Region-based Image Retrieval Using Probabilistic Feature Relevance Learning

ByoungChul Ko*†, Jing Peng**, Hyeran Byun*

*VIP. Lab. Dept. of Computer Science, Yonsei University
Shinchon-Dong, Seodaemun-Gu, 120-749, Korea
Email: {soccer1, hrbyun}@aipiri.yonsei.ac.kr

**Dept. of Computer Science, Oklahoma State University
Stillwater, OK, 74078, USA
Email: jpeng@cs.okstate.edu

† Corresponding author

Abstract

Region-based image retrieval (RBIR), a specialization of content-based image retrieval, is a promising and important research area. RBIR usually requires good segmentation, which is often difficult to achieve in practice due to several factors such as varying environmental conditions and occlusion. It is, therefore, imperative to develop effective mechanisms for interactive region-based visual query in order to provide confident retrieval performance. In this paper, we present a novel RBIR system, Finding Region In the Pictures (FRIP) that uses human-centric relevance feedback to create similarity metric on the fly in order to overcome some of the limitations associated with RBIR systems. We use features, such as color, texture, normalized area, shape, and location, extracted from each region of a segmented image to represent image content. For each given query, we estimate local feature relevance using probabilistic relevance model, from which to create a flexible metric that is highly adaptive to query location. As a result, local data densities can be sufficiently exploited, whereby rapid performance improvement can be achieved. The efficacy of our method is validated and compared against other competing techniques using real world image data.

Keywords: CBIR, FRIP, image segmentation, feature extraction, MRS, Probabilistic Relevance Learning

Originality and Contribution

In this paper, we apply a new Probabilistic Feature Relevance Learning (PFRL) to our region-based image retrieval system, FRIP (Finding Region In the Pictures) in order to improve the retrieval performance and allow progressive refinement about query results to users.

The originality and contributions of this paper are like below;

- After image segmentation, we extract 5 concise and precise features from the segmented region and then normalize these to 0~1. These features are stored in the database as an index at the preprocessing step. Especially, for exact shape matching, we design the Modified Radius-based Signature (MRS) instead of using Fourier Descriptor or moments. Even though MRS uses simple equation, it shows the invariant results about small distortion as well as rotation and scale changes of region because it provides both local and global information of shape.

- We apply a new Probabilistic Feature Relevance Learning (PFRL) to our region-based image retrieval system, FRIP, with additional improvements (history concept and momentum for updating weights).

- With this method, FRIP system is able to improve the retrieval performance and reduce the iteration time as well as learn differential feature relevance in an efficient manner by estimating the strength of each feature dimension. In addition, since the estimation process is carried out locally in the vicinity of the input query, the method is highly adaptive to query location.

1. Introduction

The explosion of multimedia data in areas such as in medicine, commercial marketing, surveillance, education, etc. has rapidly increased the need for visual information retrieval. Therefore, content-based image retrieval (CBIR) has become an important research field. Traditional CBIR techniques use text to describe image content. While simple computationally, these techniques require manual annotation of each image in a database, which is prohibitive in large image database application. In addition, textual annotation is often ambiguous and inadequate for image database search. Furthermore, since textual annotation is dependant on language, a change in annotation language renders such techniques useless. In order to overcome these problems, current CBIR tools [1],[2],[3],[4] uses visual features such as color, texture, and shape instead of text, to represent image content.

Query-by-image is a popular way in which most CBIR systems work. Because such CBIR systems typically rely 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. As a result, most recent CBIR systems, such as QBIC [1], VisualSeek [2], Netra [3], and Blobworld [4], have been region-based.

Region-based image retrieval (RBIR) is an appealing approach that works in the following way: (1) images are segmented into several regions, (2) features are extracted from each region; and (3) the set of all features is used to represent image content in an image database. At the query time, features are first extracted from either the query image, a user-provided sketch, or a region of a segmented image. These extracted features are then matched against features representing images in the database. However, RBIR usually places high premium in segmentation quality, which is often difficult to achieve in practice due to several factors such as varying environmental conditions and occlusion. Thus, RBIR provides challenges as well as opportunities for developing effective interactive mechanisms that use imperfect information to conduct visual query.

In this paper, we propose a novel RBIR system, Finding Region In Pictures (FRIP) that meets the challenges facing region-based retrieval. Specifically, our system combines probabilistic feature relevance learning (PFRL) [5] with additional properties to allow robust image segmentation and capture of differential relevance of features in an efficient manner, thereby creating flexible metrics on the fly. In addition, since the estimation process is carried out locally in the vicinity of the input query, our method is highly adaptive to query locations. As a result, our FRIP system is able to significantly improve retrieval performance and reduce user's time investment. The efficacy of our method is validated and compared against PFRL using the Corel image database.

Figure 1 near here

The rest of the paper is organized as follows. Section 2 discusses previous work on region-based image retrieval. Section 3 provides an overview of our FRIP system. Section 4, introduces a stepwise similarity matching scheme employed in FRIP. Section 5 presents a modified PFRL scheme (MPFRL) and its application to FRIP. After that, Section 6 shows experimental results exploring the performance of our technique using real-world data. Finally, Section 7 concludes this paper by pointing out possible extensions to the current work and future research directions.

2. Related Work

In general, most of RBIR systems use the linear combination of distances between individual features to evaluate the similarity.

$$Sim(X,Y)=\sum_{i=1}^{p} w_i D(x_i,y_i) \tag{1}$$

where w_i is a weight of *i*-th feature and *D* is a distance between *i*-th feature of query region and *i*-th feature of a region in the database region

In this type of computer-centric approach [12], the “best” features, and their corresponding weights for individual feature distance are fixed, which cannot adaptively model high-level concepts and user’s subjective perception. For example, if one person wants to search a ‘red car’, a retrieval system that uses low-level feature, may only look for rectangle shapes with red color. In this case, if the user is not satisfied with the retrieval results, there is no chance to obtain single or several next nearest neighbors without restarting the query process from the beginning for a higher *k*. As such, some RBIR systems [1],[2] provide a user interface that allows the user to adjust weight parameters manually based on heuristic. In this case, the adjustment of weights for a large number of features is time consuming and exhausting.

To solve the weight adjustment problem of CBIR and to make up for the limitations in the current image segmentation technique, a combination of relevance feedback mechanisms [5],[10],[11] and RBIR is more desirable. However, most CBIR systems employing relevance feedback (e.g., Ciocca [10], MARS [11], PicHunter [13] and SufrImage [14]) use global features only (See Table 1). iPURE [15] is the only known RBIR system that employing relevance feedback. iPURE uses a revised intra-query learning method of MARS to reduce the weight bias problem. However, since this kind of query-shifting mechanism [11] is a mere shifting (rotation) of the query vector, it is insufficient to achieve desired goal in many problems of practical interest.

Table 1 near here

3. Overview of FRIP System

This system includes our robust image segmentation scheme using a circular filter and description of five features. For image segmentation, by using our proposed circular filter, we can maintain natural or artificial object's shape and merge small senseless textures (e.g. strips or spots) with neighborhood regions.

In this system, we use two kinds of different sized circular filters. The big one is an 11×11 window, and the small one is a 7×7 window. Filter size is scaled up (down) according to the image size. A full description of image segmentation of FRIP is described in [7]. From the segmented image, we need to extract features from each region. The contents of image should be extracted and stored as an index. Detailed feature information is better than coarse information for retrieval accuracy. However, since expensive management of storage and comparison time is less significant than retrieval accuracy, we extract five concise and precise features. Features (color, texture, normalized area (NArea), location and shape) of each region are used to describe the content of an image.

If we use Equation (1) to evaluate the similarity between two images, they must be normalized over a common interval to place equal emphasis on every feature score since the single distance may be defined on widely varying intervals. [12]. Five feature vectors (color, texture, NArea, location and shape) are normalized as soon as features are extracted from segmented regions before they are used for distance estimation.

