# An Efficient and Flexible Matching Strategy for Content-based Image Retrieval

Mann-Jung Hsiao [1,*], Yo-Ping Huang [2], Tienwei Tsai [3], Te-Wei Chiang [4]

*[1]Department of Computer Science and Engineering, Tatung University, Taipei 104, Taiwan; [2]Department of Electrical Engineering, National Taipei University of Technology, Taipei 106, Taiwan; [3]Department of Information Management, Chihlee Institute of Technology, Taipei County 220, Taiwan; [4]Department of Accounting Information Systems, Chihlee Institute of Technology, Taipei County 220, Taiwan*

hsiao@mis.knjc.edu.tw

Received February 2, 2008

**Abstract**

With the rapid growth of multimedia applications and digital archives, content-based image retrieval (CBIR) has emerged as an important area and received lots of attentions for the past decades. In practice, there are two major problems raised by a CBIR system: the feature descriptions of images and the expressions of users' search preferences. To tackle these problems, we present a DCT-based feature descriptor coupled with an efficient way of expression for users' preferences in this paper. Our approach partitions images into a number of regions with fixed absolute locations. Each region is represented by its low-frequency DCT coefficients in the YUV color space. Two policies are provided in the matching procedure: local match and global match. In the local match, the user formulates a query by selecting the interested region in the image. Candidate images are then analyzed, by inspecting each region in turn, to find the best matching region with the query region. For those query images without clear objects, the user can select the option "global match" instead. The experimental system shows that this approach is generally effective and particularly suited for images with interested regions having features which significantly differ from the global image features. With the help of friendly GUI, our system also allows users of any experience level to effortlessly get interested images from database. [Life Science Journal. 2010; 7(1): 99 – 106] (ISSN: 1097 – 8135).

**Keywords:** Content-based image retrieval; region of interest; region-based image retrieval; discrete cosine transform.

## 1. Introduction

With the explosive growth of commerce on the Internet, businesses are advancing digital imaging even more swiftly. As digital images quickly increase in number, researchers are continually developing improved access methods to retrieve images from a large database. Generally, image retrieval procedures can be roughly divided into two approaches: annotation-based image retrieval (ABIR) and content-based image retrieval (CBIR). In ABIR, images are often annotated by words, such as time, place, or photographer. To access the desired image data, the seeker can construct queries using homogeneous descriptions, such as keywords, to match these annotations. Although ABIR potentially offers the most accurate information when images are well-named or annotated, it still has the following drawbacks[1]. First, manual image annotation is time-consuming and therefore costly. Second, human annotation is subjective. Furthermore, some images could not be annotated because it is difficult to describe their content with words. Research in the area of image databases also shows that text, specifically provided to facilitate image retrieval, is inadequate for conducting searches[2]. Even an enormous amount of descriptive text cannot adequately substitute for the content of the image itself. Unlike ABIR, CBIR tends to index and retrieve images based on their visual content. CBIR avoids many problems associated with traditional ways of retrieving images by keywords. Thus, a growing interest in the area of CBIR has been established in recent years.

CBIR is a complex and challenging problem spanning diverse algorithms all over the retrieval processes including color space selection, feature extraction, similarity measurement, retrieval strategies, relevance feedback, etc. Some general reviews of CBIR techniques can be found in the literature [3-7]. Smeulder et al. reviewed more than 200 references in this field[3]. Datta et al. studied 120 of recent approaches[4]. Veltkamp et al. gave an overview of 43 content-based image retrieval systems[5]. Deselaers et al. presented an experimental comparison for a large number of different features[6]. Liu et al. provided a comprehensive survey of the recent technical achievements in high-level semantic-based image retrieval[7]. Although various CBIR techniques have been established and good performance results were demonstrated, there are still many problems not satisfactorily solved. In order to improve the retrieval performance of a CBIR system, we propose an efficient and flexible matching strategy that employs a DCT-based feature descriptor coupled with an efficient way of expression for users' preferences.

The feature extraction method is highly essential in CBIR. Being an elementary process, the feature extraction will be invoked very frequently; therefore, it should be time-efficient and accurate. Some transform type feature extraction techniques can be applied to reduce the dimension of the vector in representing an image, such as wavelet, Walsh, Fourier, 2-D moment, DCT, and Karhunen–Loeve. Among these methods, the DCT is used in many compression and transmission areas, such as JPEG, MPEG and others. Due to the superiority in energy compacting property, the DCT is often used to extract dominant features of images. Through the extraction of the low-frequency DCT coefficients, the goal of reduction of dimensionality in feature space can be achieved. In our approach, an image is first converted to the YUV color space and then transformed into DCT coefficients for each image (or region). Psycho-perceptual studies have shown that the human brain perceives images largely based on their luminance value (i.e., Y component), and only secondarily based on their color information (i.e., U and V components). Therefore, only the Y component is considered in our approach, so as to reduce the computation cost. That is, only a block size of 4x4 DCT coefficients in the upper-left corner constitutes the feature vector of an image (or a region).

Obtaining the semantics or the meaning of an image is another important issue in CBIR. Visual feature alone is not enough to distinguish between images. Two semantically different images can have very similar visual contents. For example, indoor images have often similar red color as sunset images. Partitioning or segmenting the image into regions may reveal the "true" objects within an image. To look at the objects (or regions) in the image, instead of looking at the image as a whole, is a way to obtain the semantic of an image, which is known as region-based image retrieval (RBIR)[8]. It contributes to more meaningful image retrieval, however, the segmentation algorithms are complex and computation intensive and the segmentation results are often not correct. To solve this, some approaches break images into a fixed number of regular rectangular regions. Rudinac et al. partitioned images into 4x4 non-overlapped regions and 3x3 overlapped regions[9]. It is expectable that using more regions better results may be produced but the execution speed becomes unsatisfactory slow for a large database. Amir et al. divided images at a coarser granularity level in the IBM TRECVID video retrieval system, using a fixed 5-region layout (4 equal corner regions and an overlapping center region of the same size)[10]. Instead of pursuing sophisticated segmentation methods, we extend the Amir's spatial layout of regions to solve the problem stated above. In our CBIR system, the images are first segmented into regular regions. The user can select the region of interest (ROI) in the query image to express his/her intentions. Candidate images are then analyzed, by inspecting each region in turn, to find the best matching region with the query region. In other words, the distance between the query image and a candidate image is the smallest distance between the query region and five regions within the candidate image. The experiment results will verify that the selection of ROI is very important in CBIR.

