Region-Based Image Retrieval: Fast Coarse Segmentation and Fine Color Description

Julien Fauqueur and Nozha Boujemaa

Projet IMEDIA - INRIA, BP 105 - 78153 Le Chesnay Cedex - FRANCE

Abstract

The two major problems raised by a region-based image retrieval system are the automatic detection and visual description of regions. We adopt a *coarse detection and fine description* approach. In this paper we first present a new method of unsupervised coarse detection which provides intuitive and visually characteristic regions of interest. This segmentation scheme is based on the classification of Local Distributions of Quantized Colors (LDQC). The Competitive Agglomeration classification algorithm is used which has the advantage to automatically determine the number of classes.

Then, considering that description must be finer for regions than for images, we propose a new region descriptor of fine color variability: the Adaptive Distribution of Color Shades (ADCS). Combined with an appropriate similarity measure, the high color resolution of ADCS improves the perceptual similarity of retrieved regions compared to existing color descriptors.

Key words: image retrieval, region description, segmentation, color quantization, fine color representation, color distribution distance

Email address: {Julien.Fauqueur,Nozha.Boujemaa}@inria.fr (Julien Fauqueur and Nozha Boujemaa).

URL: http://www-rocq.inria.fr/imedia/ (Julien Fauqueur and Nozha Boujemaa).

1 Introduction

The initial content-based image retrieval paradigm, the query by image example, was first proposed in [1] and further developed in systems such as PhotoBook [2], QBic [3], Virage [4], ImageRover [5], PicToSeek [6], Ikona [7]. This paradigm allows to retrieve images in the database whose global visual appearance is similar to a given example image selected by the user. While this paradigm was useful to show the visual information retrieval viability, it is not sufficient to meet the user's need. Indeed the underlying assumption of this paradigm is that the *entire* visual content of an image is relevant for a search. Thus the user is not able to focus on a specific image part and to ignore the background. As a consequence global query by example image only allows an *approximate search* especially in a database of composite images.

We want to allow the user to specify an image part of interest and retrieve visually similar parts in other images of the database regardless the background. This query paradigm takes into account user's search preference more precisely. We must define what "parts of images" should be and how to detect them. Automatic "object" detection in images is a very hard task by sole use of visual features especially since instances of a same object can greatly differ in terms of visual appearance. Various approaches were proposed to provide a partial image representation in the context of content-based image retrieval :

- *feature backprojection:* similar parts in candidate images are identified online: flexible but time consuming at query time (e.g. proposed in [1] and used in VisualSeek [8]).
- *points of interest*: they characterize high frequency sites in images and allow a precise search on parts with salient details but at high computational expense at query time (e.g. [9]).
- *systematic image subdivision into blocks*: simple but inaccurate (see [10][11]).
- *manual subdivision*: the closest to the user's expectation but not viable for large databases (see [12]).
- *unsupervised region-segmentation*: regions are automatically detected (see [13][14]).

Among partial representations, we adopt the region segmentation approach which is a good trade-off since it is unsupervised, provides a natural approximation of objects and allows fast retrieval. To meet the requirements of a region query system, we will propose a fast segmentation technique to detect coarse and relevant regions for the user.

2 Related work

Designing a region-based query system remains a challenging and open problem : automatic detection of regions of interest is a hard task and region description must take into account the visual specificity of regions. Existing region-based query systems differ on those two points. Among these few systems we can cite Blobworld [13] and Netra [14]. In Blobworld, region segmentation is performed by classification of joint color and texture vectors with the Expectation/Maximization (EM) technique. Segmentation is approximate and many small areas are omitted. In Netra [14], segmentation is contour-based and provides satisfactory regions but is very time consuming. A more recent technique involving region-matching for image retrieval is proposed in the SIMPLIcity system [15]. The similarity between two images is measured as a combination of similarities between the regions which compose both images. But the system actually performs *global* image retrieval since all visual features in images are involved. The quality of region segmentation is not their main concern.

Concerning region visual description, most existing region query systems derive traditional global color descriptors. They consist of color distributions computed over a predefined subsampling of color space which yields about 200 colors. Choosing only 200 colors (the same for all images) among the millions of a full color space dramatically reduces the color resolution hence the retrieval precision as we will see in the experiments. However, compared to images, regions are more numerous and more homogeneous so a region-based retrieval scheme requires a higher power of visual discrimination.

Our approach differs by how we detect and describe regions. Regions should integrate more intrinsic variability to be visually more characteristic. We require that regions correspond to regions of interest for the user (potential query regions) and that they are visually characteristic for efficient retrieval. The image segmentation scheme proposed detects *coarse regions*. It is based on the classification of local color distributions evaluated over large neighborhoods with a low classification granularity. Concerning description, rather than describing regions with a predefined set of 200 colors, we propose to define an adaptive set of colors determined at a high resolution which are relevant for each region. The ADCS region descriptor will be the distribution of these colors in the region.
The key idea of *coarse region detection and fine description* is the following: the relatively high visual variability inside regions is accurately described by a high color resolution, such that regions are really specific against each other in the database. Coherence is preserved between region detection and description phases since they are formed by similar local color distributions, and retrieved using distributions of color shades. Part of this work was published in [16].

In the next section, we will explain the Competitive Agglomeration (CA) classification algorithm, an essential background technique in our work since used both at segmentation and description phases. Region extraction by classification of local color distributions will be developed in section 4. The color variability descriptor ADCS will be detailed in section 5 along with the retrieval scheme. We also present the user interface for region-based query in IKONA platform. Then experiments and results will be presented and discussed in sections 7 and 8. Retrieval performance of the ADCS descriptor will be tested against the traditional color histogram and against its combination with some simple geometrical descriptors. We conclude in section 9.

3 Visual feature grouping

For both region detection and description our approach requires an efficient scheme to group visual features of different nature in an unsupervised way. We use the Competitive Agglomeration classification algorithm, called *CA*, presented in [17]. CA has the major advantage to determine automatically the number of classes unlike other classification algorithms used in related work, such as Expectation/Maximization or K-Means. Using notations from [17], $\{x_j, \forall j \in \{1, ..., N\}\}$ denotes the set of N data we want to cluster and C the number of clusters. $\{\beta_i, \forall i \in \{1, ..., C\}\}$ denotes the prototypes to be determined. The distance between data x_j and prototype β_i is $d(x_j, \beta_i)$. The CA-classification is performed by minimizing following objective function J:

$$J = J_1 + \alpha J_2, \tag{1}$$

where :

$$J_1 = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^2 d^2(x_j, \beta_i) \text{ and } J_2 = -\sum_{i=1}^{C} [\sum_{j=1}^{N} u_{ij}]^2$$

Subject to membership constraint:

$$\sum_{i=1}^{C} u_{ij} = 1, \forall j \in \{1, ..., N\} \tag{2}$$

where u_{ij} represents the fuzzy membership degree of feature x_j to class of prototype β_i. Minimizing J_1 alone is equivalent to perform a Fuzzy C Means classification [18] which determines C optimal prototypes and the fuzzy partition U given x_j and C using distance d. A key point in CA algorithm is the introduction in objective function J of the term J_2 which can be considered as a clustering validity criterion (see [19]), which is minimum when the number of classes is minimum. Therefore J is written as a combination of two opposite effect terms J_1 and J_2. α is the competition weight between terms J_1 and J_2

in equation (1). At iteration k, weight α is expressed as :

$$\alpha(k) = \eta_0 \exp(\frac{-k}{\tau}) \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^2 d^2(x_j, \beta_i)}{\sum_{i=1}^{C} [\sum_{j=1}^{N} u_{ij}]^2} \tag{3}$$

As iterations go, α decreases so emphasis is first given to agglomeration process, then to classification optimization. α is fully determined by parameters η_0 and τ.
The algorithm is initialized with an overestimation of the number of clusters. During iterative minimization of J spurious clusters are discarded. As a consequence, the minimization of J estimates the partition and the prototypes and simultaneously determines automatically the number of classes. Spurious clusters are those whose population, defined by quantity $\sum_{j=1}^{N} u_{ij}$ for a cluster i, falls below a given threshold ϵ. Convergence is decided when prototypes are stable. The classification granularity is controlled by factors ϵ and α, through its magnitude η_0 and its decline strength with τ. The higher η_0 and τ, the higher α, so the more classes are merged. The higher ϵ and the more classes are discarded. For a given classification granularity, CA determines the optimal number of classes.

The choice of the distance measure controls the shape of detected clusters. While euclidean distance allows to detect hyperspherical clusters, the Mahalanobis distance [20] detects hyperellipsoidal clusters which are more generic. This distance takes into account cluster variance and is defined as follows :

$$d^2(x_j, \beta_i) = \mid \Sigma_i \mid^{1/n} (x_j - c_i)^T \Sigma_i^{-1} (x_j - c_i)$$

where c_i is the centroid of class of prototype β_i and Σ_i its fuzzy covariance matrix. Covariance matrix is updated as follows :

$$\Sigma_i = \frac{\sum_{j=1}^{N} u_{ij}^2 (x_j - c_i)(x_j - c_i)^T}{\sum_{j=1}^{N} u_{ij}^2}$$

CA will be used at three steps in our work with different levels of granularity:

- image color quantization (classification of color triples)
- LDQC grouping (classification of color distributions)
- region description with ADCS (classification of color triples)

For more details on competitive agglomeration the reader is referred to the original paper [17] and to [21] for details on its practical application.

