Int J Digit Libr (2000) 2: 269–287

INTERNATIONAL JOURNAL ON
Digital Libraries

Visual and textual content based indexing and retrieval

Chabane Djeraba, Marinette Bouet, Henri Briand, Ali Khenchaf

IRIN, Polytechnic School of Nantes University, 2 rue de la Houssinière, BP 92208 – 44322 Nantes Cedex 3, France;
E-mail: djeraba@irin.univ-nantes.fr

Abstract. Although there are several research and commercial systems in content-based multimedia indexing and retrieval, more effort should be focused on the quality of retrieval, particularly regarding how well the retrieval results correspond to what the user really wants. In this context, we present an approach that may contribute to the quality of indexing and retrieval. It supports both textual and visual indexing and retrieval that obtains results with a higher degree of quality, and it incorporates domain knowledge into the index without any special efforts on the part of the user.

Key words: Image – Text – Indexing – Retrieval – Content – Similarity

1 Introduction

The business potential of multimedia information is growing rapidly. Typical applications include security and surveillance systems for banking, retailing, highway access control, automatic verification and identification, and so on. Suitable tools to index and access the content need to be designed to exploit this kind of information efficiently. These tools should develop new indexing and retrieval techniques that allow users to access multimedia information while maintaining a high degree of "goodness".

Historically, two generations of systems appeared successively. They have different descriptions of content – textual and visual descriptions that refer to textual and visual documents, respectively.

The first generation of image retrieval systems use non-structured (e.g., keywords, free-text) as well as structured descriptions (e.g., conceptual graphs), supplied by the authors as a basis for indexing and retrieval. The first generation is linked to the well-known concept of "information retrieval" [18] that supports the mental aspect of information description and its specifications for retrieval and whatever techniques that are employed to carry out the operations. Historically, they referred to textual documents and started, globally, in the 1950s and 1960s [28], and evolved in the following years [27, 29]. They use essentially best-match information retrieval techniques to search a ranked list of relevant documents based, partly, on user's free-text queries. From our point of view, the web engines, even if they are recent, illustrate well this generation of systems. The documents referred to may be visual (image, video), audible (audio), or textual. Figure 1 contains an example of the web retrieval engine AltaVista.

Content description is semantically powerful, because free text is one of the best ways to describe the content naturally and semantically. In spite of this important property, they do not allow queries based directly on the visual properties of the images, and are dependent on the particular vocabulary used. So, the quality of the text-based retrieval depends on the quality of the manual description. Generally, manual descriptions tend to be incomplete and inconsistent, so the text descriptions do not, generally, allow for unanticipated search in subsequent retrievals.

The second generation of systems is a natural way to overcome the problem of text-based systems. They reuse several first-generation concepts [28], such as matching, retrieval, indexing and similarity measures. However, they introduce visual image features in indexing and queries. These features are generally extracted automatically or semi-automatically from the image, and the final users formulate their queries either by supplying sample images, or by specifying a combination of visual features.

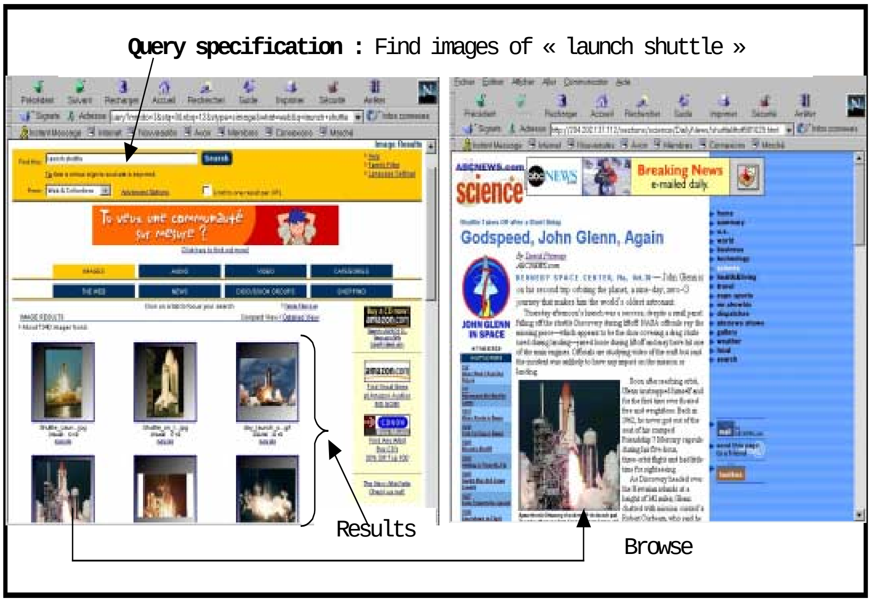


Fig. 1. The first screen contains the query "find web images characterized by the key-words launch shuttle" and a list of summarized results. The second screen contains a result display. This example illustrates the image retrieval systems based on textual indexing

The second generation of systems, started, generally, in the early 1990s. Although those systems appeared less than ten years ago, it is too difficult to compare and describe them accurately and exhaustively, because there are lots of systems and prototypes with weak test platforms. The systems of this second generation are put forward as one of the most active research domains in the multimedia field. Lots of prototypes (Visualseek [30], RECI [9], Photobook [24], SurfImage [21], and many others) that describe research results and a lesser number of industrial products (Qbic [10], Virage [17]) have been proposed. In [13] more than 40 systems have been compared. Many significant papers describe significant works. Any state of the art review [1] needs to be updated, and will certainly omit some works or prototypes. It is a real challenge to summarize, effectively, efficiently and exhaustively the state of the art that describes the present and forthcoming orientations of this formidable marathon, called the visual information system or Content-Based Multimedia Indexing and Retrieval. The domain is so active, scientifically and industrially, that a standardization process, under ISO (International Standardization Organization) supervision, has been triggered to normalize the interface description of multimedia content (images, audio, video). This process is known as (MPEG7) [19]; its aim is to normalize the description of the content in order to allow the various accesses to and extractions from the materials. In the future this standard may play an important role in comparing and testing the different retrieval systems based on "goodness".

In this context, we present a content-based visual retrieval system that enables users to formulate their queries either by supplying example images, or by specifying a combination of textual and visual features (such as colors, textures, shapes, spatial constraints). Textual attributes alone, or visual features alone, may be insufficient to describe an image correctly. Textual and visual combinations are very important to achieve good ratios of image retrieval. Textual attributes should be used to describe the secret semantic of an image. However, visual features should describe the image content such as color and shape that may be extracted automatically or semi-automatically.

However, even if the use of different descriptions has improved retrieval "goodness", it is not very powerful as these different descriptions tend to have different degrees of importance for different classes of queries. One way to resolve this problem is to consider both textual and visual descriptions with different degrees of importance (each attribute can be weighted).

The user may either give an example image to express his/her query, or specify complex queries with different degrees of importance suited to different classes of queries (a combination of several features where each feature, according to its role in the wanted image, has its own weight). So, the user may define his/her own class of images that he/she look for (i.e., introduce his/her application domain knowledge in the system), store these classes in the database, and reuse them whenever it is necessary. The storage of the application field knowledge is very important. The user may define easily and naturally, a high concept from one or several basic concepts. Moreover, the integrated knowledge is not predetermined with regard to a specific application field. It is the application domain knowledge that has an importance according to the user's experience. So, our prototype is generic and adaptable to any field. Besides, this approach overcomes the problem of incomplete knowledge inherent in most of knowledge-based systems, and enables users to organize their queries based on their perceptions of classes.

The paper is presented as follows: in Sect. 2, we present the description and the extraction process of visual and textual features. We respond to how the system extracts visual and textual features from digital images. In Sect. 3, we highlight the notion of concept-based query that combines textual and visual descriptions in indexing and retrieval, and this is the scope of our paper. In Sect. 4, we present experimental results. Finally, in Sect. 5, we highlight our principal contributions in our approach, and compared them with related works.

2 Basic feature indexing (extraction and description)

The extraction and description of image content are fundamental steps to retrieve images. The retrieval task is driven by the index that is composed of descriptors. The descriptors are attributes valued by features. The features are computed by suited extraction methods. That is why, in this section, we will answer the following questions:

- What is the content?
- How can the content be described and extracted into the database?

The content of images is composed of visual features and textual descriptions. Visual features such as color and texture are extracted automatically; color and texture concern both the whole image and the significant regions of the image. The significant regions are bordered by closed shapes. The shapes may be extracted semi-automatically, based on flood-fill, used in several photo-editing tools, and snake methods. Flood-fill methods start from a single pixel and repeatedly add adjacent pixels whose values are under a special value that depends on the application domain. Snake methods consider the user-drawn curve and automatically align it with nearby image edges. The automatic extraction of significant shapes is impossible, when the image database is independent of a specific investigation field. However, it is possible for specific fields such as marine and art images. The shapes may be extracted manually. The manual extraction of the shape takes a long time. The accuracy of the shape description added to the "good" similarity measure will certainly contribute to the "goodness" of the retrieval. The similarity measure used is "Euclidean" measure adapted to Fourier descriptors as we will see in Sect. 3. Finally, textual descriptions are inserted by the author. They are associated with the whole image and the significant regions as well.