3.1 Feature Normalization

For feature normalization, we choose to use the Gaussian normalization method [12]. Let $F_i = (f_{i1},...,f_{ik},...f_{iq})$ be the feature vector representing the *i*-th image in the database. We compute the mean, u_k, and standard deviation, σ_k, of the *k*-th feature dimension. We then we normalize the feature vectors to $N(0,1)$ according to:

$$F_i = (\frac{f_{i1}-u_1}{K\sigma_1},...,\frac{f_{ik}-u_k}{K\sigma_k},...,\frac{f_{iq}-u_q}{K\sigma_q}) \qquad (2)$$

$$= (f_{i1}^{'},...,f_{ik}^{'},...,f_{iq}^{'})$$

In Equation (2), if we assume that each feature is normally distributed and K=3, according to th 3-σ rule, the probability of an entry's value being in the range of [-1,1] is approximately 99%. A simple additional shift (Equation (3)) guarantees that 99% of feature values will be within [0,1].

$$F_i = (\frac{f_{i1}^{'}+1}{2},...,\frac{f_{ik}^{'}+1}{2},...,\frac{f_{iq}^{'}+1}{2}) \qquad (3)$$

where each $f_{i1}^{'}, f_{ik}^{'}, f_{iq}^{'}$ represents normalized feature vector within [-1,1]. If the shifted value is out of range, it is considered as an extreme value and can be discarded. To obtain u_k and σ_k of feature vectors, we use 200 random images as training data.

3.2 Extracting Color Feature

We extract the average color (*Ar, Ag, Ab*) of the RGB color space from each region instead of color histogram in order to reduce the storage space. The color distance ($d_{Q,T}^{C}$)

between query (Q) and target regions (T) is measured by the city-block distance:

$$d^{C}_{Q,T} = \left|Ar_Q - Ar_T\right| + \left|Ag_Q - Ag_T\right| + \left|Ab_{Q-}Ab_T\right| \tag{4}$$

3.3 Extracting Texture Feature

We choose the Biorthogonal Wavelet Frame (BWF) [8] as the texture feature. By BWF, we can obtain a fast and precise directional feature compare 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 sub-images. From the first-level high-pass filtered images, we calculate X-Y directional amplitude (Xd, Yd) of each region. The distance in texture ($d^{T}_{Q,T}$) between two regions, Q and T is computed by city-block distance:

$$d^{T}_{Q,T} = \left|\frac{Yd_Q}{Xd_Q} - \frac{Yd_T}{Xd_T}\right| \tag{5}$$

3.4 Extracting NArea (Normalized Area) Feature

Feature NArea is defined as the number of pixels (*NP*) of a region divided by the image size. The distance in NArea ($d^{NArea}_{Q,T}$) between query (Q) and target region (T) is computed by city-block distance:

$$d^{NArea}_{Q,T} = \left|NP_Q - NP_T\right| \tag{6}$$

3.5 Extracting Shape and Location Feature

In our FRIP system, we use two kinds of shape feature. For global geometric shape feature, we use eccentricity first. Then, for local geometric feature, we modify the radius-based shape signature (Modified Radius-based Signature: MRS) to be invariant under shape's scaling, rotation, and translation. MRS is only applied to shapes that satisfy the eccentricity at the first step. Eccentricity is used to significantly reduce the amount of computation required for shape matching at run time.

3.5.1 Eccentricity for Global Shape Feature

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).

$$E = \left| \frac{R_{Q\min}}{R_{Q\max}} - \frac{R_{T\min}}{R_{T\max}} \right| \tag{7}$$

3.5.2 MRS for Local Shape Feature

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

$$d^{P}_{Q,T} = \sqrt{(Cx_Q - Cx_T)^2 + (Cy_Q - Cy_T)^2} \tag{8}$$

To estimate MRS, the boundary of a segmented region is extracted using Sobel edge detection and edge thinning algorithms. After the boundary extraction, a shape is represented by a feature function called signature. Original signatures are invariant to translation, but they are sensitive to rotation and scaling. In our FRIP system, we modify the radius-based signature (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 (θ) is defined as the angle of axis of the least moment of inertia [9]. From a given orientation, we create a central axis that passes through the centroid with the sloping of θ degrees. Then we estimate the starting point as the farthest 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.

MRS is estimated from the starting point in the clockwise direction. However, because there exist two MRSs about a starting point, we have to consider two different directional MRSs (clockwise and counterclockwise).

Figure 2 near here

To save storage space, we obtain and store only one MRS that is estimated from the starting point in the clockwise direction. To achieve scale invariance, we use a combination of MRS and Equation (9). Equation (9) represents the differential variance between the average MRS ratio of the query region to a region in the database and individual MRS ratio.

$$d_{Q,T}^{MRS} = \frac{1}{N}\sum_{i=1}^{N}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\left|\frac{Q_i}{T_i}\right|\right)-\left|\frac{Q_i}{T_i}\right|\right]^2 \qquad (9)$$

For example, if a database region and scale-up (or down) query region have similar shapes, two shape distances are calculated to be approximately zero. Therefore, the smaller one can be chosen as the real shape distance ($d_{Q,T}^{MRS}$) between the query (Q) and the target MRS (T), even if the query region is rotated or translated. Here, to reduce the index size, we extract 12 radius (*N*) distance values with a 30 degree spacing and add these values to the region index.

By using MRS and Equation (9), our system is more robust against distortion, rotation, and scale changes of a region because they provide local information of shape with global information. Finally, we save 12 statistical properties of each region as an index to database. All properties are shown below:

Image

keys : imageNo

{

attribute int RegionNo;

attribute int AverRed(AR);

attribute int AverGreen(AG);

attribute int AverBlue(AB);

