Pattern Analysis & Applications (2001)4:153–173

RETIN: A Content-Based Image Indexing and Retrieval System

J. Fournier, M. Cord and S. Philipp-Foliguet

ENSEA/Université de Cergy-Pontoise, Cergy-Pontoise, France

Abstract: This paper presents RETIN, a new system for automatic image indexing and interactive content-based image retrieval. The most original aspect of our work rests on the distance computation and its adjustment by relevance feedback. First, during an offline stage, the indexes are computed from attribute vectors associated with image pixels. The feature spaces are partitioned through an unsupervised classification, and then, thanks to these partitions, statistical distributions are processed for each image. During the online use of the system, the user makes an iconic request, i.e. he brings an example of the type of image he is looking for. The query may be global or partial, since the user can reduce his request to a region of interest. The comparison between the query distribution and that of every image in the collection is carried out by using a weighted dissimilarity function which manages the use of several attributes. The results of the search are then refined by means of relevance feedback, which tunes the weights of the dissimilarity measure via user interaction. Experiments are then performed on large databases and statistical quality assessment shows the good properties of RETIN for digital image retrieval. The evaluation also shows that relevance feedback brings flexibility and robustness to the search.

Keywords: Classification; Image retrieval; Indexing; Interactivity; Partial query; Relevance feedback

1. INTRODUCTION

In our contemporary society, the place taken by digital documents, especially digital images, is essential. The quantity of such documents, produced by television, press agencies, hospitals, museums, police, etc. is always growing. These images or videos are often compressed and stored in different databases. They may be accessible by telecommunication networks, such as the Internet. Because of the tremendous size of these databases, there is a need for image indexing methods, search algorithms and data classification techniques.

Researchers are now able to produce effective tools for information retrieval in textual documents (Text Retrieval), for instance, the search engines used to access web sites. But if a text is represented by words (words can easily be interpreted), an image is a set of pixels, and it is not easy to interpret a set of numerical values. For example: how can the French president be automatically recognised in a picture?

Even if this question is still under investigation, progress in image processing and interpretation has led researchers to take an interest in this field since the beginning of the 1990s. Image indexing was first done by keywords and search was achieved through text retrieval techniques. The main advantage of such a representation is that it is 'high level' (semantic level), but keywords are external information which is often manually assigned to images. Researchers now use image content to automatically index and retrieve information from digital image libraries.

However, there is still a gap between the user's request, which can be expressed in semantic terms, and the reality of the low level attributes usually extracted from images. One of the most exciting goals in image indexing and retrieval is to fill this gap, and link high level interpretation and low level features. Because human beings are at the end of the image retrieval chain, and as they are the only ones to judge the retrieval quality, it is necessary (and it offers potential) to develop interactive systems (see Salton and Buckley [1] for text retrieval, and Nastar et al [2] for image retrieval).

1.1. Related Work

There are some interesting systems which now enable us to achieve an effective search guided by image content. We

Received: 5 July 2000
Received in revised form: 9 February 2001
Accepted: 15 February 2001

can distinguish two approaches for image retrieval. The first approach is *search-by-similarity* (*search-by-example*), where the goal is to find images which are similar to an example given by the user [3–7]. The other approach is called *target search*. The problem is to lead the system to the target image(s)[1] by validating a displayed set of images [8–10].

The aim of this section is not to present an exhaustive survey of the existing systems in CBIR (the interested reader can refer to Veltkamp and Tanase [11]), rather to give a few details about the main references of our work.

One of the most exhaustive works on image similarities is presented by Nastar et al, and synthesised in the search engine named Surfimage [3]. This system is generic, and based on pre-attentive similarities between the request and any other image in the collection. It gathers a wide choice of image signatures and similarity measurements, allowing the user to define and refine his query. It enables one to search any type of image database, either general or specific, and it involves the user in the retrieval process.

Another interesting system based on interactivity is presented by Schröder et al [12]. It deals with image indexing and retrieval in remote sensing image archives. In a pre-processing stage, an unsupervised Bayesian clustering of the data is done in different attribute spaces (spectral, texture, etc). Then, during the use of the system, the 'interactive learning' step allows us to build up links between the low level clusters and a label defined by the user. For instance, if the user is looking for images containing lakes, by clicking on different pixels of a lake (in the example image), he has built meta-clusters representing a lake. After each click, a posterior probability of being a lake is attributed to every pixel in this image, so the user can visually supervise the learning process. After this learning step, different probabilistic criteria are used as a retrieval score. Furthermore, the system gives the possibility of refining the learning process, and then the search, via feedback. This method is flexible, and gives good results for remote sensing image retrieval.

Another interesting system, FourEyes, has been developed by Minka and Picard [13]. Based on a learning algorithm that selects and combines feature groupings, this system uses the notion of positive and negative examples (given by the user). Thanks to a large choice of image features, this method is presented as a competition within 'a society of models'. As in the two previous systems, interactivity allows for flexibility and query refinement.

A final illustration of an interactive system is PicHunter [8]. Developed by Cox et al, it is designed to find an image similar to what the user has in mind. An original feedback approach which takes the past (all the annotations provided by the user) into account is introduced. The algorithm is based on a stochastic-comparison search: the probability of each image in the database being the target is updated thanks to comparisons carried out by the user. This Bayesian relevance feedback process is interesting, since it is not based on binary decisions (relevant or irrelevant). This kind of information is easier to assess, less arbitrary than the binary one, and it takes into account the uncertainty of human judgment.

1.2. Overview of Our System: RETIN (REcherche et Traque INteractive d'images-Retrieval and Interactive Tracking of Images)

Recognising the essential importance of the user in the retrieval process, in this paper we focus on two different aspects of interactivity: the user's adaptive formulation of the request; and the relevance feedback process. Moreover, our goal is to introduce a generic system applied to image or object search without any restrictions on the type of image contained in the collection.

Our system architecture consists of two stages, the offline processing of the database (indexing stage), and the online search (eventually completed by a relevance feedback step). Figure 1 shows an overview of the system.

In Sections 2 and 3, we focus on the offline processing, from the classification to the image signature computation. Then, Sections 4 and 5 deal with the request formulation and the search process. Sections 6 and 7 emphasise the relevance feedback for result refinement, and finally, Section 8 produces some results and quality assessment.

2. IMAGE SIGNATURES: RELATED WORK

In search-by-similarity, the goal is to find images which are 'close' to the example. It is done with respect to a given similarity measurement, and thanks to image *indexes* computed on image *features*. These features may be of various kinds (points, segments, regions, etc.), and may have different properties such as scale invariance, rotation invariance etc. They are also linked to the database's content (general or specialised database). For example, the indexes used in medical applications are different from those used for image retrieval on the web.

A *signature* is computed for each image in the database, from the set of features. This signature is a structured representation of the image, and is used as an index (it enables searches in a set of images). A lot of image transformations, such as filtering, segmentation [14], and interest points detection [15] can be used to extract features. Four attributes are currently employed in image retrieval – colour, texture, shape and position – and a lot of papers try to find the optimal colour space, or the best texture measurement.

There are a lot of ways to structure features in order to build the image signature. In Surfimage [3], the user may use texture or colour histogram. As Biernacki and Mohr [16] show, the image colour distribution can be modelled by a mixture of Gaussians, where each component stands for a dominant colour associated with its variability. Although the results are not as good as with classical histograms, this signature is economical (it is short and allows a faster search). Nastar [31] sets out an original

[1] Here the *target image* is the image(s) the user has in mind.

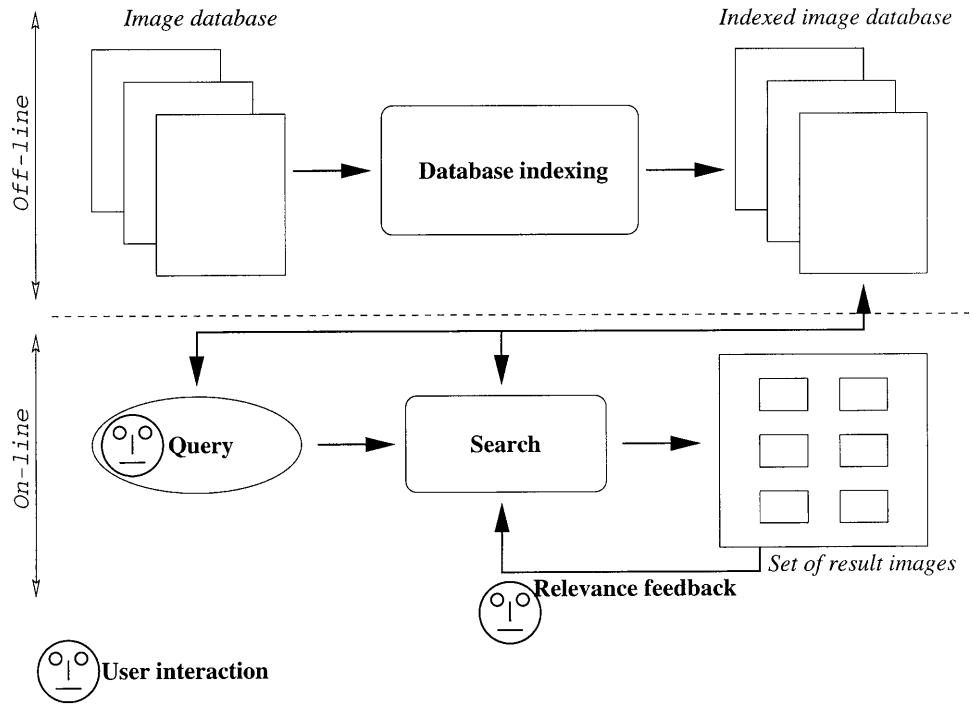

Fig. 1. The whole architecture of our system: an offline step provides the indexed image database. Given a query, an online search is performed in the indexed database. Then, a feedback step allows us to refine the set of results.

signature dedicated to object and face recognition: the Image Shape Spectrum. This index is a histogram of a function of the two principal curvatures processed on the local shape of the intensity surface. It provides an effective shape measurement, invariant in scale, in translation and in rotation, robust to noise, to occlusion and to small viewpoint changes. Another image representation is the eigenimage, introduced by Pentland, Picard and Sclaroff [4]. It is based on the calculation of the eigenvectors of the covariance matrix of the set of image features. The aim is to identify which features are the most effective for image recognition, and thus it provides an efficient similarity measurement for object recognition.

Instead of using a global characterisation, it is interesting to focus on particular areas into images. In Blobworld, Carson et al [14] introduce a system which works on blobs, i.e. regions. After the image segmentation, based on the estimation (using the EM algorithm) of the parameters of a mixture of Gaussians, the system calculates colour, texture and position features on the regions. Wood, Campbell and Thomas [18] also use regions, but with colour, texture, size, position and orientation features. Since the relative position of blobs is effective for image classification (categorisation), the image composition and, especially, the transitions (between blobs) can be very informative. Smith and Li [19] present a new signature called CRT (Composite Region Templates), which is a matrix of frequencies of the vertical transitions between the colour regions. For calculation of this kind of colour co-occurrence, the image is segmented using colour information and cut into five vertical strips. The problem is that these indexes are segmentation-dependent, which remains an unsolved problem without a human supervisor (not feasible for large image collections). Consequently, some authors look for a rigid partition of images. For example, Minka and Picard [13] divide the image into small sub-images, which are then gathered using a flexible learning algorithm based on the competition within a 'society of models'. Malki et al [20] introduce a multi-resolution signature based on a quadtree, in which every localised and structured region is then indexed by a feature histogram.

Another way to compute indexes is to detect and focus on interest points. They are processed in order to concentrate the most informative image areas. After detection, a lot of features can be computed on local patches around these points. For instance, Schmid [21] uses local grey scale invariants, but colour invariants may also be employed, as presented by Mindru [22]. Spatial location can be very informative for image retrieval, especially when working on interest points, which is why Huet and Hancock [23] introduce an extension of a classical histogram which takes into account the relative positions of the points. This signature has also been employed by Heinrichs et al [15].