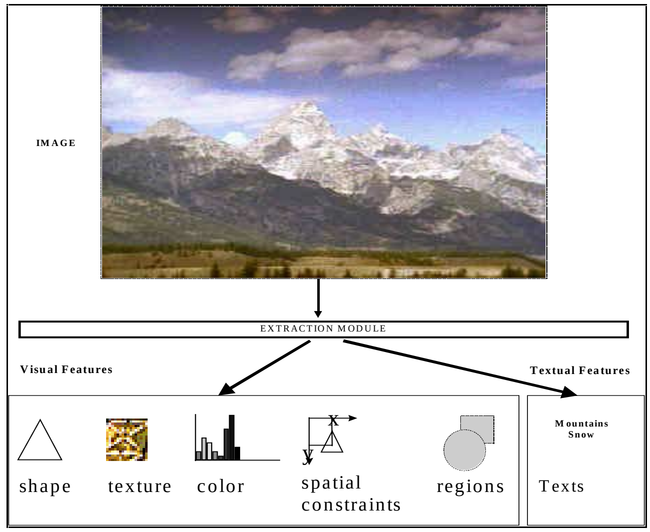


Fig. 2. Extraction and description. This figure presents the different features extracted and described in the system. Texts are extracted manually. Shapes are extracted semi-automatically. Colors, textures and spatial constraints are extracted automatically

2.1 Shape

A shape may be characterized from several points of view [25]: object description based on external (boundaries) or internal (regions) descriptions, such as elongation (Fiber Length/Fiber Width), convexity (Convex Perimeter/Perimeter), solidity (Area/Convex Area) and compactness ($\sqrt{(4/\pi)}$Area/Max. Diameter); shape reconstruction ability, such as Fourier descriptors [32], CSS [20]; mathematical (e.g., fractal, moment base invariant, Fourier transform); heuristic (e.g., elongatedness) representation; robustness of description to translation, rotation and scale transformations (e.g., Fourier transform).

The choice of implementing Fourier descriptors is based on its numerous advantages such as shape reconstruction, and robustness of description to translation, rotation, and scale transformations. Furthermore, it represents any complex shape with only a few parameters, for N harmonics, we have $2 + N * 4$ coefficients.

A shape is a closed and periodic curve $z(t) = (x(t), y(t))$ and $z(t) = z(t + T)$ and T = length of the curve, so it is decomposed in Fourier series with real coefficients on N harmonics. Thus, it is possible to characterize a shape with the first N Fourier descriptors; a Fourier descriptor being the set of coefficients a_i, b_i, c_i, d_i for the harmonic i. The first harmonic $N = 0$ gives the gravity center coordinates of the shape.

In Fig. 4, we present queries and their results that illustrate the robustness of Fourier descriptors.

Two shapes are similar even if they differ as a result of a geometric transformation such as rotation, scale, or translation. In fact, translation, rotation and scale have no effect on the module of Fourier coefficients (more or less K coefficient for scale).

$$x(t) = A_0 + \frac{1}{T}\sum_{n=1}^{N} a_n \cos\left(2\pi \frac{n}{T} t\right) + b_n \sin\left(2\pi \frac{n}{T} t\right),\ y(t) = C_0 + \frac{1}{T}\sum_{n=1}^{N} c_n \cos\left(2\pi \frac{n}{T} t\right) + d_n \sin\left(2\pi \frac{n}{T} t\right)$$

$$\text{With}\quad A_0 = \frac{1}{T}\sum_{n=1}^{N}\left(x_{n-1}\Delta t_n + \frac{\Delta x_n \Delta t_n}{2}\right),\ C_0 = \frac{1}{T}\sum_{n=1}^{N}\left(y_{n-1}\Delta t_n + \frac{\Delta y_n \Delta t_n}{2}\right),$$

$$a_n = \frac{T}{2\pi^2 k^2}\sum_{n=1}^{N}\frac{\Delta x_n}{\Delta t_n}\left(\cos\left(2\pi \frac{k}{T} t_n\right) - \cos\left(2\pi \frac{k}{T} t_{n-1}\right)\right),\ b_n = \frac{T}{2\pi^2 k^2}\sum_{n=1}^{N}\frac{\Delta x_n}{\Delta t_n}\left(\sin\left(2\pi \frac{k}{T} t_n\right) - \sin\left(2\pi \frac{k}{T} t_{n-1}\right)\right),$$

$$c_n = \frac{T}{2\pi^2 k^2}\sum_{n=1}^{N}\frac{\Delta y_n}{\Delta t_n}\left(\cos\left(2\pi \frac{k}{T} t_n\right) - \cos\left(2\pi \frac{k}{T} t_{n-1}\right)\right) \text{and } d_n = \frac{T}{2\pi^2 k^2}\sum_{n=1}^{N}\frac{\Delta y_n}{\Delta t_n}\left(\sin\left(2\pi \frac{k}{T} t_n\right) - \sin\left(2\pi \frac{k}{T} t_{n-1}\right)\right).$$

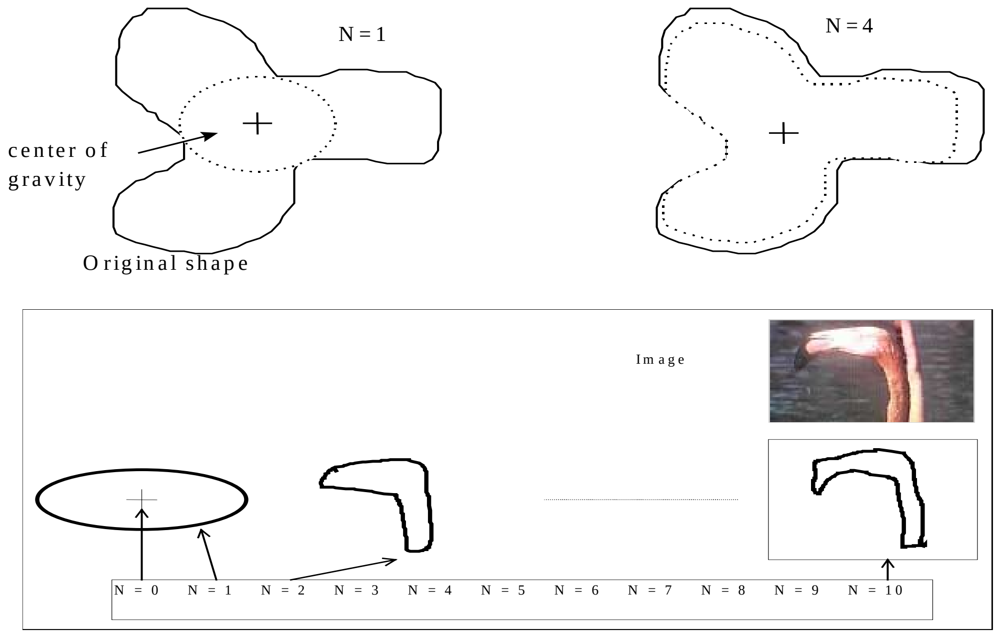


Fig. 3. Representation of a shape with Fourier descriptors. Functions $x(t)$ and $y(t)$ decomposed in Fourier series with N harmonics

2.2 Color

The color is the second important feature extracted from an image or a region.

Description. The color may be characterized in different ways, and more particularly by:

– its color distribution.
– its statistical moments: it describes the color allotment in a region or in the whole image; this is particularly interesting when a color histogram is a high-dimensional feature.
– its dominant color.

Dominant color may be deduced from the color histogram of the considered image by scanning the largest number of pixels in bins as a whole. It is defined by a three-dimensional vector, where each dimension corresponds to one component of the used color representation system (e.g., RGB). These three methods of color characterization, namely histogram, statistical moments, and dominant color, are the easiest ones in order to index a whole image or an image region. However, sometimes, they are not appropriate. There are other methods such as indexing by a binary color set, or indexing by the perimeter and angle measures. For simplification reasons, we implemented color distribution. To represent image color distribution, we use color histograms. A color histogram $H(M)$ is a vector $(h_{c_1}, h_{c_2}, h_{c_3}, ..., h_{c_n})$ where each h_{c_j} element represents the number of pixels of the color c_j (or bin) in image M. Every histogram bin represents a color in a color space (RGB, HSV, etc.).

An RGB color image may be characterized either by three histograms, or by a single color histogram. In the first case, there is a histogram for each component: red, green and blue. This means that the red histogram bins represent different red levels of red. It is the same for the green and blue histograms. In the second case, a bin represents a combination of the three RGB components. This principle may be considered in HSV color space.

A color histogram may also be represented in a three-dimensional space where the pixels of the same color are defined by a cube whose size is proportional to the number of pixels. The cube center characterizes the color in the considered color space.

A color histogram may be computed on:

– a whole image when, browsing through the database, the user gives an image as a query, which is called "queries by example". But, the major drawback of