4 Region detection by coarse segmentation

Composite natural images, such as photostock images, can encompass a broad variety of visual details. In the context of region-based image retrieval, we focus on salient image regions. We propose a coarse segmentation method to detect regions which are homogeneous in terms of photometry but encompass a certain visual variability. Fine visual details are naturally integrated within regions through the coarse detection. Region photometric variability is decided to make regions more visually characteristic from one another in the database. In addition we require our detection scheme to be unsupervised, fast and naturally provide intuitive regions for the user.
Our segmentation approach relies on the CA classification of LDQC features (*Local Distribution of Quantized Colors*). This single feature carries in itself rich photometric information of local color variability. It allows to detect uniform areas as well as textured ones, without having to combine features of different nature such as mean color and texture. Besides it is coherent with the ADCS region descriptor presented later. The coarseness of region detection is obtained by the relatively large pixel neighborhoods to determine LDQC's and the coarse CA classification granularity. Grouping similar LDQC's to generate regions leads to coarse coherent regions more naturally and requires little spatial postprocessing. In this section we will detail the segmentation algorithm in the order of its different steps :

- LDQC feature extraction
- LDQC feature grouping
- spatial consolidation

This segmentation scheme was tested on a database of 11.479 images photostock images. Results will be presented on these images.

4.1 LDQC feature extraction

The color set used to determine the local color distributions of an image should not be the entire color space but a compact and representative set of the image. A natural image can have as many as 60.000 different colors. For each image, we define this adaptive set by color quantization to dramatically reduce the number of colors without losing too much perceptual information. Then for each pixel neighborhood, local distribution is determined on the set of quantized colors providing a LDQC feature (*Local Distribution of Quantized Colors*). Then grouping of LDQC's obtained from the entire image will generate coarse regions.

The image color quantization step aims at reducing the number of color bins in the LDQC without losing too much perceptual information. They should

allow to separate various salient regions in an image. Various quantization schemes were proposed which differ in computational load and in precision (see [22]). They rely on a more or less adaptive partition (hence more or less expensive and precise) of the color space into cells. Many approaches assume the number of quantized colors is given, which we do not want. We rather want this number to depend on the image photometric complexity.

The *General Lloyd Algorithm* [23] (also referred to as *Linde-Buzo-Gray* and which is equivalent to the well-known *K-Means* classification method [24]) is a widely used color quantization scheme in the literature. It consists in estimating iteratively the optimal partition of color pixels into a fixed number of classes with the quadratic error criterion. Compared to GLA, CA algorithm (see section 3) presents the major advantage to automatically determine the number of classes. Further advantages of CA are investigated in [21]. Quantized colors are obtained as the class prototypes resulting from the CA classification of image color pixels.

Concerning color space, since classification tightly relies on the metric, a perceptually uniform color space is necessary, the most common being Luv and Lab [22] [25]. Theses spaces were designed such that color differences judged equal by a human are also equal in euclidean distance in these spaces. On contrary rgb and Hsv color spaces are not perceptually uniform [26]. Hsv space is intuitive but suffers from discontinuities (*hue* component is cyclic and *hue* and *value* components are meaningless for low *saturation*). rgb space has the advantage to avoid the transformation computation, but its topology is not representative of color similarity perceived by a human observer. The choice of color quantization by pixel classification in the Luv space [25] with the euclidean distance emerges as a natural choice. Luv was preferred to Lab due to a lower transformation cost from native rgb space. While euclidean distance will only detect hyperspherical colors clusters, we rather use the Mahalanobis distance (see section 3) to detect hyperellipsoidal clusters which provide a good model for color gradations and shades.

Initial prototypes for CA are defined from a subset of original colors. They are those which lie at the intersection of a 7×7 grid, i.e. on 36 regularly spaced sites in the image to facilitate CA convergence. So the initial color partition contains 36 quantized colors. The CA classification granularity (see section 3) was empirically chosen such that large areas with a strong texture are represented by more than one quantized color. At classification convergence, class prototypes define the set of n quantized colors. As CA determines automatically the number of classes, the number of quantized colors represents the image color variability.

After color quantization the image photometric complexity is reduced. Note that this level does not allow to detect regions directly, because coarse re-

Fig. 1. Original image has 43.217 unique colors (left) and quantized image has 27 quantized colors (right)

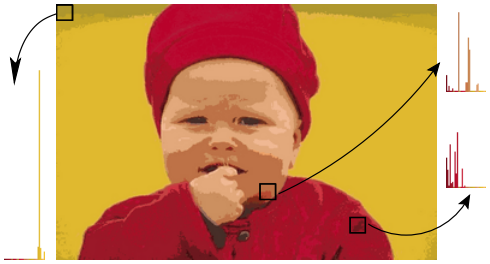


Fig. 2. Illustration of LDQC's over three different neighborhood windows.

gion homogeneity should be defined in terms of color variability rather than pointwise color information. In order to capture local visual characteristics, be they uniform or textured, LDQC feature are locally extracted in the quantized image. We slide a window over the quantized image and in each pixel neighborhood we determine the corresponding local color distribution of quantized colors (LDQC's). So in each neighborhood a LDQC feature is extracted. Figure 4.1 illustrates 3 examples of LDQC in three image sites with different color variability. In uniform neighborhoods LDQC distributions have a dominant peak (or "mode") while in textured ones they tend to be flatter. For a 500x400 image, window width is 31 pixels and evaluation step is half a window size (i.e. 16 pixels). The number of extracted LDQCs for a window is determined as follows : $(500/16) \times (400/16)=31 \times 25=775$.

4.2 LDQC feature grouping

To group extracted LDQCs using CA classification algorithm, a suitable distance is required which influences segmentation quality. Traditional distances to compare color distributions, such as Minkowski L^p distances, rely on a simple bin-wise comparison without taking into account the color information associated to each bin. They implicitly assume that bins are independent from one another which is untrue since a color similarity can be determined between colors associated to two bins. Such distances have been applied to color distributions for global image search. Although they provide satisfactory results for this problem, they turn out to be too imprecise for our purpose.

An example of their limitation is that they consider at a maximum distance two distributions which do not intersect. This does not correspond to human perception. Consider, for example, the case of two pixels neighborhoods which have different but similar colors such as shades of blue in a sky. They will be at maximum L^p distance although perceptually similar.

The color quadratic distance [27] proposed in the context of the QBic system provides a nice solution to this problem by integrating the color bin distance within the color distribution distance. We define X and Y two color distributions over the n quantized colors and write them as pairs of color/population : $X = \{(c_1, p_1^X), ..., (c_n, p_n^X)\}$ and $Y = \{(c_1, p_1^Y), ..., (c_n, p_n^Y)\}$.
The quadratic distance between X and Y is :

$$\begin{aligned} d_q(X,Y)^2 =& (X-Y)^T A (X-Y) \\ =& \sum_{i=1}^{n}\sum_{j=1}^{n}(p_i^X - p_i^Y)(p_j^X - p_j^Y)a_{ij} \end{aligned} \tag{4}$$

where $A = [a_{ij}]$ is the matrix of color similarities a_{ij} between colors c_i and c_j : $a_{ij} = 1 - d_{ij}/d_{max}$ where d_{ij} is the euclidean distance in the Luv color space and d_{max} the maximum of this distance in the color space. Note that if A denotes the identity matrix, the distance is the euclidean distance itself, i.e. $d_q(X,Y) = ||X - Y||_{L^2}$.

Figures 4 and 5 show the improvement of quadratic distance compared to L^1 distance using 3 images of "sky", "brick" and "wicker". Due to their photometric homogeneity they can be considered as pixel neighborhoods or regions. Each image was transformed according to 6 different intensity factors. So we obtain three image families (18 images total, see figure 3) which differ in photometric homogeneity and intensity. Color distributions of the 3 brightest images (corresponding to intensity factor 2) and that of the 3 darkest (factor 0.5) have been compared to the 17 other images with L^1 distance (fig. 4) and quadratic distance (fig. 5).
We observe a saturation effect of L^1 distance : most of retrieved images (fig. 4) are gathered near the maximum distance, around grade 200. As a consequence L^1 performs poorly for discriminating homogeneous images. On the other hand, in figure 5, we observe quadratic distance ability to measure the perceptual continuum between various intensities of a given image and, at the same time, to separate classes ("sky", "brick" and "wicker"), although distributions from homogeneous regions may have empty intersections. It is important to also note that the quadratic distance property of continuity with respect to intensity shift can also be shown for color shift. This better power discrimination of visual of quadratic distance compared to L^1 is necessary for LDQC grouping and will also be useful in combination with ADCS descriptor which we will present later.
Concerning classification, CA algorithm is used (see section 3) to group ex-

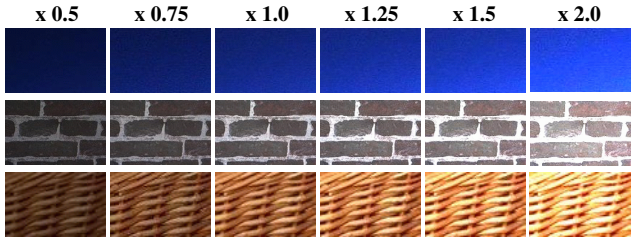


Fig. 3. 3 images extracted from actual regions are transformed according to 6 intensity factors. The 18 test images differ in intensity, color and texture.