3. RETIN: CLASSIFICATION AND INDEXING

One of our goals is to compute a compact signature in order to speed up the search. A compact but ineffective index has no interest, which is why we have tried to take into account both efficiency and effectiveness. The signature we

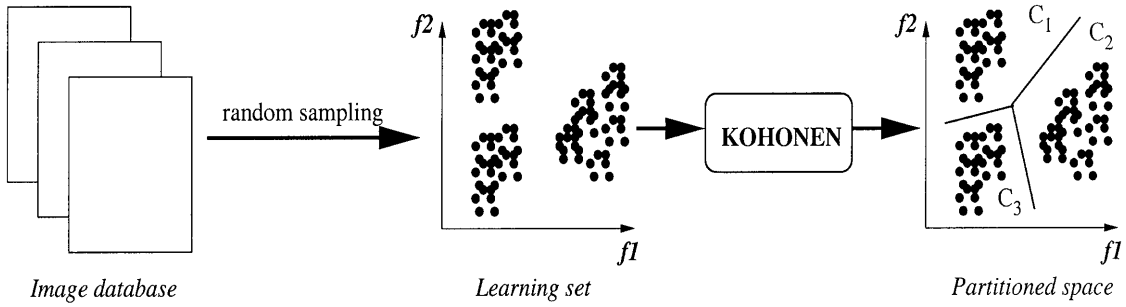

Fig. 2. Partition of the attribute space. f1, f2 are the two components of this space and C_i are the clusters.

have chosen is the statistical distribution of the image, i.e. the proportions of the clusters processed in a previous clustering stage (for different attribute spaces).

3.1. Attributes

RETIN uses colour and texture attributes. For colour, the user chooses one or several spaces (used in parallel) from the following: RGB, normalised colour rgb, HSV, CIELAB or CIELUV.

Texture is a local attribute computed thanks to Gabor filters [24]. It consists of a spectral decomposition of the signal thanks to a filter bank designed to select different frequencies and orientations. These filters are often used for texture analysis in image indexing and retrieval [25,26,15]. They have the property to model the receptive field of neurons in the visual cortex. We use a bank of 12 filters corresponding to four orientations $(0, \frac{\pi}{4}, \frac{\pi}{2}$ and $\frac{3\pi}{4})$ and three frequencies (from medium to low frequencies, i.e. 0.2, 0.1 and 0.05). For every pixel, we have obtained a vector of 12 values, representing the energy of the response of the corresponding filter.

3.2. Classification

Goal and Protocol. Classification is often used for segmentation in image indexing. For instance, Pauwels and Frederix [27] present a non-parametric clustering algorithm for segmentation, but also for region grouping. In our case, classification is just made to partition the attribute spaces. Since data are not uniformly distributed (in the attribute spaces) for a given database, we think it is interesting in terms of efficiency and effectiveness of the signature to use a data adaptive division of the attribute spaces.

The offline processing has to be fully automatic, which is why classification has to be unsupervised. We use a Kohonen neural network [28] for data clustering. It is a simple and well-known architecture which has already proven to be effective. The learning stage of the Kohonen map uses a learning set composed of a large number of pixels extracted from the digital image library. This set of pixels is randomly sampled in the images, and punctual (colour) or local (texture) attributes are computed for each of these. This set is considered as being representative of the database's content. Figure 2 shows a synoptic scheme of the classification process.

Results and Discussion. Figure 3 shows the 10 × 10 Kohonen maps obtained for a classification in the HSV (Hue-Saturation-Value) colour space for the Columbia database[2] [29] and for a general image library[3].

Features are not uniformly distributed in the feature space; data are gathered in dense regions, while other regions are completely empty. The two previous maps display the spatial repartition of the neurons in the weight space, here the HSV colourimetric space. The irregularity in the spatial locations of neurons shows the irregular and data adaptive partition of the attribute space. This irregular division is justified for effectiveness of signatures: in dense areas a fine partition is needed for a good data separation, whereas in others, the partition may be coarser without loss of information.

In conclusion, clustering is warranted in order to take into account the diversity of the collection's content. Any unsupervised classification algorithm such as LVQ (Learning Vector Quantisation), k-means, fuzzy C-means or Bayesian clustering, for example, can be used. Nevertheless, one of the main advantages of the Kohonen map is that it embeds a notion of topology. The fact that neighbouring neurons stand for neighbouring features could be integrated into the retrieval process. The main drawback is that the user has to manually set the number of neurons (the number of clusters). We thought is not essential, since this parameter can be experimentally adjusted, but as noticed by Schröder et al [12], it may be interesting to automatically tune this parameter using a Bayesian classifier with an informative criterion (such as BIC or MDL, etc.), for instance.

3.3. Indexing

Signature. Indexing is the attachment of a signature to an image. This index is external information standing for the image in the retrieval, so its relevance is crucial for the effectiveness of the search. The choice of a signature is often constrained by the retrieval goal and the database

[2] 7200 colour images of single objects.
[3] This database contains 1200 colour images of various origins – see Section 8.1.

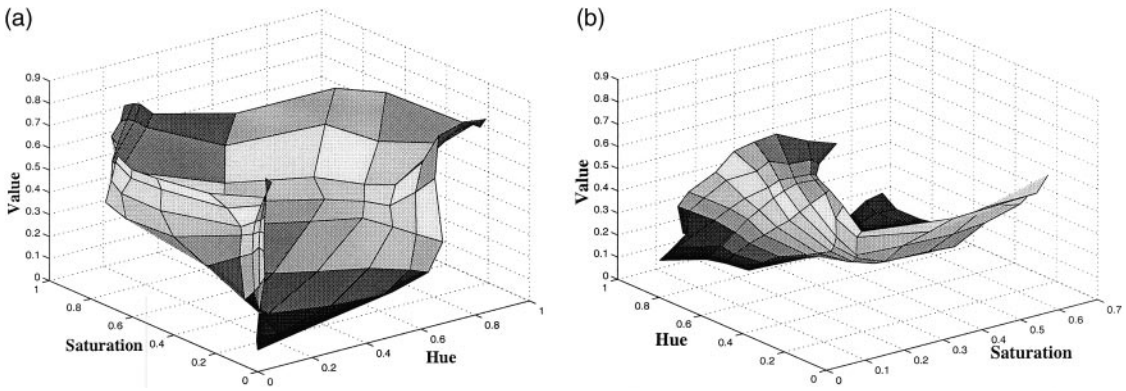

Fig. 3. Self Organising Maps (10 × 10) obtained for the Columbia database (a) and our general database (b).

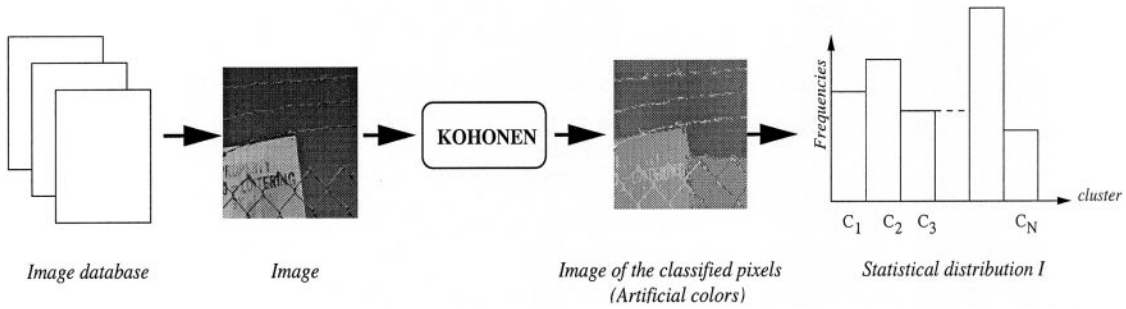

Fig. 4. Indexing process. For each image in the database, the index is of the following form: $\boldsymbol{I} = \{I_i, 1 \leq i \leq N\}$, where N is the number of clusters.

type. For instance, it is obvious that a colour histogram is useless when looking for a shape, or when the library contains grey level images.

Our system indexes the image by its statistical distribution for the previously found clusters. Each pixel is classified by the Kohonen map and the set of frequencies of each cluster constitutes the signature. The procedure is summarised in Fig. 4.

Discussion. Figure 5 shows an example of statistical distribution for a landscape image. The Kohonen map contains 100 neurons (10 × 10) and its inputs are the values in the three colour channels of the pixel in the HSV colour space.

We can notice that data are gathered on a few neurons of the Kohonen map. Two groups of clusters concentrate all the information: they correspond to the sea and the mountains. Although the clustering has been computed for the whole database, grouping within images is effective. This clustering could also be used to perform the segmentation of any image in the collection.

We have compared our statistical distribution with a classical 166-bins colour histogram computed according to Smith and Chang's method [30]. Actually, colour histograms create a lot of small population clusters, whereas our signature groups the data better. This is the result of the data adaptive partition of the attribute spaces, compared with

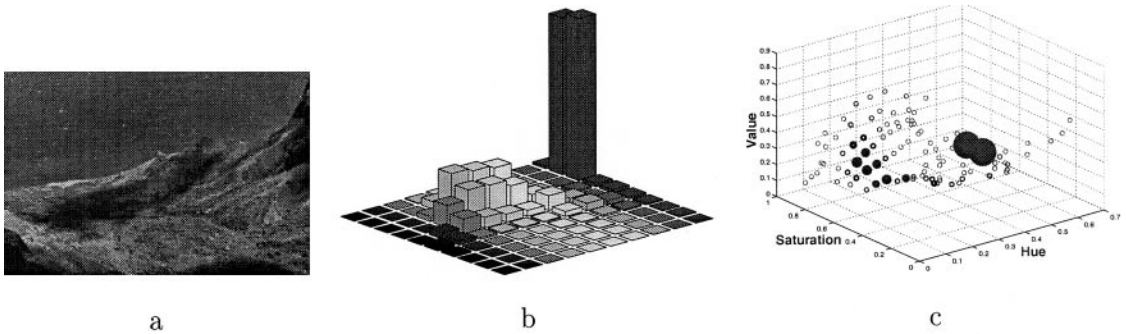

Fig. 5. A landscape image (a), its statistical distribution projected on the 2D Kohonen map (b) and its 3D representation (c) (the locations of the points correspond to the locations of the neurons in the HSV colour space, and the size of the centres is proportional to the cluster population).

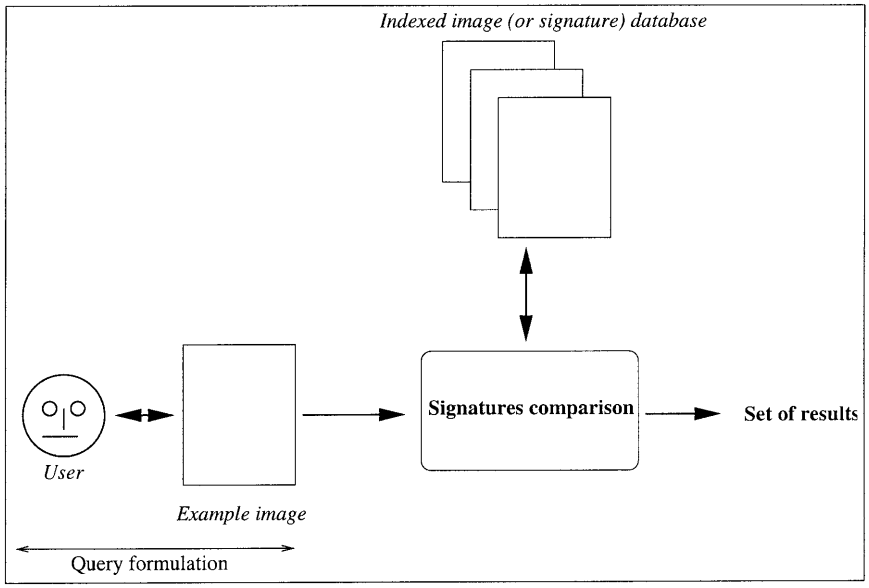

Fig. 6. The retrieval process: the user interacts with the system to formulate a query (more or less complex). Then the request signature is computed, and similar images are retrieved by comparing this signature with those of the indexed database.

the regular partition performed by classical histograms. Since it needs fewer clusters to suit the data, our signature is shorter. It is less heavy on storage, and it also reduces the computational cost of search.

4. SEARCH-BY-SIMILARITY: RELATED WORK

The search-by-similarity retrieves images according to their similarity to a request given by the user. There are two steps: the query formulation (by the user); and the search (based on a similarity function). The retrieval process is summarised in Fig. 6.

There is a strong relationship between the image signature, the request and the metric. For instance, if indexing is based both on textual information and image content, as in Westerveld [31], the system has to combine these two information sources for the final retrieval. The query is sometimes built thanks to a complex interaction with the user. As an illustration, the user can point out a region in the request image and/or choose the attributes.

Focusing on content-based image retrieval using search-by-similarity, there are two main approaches: the global query and partial query. In the first case, the whole image is taken into account for signature computation, so the retrieval is limited to a global similarity measurement. If the user wants to find images of a particular object or person, a partial similarity is more adequate.

