Interactive Exploration for Image Retrieval.

Matthieu Cord, Jérome Fournier, Philippe-Henri Gosselin, Sylvie Philipp-Foliguet

Interactive exploration for image retrieval

Matthieu Cord, Sylvie Philipp-Foliguet, Philippe-Henri Gosselin, Jérome Fournier

ETIS – CNRS UMR 8051
6 av. du Ponceau
F 95014 Cergy-Pontoise Cedex, France
E-mail: cord@ensea.fr, philipp@ensea.fr

Abstract

In this paper, we present a new version of our content-based image retrieval system RETIN. It is based on adaptive quantization of the color space, together with new features aiming at representing the spatial relationship between colors. Color analysis is also extended to texture.

Using these powerful indexes, an original interactive retrieval strategy is introduced. The process is based on two steps for handling the retrieval of very large image categories. First, a controlled exploration method of the database is presented. Second, a relevance feedback method based on statistical learning is proposed. All the steps are evaluated by experiments on a generalist database.

1 Introduction

The recent domain of image retrieval in large databases has induced a revision of the topics of image processing and pattern recognition. Image retrieval and extraction of visual information from image databases are useful in many applications. Even if there are many different application contexts, two kinds of search are usually distinguished [31]: target search and category search. Target search aims at retrieving one or a few particular images in the database. Category search aims at retrieving all the images belonging to a given category. In the latter case, the major difficulty is that images belonging to the same semantic category may have very different visual contents.

In image retrieval, fully automatic systems have given poor results. In interactive systems, the user is requested to manage the search within the database. For instance, the user may interactively annotate the results as relevant or irrelevant to his query. The system integrates these annotations through a relevance feedback. The main idea of the relevance feedback is to use the information provided by the user to improve the system effectiveness. One reason for this new need of interactivity is definitively the huge size and the

diversity of the data to be mined. Another reason is the well-known semantic gap [28] between the numerical data and their semantic meaning. The user is looking for an image or a set of images with semantics, for instance a type of landscape, whereas actual systems deal with color or texture features. The problem is even more complicated when the user is looking for a particular building, or a person, or for an abstract concept such as unemployment. These different levels of abstraction have been analyzed in [10].

All CBIR systems have to deal with two major challenges: efficient image coding and effective visual information retrieval, working with user interaction to bridge the semantic gap. In this paper, we propose solutions for both problems.

The first problem has been thoroughly studied in the first retrieval systems [24][33]. For generalist databases, the extraction of low-level features used in human pre-attentive vision, like color, texture and shape, has concentrated lots of efforts [17][36]. Concerning texture, the most popular features are moments, and features based on cooccurrence matrices, on Gabor filters or on wavelet decompositions [23]. Wavelet-based methods have been compared in [22], and authors have concluded that Gabor wavelets were the most effective. Shape features are numerous, but they depend on prior extraction of regions from the image. Concerning color features, Schettini et al. have gathered color signatures and similarity metrics employed in various indexing systems [30]: all colorimetric spaces, from RGB to HSV to CIELab or CIELuv are used by one or other of these systems. Nevertheless, the authors point out that color alone is not sufficient to index large image databases. Spatial relationship between regions are sometimes encoded in order to represent image composition. Using these features or primitives, signatures are computed.

In this paper, we present an image representation which encodes color and texture but also the spatial relations between color regions (or texture regions). For color and texture analysis, the feature space quantization is significant. To handle this problem, we propose a dynamic feature quantization scheme of the whole database. Instead of using prior criteria, experiments in the image retrieval context are carried out to select the best quantization method and its parameters.

Concerning the second challenge – effectiveness of the retrieval task –, two types of interactive approaches are usually considered [39], the geometrical approach (as search-by-similarity), and the statistical approach (as relevance function estimation or binary classification):

- The geometrical approach of relevance feedback is based either on the adaptation of the initial query or on the updating of the similarity function. In this approach, initially used for document retrieval [25], the query is represented by a vector in the feature space, and the similarity function allows to compare any image to the query. The adaptation then consists in moving the query vector, or in changing similarity parameters. Sometimes both are combined [20]. Similarity updating may be seen as a shape deformation of the search neighborhood around the query.

- The approach by relevance function estimation aims at associating a score to each image of the database, expressing the relevancy of the image to the query. A Bayesian context is often used, and the probability density function is updated considering the user annotations. The probability function may be uniformly initialized and iteratively refined in order to emphasize relevant images [3][9]. The approach using data classification treats the relevance feedback problem as a supervised learning problem. A binary classifier is learnt by using all relevant and irrelevant annotated images as input training data [39, 37].

We present a new version of RETIN, our content-based image retrieval system [12]. Several modules have been developed to deal with target and category searches using both geometrical and statistical approaches.

We first present our search-by-similarity approach based on similarity updating. Dedicated to target search, this strategy has been successfully compared to some of the best relevance feedback strategies [12].

We have considerably improved the first version of our system RETIN in order to deal with large category searches. In this context, a category is defined as a set of images with common semantic characteristics. All the relevant images are not always gathered in a single mode in the feature space. The problem is to catch all these modes. Our strategy for category search is based on a statistical approach specially dedicated to explore the database and to track multi-modal distributions. To be able to catch all the modes of a category, we propose to explicitly take into account of the distribution of the data in the feature space. The stochastic database exploration is based on a sampling of a relevance density function, and a multi-modal similarity function. After a few database exploration steps where many different images of the searched category are collected, the exploitation may start. An learning strategy based on SVM classification is then used to efficiently track large image categories. The latter step is done as soon as the exploration strategy has provided enough examples of the category.

To summarize, the main characteristics of our indexing system are the adaptive feature quantization (Section 2), and the computation of signatures composed of color, texture and spatial relationship distributions (Sections 3 and 4). Concerning the retrieval engine, two original interactive strategies are proposed for target and category searches (Sections 5 and 6). We design a stochastic exploration scheme that quickly grasps the user query concept (or semantic query) in Section 6. We model relevance feedback either by updating the similarity function or by using a binary classifier (Section 7). In Section 8, experiments on a generalist database are reported.

2 Color quantization

As color information is usually represented by a huge number of classes (often over 16 millions), it is necessary to reduce this number by a color quantization process. This quantization may be achieved by a static or

by a dynamic splitting of the color space. The difference between these two approaches is that the first one is independent of the data, whereas the second one takes into account of the distribution of colors in the feature space. The simplest method to reduce the number of classes is to split the color space into a reduced number of classes or bins. There are many methods of regular or irregular splitting. For example, the HSV space is split into 166 bins in Visual Seek [33]: the intensity axis is split into 3 intervals, the saturation axis into 3 intervals, the hue angle into 18 intervals, and the central axis of the cylinder is split into 4 bins for the gray levels.