To further allow users to express their query specifications, the feature vector is categorized into four groups to represent its average grayness and three directional texture characteristics: vertical, horizontal and diagonal. It is suitable to find out an optimal set of weights; each weight emphasizes an individual feature. However, the weights are strongly depends on the query image and the user's personal perceptions or query intentions. Therefore, a friendly user interface is employed for users to express their personal view of perceptual texture properties for the ROI. The experimental system shows that with the help of weights the performance is further improved.

This paper is organized as follows. The next section introduces color space selection and feature vector construction. Section 3 discusses related issues about the region of interest and segmentation. The similarity measurement is presented in Sec. 4 and the performance evaluation method used is given in Sec. 5. Section 6 presents experimental results. Finally, conclusions are drawn in Sec. 7.

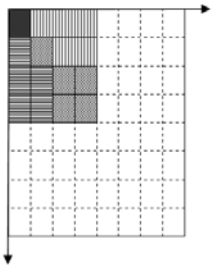


Figure 1. The upper left DCT coefficients used in our approach: (a) ■ DC, (b) ▥ vertical texture feature, (c) ▤ horizontal texture feature, and (d) ▩ diagonal texture feature.

## 2. Feature Extraction

### 2.1 Color Space

A color space is a method by which we can specify, create and visualize color. Different color spaces are better for different applications [11], for example some equipment has limiting factors that dictate the size and type of color space that can be used. Some color spaces are perceptually linear, i.e. a one-unit change in stimulus will produce the same change in perception wherever it is applied. Many color spaces, particularly in computer graphics, are not linear in this way. Among these color spaces, the YUV color space is a bit unusual. The vast majority of DVDs already store information in the YUV color space. The Y component determines the brightness of the color (referred to as luminance or luma), while the U and V components determine the color itself (the chroma). Y ranges from 0 to 1 (or 0 to 255 in digital formats), while U and V range from -0.5 to 0.5 (or -128 to 127 in signed digital form, or 0 to 255 in unsigned form). Some standards further limit the ranges so the out-of-bounds values indicate special information like synchronization.

In our approach, the YUV color space is used for two reasons: 1) efficiency and 2) ease of extracting the features based on the color tones. One another neat aspect of YUV is that you can throw out the U and V components and get a grey-scale image. Since the human eye is more responsive to brightness than it is to color, many lossy image compression formats throw away half or more of the samples in the chroma channels to reduce the amount of data to deal with, without severely destroying the image quality. Therefore, only the Y component is used in our preliminary study. There are many slightly different formulas to convert between YUV and RGB. The only major difference is a few decimal places. The following equations are used to convert from RGB to YUV spaces:

$$Y(x,y) = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y), \quad (1)$$

$$U(x,y) = 0.492(B(x,y) - Y(x,y)), \text{ and} \quad (2)$$

$$V(x,y) = 0.877(R(x,y) - Y(x,y)). \quad (3)$$

### 2.2 Feature vector

The fundamental problem for CBIR systems is to extract features for every image and to define a similarity measure for comparing images. The features serve as an image representation in the view of a CBIR system. Image contents can be defined at different levels of abstraction. At the first lowest level, an image is a collection of pixels. Pixel level content is rarely used in retrieval tasks. The raw data can be processed to produce numeric descriptors capturing specific visual characteristics called features. The most important features for image databases are color, texture and shape. In general, a feature-level representation of an image requires significantly less space than the image itself. Some transform type feature extraction methods can be applied to reduce the number of dimensions, such as Karhunen-Loeve (KLT), discrete Fourier transform (DFT), discrete cosine transform (DCT), and discrete wavelet transform (DWT), etc. Among these methods, DCT has been known for its excellent energy compacting property. It has received a great deal of attention and is widely used in image compression. For most images, most significant DCT coefficients are concentrated around the upper left corner; the significance of the coefficients decays with increased distance.

After an image is converted to the YUV color space, it is equally divided into four rectangular regions and one additional central region. Then the DCT is performed over the Y component for a whole image (global features) and five regions (regional features). Details on the segmentation will be explained in the next section. Eventually, an image is represented by one global feature and five regional features, each of which is constituted by a block of 4x4 DCT coefficients. As a result, only 96 DCT coefficients are needed for each image.

Note that the image is inherently a subjective medium and it might contain multiple concepts, that is, the perception of image content is very subjective, and the same content can be interpreted differently. For example, one user might be more interested in a different dominant feature of the image than the other, or two users might be interested in the same feature (e.g., texture), but the perception of a specific texture might be different for the two users. To solve this problem, the feature vector is categorized into four groups: the DC coefficient ($V_1$) represents the average energy of the image and all the remaining AC coefficients contain three directional feature vectors: vertical ($V_2$), horizontal ($V_3$), and diagonal ($V_4$), as shown in Figure 1. The user can give different weight for each texture feature based on their perceptions for a query.

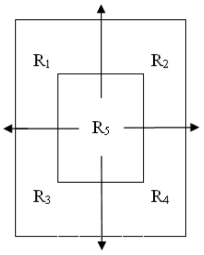


Figure 2. The five rectangular regions used in our approach.

