An Automatic Image Retrieval System for Large Multimedia Databases

Been-Chian Chien
Department of Information Engineering, I-Shou University, Taiwan, R. O. C.
Email: cbc@isu.edu.tw

Din-Yuen Chan
Department of Information Engineering, I-Shou University, Taiwan, R. O. C.
Email: dychan@isu.edu.tw

Jin-Der Li
Institute of Information Engineering, I-Shou University, Taiwan, R. O. C.
Email: m873002m@isu.edu.tw

In this paper, an object-based image retrieval system is proposed to reduce the gap between low-level features of images and high-level human concepts in large image databases. The proposed system automatically segments an image into the best number of regions to figure out the interested objects of the image. The features of each object in the image including color, shape and spatial relationship are then extracted and stored in a uniform representation. For measuring the similarity between images and satisfying human subjective concepts, the object-based similarity matching algorithms and the relevance feedback mechanism are developed. The relevance feedback mechanism will update the weights of different features inside each object and among objects by the user's response for improving the effectiveness and efficiency. The experiments show that the proposed system can capture human perception and retrieve relevant images in two or three feedback processes.

Keywords: multimedia systems, multimedia databases, image retrieval.

1. Introduction

With the advance in computer technology, the volumes and diversity of image data have been generated and stored in digital archives. In order to make use of this vast amount of image data, efficient and effective retrieval techniques need to be developed. A traditional image retrieval systems is accomplished by using keyword annotations and textual queries. In this approach, by assigning related keywords to the corresponding images, the target images can be accessed through SQL-like queries. There are several drawbacks with this approach. First, a large amount of manual effort is required while annotating the keywords on image data. Second, the keyword annotations may be distinct and inconsistent among different subjective interpretations of index creators. To overcome above drawbacks, content-based image retrieval uses objective content of images, such as color, shape, texture, and spatial relationship, as the image index. The image contents used in content-based image retrieval systems are low-level features, which can be extracted by some image processing techniques automatically. Thus, the extracted features are always consistent. Although the content-based approach improves part of drawbacks in the annotation-based approach, it loses the high-level semantic meaning of images by contraries. That is, the cognitive gap between low-level features and human high-level concepts in an image should become a problem. For example, users try to get a picture with apples on it, but systems may return the pictures with several red balls on them because of the similarity of their color and shape. The other problem is the subjectivity of human perception [Rui (1998A)]: Different users, or even the same user under different situations, may perceive the same visual content differently.
In order to solve the problems and improve the efficiency and effectiveness in content-based image retrieval, so many techniques have been developed. Some representative systems like QBIC [Niblack (1993)] developed by IBM Almaden Research Center, VisualSEEK [Smith (1997)] of Columbia University, and MARS [Rui (1998B)] of University of Illinois have been well-known by researchers. These systems adopt different low-level features including color, shape, texture and spatial relationship as their visual features. However, the low-level features describe the objective characteristics in images. It is still far from the human perception and subjectivity. The MARS system proposes a relevance feedback mechanism based on the technique of textual information retrieval to overcome the gap between similarity metrics of low-level features and the user’s perspectives. Although the system introduce the concept of interactive retrieval to approximate the user’s information need, their approach considers only on low-level image features. Their relevance feedback mechanism takes one image as an object, the feedback results will be improved by refining the weights in low-level features and metrics of image matching. Since a general user is usually attracted by the objects in an image instead of the features of color or texture, the low-level image feature is not enough for human subjectivity.

In this paper, we proposed an image retrieval system with object-based relevance feedback approach to retrieve the relevant images that are the user’s want to through the user’s interaction. For reducing the gap between human high-level concept and the low-level features of images, the proposed approach employs the similarity matching and relevance feedback based on two layers: the object layer and the image layer. First, an image is segmented into the best number of regions. We take each region as an object, then the features in each object and the spatial relationships among different objects are extracted. The similarity measure algorithms are given to match object-pairs and measure the similarity between images. Finally, the relevance feedback mechanism is developed to adjust the weights of the low-level features in each object and the spatial relationships among objects. Our system is built on the environment of PC/Windows98, the experiments show that the proposed system can capture human perception effectively and retrieve relevant images in few feedback processes efficiently.