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);
}
```

4. Stepwise Similarity Matching

In 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 a region that he/she wants to search. Finally, by the user specified constraints, such as (1) color-care/don't care (2) scale (NArea)-care/don't care, (3) shape-care/don't care, (4) location-care/don't care, the overall matching score is calculated according to Equation (10).

4.1 Stepwise Distance Estimation

After feature normalization, its associated weights are initialized to 1/q, where q is the dimension of feature space. The system carries out image retrieval using K-nearest neighbor search based on the current weights to compute the similarity between the query region and database regions. It returns the top K nearest images including similar regions. During the matching process, the following steps are taken to calculate the final score according to the user constraints.

1. At the first step, average colors (and eccentricity, in the case of shape-care) of query region are compared only with regions in the database (Equation - (4), (7)).
2. If one (or two, in the case of shape-care) of the region distance(s) between the query and the database is (are) below a threshold, the next distance calculation is permitted. By the first step, we can reduce the comparison time for shape matching significantly.
3. The directional texture distance is calculated using Equation (5).
4. By user's choice, all or a few distances are calculated (Equation - (6), (8), (9)).

$$Score = w_1 d_{Q,T}^{C} + w_2 d_{Q,T}^{T} + w_3 d_{Q,T}^{NArea} + w_4 d_{Q,T}^{P} + w_5 d_{Q,T}^{MRS} \qquad (10)$$
$$(w_1 + w_2 + w_3 + w_4 + w_5 = 1)$$

From the calculated distances, the final score is estimated (Equation (10)) and the top k nearest images are displayed in the ascending order of the final score.

5. Modified Probabilistic Feature Relevance Learning (MPFRL) for FRIP's Matching Scheme

In this section, we apply a new Probabilistic Feature Relevance Learning (PFRL) method to our FRIP system with additional improvements (history concept and momentum term for weight updating) in order to improve the retrieval performance.

5.1 Probabilistic Feature Relevance Learning (PFRL) [5]

In a two class (1/0) classification problem, the class label *c* at query *x* is treated as a random variable from the distribution {Pr(1|x), Pr(0|x)}. The *c* at *x* can be characterized by $y \mid x = \begin{cases} 1 & c \mid_x = 1 \\ 0 & c \mid_x \neq 1 \end{cases}$. This gives rise to

$$f(x) \overset{\bullet}{=} Pr(1 \mid x) = Pr(y = 1 \mid x) = E(y \mid x), \qquad (11)$$

To predict *c* at *x*, *f(x)* is estimated from a set of training data. In image retrieval, however, the "label" of *x* is known. All that is required is to exploit differential relevance of input features to image retrieval. The least-squares estimate for *f(x)*, given that *x* is known at dimension $x_i = z$, is

$$E[f \mid x_i = z] = \int f(x) p(x \mid x_i = z) dx, \qquad (12)$$

where $p(x \mid x_i = z)$ is the density of other input features. Equation (12) shows the predictive strength (probability) when the value of just one (x_i) of the input features is known. Then a feature relevance measure for query *z* can be given by

$$r_i(z) = E[f(x) \mid x_i = z] \qquad (13)$$

The relative relevance, as a weighting scheme, can then be given by

$$w_i(z) = (r_i(z))^t / \sum_{l=1}^{q} (r_l(z))^t \tag{14}$$

where t=1,2, giving rise to linear and quadratic weightings, respectively. Finally, Equation (14) can be changed following the exponential weighting scheme

$$w_i(z) = \exp(Tr_i(z)) / \sum_{l=1}^{q} \exp(Tr_l(z)) \tag{15}$$

where *T* is a parameter that can be chosen to maximize(minimize) the influence of r_i on w_i. When *T*=0, weight $w_i = 1/q$, thereby ignoring any difference between the r_i's. On the other hand, when *T* is large, a change in r_i will be exponentially reflected in w_i. The exponential weighting is more sensitive to changes in local feature relevance (15) and gives rise to better performance improvement.

In order to estimate Equation (14) and Equation (15), one must first compute Equation (13). The retrieved images with relevance feedback from a user can be used as training data to obtain estimates for Equation (13), hence Equation (14) and Equation (15). Let $\{x_j, y_j\}_1^K$ be the training data. Here x_j denotes the feature vector representing the *j*-th retrieved image, and y_j is marked with either 1 (relevant) or 0 (irrelevant) by the user as the class label associated with x_j. To compute $E[f(x) | x_i = z]$, recall that $f(x) = E[y | x]$. Thus, it follows that

$$E[f | x_i = z] = E[y | x_i = z]$$

However, since there may not be any data at $x_i = z$, the data from the vicinity of x_i at *z* are used to estimate $E[y | x_i = z]$, a strategy suggested in [6]. Therefore, Equation (13) can be estimated according to

$$\hat{E}[y | x_i = z] = \sum_{j=1}^{K} y_j 1(| x_{ji} - z | \leq \Omega) / \sum_{j=1}^{K} 1(| x_{ji} - z | \leq \Omega) \tag{16}$$

where 1(•) is an indicator function. That is, 1(•) returns 1 if its argument is true, and 0 otherwise. Ω can be chosen so that there are sufficient data for the estimation of Equation (15). Ω can be chosen such that

$$\sum_{j=1}^{K} 1(| x_{ji} - z | \leq \Omega) = C \tag{17}$$

where $C \leq K$ is a constant.

5.2 Modified PFRL (MPFRL)

The original PFRL algorithm [5] is memoryless in that retrieval in previous iterations does not contribute to feature relevance estimates in future iterations. As a result, retrieval performance may fluctuate from one iteration to the next, which might cause performance degradation. To overcome this limitation, we modify PFRL with two additional features.

First, we add a momentum term to the weight update rule. That is, since the weights from previous iteration are also determined by user's perception, they include important attributes for predicting weights for the next iteration. Furthermore, if we only use the weights estimated from the current user action, these weights may be dramatically changed or oscillated around the solution so that there may not be any increase in retrieval precision. By using this momentum term, we can prevent unstable changes of weights and help speed up learning.

Second, we keep user's entire past feedback actions in order to improve performance for future iterations. If we are to ignore feedback actions from previous iterations, rejected images, which were selected as irrelevant in the past, may be retrieved again so that performance at the next iteration may not improve in spite of increase in iteration time. Therefore, we use previous relevance feedback as *history* to help improve the performance and reduce user's time investment. Here, our history is consisted of two parts: One is the positive history and the other is the negative history.

The modified probabilistic feature relevance learning (MPFRL) algorithm works in the following way:

The first iteration is determined by weights initialized to (1/q). The resulting retrievals (images) are presented to the user using initial weights and Equation (10). Then, the user marks the retrieved images (by clicking on them with a mouse) as either relevant or irrelevant, conditioned on their similarity to the desired target image. If the user does not know if a displayed image is similar to the desired target image, the user does not need to select that image. After that, the user presses the "Refinement" button to adjust the weights for the next iteration. At this time, the new weights are estimated from the marked images using the Equation (16) and (15). In our experiments, we choose the parameter T (Equation (15)) to be one. Then, these weights are updated by

$$w_i^{t+1} = (1-\alpha)\cdot w_i^t + \alpha \cdot w_i^{t-1} \qquad (18)$$

where $0 \le \alpha \le 1$ is an adjustable parameter that determines the extent to which the two terms should be combined, and *t* presents iteration time.

The next weight, w_i^{t+1}, is updated by a linear combination of the weight of previous iteration, w^{t-1}, and the one from the current iteration, w_i^t, by Equation (15). After the weight update using Equation (18), the next weight, w_i^{t+1}, is used to carry out K-nearest neighbor search at the next iteration. To determine α, we calculate average of weight difference in the five features between iterations. To do so, we use our database images having a Sun region because it preserves consistent shape, color, texture, NArea, and location. If we do not use α (alpha 0), the difference graph is dramatically changed at every step, even though retrieval performance is good. If α is 0.2 or 0.3 (alpha 2, 3), the difference graph is changed more slowly but the retrieval performance is not superior to alpha being 0. On the contrary, if we choose α to be 0.1 (alpha 1), the difference graph is changed slowly and retrieval performance is relatively superior to others. Therefore, in this paper, we select α to be 0.1.

Figure 3 near here

During the K-nearest neighbor search step, we add *history* scheme to the weighted similarity computation using Equation (19). Here, *history* is generated by user's selection (action): at each iteration, if the user marks the k images as relevant or irrelevant, these images are recorded in the positive history $H_P = \{h_1(I_1, P_1), h_2(I_2, P_2), ..., h_i(I_i, P_i)\}$ and negative history $H_N = \{h_1(I_1, N_1), h_2(I_2, N_2)$ $,..., h_i(I_i, N_i)\}$ respectively.

$$\begin{aligned}
&\mathit{if}\ \ y \in H_P\{h_i,.., h_n\} \\
&\qquad D(x,y) = [\sum_{i=1}^{F} w_i d_i(f_i(x), f_i(y))] / \hat{K} \\
&\mathit{else\ if}\ \ y \in H_N\{h_i,.., h_n\} \\
&\qquad D(x,y) = [\sum_{i=1}^{F} w_i d_i(f_i(x), f_i(y))] \cdot \hat{K}
\end{aligned} \qquad (19)$$

In the above Equation, x is an image including the query region and y is a database image. $H_P\{h_i,..,h_n\}$ represents the set of relevant images and $H_N\{h_i,..,h_n\}$ represents a set of irrelevant images. Distance (D) of feature vectors is estimated by weighted sum of five distances (d_i, *i*: 1~5- Equation (4), (5), (6), (8), (9)) and $\hat{K}$ is a constant decision parameter for top k. In this paper, we set $\hat{K}$ to 10.

At first, positive history (H_P) checks whether the image y is included in H_P or not. If y is included in H_P, the distance between x and y is divided by the decision parameter $\hat{K}$. On the other hand, if y is included in negative history (H_N), the distance is multiplied by parameter $\hat{K}$. Here, if the iteration time is zero or image is not included in any histories, decision parameter $\hat{K}$ does not affect the distance.

From the *history* scheme, we update the distance order between the query and a database region, and return the top k-nearest neighbor images including similar regions. For example, if one image is selected as irrelevant at the previous iteration, it will get a larger distance (D) by $\hat{K}$ and has a higher probability of being dropped from the top k images. On the other hand, if one image is selected as relevant, it will get a smaller distance (D) by $\hat{K}$ and has a higher probability of being included in the top k images. This process is repeated until all of the desired images are found or the predefined precision threshold is reached. The main steps of modified probabilistic feature relevance learning algorithm is shown below.

