Journal of Visual Communication and Image Representation **11**, 302–326 (2000)
doi:10.1006/jvic.1999.0428, available online at http://www.idealibrary.com on IDEAL®

Searching Images on the Basis of Color Homogeneous Objects and their Spatial Relationship

Youssef Chahir

Laboratoire HEUDIASYC UMR CNRS 6599, Université de Technologie de Compiègne, Centre de Recherches de Royallieu, BP 20529, 60205 Compiègne Cedex, France
E-mail: ychahir@hds.utc.fr

and

Liming Chen

Département Mathématiques et Informatique, Laboratoire ICTT, Ecole Centrale de Lyon, 36, av. Guy de Collongue, BP 163-69131 ECULLY Cedex, France
E-mail: liming.chen@ec-lyon.fr

Received December 9, 1998; accepted August 5, 1999

In this paper we introduce several techniques which characterize color homogeneous objects and their spatial relationships for a more precise and efficient content-based image searching. We first present a region growing technique for efficient color homogeneous object segmentation, and then we extend the 2-D string to express spatial signatures for an accurate description of spatial relationships of objects within an image. Several optimizations, including dominant color histogram clustering, have also been proposed for an efficient search engine implementation. The experimental results that we have drawn so far show that our content-based image searching techniques give a high precision while maintaining a very good recall rate. © 2000 Academic Press

1. INTRODUCTION

Image has become an important component of today's information systems due to the explosive growth of multimedia data as enabled by recent advances in storage (DVD) and network (ATM). Content-based image retrieval (CBIR) is a key technology for easy visual data management as this is possible with alphanumeric data in large relational databases. In the past several years, CBIR has received a lot of attention in the research community [1–4]. The main problems, which are studied within the field, include the extraction of appropriate image features for possibly semantic object segmentation and the integration of different features to satisfy users' queries.

1047-3203/00 $35.00

As the description of semantic features within an image requires preliminary object segmentation, which is a very hard task, the first work for CBIR relies on global visual features, such as color histogram, which can be easily extracted from an image [5, 6]. If such an approach is simple for effective implementation and has led to interesting results, its drawback is also well known. Indeed, as the color feature is a global one, a very different image from the query image but with the same color distribution is always returned as a pertinent result according to such an approach [7]. Recent work tends to integrate spatial information to different visual features such as color histogram, texture, and shapes [8]. In this paper, we focus on content-based visual information searching. We first propose a new region-growing-based technique to segment color coherent objects from images. Then, we extend the 2-D string and incorporate other geometrical features in order to obtain a precise spatial signature describing the spatial relationships of objects within an image. Finally, several optimizations are introduced for an efficient search engine which fully explores the spatial relationships as described in images' spatial signatures. The experimental results that we have drawn so far show that our content-based image searching techniques give a high precision while maintaining a very good recall rate. The main contribution of our work is twofold:

- Efficient and robust segmentation of color homogeneous objects of an image by the region growing technique;
- Elaboration of a significant spatial signature of an image based on extended 2-D strings, enabling both visual features and fully spatial information and relationship-based querying.

The rest of the paper is organized as follows. Section 2 presents our region growing technique for segmenting color homogeneous objects within an image. Section 3 extends the classical 2-D string for a precise characterization of objects' spatial relationships. Section 4 introduces our search engine implementation and some experimental results are presented in Section 5. Section 6 discusses our work and compares it with others. In conclusion, we summarize our work and give some indications of future directions.

2. REGION-GROWING-BASED COLOR HOMOGENEOUS OBJECT SEGMENTATION

For the purpose of content-based image research, the first step is to index images by segmenting and localizing significant objects which are coherent regions according to a visual feature—the color, in the case of our paper. However, the technique we have developed can apply to other visual features such as gray level and texture. In our previous work, we used a quadtree-based split and merge algorithm to visualize homogeneous objects' extraction [9, 10]. In this work, we consider an inverse approach, the bottom-up one, which consists of region growing by pixel aggregation. In the following, we first define a neighborhood of pixels, which identifies a geometrical neighborhood of pixels, and then we introduce color homogeneity, which is used to aggregate neighbor pixels having a close color. Finally, we present our region growing algorithm for color homogeneous object extraction.

2.1. Neighborhood of Pixels

Let $p(i, j)$ and $q(m, n)$ be two pixels in an image. The distance between p and q is defined as

$$d(p, q) = \text{Max}\{|i - m|, |j - n|\}.$$

p_{36}	p_{35}	p_{34}	p_{33}	p_{32}	p_{31}	p_{30}
p_{37}	p_{16}	p_{15}	p_{14}	p_{13}	p_{12}	p_{29}
p_{38}	p_{17}	p_4	p_3	p_2	p_{11}	p_{28}
p_{39}	p_{18}	p_5	•p	p_1	p_{10}	p_{27}
p_{40}	p_{19}	p_6	p_7	p_8	p_9	p_{26}
p_{41}	p_{20}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}
p_{42}	p_{43}	p_{44}	p_{45}	p_{46}	p_{47}	p_{48}

FIG. 1. The geometric structure of region growing (N-neighbors).

Let ρ be an integer. We define ρ-neighbors of a pixel $p(i, j)$ to be all the pixels which are situated at ρ pixels from p, i.e.,

$$v_\rho(p) = \{\forall q / d_\rho(p, q) \leq \rho\}.$$

As illustrated by Fig. 1, we have:

$$v_1(p) = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}, \text{ and the cardinal of } v_1(p) \text{ is } 8;$$
$$v_2(p) = v_1(p) \cup \{p_9, p_{10}, \ldots p_{24}\}, \text{ and the cardinal of } v_2(p) \text{ is } 24;$$
$$v_3(p) = v_2(p)(\{p_{25}, p_{26}, \ldots p_{48}\}, \text{ and the cardinal of } v_3(p) \text{ is } 48.$$

Generally, for $\rho = n$, we have $v_n(p) = v_{n-1}(p)(\{p_x, p_{x+1}, \ldots p_{\sum 8*i}\}$, and the associated cardinal is

$$\sum_{i=1}^{n} 8 * i.$$

2.2. *Color Homogeneity*

Let C_1 and C_2 be the colors of pixel p and q. The distance between C_1 and C_2 is defined in RGB space by the Euclidean distance

$$d_E(C_1, C_2) = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$$

where $C_1 = (R_1, G_1, B_1)$, and $C_2 = (R_2, G_2, B_2)$. This distance allows us to define a kind of neighborhood of two colors. Indeed, two colors will be considered close if their distance is less than a threshold, say μ. In our implementation, we also proceed to a further color clustering for efficiency consideration. Consider a palette of n colors, for instance 512 as this is usually used in many systems. The cross-correlation between any couple of two colors of the palette is performed in order to identify the perceptual similarity of these two colors. The result is then sorted to obtain a clustering of close colors. A color representative is then chosen to represent each class of color issue from such an operation.

The advantage of performing the clustering of colors in a given palette is to reduce the dimension of the image's color vector without losing the search precision. Thus, it helps decrease the computational complexity in the retrieval process.