Dynamic quantization depends on data, either globally by taking into account all images of the database or individually for each image. Classification methods may be used to split the feature space but some adaptation is necessary due to the size of the data (the number of pixels in the database). An alternative consists in making clusters independently for each image. For instance, Rubner [26] uses a color palette adapted for each image. The use of image-adapted methods implies the introduction of specific distances, since the number and the significance of the bins can be different from an image to another.

After quantization, a signature may be affected to each image. Usually, it consists in the statistical distribution of the classes (estimated by histograms), but it can be reduced to a few features such as moments (mean, variance), covariance matrices, or distributions restricted to the most frequent classes present in the image.

In RETIN, we use signatures which are statistical distributions of colors resulting from a dynamic quantization of the whole database. For the dynamic quantization, we use the $k-$means method. In the $k-$means unsupervised learning algorithm, the clusters are automatically carried out using the pixels from all the database images. We use an adaptive algorithm, which means that the whole set of pixels is not simultaneously processed, which would need an enormous memory capacity, but the pixels are sequentially proposed to the clustering process. Pixels are randomly sampled from the database and processed. The only parameter is the number of classes (color bins) which must be previously fixed. To speed up the process, the database images are sequentially processed, and the random selection of pixels is done image per image.

In the following subsections, we first determine the appropriate number of color bins for the $k-$means classifier, then we compare our adaptive quantization to a static quantization, working in HSV space [33].

As none of the numerous color spaces has proved its superiority over the others for image indexing, we have chosen HSV space in order to compare with static quantization in the same space.

The CBIR experiments have been performed on the *Corel* database to compare static and dynamic quantization. This database, composed of 6000 images, is divided into categories, in order to use the classical criteria of precision and recall [32] for quality assessment. Performances are established independently for each category, but are averaged on 20 queries of the same category. In the follow-up, the displayed curves are typical results obtained for many categories[1].

[1]Alternatively, the number of images per category could be used to draw new evaluation curves, in the precision=recall

In all the experiments, the distance used to compare histograms is L_1 distance. Comparisons of various distances have been carried out in [2] [27]. Most of the time, L_1 distance is one of the most effective among geometric distances, statistical tests, and other dissimilarity measures.

2.1 Histogram size selection

Some theoretical rules may be used to tune up the resolution and the number of histogram bins. Sturges's or Scott's rules cited in [2] allow to avoid over or under-quantization. In image retrieval context, Brunelli and Mich have evaluated many feature histograms and they concluded that low-resolution histograms (with small bin numbers) are reliable [2]. For color histograms, Tran and Lenz suggest to use around 30 bins [38].

In this paper, we set the number of clusters from experimentations. Moreover, tests will be performed in section 8 with a complete retrieval system, including the feedback loop. Here, we just examine the influence of the number of classes on the retrieval results. The $k-$means algorithm is evaluated for different number of classes, from 8 to 400. Fig. 1 displays precision/recall curves (averaged over 20 queries) obtained by the $k-$means algorithm. Except for 8 classes, for which results are lower, there is no significant difference between the other four curves. As the size of signatures is related to the retrieval time, small signatures, and small number of classes have to be favored.

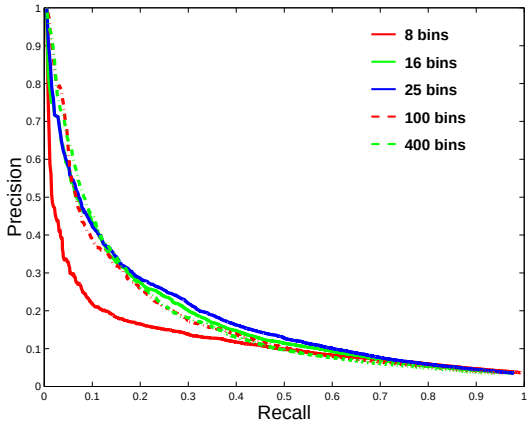


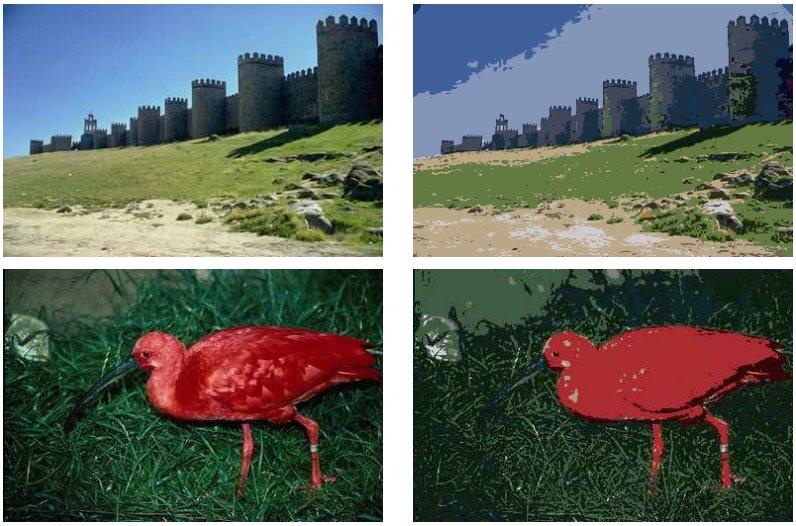
Figure 1: Precision/recall curves for different numbers of classes (sunset category).

Fig. 2 displays a palette of 25 colors which have been obtained from quantization using 500 millions of pixels randomly selected in the *Corel* database. Displayed colors correspond to the 25 class centers obtained by the $k-$means algorithm. Fig. 3 displays two examples of images before and after quantization with these 25

plane [19].

colors.

Figure 2: Color palette resulting from our adaptive database quantization. The 25 clusters are represented by their barycenters in the color feature space.



Original image Color quantized image

Figure 3: Two examples of images before and after quantization of HSV color space by adaptive $k-$means in 25 classes. The pixel values are replaced by the color of the class center.

2.2 Color quantization method selection

We compare static and dynamic quantization results. For the static quantization, we have used the method proposed by Smith and Chang on 166 bins [33] from HSV cylindric space, and our algorithm with 25 classes for the dynamic quantization. Fig. 4 displays the average precision/recall curves for 20 queries (from one category). Although the number of classes is much lower (25) for the $k-$means classifier than for the static quantization algorithm (166), the performances of the dynamic classifier are better.

