FRIP: A Region-Based Image Retrieval Tool Using Automatic Image Segmentation and Stepwise Boolean AND Matching

ByoungChul Ko, *Member, IEEE,* and Hyeran Byun

***Abstract*—In this paper, we present our region-based image retrieval tool, finding region in the picture (FRIP), that is able to accommodate, to the extent possible, region scaling, rotation, and translation. Our goal is to develop an effective retrieval system to overcome a few limitations associated with existing systems. To do this, we propose adaptive circular filters used for semantic image segmentation, which are based on both Bayes' theorem and texture distribution of image. In addition, to decrease the computational complexity without losing the accuracy of the search results, we extract optimal feature vectors from segmented regions and apply them to our stepwise Boolean AND matching scheme. The experimental results using real world images show that our system can indeed improve retrieval performance compared to other global property-based or region-of-interest-based image retrieval methods.**

***Index Terms*—Bayes' theorem, Boolean AND matching, content-based image retrieval (CBIR), finding region in the pictures (FRIP).**

I. Introduction

WITH THE explosive growth in both numbers and sizes of imaging technologies, image retrieval has attracted the interests of researchers in the field of digital libraries, image processing, and database systems. Furthermore, in the last few years, rapid improvements in hardware technology have made it possible to process, store and retrieve huge amounts of data in a multimedia format [1]. As a result, content-based image retrieval (CBIR) has been receiving widespread interest during the last decade.

The first version of Content-Based Image Retrieval was text-based retrieval scheme. Since this simple textual annotation for images was often ambiguous and inadequate for image database search, to overcome the difficulties of annotation-based method, a query-by-image [2]–[4] was proposed in the early 1990s. However, because such CBIR systems typically relied on global properties of an image, simple query-by-image may fail in situations where the global features cannot adequately capture local variations in the image. For example, if a user may wish to retrieve images containing only a "red car" regardless of their background, global image-based retrieval methods may retrieve a "red sunset" or "red flower". As a result, many recent CBIR systems have been focusing on region-based approach (QBIC [5], Netra [6], VisualSeek [7], Blobworld [8], IRM [9], and ROI [10], [11]).

The QBIC system [5] by IBM Corporation is a complete framework offering both content-based retrieval for still images and visualization for video. However, this system uses restricted classes of images and its image segmentation is not fully automatic. Netra [6] is a toolbox used for navigating large image collections. It is implemented on the www (world wide web) using Java language and uses color, texture, and shape information to organize and search the database. A distinguishing aspect of this system is its incorporation of a robust automated image-segmentation algorithm which allows an object or region-based search. However, image segmentation requires some parameter tuning and hand trimming of regions [8]. The VisualSEEk [7] is a system used for region extraction and representation based upon color set. Through a process of color set back-projection, the system automatically extracts salient color regions from images and therefore, the system is able to provide both content-based and spatial search capabilities. Although this system retrieves single or multiple regions by their spatial location, the use of additional feature vectors (e.g., scale and shape) are needed for more exacting image retrieval. In the Blobworld [8], each blob (region) in the image is described by two dominant colors, using the centroid for its location and a scatter matrix for its basic shape representation. However, the segmentation process takes over 5 min/image on a 300-MHz Pentium II PC. Integrate region matching (IRM) [9] incorporates the properties of all regions in an image by utilizing a region-matching scheme. This system allows one region to be compared with several regions a corresponding image in order to increase robustness as opposed to segmentation errors. As a result, if different new regions, having similar properties are inserted in target image, the overall image similarity may be decreased. In addition, the region of interest (ROI)-based approaches [10], [11] are reported. ROI divides the image into $n \times n$ nonoverlapping image blocks because the automatic segmentation of an image into object or region is not always reliable. This approach first computes overlapping between the user-defined ROI and individual image blocks, from which the overall similarity between the ROI and

Manuscript received October 26, 2000; revised May 16, 2003. This work was supported through the Brain Neuroinformatics Research Program sponsored by the Korean Ministry of Science and Technology (M1-0107-00-0008). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Alberto Del Bimbo.

B. Ko was with the Department of Computer Science, Yonsei University, Seoul 120-749, Korea. He is now with Samsung Electronics, Suwon 442-600, Korea (e-mail: byoungchul.ko@samsung.com).

H. Byun is with the Department of Computer Science, Yonsei University, Seoul 120-749, Korea (e-mail: hrbyun@aipiri.yonsei.ac.kr).

Digital Object Identifier 10.1109/TMM.2004.840603

the image is calculated by linearly combining these overlapping image blocks. However, this approach is sensitive to large shifts, cropping, and scaling of regions.

In this paper, we present our region-based image retrieval tool, finding region in the pictures (FRIP). Our goal is to develop an effective mechanism to overcome several limitations which are related to existing systems. To do this, we propose a new image-segmentation method using three types of circular filters, which are based on Bayes' theorem and the texture distribution of image. In addition, in order to reduce computational complexity without losing the accuracy of the search results, we extract optimal feature vectors from segmented regions and apply them to the stepwise Boolean AND matching.

This paper is organized as follows. In Section II, we describe the image-segmentation algorithm using three types of adaptive circular filters. In Section III, feature extraction methods are introduced, including the modified radius-based signature (MRS) used for shape description. Section IV is dedicated to a stepwise similarity-matching algorithm employed in our system. FRIP. Section V shows experimental results exploring the performance of our technique using real-world data. Finally, the conclusions and future works to projects are presented in Section VI.

II. Image Segmentation

Image segmentation is one of the first steps used in an image analysis and is one of the oldest issues faced in image processing [1]. It is also very important in applications such as image understanding, and region/object-based image retrieval. Especially, in the case of region-based image retrieval, the performance of retrieval is largely dependent on image segmentation. Therefore, most query-by-region systems focus on the automatic extraction of the visual content of an image to enable indexing and retrieval. Even though image segmentation is useful in many applications, semantic image segmentation is still beyond the reach of current computer vision techniques, due to the uncontrolled nature of the available images [12]. In reality, most natural images are influenced by noise, shadows, surface, and curvature. Some segmentation methods use color coordinating system for segmentation [7], [12]–[14]. However, image segmentation based only on color or gray-level usually fails to identify meaningful objects due to the instability of saturations or intensity changes of color. To overcome these problems, we propose a new image segmentation method which combines CIE L*a*b* color and edge distribution of image for region-based image retrieval.