The remainder of this paper is organized as follows: First, in Section 2, we describe the framework of content-based image retrieval and introduce the architecture of the image retrieval system used in this paper. Section 3 outlines the method of image segmentation and the features used in the system including color, shape and spatial relationship. Then, the similarity measure and relevance feedback techniques are presented in Section 4. The prototype system and experimental results are shown in Section 5. Two examples of image retrieval with relevance feedback are also given here. Finally, a summary and future work are depicted in Section 6.

2. The architecture for image retrieval

A general framework of content-based image retrieval usually consists of two main issues: the image feature extraction and the similarity measures. The function of feature extraction is to extract low-level image features automatically and store them as a feasible representation. The similarity measure is to compute the similarity between the features of query image and the image features in the feature database using a similarity metric. Based on the fundamental framework, the architecture of our system shown in Figure 1 contains several major components. Our system provides a graphical user interface to assist users in images collecting and querying. The image segmentation and feature extraction are used to analyze and represent the low-level features of images as a uniform representation. Then the features are indexed by the multi-dimensional access structure and stored in feature databases. The block of similarity matching supports the similarity measure of image features. There are two similarity matching algorithms in the proposed system. One is the matching of object-pairs, and the other is the matching of similar image. The relevance feedback gives a concise mechanism to update the weights of features for retrieving the relevant images.
In the database creation phase, we apply Lin’s color image segmentation [Lin (1997)] to segment the image into the best number of regions by CIE LUV color space automatically. After the significant objects are extracted, we compute the shape of each object in the invariant moments [Foley (1996)]. Furthermore, the location of each object is recorded as its spatial position. Then, we arrange each image as the corresponding features representation and store the features to the feature database via multi-dimensional indexing structures.

In the image query phase, the user interface in the proposed system supports two query methods for users:
1. Query by image example: The user gives a sample picture to the system, and the system will find the images that are similar to the sample picture.
2. Query by sketch: If the user cannot yield any proper picture as a query image on his hand, the system allows users to specify the query by roughly drawing the sketch.

While an image query is given, the same process of feature extraction is performed. The extracted features of the query is then compared with the features in the feature database by the similarity measure algorithms. After similarity matching, the similar images measured by low-level image features will be retrieved from the image database and be displayed on the screen. However, the retrieval results may not fully be satisfied by the user. In such circumstances, the user can mark the relevant images through the user interface interactively and refine the retrieval results by the processing of the relevance feedback mechanism repeatedly. The high-level human concept usually focuses on some objects of an image or one of the image features. The feedback from users is the best and the most natural way to capture the concepts of users. The details of the system will be presented in the following sections.

3. The feature extraction of objects

3.1 Image segmentation

The issue of segmentation is very important to image retrieval. The extraction of objects from
an image depends on good segmentation. So many segmentation approaches have been proposed in the past [Rui (1999)]. Here, we modify the Lin’s approach [Lin (1997)] to segment color images into the best number of regions. First, the image is partitioned into blocks of size $X_b \times Y_b$. Each block is assigned a representative color, which is the mean value of colors for all pixels in the block. We set each block to be an individual region initially. Then each block is merged with its neighboring blocks having the smallest color difference into a new region. Such process is iterative until only one region remains. For determining the best number of segmented regions, the performance index $Q_k$ [Duda (1973)] is defined as the sum of mean square errors between the image of the $k$ segmented region and the original image. We say that the number of regions $r$ is the best segmentation if $\delta(r)$ is maximum. The definition of $\delta(k)$ is

$$\delta(k) = \frac{Q_{k+1} - Q_k}{Q_k - Q_{k+1}}.$$  

For example, Figure 2(a) is the given original image. After the process of segmentation, the corresponding relation between the performance index $Q_k$ and the number of segmented regions $k$ is shown in Figure 2(c). The $\max\{\delta(k)\}$ will occur in the knee of curve where $k$ equals to 7. Figure 2(b) is the result of seven segmented regions. We find that the segmented result in this example approximates to human visual perception. The main four objects inside the image will be used as the significant objects to represent the image.