## 3. Region of Interest and Segmentation

Another problem for CBIR systems is that users' intentions are diverse. For example, if an image with cloudy blue sky over green grass is used as a query, one might be seeking images with cloudy blue sky as the main theme whereas another might be seeking images largely containing green grass. This issue will be examined in our experiments later. Within this great diversity, it is suggested that a CBIR system should assist users in expressing their intentions behind the query. This leads to a number of solutions that do not treat the image as a whole, but rather deal with regions within an image[12,13]. Partitioning or segmenting the image into regions may reveal the "true" objects within an image, thus contributing to more meaningful image retrieval. The interested object/region in a query is commonly defined as region of interest (ROI). For the CBIR task that the user is only interested in a portion/region of an image, it is defined as localized content-based image retrieval[13] or region-based image retrieval (RBIR)[8].

### *A. Region of interest*

In general, existing CBIR can be categorized into two major classes, namely, global methods and localized methods[12]. Global methods exploit features from the whole image and compute the similarity between images while local methods extract features from a region (portion) of an image and compute the similarity between regions. In RBIR, an image has to be segmented into regions before indexing and retrieval. However, it is hard to locate the ROI in the image when the interested object/region occupies only a small part of the image because the image background can have dominant impact on the feature extraction. The most direct way to solve the problems is to let the user select a ROI while conducting a query. As far as the target region is selected, the concept in the query image is expressed more accurately.

### *B. Segmentation*

There have been many automatic segmentation algorithms proposed in RBIR. For example, SIMPLIcity[14] and WALRUS[15] are both wavelet-based and region-based CBIR systems. An obvious drawback of such systems is that the segmentation algorithms are complex and computation intensive and the size of the search space is sharply increased due to exhaustive generation of subimages. In addition, the retrieval performance is affected significantly because the segmentation results are often incorrect. Therefore, the automatic detection of subimages remains an elusive task. In our work, we resort to a rectangular segmentation method, breaking images into a fixed number of regular rectangular regions. It seems that the use of more regions achieves a better result. However, it is not always the case and the execution speed becomes unsatisfactory slow for a large database.

The first step in our retrieval technique is to segment the image into regions that (ideally) would be easy for users to select their ROI. For this purpose, we need a segmentation method that is effective in rendering homogeneous regions in a short time. Figure 2 illustrates the five rectangular regions segmented in our approach. They are obtained by dividing the image into four non-overlapping regions and one central region with the same size as others, which is similar to the layout of the IBM TRECVID video retrieval system[10]. To avoid noise during the local match, our system allows users to select the ROI from the segmented five regions and only this region is then compared with regions in other images in the collection.

## 4. Similarity Measurement

In CBIR systems, image are in general represented and organized into *n*-dimensional feature vectors. Following to image representation, similarity measure is one of the key items in the process of image retrieval that affects the effectiveness and the efficiency of the retrieval technique. In practice, it is hard to define a distance between two sets of feature points such that the distance could be sufficiently consistent with a person's concept of semantic closeness of two images. Therefore, there are very few theoretical arguments supporting the selection of one distance over the others; computational cost is probably a more important consideration in the selection.

During the retrieval process, the query image and the database images are compared by evaluating the distance between their corresponding feature vectors. To exploit the energy preservation property of DCT, we use the sum of squared differences (SSD) as the distance function. Using a simpler distance on lower dimensional features means that computation can be saved both in the evaluation of distance and in the number of comparisons to be performed. For the local match, similarity measurement is performed based on region similarity. For the global match, the whole image is regarded as a "large" region. The distance function is defined as follows.

Overall region visual similarity relies on the combination of the four following discriptors: DC($V_1$), vertical texture ($V_2$), horizontal texture ($V_3$), and diagonal texture ($V_4$). Let $Q$ and $X$ denote the query image and a database image, respectively. $V_k$ is the $k$-th feature vector of an image or a region (In our approach, $k$ = 1 to 4). $C_n$ is a vector component in $V_k$. Assume the distance $d_k$ is the distance between the $k$-th feature vector $V_k^q$ of the ROI (e.g., the $i$-th region $R_i^q$) in $Q$ and the $k$-th feature vector $V_k^x$ of the $j$-th region $R_j^x$ in $X$. Then,

$$d_k = d(R_i^q, R_j^x) = \sum_{C_n^q \in V_k^q, C_n^x \in V_k^x} (C_n^q - C_n^x)^2. \quad (4)$$

Similarity is evaluated as a weighted aggregation of image features. $w_k$ is the weight assigned to the $k$-th feature vector in a query to express its importance. Thus the overall distance between two regions is:

$$D(R_i^q, R_j^x) = \sum_{k=1}^{4} w_k d_k. \quad (5)$$

When the user selects an interest region $R_i^q$ in the query image $Q$ and issues a query, candidate images are hence analyzed, by applying equations (4) and (5) for each region in turn, to find the best matching region in an image $X$, which having a smallest distance with the query region. The distance between $Q$ and $X$ can thus be defined as:

$$D(Q, X) = \underset{j=1\, to\, 5}{Min}(D(R_i^q, R_j^x)). \quad (6)$$

In general, the conventional computer vision recognition-based task looks for the object to be searched with as small and as accurate a retrieved list as possible. But in CBIR, the goal is to extract as many "similar" objects as possible, the notion of similarity being very loose as compared to the notion of exact match.

## 5. Performance Evaluation

CBIR aims at searching image databases for specific images that are similar to a given query image. Many CBIR systems are available but it is difficult to assess which of these systems is the best as it is not possible to compare CBIR systems quantitatively and objectively. One reason for this is the absence of a standard database to determine a standard set of quantitative performance measures.