In Carson et al [14], the authors search for regions, and the request is not an image but a region. The user may also ask for several regions at the same time, and specify the relative weights of every processed attribute. In this way, the system becomes flexible and allows for the retrieval of a part of an image, which can correspond to an object. Image segmentation is an ill-posed problem, which is why partial request without segmentation, as proposed by Malki et al [20] or Minka and Picard [13], is an effective alternative. The user now specifies the blocks of interest in the stamped query image. For Malki et al, the request is multi-resolution (quadtree). It brings flexibility to the system, since it allows several sizes of blocks, and allows one to look for a particular object in a given background.

A similarity (or dissimilarity) function aims at comparing the request to the target[4] signatures. There are many possibilities of similarity or dissimilarity function; the choice of a particular one is important, and has to suit the user's goal. As Schröder et al [12] have noticed, different probabilistic metrics lead to different results. Nevertheless, in some cases, the metric has no real importance. For example, Nastar [32] noticed that the choice of the dissimilarity function is of minor importance, since its signature (Image Shape Spectrum) is scale, translation and rotation invariant.

Another key point in CBIR is the combination or competition between features or image models. This problem is discussed in some papers [13–15], and is often viewed as a feature weighting problem solved in a more or less automatic way, thanks to user interaction. As an illustration, while FourEyes [13] and Heinrichs' [15] systems try to automatically integrate the user's expertise, in Blobworld [14] the weighting stage is manually driven.

There are a lot of different query formulations, and almost every system introduces a new one. For instance, the query can be a sketch given by the user [5,33]. Researchers are now working on 'hybrid' queries (and searches) mixing different information sources like image content, text and user interaction. Schröder et al [12] propose an interactive

[4] Here the *target image* is the image currently compared to the request.

learning procedure which builds a probabilistic query associated with an object or a semantic label (a cover-type[5] in remote sensing images). The request is translated into a set of weights of a Bayesian network linking the user's high level goal and the low-level index. It becomes obvious that user interaction (for the query formulation or during the feedback (see Section 6)) is useful to guide the search. Actually, a partial request offers a great flexibility and a lot of possibilities for the formulation and for user interaction. The drawback is that it is difficult to manage.

The visualisation of the resulting set of images appears as an important aspect of the search. The most commonly used display of result is the ranked list of images [3,12] (eventually with their respective similarity/dissimilarity values). However, some authors studied a more sophisticated visualisation scheme. For instance, Rubner [26] introduced a method that displays the set of result images as a 2D or 3D mosaic, giving an intuitive model of the whole collection. It allows us to better appreciate similarities between images, and it enables the user to easily choose to infer or not in the search by navigating in the database (through the feedback process).

An important constraint for the search is the processing complexity (it is linked to the retrieval speed). A structured image library (for instance, achieved using a tree structure [34,35]) can speed up the search, and lead to improved results.

5. RETIN: REQUEST AND SIMILARITY MEASUREMENT

This section aims at describing our request formulation and our online search process.

5.1. Query Formulation

Our system offers two types of query-by-example: the global request and the partial request (request on a part of an image):

- *Global request*: the user only brings in an example image, and the system retrieves similar ones in the database. The similarity (or dissimilarity) measurement is based on the signature presented in Section 3.3. This kind of search only deals with global similarities, and can be used for category search.
- *Partial request*: this kind of request is more flexible, and allows for the retrieval of objects or regions. This approach is close to a classical pattern recognition problem.

Our request is built on three steps: the user brings in an example image; draws a polygon of interest around an object or a region; and finally, gives a textual label to this query ('bearded man' for the partial request presented in Fig. 7). The request statistical distribution is only computed for the polygon of interest. For instance, Fig. 7 compares the statistical distributions of a global and a partial request.

This example clearly shows the changes in the request statistical distribution. The clusters corresponding to the background (i.e. the dark colours) have been removed from the request signature. The retrieval process will only focus on the face's signature.

Actually, the formulation of the query enables us to build the links between the low level clusters and a semantic label. As Schröder et al [12] show, it can be interpreted as the weighting stage of a network linking high and low level analysis of the request (see Fig. 8).

Notes.

- Our request distribution is normalised: $\sum_{i=1}^{N} w_i = 1$, where N is the number of clusters. So, the image size has no influence on the retrieval.
- Both label and partial request distributions are saved, allowing a third request type by keyword.

5.2. Dissimilarity Functions

Let us now detail the search process. Our problem is to compute a similarity (or dissimilarity) measurement between two statistical distributions. Heinrichs et al [36] and Sarrut and Miguet [37] give an overview of the main alternatives. The most commonly used distances are the Minkowski metrics (L-metrics), the Kullback–Leibler distance, for instance. To achieve a more robust distance (robust to shifts, expansions of distributions, etc.), Rubner [26] developed the Earth Mover Distance (EMD), which sees the problem of distribution matching as a transportation cost problem. Even if this metric has 'good properties' for image retrieval, its computational cost remains important.

The request image is just an example of what the user is looking for. The similarity function has to allow a flexible matching between the request distribution $\boldsymbol{R} = \{R_i,\ 1 \leq i \leq N\}$ and the target distribution $\boldsymbol{T} = \{T_i,\ 1 \leq i \leq N\}$ (where N is the number of clusters). To solve this problem, we propose two simple dissimilarity functions (d_1 and d_2) derived from the L-metrics (of order p).

Dissimilarity d_1:

$$d_1(\boldsymbol{R}, \boldsymbol{T}) = \left(\sum_{i=1}^{N} \alpha_i |R_i - T_i|^p \right)^{\frac{1}{p}} \quad \text{with} \sum_{i=1}^{N} \alpha_i = 1 \quad (1)$$

and $\alpha_i \geq 0$

In the case of the partial query, the request distribution includes only a few non-null clusters. So, all the empty clusters in the request distribution are set to 0 in the target distribution, which is then normalised to 1 $\left(\sum_{i=1}^{N} T_i = 1 \right)$. If the distribution obtained is not statistically significant (i.e.

[5] Lake, forest, etc.

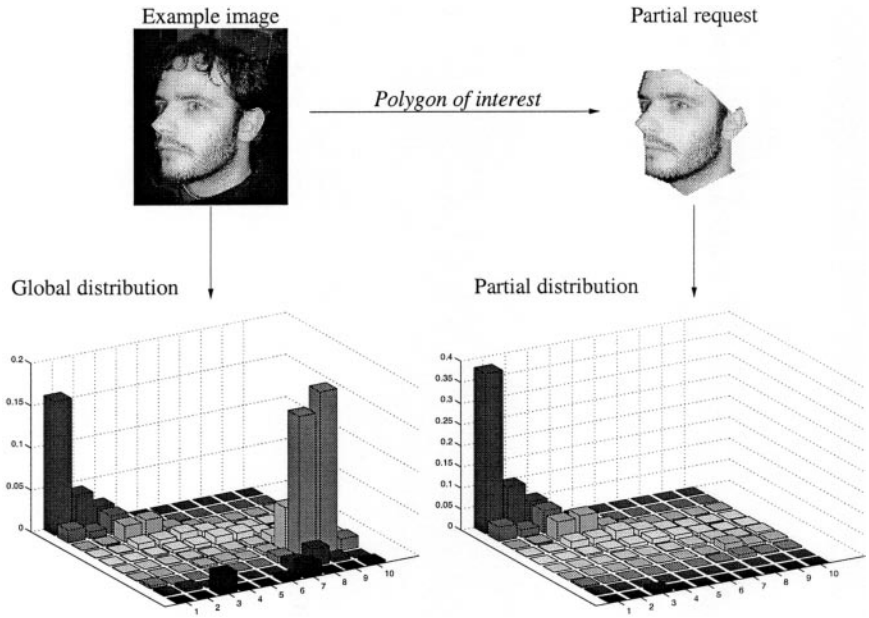

Fig. 7. Comparison between the statistical distributions of a global and a partial request. The clustering is made in CIELAB colour space through a 100 neurons Kohonen map (100 clusters).

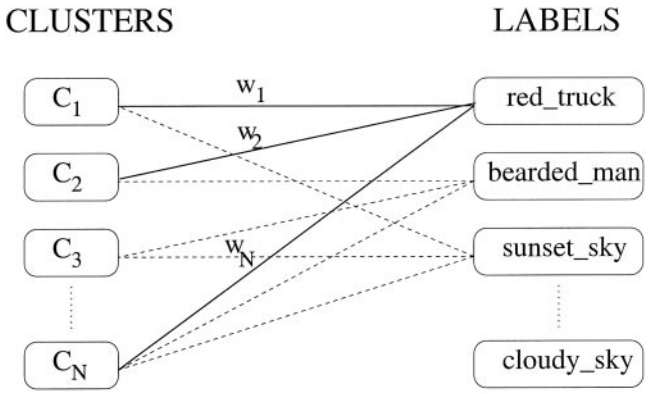

Fig. 8. Network linking the pre-processed clusters and the semantic labels. The weights w_1, w_2, ..., w_N (where N is the number of clusters) correspond to the frequencies of each cluster for the given label.

the number of remaining clusters includes less than 25% of the pixels of the target image), the corresponding target image is discarded (in other words, the corresponding target image cannot be retrieved). This test speeds up the search, since it avoids useless calculations and improves performances through the elimination of critical candidates for matching.

Dissimilarity d_2:

$$d_2(\mathbf{R}, \mathbf{T}) = \left(\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{ij} |R_i - T_j|^p \right)^{\frac{1}{p}} \quad \text{with} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{ij} = 1 \qquad (2)$$

and $\alpha_{ij} \geq 0$

This second dissimilarity measurement is more flexible, since it allows cross matching between the two statistical distributions. The key idea is that a bin-to-bin matching (as done by d_1) is not robust to small changes (like a shift) between the request and the target distribution. Because of the great flexibility provided by this function (d_2), some constraints have to prevent incoherent matchings between distant clusters. Moreover, in this case, the empty clusters (in the request distribution) are not set to 0 in the target distribution (it would break the interest of this metric).

The sets of weights $\boldsymbol{\alpha} = \{\alpha_i,\ 1 \leq i \leq N\}$ and $\boldsymbol{\alpha} = \{\alpha_{ij},\ 1 \leq i \leq N,\ 1 \leq j \leq N\}$ used in the dissimilarity functions settle the influence of each cluster. Since these measurements are designed to improve the statistical distribution matching, tuning of weight values (in the sense of result refinement) is necessary and crucial. This is achieved through user interaction (relevance feedback – see Section 7). Before the feedback process, the initial set of weights is fixed as shown in Table 1 (where $\mathbf{R} = \{R_i,\ 1 \leq i \leq N\}$ is the request distribution).

Notes.

- According to this initialisation (see Table 1), d_1 and d_2 are equivalent to the L-metrics, except for the modification of the target distribution.
- Our dissimilarity function d_1 is a metric since it respects the properties of non-negativity, symmetry, identity and triangular inequality. Nevertheless, if it is a metric for distributions, it is not a metric for images, since two different images may have the same statistical distribution.

5.3. Dissimilarity Functions for Model Competition

The previous section presented our dissimilarity functions for one image model. As mentioned in Section 3.1, the

Table 1. Initial set of weights (see text)

	Global request	Partial request
Dissimilarity d_1	$\alpha_i = \frac{1}{N}$	$\begin{cases} \alpha_i = \frac{1}{\sum_{j=1}^{N} \mathbb{I}_{R_j \neq 0}} & \text{if } R_i \neq 0 \\ \alpha_i = 0 & \text{otherwise} \end{cases}$
Dissimilarity d_2	$\begin{cases} \alpha_{ij} = \frac{1}{N} & \forall i = j \\ \alpha_{ij} = 0 & \text{otherwise} \end{cases}$	$\begin{cases} \alpha_{ij} = \frac{1}{\sum_{j=1}^{N} \mathbb{I}_{R_j \neq 0}} & \forall_i = \text{j and } R_i \neq 0 \\ \alpha_{ij} = 0 & \text{otherwise} \end{cases}$

* $\mathbb{I}_{R_j \neq 0}$ is the indicator function: $\begin{cases} \mathbb{I}_{R_j \neq 0} = 1 \text{ if } R_j \neq 0 \\ \mathbb{I}_{R_j \neq 0} = 1 \text{ if } R_j = 0 \end{cases}$

user of RETIN can put several image attributes in parallel. We are now proposing to extend these matching schemes for cooperation of (or competition between) several models.