![Figure 2: (a)The original image; (b)the image after segmentation; (c) curve of $Q_k$.](image)

### 3.2 CIE LUV color space

RGB color space is the most common way to represent colors in computer systems. According to the related researches [Gonzalez (1992)], however, RGB color space cannot catch human’s visual sense while judging the degree of similarity between two distinct colors. CIE LUV color space is more natural than RGB color space for human’s perception in color recognition. For this reason, the color segmentation depicted in above subsection uses CIE LUV color space to be the representative color space of image processing.

The transformation from RGB color space to LUV color space as the following formulas [Gonzalez (1992)]:

$$
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} =
\begin{bmatrix}
0.490 & 0.310 & 0.200 \\
0.177 & 0.813 & 0.001 \\
0.000 & 0.100 & 0.990
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
$$
where \([R, G, B]\) is the value of a pixel in RGB color space, \([L, u, v]\) is the value of a pixel in \(LUV\) color space and \([X_0, Y_0, Z_0]\) is the value of \([X, Y, Z]\) transformed by a referential color like white or black.

After the segmentation, the final color for the best number of segmented regions is the representative color of each region. We take each region as an object and assume that \(O_i\) and \(O_j\) are two objects segmented from two different images. The color difference between two \(LUV\) colors \([L_i, u_i, v_i]\) and \([L_j, u_j, v_j]\) is computed by Euclidean distance. Thus, the color difference between objects is defined as Equation (1).

\[
\delta_c(O_i, O_j) = \sqrt{[L_i - L_j]^2 + (u_i - u_j)^2 + (v_i - v_j)^2}
\]  

(1)

### 3.3 Invariant moments of shapes

The initial approach of moment is credited to Hu [6]. Several low-order central moments for a region can be used to be the shape descriptors of the region. Assume that an object is contained within a minimum boundary rectangle, \(R\), of size \(M \times N\) and \(f(x, y)\) stands for the color value in pixel \((x, y)\). The \(p, q\)’th moment of this region \(R\) is given by \(m_{pq}\),

\[
m_{pq} = \sum_{x=0}^{M} \sum_{y=0}^{N} x^p y^q f(x, y).
\]

The \(m_{00}\) is the total number of pixels in the object of \(R\). \(m_{10}\) and \(m_{01}\) are the centroids of object in \(x\)-axis and \(y\)-axis, respectively. The higher-order moments have their own geometric interpretations. However, the higher-order moments are not invariant to translation. To avoid matching error caused by the effect of translation like rotation and scaling. A translation of the origin to the centroid of the object called central moments is developed. Let \((\hat{x}, \hat{y}) = (m_{10}/m_{00}, m_{01}/m_{00})\), the normalizing central moment) \(\eta_{pq}\) yields

\[
\eta_{pq} = \frac{\left[\sum_{x=0}^{M} \sum_{y=0}^{N} (x - \hat{x})^p (y - \hat{y})^q f(x, y) \right]^{1/(p+q+1)}}{\left[\sum_{x=0}^{M} \sum_{y=0}^{N} f(x, y) \right]^{1/(p+q+1)}}.
\]

The set of invariant moments \(\{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7\}\) shown in the following is invariant to translation user rotations and scale changes:
\[
\phi_1 = \eta_{20} + \eta_{02}, \\
\phi_2 = (\eta_{20} - \eta_{02})^2 + 4 \eta_{11}, \\
\phi_3 = (\eta_{30} - 3 \eta_{12})^2 + (3 \eta_{21} - \eta_{12})^2, \\
\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{32})^2, \\
\phi_5 = (\eta_{30} - 3 \eta_{12})(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{32})^2], \\
\phi_6 = (3 \eta_{21} - \eta_{03})(\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{32})^2], \\
\phi_7 = (\eta_{11} - 3 \eta_{03})(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{32})^2] + \\
4 \eta_{11}^2 (\eta_{30} + \eta_{12})(\eta_{21} + \eta_{32})^2, \\
\phi_8 = (\eta_{11} - 3 \eta_{03})(\eta_{30} + \eta_{12})(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{32})^2].
\]