A. *Two-Level Segmentation Using Adaptive Circular Filter and Bayes' Theorem*

In our new image-segmentation algorithm, the image-segmentation process consists of two levels. At the first level, we segment an image using three types of adaptive circular filters based on the amount of image texture information the image possesses. At the second (iterative) level, small patches of an image can be merged into similar adjacent regions by region merging and region labeling.

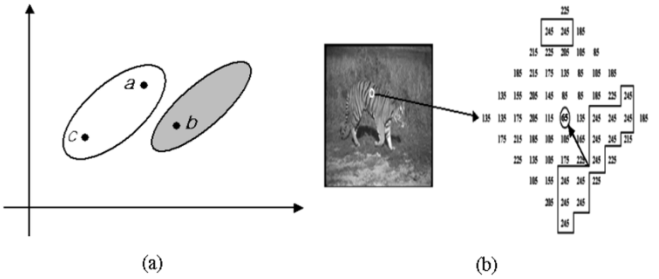


Fig. 1. (a) Advantage of clustering over distance measurement. (b) Example of circular filtering process (L^* color-band).

1) Adaptive Circular Filter: For image segmentation, the perceptually uniform CIE L*a*b* color space is used to represent color features. The L* component in the CIE L*a*b* model denotes the luminance of the color, while components a* and b* together represent the chromatic information. CIE L*a*b* model is generated by linearly transforming the RGB color space to the XYZ color space by a nonlinear transformation [14]

$$L^* = \begin{cases} 116(Y/Y_n)^{1/3} - 16, & \text{for } (Y/Y_n) > 0.008\,856 \\ 903.3(Y/Y_n), & \text{for } (Y/Y_n) \le 0.008\,856 \end{cases} \quad (1)$$

$$a^* = 500\{f(X/X_n) - f(Y/Y_n)\} \quad (2)$$

$$b^* = 200\{f(Y/Y_n) - f(Z/Z_n)\}. \quad (3)$$

In (1)–(3), (X_n, Y_n, Z_n) represents the tristimulus values and function $f(X)$ is presented as follows:

$$f(X) = \begin{cases} X^{1/3}, & \text{for } X > 0.008\,856 \\ 7.787 + 16/116, & \text{for } X \le 0.008\,856. \end{cases} \quad (4)$$

After the color transformation, the L*a*b* images are smoothed to remove noise using a median filter. Then, we apply our designed adaptive circular filters to the resulting image. As explained in [15], similarities based on clustering are in many cases far more natural than similarities based on distances alone. In the circular filter, we use this characteristic. As shown in Fig. 1(a), although point a is closer (in distance) to b than it is to c, a is more similar to c than it is to b. In the same way, although a center pixel of circular filter has different color distance with neighborhood region [Fig. 1(b)], it is clustered into neighborhood region if it satisfies our defined constraint (5). By employing our circular filter, small senseless textures (e.g., strips or spots) can be clustered into semantic neighborhood regions regardless of their similarity distances. Basically, a circular shape is appropriate to retain the natural or artificial object's shape, such as human, animal, tree or car. Fig. 1(b) shows an example of a circular filter with an overall size of 11×11 pixels.

We first calculate the color histogram in the circular filter area except for the center point. The color histogram consists of 256 bins. Here, we assume two classes. The first class (C_M) corresponds to the most frequently observed histogram bins, and the second class $(C_{\bar{M}})$ is the remaining histogram bins. The symbol M_N represents the number of bins of class C_M and $\bar{M}_N$ represents the number of bins of class $C_{\bar{M}}$. From the distribution of this histogram, we can easily estimate

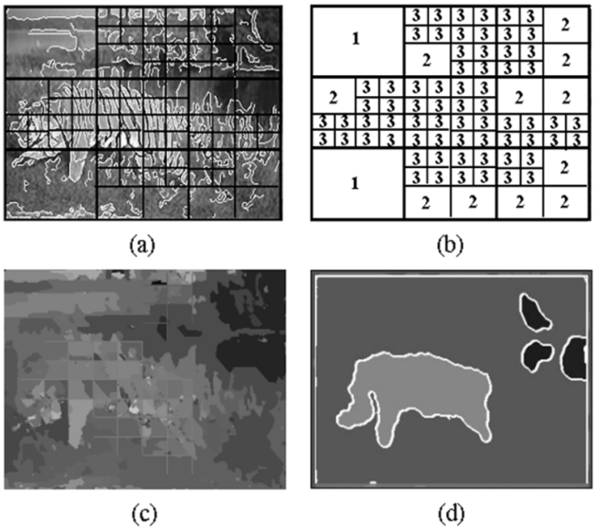


Fig. 2. Image segmentation using adaptive circular filters. (a) Image division according to the edge distribution. (b) Selected filter ("1"-7 × 7 filter, "2"-11 × 11 filter, "3"-15 × 15 filter). (c) Segmentation result by the first level segmentation. (d) Final segmented image.

the conditional probabilities $P(c_{x,y} \,|\, C_M)$ and $P(c_{x,y} \,|\, C_{\bar{M}})$. The expression $P(c_{x,y} \,|\, C_M)$ is the conditional probability of obtaining the center value $c_{x,y}$ given that it is computed from the class C_M. Similarly, $P(c_{x,y} \,|\, C_{\bar{M}})$ is the conditional probability of obtaining the center value $c_{x,y}$ given that it is computed from the class $C_{\bar{M}}$. If N is the total number of pixels included in the circular filter and $M_N, \bar{M}_N$ of them belong to C_M and $C_{\bar{M}}$, respectively, then $P(c_{x,y} \,|\, C_M) \cong M_N/N$ and $P(c_{x,y} \,|\, C_{\bar{M}}) \cong \bar{M}_N/N$. We chose the prior probabilities $P(C_M)$ to be 0.8 and $P(C_{\bar{M}})$ to be 0.2, which produced the best results during experimentation. Then, we can calculate the probability of the center point being in class C_M by using Bayes' theorem and if $P(C_M \,|\, c_{x,y})$ is over 0.5, then the center point is merged into the major class color (M_C)

$$P(C_M \,|\, c_{\mathrm{xy}}) = \frac{P(C_M)P(c_{x,y} \,|\, C_M)}{P(C_M)P(c_{x,y} \,|\, C_M) + P(C_{\bar{M}})P(c_{x,y} \,|\, C_{\bar{M}})} \tag{5}$$

$$\text{if } P(C_M \,|\, c_{x,y}) \geq 0.5, \quad \begin{array}{ll} \text{then} & c_{x,y} = M_C \\ \text{else} & c_{x,y} = c_{x,y} \end{array}.$$

Our circular filters have three different sizes: 15 × 15, 11 × 11, and 7 × 7 pixels. We determine the size of filters by experimentations: if the size of a filter is bigger than 15 × 15, images are under-segmented, and if the size of a filter is smaller than 7 × 7, images are over-segmented. Our experiments show that the 15 × 15 filter is sufficient to filter out texture information necessary to produce the desired results. If the size of an image is twice as large as that (320 × 210) of images in our database, each filter will be doubled in size as well. Also, if the size of an image is reduced by half, so is each filter.