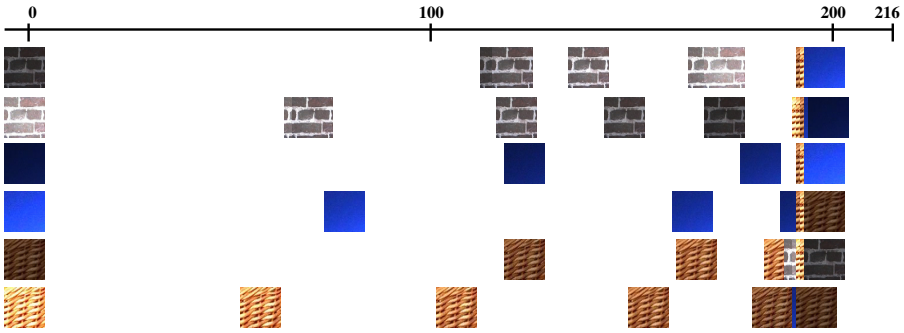


Fig. 4. Six similarity tests with L^1. On each line, images are positioned according to the L^1 distance in color distribution with the first image. Graduation depicts distance values. Very similar images are correctly ranked first, but all other images are clustered around maximum distance (around value 200) and cannot be distinguished. On average 13 images out of 18 are at a distance value in the collection $\{197, 198, 199, 200, 201\}$. This illustrates the lack of precision of L^1 distance for homogeneous data. More generally, bin-wise distances behave poorly (see text) when comparing distributions which have little or empty intersection (case of homogeneous data).

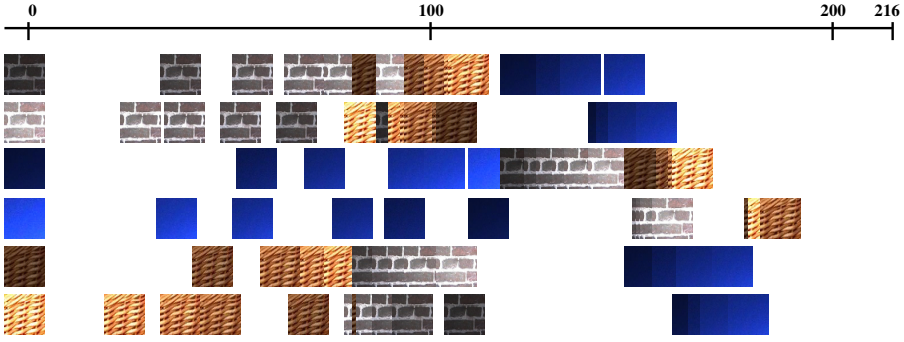


Fig. 5. Six similarity tests with d_{quad}. Same illustration as in figure 4 but with quadratic distance. Distance values are more spread out than with L^1 and provide remarkably better results. Our perception of visual *continuum* between images is more accurately measured with quadratic distance and it provides a better power of visual discrimination between classes of images.

tracted LDQCs from image using the quadratic distance. Initial prototypes for CA are defined from a subset of LDQCs. They are those which lie at the intersection of 6×6 grid, i.e. on 25 regularly spaced sites in the image. At classification convergence, the final partition provides LDQC classes and their prototypes. Segmented image is obtained by associating to each pixel the tag of the class to which its neighborhood belongs. A vote filter is then used to discard isolated tags in the image.

4.3 Spatial consolidation

By associating LDQC class tags to pixel neighborhoods, we obtain a complete image partition into adjacent regions. Some regions may still be too small to form regions of interest; so they needlessly increase the total number of regions in the database. Besides in complex scenes these small regions are often located at the frontier between two salient regions or within a salient region. They should be merged to improve regions of interest topology.

We require that each region covers at least 1.5% of total image area. Below this threshold a region is merged to its neighbor region which is the most visually similar, if it is similar enough. Two regions are said "visually similar" if their mean LDQC are close in quadratic distance. Iterative scheme for small region merging is the following : consider the smallest region of area below 1.5% which has a visually similar neighbor region, if such a small region exists, and merge it into the neighbor region. When there is no more such small regions, remaining regions below 1.5% which have no visually similar neighbor regions are considered as noise. They are discarded and not indexed.
This region merging scheme is achieved using a Region Adjacency Graph [28], or *RAG*. Region attributes are stored in graph nodes (area, color distribution, contours, position) and adjacency information in graph edges (adjacency, common contour length) as illustrated in figure 4.3.

5 Region indexing and retrieval

After segmenting images in the database, we must provide an efficient scheme to characterize region visual appearance and to compare them in a perceptually efficient manner for the user.

In this section we will propose the ADCS region descriptor which, in combination with the segmentation scheme, is part of our region-based retrieval approach of *coarse region detection and fine description*. A similarity measure will be introduced which allows to compare any kind of color distribution,

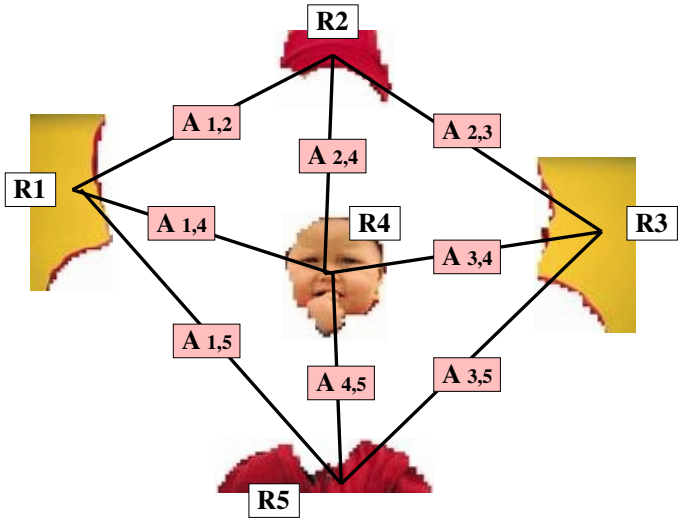


Fig. 6. RAG structure of partitioned image. RAG stores region R_i attributes in nodes and adjacency information between all region pairs (i, j) graph edges. These information is exploited for region merge and removal in the segmentation scheme.

Fig. 7. Final segmented image : regions are represented with their mean color.

including ADCS. Then retrieval results will be presented and discussed on a generic photostock image database.

5.1 ADCS, a fine and compact descriptor of region color variability

Most of existing region-based retrieval systems describe regions using the traditional global color histogram for regions which was initially proposed for query by image example. It consists in color distribution determined on a fixed color space quantization in about 200 colors : 166 colors in VisualSeek [8] and 218 in Blobworld [13] (actually reduced to 5 by singular value decomposition). In Netra [29] an average of 3.5 *dominant colors* per region is selected from a colormap of 256 colors fixed for a given database. The common point between existing color descriptors for both regions and images is the use of a fixed, hence *coarse*, color set to represent the entire color space (be it RGB, LAB,

Luv or Hsv). Choosing the same 200 colors among millions provides a coarse representation of colors. While this may be sufficient to compare global image appearance, it is not suitable for region-based retrieval. Figure 8 illustrates an example of the limitations of traditional color histogram to characterize a region.

Compared to images, regions are more homogeneous and more numerous. This statement implies that region-based retrieval requires to discriminate in the database more color distributions which are more "peaked". As a consequence, in addition to being compact, a region descriptor must provide a *fine* representation of colors.

To meet these requirements we propose to describe regions with the distribution determined on an adaptive and fine set of colors which are relevant for each region which we call *Colors Shades*. Resulting descriptor is *ADCS* for *Adaptive Distribution of Color Shades*. Color shades are obtained by a fine color quantization performed on each region using CA algorithm. We have presented this quantization scheme in section 4.1 and mentioned its advantages over LBG/GLA algorithm. Compared to fixed color space subdivision as in traditional color histograms, it naturally provides more accurate quantized colors. In each region, pixels are classified using CA with a fine granularity (see section 3) and Mahalanobis distance in Luv color space. Quantization is adaptive to each region so it provides colors which are more representative than if an *image* color quantization were performed [1]. An ADCS index is composed of the list of color shade triples and their corresponding normalized population in the region.
We use the term *color shades* rather than *dominant colors* (as in [29] and [30]) to express the presence of minor colors in ADCS descriptor. The higher the region photometric complexity and the more color shades in the ADCS descriptor. The nature and the number of color shades are specific to each region unlike with existing color descriptors. They are picked from the *entire* Luv space which contains 5.6 million colors while other descriptors can not distinguish more than about 200 colors. As a conclusion, ADCS provides a fine, compact and adaptive representation of region color variability. Figure 8 illustrates how the gain in color precision with ADCS compared to traditional histogram results in a better visual discrimination.

[1] Indeed, with an *image* color quantization, participation of region pixels is porportional to region area which implies that resulting quantized colors would be more accurate for larger regions. To avoid this undesirable effect, we rather adopt a region color quantization to describe regions.