Here, we use the set of invariant moments \( \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7\} \) to be the shape descriptors. Assume that the shapes of two objects \( O_i \) and \( O_j \) are represented by the set of invariant moments: \( \{\phi_{i1}, \ldots, \phi_{ij}\} \) and \( \{\phi_{j1}, \ldots, \phi_{jj}\} \). The moment difference between objects \( O_i \) and \( O_j \) is defined as Equation (2).

\[
\delta_{st}(O_i, O_j) = \sqrt{\sum_{k=1}^{7}(\phi_{ik} - \phi_{jk})^2} \tag{2}
\]

### 3.3 Vector-based spatial matching

The 2D String, proposed by Chang in 1987 [Chang (1987)], is the most famous spatial representation for iconic images. Many related researches have been proposed to improve 2D String’s efficiency and effectiveness of spatial reasoning and similarity matching. Since the 2D String only supports similarity matching of three types and represents spatial relationships in symbolic strings, it is hard to be combined with the other numerical features like color and shape. For example, the image retrieval method proposed by Chang [Chang (1998)] uses the 2D String as its spatial matching scheme. Their spatial matching results using 2D String can not be evaluated and ranked with other similarity metrics together.

For merging various image features as a single ranking function, we modify the vector-based approach [Chien (1998)] as the spatial measuring method in our system. The principal characteristics of Chien's vector-based approach are:
- Supporting numerical notation of coordinates.
- Providing more powerful spatial reasoning function than 2D String.
- Providing efficient and effective spatial matching algorithms.

Assume that there are two objects \( O_i \) and \( O_j \) segmented from an image. Let \((px_i, py_i)\) and \((qx_i, qy_i)\) denote the bottom-left and top-right coordinates of minimum boundary rectangle(MBR) containing \( O_i \). \( Dx_{ij} \) and \( Dy_{ij} \) are defined as the vector from \( O_i \) to \( O_j \) on x-axis and y-axis: \( Dx_{ij} = px_j - px_i \) and \( Dy_{ij} = py_j - py_i \). Let \( O_i \) be the referential object, we define the following parameters:

\[
Ix_i = |qx_i - px_i|, \ Iy_i = |qy_i - py_i|; \\
\alpha_x = \frac{Dx_{ij}}{Ix_i}, \ \beta_x = \frac{Ix_i}{Dx_{ij}}, \ \alpha_y = \frac{Dy_{ij}}{Iy_i}, \ \beta_y = \frac{Iy_i}{Dy_{ij}}; \\
\gamma_x = \frac{\alpha_x + \beta_x}{\beta_x + 1}, \ \gamma_y = \frac{\alpha_y + \beta_y}{\beta_y + 1}.
\]
The relative distance between objects \( O_i \) and \( O_j \) on \( x \)-axis and \( y \)-axis, \( f(X_{ij}) \) and \( f(Y_{ij}) \), are defined as follows:

\[
f(X_{ij}) = \begin{cases} 
    \frac{1}{(1-\gamma_x)} & \text{if } \alpha_x + \beta_x < 0; \\
    (1 - \frac{1}{\gamma_x}) + 3 & \text{if } \alpha_x > 1; \\
    2\gamma_x + 1 & \text{otherwise} 
\end{cases}
\]

\[
f(Y_{ij}) = \begin{cases} 
    \frac{1}{(1-\gamma_y)} & \text{if } \alpha_y + \beta_y < 0; \\
    (1 - \frac{1}{\gamma_y}) + 3 & \text{if } \alpha_y > 1; \\
    2\gamma_y + 1 & \text{otherwise} 
\end{cases}
\]

For two distinct pictures \( P_1 \) and \( P_2 \), assume that they both consist of two objects \( O_i \) and \( O_j \), we have the relative distances \( f(X^n_{ij}) \), \( f(Y^n_{ij}) \), \( f(X^n_{ij}^t) \), \( f(Y^n_{ij}^t) \). The spatial difference between \( P_1 \) and \( P_2 \) for \( O_i \) and \( O_j \) is defined as Equation (3).

\[
\delta_s(O^n_{ij}, O^n_{ij}^t) = \sqrt{f(X^n_{ij}) - f(X^n_{ij}^t)}^2 + [f(Y^n_{ij}) - f(Y^n_{ij}^t)]^2.
\] (3)