To apply the circular filters, we first examine texture distributions [Fig. 2(a)]. Here, we use the Canny edge detector in order to estimate the edge distribution. After that, an image is divided into nine scale1 subblocks each of the same size. For each scale1 subblock, if the sum of edge pixels is over 20% of the sum of overall pixels included in the same subblock, then subblock is divided again into four scale2 subblocks of the same size. This process is repeated until the subblock size is less than the biggest filter size (normally scale3) [Fig. 2(b)]. Then, each filter is applied to the subblocks according to the scale number. Since scale3 has the largest edge distribution, a 15 × 15 filter is applied. On the other hand, since scale1 has the smallest edge distribution, 7 × 7 filter is applied [Fig. 2(c)]. Finally, we use a region merge algorithm to merge the small patches (R) into adjacent regions that share similar color properties based on the mean of the region, R^i_m and homogeneity function (6). The algorithm is described as follows:

$$\sum_{i}^{N} \left(\left| R_{L*} - R^i_{mL*} \right| + \left| R_{a*} - R^i_{ma*} \right| + \left| R_{b*} - R^i_{mb*} \right| \right) \leq T \tag{6}$$

where N is the number of neighborhood regions and T is the minimum threshold for region merging. The results of the first-level segmentation are shown in Fig. 2(c). By observing the first level segmentation, we can initialize similar regions that are not merged by normal region merge algorithm, according to their texture distribution and erode the senseless small regions to the background.

B. Iterative Level Using Region Labeling and Iterative Region Merging

After the first-level segmentation, some regions may be declared to be similar if they have similar color properties, even though they are in fact semantically different regions. Therefore, we label each region as different regions using the connected-component algorithm to protect different regions from merging into the same region. Each region that has the same color value is projected to a buffer as a specific color value. This binary image is raster scanned from left to right and from top to bottom. In this step, different region numbers are assigned to each region.

After region labeling, if the number of regions is over 30, we repeat the circular filtering starting with the smallest filter (7 × 7) in order to merge small patches into adjacent regions that are not merged at the first level, and then perform region merging with a threshold T. The threshold T increases as iterations continue until the number of regions is reduced to below 30. Furthermore, in our experiments, a homogeneous region is also merged into the biggest neighborhood region if its size (number of pixels) is less than 0.3% of the image. Fig. 2(d) shows a final segmented image. Here, to save feature descriptions storage space for our region-based image retrieval application, we restrict the maximum number of regions to 30. Fig. 3 shows some color segmentation results produced by our algorithm.

III. Feature Extraction

From a segmented image, we need to extract features from each region. During this step, features should be described briefly and effectively. Although detailed feature information is better than coarse information for retrieval accuracy, its performance is known to degrade as the number of dimensions

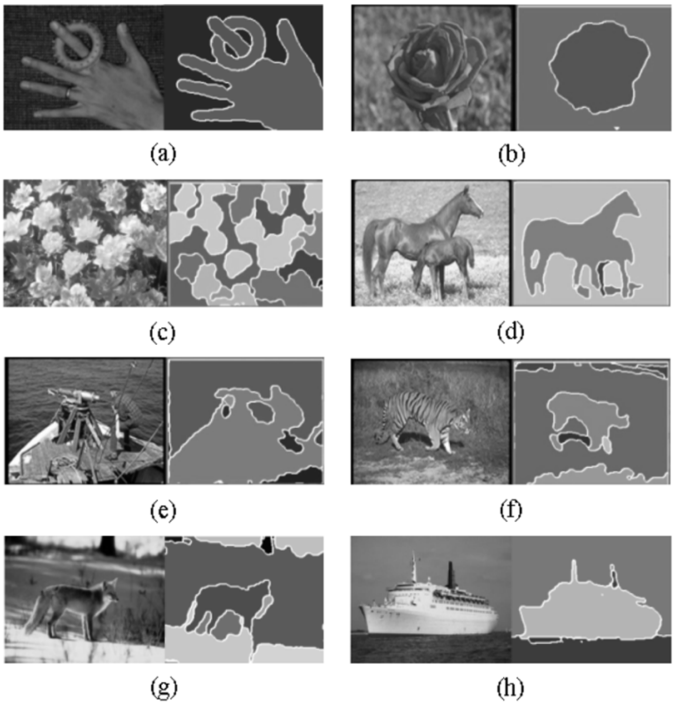


Fig. 3. Some color segmentation results. Left: original image. Right: its corresponding segmentation. (a) Finger. (b) Flower. (c) Garden. (d) Horse. (e) Fishing boat. (f) Tiger. (g) Fox. (h) Ship.

increase (*the curse of dimensionality*). Therefore, in this system, we extract five concise and precise features. In this system, we extract color, texture, NArea (scale), shape, and location features. At this time, since the different features may be defined along intervals of widely varying values, they are normalized to lie between a common interval. Five feature vectors are normalized to [0–1] using the Gaussian normalization method [16] as soon as features are extracted from segmented regions before they are used for distance estimation.