2.3. Color Homogeneous Objects' Extraction by Region Growing

A color homogeneous object in an image is a coherent area according to the color neighborhood as defined in the previous section. The region growing process consists of gathering neighbor pixels from a starting point on the basis of color homogeneity. More precisely, the process starts from a pixel and tries to determine whether neighboring pixels belong to the same region of the starting pixel. This process eventually leads to growing a homogeneous region until no more neighboring pixels can be added to the same region on the basis of the color homogeneity.

The following structure characterizes the attributes of a coherent object: a region identifier, its color representative, its minimum bounding rectangle (MBR), its barycenter, and the set of pixels covering the coherent region.

```
struct pixel {
      int flag; /*Pixel identification*/
      int color;  /*Color Of pixel*/
      int x,y;  /*Coordinates of pixel*/
};

typedef vector ⟨pixel⟩ VectorOfPixel;

Struct OBJECT {
    int num_reg;; /*Object identification*/
    int color;
    /* Minimum Bounding Rectangle Coordinates*/
    int xi, xs, yi, ys;
    int x_centroid, y_centroid; /*barycenter*/
    VectorOfPixel Region; /*Set of pixels*/
}
```

Figure 2 depicts the region growing procedure. This function builds a region growing by pixel aggregation and returns its characteristics. The process starts with pixel $p = (x, y)$ and continues to grow within the image I according to a pixel neighborhood threshold ρ until no more pixels can be added to the region. This aggregation process makes use of a stack, *UCN* (unvisited candidate neighbor pixels), in order to memorize the list of neighboring candidate pixels which have not been visited yet. A neighboring pixel according to the neighborhood threshold ρ is considered as a candidate pixel if it has not been visited by the region growing process and has a close color with regard to the color homogeneity threshold μ. Thus, starting from the pixel $p = (x, y)$, the process piles up into UCN, the neighboring candidate pixels.

In order to extract all color homogeneous objects, the region growing process has to be repeated for all unvisited pixels in an image. Figure 3 illustrates the result of such a process.

The original image, as depicted on top in the figure, contains six homogeneous regions. The clustering of the associated color palette gives three classes, respectively represented by colors black, green, and red. The region growing process, applied to the original image, identifies five color homogeneous objects. The first one is an object composed of two neighbor regions, each region having a different color of the same cluster represented by the black color; The second and third objects, a triangle and a circle, have colors of the

```
Function Region_Growing( )
Input:
    P : starting point in the image I;
    ρ: neighborhood threshold  ; {1, 2, or 3}
    μ : color homogeneity threshold ;
Output :
    the color coherent region R covering the starting pixel P;

Initialization :
    Pile up the starting pixel P into UCN ;
    {for the MBR}
    xi=xs=P.x;
    yi=ys=P.y;
    {for the barycenter}
    sx_centroid=0; sy_centroid=0;
    nb_pixels=0;
    {for the color representative of the region}
    Mean_Color=color_of_P ;

Step 1: {A new coherent pixel is added to the region}
    pop out a candidate pixel, say Q,  from UCN;
    insert Q into R.Region;
    mark pixel Q as visited;
    update number of pixels of the 4 quadrants
    {determine the coordinates of the new MBR}
    if (Q.x < xi) xi = Q.x;
    if (Q.y< yi) yi = Q.y;
    if (Q.x > xs) xs = Q.x;
    if (Q.y> ys) ys = Q.y;
    {accumulation for the computation of the barycenter}
    sx_centroid += Q.x;    sy_centroid += Q.y;
    {increment of the number of pixels}
    nb_pixels++;
    {Mean_Color is the average color value}
    update Mean_Color

Step 2 : {computing the neighbor candidate pixels}
    For each neighbor pixel of Q, say T,  w.r.t. ρ
        If  T is not visited yet and T's color is close to Q's
        one according to μ, then pile up R into UCN ;

Step 3 : {iteration}
    Repeat Steps 1 and 2 until the pile UCN is empty ;

Step 4 :  {Outputting the final result of the region R}
    {computing the MBR}
    R.xi = xi ;R.yi=yi;
    R.xs=xs; R.ys=ys;
    {the color representative}
    R.color = Representative color of  Mean_Color;
    {computing the barycenter}
    R.x_centroid = sx_centroid /nb_pixels;
    R.y_centroid = sy_centroid /nb_pixels;
```

FIG. 2. Region growing pseudo-algorithm.

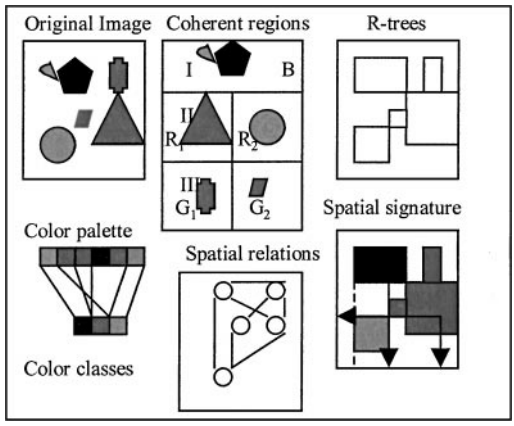


FIG. 3. Segmentation and indexing process.

third cluster represented by the red color. Finally, the fourth and fifth objects are coherent regions with colors of the second cluster represented by the green color.

3. IMAGE SPATIAL SIGNATURE BY OBJECT SPATIAL RELATIONSHIP CHARACTERIZATION

3.1. 2-D-R++-String Signature

Object spatial-relationship identification aims at characterizing spatial arrangement relationships among objects, giving birth to an image spatial signature. Encoding spatial information in the representation of an image not only improves the retrieval quality but also enables users to query databases of images based on the spatial arrangement of subimages. Most approaches for describing spatial relationships among objects consist of using a single content descriptor such as 2-D-string [11] and its variants (2-D-E-strings [12], 2-D-G-strings [13], 2-D-C-strings [14], and 2-D-B-strings [15].

The major drawback of such a representation is its fuzziness, as its associated spatial operators are not sufficient to give a complete description of the spatial relationships that can exist among objects. For instance, when an object is included in another one and these two objects have the same barycenter, the 2-D string representation cannot express this particular spatial relationship. Figure 4 contains four images illustrating such fuzziness.

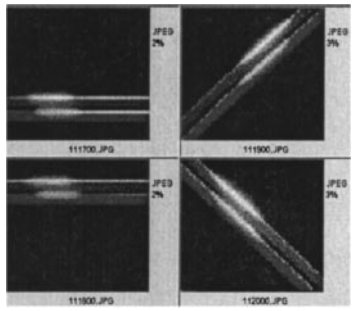


FIG. 4. Images that have the same 2-D string but which are spatially different.

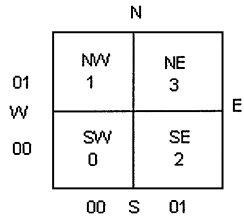


FIG. 5. Image quadrant labeling.

This consideration has led us to extend 2-D string representation with other features, which have been extracted during the object segmentation process. The best representation of a spatial relationship among objects would take into account all points of the contour of each object, but such a method would greatly increase the computational complexity as well as the storage requirements.