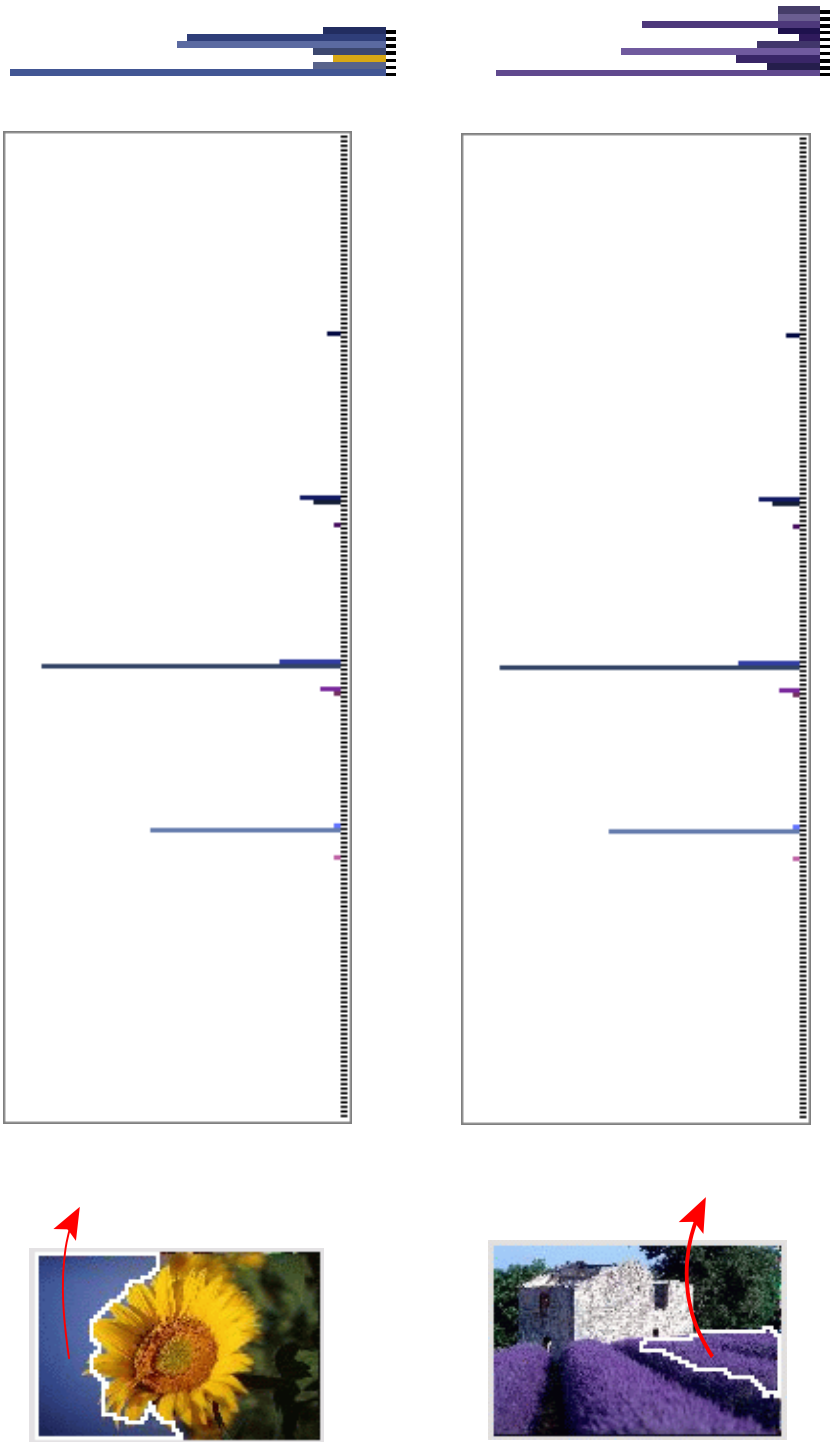


Fig. 8. Limitation of traditional color histogram : two perceptually different regions may have almost same histograms (middle images). Its coarse color resolution brings colors which are perceptually different in the same bins, while ADCS color shades provide a more accurate description (top).

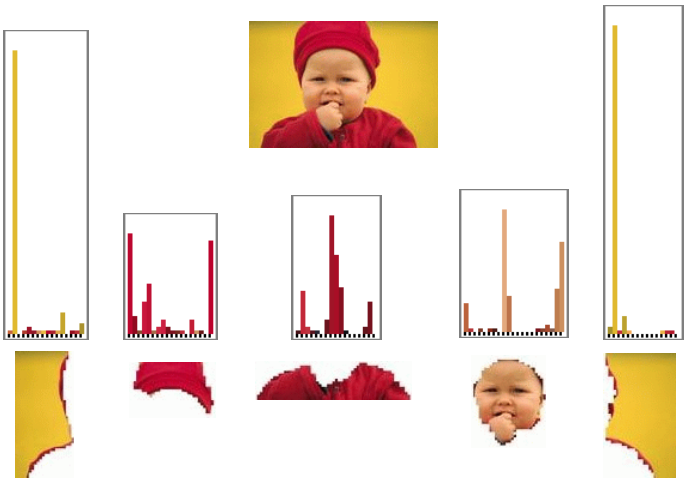


Fig. 9. Original image, its detected regions and their respective ADCS distributions. Color bin order does not matter. We remark that textured regions corresponding to the hat and the coat are represented by different shades of red and the uniform yellow background by a dominant yellow peak and a few other minor colors. The set of color shades obtained is perceptually relevant for each region.

5.2 Generalized form of color quadratic distance

We now address the problem of similarity measure between two ADCS distributions to retrieve regions. Since ADCS relies on a region-dependent color quantization, two ADCS index have different color sets and usual bin-wise histogram distances are not applicable. Color quadratic distance is again a relevant choice for three reasons. The first reason is practical : we will see that it can be applied to distributions with different quantizations. The second is its power of visual discrimination shown in section 4.2 relatively to intensity and color shifts. The third one is it provides a consistent similarity measure even between two distributions with empty intersection unlike usual bin-wise distances (such as L^p). Indeed since ADCS color resolution is fine, two ADCS distributions are very likely to have an empty intersection, i.e. no color shades in common.

To our knowledge only two distances were proposed to compare distributions expressed on different quantizations : *Earth Mover Distance* [31] and *Weighted Correlation* [32]. Both papers deal with global image retrieval and quantization is performed at image level. The first distance requires solving iteratively a linear optimization problem; so it is complex and computationally expensive. The second is faster (in $\mathcal{O}(NN')$ where N and N' are the number of colors in each distribution) but is defined specifically for their quantization algorithm, which is a variation of K-Means.

To compare two ADCS distributions we propose to express the quadratic

distance in its *generalized form*. We will show that quadratic distance allows to compare color distributions based on *different* quantizations, unlike what other papers claimed [31][33][32][34]. In section 4.2, quadratic distance was used in its original form to compare LDQC distributions which have same quantizations. We propose to rewrite its expression from formula (4) in order to remove terms which involve bin difference. For a given query region of ADCS X similar regions in the database are such that their ADCS Y minimizes $d_{quad}(X,Y)$. Let us write X and Y as pairs of color/population : $(c_1^X, x_1), ..., (c_{n_X}^X, x_{n_X})$ and $(c_1^Y, y_1), ..., (c_{n_Y}^Y, y_{n_Y})$. We define X' and Y' as the extensions of distributions X and Y over the *entire* color space (LUV here) as follows : X' has the same values $\{x_1, ..., x_{n_X}\}$ as X on the set $\{c_1^X, ..., c_{n_X}^X\}$, and 0 on the rest of the space. Y' is defined likewise from Y. So we have $d_{quad}(X', Y') = d_{quad}(X, Y)$. Since X' and Y' are defined over the same color space (the entire space), $d_{quad}(X', Y')$ can be expressed. We note A the matrix of similarity between all colors in the entire space. We get :

$$\begin{aligned} d_{quad}(X,Y)^2 =& d_{quad}(X',Y')^2 \\ =& (X'-Y')^T A (X'-Y') \\ =& X'^T A (X'-Y') - Y'^T A (X'-Y') \\ =& X'^T A X' - X'^T A Y' - Y'^T A X' + Y'^T A Y' \end{aligned}$$

Symmetry of matrix A implies :

$$d_{quad}(X',Y')^2 = X'^T A X' + Y'^T A Y' - 2X'^T A Y'$$

By construction of X' and Y', we have :

$$X'^T A X' = X^T A^X X \ , \ Y'^T A Y' = Y^T A^Y Y \ , \ X'^T A Y' = X^T A^{XY} Y$$

where matrices A^X, A^Y and A^{XY} are the restrictions of matrix A which express color similarities between, respectively, X's color shades with themselves (matrix of dimension $n_X.n_X$), those of Y with themselves (dimension $n_Y.n_Y$) and those of X with those of Y (dimension $n_X.n_Y$). We obtain the following formula for $d_{quad}(X,Y)^2$ in which no more bin-wise difference appears :

$$d_{quad}(X,Y)^2 = X^T A^X X + Y^T A^Y Y - 2X^T A^{XY} Y$$

and in scalar form we have :

$$d_{quad}(X,Y)^2 = \sum_{i,j=1}^{n_X} x_i x_j a_{c_i^X c_j^X} + \sum_{i,j=1}^{n_Y} y_i y_j a_{c_i^Y c_j^Y} - 2\sum_{i=1}^{n_X}\sum_{j=1}^{n_Y} x_i y_j a_{c_i^X c_j^Y} \qquad (5)$$

Expression (5) is the generalized form of color quadratic distance between distributions X and Y determined on any color sets $\{c_1^X, ..., c_{n_X}^X\}$ and $\{c_1^Y, ..., c_{n_Y}^Y\}$ with respective color populations $\{x_1, ..., x_{n_X}\}$ and $\{y_1, ..., y_{n_Y}\}$. This expression is used to compare query ADCS with candidate ADCS in the database. It

can be used more generally with any kind of color distributions with *different* color quantizations. This computation is actually optimized at query time by pre-computing the first two terms in (5) for each region at indexing time such that only the last cross term needs to be evaluated.
Note that in Netra [14] and in MPEG7 [30], the distance used to compare dominant colors distributions is an approximation of the generalized form of color quadratic distance in which colors in each distributions are at maximum distance in color space. We do not make such an assumption since color shades of an ADCS index can have various degrees of similarity.