A. Extracting Color Features

We extract the average (Al, Aa, Ab) and variance (Vl, Va, Vb) of the CIE-L*a*b* color space associated with each region instead of color histogram in order to reduce storage space. The color distance $(d_{Q,T}^{C})$ between query (Q) and target regions (T) is measured by the following city-block distance:

$$d_{Q,T}^{C} = \left(\sum_{C=Al,Aa,Ab} |Q_C - T_C| + \sum_{C=Vl,Va,Vb} |Q_C - T_C| \right) \Big/ 2. \quad (7)$$

B. Extracting Texture Features

We choose the biorthogonal wavelet frame (BWF) [17] as the texture feature. Through the BWF, we can obtain a fast and precise directional feature compared with multi-resolution method and get the same size of low-pass and high-pass images from the original image. Each high-pass image is decomposed again into X-Y directional subimages. From the first-level high-pass filtered images, we calculate the X-Y directional amplitude (Xd, Yd) of each region. The distance in texture $(d_{Q,T}^{T})$ between the subsequent two regions, Q and T, is computed by the following city-block distance:

$$d_{Q,T}^{T} = \left| \frac{Yd_Q}{Xd_Q} - \frac{Yd_T}{Xd_T} \right|. \quad (8)$$

C. Extracting NArea (Normalized Area) Feature

The featured NArea is defined as the number of pixels (NP) of a region divided by the image size. The distance in NArea $(d_{Q,T}^{\mathrm{NArea}})$ between query (Q) and target region (T) is computed by using the following city-block distance:

$$d_{Q,T}^{\mathrm{NArea}} = |\mathrm{NP}_Q - \mathrm{NP}_T|. \quad (9)$$

D. Extracting Shape and Location Feature

In our FRIP system, we use two kinds of shape features. For the global geometric shape feature, we use eccentricity first. While for the local geometric feature, we propose the MRS (modified radius-based shape signature) to be invariant under shape's scaling, rotation, and translation. This MRS is only applied to shapes that satisfy eccentricity during the first step in Boolean AND matching. In order to calculate eccentricity, we first estimate the bounding rectangle for each segmented region. From the major axis $(R_{\max})$ and the minor axis $(R_{\min})$, we can obtain eccentricity at the similarity comparison step. Eccentricity (E) is defined as the ratio of $R_{\max}$ to $R_{\min}$ between query (Q) and target region (T)

$$E = \left| \frac{R_{Q\min}}{R_{Q\max}} - \frac{R_{T\min}}{R_{T\max}} \right|. \quad (10)$$

After the calculation of eccentricity, the centroid is computed from each region. The location of each region is defined by its centroid (C_x, C_y). Here, spatial location distance $(d_{Q,T}^{P})$ between two regions is measured by the following Euclidean distance:

$$d_{Q,T}^{P} = \sqrt{(Cx_Q - Cx_T)^2 + (Cy_Q - Cy_T)^2}. \quad (11)$$

To estimate MRS, the boundary of a segmented region is extracted using Sobel edge detection and edge thinning algorithms. After boundary extraction, a shape is represented by a feature function called a signature. Original signatures are invariant to translation, but they are sensitive to rotation and scaling. In our FRIP system, we propose the MRS in order to make it invariant to scaling and rotation as well as translation. First, to achieve rotation invariance, we calculate the orientation of a region. Orientation is the angle between the major axis of the object and X-axis. From a determined orientation, we create a central axis that passes through the centroid having a slope

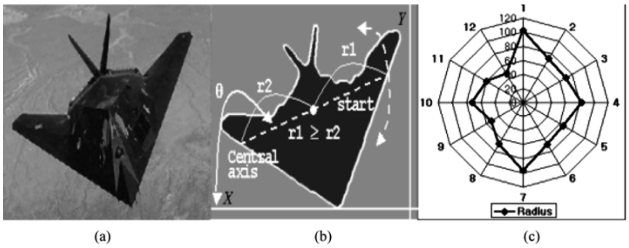


Fig. 4. Region signature. (a) Original image. (b) Region boundary. (c) Polygon by MRS.

of θ degrees. We estimate the starting point as the most distant point from the centroid to two symmetric boundary points aligned with the central axis. With this method, we can maintain the same starting point, even though the shape is somewhat distorted by its movement or occlusion (Fig. 4).

To save storage space, we obtain and store only one MRS which is estimated from the starting point and in the clockwise direction. However, due to the fact that two MRSs exist in two directions from the starting point, we have to consider two different directional (flip case) MRSs (clockwise and counterclockwise). To extract feature points, the original radius-based signature extracts N radius distance values per constant degrees. However, shape is not always simple and if the centroid exists outside of the shape, the first method gives incorrect distance signatures. Chang *et al.* [18] constructed the distance function from the centroid to the high curvature points. Even though this method produces more reasonable representation on shape regardless of the location of the centroid, if it may have different number of feature points, it does not correspond with the purpose of our system which reduces the feature dimension. The MRS extracts 12 radius (N) distance values from uniformly sampled boundary points to the centroid and add these values to the region index. In this method, even though the centroid exists outside of a region, the signature has a similar configuration with original region.

To achieve scale invariance, we use a combination of MRS and (12). Equation (12) represents the differential variance between the average MRS ratio of the query region (Q_i) to a region in the database (T_i) and individual MRS ratio

$$d_C = \frac{1}{N}\sum_{i=1}^{N}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\left|\frac{Q_i}{T_i}\right|\right) - \left|\frac{Q_i}{T_i}\right|\right]^2 \tag{12}$$

$$(d_1 : \text{clockwise}, d_2 : \text{counterclockwise})$$

$$d_{Q,T}^{\text{MRS}} = \min(d_1, d_2). \tag{13}$$

For example, if a database region and scale-up (or down) query region have similar shapes, these two shape distances (d_1, d_2) are to be approximately zero and the smaller one can be chosen as the real shape distance $(d_{Q,T}^{\text{MRS}})$ between the query (Q) and the target MRS (T) by (13) regardless of its rotation or translation. By using MRS and (13), our system is more robust against at measuring distortion, rotation, and scale changes of a region because they provide local shape information guided by global properties.

Finally, we save 27 statistical properties of each region as an index in the database. All properties are shown as follows:

```
Image
keys: imageNo
{
attribute int RegionNo;
attribute int AverRed(Ar);
attribute int AverGreen(Ag);
attribute int AverBlue(Ab);
attribute float VarRed(Vr);
attribute float VarGreen(Vg);
attribute float VarBlue(Vb);
attribute int Normalized Area (NArea);
attribute int CenterOfX(Cx);
attribute int CenterOfY(Cy);
attribute int MajorLength(Rmax);
attribute int MinorLength(Rmin);
attribute Array⟨int⟩ Signature [12];
attribute float AmplOfXDirect(Xd);
attribute float AmplOfYDirect(Yd);
attribute float AmplOfXYDirect(XYd);
}
```

IV. Stepwise Similarity Matching

In general, a linear search is adequate for obtaining "real-time" results in small database. Furthermore, the cost of indexing multidimensional data is exponential in the dimension of the data, and if the dimensions beyond the range of 12–15, it is more efficient to do a linear search [19]. However, in the case of region-based image retrieval, the similarity of each query region should be calculated against all the regions extracted from the images in the database. As the number of regions increases in the database, so does the number of comparisons required to compute similar retrievals. In general, because such sequential matching is computationally expensive, a two-pass approach is often adopted [19]–[22] in the tradeoff effectiveness and efficiency. In this case, first, it retrieves a set of N images using an inexpensive search algorithm (*quick-and-dirty*) and next, a more sophisticated matching technique is used. Therefore, the choice of N is important here. To improve retrieval performance and reduce the total retrieval time, we use the AND operator with independent color and shape features as a first step in our stepwise matching. Once we find a sufficient number of candidate images, we apply a linear combination of feature distances to the retrieved set by using AND operator for similarity computation.