There are two commonly used performance measures in CBIR: the precision rate $p$ and the recall rate $r$, which are adapted from textual information retrieval. They are defined as

$$p_M = \frac{n_r}{M}, \text{ and} \quad (7)$$

$$r_M = \frac{n_r}{N_r}, \quad (8)$$

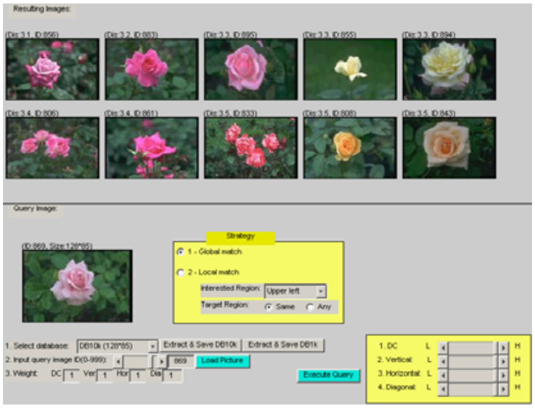
where $M$ is the total number of retrieved images, $N_r$ is the total number of relevant images in the database, and $n_r$ is the number of relevant images retrieved. It is assumed that the user inspects the $m$ first images of the ranked list. So for every $m$ = 1, 2, …, $M$, precision $p_m$ and the recall $r_m$ are calculated. Note that considering one additional image may raise the precision or lower it.

This process can be interpreted as looking only at the locally optimal answer sets for which recall and precision cannot be improved simultaneously by inspecting further images.

In practice, when the number of relevant images is greater than the size of the answer list, recall is meaningless as a measure of the retrieval quality. To overcome this problem, a measure called efficacy is introduced[16]:

$$\eta_M = \begin{cases} n_r / N_r \text{ , } if \ N_r \le M \text{ ,} \\ n_r / M \text{ , } if \ N_r > M \end{cases} \tag{9}$$

where $\eta_M$ is called the efficacy of retrieval for a given short-list of size *M*. If $N_r \le M$, $\eta_M$ becomes the traditional recall of information retrieval; if $N_r > M$, $\eta_M$ is indeed the precision of information retrieval.


Resulting Images:

(Dis:3.1, ID:856) (Dis:3.2, ID:883) (Dis:3.3, ID:895) (Dis:3.3, ID:855) (Dis:3.3, ID:894)

(Dis:3.4, ID:806) (Dis:3.4, ID:861) (Dis:3.5, ID:833) (Dis:3.5, ID:808) (Dis:3.5, ID:843)

Query Image:

(ID:869, Size:128*85)

Strategy

● 1 - Global match

○ 2 - Local match

Interested Region: Upper left

Target Region: ● Same ○ Any

1. Select database: DB10k (128*85)

2. Input query image ID(0-999): 869

3. Weight: DC 1 Ver 1 Hor 1 Dia 1

Figure 3. The system interface, browsing a query's result.

## 6. Experimental Results

In order to implement, test, and validate our approach, we develop a CBIR system with a general-purpose image database including 1,000 color images, which was downloaded from the WBIIS database[17]. Five categories of images, including eagles, brown animals, cloudy sky, mountain scene, and flowers were selected as benchmark images; each of these images was used in turn to query the image database. The returned images are considered relevant (or hit) once they appear in the same category to the query. Unlike recognition-based systems, CBIR systems require versatility and adaptation to the user, rather than the embedded intelligence desirable in recognition tasks. Therefore, design efforts in our CBIR system are devoted to combine light computation, great flexibility and friendly user interface such that the system is evaluative enough to let users conduct a query, modify parameters, and browse the results for benchmarking.

During the query process, it's hard to guess what object is the target of users. In our system, users can specify a whole image (global match) or a particular ROI (local match) as a query. Figure 3 is the interface of our system, where the user can specify the ROI, adjust the weight of each feature, and inspect the retrieved results. The ROI capabilities in the system, allowing the expressions of interested region in the query image, are highly appealing to capture a certain level of semantics and can be used much in the same way as words. For example, the system will accept queries like "Find me all the images containing the contents in the upper left region of the query image." To accomplish this, the user can select any one of the regions (upper left, upper right, lower left, lower right, and center) for local match (or regional match). For those query images without clear objects, the user can select the option "global match" to conduct the query. For a single region query (i.e., regional match), there are two options" "same" and "any", which means the ROI of the query image is compared with the "same" or "any" region of the candidate images.

After the user loads a query image and selects the ROI, the system first initializes a set of uniformly distributed weights for features (i.e., the default weights). Then the user's specific information needs can be described by adjusting the weight of each feature. To achieve the best possible results, the feature vector of the ROI is compared with the feature vectors of all predefined regions of candidate images. Through exhaustive searches, the top ten in similarity are displayed to the user, ranked in the ascending order of the distance to the query image from the left to the right and then from the top to the bottom. Note that the similarity is computed on the basis of the default or adjusted weights, modeling what users see when they look at the query image.