After the feature extraction, the objects and corresponding features are extracted from images and can be represented by multi-dimensional vectors. We model an image as a triple:

\[ [\text{Image_id, Object_set, Feature_list}] \]

\( \text{Image_id} \) is the identity of an image, \( \text{Object_set} \) is the set of objects contained in the image, and \( \text{Feature_list} \) is the features of each object in the image, as \( \langle O_1; F_1 \rangle, ..., \langle O_n; F_n \rangle \), where the number of objects is \( n \). In our system, the features in \( \text{Feature_list} \) are

\[ F_i = [L, u, v] \land [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7] \land [LD_x, LD_y, RU_x, RU_y]. \]

\( [L, u, v] \) and \( [\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7] \) are the features of color and moments as described earlier. \( LD_x, LD_y, RU_x, RU_y \) are the bottom-left and top-right locations of object’s MBR. The above features will be able to be easily indexed by high-dimensional data structures such as k-d tree and R-tree to improve the performance of access.

### 4. Similarity matching and relevance feedback algorithms

#### 4.1 Similarity matching algorithms

The similarity measure of image features is an another important topic for image retrieval systems. A good measure can retrieve relevant images to satisfy users’ request effectively. Since the proposed system based on object features, the similarity measure includes two main matching algorithms. The first algorithm is to find the best matching of objects belonging to the query image and the target image respectively. The second algorithm then gives the overall measure of different features and similarity ranking of images.

We discuss the object-pair matching algorithm first. The goal of this algorithm is to find the best object-pairs whose objects belongs to the query image and the target image. The similarity of two objects is measured by their representative colors and shape descriptors, that is, the values of \( [L, u, v] \) and the set of seven invariant moments \{ \phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7 \}. Assume that there are \( m \) objects in query image \( P_1 \) denoted as \( O_1, ..., O_m \), and the target image \( P_2 \) in which contains \( n \) objects denoted as \( O_1, ..., O_n \). The best object-pairs are found by the following steps:
Step 1: Let the objects in \(P_1\) and \(P_2\) construct a matrix of size \(m \times n\) denoted as \(D_{mn}\). We have \(D_{mn} = [d_{ij}]_{m \times n}, 1 \leq i \leq m, 1 \leq j \leq n\), where \(d_{ij}\) is an element in the matrix which stands for the difference between \(i\)th object \(O_i\) in \(P_1\) and the \(j\)th object \(O_j\) in \(P_2\). The \(d_{ij}\) is defined as

\[
d_{ij} = w_c \Delta_c(O_i, O_j) + w_s \Delta_m(O_i, O_j)
\]

where \(\Delta_c(O_i, O_j)\) is the normalized color difference defined in Equation (1), \(\Delta_m(O_i, O_j)\) is the normalized moment difference defined in Equation (2), and \(w_c, w_s\) are the corresponding weights of color difference and moment difference, respectively.

Step 2: Let \(\min\{d_{ij}\}\) be the minimum element of \(D_{mn}\). If \(\min\{d_{ij}\}\) is found in \(i = a\) and \(j = b\), the object \(O_a\) in \(P_1\) and the object \(O_b\) in \(P_2\) will be grouped into an object-pair.

Step 3: Discarding the \(a\)th row and the \(b\)th column in \(D_{mn}\), we get a new matrix \(D_{(m-1)(n-1)}\). The Step 2 is repeated by the new matrix until all objects in either picture \(P_1\) or \(P_2\) have all been matched or the remaining differences in \(D_{(m-1)(n-1)}\) are larger than a specified value.