In the FRIP system, the actual matching process is to search for the k elements in the stored region set closest to the query region. After regions are segmented, the user selects one or multiple regions that he/she wants to search. Finally, by the user-specified constraints, such as: color-care/do not care scale (NArea)-care/do not care, shape-care/do not care, and location-care/do not care, the overall matching score is calculated according to (14).

A. Stepwise Boolean AND Matching for Search Pruning

In the FRIP system, we use five features: color, texture, shape (shape1: eccentricity, shape2: MRS), and location. Given

TABLE I
PRECISION VALUES ON RANDOMLY SELECTED POSSIBLE FEATURE PAIRS

Feature pairs	Precision (%)
Color, texture	70
Color, scale	70
Color, shape1	80
Shape1, scale	75
Shape1, location	30
Scale, location	40

a query region, the following steps are taken to calculate the final rank according to user constraints.

1) During the first step, check the number of features in the query regions. In the case of both color and shape-care, we calculate color distance (7) and eccentricity (10) between each region included in a database image and the query region. Regions whose color and eccentricity distances to the query are below given thresholds are selected as candidate regions for subsequent distance calculation. If the user chooses one feature between two features, only one distance is calculated. In this system, we set thresholds (a threshold for color: 0.25, a threshold for eccentricity: 0.35) to retrieve an average of 10%–20% of database regions as candidate regions through the experimentations.

   Notice that this step allows us to significantly reduce the number of evaluations required for feature matching. For example, suppose that our database has 1000 images and there are 30 regions per image. In general, 5(features) × 30(regions) times per one image are needed which results in 150 000 evaluations for the entire database. However, our method requires only 60 (2×30) evaluations per image and at most 60 000 evaluations overall.

   In order to select the features to be used during the first step, we pick two features from the feature set $\{$color, texture, scale, shape1, location$\}$. To do this, we randomly selected possible feature pairs ($\{$color, texture$\}$, $\{$color, scale$\}$, $\{$color, shape1$\}$, $\{$shape1, scale$\}$, $\{$shape1, location$\}$, $\{$scale, location$\}$) and test them against the Sun region as a query because it preserves a consistent shape, color, texture, NArea, and location. Because shape2 has high dimensionality (12), we do not consider it as a first step feature in order to reduce the complexity. Table I shows the results of the experiments on six possible feature pairs. Since the average precision on $\{$color, $\mathrm{shape1}\}$ is 80% more effective when hbox-compared to others, we adopted two features, color and shape1 as the features used for Boolean AND matching.
2) As determined by the users, either all or a few distances are calculated using (8), (9), (11), and (13).
3) From the calculated distances, the final distance is estimated by (14) and the top k nearest images are displayed in ascending order of the final distance

$$\mathrm{Sim}(X,Y) = \sum_{i=1}^{r} \left(\sum_{j=1}^{p} w_j \cdot D_j \left(q_j^i, t_j \right) \right) \tag{14}$$

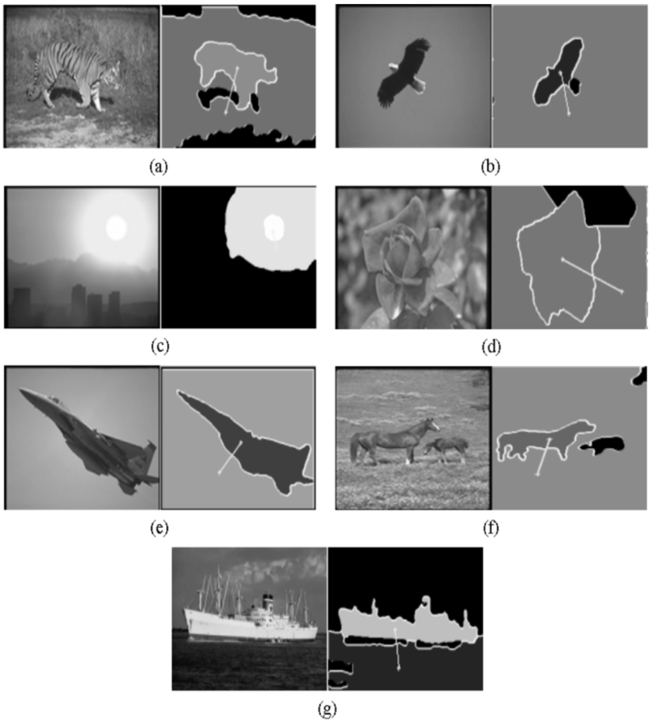

(a) (b) (c) (d) (e) (f) (g)

Fig. 5. Query examples for multiple regions (left: original image, right: query regions). (a) Query #1: tiger with green field. (b) Query #2: eagle with blue sky. (c) Query #3: sun with yellow sunset. (d) Query #4: flower with green leaves. (e) Query #5: airplane with blue sky. (f) Query #6: horse with green field. (g) Query #7: ship with blue sea.

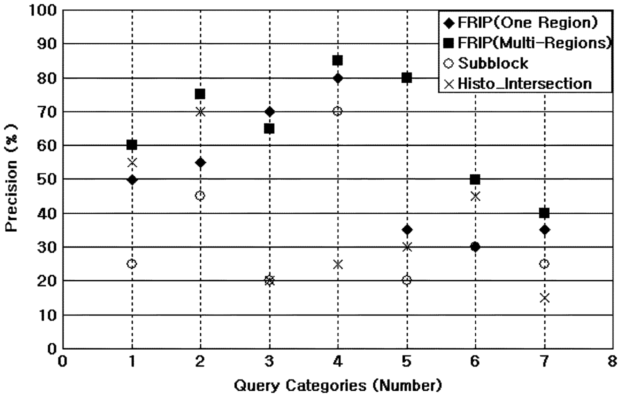
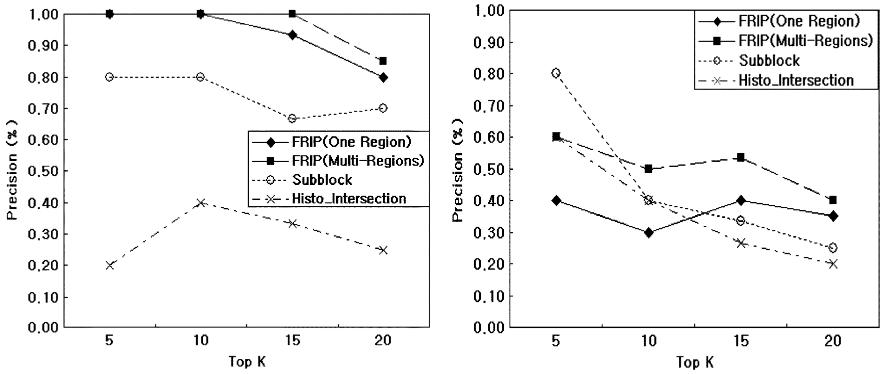