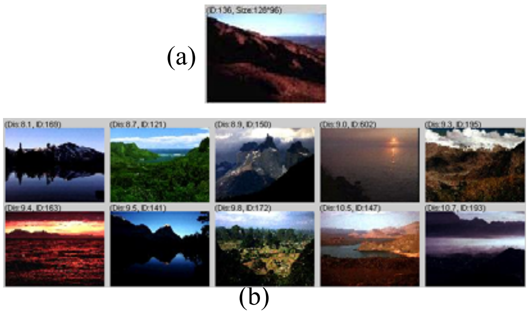
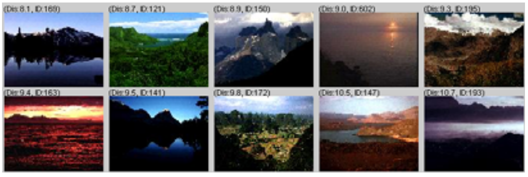
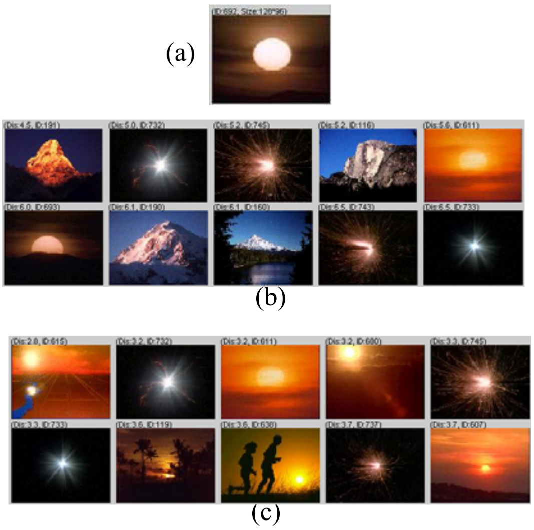


Figure 4. (a) The query image; (b) retrieval results using global match.

Several queries are first given for better illustrating the use of global match and local match. For the first query, an image of a mountain scene is given as the query image, shown as Figure 4(a). Since no obvious object/region appears in the image, the user can not pick the region which is perceptually meaningful as the ROI. Thus, the user selects the option "global match" to conduct the query. Figure 4(b) gives a very promising result. In the second query image, as shown in Figure 5(a), there is a round bright object in the center, surrounded by darkness and a faint glow. Figure 5(b) shows the results using the global match while Figure 5(c) shows the results using the local match with the option "any". It is observed that local match explores more interesting images than global match, which returns images with shinning objects at other regions. Another example of local match often happens in the real world.



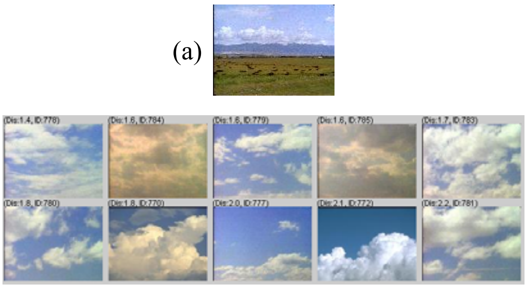


(a)

(b)

(c)

Figure 5. (a) The query image; (b) retrieval results using global match; (c) retrieval results using local match (options: "center" and "any").



(a)

(b)

Figure 6. (a) The query image; (b) retrieval results using local match (options: "upper left" and "any")

If a user wants to find out images with cloudy blue sky, but only has an image like Figure 6(a) in hand, he/she can selects the ROI that contains his/her target object. It can be seen that the result in Figure 6(b) matches the user's needs, without containing grass "objects" as in the query image.

A series of experimental results are collected to show the feasibilities of our solution. The next experiment examines the appropriateness of the feature vector (i.e., Y-component) and weights adjustment. The benchmark query targets are manually picked from five commonly accepted categories. Tables 1 to 5 show all the query results; each for one individual category. Table 6 summarizes the search results for all of the categories. It can be noticed that the average efficacy is improved from 72.38% to 79.05% if the proper weights are given. Some of the query examples even result in great improvements by giving different weights. For instance, while using ID 240 as the query image, the number of the retrieved relevant images increases from 5 to 9 if the weights adjusted from (1,1,1,1) to (0,1,1,1). Generally speaking, in the query process, users can adjust the weights through a set of experiments, with the goal of learning which features are most likely to contribute to the query.

The following experiment verified the necessity of ROI. We purposely choose ten queries with poor performance on the global match, compared with the same queries on the local match. Table 7(a) shows the retrieval results for global match, where the queries do not perform ROI selection and use the whole image for feature construction. In the local match, as shown in Table 7(b), users can select their ROI with the same weights (i.e., default weights) as those in the global match. It is obvious that ROI selection makes great improvement raising efficacy from 44% to 56%. Next, to further evaluate the effectiveness of the weight adjustment, we conduct queries with the same ROI selections but different weight settings. From Table 7(c), it can be seen that the weight adjustment can improve the discrimination ability as in the experiment 1, raising efficacy from 66% to 81%.

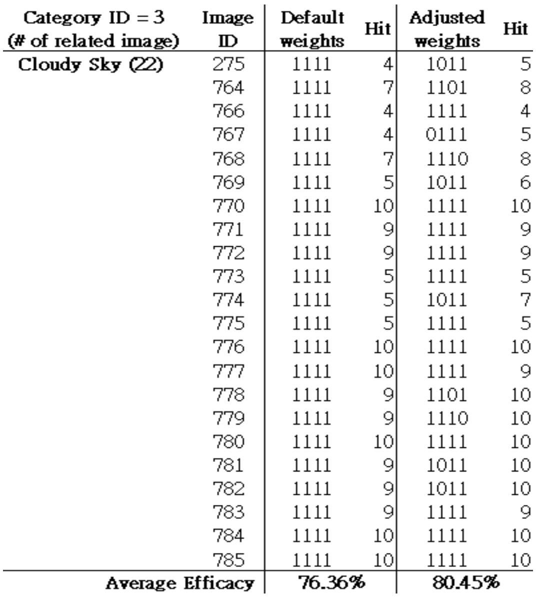
Table 1. The average efficacy of global match for category 1.

| Category ID = 1<br>(# of related image) | Image ID | Default weights | Hit | Adjusted weights | Hit |
|---|---|---|---|---|---|
| Eagle (9) | 275 | 1111 | 4 | 1011 | 5 |
| | 276 | 1111 | 7 | 1110 | 8 |
| | 277 | 1111 | 5 | 1111 | 5 |
| | 278 | 1111 | 5 | 1111 | 5 |
| | 281 | 1111 | 2 | 1011 | 6 |
| | 285 | 1111 | 6 | 1011 | 7 |
| | 286 | 1111 | 6 | 1111 | 6 |
| | 287 | 1111 | 6 | 1111 | 6 |
| | 289 | 1111 | 6 | 1111 | 6 |
| Average Efficacy | | 58.02% | | 66.67% | |

Table 2. The average efficacy of global match for category 2.