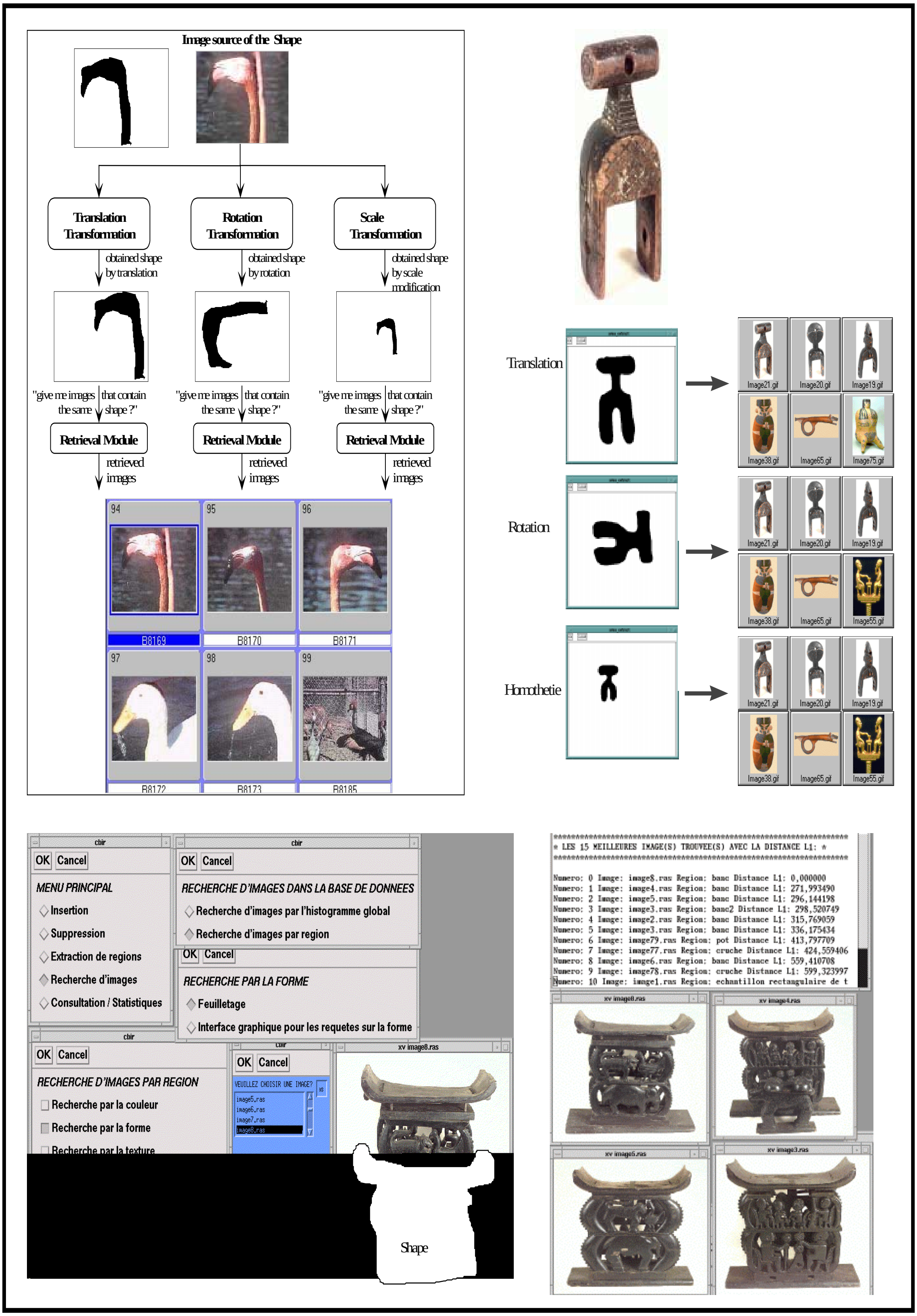


Fig. 4. Three examples of queries that highlight Fourier descriptor robustness to translation, rotation, and scale transformations

using a global histogram, is that it does not allow the detection of the spatial arrangement of image objects or image colored regions. The two following cases resolve this problem by computing a local histogram.
– an image area when the considered image is divided into several identical areas.
– an image region.

Extraction. In the first step of the extraction process, based on image format (e.g., jpeg), the region and image colors are extracted and represented in the RGB model. In the second step, based on the RGB model, the color is transformed and represented in the HSV model, characterized by three means H (Hue – the color), S (Saturation – the vividness of the color) and V (Value – the brightness of the color). The HSV model is better than the RGB model, in which certain ambiguities appear between colors (such as yellow and green). In the object-oriented database, we define two classes RGB and HSV colors. RGB instances are histograms with RGB model. An instance represents a histogram of an image or a histogram of an image region. It is created when an image is indexed, because it corresponds to a physical representation of an image. A physical image is a vector in which each pixel is coded by three bytes, for respectively red, green and blue. So, the instance of RGB class is created first, and it is converted to HSV model. This conversion implies the creation of an instance of HSV class. In the first versions of our system, we used only RGB model, in the followed versions we used HSV models. So for historical reasons, we kept the two models. Another reason for keeping RGB, is that the indexing process is faster than the indexing process with HSV, because there is no conversion step from RGB to HSV on each pixel. So the user has the possibility to index only by "RGB model", with its disadvantages, when he/she wants quick indexing.

The first class, called RGB histogram is structured by three the attributes red, green and blue and two methods. Each attribute takes as values an object structured by two attributes: the variation and the mean of the color. The two methods compute, respectively, for each color (red, blue or green) the mean and the variation of the color.

The second class, called HSV histogram, contains a 'structure' part that includes the histogram of colors and a set of distances, and a 'method' part that includes the methods that compute the attributes of the 'structure' part. The colors of regions and images are quantified, so color histograms have the same cardinal. We associate with the color descriptions "moment-degree" and "mean" that may be helpful for more accurate color query. Each element of the histogram represents the number of pixels that have the suited color. So, comparing the colors of two regions is equivalent to computing the distance between the histogram of the target region and that of the source region. Before submitting the query, the user may choose the distance.

2.3 Texture

The texture is the third important feature extracted from image regions. It is very difficult to find a generic texture representation that may be independent of the application domain. However, for pragmatic reasons, we represent two simple classes of texture that may be extended, using object technologies, to other texture classes suited to specific application domains. The class of texture considered is not a powerful texture representation, but may be interesting for the retrieval process when mixing it with other features (color, shape, keywords). In the first class, the texture of the region is represented by statistical formulas based on four moments. In object-oriented modeling, we define the class texture structured by a set of attributes that model the four statistical moments and the number of pixels of the region. The four moments are computed, respectively, by four suitable methods. In the second class, the texture of the region is represented by the histogram of the gray differences. We consider a distance as a segment between two points. The direction of the distance and the length of the distance are dependent on the application domain. Based on the histogram, the system computes the coarseness, the variance, the contrast and the directionality.

2.4 Spatial localization

The spatial localization of regions is the third important image feature extracted from an image. It locates the relations between image regions and computes the distance between two points in those regions. We consider each region in a minimal rectangle. The distance considers left and top points of each rectangle.

The gravity center of the region (x_i, y_i) is associated with the first harmonic $N = 0$ of Fourier coefficients. The gravity center is computed by means the first harmonic $(N = 0)$ of Fourier descriptors.

The image is subdivided into nine areas of identical dimensions, each one identified by an identifier. The position of a region is deduced from the position of the gravity center in one of these nine areas. Finally, the localization is stored in the database.

The way coordinates are associated with objects is primitive; we use Fourier coefficients with $N = 0$. So, the spatial localization of regions is simple. The spatial localization is an interesting feature, because it is extracted without any effort from the user. In addition, based on spatial localization, spatial relationships are extracted automatically. So, the user can ask questions in the form of "find images that contain two regions": region1 and region2 characterized, respectively, by texture1, color1 and texture2, color2, and region1 is "on the left of" region2. The relation between region1 and region2 is deduced automatically from their localization.

Furthermore, mixing this feature with texture, color and shapes will certainly give a rich description of images and queries. The spatial localization is not in the scope of the paper, it is an additional feature that complements the other features.

"Why nine areas?" This number of areas is arbitrary for the moment, and may be specified for any application fields. For example, the images of certain application fields contain very small significant regions, should have more than nine areas. However, if images are composed of one dominant region, there is no need to have nine areas, one area is enough. The most important point of this feature is the principle of image subdivision in any number of areas, and automatic extraction of the region gravity centers and spatial relations between regions.

3 Concept-based query

Images contain rich features from which we can deduce many high-level classes of images. The full semantics of an image is specified through multiple visual and textual forms. The use of either text or visual descriptions is not enough to fully describe the semantic content of images, because each form on its own has limited expressive power. For example, colors, textures, and shapes are good for capturing visual aspects of a region. However, they are not very accurate to represent real-world concepts (real world meaning regions such as person, sea, sun, etc.) present in images. Text descriptions, on the other hand, are good at describing high level classes. However, they are weak at capturing visual descriptions, and tend to be incomplete.

To overcome the representation problems of each form of description, both forms should be used together to provide a more complete description of the image contents. The use of different forms has improved retrieval effectiveness, but it is not very powerful as different forms tend to have different degrees of importance for different classes of queries.

One way to resolve this problem is to use different specification forms, textual description and visual content, with different degrees of importance. The use of different forms with different degrees of importance is based on the human perception which filters and discriminates suitable forms of descriptions when selecting certain kinds of images. One typically starts with a stereotyped model of the concept that one is looking for (car, bus, plane, flower, sea, sun, etc.). There should be multiple representations of the same concept. The stereotyped model contains information that can be used to identify the concept in the image in terms of other sub-concepts and image content attributes. This is similar to the idea of patterns. The degree of similarity between the content of an image and the concept representation is used to compute the relevance of the image.

For example, if we look for images of canary birds, we will look for, in order of importance, certain specific shapes, colors, and textures that characterize a canary bird, together with typical text words used to describe the canary bird. At a higher level, if we look for images of canary birds in a park, we may start with a mental image of a large green field consisting of trees, possibly some flowers, a pond, and birds. Thus, the stereotyped model of the concept birds in a park is composed of a number of primitive sub-concepts. Each primitive concept is in turn defined in terms of a set of image content attributes. With this representation, we can successively build up a hierarchical model of real-world concepts together with clues for their detection in terms of colors, textures, and text descriptions at a primitive level.

3.1 Description

An image contains many different abstractions. It is impossible to define a common knowledge base shared by all users. The aim of our approach is to provide a mechanism for the users to incorporate domain knowledge into their queries. This approach overcomes the problem of incomplete knowledge inherent in most knowledge-based systems, and also allows the users to organize their queries based on their perceptions of the concept.

We represent the query as a composite hierarchy of abstractions, except that in our case, the leaf concepts are defined with different features including text, colors, textures, shapes, and spatial localization. Each abstraction is described by its name and its relationships with other concepts.

The figure shows the composite description of the concept "Shuttle-Flight-Launch". Here the concept "Shuttle-Flight-Launch" is defined by two attributes, textual and visual descriptions. The visual description is defined by the color and shape attributes. At the leaf level of the concept composition, the text attribute is defined as a composition of text strings "Shuttle", "Launch" and "Flight". The color attribute is defined as a composition of orange color, while the shape attribute is defined to be of the "Shuttle" type. The shuttle color and shape are specified by the user with a visual tool.