Fig. 6. Performance comparison of seven queries by precision among FRIP using one region, multiple regions, Subblock, and Histogram intersection method. The average precision of each method is about 50.7%, 65%, 33.6%, and 37.1%, respectively.

where w_j is the weight associated with the ith feature and it is initialized to $1/p$. The symbol w_j is adjustable by the operator's decision. The symbol D_j represents a metric distance between the ith feature of query and that of an image in the database. The symbol r represents the number of regions for multiple regions query. In this system, we consider only absolute location of regions. Relative location of regions and their spatial relationships will be added in the next version of our system.

V. EXPERIMENTAL RESULTS AND EVALUATION

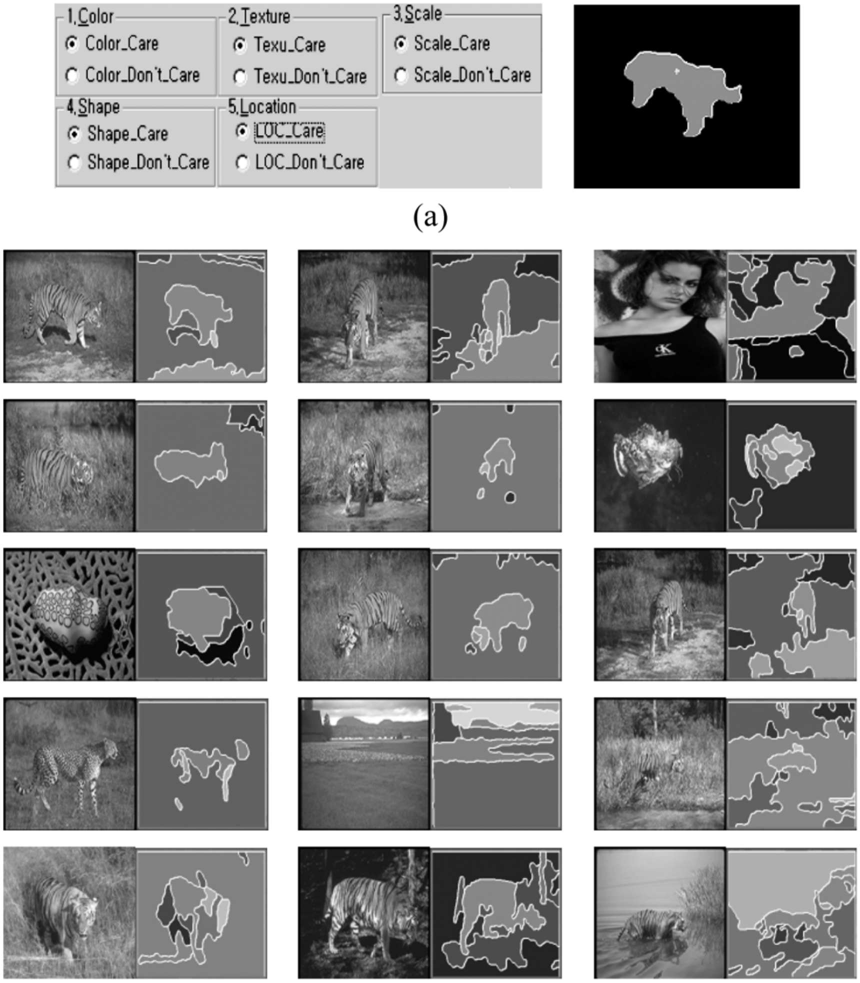
This system is developed in Visual C++ 6.0 language as an off-line system. The average segmentation time requires approximately 20 s per image and the average retrieval time



(a)

(b)

Fig. 7. (a) Variation of precision according to the top K (query: flower, best case). (b) Variation of precision according to the top K (query: ship, worst case).



1.Color
● Color_Care
○ Color_Don't_Care

2.Texture
● Texu_Care
○ Texu_Don't_Care

3.Scale
● Scale_Care
○ Scale_Don't_Care

4.Shape
● Shape_Care
○ Shape_Don't_Care

5.Location
● LOC_Care
○ LOC_Don't_Care

(a)