```
1. Let i be the current query; initialize the weight vector w to $\{1/q\}_1^q$.
2. Compute k nearest images using initial w and Equation (10).
3. User marks n images as relevant or irrelevant.
4. Initiate history $H_P$ and $H_N$
5. While [(precision < $\theta$) or (user is not satisfied)] Do
    (a) Tset← {marked n images}
    (b) Estimate w from Equation (15) and (16) using training data in Tset
    (c) Update new weights from Equation (18)
    (d) Compute k nearest images using new weights and history using Equation (19)
    (e) User marks the n images as relevant or irrelevant.
    (f) Update history $H_P$ and $H_N$.
```

Figure 4 near here

6. Experimental Results

We have performed a variety of queries using a set of 3,000 images from the *WWW* and Corel photo-CD, containing various categories such as natural images (e.g. landscape, animals) and synthetic images (e.g. graphics, drawing). This system is developed in Visual C++ 6.0 language as an off-line system. The average segmentation time requires approximately 30 seconds per image and the average retrieval time requires 2 seconds per 1000 images using a Pentium PC, 450 MHz. The retrieval results are accessible at *http://vip.yonsei.ac.kr/Frip.*

Figure 5 near here

The experiments are carried out on 8 specific domain data such as sun, tiger, car, eagle, airplane, and flower between MPFRL with that of PFRL.

First, a user chooses a query image and pushes the segment button. Then, the user selects *k* and "double click" the region that he/she wants to retrieve. A user can choose color, NArea (scale), shape, and location (texture is included as a default condition) constraints in order to search regions more precisely. In this experiment, we choose all user constraints and top 18 nearest neighbors are returned that provide necessary relevance feedback.

In all the experiments, the performance is measured using the average retrieval precision.

$$precision = \frac{Positive\ Retrievals}{Total\ Retrievals} \qquad (20)$$