In fuzzy query specification, it is important to describe the similarity-based matching that denotes the best matching, rather than the exact matching, to deduce the degree of similarity between the target and the source images. So, we use a Composition-of relationship. In our example, the Composition-of relationship is used to describe the composition of content attribute in terms of a weight of 40% for region1 description and a weight of 60% for region2 description. Region1 is composed of 80% of orange color and region2 of the shuttle shape. It means that relevant images should preferably contain the desired content composition. It does not require the features to be present in the relevant images as in And-relationships. It is also insufficient for a relevant image to contain just 80% of orange color as expressed by the Or-relationship. And- and Or-relationships are relationships that, respectively, deduce the conjunction and the disjunction of the specified descriptions. These relationships are generally used in Boolean "exact matching".

The composition relationship is also used at a higher level to combine visual and textual descriptions. In the "Shuttle-flight-Launch" concept, the value of "Shuttle-flight-Launch" is deduced by combining the weighted evidence of the text and visual descriptions.

The user may associate weight to an attribute or a concept to denote its importance in inferring the higher-level concept. With reference to the "Shuttle-Flight-Launch" concept, weights of 60% and 40% are associated with the textual and the visual descriptions, respectively. The weight denotes the degree of importance judged by the user, based upon his/her experience. The weights will influence the combination of descriptions and ultimately the ranking of the images retrieved. The weight is a real number in the interval [0,1]. If the weight is omitted for an attribute, it is assumed to be 1 (i.e., big importance). The sum of all weights associated with a relationship should be less than or equal to 1. In Fig. 5, we present the high level specification of the user concept "Shuttle-Flight-Launch".

We can reuse concepts to construct higher-level concepts. An example of higher level concept is the "Shuttle Flight Launch" in a daylight (Fig. 6). This concept, "Shuttle-Flight-Launch-Daylight", is composed of two sub-concepts, namely "Shuttle-Flight-Launch" and "Daylight". In a similar manner, queries expressing composite concepts could also be constructed.

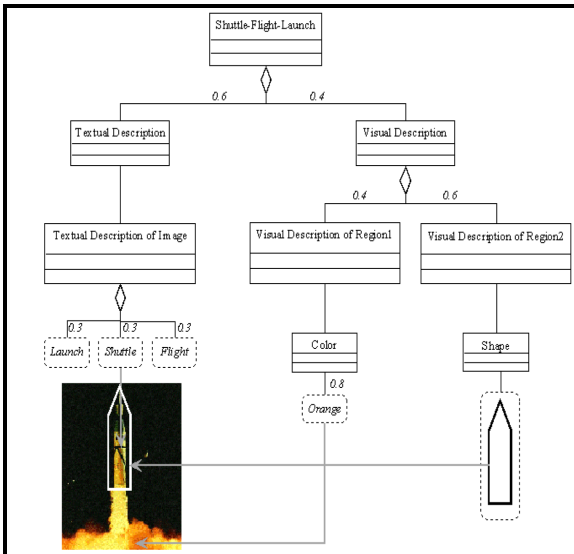
3.2 Query processing

If the query specification is syntactically and semantically correct, the system constructs the query concept in order to match it with the content of images stored in the database. The matching consists in computing the similarity distance between the concept specified in the query and the database image content. The returned images are then sorted according to distances.

The similarity computation starts from the leaf nodes (e.g., "Launch" "Shuttle" "Flight" "Color" "Shape"). For each leaf attribute value that describes a certain feature, we compute its similarity, using a suitable distance, with the same features of database images. For color and texture, we have implemented a set of similarity distances: histogram distance L1, Euclidean distance L2, histogram intersection distance, moment distance L1, and quadratic distance. These distances have been chosen because they illustrate the advantages and disadvantages of the current approaches. Histogram distance L1 and Euclidean distance L2 are the simplest ones, histogram intersection distance, moment distance L1, and quadratic distance are generally more efficient when the result images are voluminous, but they are more complex. For example, quadratic distance is more powerful, but it is time-consuming compared to the first two distances. For pragmatic reasons, before triggering the retrieval process, the distance used is selected by the user out of the set of distances.

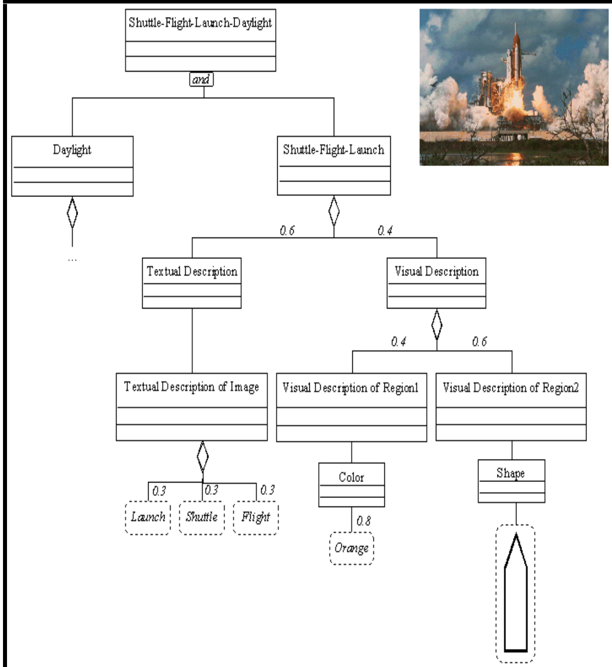
We limit our presentation to quadratic distance. We consider H and I histograms of, respectively, source image (query image) and target image (image database). They are defined by an array (vector) $(h_{c_1}, h_{c_2}, h_{c_3}, \ldots, h_{c_n})$, in which each element h_{c_j} represents the number of pixels of color c_j in the image. N is the number of pixels in the image. The distance between two histograms of colors H and I is represented by:

$$d_Q(H, I) = \sqrt{(H - I).A.(H - I)^T}\,,$$



(***Concept*** Shuttle-Flight-Launch)
(***Description*** Shuttle-Flight-Launch "It came in for a perfect landing at the Kennedy Space Center at 6:47 a.m. EDT on Thursday, July 17, 1997, following a virtually perfect scientific mission that resulted in many discoveries that will be important both on earth and for the International Space Station")
(***Attribute*** Shuttle-Flight-Launch
(***Composition-of***
(***TextualDescription***, 0.60,
(***Composition-of***
(Image, 1,
(***Composition-of***
("Shuttle", 0.30),
("Flight", 0.30),
("Launch", 0.30))))),
(***VisualDescription***, 0.40,
(***Composition-of***
(Region1, 0.40,
(***Composition-of***
(Color, 1,
(***Composition-of***
(, 1, 0.80))))),
(Region2, 0.60,
(***Composition-of***
(Shape, 1,
(***Composition-of***
(, 1)))))))).

Fig. 5. "Shuttle-Flight-Launch" concept: composition hierarchy. "Shuttle-Flight-Launch" concept: specification




```
(Concept Shuttle-Flight-Launch-Daylight)
(Description Shuttle-Flight-Launch-Daylight "……")
(Attribute Shuttle-Flight-Launch-Daylight
    (And
        (Concept Shuttle-Flight-Launch)
        (Description Shuttle-Flight-Launch "It .....")
        (Attribute Shuttle-Flight-Launch
            (Composition-of
                (TextualDescription, 0.60,
                    (Composition-of
                        (Image, 1,
                            (Composition-of
                                ("Shuttle", 0.30),
                                ("Flight", 0.30),
                                ("Launch", 0.30))))),
                (VisualDescription, 0.40,
                    (Composition-of
                        (Region1, 1,
                            (Composition-of
                                (Color, 1,
                                    (Composition-of
                                        ( , 1, 0.80))))),
                        (Region2, 0.60,
                            (Composition-of
                                (Shape, 1,
                                    (Composition-of
                                        ( , 1))))))))),
        (Concept Daylight)
        (Description Daylight ".....")
        (Attribute Daylight
            (Composition-of
                (TextualDescription, 0.50,
                    (Composition-of
                        (Image, 1,
                            (Composition-of
                                ("Daylight", 1))))),
                (VisualDescription, 0.50,
                    (Composition-of
                        (Image, 1,
                            (Composition-of
                                (Color, 1,
                                    (Or
                                        ( , 0.70),
                                        ( , 0.30)))))))))).
```

Fig. 6. "Shuttle-Flight-Launch-Daylight" concept: composition hierarchy. "Shuttle-Flight-Launch-Daylight" concept: specification

where H and I are, respectively, the color histograms of M and N images. A is similarity matrix $(n \times n)$. Quadratic distance may, also, be described in the following form:

$$d_Q^2(H, I) = \sum_{p=1}^{n} \sum_{q=1}^{n} a_{pq}(h_p - i_p)(h_q - i_q) .$$

This distance is the best one because it correctly appreciates color similarity, but it is time-consuming compared to other distances (e.g., L1).

As for the shape, first, the source shape undergoes a transcription into the model: Freeman code followed by 11 quadruplets of Fourier descriptors (a_n, b_n, c_n, d_n) where n varies from 1 to 11. Second, the matching process between the Fourier descriptors of z' shape of the image database and the Fourier descriptors of z shape of the considered query, is triggered by computing the distance between z and z', namely:

$$d(z, z') = \sqrt{\sum_{n=1}^{P} \left(|Z'_n| - \hat{K}.|Z_n| \right)^2} .$$

For z and z' shapes, we have a positive constant K, and for any $n \neq 0 : |Z'_n| = K.|Z_n|$, where:

$$|Z_n| = \sqrt{|X_n|^2 + |Y_n|^2} = \sqrt{a_n^2 + b_n^2 + c_n^2 + d_n^2} .$$

That is to say, the shapes are identical but for one geometric transformation. The translation, scale, and rotation have no effect on the module of Fourier coefficients. $\hat{K} = \frac{1}{P} \sum_{n=1}^{P} \frac{|Z'_n|}{|Z_n|}$ is an estimation of K which minimizes the error on the first P (e.g., 11) coefficients of Fourier.