We observed this behavior for many categories; in order to provide statistics for the whole database, we present capacity curves [35]. The capacity curve is defined as the histogram of dissimilarities between all pairs of images of the database [35]. It allows to appreciate the dispersion of signatures within the search

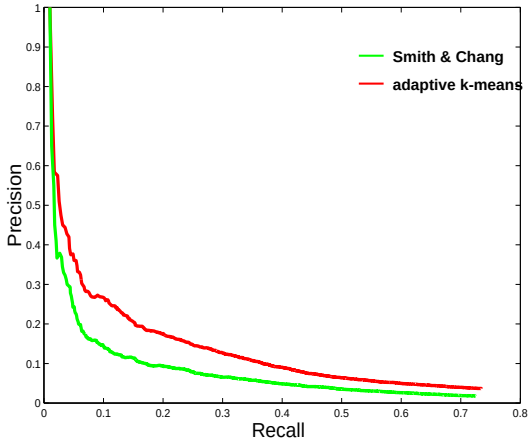


Figure 4: Precision/recall curves for two quantization methods: Smith and Chang with 166 bins and our adaptive method with 25 colors.

space. More this dispersion increases, better is the discrimination quality of the signature. They have been computed for the color histogram of Smith and Chang and for our color signature (fig. 5). One can observe that image dispersion in the search space is larger with our color signature than with Smith and Chang histogram. The discriminatory ability of our approach is higher, which confirms previous results.

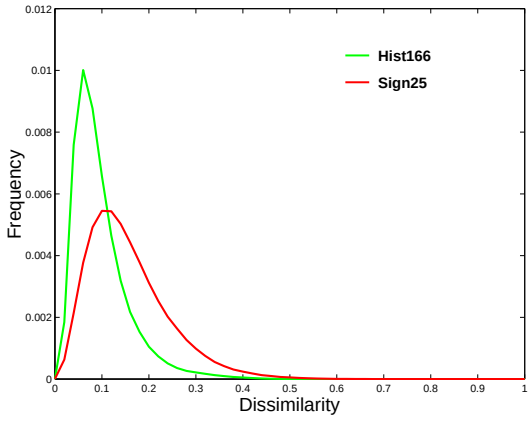


Figure 5: Capacity curves for the color histogram in 166 classes of Smith and Chang and for our color signature (color space HSV) of 25 classes (Sign25).

Our approach by dynamic quantization of the feature space provides a more effective indexing of the database, compared to a static histogram; image signatures are better scattered in the search space and retrieval results are better. A major advantage of the dynamic approach is the reduction of the size of the signature without performance deterioration.

Even if the results depend on the image categories and on the database, all our experiments show that for a generalist database (around $10,000$ images), a small number of classes obtained by a dynamic clustering of the database is sufficient to build efficient color signatures. We have adopted this dynamic quantization in the RETIN system with 25 color classes (as the default value).

3 Spatio-colorimetric indexing

Color distribution is not sufficient to encode all color information, because it gives no information about spatial localization of each color in the image. Some methods integrate spatial information, for example color correlograms [18], spatio-colorimetric histograms [6] and Composite Region Templates (CRT) [34]. Another solution is to segment the image and to store spatial relationship between regions. However, automatic segmentation of a whole database is not an easy task. Using manual annotations, complex spatial relationships have been modeled and exploited in a pictorial data retrieval context [1]. Starting from the works of Smith and Li [34], we propose a new spatio-colorimetric indexing without segmentation.

3.1 Spatio-colorimetric quantization without segmentation

The main idea is to store the vertical color transitions within the image. In generalist image databases, only vertical transitions are of importance. The reason is that a symmetry over an horizontal axis greatly changes our perception of the image, while a vertical symmetry weakly changes it: in landscape images, the sky is usually on the top of the image !

Instead of segmenting the image, we start from the quantized image with N color classes as explained in section 2.1, and we split it into rectangular blocks without overlap. Each block is then represented by the most frequent color class in the block. Resulting block-image is like a low-resolution version of the quantized image (see Fig 6). The number of blocks is a parameter of the method which must be chosen according to the size of the image objects. Tests have been performed and are presented in section 3.2. The frequency of top-down transitions of colors between adjacent or not-adjacent blocks belonging to the same block column is then computed and stored in a matrix. Unlike CRT technique, transitions between blocks of same color are counted only if they are separated by at least one block from another color. This introduces scale invariance since only transitions are counted, and robustness towards the block size, since adjacent blocks of same color are not counted: over-segmentation in small blocks is thus overcome.

One example of block image and its matrix of color transitions are displayed in Fig. 6. The matrix is mainly made up of a large peak corresponding to blue/white transitions, which represents the vertical sky/snow transitions in the image.

Transition matrices are large and very sparse. It would be expensive to keep the whole matrix as signature,

so the information contained in matrices is reduced to N^2 components by PCA.

Original image | Block image | 3D representation of colorimetric transition matrix

Figure 6: Example of transition matrix with 15×15 blocks.

3.2 Method tuning and comparative results

The only parameter of our spatio-colorimetric signature is the number of blocks. Fig. 7 compares retrieval results for several numbers of blocks from 5×5 to 45×45. Curves show that 15 or 30 is a good choice for the number of blocks. The more image is degraded by a coarse splitting, the more performances decrease. On the other hand, using more blocks does not improve results. Considering the mean size of objects contained in the *Corel* database, which is typical of generalist image databases, the number of blocks can be fixed to 15×15.

We have carried out comparisons between signatures using CRT and our spatio-colorimetric signatures. In Fig. 8, precision/recall curves have been obtained for 20 queries with the CRT signature and with our spatio-colorimetric signature with 225 (15×15) blocks, and after reduction to 25 classes. In order to respect the original CRT method, we have used the similarity function of Smith and Li for CRT retrieval [34]. Our signature gives the best results. The reasons are that we have a better color adaptation to data through the dynamic color quantization and a better spatial adaptation thanks to the splitting into small blocks, which is more accurate than the coarse segmentation proposed by Smith and Li. Over-segmentation, which could be criticized, is not a drawback in our scheme, because pairs of adjacent blocks with the same color are not counted.

In Fig. 9, we reported retrieval results when the system uses either only the color signature, or both color and spatio-colorimetric signatures. The ten most similar images are displayed in decreasing rank of similarity.

[2] N is the number of color clusters.

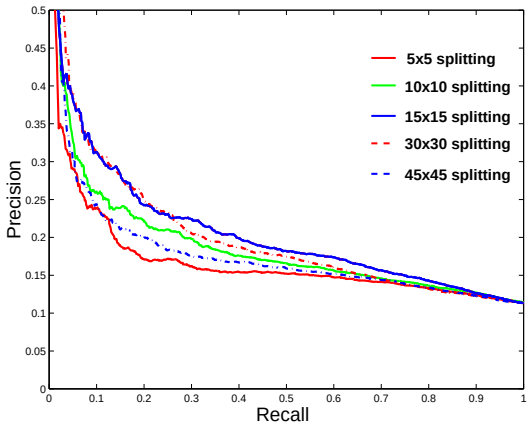


Figure 7: Precision/recall curves for various splittings using our spatio-colorimetric signature (landscape category).