5.3 Visual similarity search

In the visual search of similar regions ADCS descriptor captures photometric information. However, depending on the type of searched regions, additional geometrical descriptors are necessary to improve the retrieval precision. In addition to ADCS, following features are computed for each region : area, position and compactness. Region area feature is normalized with respect to image area. Position is the region centroid whose coordinates are normalized with respect to image width and height. Compactness is the ratio of the sum of region contour lengths divided by region area; it is maximum for a disc and low for a thin, elongated or irregular shaped region. Integration of sophisticated region shape descriptors was judged irrelevant since regions obtained by segmentation on natural image database are not precise enough. Compactness feature provides sufficient shape information for our problem.

Overall region visual similarity relies on the combination of the 4 following descriptors : ADCS, area, position and compactness using quadratic distance (generalized form), L^1, L^2, and L^1, respectively. The overall similarity measure between a query region R_Q and a candidate region R_C is a linear combination of those four distances :

$$\begin{aligned} d_{final}(R_Q, R_C) = \; & \alpha_{ADCS}.d^{ADCS}_{quad}(R_Q, R_C) + \\ & \alpha_A.d^{A}_{L_1}(R_Q, R_C) + \\ & \alpha_P.d^{P}_{L_2}(R_Q, R_C) + \\ & \alpha_C.d^{C}_{L_1}(R_Q, R_C) \end{aligned} \tag{6}$$

α_{ADCS}, α_A, α_P and α_C are the relative weights of importance of features ADCS, area, position and compactness, respectively. They are set in the query interface, as we will see later, depending on the user requirements.

Since geometric feature distances are much faster to compute than quadratic distance on ADCS descriptor, query time can be reduced by rejecting unlikely candidate regions : for a given query region R_Q and a candidate R_C, if a value

among d_{quad}^{ADCS}, $d_{L_1}^{A}$, $d_{L_2}^{P}$, $d_{L_1}^{C}$ is too high, candidate R_C is rejected and more complex quantity d_{quad}^{ADCS} is not computed. If geometric weights are set to zero, this rejection strategy is not applied since distance is reduced to d_{quad}^{ADCS}. Rejection strategy works as follows :

initialization: $d_{final} = 0$
if ($\alpha_A > 0$ and $d_{L^1}^{A} >$ `AREA_THRESHOLD`)
    so $d_{final} =$ `PSEUDO_INFINITE_VALUE`
else
    $d_{final} += d_{L^1}^{A}$
    if ($\alpha_C > 0$ and $d_{L^1}^{C} >$ `COMPACTNESS_THRESHOLD`)
        then $d_{final} =$ `PSEUDO_INFINITE_VALUE`
    else
        $d_{final} += d_{L^1}^{C}$
        if ($\alpha_P > 0$ and $d_{L^2}^{P} >$ `POSITION_THRESHOLD`)
            then $d_{final} =$ `PSEUDO_INFINITE_VALUE`
        else
            $d_{final} += d_{L^2}^{P}$
            $d_{final} += d_{quad}^{ADCS}$

`PSEUDO_INFINITE_VALUE` is an arbitrary large distance value. `AREA_THRESHOLD` , `COMPACTNESS_THRESHOLD` , `POSITION_THRESHOLD` are defined as half of the maximum distances. We will see in section 8 that this rejection strategy speeds up the search process.

6 User Interface

Our region-based image retrieval system is integrated in IKONA software platform [7], built upon a client/server architecture (server written in C++ and client user interface in Java).

IKONA interface allows browsing database thumbnails (see screenshot 6). The user can click any detected region in each thumbnail to specify the query region. A second window allows to adjust relative importance of features (weights α_{ADCS}, α_A, α_P and α_C) depending on the relevance of geometrical feature for the type of searched regions.
The server retrieves images which have a region which minimizes the visual distance d_{final} to the selected query region. Retrieved regions are identified by white contours in the interface.

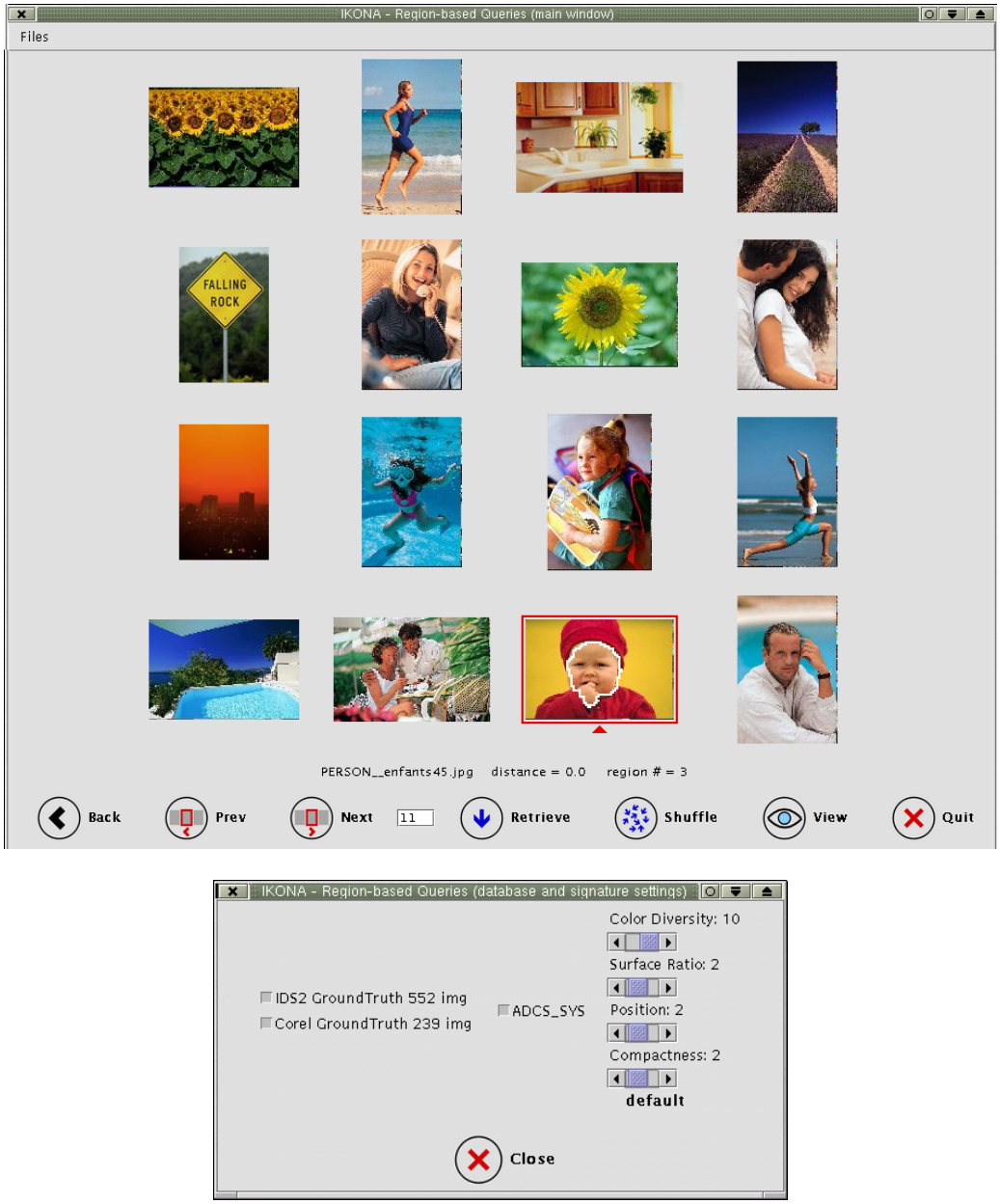


Fig. 10. IKONA region query user interface. A random view of the database (top) and settings window (bottom) to adjust relative importance of color diversity (ADCS), area, position and compactness. Each region in any thumbnail is clickable in the main window.

7 Experiments

Our system was tested with a standard 2GHz/512Mo PC. The test database contains 11.479 images (mostly color and some black and white) including 792

texture images from Vistex database [2], 552 from *Images Du Sud* [3] photostock and 10.135 from *Corel* photostock [4]. The last two databases contain natural images of flowers, drawings, portraits, landscapes, architecture, people, fruit, garden, cars, kitchen, ...