To fully capture the spatial relationship among objects in an image, we have proposed the 2-D-R++ string. First, MBR is used to frame objects within an image. Each object is then localized by a Peano key [6], which gives the spatial position of the center of gravity, associated with the MBR. The basic 2-D string is extended to include, for each object, its MBR, its barycenter, its relative area, and a chain code, which aims at giving more precise information on the spatial distribution of the object. Indeed, an image is divided into four quadrants: 0 (SW), 1 (NW), 2 (SE), and 3 (NE) [16]. While segmenting a color coherent object, the region growing process also looks at the distribution of the object into the four quadrants, sorting them according to the number of pixels of the object on each quadrant.

Each image has four edges NW, NE, SW, and SE. Figure 5 illustrates the labeling of image quadrants.

Our approach consists of adding a set of new symbols which give information on the spatial positions of objects, thus enabling a more precise description of spatial relationships such as *overlap inverse*, *contain*, *meet*, *begin*, *end*, *overlap*, *equal*, and *less than*. Once these spatial relationships are represented, the image retrieval using these spatial relationships becomes a classical 2-D sequence matching. The set of MBRs together with all spatial features are indexed using the R-tree structure [17], providing a dynamic structure for rectangles indexing.

Each object in the image is framed by a MBR, thus having two pairs of begin and end coordinates, one for the x-axis and the other for the y-axis. The following features are added to the spatial operators [18] which are recalled in Table 1.

- Each object R with size s is denoted as R_s, where s is the total pixels of the object.

TABLE 1
Definition of Spatial Operators

Notations	Conditions
A < B	end(A) < begin(B)
A = B	begin(A) = begin(B), end(A) = end(B)
A\B	end(A) = begin(B)
A % B	begin(A) < begin(B), end(A) > end(B)
A] B	begin(A) = begin(B), end(A) > end(B)
A [B	begin(A) < begin(B), end(A) = end(B)
A / B	begin(A) < begin(B) < end(A) < end(B)

◆ An object R is denoted ${}_{\chi}R$, where χ is the chain code of R giving the object distribution into quadrants in a decreasing order.

◆ Operator "$<$" is extended to include the distance d between two objects A and B, denoted as A $<_d$ B, where $d = \text{begin(B)} - \text{end(A)}$.

◆ Operator "%" is extended with the two distances (d, d') between objects A and B, denoted as A $\%_{d,d'}$ B, where $d = \text{begin(B)} - \text{begin(A)}$ and $d' = \text{end(A)} - \text{end(B)}$.

◆ Operator "]" is enriched with the distance d between objects A and B, denoted as A $]_d$ B, where $d = \text{end(A)} - \text{end(B)}$.

◆ Operator "[" is extended with distance d between objects A and B, denoted as A $[_d$ B, where $d = \text{begin(B)} - \text{begin(A)}$.

◆ Operator "\" is parameterized with distances d_1, d_2, d_3 between objects A and B, denoted as A $\setminus_{d_1,d_2,d_3}$ B, where $d_1 = \text{begin(B)} - \text{begin(A)}$, $d_2 = \text{end(A)} - \text{begin(B)}$, and $d_3 = \text{end(B)} - \text{end(A)}$.

3.2. Spatial Relationship Graph

Once the color coherent objects are segmented by the region growing process, an image is also described by the spatial relationships of the objects within the image. All these spatial relationships are captured by a complete weighted graph called the spatial relationship graph (SRG). In such a graph, each node represents an object within the image while each vertex represents all spatial relationships captured by spatial operators between two objects. Indeed, we have simplified the representation of such a SRG in our implementation. In our current prototype, each node of a SRG only contains the color representative of the underlying object, although other features associated with an object may be added to the node later.

3.3. Example

Figure 6 is a picture of a young lady. Figure 7 gives the same image obtained after the region growing segmentation process. Figure 8 is the image where we only keep the first six dominant homogeneous regions whose features are detailed in Table 2. Figure 9 illustrates the spatial arrangement of the minimum bounding rectangles associated with these first six homogeneous regions.

FIG. 6. (a_0) Example image.

FIG. 7. (b_0) Segmented image by the region growing method.

Each homogeneous region has a MBR (xi, yi, xs, ys), a barycenter (x_centroid, y_centroid), a size (s), and a chain code (c).

In this example, each homogeneous region is labeled by a letter (A . . . F). In our current prototype, each homogeneous region is represented simply by its color representative.

Now we show the way we obtain the 2-D-R++ strings which characterize spatial relationships of the six dominant homogeneous objects, using the features added to spatial operators in Table 1 and applying them to the image in Fig. 8.

2-D-R++-strings. According to Table 2, we can identify the dominant homogeneous regions by the following representation:

$$_{32}\mathrm{A}_{30579},\ _{10}\mathrm{B}_{27064},\ _{02}\mathrm{C}_{11104},\ _{3102}\mathrm{D}_{3891},\ _{13}\mathrm{E}_{3463},\ \text{and}\ _{02}\mathrm{F}_{2050}.$$

▲ 2-D-String based on the barycenter of the homogeneous regions: (B $<_{101}$ E $<_6$ C $<_5$ D $<_1$ F $<_{112}$ A, E $<_{47}$ B $<_{15}$ A $<_5$ D $<_{77}$ C $<_{62}$ F)

▲ 2-D-String based on MBRs: (B $<_{79}$ C $<_2$ F $<_{38}$ E $<_{15}$ B $<_7$ D $<_{57}$ D $<_6$ E $<_{13}$ A $<_{14}$ F $<_6$ C $<_{99}$ A, A = B $<_{38}$ E $<_{44}$ D $<_{32}$ E $<_{36}$ C $<_{41}$ D $<_{58}$ F $<_{26}$ A $<_1$ B $<_2$ C $<_4$ F)

FIG. 8. (b'_0) Image represented by its first six dominant homogeneous regions.

TABLE 2
Selection of Dominant Homogeneous Regions Sorted by Decreasing Order

Homogeneous Regions	MBR	Gravity center	Size	Chain code
 A	(218,6) (337,281)	(281,136)	30579	32
 B	(1 , 6) (135 ,280)	(56,121)	27064	10
 C	(80,156) (238, 282	(163,218)	11104	02
 D	(142,88) (199, 197)	(168,141)	3891	3102
 E	(120, 44) (205, 120)	(157, 74)	3463	13
 F	(82,255) (232,286)	(169,280)	2050	02

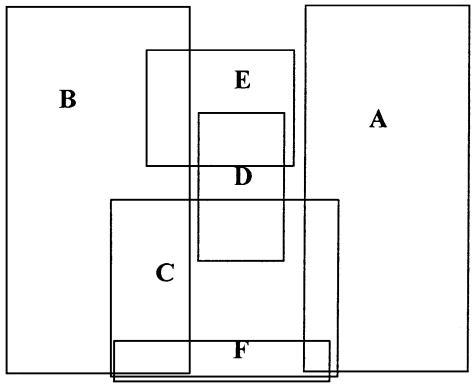


FIG. 9. MBRs of dominant homogeneous regions.