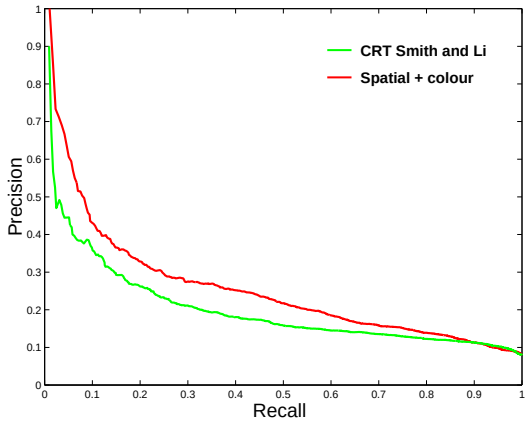


Figure 8: Precision/recall curves using the Smith and Li signature and our spatio-colorimetric signature (elephant category).

The spatial information clearly improves the results in this search, moving away images of mountains, where blue areas are not in the center of the image.

(a) Without spatial information (b) With spatial information

Figure 9: Top results for a search of doors in the *Corel* database. Images are ranked by decreasing similarity from top to bottom, and from left to right. The query is the top left hand image.

4 Texture

The same principle of quantization can be applied to any feature space. For example, texture is often represented by wavelet coefficients or by features obtained with Gabor filters. We use a Gabor filter bank of three frequencies and four orientations, which leads to a 12-dimension vector for each pixel. Quantization is performed by the $k-$means algorithm which in this case works in a 12-dimension space (3 frequencies and 4 orientations) instead of the 3-dimension HSV space. Two examples of quantization of the texture space are displayed in Fig. 10.

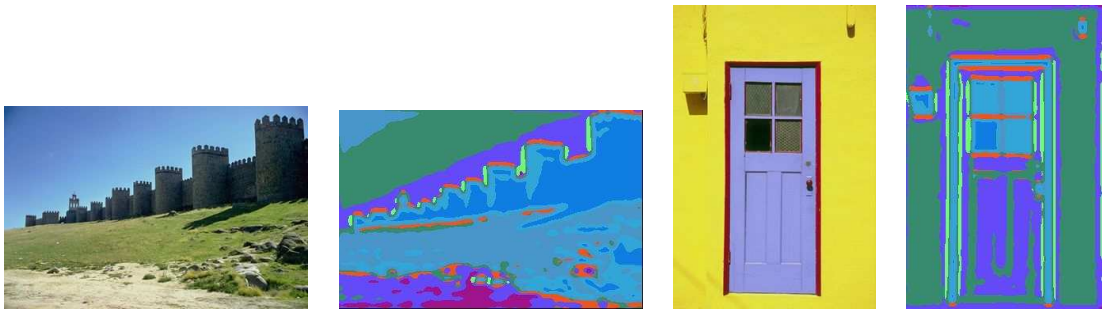


Figure 10: Two examples of images before and after quantization of the texture space by adaptive $k-$means into 25 classes. A color is randomly attributed to each texture class.

As for color composition, vertical transitions of textures can be stored in a vector representing texture image composition.

To summarize, our signature is made of four vectors, two vectors dedicated to color and two vectors dedicated to texture. The first vector represents the color distribution obtained by $k-$means clustering from HSV space, as explained in section 2.2. The second vector represents spatio-colorimetric transitions as presented in section 3. The other vectors represent texture in the same way as color. In order to easily combine similarities in various spaces, we take the same number of clusters for all the spaces. Experiments on the *Corel* database have shown that $N = 25$ classes (for any feature space) realizes a good tradeoff between the size and the richness of the resulting signature. We have adopted this value as the default one in RETIN.

5 Target search: similarity updating

5.1 Introduction

Target search strategy is working as follows: the user presents an example of the images he is looking for, and the system extracts from the database the images most similar to the query. Given a set of results, the user indicates for each image if it is relevant or irrelevant. Relevance feedback is then applied. Two main approaches can be distinguished and combined. The first one is directly inspired by text retrieval and consists in query refinement, i.e. a mean query is computed from relevant and irrelevant examples provided by user [28]. The second approach is similarity updating. For instance, some techniques refine results through tuning of weights associated to each feature space. Actually, feature weights are either manually tuned by user [4] or automatically updated via user annotations [29].

Our CBIR system RETIN includes a relevance feedback stage with similarity refinement [11]. Our similarity function is first introduced, and the feedback scheme managing competition between features is detailed.

5.2 Similarity updating strategy

The similarity is computed in each feature space and the set of similarities is then merged. We use a hierarchical model [29] where merging is achieved by a linear combination of all the feature space similarities.

The system compares a query image R to any image I_i of the database, these images are indexed by M statistical distributions (one for each feature) of N classes respectively noted $\boldsymbol{R}$ and $\boldsymbol{I}_i$: $\boldsymbol{R} = \{R_k(q), 1 \leq q \leq N, 1 \leq k \leq M\}$ and $\boldsymbol{I}_i = \{I_{ik}(q), 1 \leq q \leq N, 1 \leq k \leq M\}$. The similarity is computed as a double

weighted sum:

$$S(\boldsymbol{R}, \boldsymbol{I}_i) = \sum_{k=1}^{M} \beta_k s_k(\boldsymbol{R}, \boldsymbol{I}_i) \ \text{, with } s_k(\boldsymbol{R}, \boldsymbol{I}_i) = 1 - \sum_{q=1}^{N} \alpha_{kq} \left| R_k(q) - I_{ik}(q) \right| \tag{1}$$

with $\beta_k \in I\!R^+$, $\alpha_{kq} \in I\!R^+$. Weights β_k (resp. α_{kq}) manage the competition between features (resp. between bins), they are normalized: $\sum_{k=1}^{M} \beta_k = 1$ and $\sum_{q=1}^{N} \alpha_{kq} = 1$ ($\forall k \in [1, M]$). With the signature normalization, we also have: $0 \leq s_k(\boldsymbol{R}, \boldsymbol{I}_i) \leq 1 \ \forall k$ and consequently: $0 \leq S(\boldsymbol{R}, \boldsymbol{I}_i) \leq 1$. The dissimilarity function may be deduced from the similarity one: $D(\boldsymbol{R}, \boldsymbol{I}_i) = 1 - S(\boldsymbol{R}, \boldsymbol{I}_i)$.