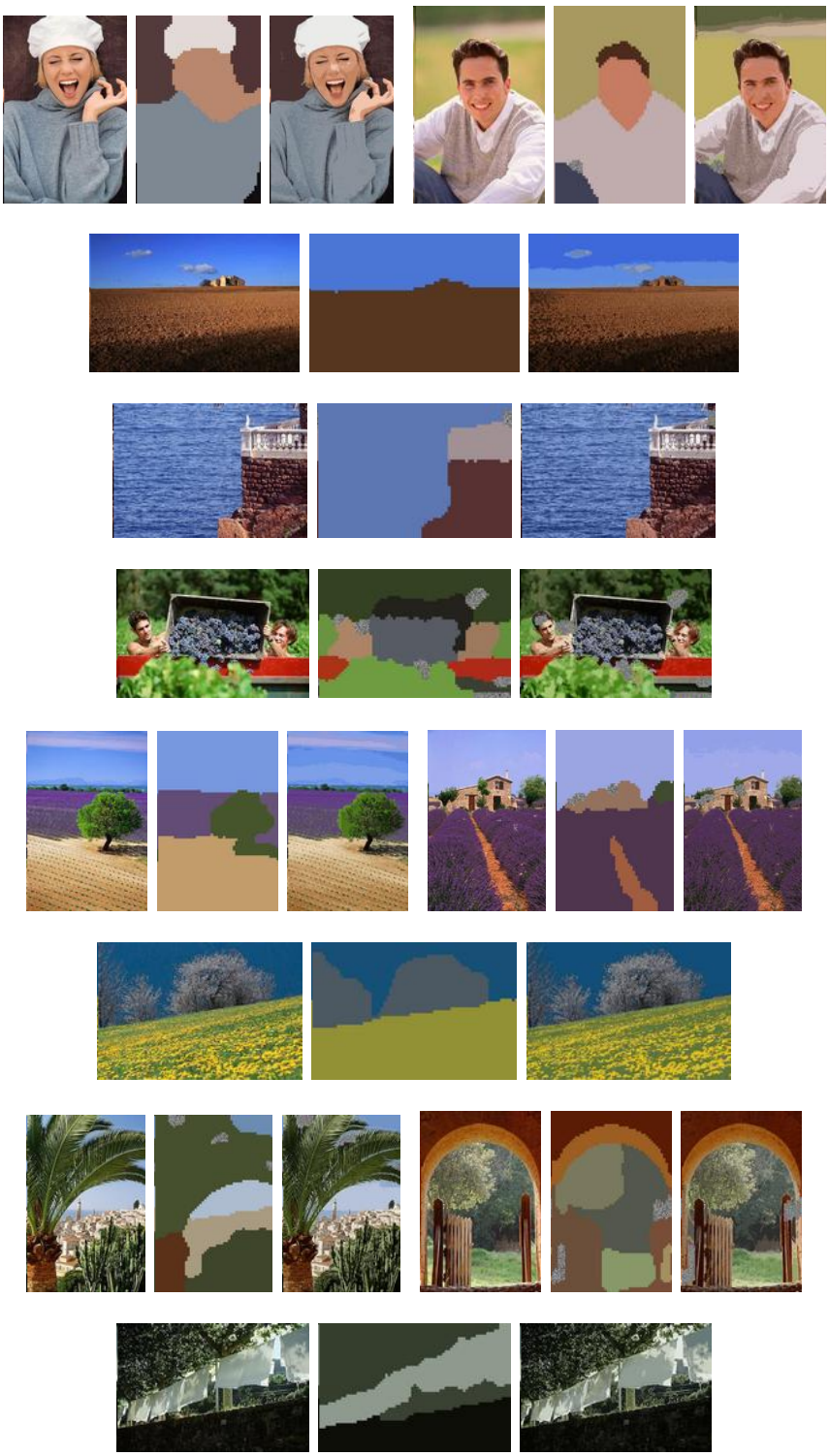
Evaluation of performance of a content-based image retrieval system is a hard task since it depends on human perceptual judgement, on the domain of application and database content (photostock images here). For a region-based system, it is harder since it requires the construction of a region groundtruth database. Our region groundtruth database is partly built from the 88 classes of texture patterns (9 images per class) from Vistex database which can be considered as regions obtained by our coarse detection scheme. The rest of the groundtruth database consists of detected regions, which we have manually labelled them to associate them to one of the three following classes : person (skin region), lavender and swimming pool. The 88 Vistex classes allow to present precision results over a large number of tests and the 3 manually defined classes are used to investigate practical query scenarios. Within each class, regions refer to the same *semantic* object and are also perceptually similar.

8 Results

8.1 Region detection

Even in complex natural scenes extracted regions present a coherent visual appearance and are generally intuitive for the user. The coarse segmentation proves its ability to form regions which can encompass different shades of the same hue, strong textures, isolated spatial details. Such perceptual variability makes each region more specific against other regions in the database. Discarded regions (shown as small grey regions in examples of figure 11) represent a very small fraction of image areas. Hard segmentation cases correspond to scenes with many fine details. For such images, we may consider the alternative of points of interest for partial description as in [9].
Some segmented images are presented in figure 11. More examples can be seen at: `http://www-rocq.inria.fr/~fauqueur/ADCS/` . The segmentation process is fast (1.9 seconds on average per image) which is suitable for large image databases. 56,374 regions were automatically extracted from the 11,479 images (average of 5.2 regions per image on Corel and Image Du Sud images).

[2] http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html
[3] http://www.imagedusud.fr
[4] http://www.corel.com



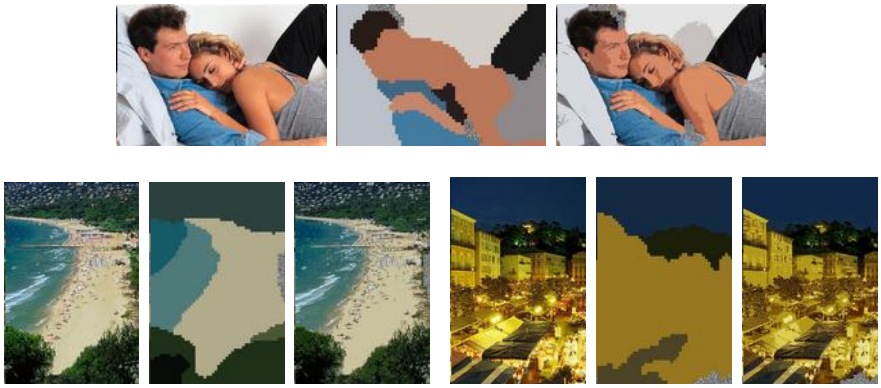


Fig. 11. Illustration of coarse segmentation and fine description. Each triple of images consists of the original image, the image of detected regions represented with their mean color and the image of regions with color shades used for indexing. Small discarded regions are shown in grey. The high perceptual similarity between each original image and the image of color shades shows the accuracy of the ADCS descriptor.

8.2 Region description

In figure 11, the third image of each example is created from the ADCS color shades of each detected region. The high visual similarity between these images and their corresponding original image shows the precision of ADCS color variability description.

A total of 963,215 colors shades from the Luv space was automatically determined to index the 56,374 regions (average of 17 color shades per region). 690,419 of these colors were unique (to be compared with the 200 fixed bins of a traditional histogram). Extracting region ADCS index from an image took 0.8s on average. A region index takes an average of 69 bytes (a scalar is stored as byte) which makes it three times more compact than a traditional color histogram index.

- number of images : 11,479
- number of regions : 56,374
- total number total of color shades : 963,215
- total number total unique shades : 690,419
- number of colors per region : 17
- indexing time per image : 0.8s

Tests were achieved to compare retrieval precision on our region groundtruth database between ADCS and the coarse color representation of traditional descriptors. Comparison was performed with an Luv color histogram based on a systematic subdivision of color space into 6 values per component, which yields 216 bins and using L^1 distance for retrieval.

8.3 Retrieval

Retrieval is performed by exhaustive index comparison with query region index, i.e. all regions in database are compared to the query region. Average retrieval time among the 56,374 regions is 0.8s with sole ADCS descriptor and 0.5s with combination of geometric descriptors (faster time is due to rejection strategy detailed in section 5.3).

8.3.1 Qualitative evaluation

In practice, various types of queries performed with the presented approach always return regions which have a consistent visual similarity with the query region whether it be uniform, textured or encompassing different shades of a given color. Retrieved regions give an impression of visual continuum along the ranks. In comparison with the 216 color histogram, region retrieval with ADCS is slower but always returns regions which are more satisfactory perceptually. When taking into account position, area and compactness information (by setting combination weights to positive values in user interface) the improvement in visual relevance of retrieved regions is usually noticeable, in addition to speeding up search time.

Depending on the type of searched regions, results may be more or less relevant in terms of semantics because semantics and visual description do not always have a one-to-one correspondance. For instance a small and uniformly black query region may depict very different objects as well as shadow areas. More generally, even if a region is correctly detected, we observed that its visual appearance is not always specific to a single class of "objects" in an heterogeneous image database. Conversely some semantic "objects" can have very different visual appearance, such as "dog", "cloth", "car". To build a region groundtruth database we favored regions which present a strong correlation between semantics and visual appearance (such as lavender, skin, swimming pool) for a more meaningful numerical evaluation of performance. In table below, we give some examples of correlation between semantics and visual appearance which we observed when performing region queries :

size	position in image	hue	color variability	likely "object"
large	bottom	white	low	snow
large	bottom	purple	high	lavender field
large	top	blue	shades	sky
not discriminant	*not discriminant*	cyan	low	swimming-pool
small to medium	center	light pink	shades	skin

Table 1. Examples of regions for which a strong correlation was observed between their visual description and their semantics.