TABLE 3
A Summary of Spatial Relationships between Homogeneous Regions of the Image in Fig. 8

$\Gamma_{x,y}$	A	B	C	D	E	F
A	*	B < A B] A	C/A A/C	D < A A % D	E < A A % E	F/A A/F
B	*	*	B/C B/C	B < D B % D	E/B B % E	B/F B/F
C	*	*	*	C % D D/C	C % E E < C	C % F C/F
D	*	*	*	*	E % D E/D	F % D D < F
E	*	*	*	*	*	F % E E < F
F	*	*	*	*	*	*

The first component of this string is the projection of all homogeneous regions on the X-axis while the second is on the Y-axis.

We can deduce from this representation that the largest part of the homogeneous region A is localized in quadrant NE; moreover the remainder is in quadrant SE and is composed of 30,579 pixels.

For instance, B < C in the first component indicates that the homogeneous region labeled by the color representative B is at the left of the homogenous region labeled by C.

Table 3 summarizes all spatial relationships between homogeneous regions, and Table 4 simplifies their notations.

As we can see in Fig. 9, the spatial relationship is (B < A, B] A) between homogeneous regions A and B. In Table 4, we note that the spatial relationship for the couple (A, B) is $(-<, -])$. This means that the sense of the direction is B < A and B] A. Generally the spatial operator Γ between any couple of homogeneous regions (U, V) is $(-\Gamma)$, which means that the right spatial relationship on the axis is $V\ \Gamma\ U$. Obviously, while computing the 2-D-string corresponding to barycenters, we also store distances for precise querying. For instance, if Γ is "]", we memorize $A\ ]_d\ B$, where $d = \text{end}(A) - \text{end}(B)$.

Once the 2-D-R++-string representation is computed, we build a complete graph, called the *SRG*, to summarize and identify spatial orientation between any couple of homogeneous regions [19]. Figure 10 gives the SRG associated with the image in Fig. 8.

TABLE 4
Simplified Notation of Spatial Relationships as It Is Implemented for the Image in Fig. 8

U	V				
U $\Gamma_{x,y}$ V	B	C	D	E	F
A	$(-<, -])$	$(-/, /)$	$(-<, \%)$	$(-<, \%)$	$(-/, /)$
B		$(-/, /)$	$(<, \%)$	$(-/, \%)$	$(-/, /)$
C			$(\%, -/)$	$(\%, -<)$	$(\%, /)$
D				$(-\%, -/)$	$(-\%, <)$
E					$(-\%, <)$

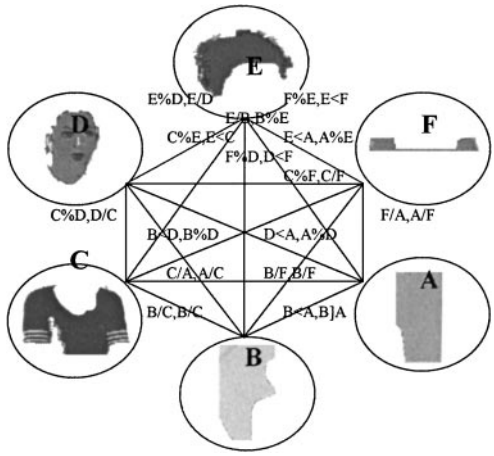


FIG. 10. The SRG corresponding to the image in Fig. 8.

4. SEARCH ENGINE IMPLEMENTATION

For searching images from the database, a user gives a query image, which may be selected from the database. This query image is then compared with all images of the database, first on the basis of global color histogram comparison to perform a first filtering and then on the basis of spatial signatures. For instance, on a base of 22,500 images, the filtering operation using the global color histogram comparison usually selects a set of 100 to 150 candidate images to which spatial signature comparison is then performed. Here we assume that the input of the search engine is the spatial signature of the query image. If the query image comes from the database of images, we already have its spatial signature calculated during its insertion into the base. When the query image is not selected from the database, we first apply our region growing procedure to segment a color homogeneous object and determine its spatial signature as explained in the previous two paragraphs.

4.1. Color Histogram-based Filtering

The color histogram is the color distribution in an image; thus, it is a global visual feature. Its major advantages are its simplicity in computation and its insensitivity with respect to image rotation and translation. In order to enable a correct comparison of images, we first normalize the color histogram and then a similarity is computed between two images using a similarity metric.

A color histogram is normalized by the formula

$$H^N(\mathrm{im}_n, i) = \frac{H(\mathrm{im}_n, i)}{\sum_i H(\mathrm{im}_n, i)},$$

where $H(\mathrm{im}_n, i)$ is the histogram of an image im_n. The index i represents a histogram bin. A similarity distance between two color histograms, $H(\mathrm{im}_n)$ and $H(\mathrm{im}_m)$, each consisting

of n bins, is quantified by the following metric,

$$d_E^2(H(\text{im}_m), H(\text{im}_n)) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}(H(\text{im}_m, i) - H(\text{im}_n, i))\,(H(\text{im}_m, j) - H(\text{im}_n, j)),$$

where the matrix a_{ij} represents the similarity between the colors corresponding to bins i and j, respectively. This matrix needs to be computed from human visual perception studies. Note that if a_{ij} is the identity matrix, then this measure becomes a Euclidean distance.

4.2. *Optimization Based on the Colors of Dominant Objects*

Actually only a small number of colors, formed by the colors of dominant objects within an image, is used to construct an approximate representation of color distribution in our implementation. This simplification does not alter the precision of the search result, since only colors of dominant objects are most visually significant and capture the human attention, as observed by Gong *et al*. [20]. Our current implementation makes use of this observation and limits color classes to the colors of the first 30 dominant color coherent objects ($\text{Th}_4 = 30$). Furthermore, in our experiments, a homogeneous region is ignored if its size (number of pixels) is less than $\text{Th}_3 = 0.3\%$ of the image.

Figure 11 also gives an illustration of this optimization applied on original images (a_i). The images on the right give the optimized images; they are obtained in keeping only the dominant homogeneous regions.

In our previous implementation, we used the first 50 dominant colors for this optimization. The experiments that we drew have led us to consider rather the colors of dominant objects within an image, because a color may be dominant but it may not represent dominant objects which are the most visually significant within an image.

4.3. *Spatial Signature-Based Images' Comparison*

The color histogram-based filtering selects from the database a set of candidate images as results to the query image. The further step consists of using spatial signatures to compute a list of these candidate images sorted by their similarity as compared to the query image.

The comparison process based on spatial signature has as input the SRG of the query image. Recall that a SRG captures all the spatial relationships of objects within an image. For each pair of nodes in the SRG, say A and B, and for each spatial relationship, say $\text{A} < \text{B}$, between these two objects represented by their color A and B, a vote is incremented for each candidate image having also such a spatial relationship $\text{A} < \text{B}$. The final result is obtained by sorting candidate images in decreasing order on the basis of the votes obtained by each candidate image.

Table 5 summarizes the attributes characterizing color homogeneous objects such as color representative, spatial distribution, MBR, and size.