We give the initial values of weights \(w_c = w_s\) and \(w_c + w_s = 1\), where \(w_c\) and \(w_s\) are in \([0, 1]\). The weights will be adjusted automatically by the mechanism of relevance feedback. We will discuss the relevance feedback in the next subsection. The algorithm of object-pairs matching is shown in the following:

**Algorithm: object-pairs matching algorithm**

Input: the color and moment features of pictures \(P_1\) and \(P_2\)

Output: the set of object-pair

```{ for i = 1 to m
    for j = 1 to n
        compute \(D_{mn}\), \(d[i,j] = w_c \Delta_c(O_i, O_j) + w_s \Delta_m(O_i, O_j)\);
    object-pair = \(\emptyset\);
    \(k = 0\);
    \(d[a,b] = \min\{d[i,j] | 1 \leq i \leq m, 1 \leq j \leq n\}\);
    While \((k < \min(m,n))\) and \((d[a,b] \leq \text{MAX\_DIS})\) // MAX\_DIS is a specified
        \{ // maximum value of difference
            object-pair = object-pair \cup \{(O_a,O_b)\};
            for j = 1 to n
                \(d[i,j] = \infty\);
            for i = 1 to m
                \(d[i,b] = \infty\);
            \(k = k + 1\);
            \(d[a,b] = \min\{d[i,j] | 1 \leq i \leq m, 1 \leq j \leq n\}\);
        }
    }
```

After object-pairs are found, the similarity of two pictures is measured by the similarity matching algorithm. In addition to the features of color and shape, the similarity matching algorithm also considers the spatial relationships among the object-pairs.

Step 1: Assume that the object-pairs matching algorithm finds \(k\) object-pairs for the given two pictures \(P_1\) and \(P_2\). The set of object-pairs is \(\{(O_1^1,O_2^1),(O_1^2,O_2^2),\ldots,(O_1^k,O_2^k)\}\).

Step 2: The dissimilarity between pictures \(P_1\) and \(P_2\), denoted \(\text{Dis}(P_1,P_2)\), is computed by the Equation (5).
\[ \text{Dis}(P_1, P_2) = W_c \psi_{\text{color}}(P_1, P_2) + W_s \psi_{\text{shape}}(P_1, P_2) + W_{sr} \psi_{\text{spatial}}(P_1, P_2) \]

where

\[ \psi_{\text{color}}(P_1, P_2) = \sum_{i=1}^{k} \Delta_C(O^c_i, O^c_j), \]

\[ \psi_{\text{shape}}(P_1, P_2) = \sum_{i=1}^{k} \Delta_M(O^m_i, O^m_j), \]

\[ \psi_{\text{spatial}}(P_1, P_2) = \sum_{i=1}^{k} \sum_{j=1}^{k} \Delta_S(O^s_i, O^s_j). \]

The \( \Delta_C(O_i, O_j) \) is the normalized color difference and the \( \Delta_M(O_i, O_j) \) is the normalized moment difference defined as the object-pairs matching algorithm. The \( \Delta_S(O_i, O_j) \) is the spatial difference equal to the \( \delta_S(O_i, O_j) \), since the \( \delta_S(O_i, O_j) \) has been normalized itself. The values of weights \( W_c, W_s \) and \( W_{sr} \) are all equal initially and \( W_c + W_s + W_{sr} = 1 \), where \( 0 \leq W_c, W_s, W_{sr} \leq 1 \). Then the weights will be also changed by the same relevance feedback mechanism described in the next subsection. We use the degree of dissimilarity, \( \text{Dis}(P_1, P_2) \), to determine the relevant degree of the similar images in the image database. The smaller values of \( \text{Dis}(P_1, P_2) \) is, the more similar the two pictures \( P_1 \) and \( P_2 \) are. The algorithm of similarity matching is shown in the following:

**Algorithm: similarity matching algorithm**

Input: the features of pictures \( P_1 \) and \( P_2 \) including color, moments and spatial locations

Output: the degree of dissimilarity between \( P_1 \) and \( P_2 \)

\{ 
object-pairs = object-pairs-matching(the color and moments of \( P_1 \) and \( P_2 \));
\]

\[
\text{k} = |\text{object-pairs}|; \quad // \text{the number of elements in the set of object-pairs}
\]

for \( i = 1 \) to \( k \)

\[
\text{compute } \psi_{\text{color}}(P_1, P_2), \psi_{\text{shape}}(P_1, P_2) \text{ and } \psi_{\text{spatial}}(P_1, P_2);
\]

\[
\text{Dis} = W_c * \psi_{\text{color}}(P_1, P_2) + W_s * \psi_{\text{shape}}(P_1, P_2) + W_{sr} * \psi_{\text{spatial}}(P_1, P_2);
\]