Research by spatial constraints, and more particularly by region localization in the image, follows two options. In the first one, the user selects a localization among nine possibilities (middle, right, left, top, bottom, top and right, top and left, bottom and right, and bottom and left). In the second one, the user selects a region of an image, and the system considers the localization of that region in the query.

In spatial localization, there is no similarity measure. The system considers only the images that respect the localization specified in the query. This can be seen as "exact matching"

For the text, the distance is equal to 1 if the words matched are the same or synonymous, and 0 otherwise, but we may use one of the various vector space information retrieval models. The distances computed at the leaves are propagated up the composite concept.

In the Composition-of node, the propagated value is deduced by computing the vector sum distance values of the component distances.

$$\text{Distance\_Composition-of} = \sum_{i=1}^{n} (\text{Weight}_i * \text{Distance}_i)\,,$$

where Distance_i and Weight_i represent respectively the similarity and the weight associated with the component node i. If we use the Or-node, the propagated value is deduced by selecting the maximum distance value of the component nodes,

$$\text{Distance\_OR} = \text{Max}(\text{Distance}_1, \text{Distance}_2, \ldots, \text{Distance}_n)\,,$$

where distance_i is the distance value associated with component node_i. However, if we use the And-node, the propagated value is deduced by selecting the minimum distance value of the component nodes,

$$\text{Distance\_AND} = \text{Min}(\text{Distance}_1, \text{Distance}_2, \ldots, \text{Distance}_n)\,,$$

where Distance_i is the value associated with component node i.

The figure illustrates how distances are propagated from the basic features to the composite feature. We assume that the arcs without weights are equal to 1. If we have an image in which the similarity values are equal to 1, 1, 0.5, and 0.8 obtained from text, color, and shape features, respectively, the final weight obtained for the composite concept is 0.62. It corresponds to the distance

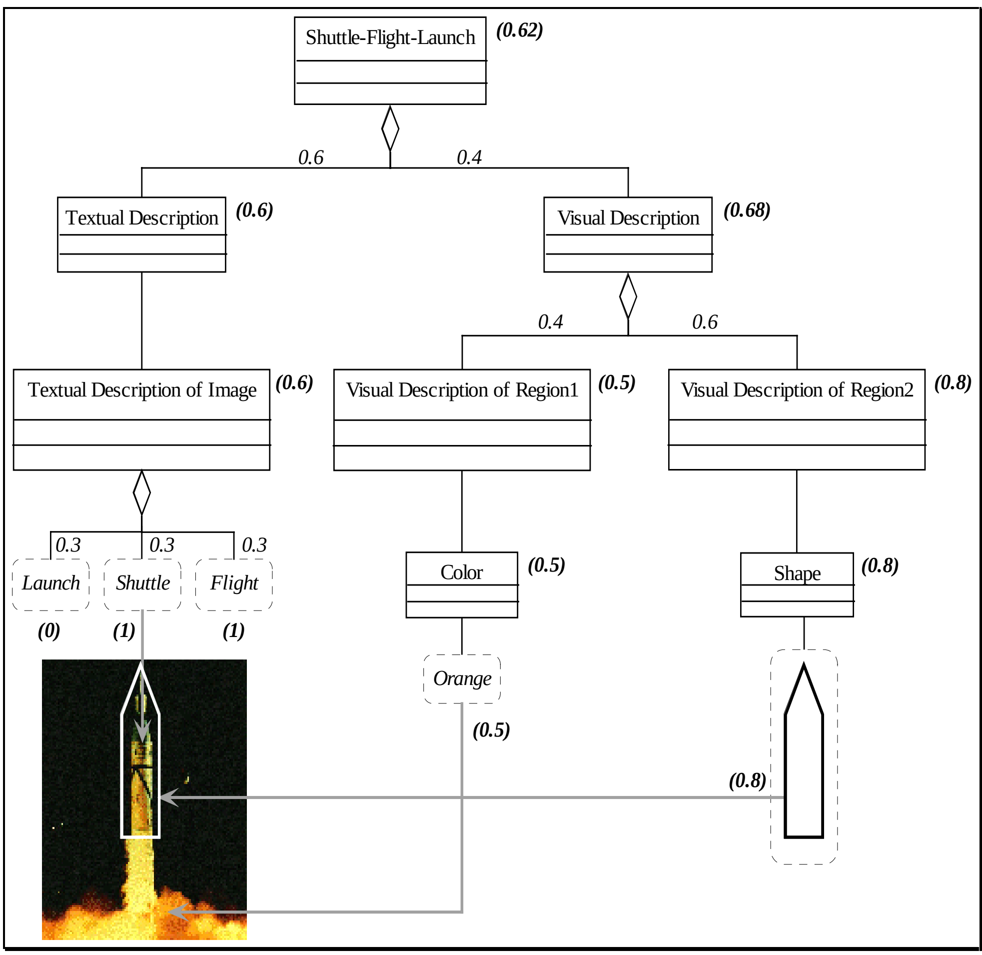


Fig. 7. Similarity propagation

between the query concept and the target image of the database.

3.3 Knowledge integration

The result of the integration takes the form of classes. Each query is considered as a potential class. The positive results of the query become instances of the potential class. So, when the user validates the concept supported by the class, the system creates the class and associates the instances (positive results) with the class. The creation of the class is dynamic. This notion revolves around the notion of scheme evolution, that is an important notion of the object-oriented database. So, the different classes, validated by the users, correspond to queries that correspond to real world concepts. The creation of a class is possible with the object-oriented model that implements our prototype. The creation, deletion, and modification of the classes are class methods defined in the meta-class level of the object-oriented model. For different queries, the system creates different classes. However, the queries that reuse previous classes, imply the creation of sub-classes. We impose a minimum number of instances in the classes. The system refuses the creation of classes that have too few numbers of instances. In our case, we consider that each created class should contain at least 0.1% of the total number of images in the database. For example, for our data set (30 000 images), each created class is validated when the cardinality of the class is greater than or equal to 30 images. The structure of a class is composed of the visual features (shape, color, texture) and the textual description of the query. It also contains the weight value of each feature.

3.4 Query check

Users continue to prefer highly-customized user interfaces that are very closely linked to specific application fields. However, limited usability testing was performed and when given enough training, users were able to use query language for simple database tasks. The only current operating alternative continues to be text, mainframe, thus making any direct comparisons somewhat awkward to make. Subjectively, however, the visual, highly-graphical approach was preferred over the above alternatives. That is why we developed a query visual interface and a functional representation interface. The visual interface gives users the possibility to specify their query by selecting color, texture, shapes, constraints localization, and key words with different weights. It also gives them the possibility to specify a query by example, such as "find images that are similar to the example image". Then, the system automatically generates a functional representation.

In case the user specifies a query that is very difficult to specify with the visual interface, he/she may specify it directly in the functional representation interface. A parser has been designed to eliminate syntactic errors such as color without reference or missing bracket, etc., and semantic errors, such as the absence of color or shape reference, or the sum of weights greater than 100%, etc. The user query specification, based on the functional representation interface, is not easy, however, it may be interesting in certain cases in which the user needs to specify a very complex query. In this case the visual interface may be insufficient. We believe, that generally, the visual interface will be the most frequently used. When the query is syntactically or semantically incorrect, the query is refused. In other words, queries can be constructed with a functional representation interface, or they can be generated dynamically within an example or within vision application using the user's interface. Queries are classes and their positive results are instances. When validated, query classes and instances become persistent and reusable. Users may keep libraries of query templates, which can be built incrementally, tested separately, cloned and linked together to form more complex queries.

3.5 Query by examples

The following examples (Figs. 8–10) illustrate "query by examples" that are based on a combination of the features presented previously. Query by examples specifies a query that means "find images that are similar to those specified". The query may be composed of several images. Several images match the "goodness" of retrieval. The following examples illustrate well the kind of results we can obtain when we use, on the one hand, visual queries and, on the other, visual and textual queries.

In the following example (Fig. 8), the query based on visual features. The query is: "find images that are similar to the image presented in the example". In visual description there are 50% for texture, 50% for color, 0%, for shape and 0% for spatial constraints. There is no specification of texts, so the system will scan all images of the database without considering the textual annotations. As results, we obtain a few images of waterfalls, because there are lots of images that are visually similar to the example, but they do not contain a waterfall. There are ten positive images.

To match the description of the waterfall, several examples are presented (e.g., four). This property makes possible the refinement of retrieval based on the feedback (results of previous queries). The results obtained tend to have more "goodness". As previously, the visual descriptions are specified as follow: 50% for texture, 50% for color, 0% for shape, and 0% for spatial constraints (Fig. 9).