The searching process makes use of SRG extracted from the images in the database. For efficiency consideration, a kind of "inverted file" is implemented to represent dominant homogeneous regions from images (see Table 6) and the SRG of all the images in the database. Indeed, to each spatial relationship, Γ, is associated a file of the form as illustrated by Table 7. For instance, the first line contains in the left column the spatial

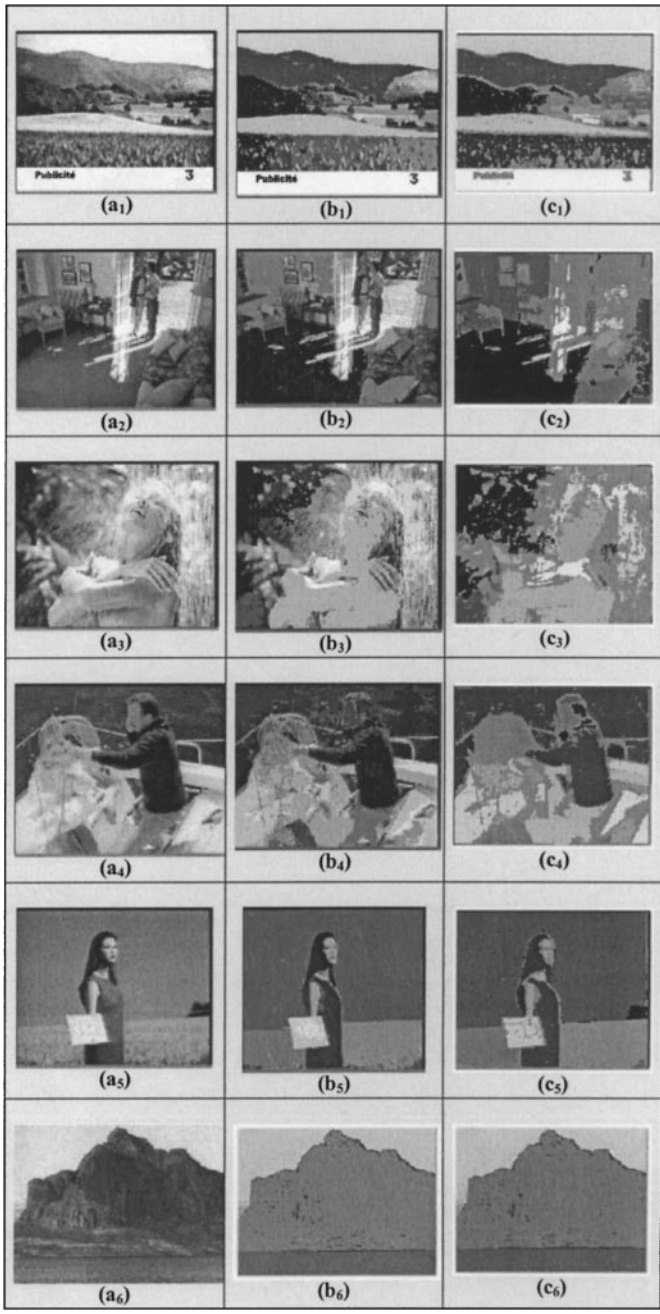


FIG. 11. Segmentation results. (a) orignals, (b) segmentation results, using ρ and μ, and (c) results when also using Th_3 and Th_4.

arrangement of two object colors, $C_1 \Gamma C_2$, and the right column gives all image identifiers having the same color spatial arrangement together with the number of repetition. Thus, $I_5(4)$ indicates that image I_5 contains four pairs of objects having the same spatial arrangement $C_1 \Gamma C_2$. This file is updated each time an image I is inserted into the database.

TABLE 5
Homogeneous Regions and Their Attributes

Region ID	C_R	(x_i, y_i)	(x_s, y_s)	(x_c, y_c)	Area	Code
${}_1R_1$	C_1	${}_{11}P_i$	${}_{11}P_s$	${}_{11}P_c$	${}_{11}S$	${}_{11}C$
${}_1R_2$	C_5	${}_{12}P_i$	${}_{12}P_s$	${}_{12}P_c$	${}_{12}S$	${}_{12}C$
...	...					
${}_nR_m$	C_x	${}_{nm}P_i$	${}_{nm}P_s$	${}_{nm}P_c$	${}_{nm}S$	${}_{nm}C$

Note. Color representative, C_R; MBR coordinates $[x_i, y_i, x_s, y_s]$; region barycenter (x_c, y_c); region size, area; and code chain, code. Peano keys represent the coordinates.

We can see that the crossing of the SRG composed of six nodes, representing the six homogeneous regions, necessitates 15 operations.

Generally, the crossing of a SRG composed of n nodes necessitates S operations, as described by the following formula:

$$S = (n-1) + (n-2) + \cdots + 1 = \sum_{i=1}^{n}(n-i) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}.$$

For each image composed of n homogeneous regions, we must memorize $n(n-1)/2$ spatial relationships. Each spatial relationship concerns two directions as regard the horizontal and the vertical axis.

Figure 11 shows other results of color-based image segmentation resulting from the region growing process. For each image, Fig. 11a shows an original image while Fig. 11b gives the result from the region-growing-based segmentation. Images are respectively in the format 320×240, 352×288, 352×288, 352×288, 352×288, 320×240, and each pixel is represented by the RGB color coordinates with eight bits of precision (24 bits/pixel).

In addition to the threshold value used for a neighborhood of pixels ($\mathrm{Th}_1 = \rho = 3$ pixels) and the one for color homogeneity ($\mathrm{Th}_2 = \mu = 6$), we also consider two other threshold values, Th_3 and Th_4, for performance consideration:

- Th_3 is the threshold value for a minimum number of pixels in order to consider a homogeneous region as significant. It is formulated as a percentage of the image. $\mathrm{Th}_3 = 0.3\%$.
- Th_4 is the threshold value for the maximum number of the most dominant homogeneous regions that we consider. It is limited to 30.

Table 8 gives various experimental results with regard to these images.

During the searching process, for each spatial relationship Γ extracted from the query image, for instance, $C_i \, \Gamma \, C_j(m)$ indicating there are m pairs of objects having colors C_i and C_j and satisfying the spatial arrangement Γ, we look directly into the line $\langle C_i, C_j \rangle$

TABLE 6
File Structure Associated with Images

Image identifiers	Corresponding regions
I_1	${}_1R_1, {}_1R_2, {}_1R_3$
I_2	${}_2R_1, {}_2R_2, {}_2R_3, {}_2R_3,$
...	
I_n	${}_nR_1, {}_nR_2, {}_nR_3, \ldots {}_nR_m$

TABLE 7
File Structure Associated with Each Spatial Relationship Γ

Pairs of object colors	Image identifiers
C_1, C_2	$I_1(2), I_5(4), I_{89}(1)$
C_2, C_{15}	$I_5(4), I_{49}(1), I_{113}(3), I_{133}(7)$
...	
C_i, C_j	$I_x(u), I_y(v), \ldots, I_z(w)$

in the file Γ and obtain all the image identifiers of the database having the same spatial arrangement $C_i \Gamma C_j$. Each image identifier, for instance $I_x(u)$, increments then its votes by u if u is less than m; otherwise m as the maximum number of the spatial arrangements in the query image is m.