\}

4.2 Relevance feedback mechanism

In the proposed system, the relevance feedback mechanism is employed in the matching of object pairs and the similarity matching of images. The approaches of updating weights used in matching object pairs and measuring similar images are the same except the number of input parameters. We try to match the best object-pairs by updating the weights of low-level features in individual object. And the global weights are used to capture the user’s high-level concept about the information need of color, shape or spatial on objects according to the user’s feedback. Unlike the other relevance feedback approaches, in our system, the weight updating is based on the dissimilarity instead of similarity. The relevance feedback is proceeded as the following steps:

Step 1: The similarity matching algorithm finds the similar images from databases and lists the images in the order of similarity for a given query image. The user can restrict the maximum number of retrieved images.

Step 2: For each retrieved image, the user marks whether it is relevant or not.

Step 3: The system updates the weights including the low-level features in objects and the global features of images according to the user’s feedback in Step 2.

Step 4: Go to Step 1 and start a new retrieval until the user satisfies the retrieval results.

In general, there are two ways to browse and mark the relevant images. One is that users give a threshold to retrieve and rank the similar images that are above a specified degree of similarity. The other way is to restrict the maximum number of images by the ranking of
retrieval results. The proposed approach in Step 1 uses the second method. The update of weights in Step 3 is described in the following. Assume that the number of features used in the similarity measure is $K$. Let $\omega_i'$ be the weight of the $i$th feature, where $0 \leq \omega_i' \leq 1$, $\omega_1'+\omega_2'+\ldots+\omega_K'=1$; and $N_i$ be the number of images marked by the user at the $t$th feedback time. $D^t_j$ is the difference measured by the $i$th feature of the $j$th marked relevant image at that time. We give the same values to the weights of different features initially while the first measuring is made. That is,

$$\omega_i^0 = \frac{1}{K}, \quad \text{for } 1 \leq i \leq K.$$

After the images are retrieved by the initial weights, the user marks $N_t$ relevant images from the retrieval results. The update of weights is considered to be the ratio of relative difference of different features. The adjusting functions are as follows:

$$\sigma_i^{t+1} = 1 - \frac{\sum_{j=1}^{N_t} D^t_j / \sum_{j=1}^{K} \sum_{i=1}^{K} D^t_j}{\sum_{i=1}^{K} \sigma_i^{t+1}},$$

$$\omega_i^{t+1} = \frac{\sigma_i^{t+1}}{\sum_{i=1}^{K} \sigma_i^{t+1}}.$$

Since $D^t_j$ represents the dissimilarity, the ratio has to be subtracted by one. Finally, the weights should be less than or equal to one and the summation equals to one.

Applying above adjusting function to the object-pairs matching algorithm will effect the matching of object-pairs between images. It can help system to capture the significant objects that are interesting to the user. Then the update of weights used in the similarity image matching algorithm is used to decide the global importance of object features, since $\psi_{\text{color}}$, $\psi_{\text{shape}}$, and $\psi_{\text{spatial}}$ have summed the differences of independent features among objects. By using the technique of relevance feedback to adjust the weights of different features, the relevant images concerned by the user will be mostly retrieved in several times of feedback processes.

5. Experimental results

For demonstrating the efficiency and effectiveness of the proposed approach, we build a prototype system on PC with Windows 98. The total number of tested images is about 2500. Types of the images include about 1000 images of photographs with the same size of 256×256 and about 1500 images of trademark with different size. First, the test images are segmented and the features are extracted by the systems automatically. Then the feature database is created and query processes can be started.