At the beginning, weights are equal and normalized. After computing global similarities between each image and the query, the images the most similar to the query are displayed. After user annotations, the set of relevant ($\mathcal{E}^+$) and irrelevant ($\mathcal{E}^-$) images constitute a learning set ($\mathcal{E}$). It is used to update weights by a LMS optimization. Therefore, it is easy to compute both α and β parameter updating [12]:

Given a learning rate μ ($\mu > 0$), one iteration of feedback for $I_i \in \mathcal{E}$ is:

$\beta_k \leftarrow \beta_k + \mu \left(\mathbb{1}_{I_i \in \mathcal{E}^+} - S(\boldsymbol{R}, \boldsymbol{I}_i) \right) s_k(\boldsymbol{R}, \boldsymbol{I}_i)$

$\alpha_{kq} \leftarrow \alpha_{kq} + \mu \left(\mathbb{1}_{I_i \in \mathcal{E}^+} - S(\boldsymbol{R}, \boldsymbol{I}_i) \right) \beta_k \left| R_k(q) - I_{ik}(q) \right|$

6 Category search: on-line semantic query learning

Search-by-similarity strategies with relevance feedback are well adapted for target search. To retrieve a few images close to a query image, our search-by-similarity method is effective. The problem is much more complicated when the user is looking for large image categories. In this case, all relevant images have common semantic characteristics but are not always gathered in a single mode in the feature space. The problem is to catch these various modes. Without specific strategy, search-by-similarity methods only retrieve images close to the first query, and so, it is very hard to track other modes of the searched category.

Any system needs an efficient strategy for exploring the database in order to catch complex image category distributions. Statistical learning approaches which perform binary classification do not really manage exploration. They need enough initial training data in several modes in order to get good classifications [37]. Chang proposes a two-step sequential process to get some relevant images before doing the classification [5]. Bayesian framework has been proposed [9][13] for relevance feedback. Some kind of exploration is implicitly managed, but the goal is not to retrieve categories, and the exploration is not easy to tune.

We propose a statistical approach to explore the database and to track multi-modal distributions. The basic idea is to modify the selection scheme (based on similarity ranking). A relevance probability is attributed to each image. The system uses this probability to sample and display new images. The probability function is defined to ensure that, during the first steps of a search session, any image, even far from the query, could

be selected. When starting a retrieval session with one image from one mode, this strategy makes possible to select images from other modes.

This approach allows us to have a straight control of the exploration with intuitive parameters very easy to tune. This strategy is inspired by simulated annealing techniques [21].

6.1 Stochastic exploration approach

Let us note $\mathcal{SQ}$ (for semantic query), the set of L images that have been annotated as relevant since the beginning of the retrieval session, $\mathcal{SQ} = \{R_l, 1 \leq l \leq L\}$.

The idea is to assign to each image of the database a probability to be relevant towards the searched category. A Boltzmann distribution on the dissimilarity $D()$[3] is then used to compute the image probability:

$$P_{\mathcal{SQ}}(I = I_i) = \frac{1}{Z_T} \times \exp(\frac{-D(\mathcal{SQ}, \boldsymbol{I_i})}{T}) \tag{2}$$

where Z_T is the sum of the exponential values over all the images of the database and T the parameter which tunes the size of the search subspace.

At each iteration of the interactive search, the system samples and displays images according to the probability $P_{\mathcal{SQ}}$. All images that the user annotates as relevant are added to the set $\mathcal{SQ}$.

When the parameter T is high, the influence of the dissimilarity to $\mathcal{SQ}$ is small, and thus, the neighborhood explored around the set $\mathcal{SQ}$ is broad. When T decreases, the influence of the dissimilarity to $\mathcal{SQ}$ increases in the probability computation. The search space cuts down around $\mathcal{SQ}$. During first iterations, the database exploration is favored and new examples are added to the query, allowing to catch many modes of the searched category.

The $\mathcal{SQ}$ content information accumulated during first steps may be fully exploited in a second step.

6.2 Semantic query similarity function

The similarity between an image and the set $\mathcal{SQ} = \{R_l, 1 \leq l \leq L\}$ is different from the similarity between two images defined in section 5. For an image I_i (indexed by $\boldsymbol{I}_i$) and for a search based on M feature spaces, the similarity measurement between I_i and the semantic query is calculated as follows:

$$S(\mathcal{SQ}, \boldsymbol{I}_i) = \sum_{k=1}^{M} \beta_k s_k(\mathcal{SQ}, \boldsymbol{I}_i) \tag{3}$$

where $s_k(\mathcal{SQ}, \boldsymbol{I}_i)$ is the similarity in the k^{th} feature space, $\beta_k \in \mathbb{R}^+$. Many similarity functions $s_k(\boldsymbol{R}_l, \boldsymbol{I}_i)$ have been tested and a similarity based on L_1 distance has been adopted in our experiments (with normalization): $s_k(\boldsymbol{R}_l, \boldsymbol{I}_i) = 1 - d_{L_1}(\boldsymbol{R}_l, \boldsymbol{I}_i)$.

[3]The extension of $D(\boldsymbol{R}_l, \boldsymbol{I}_q)$ to $D(\mathcal{SQ}, \boldsymbol{I}_q)$ is presented in the next section.

To merge similarities $s_k(\boldsymbol{R}_l, \boldsymbol{I}_i)$, we use the following *barycenter* operator: $s_k(\mathcal{SQ}, \boldsymbol{I}_i) = \frac{\sum_{l=1}^{L} s_k(\boldsymbol{R}_l, \boldsymbol{I}_i)^2}{\sum_{l=1}^{L} s_k(\boldsymbol{R}_l, \boldsymbol{I}_i)}$ This strategy allows to take the multi-modality into account [8].

Due to the normalization, $s_k(\mathcal{SQ}, \boldsymbol{I}_i) \leq 1 \; \forall k$, and the similarity values $S(\mathcal{SQ}, I_i)$ are then between 0 and 1. The dissimilarity may be expressed as follows:

$$D(\mathcal{SQ}, I_i) = 1 - S(\mathcal{SQ}, I_i)$$

6.3 Parameter tuning

A decreasing law for the parameter T has to be fixed. In CBIR context, the number of iterations must be small in order not to discourage the user. We use an exponential decay[4]:

$$T_j = C^j \times T_0 \tag{4}$$

where T_0 and C are constants ($C < 1$), and j indicates the user interaction steps. Constants have to be fixed for an acceptable number of feedback iterations. Actually, we propose to base them on maximal dissimilarities in the feature space. This approach will allow to handle more intuitively the exploration process.