We "may" obtain better results when combining visual, as done previously, and textual descriptions with, respectively, 50% visual description and 50% textual description. In textual description, the query specifies the keyword "waterfall", and the visual description contains the visual features. In the visual description there are

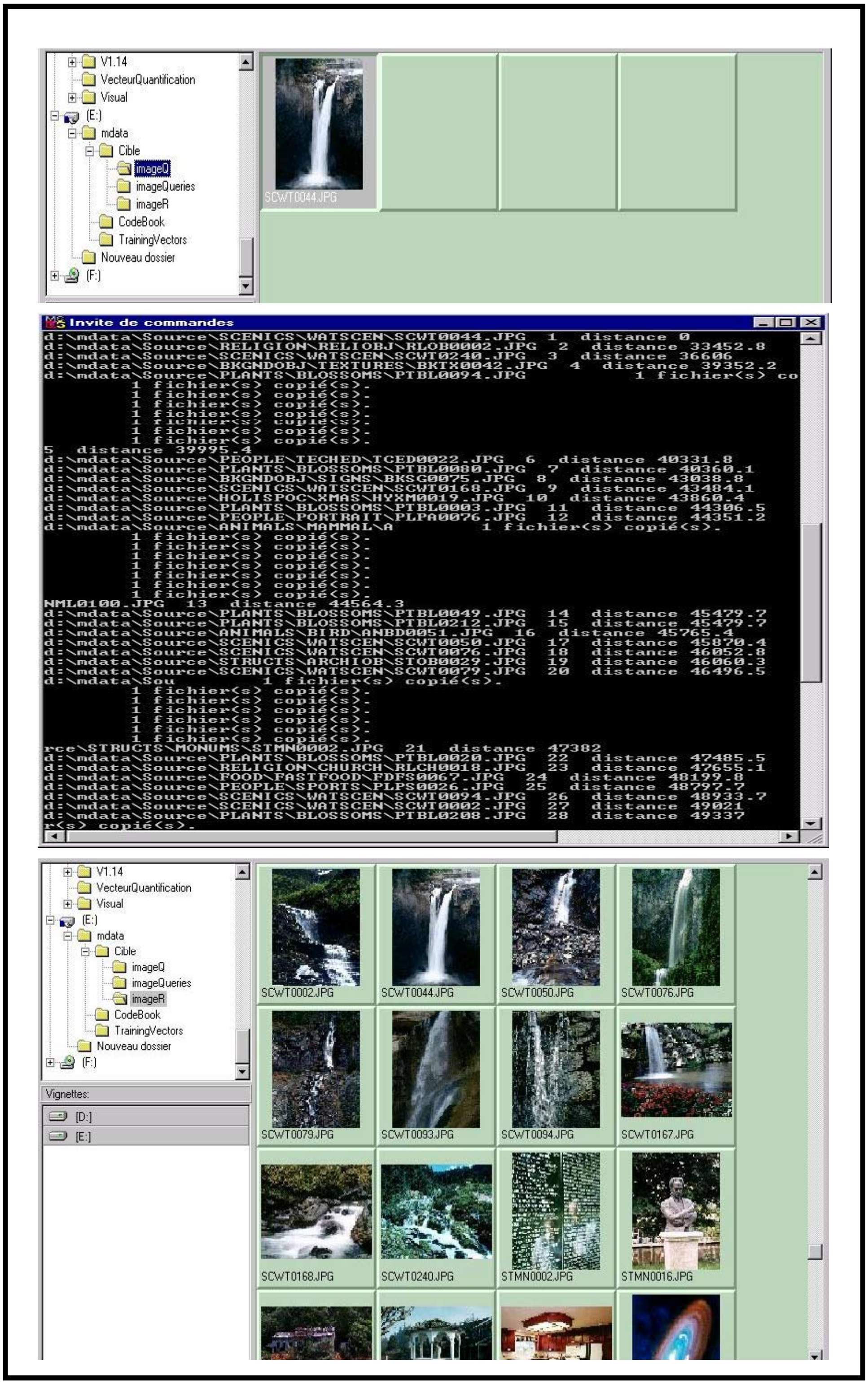


Fig. 8. Find images that are similar to the example: "waterfall"

50% for texture, 50% for color, 0% for shape, and 0% for spatial constraints. The results are better because the annotation "waterfall" exists in the dictionary (Fig. 10).

So the "goodness" is dependent on the particular vocabulary used, in our case "waterfall". If the manual descriptions were incomplete, in our case there was no "waterfall" in the dictionary, then we would be in the presence of unanticipated search in subsequent retrievals.

4 Experimental results

We have conducted extensive experiments of varied data sets to measure the performance of the concept-based query. First, we present the metrics used to measure retrieval performance followed by a description of the data sets. Finally, we present the results along with our observations.

4.1 Evaluation method

The retrieval system can be evaluated by considering its capacity to effectively retrieve information relevant to a user. It is called the retrieval goodness. Retrieval goodness is measured by recall and precision metrics [27, 29]. For a given query and a given number of images retrieved, recall gives the ratio between the number of relevant images retrieved and the total number of relevant images in the collection considered. Precision gives the ratio between the number of relevant images retrieved and the

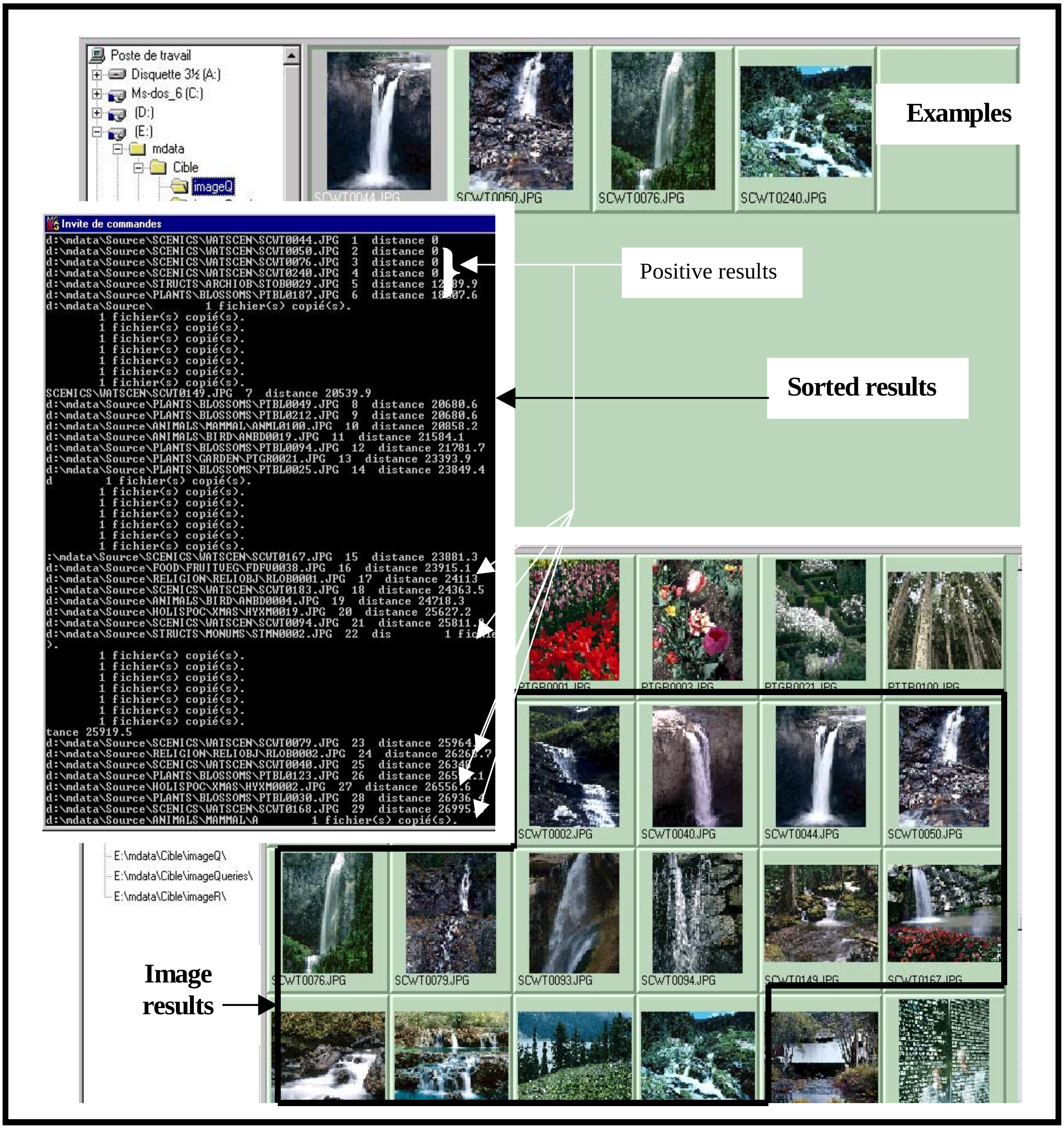


Fig. 9. "Find images that contain waterfalls"

Fig. 10. "Find images that are visually similar to the example and described by the keyword waterfalls"

number of retrieved images.

$$\text{Precision} = |\text{relevant} \cap \text{results}| / |\text{results}|$$
$$\text{Recall} = |\text{relevant} \cap \text{results}| / |\text{relevants}|$$

Recall and precision values for a system can be represented in a recall and precision graph [26], where the precision of the system is plotted as a function of the recall. This representation allows us, for instance, to measure the precision at different recall points.

4.2 Data sets

We have conducted experiments on a data set which is a collection of images covering a wide range of categories including animals, panorama, archive, flowers, scenery, people, nature, etc. This collection contains around 30 000 images. All images are catalogued into broad categories and each image carries an associated description. In this case, manually separating the collection into relevant and non-relevant sets was infeasible due to the size. It is not simple to obtain the exact number of

Table 1. Experimental results

Queries	Concept-based queries		Non-concept-based queries	
	Precision	Recall	Precision	Recall
Query 1	80%	59%	30%	70%
Query 2	65%	68%	60%	55%
Query 3	55%	70%	40%	65%
Query 4	65%	65%	50%	57%
Query 5	75%	64%	80%	42%
Query 6	75%	60%	75%	50%
Query 7	80%	57%	70%	53%
....				
Query 98	85%	55%	40%	62%
Query 99	87%	50%	55%	56%
Average	80%	85%	55%	70%

positive results in the large image database. Instead, we estimate the cardinality of the result set of each query. The pre-classification of images in semantic categories such as panorama, flowers, etc., simplifies our estimation process. For certain images, we pre-determine, by hand and by exhaustive browsing, a set of relevant image results. For other images, we start the search by issuing the query involving textual descriptions or visual features, to obtain the first set of images results. The set is refined and expanded by using query by example and exhaustive browsing.