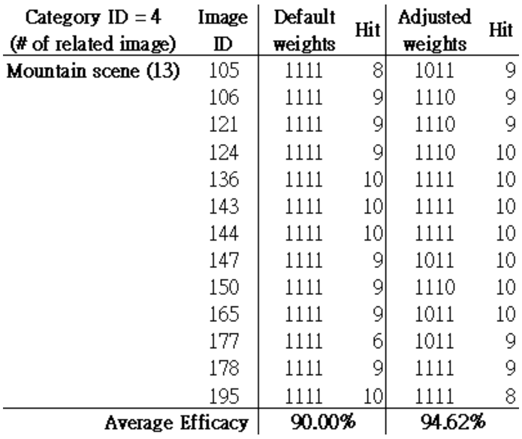
| Category ID = 2<br>(# of related image) | Image ID | Default weights | Hit | Adjusted weights | Hit |
|---|---|---|---|---|---|
| Brown animals (18) | 219 | 1111 | 7 | 1111 | 7 |
| | 220 | 1111 | 10 | 1111 | 10 |
| | 221 | 1111 | 9 | 1110 | 10 |
| | 222 | 1111 | 8 | 1111 | 8 |
| | 223 | 1111 | 6 | 1111 | 6 |
| | 224 | 1111 | 6 | 1111 | 6 |
| | 225 | 1111 | 8 | 1111 | 8 |
| | 235 | 1111 | 7 | 1110 | 8 |
| | 236 | 1111 | 6 | 1110 | 7 |
| | 237 | 1111 | 6 | 1110 | 8 |
| | 239 | 1111 | 6 | 1111 | 6 |
| | 240 | 1111 | 5 | 0111 | 9 |
| | 241 | 1111 | 9 | 1101 | 10 |
| | 242 | 1111 | 5 | 0111 | 6 |
| | 243 | 1111 | 3 | 1101 | 5 |
| | 244 | 1111 | 4 | 1111 | 4 |
| | 245 | 1111 | 2 | 1100 | 5 |
| | 246 | 1111 | 9 | 0111 | 10 |
| Average Efficacy | | 64.44% | | 73.89% | |

Table 3. The average efficacy of global match for category 3.



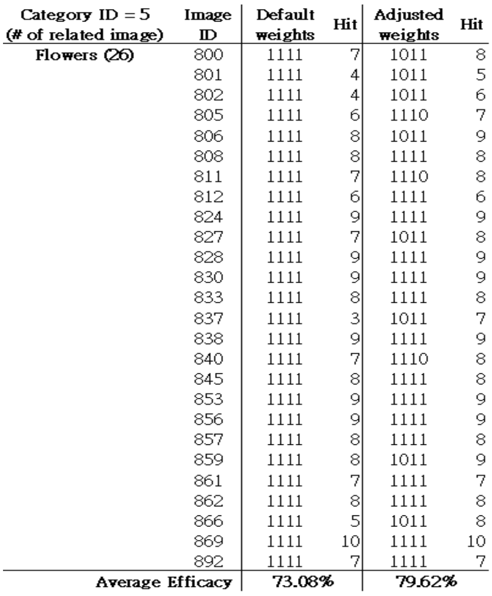
| Category ID = 3<br>(# of related image) | Image ID | Default weights | Hit | Adjusted weights | Hit |
|---|---|---|---|---|---|
| Cloudy Sky (22) | 275 | 1111 | 4 | 1011 | 5 |
| | 764 | 1111 | 7 | 1101 | 8 |
| | 766 | 1111 | 4 | 1111 | 4 |
| | 767 | 1111 | 4 | 0111 | 5 |
| | 768 | 1111 | 7 | 1110 | 8 |
| | 769 | 1111 | 5 | 1011 | 6 |
| | 770 | 1111 | 10 | 1111 | 10 |
| | 771 | 1111 | 9 | 1111 | 9 |
| | 772 | 1111 | 9 | 1111 | 9 |
| | 773 | 1111 | 5 | 1111 | 5 |
| | 774 | 1111 | 5 | 1011 | 7 |
| | 775 | 1111 | 5 | 1111 | 5 |
| | 776 | 1111 | 10 | 1111 | 10 |
| | 777 | 1111 | 10 | 1111 | 9 |
| | 778 | 1111 | 9 | 1101 | 10 |
| | 779 | 1111 | 9 | 1110 | 10 |
| | 780 | 1111 | 10 | 1111 | 10 |
| | 781 | 1111 | 9 | 1011 | 10 |
| | 782 | 1111 | 9 | 1011 | 10 |
| | 783 | 1111 | 9 | 1111 | 9 |
| | 784 | 1111 | 10 | 1111 | 10 |
| | 785 | 1111 | 10 | 1111 | 10 |
| Average Efficacy | | 76.36% | | 80.45% | |

Table 4. The average efficacy of global match for category 4.