Let us first specify that dissimilarity D used for parameter tuning was equalized beforehand. We consider the probability δ of selecting an image whose dissimilarity is lower than a threshold d_{bound}. According to equation (2), we have:

$$\delta = \sum_{I_i \in \; db | D(\mathcal{SQ}, \boldsymbol{I}_i) \leq d_{\text{bound}}} P_{\mathcal{SQ}}(I = I_i) = \sum_{I_i \in \; db | D(\mathcal{SQ}, \boldsymbol{I}_i) \leq d_{\text{bound}}} \frac{1}{Z_T} \exp(\frac{-D(\mathcal{SQ}, \boldsymbol{I}_i)}{T})$$

With the notation $d_i = D(\mathcal{SQ}, \boldsymbol{I}_i)$, $\forall \; I_i \; \in \; db$, after ranking, δ may be expressed as follows:

$$\delta = \sum_{0 \leq d_i \leq d_{\text{bound}}} \frac{1}{Z_T} \exp(\frac{-d_i}{T})$$

To find an explicit relation between T, d_{bound} and δ, D may be considered as continuous (reasonable hypothesis since the database contains many images). After equalization of D, this leads to the following approximation:

$$\delta \approx \frac{\int_0^{d_{\text{bound}}} \exp(\frac{-x}{T}) dx}{\int_0^{\infty} \exp(\frac{-x}{T}) dx}$$

[4]In simulated annealing techniques [21], a combination between high initial parameter T (called temperature parameter) and slow cooling strategy is unsuited. For time consuming constraints, exponential decay is often preferred.

i.e.:

$$d_{\text{bound}} = -T \cdot \ln(1-\delta) \tag{5}$$

T_0 is calculated according to the maximum dissimilarity $d_{\text{bound}} = D_{\text{MAX}}$ at the beginning of the retrieval. So, we have (*cf.* eq. 5):

$$T_0 = \frac{-D_{\text{MAX}}}{\ln(1-\delta)}$$

In the same way, d_{max} is defined as the maximal dissimilarity at the last step of the exploration. By choosing the number n of iterations during the exploration, it is possible to fix up the final value of T:

$$T_{\text{final}} = T_n = \frac{-d_{\text{max}}}{\ln(1-\delta)}$$

The constant C may be computed thanks to eq. 4 in the following way:

$$C = \sqrt[n]{\frac{d_{\text{max}}}{D_{\text{MAX}}}}$$

To summarize, four parameters handle the exploration process:

- δ, close to 1, set to $1 - 10^{-5}$ in all tests.
- D_{MAX}. In our experiments, the whole database is selected. In this case, D_{MAX} is the dissimilarity of the image of the database the furthest away from the query.
- d_{max}. This value may be tuned thanks to the number of images in which the system is looking for at the end of the exploration process. We set this number to $20 \times N_{\text{disp}}$ images in our experiments (where N_{disp} is the number of images displayed at each iteration).
- n the number of iterations. From 5 to 10 iterations is a nice tradeoff between short search and effective exploration of the database.

Once the parameters are set, the process may start. One iteration of the stochastic exploration algorithm is as follows:

– Step 1:	*For each image I_i, $P_{\mathcal{SQ}}(I = I_i)$ calculation (eq. 2)*
– Step 2:	*$P_{\mathcal{SQ}}()$ sampling and display of images*
– Step 3:	*Image annotation and updating of semantic query $\mathcal{SQ}$*
– Step 4:	*Decreasing of T (eq. 4)*
– Step 5:	*Go to step 1 until the end of the exploration*

7 Category search: semantic query exploitation

7.1 Context

When the semantic query is rich enough, it makes sense to use it to the full extent to get as many relevant images as possible. All the examples that the user has annotated as irrelevant during the exploration step are also stored to be exploited during this second step.

Relevance feedback may be used to refine the semantic query similarity function introduced in section 6.2. Weight competition on β parameters (cf. section 5) has been applied. We called it the SQRF technique (semantic query relevance feedback).

Recently, statistical learning approaches have been introduced in CBIR context and have been very successful [37, 5]. Discrimination methods may significantly improve the effectiveness of visual information retrieval tasks. However, CBIR is a very specific classification task. There are very few training data during the retrieval process, and the input space dimension is very high. Support Vector Machines (SVM) seem to be a good solution in such a context because they are dedicated for binary classification and are well adapted to these specificities. They usually have good classification performances with few training data and high input space.

However, SVM need a minimum of examples to obtain good discrimination and generalization properties. For this reason, we always start category search session with exploration strategy before SVM classification.

7.2 SVM parameter setting

Let $(\boldsymbol{I_q})_{q\in[0,l-1]}$, $\boldsymbol{I_q} \in \mathbb{R}^p$ be the feature vectors representing annotated images, and $(y_q)_{q\in[0,l-1]}$, $y_q \in \{-1, 1\}$ be their respective annotations (1 = relevant, -1 = irrelevant).

The aim of the SVM classification method is to find the best hyperplane separating relevant and irrelevant vectors maximizing the size of the margin (in between both classes). Initial method assumes that relevant and irrelevant vectors are linearly separable. To overcome this problem, kernels $k(.,.)$ have been introduced. It allows to deal with non-linear spaces. Moreover, a soft margin may be used, in order to tolerate noisy configuration. It consists in a very simple adaptation by introducing a bound C in the initial equations [40]. The resulting optimization problem may be expressed as follows:

$$\begin{array}{l} \text{argmax}_{\boldsymbol{\alpha}} \sum_{q=0}^{l-1} \alpha_q - \frac{1}{2} \sum_{q,j=0}^{l-1} \alpha_q \alpha_j y_q y_j k(\boldsymbol{I_q}, \boldsymbol{I_j}) \\ \text{with} \left\{ \begin{array}{l} \sum_{q=0}^{l-1} \alpha_q y_q = 0 \\ \forall q \in [0, l-1] \quad 0 \leq \alpha_q \leq C \end{array} \right. \end{array} \tag{6}$$

Thanks to the optimal $\boldsymbol{\alpha}^\star$ value, the distance between a vector $\boldsymbol{I}_i$ and the separating hyperplane is used to

evaluate how the image I_i is relevant:

$$f(\boldsymbol{I_i}) = \sum_{q=0}^{l-1} y_q \alpha_q^\star k(\boldsymbol{I_i}, \boldsymbol{I_q}) \tag{7}$$

The kernel function, used in the SVM algorithm, has to be determined. Most popular kernels are Gaussian and polynomial ones. We selected a Gaussian kernel $k(\boldsymbol{I}_i, \boldsymbol{I}_j) = exp(-\frac{d^2(\boldsymbol{I}_i, \boldsymbol{I}_j)}{2\sigma^2})$ because we have no prior assumption on input data configuration. Moreover, distance in Gaussian kernel may be chosen according to the type of feature vectors. For instance, we use a χ^2 distance which is well suited for histograms, and in that case, $\sigma = 1$.

8 Experiments

We display in this section some experiments on $6,000$ images from the *Corel* database introduced in section 2. In order to make quality assessment, reference categories are used to generate many experiments and make statistics on precision and recall criteria [32]. As there are high variations in the number of images in each category, performances are established independently for each category, but are averaged on many queries of the same category.

Category retrieval is evaluated[5]. First, results and quality assessment are done about the exploration step. The exploitation of the semantic query is then reported.

Feature vectors are composed of four index vectors presented in sections 2,3,4, but the user may select or un-select some of them. For category retrieval statistical computation, only color and texture features are considered.

On figure 11, we present three different results:

- top-15 result with no feedback (top),
- top-15 result after 5 iterations of our similarity feedback strategy with a single query image without exploration (middle),
- result with the exploration process to build the semantic query. Images are not ranked, but most of the images of the semantic query $\mathcal{SQ}$ (obtained after 5 iterations) are displayed (bottom).