4.3 Results and analysis

The recall and precision graphs for our system are computed as follows. One hundred reference ("query") images are selected from a test collection. A sub-set of images is selected per category (waterfalls, fires, panorama, etc.). For each image, a concept-based query is formulated. For an image reference, we associate a concept-based query that mixes visual features and textual descriptions (color + text, shape + text, texture + text, spatial-localization + text, color + texture + text, color + texture + shape + text). We also associate a non-concept-based query that uses only text descriptions or visual descriptions such as shapes, colors or textures, but not both. In certain reference images, we use only colors; in others shapes and colors; and in others textures and colors.

To demonstrate the goodness of the concept-based queries, the results of the concept-based queries are compared with the results of queries that do not use the concept-based queries. The queries that do not use knowledge combine either visual features (color, texture, spatial localization) or text, but not both. Since it is not possible to retrieve all relevant images, our experiment evaluates only the first 100 ranked images.

Regarding the table that summarizes some results, it is obvious that the use of domain knowledge leads to improvements in both precision and recall over majority queries tested. The average improvements of concept-based queries over non-concept-based queries are 25% for precision and 15% for recall. The resulting recall and precision graph is shown in Fig. 11.

The graph shows that, over all, precision and recall are better for concept-based queries (queries that mix visual features and textual descriptions with different degrees of importance) than for queries that use only visual features such as color or shapes or textures or textual descriptions, but not both.

We interpret these graphs as follows: a higher curve is better than a lower curve. This would mean that, for most query points, the precision points are better than recall points. The graph represented is globally monotonic. However, we think that the graph may not be monotonic for a different sample of queries.

We observe that for a sub-set of non-concept-based queries, the precision is better than the precision of concept-based queries. A possible explanation for this is that the weight of features used in concept-based queries is too restrictive. The weight of features may perturb the goodness even if it is determined subjectively.

The general principle of "the larger the retrieved set, the higher the recall, and the lower the precision" is observed in Fig. 12.

5 Contributions and comparison with other approaches

The contributions of our approach revolve around the simultaneous presence of the following nine points:

1. From visual points of view, we implement shape, color, texture, and spatial localization features. Several current systems do not implement all these features together. The color is the most commonly one [22]. Certain approaches use color pairs to model the distinct boundaries and relationships between objects in an image, in order to eliminate the contributions of the image background and to perform semantic object retrieval. Others consider the use of segmented im-

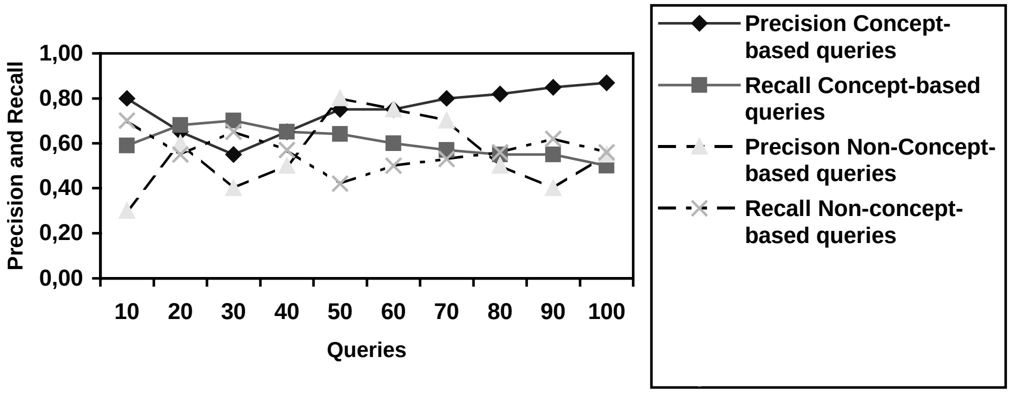


Fig. 11. The recall and precision graphs of the retrieval system. The following graphs are compared: *Solid line*: retrieval of concept-based queries. *Dotted line*: retrieval of non-concept-based queries. In general, *solid lines* are above *dotted lines* in terms of recall and precision

ages in retrieval, coupled with the use of perceptually similar colors, substantial improvements in retrieval effectiveness have been obtained. In addition, others develop a color-spatial model, which encodes image contents as a set of dominant colors, along with their spatial distributions in the image. Measures and tests have proved that a retrieval system that combines at least colors and textures could achieve a retrieval effectiveness of up to 60% [3] in recall and precision.

2. In our approach, we combine visual (shape, color, texture, spatial constraints) with textual description in queries and indexing, and we associate different degrees of importance with the different features, whether they be visual or textual. Although several systems for image retrieval exist, very few of them mix visual and textual features with different degrees of relevance in order to index and retrieve images. The CHABOT system, for example, uses simple color and text analysis strategies for image retrieval [22], but there is no representation of texture, shape, spatial localization features. Visual information retrieval systems, such as SurfImage [21], VisualSeek [30], MARS [23], are generally limited to visual features that are usually extracted automatically. Classical information retrieval systems, such as [16], have developed techniques based on semantic description of the content limited to textual annotations. Content-based queries are proposed in image databases [23]. It describes how content-based retrieval techniques applied to textual documents in the area of automatic information retrieval, can be adapted for ranked retrieval in image databases. In particular, it discusses the ranking and retrieval algorithms developed in the system based on the Boolean retrieval model. However, there is no mixing of textual and visual features in indexing and retrieval. [16] develops an approach that consists of a video text data model based on free text annotations associated with logical video segments and a corresponding query language. The model allows free text annotations on logical video segments, as well as user queries based on the temporal and interval relations between annotated logical video segments. The video indexing and query processing are based on information retrieval techniques, and query results can be ranked according to their relevance to the semantic content of the video data. However, there are no visual descriptions of the content, the indexing is restricted to manual annotations. More generally, many operational systems are based on textual annotations, called a thesaurus. A system that offers support for both alphanumeric data attached to images, and content-based query utilizing image examples is proposed in [31]. Content-based retrieval, specifically Query by Image Example, is made possible by the self-organizing hierarchical optimal subspace and learning framework for object recogni-

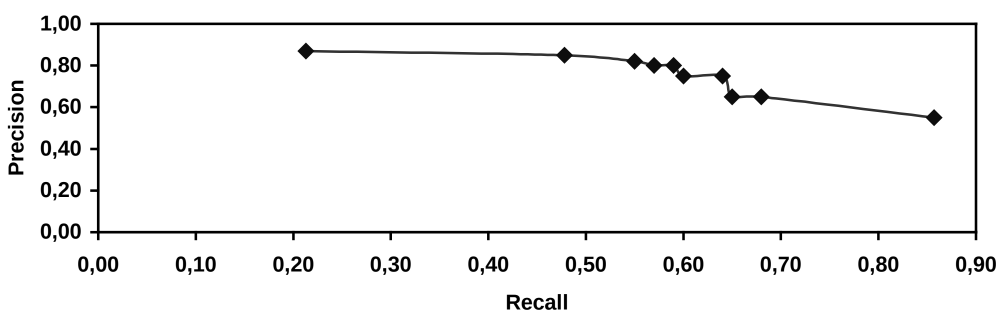


Fig. 12. Precision at recall levels

tion. The system analyses all images in the database and builds a hierarchical structure for efficient search. It uses the theories of optimal linear projection to generate the hierarchical tessellation of a space defined by the training images. This space is generated using the Discriminant Karhunen-Loève projection, where two sets of features are generated. The most expressive features are useful for characterizing a collection of images that belong to the same class. The most discriminating features are useful for determining to which class a particular image belongs. The system has been designed and experimented for face recognition. So, the visual content description is specific to face recognition. The mixing strategy between visual and textual features is based on "conjunctive" strategy without any possibility for the user to specify different degrees of importance to different features. This point is the direct result of the specific application domain: face recognition. Finally, there is no strategy to manage the user's knowledge based on real-world concepts.

3. The approach presented in our work is generic and independent of a specific application domain. It may be adaptable to any specific application domain, and there is no need to reconsider all the whole system when we apply it in a specific application field. In other systems such as in MACS, they allow querying and integration of image data and heterogeneous data sources including image and video [2]. The MACS system includes accessing images annotated textually and with captions, as well as accessing face recognition algorithms to manipulate a face database. The system is specific to face recognition domain, so visual features are focused on this specific domain of application.
4. The concepts that we look for (car, bus, plane, flower, sea, sun, etc.) are stereotyped. These concepts are implemented by classes, in terms of object paradigms, and the positive results of the queries are instances of the classes.
5. The approach supports different concept compositions to obtain more complex concepts (sea concept + plane concept = plane on the sea concept). The "plane on the sea" concept is composed of two sub-concepts "sea" and "plane". This composition may have different degrees of importance, so the concept "sea" may have more importance than the concept "plane". The degree of similarity between the content of an image and the concept representation is used to compute the relevance of the image.
6. The approach implements the building of hierarchical composition of real world concepts. The stereotyped model of the concept "plane on the sea at sunrise" is composed of a number of concepts ("sea at sunrise", "plane"). The concept of the "sea at sunrise" is composed of two primitive concepts ("sea" and "sunrise"). "plane" is itself a primitive concept. The primitive concepts are in turn defined in terms of a set of image content features (shape, texture, color, text). With this representation, we can successively build up a hierarchical model of real-world concepts together with clues for their detection in terms of colors, textures, and text descriptions at a primitive level. The building of hierarchical composition of real-world concepts integrates knowledge in the database, and allows complex user query specifications.
7. The approach implements a powerful shape representation. QBIC [10] uses global shape features such as area and circularity to retrieve similarity shape objects. However, it has a limited precision of the global shape representation. Thus, we use a mathematical model which is one of the best: the Fourier model [32]. The Fourier model has very interesting advantages: the shape can be reconstructed from the descriptors; it has a mathematical description rather than a heuristic one; and finally, the model supports the robustness of description to translation, rotation, and scale transformations. An important contribution of our representation is our extension of Fourier model shape description. This extension considers the matching process. In this extension, we consider the shape $z(t)$ composed of two signals: $x(t)$ and $y(t)$. So, $z(t) = (x(t), y(t))$. $x(t)$ represents the different values of x, and $y(t)$ represents the different values of y. $y = \text{shape}(x)$. Shape is the function in which we have the coordinates of the shape points. t indicates the different indices of the signal $z.t = 0, T-1$. T is the period of the signal, and $T =$ number of x values and y values. That is why we have two suites of coefficients $S(an, bn)$ and $S(cn, dn)$ that represent the Fourier coefficients of $x(t)$ and $y(t)$, respectively. In this extension, we modify the similarity measures (Euclidean distance) in order to consider the coefficients of the two signals $x(t)$ and $y(t)$.
8. As presented in the introduction, retrieving images by content has been studied recently [9, 10, 17, 30], and matching similar features remains a difficult problem. Therefore, it is essential to use a cooperative answering technique [5] to provide approximate matching of the features. Previously proposed cooperative database systems used the rule-based approach [6], and are therefore not scalable. In our approach, knowledge is structurally represented at different levels of semantics by the concept hierarchy and can be automatically generated from user queries. Our knowledge acquisition uses the technique of unsupervised conceptual clustering [5]. In order to provide a powerful textual and visual retrieval of image features, we have associated this knowledge representation with visual and textual features.
9. Current research in image databases tackles individual components of the overall image database problem. Models exist for representing images, and query languages for retrieving images by content have also