$\boldsymbol{\alpha}$ is an intra-model weight set adjusting the importance of each cluster in the dissimilarity function (for a single image model). Let us now introduce $\boldsymbol{\beta} = \{\beta_k, 1 \leq k \leq M\}$, the inter-model weight set tuning the influence of each model in the final dissimilarity measurement. This formalism allows a consistent joint updating of the intra- and inter-model weights. As mentioned in the previous section, their values are settled due to relevance feedback. Let us call $\mathbf{R}^{(k)} = \{R_i^{(k)}, 1 \leq i \leq N^{(R)}\}$ and $\mathbf{T}^{(k)} = \{T_i^{(k)}, 1 \leq i \leq N^{(R)}\}$, respectively the request and the target distributions for the model number k ($1 \leq k \leq M$ and $1 \leq M \leq 6$, where M is the number of models used and $N^{(k)}$ is the number of clusters for the model number k).

In the context of model competition, our dissimilarity functions is:

Dissimilarity D_1:

$$D_1(\mathbf{R}, \mathbf{T}) = \sum_{k=1}^{M} \beta_k \left(\sum_{i=1}^{N^{(k)}} \alpha_i^{(k)} |R_i^{(k)} - T_i^{(k)}|^p \right) \quad (3)$$

with $\sum_{i=1}^{N^{(k)}} \alpha_i^{(k)} = 1$, $\sum_{k=1}^{M} \beta_k = 1$ and $\alpha_i^{(k)} \geq 0$, $\beta_k \geq 0$.

Dissimilarity D_2:

$$D_2(\mathbf{R}, \mathbf{T}) = \sum_{k=1}^{M} \beta_k \left(\sum_{i=1}^{N^{(k)}} \sum_{j=1}^{N^{(k)}} \alpha_{ij}^{(k)} |R_i^{(k)} - T_j^{(k)}|^p \right) \quad (4)$$

with $\sum_{i=1}^{N^{(k)}} \sum_{j=1}^{N^{(k)}} \alpha_{ij}^{(k)} = 1$, $\sum_{k=1}^{M} \beta_k = 1$ and $\alpha_{ij}^{(k)} \geq 0$, $\beta_k \geq 0$.

The sets $\boldsymbol{\alpha}^{(k)} = \{\alpha_i^{(k)}, 1 \leq i \leq N^{(k)}\}$ and $\boldsymbol{\alpha}^{(k)} = \{\alpha_{ij}^{(k)}, 1 \leq i \leq N^{(k)}, 1 \leq j \leq N^{(k)}\}$ ($1 \leq k \leq M$) are the intra-model weights for the model number k; their initialisation is the same as in Section 5.2. The inter-model weights $\boldsymbol{\beta} = \{\beta_k, 1 \leq k \leq M\}$ are initialised with the same value:

$\beta_k = \frac{1}{M}$ ($1 \leq k \leq M$).

6. RELEVANCE FEEDBACK: RELATED WORK

The set of initially retrieved images includes wrong answers. It may be explained by a problem in the request definition, in the signature or in the similarity function. The wrong images cannot be automatically interpreted as outliers, since they are 'similar' to the request according to the similarity function. Even if the system suits the target application, the search-by-similarity may not satisfy the user's intention. This is due to a gap between the semantic request and the syntactical information extracted from images. The user's expertise allows one to overcome this problem and refine the search. Moreover, it is obvious that the retrieval depends upon the user's aim: a subjective goal is hard to reach through an objective search process. So, there is no satisfactory search without a strong interaction involving the user.

The result refinement guided by the user is called *relevance feedback*. Research in text retrieval has already proven its utility [1], and image retrieval effectiveness can also be improved through relevance feedback. It brings flexibility and adaptability to the search through the integration of the user's expertise and subjectivity. The first proposed approach directly uses the technique developed in TR [38]. It is based on the calculation of an optimal request by additions and subtractions of the relevant and irrelevant image vectors, to the initial query vector [39].

Another technique for feedback is used in QBIC [5] and

Blobworld [14]. The user is considered as an expert who manually adjusts the weights of the attributes. The problem with this approach is that it relies on complex and uncertain human expertise.

As mentioned in Section 1, we can distinguish between two types of systems integrating relevance feedback. On the one hand, there are search-by-similarity systems [2,15] that try to extract images close to a query. On the other hand, there are systems for target search [8,9] where the goal is to look for an image that the user has in mind without any initial example image. It is repeatedly guided by the user's reactions on a set of displayed images.

In search-by-similarity, the example image stands for a seed in the search space. There are two ways to refine the results in this case. In the first, the feedback allows us to shift the seed because of the user's reactions [2,40,12]. Nastar and Meilhac [2,40] compute a new query which takes into account both the positive and negative examples given by the user after the initial search. It is based on a parametric [2] or a non-parametric [40] estimation of the probability distributions of the relevant and irrelevant images. According to these estimations, a new query is drawn at random according to the estimated probability densities. In Schröder et al [12] (see Section 4), the user selects one of the result images to refine his/her query via an interactive learning procedure. There is no fusion of queries. The second manner in which to refine the results is to tune the similarity function. For instance, Heinrichs [15] uses the rank (in the retrieved list of images) of the labelled images to compute a new set of feature weights. The main idea is that a feature which better ranks relevant images has to be reinforced contrary to that which better ranks irrelevant ones. Minka and Picard [13] also use this kind of competition between models. For this second approach, the seed is fixed, but the shape of the search neighbourhood (the neighbourhood in which similar images are retrieved) changes with the reactions of the user.

In the target search strategy, as there is no initial request, there is no seed. Every image of the database has an equal probability of being the target. Then, according to the user's reactions on a set of result images, the system updates these probabilities. Cox et al [8] try to build a probabilistic model of the human behaviour, based on a 'stochastic-comparison search' algorithm which enables comparisons like: 'image A is more relevant than image B'. The relative judgments make the algorithm more flexible and effective, because it takes into account the notion of uncertainty of the human judgment. Geman and Moquet [9] carry out a similar kind of Bayesian relevance feedback, but with a stochastic search providing a sequence of random metrics. Since they do not deal with query drift, Müller et al [10] propose the extension of these Bayesian methods in the case of moving targets, i.e. when the feedback annotations are inconsistent with the earlier ones.

7. RETIN: FEEDBACK PROCESS

Since RETIN is based on search-by-similarity, as explained in Section 6, we can refine the query or the metric. Our relevance feedback optimises the metric in a particular feature space (i.e. for a given model), but also manages the competition between image models.

7.1. Feedback Rule

Our dissimilarity functions for a single image model (see Section 5.2) are weighted sums of the cluster-to-cluster dissimilarities (one distribution for the request and one distribution for the target image). The preliminary retrieval gives a set of results that the user annotates (in this framework, to annotate means to label the images as relevant or irrelevant). Then, the system computes a new set of weights adapted to the user's reactions.

The aim is to increase the weights of the reliable clusters, and to decrease the others. We consider that a cluster is reliable if the matching between the request and the target is correct (for this cluster). We use the Least Mean Squares (LMS) rule [41] to perform the weights updating. The minimisation of the LMS criterion allows the system to learn the statistics of the target distributions. It minimises the quadratic error between the dissimilarity measurement obtained for a target image T (compared with the request R) and the desired output for this image S_d. The desired output is set through the user's relevance annotation. An image considered as relevant should have a small dissimilarity (d_1, $d_2 \in [0, 1]$ with respect to the query:

$$\begin{cases} S_d = 0 & \text{if } T \text{ is relevant} \\ S_d = 1 & \text{otherwise} \end{cases}$$

Given $\mu(\mu > 0)$, a learning rate (it may be constant or it may decrease) and N, the number of clusters of the statistical distribution, the updating rules are (see the similarity functions in Section 5.2):

- *For dissimilarity* d_1:

$$\alpha_i^* = \alpha_i + \mu(S_d - (d_1(\mathbf{R}, \mathbf{T}))^p)|R_i - T_i|^p \quad 1 \le i \le N$$

- For *dissimilarity* d_2:

$$\alpha_{ij}^* = \alpha_{ij} + \mu(S_d - (d_2(\mathbf{R}, \mathbf{T}))^p)|R_i - T_j|^p \quad 1 \le i \le N, \; 1 \le j \le N$$

Note. The LMS criterion provides a strict theoretical framework (especially for convergence), which explains its use in many contexts and applications.

7.2. Feedback Rule for the Competition Between Models

We now present the feedback rules for the competition between image models as an extension of the rules introduced in Section 5.3. The LMS rules result from the minimisation of the Quadratic Error:

$$Err = \frac{1}{2}(S_d - D_i)^2 \quad (i = 1, 2)$$

where S_d satisfies the conditions imposed in Section 7.1.

The optimisation of these criteria leads to the following

updating rules (we use the notations introduced in Section 5.3):

- *For dissimilarity* D_1:

$$\beta_k^* = \beta_k + \mu(S_d - D_1(\mathbf{R}, \mathbf{T}))\ (d_1^{(k)}(\mathbf{R}, \mathbf{T}))^p$$

$$\alpha_i^{(k)*} = \alpha_i^{(k)} + \mu(S_d - D_1(\mathbf{R}, \mathbf{T}))\beta_k |R_i^{(k)} - T_i^{(k)}|^p$$

with $1 \le i \le N^{(k)}$ and $1 \le k \le M$.

- *For dissimilarity* D_2:

$$\beta_k^* = \beta_k + \mu(S_d - D_2(\mathbf{R}, \mathbf{T}))\ (d_2^{(k)}(\mathbf{R}, \mathbf{T}))^p$$

$$\alpha_{ij}^{(k)} = \alpha_{ij}^{(k)} + \mu(S_d - D_2(\mathbf{R}, \mathbf{T}))\ \beta_k |R_i^{(k)} - T_j^{(k)}|^p$$

with $1 \le i \le N^{(k)}$ and $1 \le j \le N^{(k)}$ and $1 \le k \le M$.

These rules are error back-propagation rules. The intra- and inter-model weights are adapted, so it leads to the intra-model dissimilarity optimisation and the models competition, at the same time.

Note. This relevance feedback scheme allows us to track the user's goal, even if it changes through the time.

7.3. Feedback Protocol

The relevance feedback is based on a simple user interaction. After a retrieval run, the system displays the best ranked images (the number of displayed images is set by the user) according to the dissimilarity function. Then, the user labels all these images as relevant or irrelevant (he clicks on the left button of the mouse for a relevant image and the right button for an irrelevant one). Thanks to the annotated images, the system sequentially updates the weights using the rules presented in Sections 7.1 and 7.2, and discards all the irrelevant images from the explored image set. It means that these 'wrong' images will not disturb the future searches (until the next query). After this step, the updated weights allow the system to retrieve and display a new set of images approximating the user's guess.

In such a process, it is essential to check that the weight values are not divergent. Figure 9 shows an example of the weight evolution along 40 feedback iterations for a competition between three image models (HSV colour, CIELAB colour and texture). This experiment has been done on our general database (1200 colour images of various types), for 25-cluster classifications and for 10 annotated images (at each feedback step).

This example provides a good illustration of the weight evolution during the feedback process, because it clearly shows that a hierarchy appears between the models (Fig. 9(a)) and between the clusters (Fig. 9(b, c, d)). It shows how the search focuses on the most reliable models, and on the most reliable clusters for a given model. Here, we notice that texture is the most discriminant model, and that a few clusters (for each model) are useless for the search. Nevertheless, the evolution of the weights is often irregular, and the convergence rate changes according to the μ value: the more μ is large, the more the convergence is fast, but the more the residual fluctuation phenomenon is enhanced. In our system, the learning rate is constant and has been empirically chosen.

Finally, the weight stability and the search coherence depend upon the reliability of the user's annotations; moreover, the correct weight evolution does not ensure the user's satisfaction. Actually, the success of the retrieval is linked to the signatures' richness and flexibility, and is also linked to the initial request. For the quality assessment (Section 8), we will try to quantify the convergence towards the user's goal.

8. RESULTS AND QUALITY ASSESSMENT

8.1. Introduction

We have tested our system on four different databases. The first is the Columbia database [29], which contains 7200 colour images of isolated objects (100 objects taken at 5 degrees incremented in pose = 72 shots per object). This database is suitable to evaluate the performances of our system for object recognition. The second database is our man-made *compound* database. It contains 3000 images combining two objects extracted from the Columbia digital image library. It is used to evaluate the partial request ability to separate two objects in the same image. The third is our *general* database, it is made of 1200 images of different origins (from the web, from the IGN aerial images set[6], from the VisTex database [42], etc. – it is composed of animals, cars, textures, aerial images, portraits, landscapes, etc.). This library is very interesting in order to appreciate the retrieval effectiveness for categorisation because of its wide content diversity. The fourth is the *Annotated groundtruth database* (ANN) [43] of the University of Washington, which provides 493 photographs of different topics. The statistics presented in this section only concern the Columbia and the general database, because the ground truth is not available on the compound digital library, and because the ANN collection is too small.