The complexity for the vote algorithm is $\theta(S \times c)$, with $S = n(n-1)/2$, where n is the number of the extracted homogeneous region and c is the number of candidate images.

In the worst case, for a given line, we find all c candidate images ($c = 100$ to 150 images) selected after filtering by comparison between histograms.

In principle, there is a vote for each needed additional precision. The time of vote grows in the following order:

- Vote for the same or closet spatial arrangement
- Vote for the comparison between sizes of objects
- Vote for the comparison of distances between each couple of objects.

The choice of a good mechanism of a vote for all these additional conditions is a crucial problem. The mode of vote must take account of the satisfaction of partial conditions, for example, when a portion of an object respects the same spatial distribution like a query object.

For this reason, in our experiences, we do not yet take account of the size or of the spatial distribution of objects within an image.

TABLE 8
Number of Regions before and after Thresholding and Time Complexities

Image	R_T	R_E	R'_E	T_1	T'_1	T_2
a_0	29918	4401	11	5	4	1
a_1	56649	12942	11	7	4	2
a_2	60215	18232	11	7	5	3
a_3	69609	17695	13	8	5	3
a_4	59445	14955	19	7	5	3
a_5	49514	7858	9	5	4	2
a_6	49502	3058	4	3	2	1

Note. R_T, the total number of connected components, without any thresholding, which corresponds to Fig. 11a. R_E, the number of homogeneous rigions extracted by the region growing segmentation process, thus only taking account of thresholds ρ and μ. The images correspond to Fig. 11b. R'_E, the number of homogeneous regions really considered for the spatial relationship characterization, thus using all thresholding (Th_1, Th_2, Th_3, and Th_4) which corresponds to Fig. 11c. T_1, CPU time (in seconds) related to the region growing segmentation algorithm in Fig. 2 when is applied to all R_E regions extracted (Fig. 11b). T'_1, CPU time (in seconds) related to the region growing segmentation process in Fig. 2 when it is only applied to R'_E homogeneous regions extracted (Fig. 11c). T_2, CPU time with regard to the extraction of 2-D-R++-string and the construction of the SRG (Fig. 11c). All measures of these CPU times (in seconds) are realized on a PC Pentium-Pro 200 platform.

5. EXPERIMENTAL RESULTS

Current implementation of our program runs on the PC Pentium-Pro 200 platform. The system has been implemented using Visual C++ 4.0 Standard Template Library. The tests have been performed on a set of 15500 test images. The results have been judged in these experiments by human observers.

5.1. Searching Based on Colors of Dominant Objects

The first query is to retrieve images based on a color histogram. The search process has been made by comparison of colors of the first thirty dominant objects. Figure 12 illustrates some results of images retrieved (the first 28 closest images).

Query images and the first 28 most similar images extracted by the search engine from the database are shown in Fig. 12, sorted according to decreasing similarity, first by line then from left to right.

As we can see, the experimental result provides a sorted list of images whose colors of dominant objects are the closest to the ones of the query image.

For queries based on colors of dominant objects, the query response time varies between 10 and 15 s.

5.2. Spatial Signature-Based Searching

This kind of query also makes use of the spatial relationship among the dominant objects, and images that resulted from the searching process are also sorted by their spatial relationship similarity degree, which is measured by the weight of votes. The more votes a candidate image obtains, the closer spatial arrangements it has between any couple of objects as compared to the query image.

As we can see in Fig. 13a, the first images that follow the query image are those having the closest disposition of their dominant objects as compared to the one in the query image, that is sky on mountain on water. The best matches have a similar arrangement among the dominant regions, which are also similar.

Figure 13b gives another example of image retrieval based on the spatial disposition of dominant objects. The query image shows a brown horse on the green meadow. As we can see in the figure, all images retrieved respect this spatial arrangement such that the most dominant object, with a green color (or a close color) representing the meadow, contains another one with a brown color or a close color, representing the animal.

For all queries using a spatial signature of dominant objects, the query response time varies between 10 and 20 s.

5.3. Recall and Precision Assessment of the Searching Engine

For retrieval effectiveness assessment, we calculated two rates in terms of *recall* and *precision* [21]. Recall that the rate of precision is defined as the ratio of relevant images in the answer on the images retrieved, while the rate of recall is defined as the ratio of the relevant images in the answer on all relevant images in the database.

Let α be the set of images retrieved as a response to the query, and let β the set of images in the database that are considered to be relevant to the query. Then we have

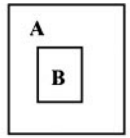
$$\text{Precision} = (\alpha \cap \beta)/\alpha$$

$$\text{Recall} = (\alpha \cap \beta)/\beta.$$

FIG. 12. Query results based on dominant colors.

FIG. 13. (a) First set of experiments based on spatial arrangement. (b) Second set of experiments based on spatial arrangement.

b



In the following queries, we search the same arrangement like the arrangement between two homogeneous regions A and B represented by their MBR and their color representatives.