Let us note that the same number of annotations has been applied for the three experiments.

The user is looking for the castle category in this experimentation and the initial query is the castle picture of the first line, left column of top frame.

[5]Experiments on target search with relevance feedback are not reported here. Interested reader can find in the paper [12] a complete evaluation of our method and a comparison to leader techniques.

One can notice that the result without feedback is really poor; the color distributions and the transitions seem to be very close in the returned images, but there are no castles before the ninth rank. Next, the feedback strategy SQRF is able to find relevant images (middle window result), but the number of relevant retrieved images remains low. Finally, the exploration strategy gives by far the best results (bottom window). Many castle images of very different kinds have been retrieved without using more user annotations than other methods.

In figure 12, a quality assessment is realized over 20 distinct queries of the flower category. The performance criterion is the recall according to the number of iterations. We also computed performances of a random search. In the beginning of our controlled exploration, the system returns many images even far from the query in order to catch the category diversity. The recall criterion is weak during these first iterations, and then quickly increases after four or five feedbacks. When the semantic query has caught enough modes of the category, performances are higher than with simple competition strategy.

Our exploration strategy is effective to build a powerful semantic query, which makes the accumulation of many relevant images easy. After some iterations, more relevant images are retrieved than using traditional search-by-similarity methods.

In figures 13 and 14, the last part of our category retrieval strategy is evaluated and compared for two categories. Two methods have been introduced in section 7, the semantic query relevance feedback technique (SQRF), and the SVM binary classification. They are used after the exploration and compared to the relevance feedback technique without exploration. The efficiency of the exploration method is confirmed by these precision/recall curves: the technique performs better with exploration. The SVM classification always gives the best results. These results have been confirmed by tests on many categories from this database and from other generalist databases [16, 14]. One can notice that performances are better for the cavern category than for the flower category. The cavern category is simpler (50 images) than the flower one (200 images sparsely distributed in the feature space). These properties explain the difference of retrieval effectiveness.

As far as the time consuming is concerned, the main computational needs is the $O(N)$ computation (where N is the number of images in database) of the distance between any image and the query (step 1 in section 6), or for the SVM method, the fellowship to the relevant class (function $f(.)$ in section 7.2) on the whole database. Other requirements are negligible against N. In our experiments, all methods need at most a few seconds to compute new results with a Pentium 3 GHz. About the main memory space (RAM), we need to store feature vectors ($N \times p$ doubles) and kernel cache lines ($N \times c$ doubles) for the SVM computation, where p is feature vector dimension, and c the number of lines to cache. Other requirements are negligible against N. In the experiments, about 3 Mo are used by the feature vectors, and 10 Mo for the kernel cache.

Figure 11: Retrieval strategy comparison: the top-15 result without feedback (top frame), the top-15 result using the simple feedback retrieval strategy (middle), the exploration strategy result (bottom).

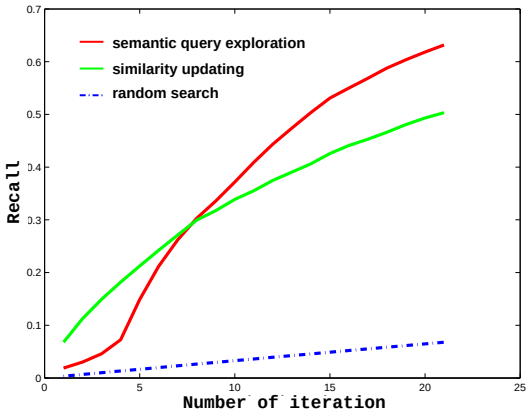


Figure 12: Exploration evaluation: recall according to the number of feedback iterations.

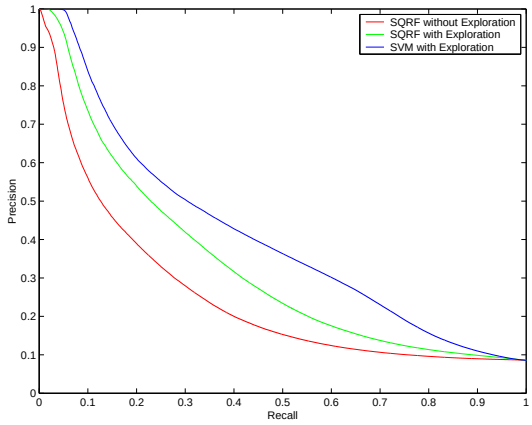


Figure 13: Precision/recall curves for semantic query relevance feedback and SVM methods with or without exploration strategy (flower category).

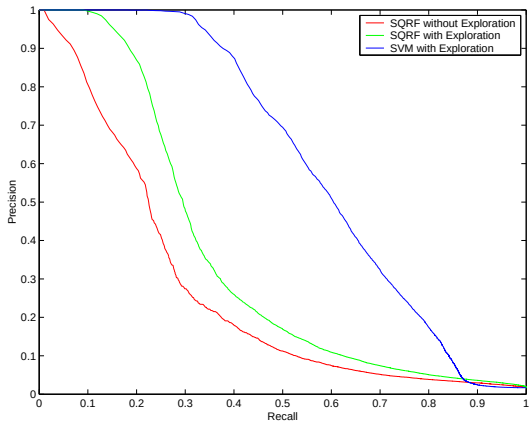


Figure 14: Precision/recall curves for semantic query relevance feedback and SVM methods with or without exploration strategy (cavern category).

9 Conclusion

We proposed a new system to take up the double gauntlet of CBIR: powerful image signature and efficient interactive retrieval strategy.

Color and texture indexing are considered. We carried out comparative tests concerning color and texture space quantization in the framework of CBIR. As a result of a lot of experiments, we have chosen an adaptive quantization method with an efficient parameter tuning. Adaptive quantization is more effective than static one. Thanks to this database quantization, the number of clusters in the quantization can be drastically reduced. Typically, 25 clusters produce satisfactory results for databases of about $10,000$ images. It is used in RETIN as the default value to quantify color and texture spaces. We also encoded spatial information through vertical cooccurrences of colors and textures. This is a simple and effective way to build signatures including the spatial distribution of color and texture features.

We proposed an original method for image category retrieval including an exploration step of the database. As the searched category often has a multi modal distribution in the feature space, we developed an approach to explicitly model this complexity. During the retrieval, query and similarity are modified to take advantages of the user annotations. We introduced a semantic query, which is composed of all the relevant images as the search advanced. To select new images for labeling, our process is based on a controlled exploration strategy of the database. The control parameter is integrated in a global relevance function. Due to this new formulation, an explicit feature space exploration is proposed to the user. Many images, scattered in feature spaces, may be retrieved during this exploration process. We also proposed a SVM binary classification. It allows to retrieve most of the images of the searched category starting from the semantic query obtained at the end of the exploration step. Experiments and quality assessment on a large database have been carried

out to evaluate our approach. The combination of the exploration-based approach with the classification process gives outstanding results when large and complex categories are considered. Experiments have shown that the statistical approach performs better than the geometrical approach for category retrieval.