8.3.2 Quantitative evaluation

Quantitative evaluation of retrieval precision of our approach was tested by considering each region from the groundtruth database as query region. Groundtruth classes are the following :

- 88 Vistex classes (792 images with 792 labelled regions)
- "lavender" class (108 images and 134 labelled regions)
- "people" class (371 images and 634 labelled regions)
- "swimming pool" class (26 images and 29 labelled regions)

The total number of labelled regions is 1589. Each of these 1589 regions was used as a query region. Among top ranked regions (up to rank 50), precision at rank k is defined as the ratio of the total number of correct retrieved regions up to rank k divided by k. Figure 12 shows the precision curves obtained from these automatic queries using ADCS, 216 bin histogram and the combination of ADCS with geometrical descriptors proposed above. For presentation sake, results on Vistex 88 groundtruth classes are shown on the average over the 88 classes.

In figure 12, precision curves show that for all groundtruth classes, ADCS improves precision compared to traditional color histogram. This improvement is variable depending on classes. The gain in precision is coherent with the finer color representation of ADCS combined to the quadratic distance. Figure 13 illustrates a query scenario on a lavender region using traditional color histogram and ADCS. Figure 14 illustrates a query on a snow region with sole ADCS and the combination of ADCS with geometric descriptors.

We observed that the information of number of color shades (which expresses the photometric complexity) is exploited by ADCS descriptor : regions with many color shades are matched with regions with many color shades and

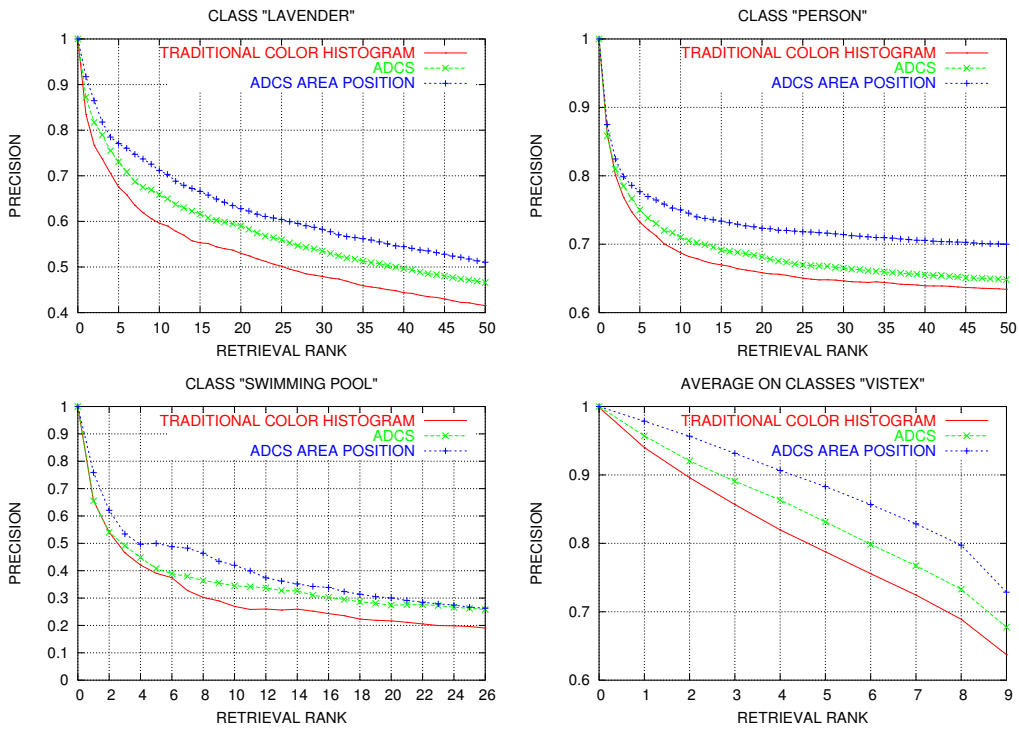


Fig. 12. Precision curves on classes "lavender", "person", "swimming pool" and "Vistex" using 3 indexing schemes: traditional histogram vs. ADCS vs. ADCS combined with area and position.

conversely for regions with one or few color shades. False positives among retrieved regions are due to regions which present similar visual appearance but different semantic content.

Combination of ADCS with the simple presented geometric features leads to another remarkable improvement except for the swimming pool class for which the gain is almost zero. In our groundtruth database, region labelled as swimming pool had similar photometric content but were of different areas at different locations within images. As stated before, combination with geometric features is not necessarily relevant for all types of target regions. However for regions such as *lavender*, *sky*, *skin* or *snow* (see query example of figure 14), they are particularly discriminant.

8.3.3 Scalability and optimization

Although region retrieval is performed by exhaustive search on the database of 11,479 images and 56,374 regions, retrieval time is low (0.5s). We usually consider that two seconds is a maximum for the user to wait for query results. To handle larger databases, beyond 50,000 images, we plan to investigate methods to speed up the region search process. The first solution is to optimize the quadratic distance by using the computationally cheaper bounding distance

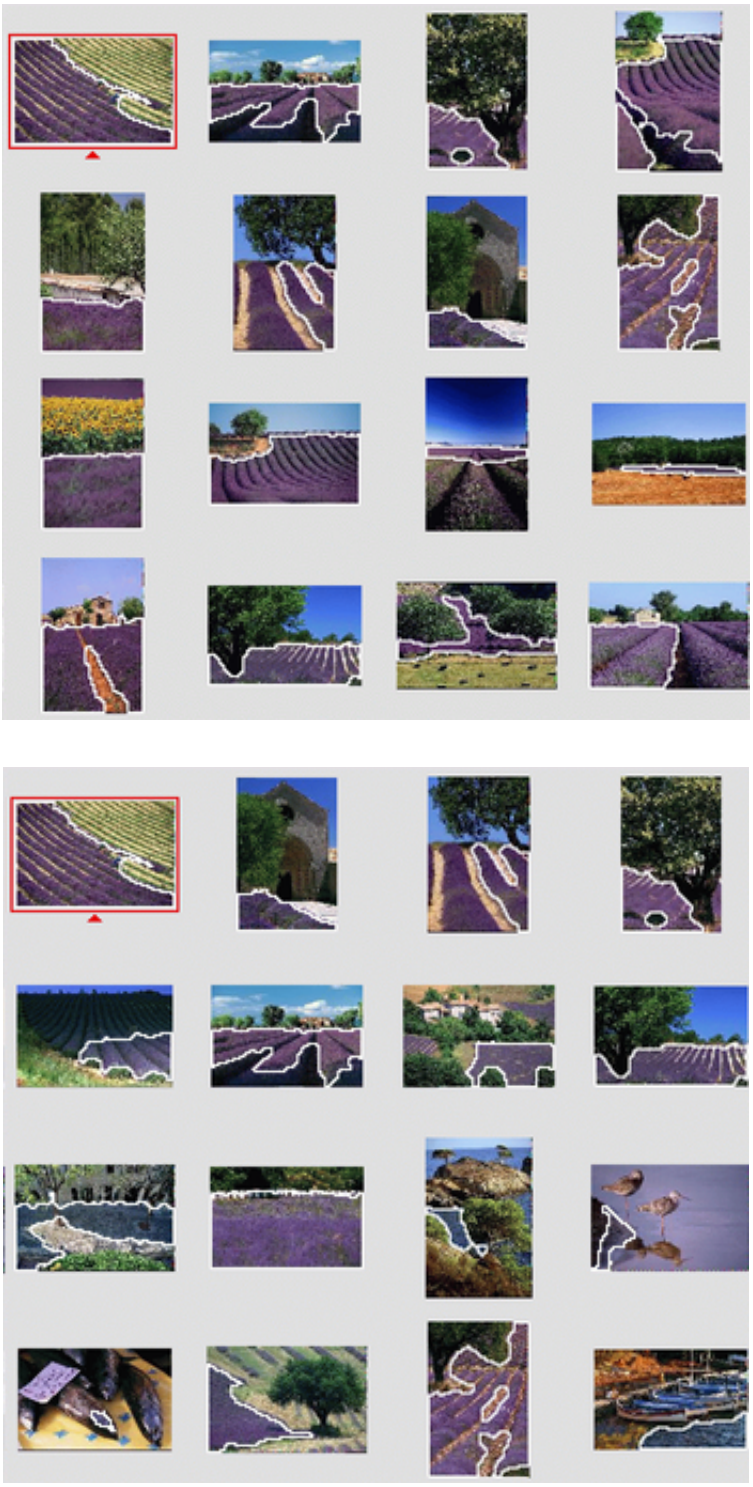


Fig. 13. Retrieval from top-left lavender region: using ADCS (top) and the 216 bin traditional color histogram (bottom). Retrieved regions in images are identified by their white contours. The traditional color histogram can not distinguish shades of purple with shades of blue. No geometrical descriptor is used here.