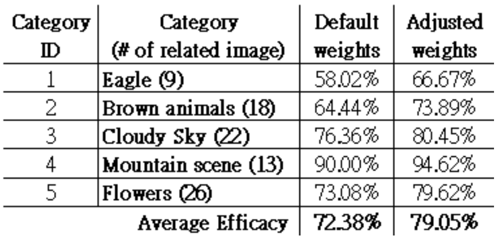
| Category ID = 4<br>(# of related image) | Image ID | Default weights | Hit | Adjusted weights | Hit |
|---|---|---|---|---|---|
| Mountain scene (13) | 105 | 1111 | 8 | 1011 | 9 |
| | 106 | 1111 | 9 | 1110 | 9 |
| | 121 | 1111 | 9 | 1110 | 9 |
| | 124 | 1111 | 9 | 1110 | 10 |
| | 136 | 1111 | 10 | 1111 | 10 |
| | 143 | 1111 | 10 | 1111 | 10 |
| | 144 | 1111 | 10 | 1111 | 10 |
| | 147 | 1111 | 9 | 1011 | 10 |
| | 150 | 1111 | 9 | 1110 | 10 |
| | 165 | 1111 | 9 | 1011 | 10 |
| | 177 | 1111 | 6 | 1011 | 9 |
| | 178 | 1111 | 9 | 1111 | 9 |
| | 195 | 1111 | 10 | 1111 | 8 |
| Average Efficacy | | 90.00% | | 94.62% | |

Table 5. The average efficacy of global match for category 5.



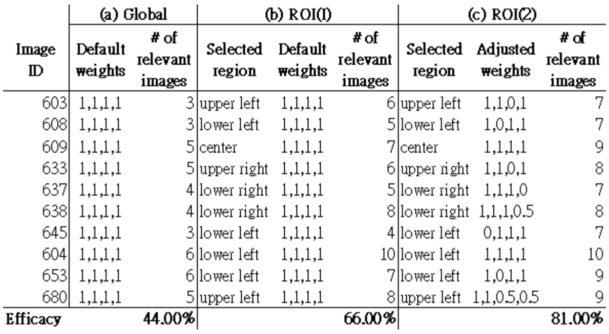
| Category ID = 5<br>(# of related image) | Image ID | Default weights | Hit | Adjusted weights | Hit |
|---|---|---|---|---|---|
| Flowers (26) | 800 | 1111 | 7 | 1011 | 8 |
| | 801 | 1111 | 4 | 1011 | 5 |
| | 802 | 1111 | 4 | 1011 | 6 |
| | 805 | 1111 | 6 | 1110 | 7 |
| | 806 | 1111 | 8 | 1011 | 9 |
| | 808 | 1111 | 8 | 1111 | 8 |
| | 811 | 1111 | 7 | 1110 | 8 |
| | 812 | 1111 | 6 | 1111 | 6 |
| | 824 | 1111 | 9 | 1111 | 9 |
| | 827 | 1111 | 7 | 1011 | 8 |
| | 828 | 1111 | 9 | 1111 | 9 |
| | 830 | 1111 | 9 | 1111 | 9 |
| | 833 | 1111 | 8 | 1111 | 8 |
| | 837 | 1111 | 3 | 1011 | 7 |
| | 838 | 1111 | 9 | 1111 | 9 |
| | 840 | 1111 | 7 | 1110 | 8 |
| | 845 | 1111 | 8 | 1111 | 8 |
| | 853 | 1111 | 9 | 1111 | 9 |
| | 856 | 1111 | 9 | 1111 | 9 |
| | 857 | 1111 | 8 | 1111 | 8 |
| | 859 | 1111 | 8 | 1011 | 9 |
| | 861 | 1111 | 7 | 1111 | 7 |
| | 862 | 1111 | 8 | 1111 | 8 |
| | 866 | 1111 | 5 | 1011 | 8 |
| | 869 | 1111 | 10 | 1111 | 10 |
| | 892 | 1111 | 7 | 1111 | 7 |
| Average Efficacy | | 73.08% | | 79.62% | |

Table 6. The comparison of global match for five categories.



| Category ID | Category<br>(# of related image) | Default weights | Adjusted weights |
|---|---|---|---|
| 1 | Eagle (9) | 58.02% | 66.67% |
| 2 | Brown animals (18) | 64.44% | 73.89% |
| 3 | Cloudy Sky (22) | 76.36% | 80.45% |
| 4 | Mountain scene (13) | 90.00% | 94.62% |
| 5 | Flowers (26) | 73.08% | 79.62% |
| | Average Efficacy | 72.38% | 79.05% |

Table 7. The comparison of global match and local match (ROI).



| | (a) Global | | (b) ROI(I) | | | (c) ROI(2) | | |
|---|---|---|---|---|---|---|---|---|
| Image ID | Default weights | # of relevant images | Selected region | Default weights | # of relevant images | Selected region | Adjusted weights | # of relevant images |
| 603 | 1,1,1,1 | 3 | upper left | 1,1,1,1 | 6 | upper left | 1,1,0,1 | 7 |
| 608 | 1,1,1,1 | 3 | lower left | 1,1,1,1 | 5 | lower left | 1,0,1,1 | 7 |
| 609 | 1,1,1,1 | 5 | center | 1,1,1,1 | 7 | center | 1,1,1,1 | 9 |
| 633 | 1,1,1,1 | 5 | upper right | 1,1,1,1 | 6 | upper right | 1,1,0,1 | 8 |
| 637 | 1,1,1,1 | 4 | lower right | 1,1,1,1 | 5 | lower right | 1,1,1,0 | 7 |
| 638 | 1,1,1,1 | 4 | lower right | 1,1,1,1 | 8 | lower right | 1,1,1,0.5 | 8 |
| 645 | 1,1,1,1 | 3 | lower left | 1,1,1,1 | 4 | lower left | 0,1,1,1 | 7 |
| 604 | 1,1,1,1 | 6 | lower left | 1,1,1,1 | 10 | lower left | 1,1,1,1 | 10 |
| 653 | 1,1,1,1 | 6 | lower left | 1,1,1,1 | 7 | lower left | 1,0,1,1 | 9 |
| 680 | 1,1,1,1 | 5 | upper left | 1,1,1,1 | 8 | upper left | 1,1,0.5,0.5 | 9 |
| Efficacy | | 44.00% | | | 66.00% | | | 81.00% |