First, evaluation of the retrieval without feedback is presented (Section 8.2), then the quality improvement by relevance feedback is assessed (Section 8.3).

8.2. Experiments without Relevance Feedback

This section focuses on the retrieval effectiveness without relevance feedback. Based on a rigorous evaluation protocol, the performances of RETIN are compared with a colour histogram-based method, and the parameter influence is estimated.

Some Results. Figure 10 shows the results for two global queries on the Columbia database and on our general database.

[6] IGN: Institut Géographique National (French National Geographic Institute).

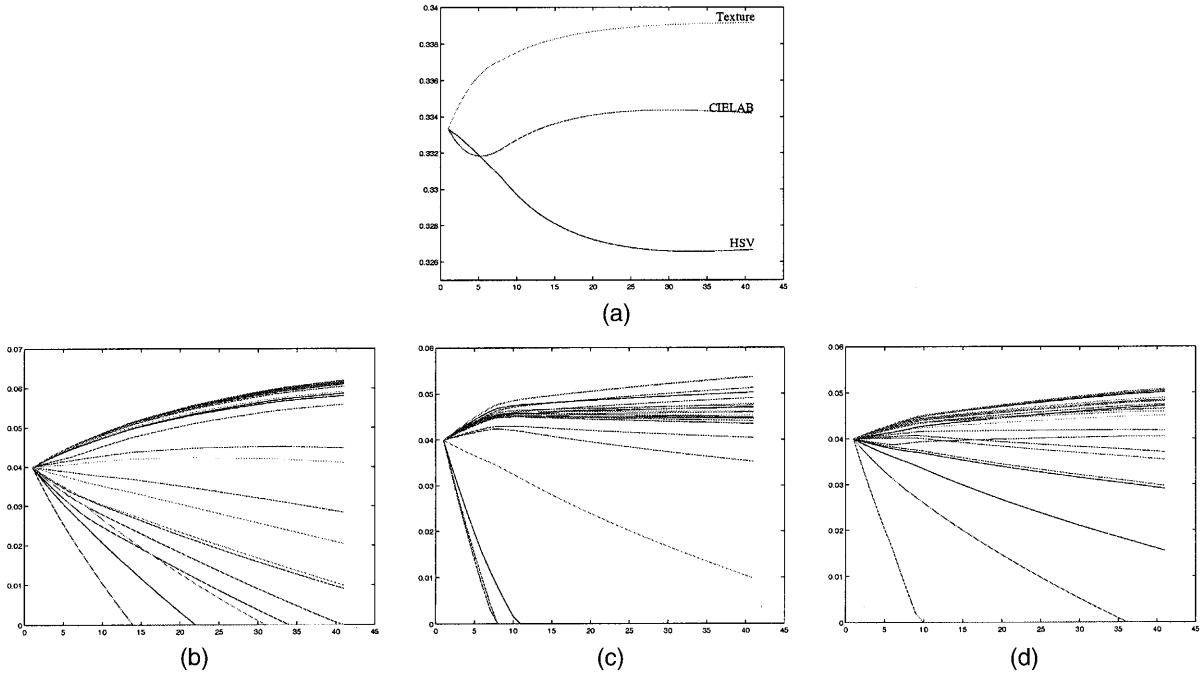

Fig. 9. The dissimilarity measurement weights as functions of the number of feedback iterations (40 iterations). (a) Inter-model weights, (b) HSV-model weights (25 curves corresponding to 25 weights), (c) CIELAB-model weights, (d) texture-model weights.

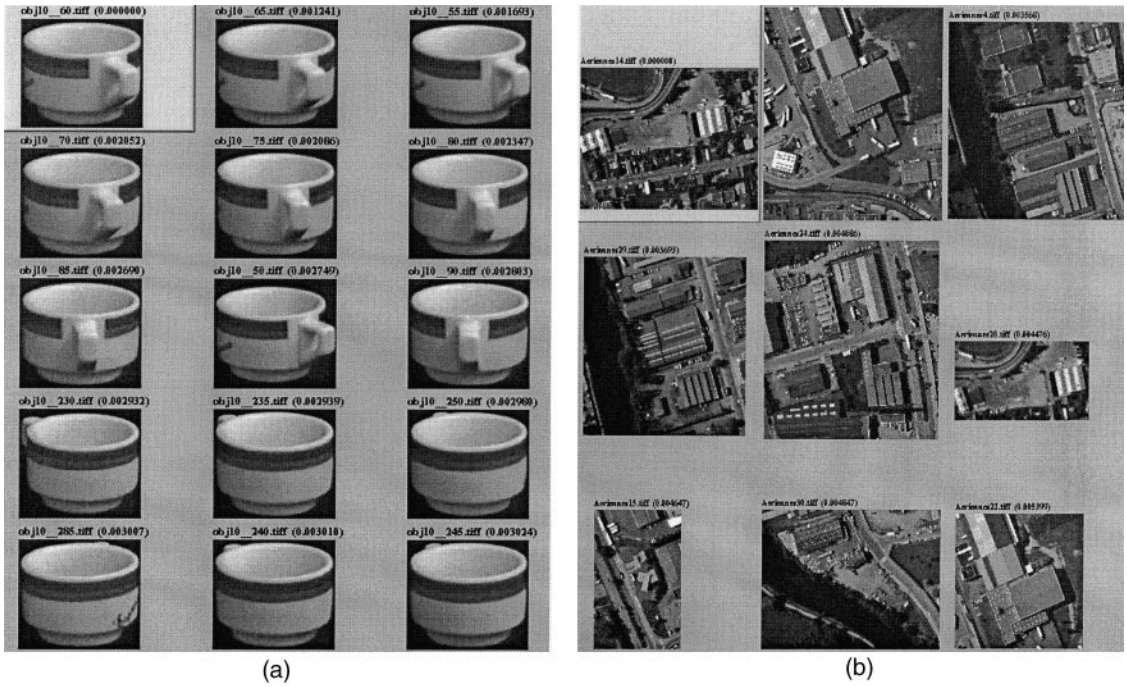

Fig. 10. Results of the search for a global query. (a) Retrieval of cups in the Columbia database using colour attributes (100 clusters in the HSV colour space); (b) retrieval of aerial images in the general database using colour and texture attributes (100 clusters in the HSV colour space and 100 in the texture space). The request is the top left-hand image and the results are ranked by increasing dissimilarity values (from left to right and top to bottom).

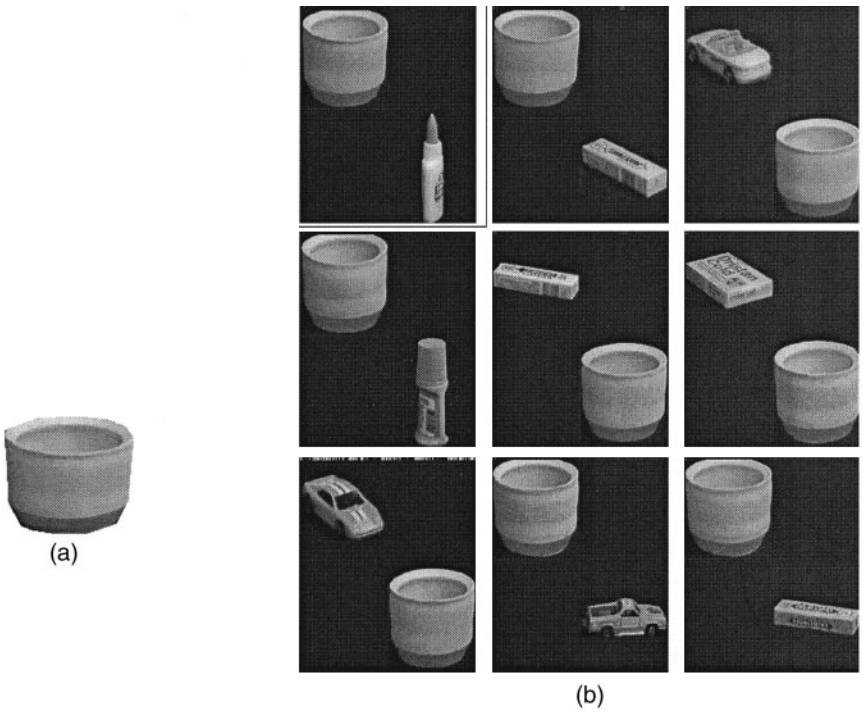

Fig. 11. Results of the search for a partial request on the object *pot*. (a) Region of interest (corresponding to the *pot*) in the example image; (b) ranked list of result images. The search is based on color attributes (100 clusters in HSV colour space).

We notice that all the retrieved images are relevant. For the Columbia database, the first wrong image appears at the 37th position in the ranked list of results. Actually, according to our tests, the system retrieves a wide number of relevant images[7] in the Columbia database, whereas the task is harder in the general database.

Let us introduce results stemming from experiments for a partial request on the compound database (see Fig. 11).

The nine retrieved images contain the pot. Generally, the partial request allows us to retrieve a single object in the compound database. In fact, the problem here is to separate the statistical distributions of the two objects, whereas for the global request, both distributions are mixed, the partial query focuses on one particular object and builds its own distribution. If the objects do not have too many common clusters, a majority of relevant images are retrieved.

Evaluation Protocol. To quantitatively evaluate the retrieval effectiveness of our system, we use the classical and relevant [44] criteria: precision and recall. If A is the whole set of relevant images contained in the database and B is the set of retrieved images, precision and recall are defined as:

- $\text{precision} = \dfrac{|A \cap B|}{|B|} = \dfrac{\text{number of relevant images retrieved}}{\text{number of retrieved images}}$
- $\text{recall} = \dfrac{|A \cap B|}{|A|} = \dfrac{\text{number of relevant images retrieved}}{\text{number of relevant images}}$

Note. $|A \cap B|$ is often called the number of detections.

A *ground truth* is necessary to provide the relevance of any retrieved image. For the Columbia database, an image is relevant if it belongs to the category of the request (i.e. it is the same object). Nevertheless, for our general database, it is more difficult to estimate if an image is good with respect to a query. We have manually clustered the 1200 images into 14 categories (aerial images, lions, sunsets, cars, etc.).

To assess the various performances of our system, precision and recall are computed for an increasing number of retrieved images (i.e. the search is done for 1, 2, 3, etc., until 200 retrieved images). For a given database and a given image category, this protocol is repeated for several requests, and the quality criteria are averaged over all these queries. Three curves are then drawn: (average) precision versus number of retrieved images; (average) recall versus number of retrieved images; and (average) precision versus (average) recall.

Note. In the Columbia database, there are 72 images per object, that is to say, 72 relevant images per search. For the general database, we separate the evaluation of each image class because the number of relevant images is not constant between categories.

Comparison between Colour Histograms and our Signature. We have compared the retrieval performances of our

[7] A relevant image suits the user's goal, i.e. it belongs to the category of the request.

colour signature with a colour histogram-based method. This last signature is presented by Smith and Chang [30]; it partitions the HSV colour space into 166 bins (18 levels on Hue, 3 on Saturation, 3 on Value and 4 on grey levels).

Let us first introduce the results for our general database. Figure 12 shows the averaged quality criteria obtained for global requests performed in three image categories (aerial images, elephants and lions).

The first important fact is that our system performs better than the colour histogram for every image category. Our signature retrieves relevant images in the database better, whatever the number of results. It is due to the fact that for an equivalent number of clusters, our signature is adapted to the database's content diversity.

Another interesting issue is the absolute performances. For aerial images, the absolute performances of the colour histogram and our signature are superior to the other categories (here elephants and lions). Actually, aerial images are well separated from the other images in the database, thanks to colour attributes.

Let us now study the retrieval performances for the Columbia database. Figure 13 compares the evolution of the quality criteria for three approaches: the colour histogram; our signature with a global request (see Section 5.1); and our signature with a partial request (see Section 5.1).

The first interesting observation is that the three curves follow the same tendency. The retrieval properties are equivalent for these methods, and the linear tendency of the recall curves until 72 retrieved images shows that they all allow us to find a large set of relevant images in a restricted set of results. After 72, since the number of remaining relevant images is weak, the recall grows less rapidly and an inflexion point appears.

Our signature still performs better than the classical colour histogram. We also notice that the partial request is more effective and more appropriate for the search in this database. It stems from the fact that this query mode focuses on the object clusters, and eliminates all the disturbing classes of the background.