Figure 6 shows the performance achieved by MPFRL and PFRL. Here, we can see that the precision of PFRL is either stable, linearly increase or decrease in spite of increased iteration time, but the precision of MPFRL is never goes down and improves steadily, because MPFRL uses the history concept.

Figure 6 near here

Figure 7 shows 5 weight changes on 4 categories as a function of iteration. Here, $w_i(i:1\sim 5)$ is a weight for each of five features (color, texture, NArea, shape and location). In Figure 7, it can be seen that for all the queries, the weights associated with the relevant features are increased and the weights associated with the irrelevant ones are decreased after learning has taken place. These results show convincingly that our method can capture local feature relevance.

Figure 7 near here

Figure 8, 9, 10 and 11 show top 10 retrieval results about query region with corresponding segmented images.

Figure 8 near here
Figure 9 near here
Figure 10 near here
Figure 11 near here

7. Conclusion

This paper presents a region-based image retrieval system, FRIP, that is able to tolerate, to the extent possible, region scaling, rotation and translation, and that incorporates probabilistic feature relevance learning with two additional properties (history concept and weight momentum) to enable our system to perform region-based image retrieval in a desired manner. The experimental results using real image data show that our MPFRL algorithm can indeed rapidly improve region-based retrieval performance of an image database system. Furthermore, since our relevance estimate is local in nature, the resulting retrieval, in terms of the shape of the neighborhood, is highly adaptive and customized to the query location.

A potential extension to the technique described in this paper is to consider additional derived variables (features) for local relevance estimate, thereby contributing to the distance calculation. The derived features are functions, such as linear functions, of the original features. When the derived features are more informative, huge gains may be expected. On the other hand, if they are not informative enough, they may cause retrieval performance to degrade since they add to the dimensionality count. The challenge is to be able to have a mechanism that computes such informative derived features efficiently.