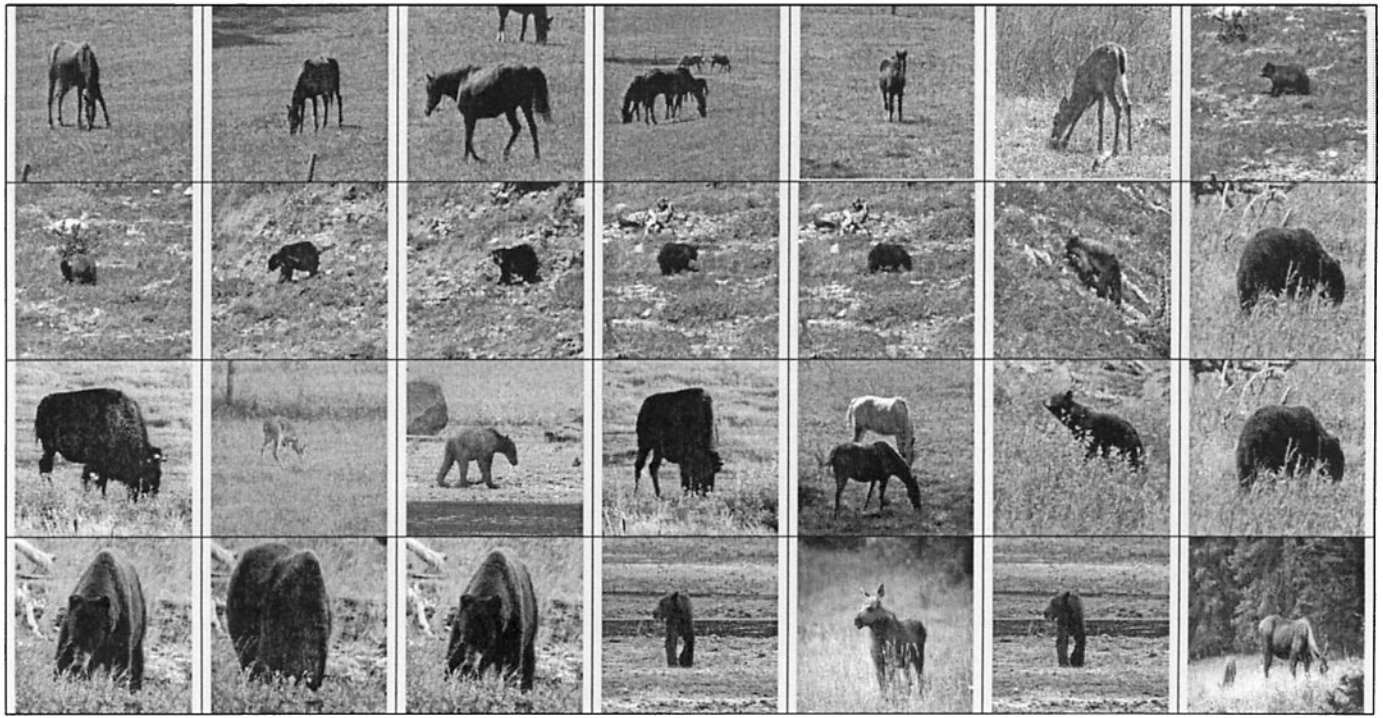


FIG. 13—*Continued*

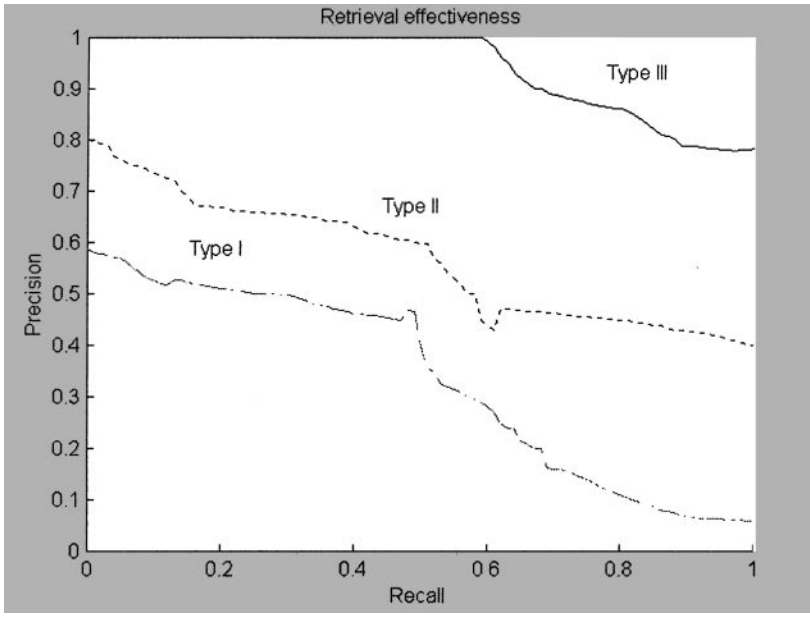


FIG. 14 Retrival effectiveness of 100 generated queries on a database of 15,500 images with three query methods: Type I, Type II, and Type III.

Figure 14 shows three different curves comparing experimental results in terms of precision and recall rates of three search results using three different search criteria. In this experiment, 100 arbitrary query images were used to evaluate the search engine quality.

The Type I curve corresponds to the precision/recall rate when only a color histogram is used to retrieve the most similar images from the database as compared to query images; the Type II curve is the precision/recall rate when dominant objects are used for the retrieval; and finally the Type III curve measures also the retrieval effectiveness when spatial relationships among color homogeneous objects are expressed in 2-D-R++ string, and the SRG with the object represented by its MBR and its barycenter is used to describe spatial relationships of objects and to answer the query.

As we can observe from the experimental results in Fig. 14, results of queries using colors of dominant objects (Type II) performs much better than queries only using a color histogram (Type I). We also see that the color/spatial query strategy (Type III) based on spatial operators performs much better than a query based on colors of dominant objects. Indeed, while the worst precision rate is obtained when the search engine uses only a color histogram, the best precision rate, above 0.8 for all 100 query images is achieved by a search using both colors of dominant objects and their spatial relationships. The use of colors of the dominant objects already improves the color histogram-based search, while taking spatial relationships into account gives further improvement on the rate of precision.

Most importantly, while Type II or I queries may improve the recall rate to the detriment of the precision rate, the Type III query always gives a recall rate above 0.6 while keeping the precision rate above 0.8.

However, as the searching process is based on the arrangement of the dominant homogeneous regions rather than semantic contents of images, it also suffers from the same

limitation as low level feature-based image retrieval. It is observed that in certain classes of images the results depend heavily on the dominant colors of the dominant objects in the images, which could be background, foreground, or large objects. Consequently, the retrieval effectiveness may vary with the type of application domain.

6. RELATED WORK AND DISCUSSION

The principal characteristic of our segmentation method is that it takes into account both color similarity and spatial proximity.

Region growing is a simple and sophisticated method for image segmentation which was proposed in a context other than content-based image retrieval [22]. Usually the performance of region growing methods suffers from noise in the image and therefore the segmentation result becomes inaccurate. This fact led us to apply the region growing method to an image divided into quadrants and base our pixel aggregation on dominant colors with 3-neighbor pixels. On the characteristics of a pixel and its neighborhood, several criteria based on spatial similarity can be used to decide whether a pixel belongs to a homogeneous region. These criteria can be defined from local or global considerations. In our approach we have used the neighborhood homogeneity criterion based on color homogeneity which corresponds to a local comparison between adjacent pixels and its neighbors.

Most color quantization processes use techniques based on thresholding color histograms without taking into account contextual information relative to spatial repartition. The results obtained usually do not corroborate to visual judgment. Note that this thresholding can also be viewed as a classification problem for which discrimination rules are required such as the separation of classes (i.e., variance between classes is maximal) and the homogeneity of classes (i.e., local variance is minimal). This problem is one of the most important in color data analysis in that there is no order relation between colors and the visual perception which is very sensitive to color differences [23].

Components segmented by a simple connected components method [24a] are regions obtained by aggregation of directly connected neighbor pixels. Therefore, each component is only composed of pixels which are connected to each other. Our region growing technique appears as a more general segmentation technique. Because of the threshold ρ leading to the consideration of pixels within a certain distance, a homogeneous region can comprise both connected and unconnected close pixels which are neighboring by their distance from/to each other and which share the same criterion of color homogeneity. In fact, such a homogeneous region is composed of separate close connected components in the sense of a simple connected component method.

The advantage of our method is that it enables us to take account of the neighborhood between the connected components directly during the segmentation process and to avoid other passages to link these components. Moreover, our method enables us to extract homogeneous regions from an arbitrary cloud of points which are not necessarily linked, as this is often the case for textured images.

In our current implementation, the color homogeneity is based on the RGB space, which has its well-known drawbacks. The RGB space is a color representation sensible to light effects; for instance, two totally different colors may be considered more similar than two different shades of an identical color [24b].