Given these retrieval performances, the search appears

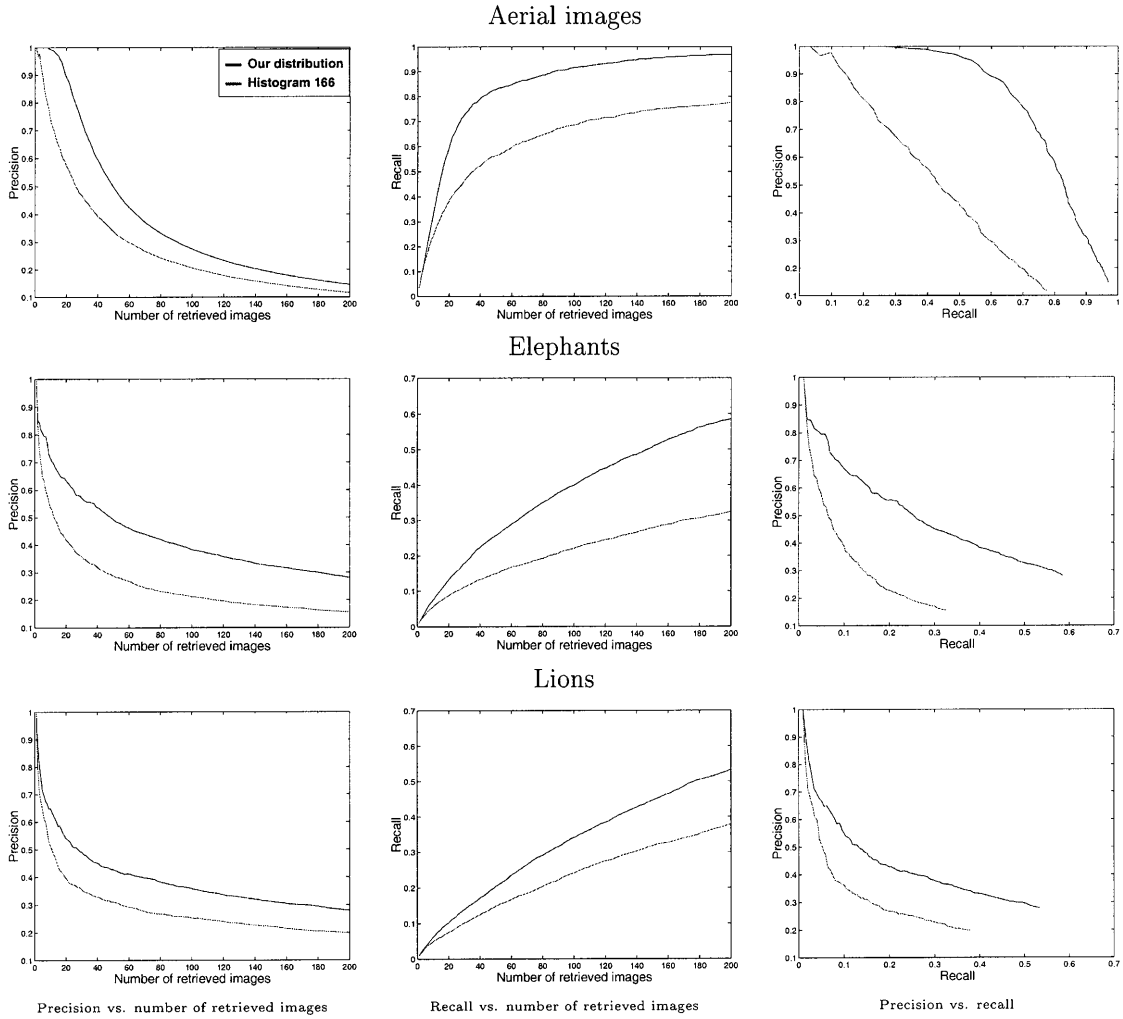

Fig. 12. Comparison between the colour histograms and our colour signature (169 clusters in the HSV colour space) for the general database. The image distributions are compared using the same L_1-metric, and the quality criteria are averaged over all the possible requests belonging to the category. The number of relevant images is 30 for aerial images, 96 for elephants and 105 for lions.

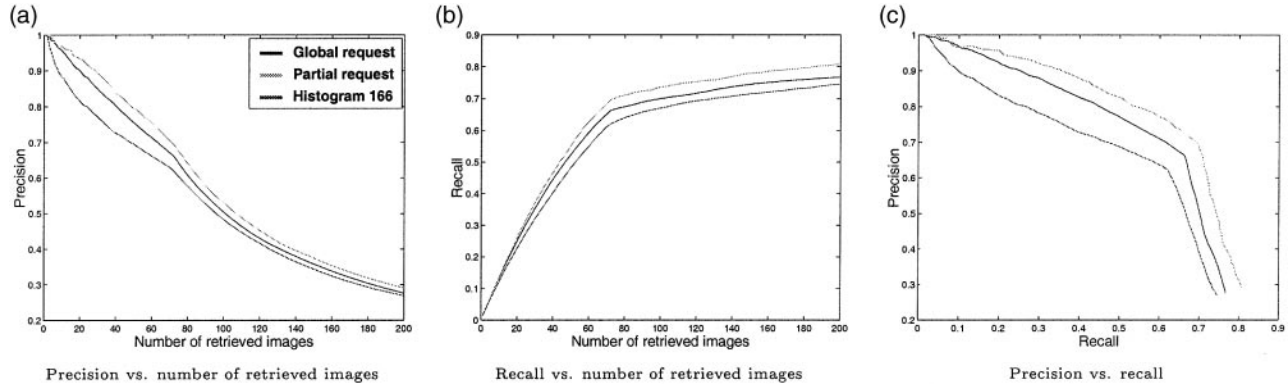

Fig. 13. Comparison between the colour histograms, our global colour signature and our partial colour signature (169 clusters in the HSV colour space) for the Columbia database. The image distributions are compared using the same L_1-metric and the quality criteria are averaged over 100 queries (1 query per object). The number of relevant images is 72 for each object.

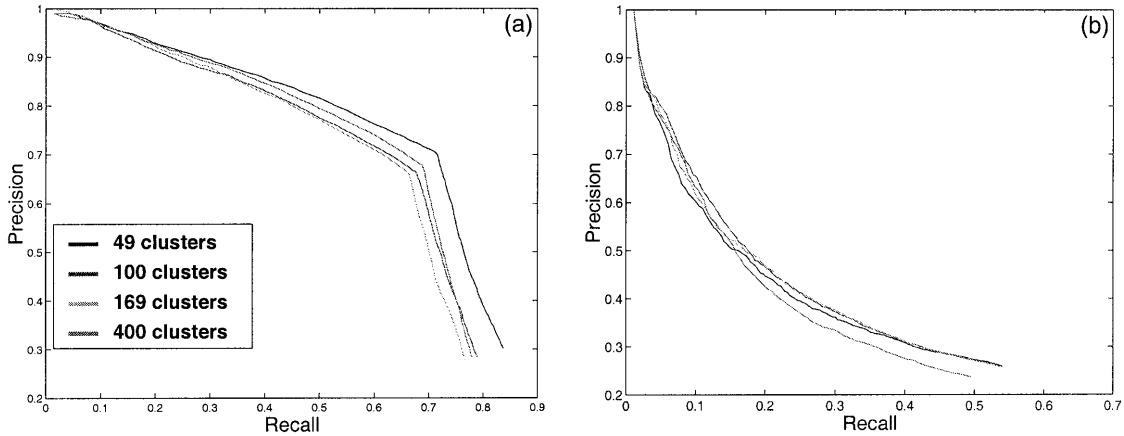

Fig. 14. Comparison of the retrieval performances obtained for classifications (in the HSV colour space) using different numbers of clusters (49, 100, 169, 400). (a) Results for the Columbia database (averaged over 72 global queries); (b) results for the general database (averaged over 105 global queries on the lion category).

as more critical in the general database than in the Columbia image collection. The search-by-similarity in this last database is not the most difficult task, because it contains isolated objects that are well discriminated due to colour attributes. This is the reason why the gap between the retrieval performances of our adaptive signature and the colour histogram is not large. Our system is, in fact, designed for the search in a general database. Nevertheless, since the content differences are larger than in a specialised image collection, the generalisation of the example provided by the user is more difficult, and the performances decrease.

Parameter Influence. After the quality assessment of the retrieval process without feedback, it is interesting to investigate the influence of the parameter.

The first studied parameter is the number of clusters (the number of neurons on the Kohonen map) or the number of bins of our signature. Figure 14 provides a comparison between the retrieval performances obtained for an increasing number of clusters.

These results show that retrieval performances are close whatever the number of clusters. The search is robust and the best performances can be attained for only 49 clusters on both collections. Our system allows us to build and use short signatures, which decreases the search duration and reduces the memory needed for the storage.

Another interesting input parameter of the system is the dissimilarity function used for signature comparison. We have tested five functions: L_1, L_2, L_∞, the Kullback–Leibler distance [36] and the cross correlation (CC)[8]. The retrieval performances are presented in Fig. 15.

Note. For the search without feedback based on one image model, the L-metrics are equivalent to our dissimilarity function d_1 (see Section 5.2).

The general tendency points to a loss in the retrieval quality when the order of the L-metrics grows. L_1 gives better results than L_2, and L_2 gives better results than L_∞. When the order is high, the L-metric becomes more sensi-

[8] To obtain a dissimilarity function, we use: 1-CC.

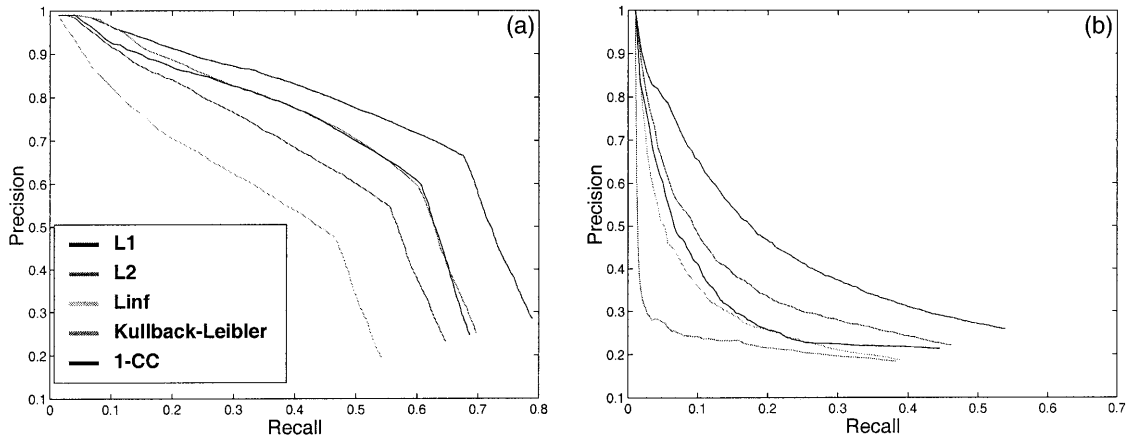

Fig. 15. Comparison of the retrieval performances obtained for five metrics. The search is based on a 100-cluster classification in the HSV colour space. (a) Results for the Columbia database (averaged over 72 global queries); (b) results for the general database (averaged over 105 global queries on the lion category).

tive to small changes between the target and the request distributions. The Kullback–Leibler distance and the cross correlation have similar behaviour and give interesting results on the Columbia digital library. Nevertheless, for the general database, the performances obtained with these two distances are very low compared to the L_1 effectiveness.

Finally, when we focus on the low recall values, the absolute performances are not very different according to the number of clusters, or according to the metric. It is important, since the user wants to retrieve a maximum of relevant images in a small set of retrieved images. Our system is robust for the low recall values, i.e. for a small set of results.

8.3. Experiments with Relevance Feedback

This section describes the relevance feedback influence on the retrieval effectiveness. After a brief exhibition of some search results and presentation of our evaluation protocol, we study the feedback contribution to the absolute performances of the retrieval, and for the competition between models.

Some Results. Figures 16 and 17 show the results of two searches with and without feedback in the Columbia database and in the ANN digital image library. The user brings an image as a query (the top left-hand image), the system retrieves an initial set of results (a), user annotates as relevant and irrelevant each of these images (the user's annotations are advisable on the images) and the system performs a refined search (b).

In Fig. 16, we see that the initial search retrieves only five 'good' images (Fig. 16(a)), and that three objects are red, whereas the query object is yellow. Actually, in this example, the classification has been made with only 25 clusters in the HSV colour space. It leads to a coarse quantification where yellow and red are mixed. But thanks to relevance feedback, all the outliers are removed (Fig. 16(b)). In the ANN image collection, the use of colour alone (without texture attributes) can lead to inconsistent retrievals (see Fig. 17(a)), but human expertise easily helps in removing these wrong images. These two examples are good illustrations of the retrieval improvement by relevance feedback. Nevertheless, success is not ensured, and the system sometimes needs more than one or two feedback steps to significantly update the results.