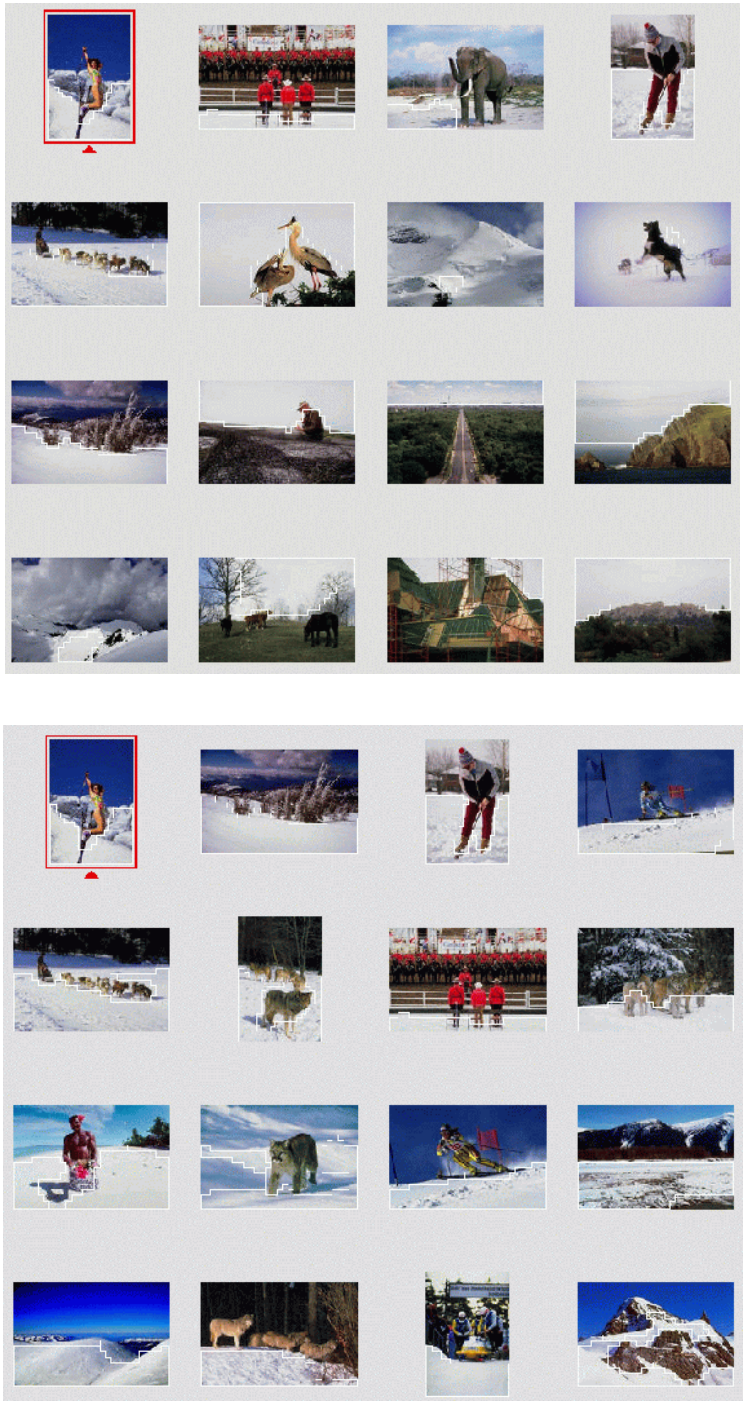


Fig. 14. Retrieval from top-left snow region: using ADCS (top) and the combination of ADCS with the area and position (bottom). Although retrieved regions with ADCS only are relevant in terms of photometry, the size and position greatly improve the quality of retrieval. They are very discriminant for snow region retrieval.

proposed in [27]. The second solution is to prune the search by prestructuring the database using a tree structure, hash table or region categories. Region categories, as defined in [35] for another query paradigm, could be exploited.

9 Conclusions

We presented a novel scheme for coarse automatic image segmentation and fine region description to perform region-based queries in a generic image database. The key idea of this paper is to detect visually specific regions of interest and match them with the fine descriptor to improve the retrieval results.

Segmentation is fast and detects coarse regions which are intuitive for the user. The technique relies on the classification of LDQC's evaluated over large neighborhoods with the generalized form of the quadratic distance as similarity measure. To describe regions, we focused on color since it is known to be the perceptually most relevant photometric descriptor for generic images. The proposed ADCS signature provides a representation of region color variability with more accuracy than existing region color descriptors. It improves the region retrieval precision. We saw that combining ADCS with two simple geometrical descriptors leads to an additional significant gain in retrieval precision.

In future work we plan to investigate methods to speed up the region retrieval process for databases with more than 50,000 images using region categorization and similarity measure optimization. We also consider extending ADCS fine color description and matching scheme to color descriptors which integrate spatial information (such as [30] or [36]). Finally our other prospect concerns multiple region queries using the RAG representation of images.

References

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

[2] A. Pentland, R. Picard, S. Sclaroff, *Photobook: Content-based manipulation of image databases*, SPIE Storage and Retrieval for Image and Video Databases II (1994).

[3] M. Flickner, al.,*Query by image and video content: the QBic system*, IEEE Computer 28 (9) 23–32 (1995).

[4] A. Gupta, al., *The Virage image search engine: an open framework for image management*, SPIE Storage and Retrieval for Image and Video Databases

(1996).

[5] S. Sclaroff, L. Taycher, M. L. Cascia, *Imagerover: A content-based image browser for the world wide web*, IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL) (1997).

[6] T. Gevers, A. Smeulders, *The PicToSeek WWW image search system*, in ICMCS, Vol. 1 pp. 264–269 (1999).

[7] N. Boujemaa, J. Fauqueur, M. Ferecatu, F. Fleuret, V. Gouet, B. Le Saux, H. Sahbi, *Ikona: Interactive generic and specific image retrieval*, International workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR), Rocquencourt, France (2001).

[8] J. R. Smith, S. F. Chang, *VisualSEEk: A fully automated content-based image query system*, ACM Multimedia Conference, Boston, MA, USA, (1996).

[9] V. Gouet, N. Boujemaa, *Object-based queries using color points of interest*, IEEE Workshop on Content-Based Access of Image and Video Libraries, (CBAIVL) (2001).

[10] B. Moghaddam, H. Biermann, D. Margaritis, *Defining image content with multiple regions of interest*, IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL) (1999).

[11] J. Malki, N. Boujemaa, C. Nastar, A. Winter, *Region queries without segmentation for image retrieval by content*, Proc. of International Conference on Visual Information System (VIS'99) pp. 115–122 (1999).

[12] A. Del Bimbo, E. Vicario, *Using weighted spatial relationships in retrieval by visual contents*, IEEE workshop on Image and Video Libraries (1998).

[13] C. Carson, al., *Blobworld: A system for region-based image indexing and retrieval*, International Conference on Visual Information System (1999).

[14] W. Y. Ma, B. S. Manjunath, *Netra: A toolbox for navigating large image databases*, Multimedia Systems, 7 (3) 184–198 (1999).

[15] J. Z. Wang, J. Li, G. Wiederhold, *Simplicity: Semantics-sensitive integrated matching for picture libraries*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) (2001).

[16] J. Fauqueur, N. Boujemaa, *Region-based retrieval: Coarse segmentation with fine signature*, IEEE International Conference on Image Processing (ICIP) (2002).

[17] H. Frigui, R. Krishnapuram, *Clustering by competitive agglomeration*, Pattern Recognition 30 (7) 1109–1119 (1997).

[18] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Functions*, Plenum, New York NY (1981).

[19] N. Boujemaa, *On competitive unsupervized clustering*, International Conference on Pattern Recognition (ICPR'00) (2000).

[20] E. E. Gustafson and W. C. Kessel, *Fuzzy clustering with a fuzzy covariance matrix*, IEEE CDC, San Diego, California (1979).

[21] J. Fauqueur, *Image Retrieval by Regions of Interest*, PhD dissertation, INRIA, (2003), in preparation.

[22] S. Sangwine, R. Horne, *The colour image processing handbook*, Chapman and Hall (1999).

[23] Y. Linde, A. Buzo, R. M. Gray, *An algorithm for vector quantizer design*, IEEE Transactions on Communications COM-28 84–95 (1980).

[24] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proc. of the Fifth Berkeley Symp. on Math. Stat. and Prob. 1 281–296 (1967).

[25] G. Wyszecki, W. S. Stiles, *Color Science: concepts and methods, quantitative data formulae*, John Wiley, New York (1982).

[26] P. K. Robertson, *Visualizing color gamuts : A user interface for the effective use of perceptual color spaces in data displays*, IEEE Computer Graphics and Applications 50–64 (1988).

[27] J. Hafner, H. Sawhnay, al., *Efficient color histogram indexing for quadratic form distance functions*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 17 (7) 729–736 (1995).

[28] T. Pavlidis, *Structural Pattern Recognition*, Springer-Verlag, Berlin (1977).

[29] Y. Deng, B. S. Manjunath, *An efficient low-dimensional color indexing scheme for region based image retrieval*, Proc. IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP'99), Phoenix, Arizona (1999).

[30] B. Manjunath, P. Salembier, T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, Wiley, ISBN: 0-471-48678-7 (2002).

[31] Y. Rubner, C. Tomasi, L. Guibas, *The earth mover distance as a metric for image retrieval*, Stanford University Technical Report (1998).

[32] W. K. Leow, R. Li, *Adaptive binning and dissimilarity measure for image retrieval and classification*, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'01) (2001).

[33] J. Puzicha, J. Buhmann, Y. Rubner, C. Tomasi, *Empirical evaluation of dissimilarity measures for color and texture*, IEEE International Conference on Computer Vision (ICCV) (1999).

[34] S. Gibson, R. Harvey, *Morphological color quantization*, IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'01) (2001).

[35] J. Fauqueur, N. Boujemaa, *Image retrieval by composition of region categories*, IEEE International Conference on Image Processing (ICIP) (2003).

[36] C. Vertan, N. Boujemaa, *Using fuzzy histograms and distances for color image retrieval*, Challenge of Image Retrieval, Brighton (2000).