However, our technique is general. The proposed algorithm operates in RGB color space and it can be naturally extended to other color spaces, as L*a*b* or L*u*v* uniform color spaces [25]. Nevertheless this implies computing new threshold values for these color spaces to perform homogeneous region segmentation and in choosing an appropriate metrics.

The use of spatial signatures enables the search of images having the same spatial arrangement of the objects as compared to the query image. In this kind of search, one is likely more interested by the layout of the contents captured by an image rather than the pure content itself. In addition, it is expected that two images have almost the same contents if they have a maximum of similar dominant objects in common, without taking account of the spatial arrangement of the objects. In the case where some objects move within an image, the new image will be searched out close to the original query image, only using the global color histogram comparison. The new image would be considered a closer one if it also contains almost the same dominant objects with respect to the original query image and still much closer if the image with moving objects follows the same spatial arrangement of the objects within the query image. One can still refine more the searching result by also taking account of object sizes as well as their positions.

Thus, the use of a spatial signature makes it possible to refine queries with more precision or to specify new types of requests which, otherwise, would be impossible, for example, finding out all the images in which one has a large red object on a small green object. It can be used directly as a signature of similarity from which one can obtain the close images. In our application, the spatial signatures are used to filter the answers according to a required degree of accuracy. The searching criteria allowing an increased accuracy are first color histograms, followed by the number of dominant objects, then simple spatial signature, and the spatial signature enriched with object size and position.

Our search engine is designed to handle with heterogeneous image databases. Consequently, features extracted from images may be very different. In order to decrease the computation complexity without losing the accuracy of the search result, we have used several appropriate thresholds to limit the expansion and the growth of characteristic. For starters we consider a region as a homogeneous object only if its size, i.e., the number of pixels, exceeds a threshold. In our current implementation, a homogeneous region is kept after the region growing process if its size represents at least 0.5% of pixels of the entire image. For less than 0.5% of an image, the region is a detail, which is not visually insignificant. On the other hand, in order to have a reasonable size of the SRG capturing the spatial relationships of objects within an image, we have limited the nodes to the first 30 dominant objects. Thus, only the spatial relationships of these objects are memorized for the search engine. The experiments that we have drawn show that this threshold is rather a good compromise between the search accuracy and efficiency consideration.

7. CONCLUSION

In this paper we introduced several techniques which characterize color homogeneous objects and their spatial relationships for an efficient content-based image searching. We presented a region growing technique for efficient color homogeneous objects segmentation

and extended the 2-D string to an accurate description of spatial information and relationships. Several optimizations, including dominant color histogram clustering, have also been proposed to an efficient search engine implementation. The experimental results that we have drawn so far show that our content-based image searching techniques gives a high precision while keeping a very good recall rate.

Our method emphasizes several objectives in order to improve content-based image retrieval, such as:

- automated extraction of localize coherent regions and visual features
- querying by both feature and spatial information
- development of techniques for fast indexing and retrieval.

Currently we have been working on the extension of our search engine to include other visual features such as textures and simple forms. We have also applied our technique for moving objects' tracking in video.

REFERENCES

1. V. N. Gudivada *et al*. (Guest Eds.), Special Issue on Content-Based Image Retrieval Systems, *Computer* **28**(9), 1995.
2. T. Chiuch, Content-based image indexing, *Proc. of the 20th VLDB Conf., Santiago, Chile, 1994*, pp. 582–593.
3. A. D. Narasimhalu (Guest Ed.), Special Issue on Content-Based Retrieval, *ACM Multimedia Systems* **3**(1), 1995.
4. A. Ralescu and R. Jain (Guest Eds.), Special Issue on Advances in Visual Information Management Systems, *J. Intelligent Information Systems* **3**(3), 1994.
5. M. Flickner *et al*., Query by image and video content: The QBIC system, *IEEE Comput*. **28**(9), 1995, 23–32.
6. Alex Pentland *et al*., Photobook: Content-based manipulation of image databases, *Internat. J. Comput. Vision* **18**, 1996, 233–254.
7. Greg Pass *et al*., Comparing images using color coherence vectors, *ACM Multimedia'96, November 1996*.
8. J. R. Smith and S. F. Chang, Visual SEEK: A fully automated content-based image query system, *ACM Multimedia, Nov. 96*.
9. Y. Chahir and L. Chen, Peano key rediscovery for content-based retrieval of images, *SPIE, M.S.A S. II*, pp. 172–181, November 1997.
10. Y. Chahir and L. Chen, Outils d'indexation et de distribution de données multimédias pour le web, *NTICF '98, Autoroutes de l'information 2*, Rouen, France, November 1998. [http:\\ nticf98.insa-rouen.fr]
11. S. K. Chang *et al*. Iconic indexing by 2D Strings, *IEEE Trans. Pattern Anal. Mach. Intell*. **3**, 1987, 413–428.
12. E. Jungert, Extended symbolic projection as a knowledge structure for image database systems, *Fourth BPRA Conference on Pattern Recognition*, pp. 343–351, Springer-Verlag, Berlin/New York, 1988.
13. S. Chang, E. Jungert, and Y. Li, Representation and retrieval of symbolic pictures using generalized 2-D strings, *Proc. SPIE: Visual Commun. Image Process. IV 1989*, pp. 1360–1372.
14. S-Y Lee and F.-J. Hsu, Spatial reasoning and similarity retrieval of images using 2D C String knowledge representation, *Pattern Recogn*. **25**, 1992, 305–318.
15. S. Lee *et al*. Signature files as a spatial filter for iconic image database, *J. Visual Lang. Comput*. **92**, 305–397.
16. Y. Chahir and L. Chen, *Spatialized Visual Features Based Image Retrieval, CATA, Cancun, Mexico, 1999*.
17. A. Guttman, R-trees: A dynamic index structure for spatial searching, in *ACM Proc. Internat. Conf. Manag. Data, June 1984*, pp. 47–57.
18. P. Huang, Indexing pictures by key objects for large-scale image databases, *Pattern Recogn*. **30**, 1997, 1229–1237.
19. V. Gudivada and V. Raghavan, Design and evaluation of algorithms for image retrieval by spatial similarity, *ACM Trans. in Inform. Systems* **13**, 1995, 115–144.
20. Y. Gong *et al*., An image database system with content capturing and fast image indexing abilities, *Proc. Internat. Conf. Multimedia Computing and Systems, Boston, MA, 1994*, 121–130.

21. C. J. van Rijsbergen, *Retrieval Effectiveness, Information Retrieval Experiment in K. S. Jones, MA, 1981*, pp. 32–43.
22. R. Haralik and L. Shapiro, Survey: Image Segmentation Techniques, *Comput. Vision Graphics Image Process.* **85**, 100–132.
23. A. Tremeau and N. Borel, A region growing and merging algorithm to color segmentation, *Pattern Recogn.* 1191–1203, 1997.
24. Arthur R. Weeks, Jr., *Fundamentals of Electronic Image Processing*, SPIE/IEEE Press, New York, (a) pp. 440–445; (b) pp. 1228–1293.
25. H. R. Kang, Color Technology for Electronic Imaging Devices, *SPIE*, Bellingham, WA, 1996.