Evaluation Protocol. There is no common quality assessment protocol between the research teams working on image retrieval refinement by relevance feedback and the propositions made by Salton [1] in text retrieval. We suggest the use of the classical quality criteria (precision and recall) for the running of the system close to real conditions (i.e. only a few annotated images at each step).

Given n retrieved images, the system automatically labels as relevant or irrelevant (thanks to the ground truth available on the database) the first images n_l, updates the weights of the dissimilarity metric and starts a new search. The process can be iterated more than once.

Search Refinement by Relevance Feedback. Figure 18 gives the absolute performances obtained through relevance feedback for searches performed on the general database and for an increasing number of iterations. The first 30 images are annotated at each step (i.e. $n_l = 30$).

We notice that the retrieval effectiveness increases significantly thanks to relevance feedback. After one step, performances are better for low recall values (< 0.17), or for a small set of results. For five steps, the tendency of the precision-recall curve is totally different; precision tends to be equal to 1 for recall values lower than 0.2. Since we are interested in finding a high number of correct images in a small number of results, these performances are satisfying.

The recall values are always lower than 0.41. In fact, the absolute number of relevant images retrieved is limited by large content variations within this image category. The performances obtained thanks to relevance feedback show that our process is effective to retrieve more relevant images

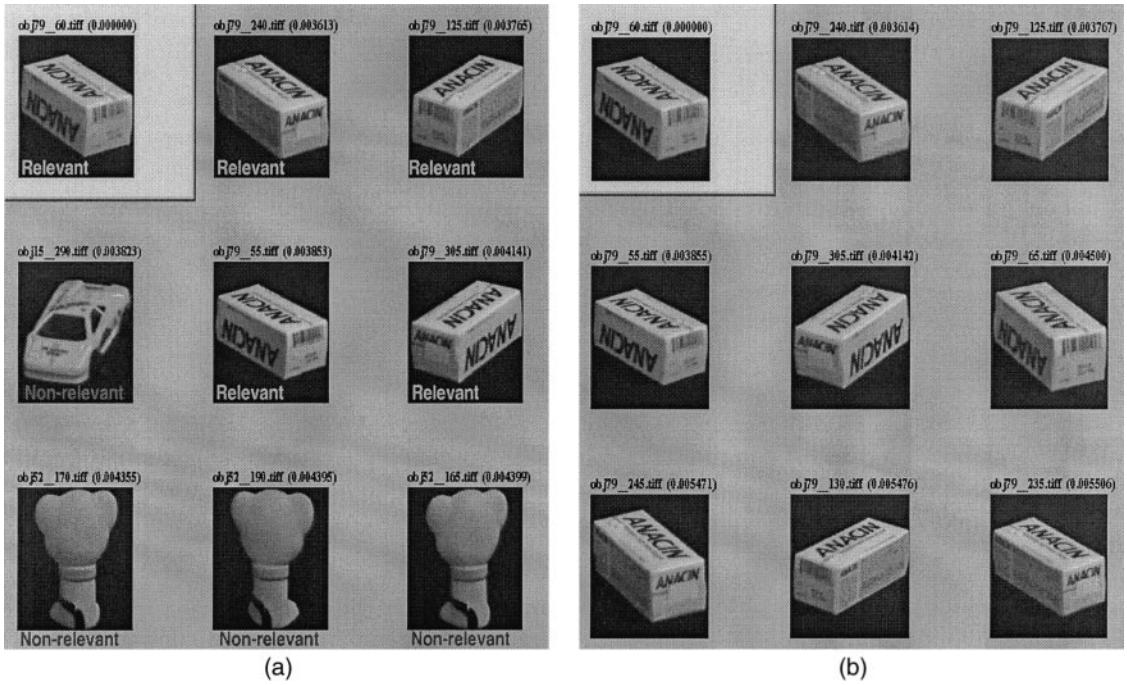

Fig. 16. An example of the relevance feedback contribution for the retrieval in the Columbia database. The search uses a global request and colour attributes (25 bins distributions in the HSV colour space). (a) Initial ranked set of results; (b) results after two feedback steps (18 annotations).

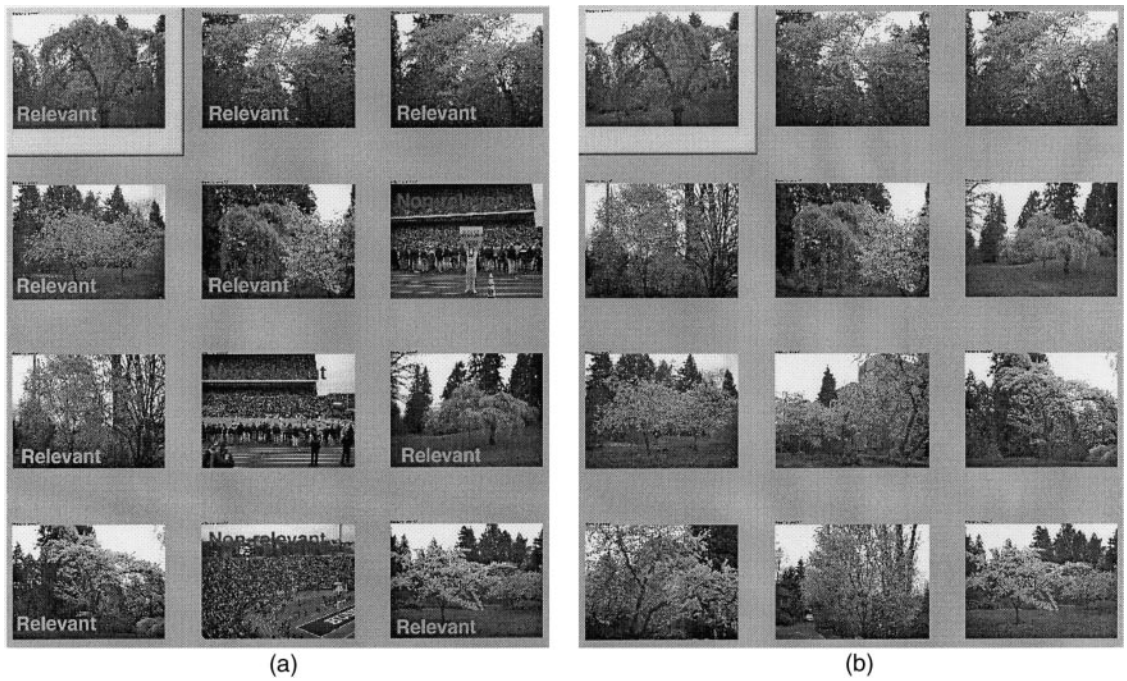

Fig. 17. An example of the relevance feedback contribution for the retrieval in the ANN database. The search uses a global request and colour attributes (100 bins distributions in the HSV colour space). (a) Initial ranked set of results; (b) results after two feedback steps (18 annotations).

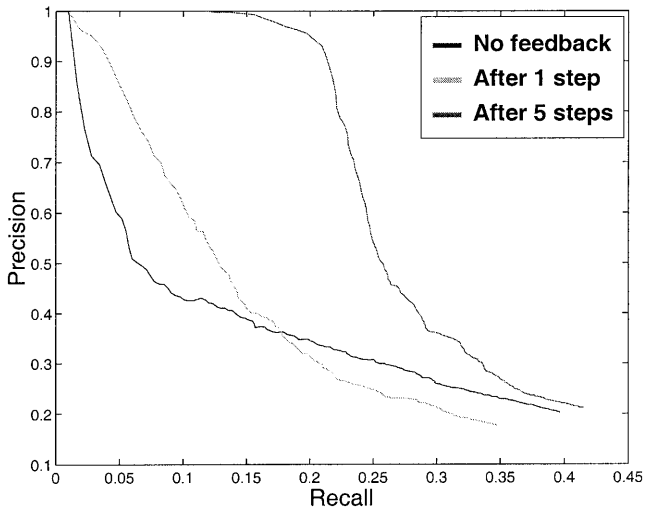

Fig. 18. Retrieval refinement by relevance feedback. Comparison of the performances without feedback, after one iteration (or one step) and after five iterations. Criteria are averaged over 30 different global queries of the bear category. Searches are based on colour (100 clusters in the HSV colour space) and texture attributes (100 clusters) and D_1 (5.3) is used as a dissimilarity function.

(the higher recall value increases), but also allows to better rank the good images in the result list.

Relevance Feedback and Competition between Models. Figure 19 compares the retrieval criteria obtained for two single image models (colour and texture), and for the competition of these two models. As for the previous tests, 30 images are annotated at each feedback step ($n_l = 30$).

Without feedback, the retrieval performances of the three models (two singles and one compound model) are weak. No clear tendency appears. The compound model is superior to the others, but for low recall values, the colour alone gives equivalent results. Moreover, the texture model becomes better than the colour one for recall values higher than 0.34. Due to relevance feedback, the absolute performances increase, and a clear hierarchy appears between the three models. The texture model is greater than the colour alone, and the competitive model overcomes the single ones. It shows that our weighted metric is correctly updated, and that the relevance feedback correctly manages the competition between the attributes.

Comparison with a Leading Relevance Feedback Method. Heinrichs et al [15] introduce a new relevance feedback method for the combination of similarities issued from different image models. The goal is to increase the weight of a model if it better ranks the relevant images than the irrelevant ones. If $\tilde{r}_i^N$ (resp. $\tilde{r}_i^R$) is the mean rank of model number i computed on the irrelevant (resp. relevant) image set, and β_i is the weight of the corresponding model, the updating rule is:

$$\beta_i^* = \beta_i \times \frac{\tilde{r}_i^N}{\tilde{r}_i^R}$$

Figure 20 shows the compared performances of Heinrichs' technique and our feedback rule.

The absolute performances given by both methods are very close. Our rule is slightly better than Heinrichs' feedback, because of the intra-model metric optimisation. Nevertheless, the difference is very hard to appreciate, because both methods provide a nice search behaviour. They do improve the precision-recall curve tendency obtained without feedback, by focusing on the most discriminant model. The important issue in regard to this comparison is that our relevance feedback is not in contradiction with another feedback technique. We still have to study the convergence duration of our rules in order to optimise the updating scheme. For example, we have to effectively set the learning rate (μ – see Sections 7.1 and 7.2) employed by the LMS rule.

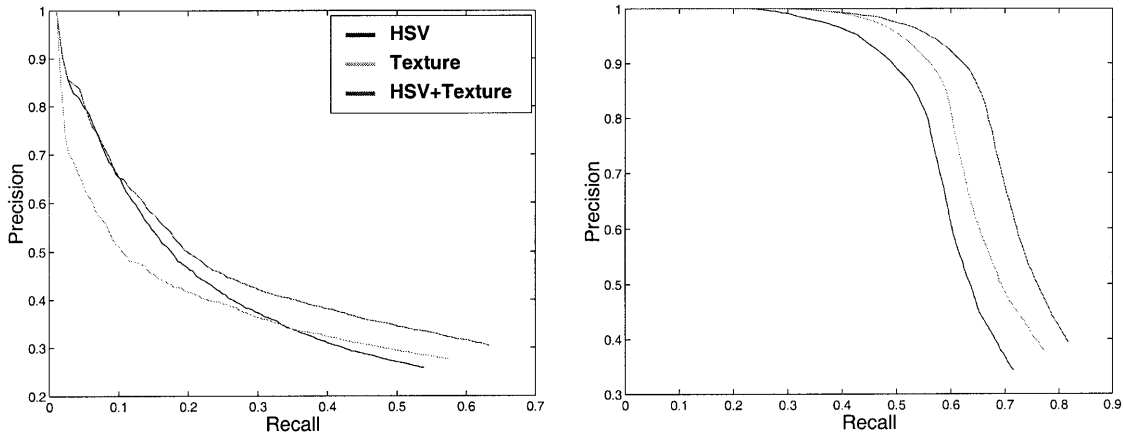

Fig. 19. Comparison of the performances for a single colour attribute (100 clusters in the HSV colour space), a single texture attribute (100 clusters) and competition between these two models. (a) Without feedback; (b) after five steps – criteria are averaged over 30 different global queries of the lion category; d_1 is used as a dissimilarity function for the single models, and D_1 is used for the competition.