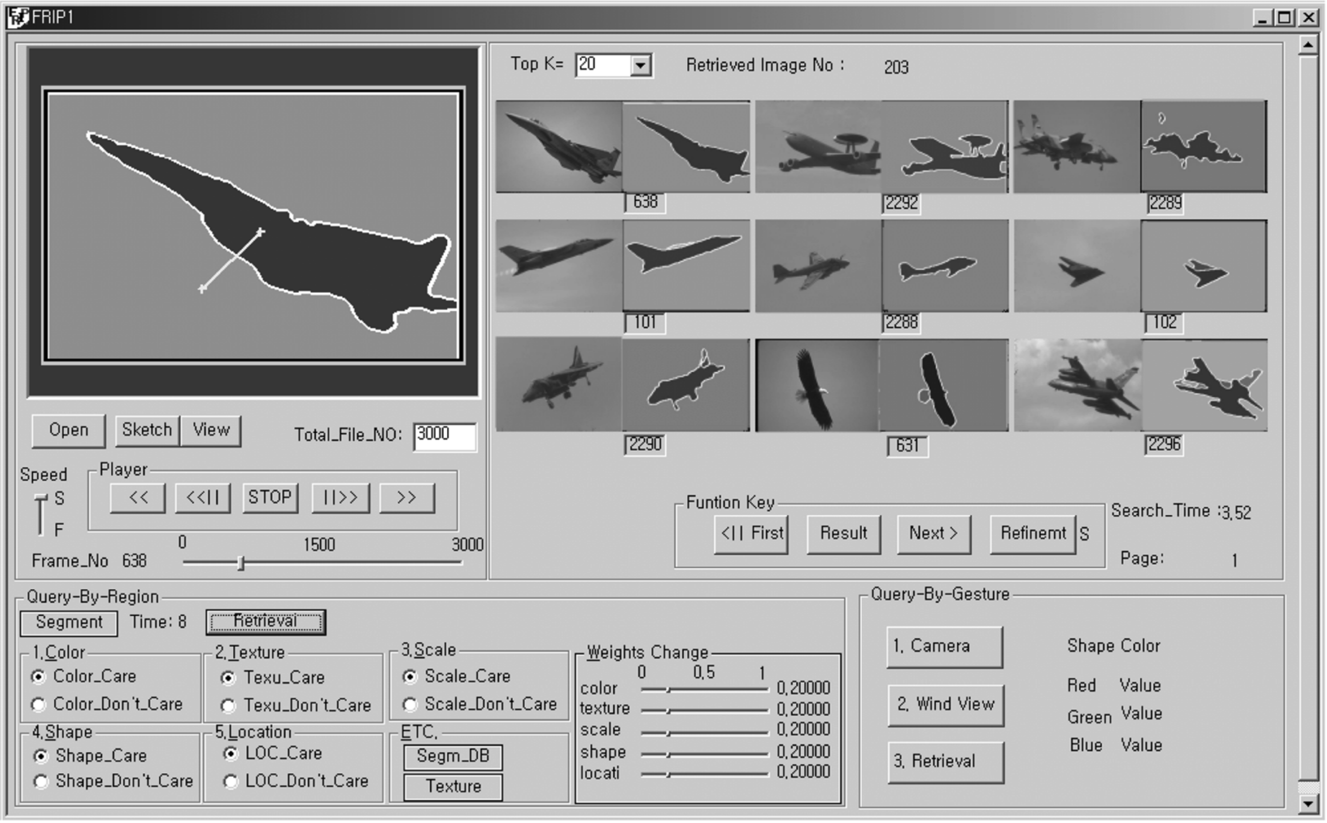
(b)

Fig. 8. Image search for "tiger" region. (a) Query constraints which are formed by the user and a selected query region. (b) Top 15 retrievals from database.

requires 3.4 s on 3000 images using a Pentium PC, 866 MHz.[1] We have performed experiments using a set of 3000 images from the Corel-photo CD, covering a wide variety of content ranging from natural images to graphic images without preselected categories.

To complete a query, the user pushes the segmentation button and selects one or several regions. After that, the user selects the number of retrieval images k and clicks retrieval button that he/she wants to retrieve. A user can choose color, texture, NArea (scale), shape, and location constraints in order to search regions more precisely. In this experiment, we choose all user constraints and the top 20 nearest neighbors are returned. In general, experimental verification of an image retrieval system is a very

[1] The retrieval results are accessible at http://cvpr.yonsei.ac.kr/Frip.



FRIP1

Top K= 20 Retrieved Image No : 203

Total_File_NO: 3000

Frame_No 638

Search_Time :3.52

Page: 1

Query-By-Region: Time: 8

1.Color: ● Color_Care ○ Color_Don't_Care

2.Texture: ● Texu_Care ○ Texu_Don't_Care

3.Scale: ● Scale_Care ○ Scale_Don't_Care

4.Shape: ● Shape_Care ○ Shape_Don't_Care

5.Location: ● LOC_Care ○ LOC_Don't_Care

Weights Change: color 0.20000, texture 0.20000, scale 0.20000, shape 0.20000, locati 0.20000

Query-By-Gesture: Shape Color — Red Value, Green Value, Blue Value

Fig. 9. Retrieval results of query "airplane with blue sky" (left: original image, right: segmented image).

difficult task. There are few if any standard datasets, and there are no widely "agreed-upon" evaluation metrics [23]. To validate the effectiveness of our approach, we compare the retrieval performance of our system with the Flexible-Subblock algorithm [24], which is similar to the ROI method and histogram intersection method [25]. The test is performed on seven specific classes, using a query function such as: 1) tiger; 2) eagle; 3) sun; 4) flower; 5) airplane; 6) horse; and 7) ship. We also carried out additional evaluation on multiple regions query. For a multiple regions query, we select the same query region with one or more other regions. That is: 1) a tiger with a green field; 2) an eagle with a blue sky; 3) a sun with a yellow sunset; 4) a flower with green leaves; 5) airplane with blue sky; 6) a horse with green field; and 7) a ship with blue sea. Fig. 5 shows the seven query regions selected for testing.

Fig. 6 shows the retrieval performance between the FRIP with one region, the FRIP with multiple regions, the flexible-subblock and histogram intersection. In all experiments, performance is measured using average retrieval precision. As shown in Fig. 6, the overall performance of our two approaches outperforms the other two methods as by percentages of 50.7%, 65%, 33.6%, and 37.1%. In the case of query number 3, even though it has additional yellow sunset information, many false positives are retrieved because other sunset images included in our database have a different color sunset (e.g., red). Therefore, a query for only one region produces better performance than a query for multiregions.

Fig. 7 indicates that the variation of precision according to the top K. Fig. 7(a) shows the precision of the best case (query-flower) and Fig. 7(b) shows the precision of the worst case (query-ship). Even though the ship has almost the same color and texture properties except for shape, it shows significantly reduced performance than the flexible-subblock and histogram intersection method within the top ten.

Figs. 8 and 9 shows some retrieval results of the FRIP system.

VI. Conclusion and Future Works

This paper presents a region-based image retrieval system and a FRIP that is able to accomodate, to the extent possible, region scaling, rotation, and translation. For semantic region segmentation, we proposed three types of circular filters, which are based on both Bayes' theorem and texture distribution. In addition, in order to decrease the computation complexity without losing the accuracy of the search results, we have extracted optimal feature vectors from segmented regions and apply them to the stepwise Boolean AND matching. The experimental results using real image data show that our FRIP system can indeed improve retrieval performance when compared to global property-based or ROI-based method.