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments.

Reference

[1] Faloutsos, M. Flickner, W.Niblack, D. Petkovic, W. Equitz, R. Barber. Efficient and Effective Querying by Image Content. Research Report #RJ 9203 (81511), IBM Almanden Research Center, San Jose, Aug. 1993.

[2] J.R. Smith and S.F. Chang. VisualSEEk: A Fully Automated Content-Based Image Query System. ACM Multimedia, Boston MA, 1996.

[3] W.Y.Ma, B.S. Manjunath. Netra: A toolbox for navigating large image database. IEEE International Conference on Image Processing, 1997.

[4] Carson, M.Thomas, S. Belongie, J.M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In Proc. Int. Conf. Visual Inf. Sys, 1999.

[5] Peng, J., Bhanu, B., and Qing, S. Probabilistic Feature Relevance Learning for Content-Based Image Retrieval. Computer Vision and Image Understanding, 1999; 75(1): 150-164.

[6] J.H. Friedman. Flexible Metric Nearest Neighbor Classification. Theh. Report, Dept. of Statistics, Stanford University, 1994.

[7] ByoungChul Ko, Hae-sung Lee, Hyeran Byun. Region-based Image Retrieval System Using Efficient Feature Description. Proceedings of the 15th International conference on Pattern Recognition, 2000;4(4):283-286.

[8] Sideney Burrus, Ramesh A. Gopinath, Haitao Guo. Introduction Wavelets and Wavelet Transforms. A primer, Prentice-Hall, 1998.

[9] Anil K. Jain. Fundamentals of digital image processing. Prentice hall international editions, 1989.

[10] G. Ciocca, R. Schettini. A relevance feedback mechanism for content-based image retrieval. Information Processing and Management, 1999; 3:605-632.

[11] Yong Rui, Thomas S. Huang and Sharad Mehrotra. Content-based image retrieval with relevance feedback in MARS. IEEE International Conference on Image Processing, Santa babara, Oct. 1997; 815-818.