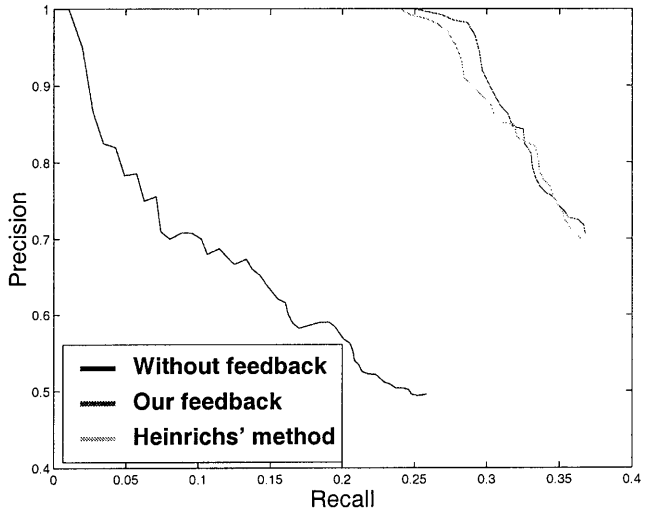

Fig. 20. Comparison of the performances of our relevance feedback rule (associated with the metric D_1) to Heinrichs' leading method for the competition between a colour model (100 clusters in the HSV colour space) and a texture model (100 clusters). Criteria are averaged over 15 different global queries of the elephant category. Five steps of 30 annotations are done for each method.

8.4. Discussion

Our system is effective for the search in the general image collection, as well as in the Columbia database. We have shown that our colour signature gives better results than a colour histogram, but we have also shown that it is robust to the choice of the dissimilarity function, and to the number of clusters used for the attribute space quantisation. This last point is important, since it means that the use of short signatures does not lead to a drop in retrieval performances. The online search is faster, and the memory space required for signature storage is smaller.

It appears that the relevance feedback is effective to refine the results of the search; it enables the user's retrieval to be less dependent on the initial query. It helps to pinpoint the user's goal through its annotations.

Our quality assessment is based on intensive and rigorous statistics. For instance, thousands of searches have been performed to obtain reliable results. Nevertheless, the protocol of evaluation is debatable, since the ground truth is subjective. To take into account the notion of inaccuracy and doubt in setting up the ground truth, the categories could be fuzzy, i.e. a given image should belong to several categories at the same time. Actually, there is a gap between the subjectivity of the retrieval process and the rigidity of the quality assessment. Given these issues, it seems to be important for the CBIR community, as noticed by Müller et al [45], to build a common test set with a strict evaluation protocol and ground truth.

9. CONCLUSION AND FUTURE WORK

We have introduced a general system for digital image retrieval. During the offline stage, an unsupervised classification is processed in each attribute space, thanks to a learning set which takes into account the collection diversity. The images are indexed by their statistical distributions computed over an irregular partition of each attribute space; the signature is self-adaptive to the database. The online search is initialised through an example image provided by the user, in which he can select a region of interest. Given the first set of results, the user has the possibility to react by specifying the relevance or irrelevance of each displayed image. This user's expertise is integrated by a relevance feedback process which tunes the dissimilarity function. The key notion of our system is the search flexibility introduced through user interaction.

An intensive and rigorous quality assessment has been carried out for two wide databases (a general one and an object database) containing approximately 10,000 images. The comparison with a colour histogram-based method shows how effective our signature is. Our system is also robust to parameter choices, particularly to the number of partitions in the attribute spaces. It enables us to use short image signatures, optimising memory space and search processing time. Otherwise, the quality assessment of the relevance feedback allowed us to quantify the result improvement via the intra- and inter-model dissimilarity optimisation.

Since our approach manages the competition between image models, an extension of our work deals with the integration of new attributes. For instance, image features taking into account the spatial information could be useful to the system.

Flexibility is also a very important aspect of the search. We think that effectiveness can be improved by navigating within the image collection. Actually, for a given query, the search is restricted (by this query) to a small area in the search space. The retrieval process will become more efficient in terms of user's satisfaction if the algorithm is able to look for groups of images scattered in the whole database. Moreover, the navigation provides a good way for the user to find a request image into the collection.

Acknowledgements

All our programs have been developed using the Image Understanding Environment TargetJr: http:/www.targetJr.org/

The aerial images are provided by IGN (Institut Géographique National).

References

1. Salton G, Buckley C. Improving retrieval performance by relevance feedback. J Am Soc Infor Sci, 1990; 41(4):288–297
2. Nastar C, Mitschke M, Meilhac C. Efficient query refinement for image retrieval. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98), Santa Barbara, CA, June 1998; 547–552
3. Nastar C, Mitschke M, Meilhac C, Boujemaa N. Surfimage: a flexible content-based image retrieval system. ACM-Multimedia 1998, Bristol, UK, September 12–16 1998; 339–344

4. Pentland A, Picard R, Sclaroff S. Photobook: Tools for content-based manipulation of image databases. SPIE Conf. on Storage and Retrieval of Image Databases II, San Jose, CA, 1994; 34–47
5. Flickner M. Query by image and video content: the qbic system. IEEE Computer 1995; 28(9):23–32
6. Kelly PM, Cannon TM, Hush DR. Query by image example: the candid approach. SPIE Storage and Retrieval for Image and Video Databases III, 1995; 2420:238–248
7. Ma WY, Manjunath BS. Netra: A toolbox for navigating large image databases. ACM Multimedia Systems 1999; 7(3):184–198
8. Cox IJ, Miller ML, Minka TP, Yianilos PN. An optimized interaction strategy for bayesian relevance feedback. CVPR'98, Santa Barbara, CA, June 1998; 553–558
9. Geman D, Moquet R. A stochastic feedback model for image retrieval. RFIA'2000, Paris, France, February 2000; III:173–180
10. Miller W, Squire DM, Muller H, Pun T. Hunting moving targets: an extension to bayesian methods in multimedia databases. Technical report 99.03, Computer Vision Group, Computing Science Center, University of Geneva, Genve, Switzerland, July 1999
11. Veltkamp RC, Tanase M. Content-based image retrieval systems: A survey. Technical report UU-CS-2000–34, Department of Computing Science, Utrecht University, October 2000
12. Schröder M, Rehrauer H, Seidel K, Datcu M. Interactive learning and probabilistic retrieval in remote sensing image archives. IEEE Trans Geoscience and Remote Sensing September 2000; 38:2288–2298
13. Minka TP, Picard RW. Interactive learning with a "society of models". Pattern Recognition 1997; 30:565–581
14. Carson C, Thomas M, Belongie S, Hellerstein JM, Malik J. Blobworld: A system for region-based image indexing and retrieval. Third Int Conf on Visual Information Systems, June 1999
15. Heinrichs A, Koubaroulis D, Levienaise-Obadia B, Rovida P, Jolion JM. Image indexing and content based search using pre-attentive similarities. RIAO2000, Paris, France, April 2000; 2:1616–1631
16. Biernacki C, Mohr R. Indexation et appariement d'images par modèle de mélange gaussien des couleurs. Technical report RR-3600, INRIA, January 1999
17. Nastar C. Indexation d'images par le contenu: un etat de l'art. In CORESA'97, March 1997
18. Wood MEJ, Campbell NW, Thomas BT. Iterative refinement by relevance feedback in content-based digital image retrieval. ACM Multimedia 98, Bristol, UK, September 1998; 13–20
19. Smith JR, Li CS. Image classification and querying using composite region templates. Computer Vision and Image Understanding 1999; 75(1/2):165–174
20. Malki J, Boujemaa N, Nastar C, Winter A. Region queries without segmentation for image retrieval by content. Third International Conference on Visual Information Systems (Visual'99), Amsterdam, The Netherlands, June 2–4 1999; 115–122
21. Schmid C, Mohr R. Matching by local invariants. Technical report RR-2644, INRIA, August 1995
22. Mindru F, Moons T, Van Gool L. Recognizing color patterns irrespective of viewpoint and illumination. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), Fort Collins, CO, June 1999; 1:368–373
23. Huet B, Hancock ER. Relational histograms for shape indexing. Int Conf on Computer Vision 1998; 563–569
24. Gabor D. Theory of communication. J Institute of Electrical Engineers, January 1946; 93(21)(Part III):429–457
25. Manjunath BS, Ma WY. Texture features for browsing and retrieval of image data. IEEE Trans Pattern Analysis and Machine Intelligence (Special Issue on Digital Libraries) August 1996; 18(8):837–842
26. Rubner Y. Perceptual Metrics for Image Database Navigation. PhD thesis, Stanford University (http://robotics.stanford.edu/~rubner/), May 1999
27. Pauwels EJ, Frederix G. Finding salient regions in images: Non-parametric clustering for image segmentation and grouping. J Computer Vision and Image Understanding July/August 1999; 75(1/2):73–85
28. Kohonen T. The self-organizing map. Proc IEEE 1990; 78(9)
29. Columbia object image library (coil-100). Department of Computer Science, Columbia University, http://www.cs.columbia.edu/CAVE/research/softlib/coil-100.html
30. Smith JR, Chang SF. Visualseek: A fully automated content-based image query system. ACM Multimedia Conference, Boston, MA, November 1996; 87–98
31. Westerveld T. Image retrieval: Content versus context. RIAO'2000, April 2000; 276–284
32. Nastar C. The image shape spectrum for image retrieval. Technical report RR-3206, INRIA, July 1997
33. Del Bimbo A, Pala P. Visual image retrieval by elastic matching of user sketches. IEEE Trans PAMI February 1997; 19(2):121–132
34. Schweitzer H. Organizing image databases as visual-content search trees. Image and Vision Computing 1999; 17:501–511
35. Yianilos PN. Data structures and algorithms for nearest neighbor search in general metric spaces. Fourth ACM-SIAM Symposium on Discrete Algorithms January 1993; 311–321
36. Heinrichs A, Koubaroulis D, Rovida P, Levienaise-Obadia B, Jolion JM. Robust image retrieval in a statistical framework. Technical report rr-04.99, Ruhr-Universitat Bochum, Germany, April 1999
37. Sarrut D, Miguet S. Similarity measures for image registration. First European Workshop on Content-Based Multimedia Indexing, Toulouse, France, October 1999; 263–270
38. Squire DM, Muller W, Muller H, Pun T. Content-based query of image databases: inspirations from text retrieval. Pattern Recognition Letters (special edition for SCIA'99) 2000; 21(13–14):1193–1198
39. Rui Y, Huang T, Mehrotra S, Ortega M. A relevance feedback architecture for content-based multimedia information retrieval systems. IEEE Workshop on Content-Based Access of Image and Video Libraries 1997; 92–89
40. Meilhac C, Nastar C. Relevance feedback and category search in image databases. IEEE International Conference on Multimedia Computing and Systems (ICMCS'99), Florence, Italy, June 1999; 512–517
41. Widrow B, Hoff ME. Adaptive switching circuits. IRE WESCON, New York, 1960; 96–104
42. Vision texture database (vistex). Vision and Modeling group, MIT Media Lab, http://vismod.www.media.mit.edu/vismod/imagery/Vision Texture/
43. Annotated groundtruth database (ann). Department of Computer Science and Engineering, University of Washington, http://www.cs.washington.edu/research/imagedatabase/
44. Smith JR. Image retrieval evaluation. In IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'98) June 1998; 112–113
45. Müller H, Müller W, Squire DM, Marchand-Maillet S, Pun T. Performance evaluation in content-based image retrieval: Overview and proposals. Pattern Recognition Letters 2001 (to appear)

Jérôme Fournier is now doing a PhD after completing his masters degree in image and signal processing. He is working with ETIS (image and signal processing research team), a French laboratory associated with the CNRS. He specialises in content-based image indexing and retrieval. His work essentially deals with flexible image signatures and adaptive search guided by the user.

Matthieu Cord received the PhD degree in image processing from the University of Cergy-Pontoise, France, in December 1998. He was a post-doctoral researcher at the Katholieke Universiteit Leuven, Belgium, where he worked on the 3D modelling of aerial images. He is currently an associate professor with the ETIS Laboratory at the University of Cergy-Pontoise, France, in the image processing research group.

Sylvie Philipp-Foliguet taught mathematics in high schools from 1977 to 1984. She received her PhD degree in computer science from the University of Paris 6, France, in 1988. The subject of her thesis was the detection of defects in radiographic images of weld, by texture analysis. She is currently a Professor at the ENSEA of Cergy-Pontoise, and she works with ETIS, where she manages the image processing research group. Her main interest fields are segmentation, image analysis and interpretation, and more recently, content-based image retrieval.

Correspondence and offprint requests to: J. Fournier, ENSEA/Université de Cergy-Pontoise, 6, av. du Porceau, F95014 Cergy-Pontoise cedex, France. Email: fournier@ensea.fr