been developed like in QBIC and Virage [17]. Modeling and query languages in conventional alphanumeric databases have also been developed [16]. However, to support image database requirements in our system, we need to support combined textual and image requirements. Our visual query language supports knowledge-based concept query answering. Other visual query languages such as [7] can only support a subset of the requirements for the image query language provided by our approach. Many existing works also model and query image information [5, 24]. As in [5], concept-based query is used as the formalism in our work to combine the visual language and its processing. Due to the fact that our functional language is convertible to Object Query Language as recommended by ODMG, our processing language may be more directly compatible with commercially available database engines than in [4, 24].

Finally, the work reported in this paper is being extended in an important direction that mixes two distinct historical features, textual and visual features, in the same real-world concept. These two kinds of features belong to two historical classes of systems, respectively, visual information systems triggered by William Grosky [12], and information retrieval initiated by Moores [18], Salton, and others. Our combination considers the two kinds of features in a description scheme. The representation of the knowledge of the domain is incorporated in the database in the form of concepts implemented by classes. To limit the explosion of the concepts, the system limits the number of created classes. When the maximal number is reached, the class that contains the smallest number of instances is destroyed.

6 Conclusion

We have presented a visual and textual content-based retrieval system. The system has two major components: extraction and queries. We combine manual, semi-automatic, and automatic approaches in order to extract image region features. In queries, the user may use both visual and textual features with different degrees of importance in order to specify the full semantics of the required images through both visual and textual forms.

A query-based abstraction is introduced to stereotype image model descriptions. A concept is organized in a composite hierarchy of sub-concepts. The primitive concept is defined in terms of a set of image content attributes text, color, texture, shape, and localization. The matching between the source and the target concept computes the similarity distance, and propagates it by computing the vector sum distance value through the concept composite hierarchy. Users specify the query as a composite concept incorporating application knowledge. A concept may be a basic concept (color, texture, shape, spatial localization, or keywords) or a high-level concept. The leaf nodes of the concept provide basic features of the image contents. These basic features include text, color, shape, and constraint localization specifications. The weights computed and propagated up to the root concept correspond to the degree of similarity between target and source images.

The user may either give an example image to express his/her query, or specify complex queries with different degrees of importance suited to different classes of queries (a combination of several features where each feature, according to its role in the wanted image, has its own weight). So, the user may define his/her own concepts, (i.e., introduce his/her application domain knowledge in the system), store these concepts in the database and reuse them whenever it is necessary. The storage of the application field knowledge is very important. The user may define easily and naturally, a high concept from one or several basic concepts. Moreover, the integrated knowledge is not predetermined with regard to a specific application field. It is the application domain knowledge that has an importance according to user's experience. So, our prototype is generic and adaptable to any field. Besides, this approach overcomes the problem of incomplete knowledge inherent in most of knowledge-based systems, and enables users to organize their queries based on their perceptions of concepts.

References

1. Aigrain, P., Zhang, H.-J., Petkovic, D.: Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review. Multimedia Tools and Applications. An International Journal 3(3): 179–202, November 1996
2. Brink, A., Marcus, S., Subrahmanian, V.S.: Heterogeneous Multimedia reasoning. IEEE Computer 28(9): 33–39, 1995
3. Chua, S.K.: Content-based Image retrieval. MSc Dissertation. National university of Singapore 1994
4. Chua, T., Teo, K., Ooi, B., Tan, K.: Using domain knowledge in querying image databases. Multimedia Modeling. Towards the information Superhighway. France, Toulouse, 12–15 November 1996
5. Chu, W.W., Chiang, K., Hsu, C.C., Yau, H.: An error based conceptual clustering method for providing approximate query answer. Comm. ACM, Virtual Extension Ed., Dec. 1996
6. Cuppens, F., Demolombe, R.: How to recognize interesting topics to provide cooperative answering. Information Systems 14(2): 163–173, 1989
7. Cruz, I.F.: Doodle. proc. SIGMOD, 1992
8. Djeraba, C., Bouet, M.: Knowledge-based query in a visual information system. IEEE ADL 1998, Santa Barbara, California, 22–24 April 1998
9. Djeraba, C., Bouet, M.: Digital Information Retrieval. ACM CIKM'97, Las Vegas, USA, November 1997
10. Faloutsos, C., Flickner, M., Niblack, W., Petrovic, D., Equitz, W., Barber, R.: Efficient and Effective Query by Image Content. Research report, IBM Alameda Research Center, 1993
11. Grosky, W.I., Fotouhi, F., Jiang, Z.: Using Metadata for the Intelligent Browsing of Structured Media Objects. In: Sheth, A., Klas, W. (eds.). Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data McGraw Hill Publishing Company, New York, 1998, pp. 123–148
12. Grosky, W.: Toward a data model for integrated pictorial data bases. Computer, Vision, Graph. Image Proc. 25: 371–382, 1984

13. Gudivada, V.N.: Distributed Semantic Multimedia Information Retrieval. Tutorial report, ICAST'97 in conjunction with ICMIS'97, Motorola University, Chicago, USA, April 1997
14. Gupta, A., Jain, R.: Visual Information Retrieval. A communication of the ACM 40(5): May 1997
15. Hafner, J., et al.: Efficient Color Histogram Indexing for Quadratic Distance Functions. In IEEE Transaction on Pattern analysis and Machine Intelligence, July 1995
16. Jiang, H., Montesi, D., Elmagarmid, A.K.: Integrated Video and Text for Content-based Access to Video Databases. MTAP Journal 9: 227–249, 1999
17. Jain, R.: Infoscopes: Multimedia Information Systems. Virage Inc. and University of California at San Diego, 1995
18. Moores, C.N.: Datacoding applied to mechanical organization of knowledge. AM. Doc. 2: 20–32, 1951
19. MPEG-7 Requirements Document. Doc. ISO/MPEG N2727, MPEG Seoul Meeting, March 1999
20. Mokhtarian, F., Mackworth, A.K.: A theory of multiscale, curvature-based shape representation for planars curves. In IEEE Trans. On Pattern Analysis Machine Intelligence. PAMI-14(8): 789–805 1992
21. Nastar, C., Mitschke, M., Meilhac, C., Boujemaa, N.: Surfimage: a Flexible Content-Based Image Retrieval System. ACM-Multimedia 1998, Bristol, England, September 12–16, 1998
22. Ogl, V.E., Stonebraker, M.: Chabot: Retrieval from a relational Database of Images. IEEE Computer 28, December 1995
23. Ortega, M., Rui, Y., Chakrabarti, K., Porkaew, K., Mehrota, S., Huang, T.: Supporting Ranked Boolean Similarity Queries in MARS. IEEE Trans. On Knowledge and Data Engineering 10(6): November/December 1998
24. Pentland, A., Picard, R.W., Sclaroff, S.: Photobook: Tools for Content-Based Manipulation of Image Databases. In Proc. of SPIE-94, pages 34–47, Bellingham, Washington, 1994
25. Pavlidis, T.: Review of Algorithms for Shape Analysis. Journal Computer Graphics Image Process 7(2): 243–258, April 1978
26. Raghavan, V., Jung, G., Bollman, P.: A Critical Investigation of Recall and Precision as Measures. ACM Transactions on Information Systems 7(3): 205–229, 1989
27. Keith van Rijsbergen, C.J.: Information retrieval. Second edition, London: Butterworths, 1979
28. Salton, G.: Automatic Information Organization and Retrieval. McGraw Hill Book Co, New York, 1968, Chapt. 4
29. Salton, G., McGill: Introduction to Modern Information Retrieval. McGraw-Hill, New York, NY 1983
30. Smith J.R., Chang, S.F.: VisualSEEK: a fully automated content-based image query system. ACM Multimedia'96, November 1996
31. Swets, D.L., Pathak, Y., Weng, J.: An image database system with support for traditional alphanumeric queries and content-based queries by example. Journal MTAP 7: 181–212, Kluwer Academic publishers, 1998
32. Zahn, C.T., Roskies, R.Z.: Fourier descriptors for plane closed curves. IEEE Trans. On Computers, 1972