[12] Yong Rui, Thomas S, Michael Ortega, and Sharad Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. IEEE Transaction on Circuits and Systems for Video Technology, Vol. 8, No. 5, Sept. 1998; 644-655.

[13] I,J. Cox, M.L. Miller, T.P. Minka, et al. An optimized interaction strategy for Bayesian relevance feedback. In Proceeding of IEEE CVPR'98. Santa babara, 1998

[14] C. Nastar, M. Mitschke, and C. Meihac. Efficient query refinement for image retrieval. In Proceeding of IEEE CVPR'98. Santa babara, 1998.

[15] Gaurav Aggarwal, Pradeep Dubey, et al. iPURE: Perceptual and user-friendly retrieval of images. In Proceeding of IEEE International Conference on Multimedia and Expo 2000, 2000;3(2);693-696.

Appendix. Copies of all figures and tables

Table 1. CBIR system classification

System type	Characteristics	CBIR System
Relevance feedback system	• Only use global features	MARS, PicHunter, SurfImage
Region-based system	• Image segmentation • Use region level features	Netra, Blobworld
Others	• Coarse image segmentation • User modifies the global features manually	QBIC, VisualSEEk
Region-based relevance feedback system	• Image segmentation • Use region level features • User modifies the feature parameters manually or automatically	FRIP, iPURE

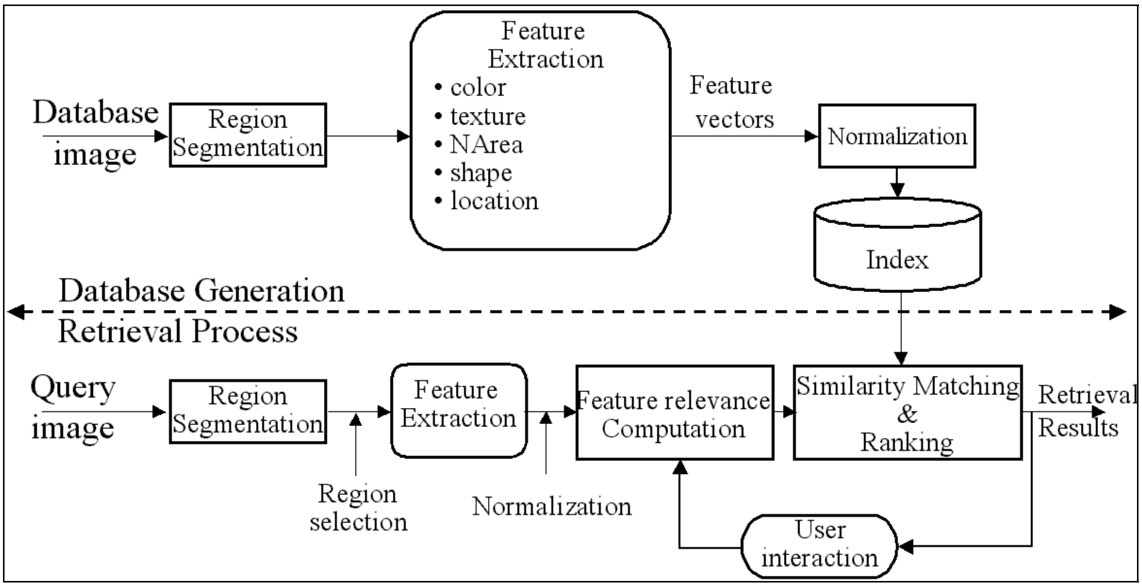
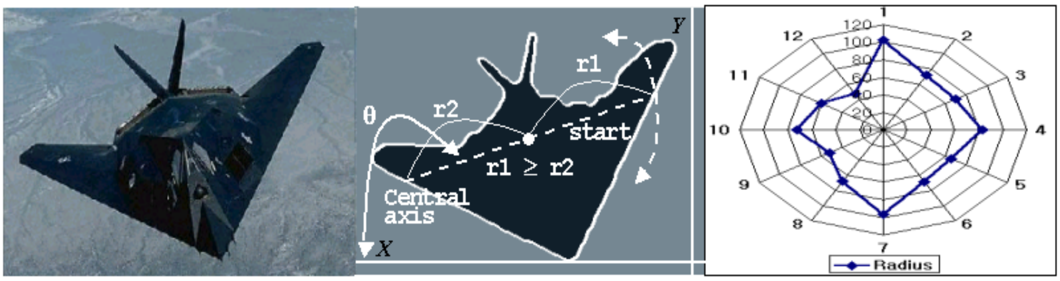


Figure 1. FRIP system architecture.



(a) (b) (c)

Figure 2. Region signature: (a) original image (b) region boundary (c) polygon by MRS

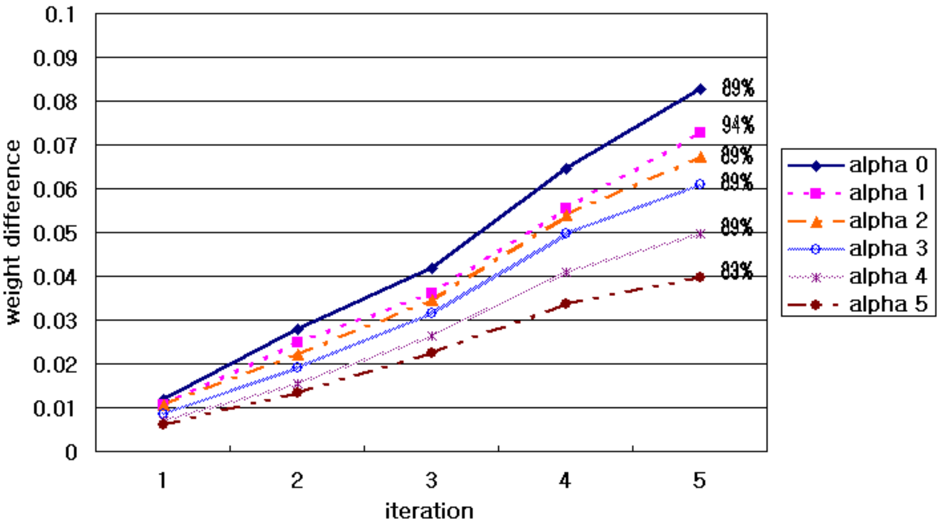


Figure 3. Cumulative distribution of weight difference (% : precision)

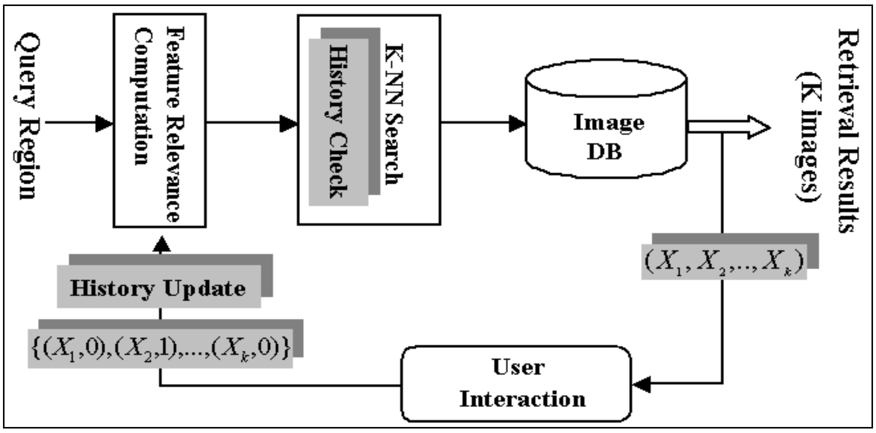
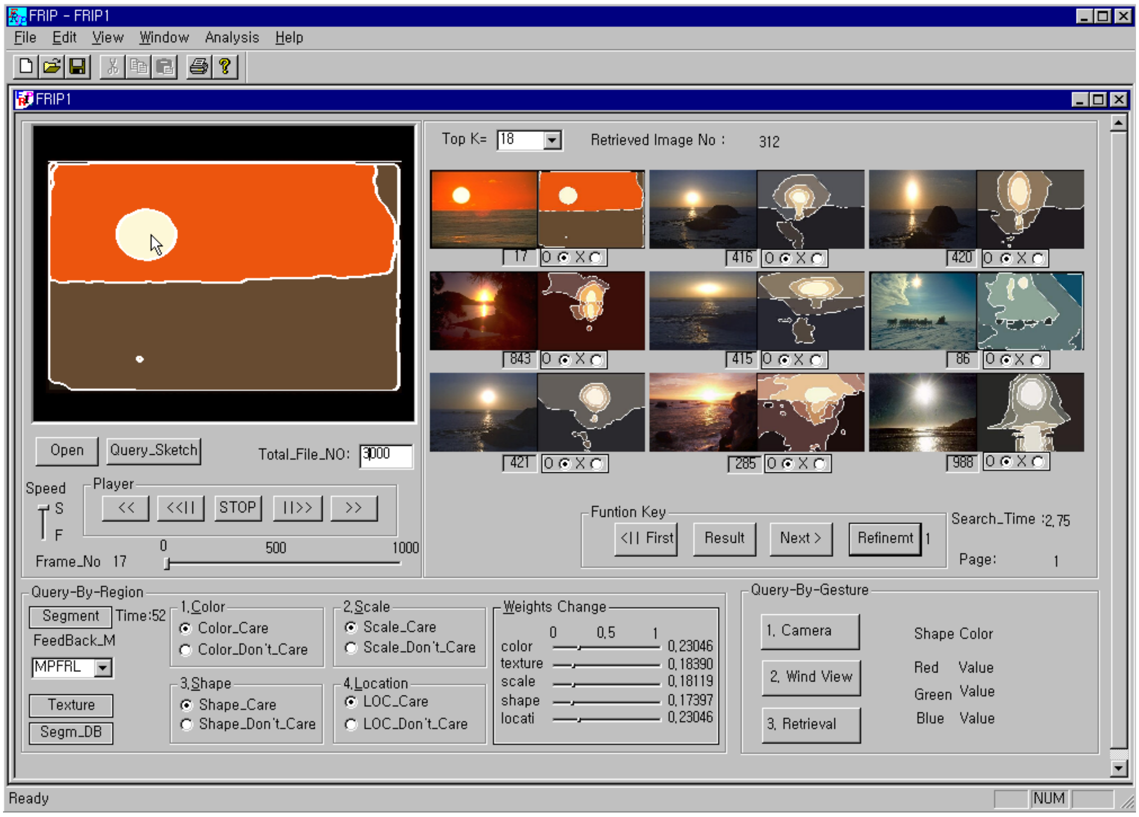


Figure 4. *FRIP*'s matching procedure using MPFRL



Top K= 18

Retrieved Image No : 312

Total_File_NO: 3000

Frame_No 17

Refinemt 1

Search_Time :2.75

Page: 1

Time:52

FeedBack_M: MPFRL

1.Color: ● Color_Care ○ Color_Don't_Care

2.Scale: ● Scale_Care ○ Scale_Don't_Care

3.Shape: ● Shape_Care ○ Shape_Don't_Care

4.Location: ● LOC_Care ○ LOC_Don't_Care