In general, a region-based image retrieval (RBIR) usually places high premium on segmentation quality, which is often difficult to achieve, in practice, due to several factors such as varying environmental conditions and occlusion. Thus, a RBIR provides challenges as well as opportunities for developing effective interactive mechanisms that use imperfect information to conduct visual query. Currently, we have been working on the extension of our search engine to include a relevance feedback learning algorithm in order to overcome some of the limitations associated with RBIR systems such as imperfect image segmentation, the weight adjustment problem, and an incorrect feature vector due to the distortion of region. Our new relevance feedback algorithm is an advanced version of our previous research [26] and it is designed to support four levels of relevance feedback [27] such as highly relevant, relevant, irrelevant and highly irrelevant. Through our additional approach, the system can capture the user's intension more precisely and improve performance for future iterations by recording all users' feedback actions. In addition, we have been upgrading our multiple regions query to consider not only absolute position but also relative position of regions with their spatial relationships [28]. We currently continue to conduct research on the above-mentioned mechanisms and plan to report our results in a forthcoming paper.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments in improving the quality of this work.

REFERENCES

[1] A. Moghaddamzadeh and N. Bourbakis, "A fuzzy region growing approach for segmentation of color images," *Pattern Recognit.*, vol. 30, no. 6, pp. 867–881, 1997.

[2] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.

[3] M. Stricker and M. Orengo, "Similarity of color images," *SPIE: Storage Retrieval Image and Video Database III*, vol. 2420, pp. 381–392, Feb. 1995.

[4] Y. Gong, H. Zhang, H. Chuant, and M. Skauuchi, "An image database system with content capturing and fast image indexing abilities," in *Proc. Int. Conf. Multimedia Computing and Systems*, May 1994, pp. 121–130.

[5] M. Flickner, H. Sawhney, W. Niblack, and J. Ashley, "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, no. 9, pp. 23–32, Sep. 1995.

[6] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image database," in *Proc. Int. Conf. Image Processing*, vol. 1, 1997, pp. 568–571.

[7] J. R. Smith and S. F. Chang, "VisualSEEk: A fully automated content-based image query system," *ACM Multimedia*, pp. 87–98, 1996.

[8] M. Carson, S. Thomas, J. M. Belongie, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *Proc. Int. Conf. Visual Information Systems*, 1999, pp. 509–516.

[9] J. Li, J. Z. Wang, and G. Wiederhold, "IRM: Integrated region matching for image retrieval," *ACM Multimedia*, pp. 147–156, 2000.

[10] B. Moghaddam, H. Biermann, and D. Margaritis, "Defining image content with multiple regions-of-interest," in *Proc. IEEE Workshop on CBAIVL*, 1999, pp. 89–93.

[11] Q. Tian, Y. Wu, and S. Thomas, "Combine user defined region-of-interest and spatial layout for image retrieval," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2000, pp. 746–749.

[12] Y. Deng, B. S. Manjunath, and H. Shin, "Color image segmentation," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, vol. 2, 1999, pp. 447–451.

[13] N. Ikonomakis, K. N. Pataniotis, and A. N. Venetsanopoulos, "Unsupervised seed determination for a region-based color image segmentation scheme," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 2000, pp. 537–540.

[14] A. H. Kan and W. J. Fitzgerald, "A general method for unsupervised segmentation of images using a multiscale approach," in *Proc. 6th European Conf. Computer Vision, Part II*, Dublin, Ireland, Jul. 2000.

[15] E. J. Pauwels and G. Frederix, "Finding salient regions in images," *Comput. Vis. Image Understanding*, vol. 75, pp. 73–85, 1999.

[16] Y. Rui, S. Thomas, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 644–655, Sep. 1998.

[17] S. Burrus, R. A. Gopinath, and H. Guo, *Introduction Wavelets and Wavelet Transforms*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[18] C. C. Chang, S. M. Hwang, and D. J. Buehrer, "A shape recognition scheme based on relative distances of feature points from the centroid," *Pattern Recognit.*, vol. 24, no. 11, pp. 1053–1063, 1991.

[19] H. Sundaram and S. F. Chang, "Efficient video sequence retrieval in large repositories," in *Proc. SPIE Storage and Retrieval for Image and Video Database VII*, vol. 3656, 1999, pp. 108–119.

[20] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Image indexing using color correlogram," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, May 1997, pp. 369–380.

[21] K. A. Hua, K. Vu, and J. H. Oh, "SamMatch, a flexible and efficient sampling-based image retrieval technique for large image databases," *ACM Multimedia*, pp. 225–234, 1999.

[22] J. Hafner *et al.*, "Efficient color histogram indexing for quadratic form distance function," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 7, pp. 729–736, Jul. 1995.

[23] K. Tieu and P. Viola, "Boosting image retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, 2000, pp. 228–235.

[24] B. C. Ko, H. S. Lee, and H. Byun, "Flexible subblock for visual information retrieval," *Electron. Lett.*, vol. 36, no. 1, pp. 24–25, Jan. 2000.

[25] M. J. Swain, "Interactive indexing into image database," in *Proc. SPIE: Storage and Retrieval Image and Video Database*, Feb. 1993, pp. 95–103.

[26] B. C. Ko, J. Peng, and H. Byun, "Region-based image retrieval using probabilistic feature relevance feedback," *Pattern Anal. Applic. (PAA)*, vol. 4, pp. 174–184, 2001.

[27] B. C. Ko and H. Byun, "Probabilistic neural networks supporting multi-class relevance feedback in region-based image retrieval," in *Proc. Int. Conf. Pattern Recognition*, vol. IV, Aug. 2002, pp. 138–141.

[28] ——, "Multiple regions and their spatial relationsgip-based image retrieval," in *Proc. Int. Conf. Image and Video Retrieval*, vol. 2383, Jul. 2002, pp. 81–90.

ByoungChul Ko (M'00) received the B.S. degree in computer science from Kyonggi University, Seoul, Korea, in 1998, and the M.S. and Ph.D. degrees in computer science in 2000 and 2004, respectively, from Yonsei University, Seoul.

He is currently a Research Engineer at Samsung Electronics, Suwon, Korea. His research interests include multimedia information retrieval, computer vision, face detection and tracking, and pattern recognition.

Mr. Ko received the Excellent Paper Award in 2000 from the Korea Information Science Society (KISS).

Hyeran Byun received the B.S. and M.S. degrees in mathematics from Yonsei University, Seoul, Korea, and the Ph.D. degree in computer science from Purdue University, West Lafayette, IN.

She was an Assistant Professor at Hallym University, Chooncheon, Korea, from 1994 to 1995. Since 1995, she has been an Associate Professor of Computer Science at Yonsei University. Her research interests include multimedia, computer vision, image, and video processing, artificial intelligence, and pattern recognition.