## 7. Conclusions

In CBIR, the queries are often exploratory searches to decide which image is relevant for a certain purpose during the retrieval. For example, users usually search the web for images without assuming that the objects they are looking for are unique. Any objects described by the same information are good enough for a user's generic need. Therefore, CBIR systems require versatility and adaptation to the users, rather than the sophisticated intelligence for exact recognition tasks. Another problem in CBIR is that it is hard to guess what object is the target of users. In practice, the region of interest (ROI) is easy to observe but hard to isolate through automatic region analysis. Even though automatic segmentation can isolate multiple objects in an image, users still have to explicitly select which is their interested one. Thus, it is quite useful if a priori knowledge of the ROI is provided by users.

Under above considerations, this paper presents a general and effective feature descriptor and provides a simple way for users to express their ROI. The retrieval can be either global or local, focusing on the whole image or a particular region. During the retrieval procedures, users can adjust the weight of each individual feature through a series of experiments, with the goal of learning which features are most likely to contribute to the query. The experiment results show that the proposed DCT-based feature descriptor, coupled with a selection of ROI, can easily extend the capability of CBIR to RBIR. It is also observed that the introducing of weight adjustment explores more interesting images and improves the retrieval performance. Some promising properties of our approach are highlighted as follows:

1. It is very computationally efficient and provides good-enough performance.
2. It bridges the semantic gap by providing the

functionality of ROI.

3. It offers great flexibility for users to express their personal view on each individual feature.
4. It is simple and easily implementable.

Now the system is ready for retrieving the images through either global or local match, in which queries can be formulated at a simple semantic level. The experiments show that our approach provides a generic and efficient solution for image retrieval. However, we found that the color tones of the retrieved images are not always similar to that of the query image even though they are similar from the viewpoint of texture or shape. This is because only Y component is used in the feature vector for retrieval efficiency. In the future, we are going to consider the use of the U and V components and find out a balance point between efficiency and accuracy. In addition, from the computational point of view, CBIR systems are potentially expensive and have response times growing with the ever-increasing sizes of the databases associated to them. To address this challenge, our continuing study includes designing a multiple-stage CBIR system and developing a filter that eliminates those candidate images with widely distinct features at an earlier stage.

*Correspondence:
hsiao@mis.knjc.edu.tw;
M.-J. Hsiao is currently an instructor at Kang-Ning Junior College of Medical Care and Management.

## References

1. Brahmi D, Ziou D. Improving CBIR systems by integrating semantic features. Proc. RIAO France 2004; 291-305.
2. Besser, Howard. Visual Access to Visual Images: The UC Berkely Image Database Project. Library Trends 1990; 38(4); 787-798.
3. Smeulders A W M, Worring M, Santina S, Gupta A, Jain R. Content-based image retrieval at the end of the early years. IEEE Trans. Patt. Anal. Mach. Intell. 2000; 22(12); 1349-1380.
4. Datta R, Li J, Wang Z. Content-based image retrieval - approaches and trends of the new age. Proc. Int. Workshop on Multimedia Information Retrieval, ACM 2005; 253-262.
5. Veltkamp R C, Tanase M. Content-based image retrieval systems: A survey. Tech. Rep. UU-CS-2000-34, Utrecht University, Available online from: http://give-lab.cs.uu.nl/cbirsurvey/cbir-survey.pdf.
6. Deselaers T, Keysers D, Ney H. Features for image retrieval: an experimental comparison. Information retrieval 2008; 11(2); 77-107.
7. Liu Y, Zhang D, Lu G, Ma W Y. A survey of content based image retrieval with high-level semantics. Pattern Recognition 2007; 40(1); 262–282.
8. Islam M M, Zhang D, Lu G. Comparison of retrieval effectiveness of different region based image representations. The 6th Int. Conf. on Information, Communications & Signal Processing 2007; 1-4.
9. Rudinac S, Uscumlic M, Rudinac M, Zajic G, Reljin B. Global Image Search vs. Regional Search in CBIR Systems. Proc. of the 8th Int. Workshop on Image Analysis for Multimedia Interactive Services 2007.
10. Amir A, Berg M, Chang S-F, Hsu W, Iyengar G, Lin C-Y, Naphade M, Natsev A, Neti C, Nock H, Smith J, Tseng B, Wu Y, Zhang D. IBM research Trecvid-2003 video retrieval system. Proc. of NIST TrecVid 2003.
11. Ford A, Roberts A. Colour Space Conversions. Available at http://www.poynton.com/PDFs/coloureq.pdf 1998.
12. Li W-J, Yeung D-Y. Localized Content-Based Image Retrieval Through Evidence Region Identification. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2009.
13. Rahmani R, Goldman S A, Zhang H, Cholleti S R, Fritts J E. Localized content-based image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 2008; 30(11);1902-1912.
14. Wang Z, Li J, Wiederhold G. SIMPLIcity: Semantic-Sensitive Integrated Matching for Picture Libraries. IEEE Trans. on Pattern Analysis and Machine Intelligence 2001; 23(9); 947-963.
15. Natsev A, Rastogi R, Shim K. WALRUS: A Similarity Retrieval Algorithm for Image Databases. IEEE Trans. on Knowledge and Data Engineering 2004; 16(3); 301-316.
16. Huang PW, Dai SK. Design of a Two-Stage Content-Based Image Retrieval System Using Texture Similarity. Information Processing and Management 2004; 40(1); 81-96.
17. Wang J Z. Content Based Image Search Demo Page. Available at http://bergman.stanford.edu/~zwang/ project /imsearch/WBIIS.html 1996.