color	0.23046
texture	0.18390
scale	0.18119
shape	0.17397
locati	0.23046

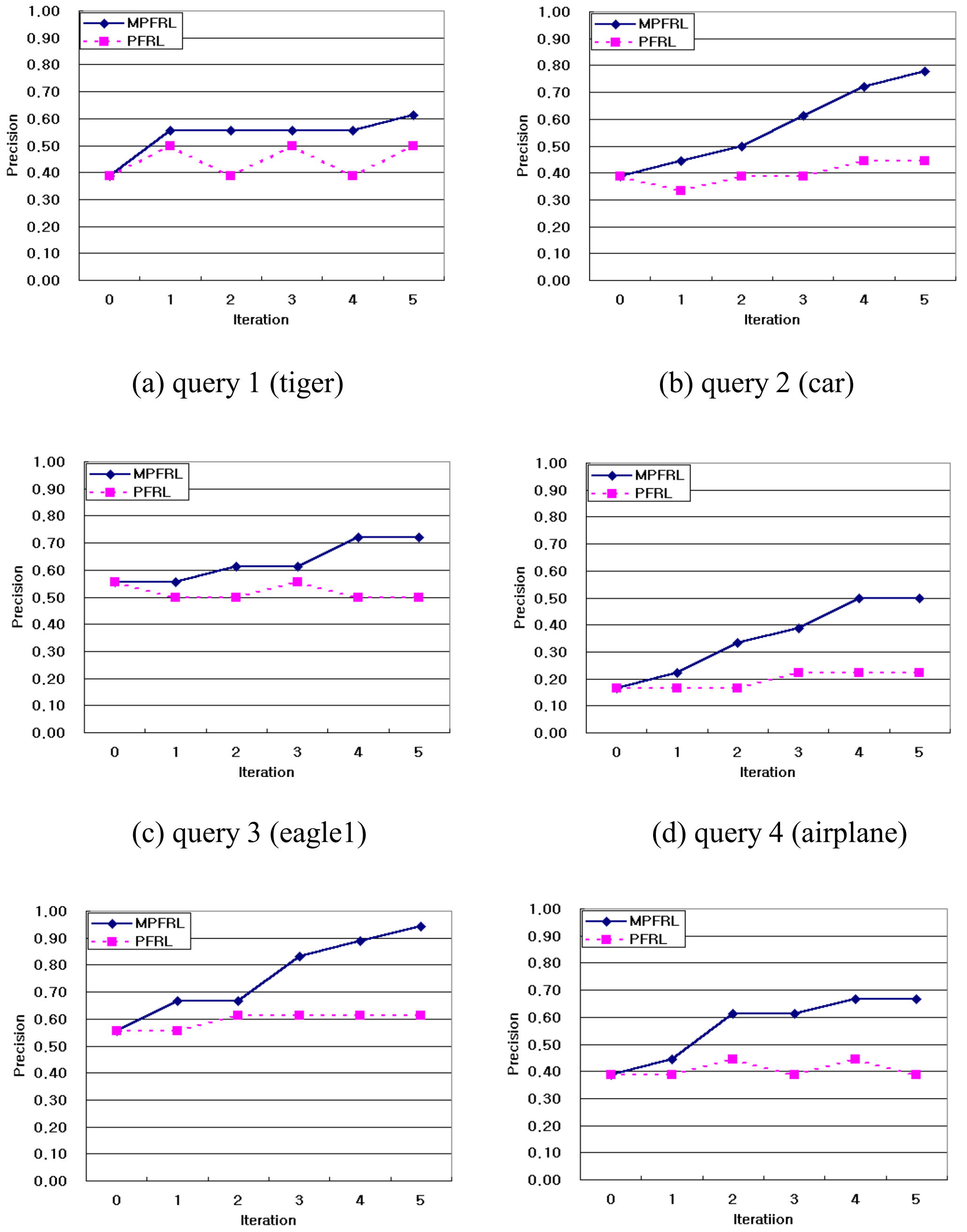
Shape Color

Red Value

Green Value

Blue Value

Figure 5. User interface of *FRIP* system



(a) query 1 (tiger)

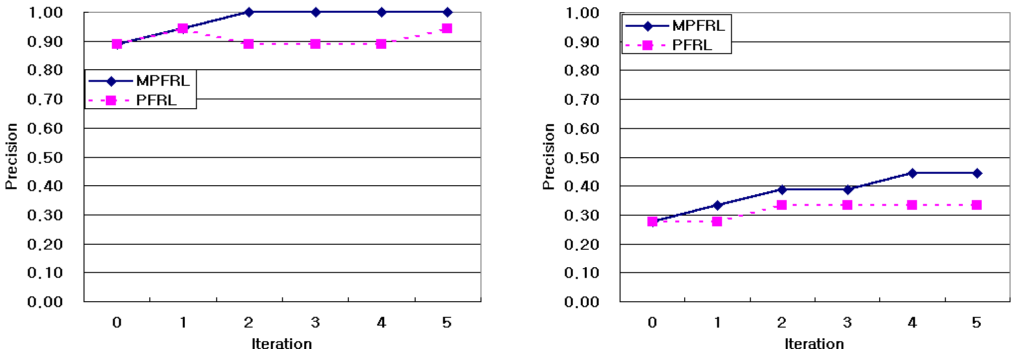
(b) query 2 (car)

(c) query 3 (eagle1)

(d) query 4 (airplane)

(e) query 5 (sun)

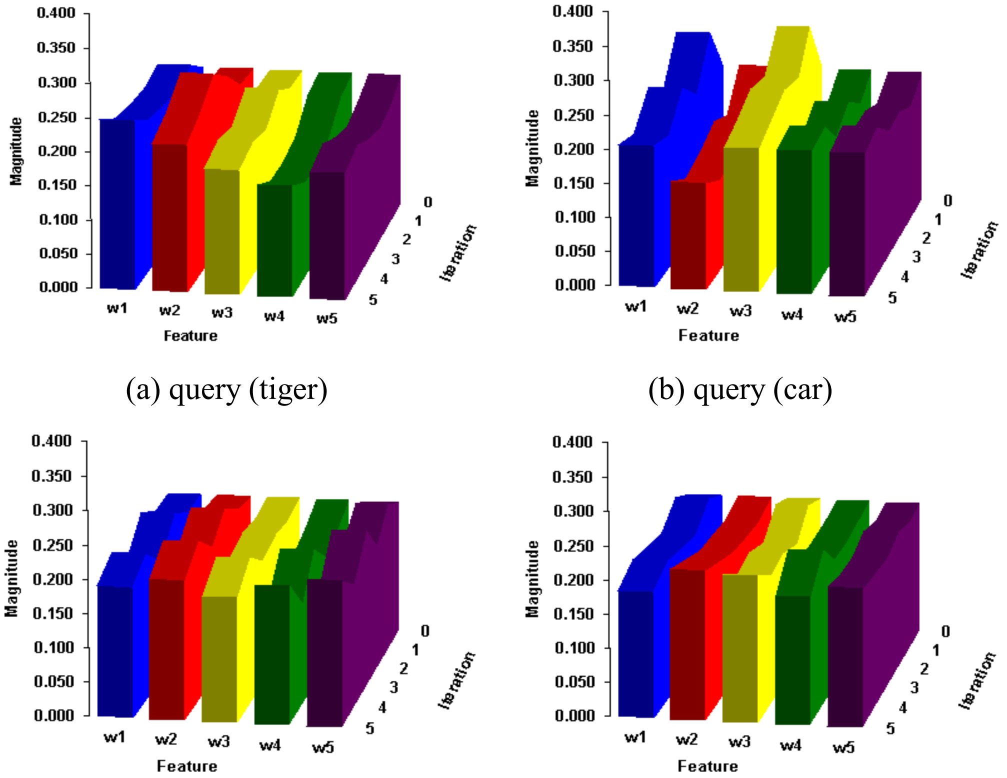
(f) query 6 (eagle2)



(g) query 7 (flower)

(h) query 8 (white bear)

Figure 6. Performance of 8 specific regions between MPFRL and PFRL



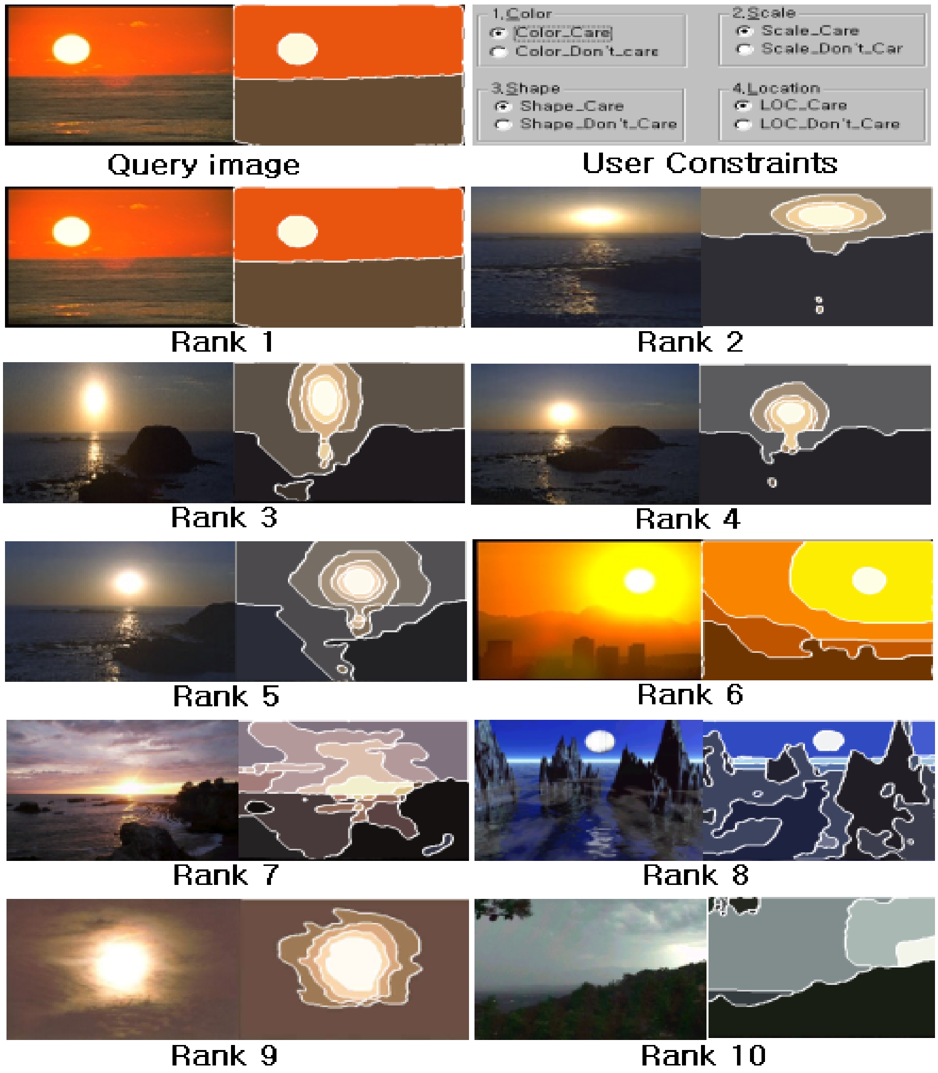
(a) query (tiger)

(b) query (car)

(c) query (eagle1)

(d) query (sun)

Figure 7. Weight changes of features according to the iteration



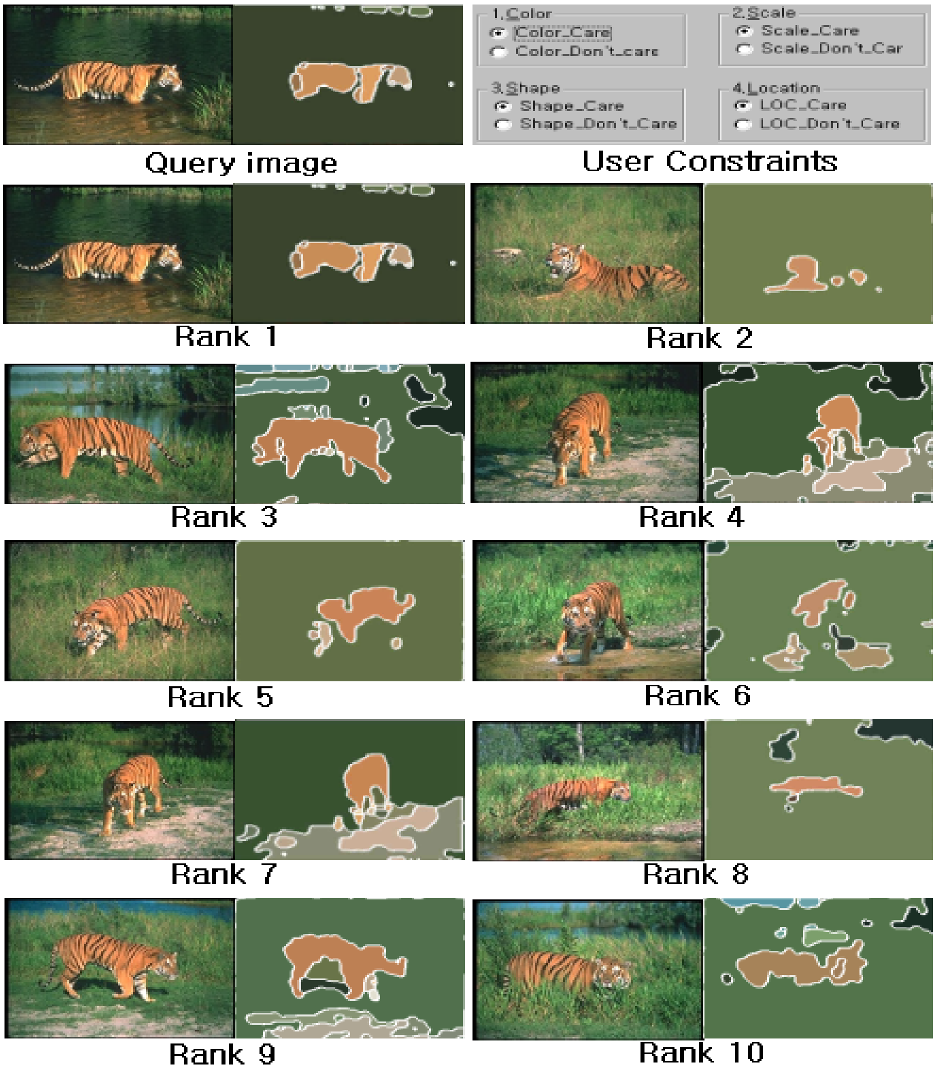
1.Color
● Color_Care
○ Color_Don't_care

2.Scale
● Scale_Care
○ Scale_Don't_Car

3.Shape
● Shape_Care
○ Shape_Don't_Care

4.Location
● LOC_Care
○ LOC_Don't_Care

Figure 8. Retrieval results (query region: sun-5 relevance feedback, top-10):Left is the original image, right is the segmented image



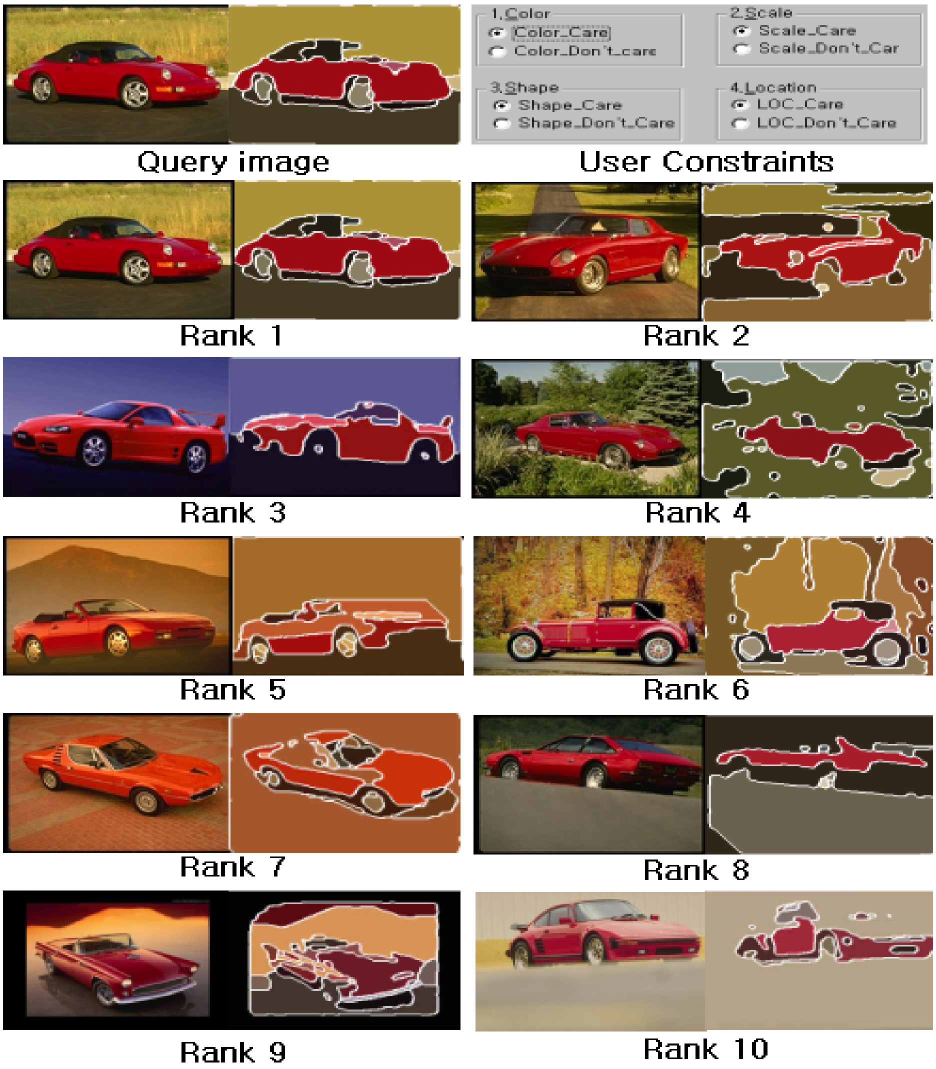
1.Color
● Color_Care
○ Color_Don't_care

2.Scale
● Scale_Care
○ Scale_Don't_Car

3.Shape
● Shape_Care
○ Shape_Don't_Care

4.Location
● LOC_Care
○ LOC_Don't_Care

Figure 9. Retrieval results (query region: tiger-5 relevance feedback, top-10):Left is the original image, right is the segmented image



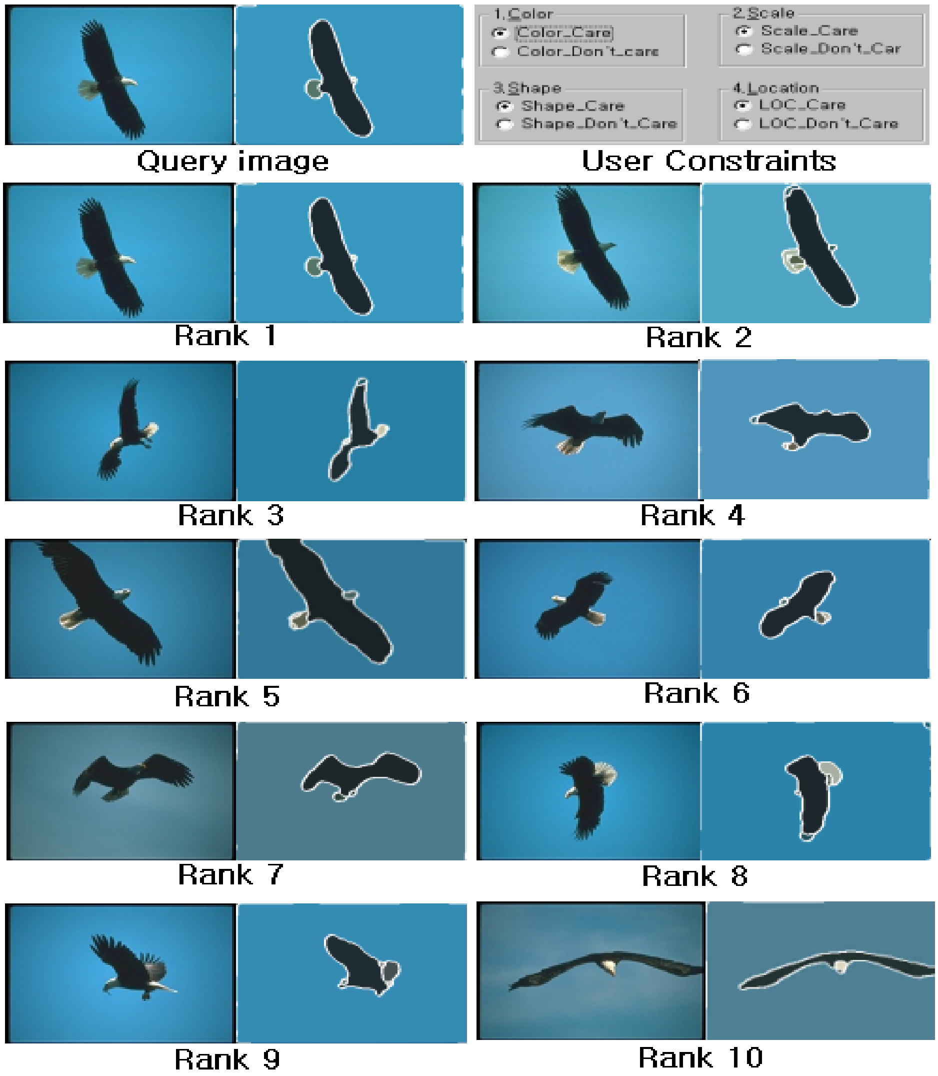
1.Color
● Color_Care
○ Color_Don't_care

2.Scale
● Scale_Care
○ Scale_Don't_Car

3.Shape
● Shape_Care
○ Shape_Don't_Care

4.Location
● LOC_Care
○ LOC_Don't_Care

Figure 10. Retrieval results (query region: car-5 relevance feedback, top-10):Left is the original image, right is the segmented image



1.Color
● Color_Care
○ Color_Don't_care

2.Scale
● Scale_Care
○ Scale_Don't_Car

3.Shape
● Shape_Care
○ Shape_Don't_Care

4.Location
● LOC_Care
○ LOC_Don't_Care

Figure 11. Retrieval results (query region: eagle-5 relevance feedback, top-10):Left is the original image, right is the segmented image