The experimental results are shown in Figure 3, Figure 4 and Figure 5. Our system can accept a picture image or a sketch to be the query image. Here, we give the query images by making a sketch. For the first query image shown in Figure 4(a), three objects with different color and shape are given. The circle is yellow, the triangle is blue, and the thin rectangle is pink. We first compare our system with Mehtre's [Mehtre (1998)], the result is as Figure 3. In this experiment, the weights of $W_c$, $W_s$, and $W_o$ are directly set to 0.2, 0.4 and 0.4, respectively; and the parameters of Mehtre's system use the values recommended in [Mehtre (1998)]. Figure 3 shows that our system retrieves the relevant images in a better ranking order correctly. However, the ranking order of Mehtre's system can not reflect the human requirements on the spatial relationship of objects. In our system, of course, the decision of the weights is the key of the system. For solving the human subjectivity, the best weights can be found by the processing of user's feedback. The maximal number of images browsed is set to seven. The results of retrieval and relevance feedback are shown in Figure 4(b). For the initial retrieval, the weights are 0.5 in
object layer and 0.333 in the image layer, respectively. The relevant images are the images ranked 1, 2, 3, and 6. We mark these relevant images and start the first retrieval of relevance feedback. The result appears a new image with ranked 7 that is not contained in the top seven images in initial retrieval. The weights of the image layer used in this measure are shown in Table 1. And then, the second relevance feedback is proceeded by marking the image ranked 1, 2, 3, 5, and 7. We find that the final result will list all relevant images on the top of ranking. Since the color of thin rectangle in target images are not pink but red, the final weights in the image layer reflect this characteristics. The second example is shown in Figure 5. The sketch given in Figure 5(a) is a black ellipse with an orange ellipse inside. It looks like a black collar around the orange ellipse. We demonstrate the retrieval of photographs in this example. The initial retrieval obtains many images with different high-level concepts. The relevant images we retrieved are the images ranked 3 and 6. After marking the two images, the result of relevance feedback has been able to rank the images with similar content to the top of ranking list. The other images list on the result have similar color distribution mostly, because the weights of similarity measure are updated into about 2:1:1 in color:shape:spatial relationship order.

In the experiments, most of the image queries can get the relevant images that is the user want to in two or three relevance feedback processes. In general, the reason that we can not find the similar image to fit the user’s request is no such relevant image with similar content in the image database. In some cases, we miss the relevant images because the user’s conceptual level is higher than the level of object. For example, the user may give a picture with a rose to retrieve the same kind of rose. However, the system cannot recognize whether the flower on a picture is a rose or not. Actually, even a general user cannot distinguish the different kinds of rose except the expert. Thus, combining higher conceptual semantics with our system is the research direction of future.

6. Conclusions

Content-based image retrieval is one of the most important techniques for retrieving multimedia data. It is difficult for researchers to create the so-called fully automatic image retrieval to capture human perception. The previous researches proposed the interactive retrieval approach focusing on the different low-level image features and similarity metrics. In this paper, we motivate an object-based model and integrate the system with interactive retrieval techniques to reduce the gap between high-level human concept and the low-level image features. The proposed approach segments an image into feasible regions first. We take each region as an object, then the features in the object and the relations among different objects are extracted. The similarity measure algorithms are given to match object-pairs and measure the similarity between images. Finally, the relevance feedback mechanism is developed to adjust the weights of measure functions in the two proposed algorithms. The experimental results show us that the proposed approach is effective and efficient. On expanding the object model with visual perception to integrate with human conceptual semantics is the future work. The interpretation of linguistic terms based on fuzzy set theory is one of the possible solutions.
Table 1: The weights of the retrieval in query image 1.

<table>
<thead>
<tr>
<th>( t )th feedback</th>
<th>Weights</th>
<th>( W^{t'}_c )</th>
<th>( W^{t'}_i )</th>
<th>( W^{t'}_{w^t} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.333</td>
<td>0.333</td>
<td>0.333</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.500</td>
<td>0.167</td>
<td>0.333</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.225</td>
<td>0.500</td>
<td>0.275</td>
<td></td>
</tr>
</tbody>
</table>
Figure 5: (a) The query image 2; (b) The retrieval results of query image 2.

Table 2: The weights of the retrieval in query image 2.

<table>
<thead>
<tr>
<th>t' th feedback</th>
<th>Weights</th>
<th>( W_{c}^{'} )</th>
<th>( W_{s}^{'} )</th>
<th>( W_{sr}^{'} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.333</td>
<td>0.333</td>
<td>0.333</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.488</td>
<td>0.268</td>
<td>0.244</td>
<td></td>
</tr>
</tbody>
</table>

References