We are currently working on the integration of our exploration strategy in the statistical framework of active learning [7, 14]. Other investigations concern the analysis of the semantic queries stored during many retrieval sessions. This semantic information is very rich and should be helpful for future category searches [15].

References

[1] S. Berretti, A.Del Bimbo, and E.Vicario. Weighted walkthroughs between extended entities for retrieval by spatial arrangement. *IEEE Transactions on Multimedia*, 5(2):52–70, 2003.

[2] R. Brunelli and O. Mich. Histograms analysis for image retrieval. *Pattern Recognition*, 34:1625–1637, 2001.

[3] G. Caenen, G. Frederix, A.A.M Kuijk, E.J. Pauwels, and B.A.M. Schouten. Show me what you mean! PARISS: A CBIR-interface that learns by example. In *International Conference on Visual Information Systems (Visual'2000)*, volume 1929, pages 257–258, 2000.

[4] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third Int. Conf. on Visual Information Systems*, June 1999.

[5] E. Chang, B. T. Li, G. Wu, and K.S. Goh. Statistical learning for effective visual information retrieval. In *IEEE International Conference on Image Processing*, Barcelona, September 2003.

[6] L. Cinque, S. Levialdi, and A. Pellican. Color-based image retrieval using spatial-chromatic histograms. In *IEEE International Conference on Multimedia Computing and Systems*, 1999.

[7] D. Cohn. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[8] M. Cord, J. Fournier, and S. Philipp-Foliguet. Exploration and search-by-similarity in cbir. In *IEEE SIBGRAPI'03*, Sao Carlos, Brazil, October 12 - 15 2003.

[9] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, and P.N. Yianilos. The bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, 2000.

[10] J.P. Eakins. Towards intelligent image retrieval. *Pattern Recognition*, 35:3–14, 2002.

[11] J. Fournier, M. Cord, and S. Philipp-Foliguet. Back-propagation algorithm for relevance feedback in image retrieval. In *IEEE International Conference in Image Processing (ICIP'01)*, volume 1, pages 686–689, Thessaloniki, Greece, October 2001.

[12] J. Fournier, M. Cord, and S. Philipp-Foliguet. Retin: A content-based image indexing and retrieval system. *Pattern Analysis and Applications Journal, Special issue on image indexation*, 4(2/3):153–173, 2001.

[13] D. Geman and R. Moquet. A stochastic feedback model for image retrieval. In *RFIA'2000*, volume III, pages 173–180, Paris, France, February 2000.

[14] P.H. Gosselin and M. Cord. Retin al: An active learning strategy for image category retrieval. In *IEEE International Conference on Image Processing (ICIP)*, Singapore, October 2004.

[15] P.H. Gosselin and M. Cord. Semantic kernel updating for content-based image retrieval. In *IEEE International Workshop on Multimedia Content-based Analysis and Retrieval (MCBAR)*, Miami, Florida, USA, December 2004.

[16] P.H. Gosselin, M. Najjar, M. Cord, and C. Ambroise. Discriminative classification vs modeling methods in cbir. In *IEEE Advanced Concepts for Intelligent Vision Systems (ACIVS)*, Brussel, Belgium, September 2004.

[17] J. Hafner, H.S. Sawhney, and W. Equitz. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[18] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997.

[19] D. P. Huijmans and N. Sebe. Extended performance graphs for cluster retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pages 26–31, Hawaii, Dec 2001.

[20] Y. Ishikawa, R. Subramanya, and C. Faloutsos. MindReader: Query databases through multiple examples. In *24th VLDB Conference*, pages 218–227, New York, 1998.

[21] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[22] W. Ma and B. Manjunath. Image indexing using a texture dictionnary. In *SPIE Conference on Image Storage and Archiving System*, volume 2606, pages 288–298, Philadelphia, Pennsylvania, October 1995.

[23] B. Manjunath and W. Ma. Browsing large satellite and aerial photographs. In *IEEE International Conference on Image Processing*, volume 2, pages 765–768, Lausanne, Switzerland, September 1996.

[24] W. Niblack, R. Barber, W. Equitz, M. Flickner, E.H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC project: Querying images by content, using color, texture, and shape. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 173–187, February 1993.

[25] J.J. Rocchio. Relevance feedback in information retrieval. In *G. Salton editor, The SMART retrieval system : Experiments in Automatic Document Processing*. Prentice Hall Inc., 1971.

[26] Y. Rubner, C. Tomasi, and LJ. Guibas. The earth mover's distance as a metric for image retrieval. Technical Report STAN-CS-TN-98-86, Department of Computer Science, Stanford University, September 1998.

[27] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *IEEE International Conference on Computer Vision*, pages 59–66, Bombay, India, January 1998.

[28] Y. Rui, T. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture for content-based multimedia information retrieval systems. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 92–89, 1997.

[29] Y. Rui and T.S. Huang. Optimizing learning in image retrieval. In *Conf on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 236–243, Hilton Head, SC, June 2000.

[30] R. Schettini, G. Ciocca, and S. Zuffi. Color in databases: Indexation and similarity. In *First International Conference on Colour in Graphics and Image Processing CGIP'2000*, Saint-Etienne, France, October 2000.

[31] A.W.M Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.

[32] J.R. Smith. Image retrieval evaluation. In *IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'98)*, pages 112–113, June 1998.

[33] J.R. Smith and S.F. Chang. VisualSEEK: a fully automated content-based image query system. In *ACM Multimedia Conference*, pages 87–98, Boston, USA, November 1996.

[34] J.R. Smith and C.S. Li. Image classification and querying using composite region templates. *Computer Vision and Image Understanding*, 75(1-2):165–174, July-August 1999.

[35] M. Stricker and M. Swain. The capacity of color histogram indexing. In *Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 704–708, 1994.

[36] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[37] Simon Tong. *Active Learning: Theory and Applications*. PhD thesis, Stanford University, 2001.

[38] L.V. Tran and R. Lenz. PCA-based representation of color distributions for color-based image retrieval. In *International Conference in Image Processing (ICIP'01)*, volume 2, pages 697–700, Thessaloniki, Greece, October 2001.

[39] N. Vasconcelos and M. Kunt. Content-based retrieval from image databases: current solutions and future directions. In *International Conference in Image Processing (ICIP'01)*, volume 3, pages 6–9, Thessaloniki, Greece, October 2001.

[40] K. Veropoulos. Controlling the sensivity of support vector machines. In *International Joint Conference on Artificial Intelligence (IJCAI99)*, Stockholm, Sweden, 1999.