A General and Effective Two-Stage Approach for Region-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, China
hsiao@mis.knjc.edu.tw

Abstract
Content-based image retrieval (CBIR) has received substantial attentions for the past decades. It is motivated by the rapid accumulation of large collections of digital images which, in turn, create the need for efficient retrieval schemes. Many research works further utilize regional features to obtain the semantics of images for better retrieval performance. In this paper, a two-stage retrieval strategy is presented to improve the performance of region-based image retrieval (RBIR). In this approach, an image is first segmented into a fixed number of rectangular regions. Then, each region is represented by its low-frequency discrete cosine transform (DCT) coefficients in the YUV color space. At the first stage of retrieval, the threshold-based pruning (TBP) serves as a filter to remove those candidates with widely distinct features. At the second stage, a more detailed feature comparison (DFC) is conducted over the remaining candidates. In the experimental system, users can represent their region of interest (ROI) by selecting different strategies, setting parameter values, and/or adjusting the weights of features as the search progresses. The experimental results show that both efficiency and accuracy can be improved by using the proposed two-stage approach. [Life Science Journal 2010;7(3):73-80]. (ISSN: 1097-8135).

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

1. Introduction

Digital content is becoming an important medium for image collection and exchange. Given the exploring market on digital photo and video cameras, the fast growing amount of image content further increases the need for image retrieval systems. Along this line, textual annotation of images is a simple and convenient way to express the image content. For example, modern search engines and their image search offspring have enabled significant progress in domains where visual content is tagged with text descriptions, but they only analyze metadata, not the images themselves, and thus are of limited use in many practical scenarios[1]. In practice, human annotation is not only subjective but also time-consuming. Besides, there is always a big gap in creating a mapping between words and visual features. To avoid many problems associated with annotation-based approaches, content-based image retrieval (CBIR) tends to index, sort, filter, and search images based on their visual content.

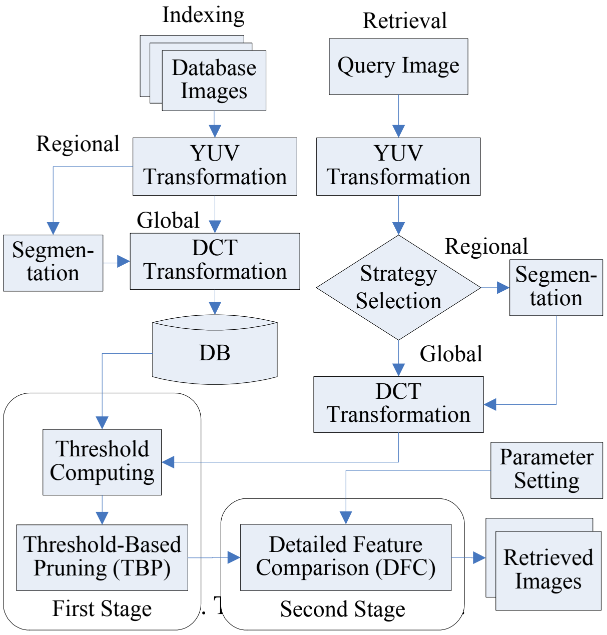
CBIR has received substantial attentions for the past decades. Some general reviews of CBIR literature can be found in[2-5]. Smeulder et al. reviewed more than 200 references in this field[2]. Datta et al. studied 120 of recent approaches[3]. Veltkamp et al. gave an overview of 43 content-based image retrieval systems[4]. Inoue reviewed the current research activities surrounding image access from the following aspects: information retrieval and organization technology, the infrastructure that enables large scale data processing, issues in human-system interaction, and the social issues[5]. These CBIR methods can be categorized into two major classes, namely, global methods and regional (or localized) methods[6]. Global methods exploit features characterizing the global view of an image while regional methods extract features from a region (or an object) of the image representing the visual content of it. Although global features can be extracted easily, in many cases, regional features contribute more meaningful image retrieval. To look at the regions (or objects) 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)[7].

In RBIR, an image is required to be segmented into a number of regions with the aim of extracting the objects within it. However, there is no unsupervised segmentation algorithm that is always capable of segmenting an image into its constituent objects, especially when considering a database containing a collection of heterogeneous images. Therefore, an inaccurate segmentation may result in an inaccurate representation and hence in poor retrieval performance[8]. A number of RBIR systems has been presented[9-12]. In Blobworld[9], objects are recognized by segmenting the images into regions (blobs) that have roughly the same color and texture. The distance between two images is calculated as the distance between the blobs in terms of color and texture. The Netra system[10] segments images into region of homogeneous color and then uses the color, texture, shape and spatial properties for measuring similarity. The SIMPLIcity system[11] uses semantics classification methods, a wavelet-based approach for feature extraction, and integrated region matching (IRM) based upon image segmentation. The WALRUS system[12] uses wavelet-based retrieval of user-specified scenes. Each image is broken into overlapped sliding windows of varying sizes. The signature for each window is described by the lowest frequency band of the Haar wavelet transform for the window. All these systems suffer the same problem that the segmentation may not have yielded regions close to the human perception of an object. The problem will become worse if a complex background is present in the image or no clear object is contained in the image. Besides, most automatic segmentation methods in RBIR include sliding-window search and region growing by pixel aggregation, region splitting and merging techniques, which are complex and computation intensive. Therefore, the size of the search space is sharply increased due to exhaustive generation of regions.

To cope with the above problems, this paper presents a two-stage retrieval strategy to model a RBIR framework. It includes three main parts: segmentation, regional feature extraction, and retrieval strategy selection. Segmentation and regional features are considered as black boxes, which means any segmentation algorithm and descriptor can be used within this framework. To reduce the problem of image segmentation, the images are first segmented into a fixed number of rectangular regions. Then, each region is described by its low-frequency discrete cosine transform (DCT) coefficients in the YUV color space. While conducting a query, the user can select the region of interest (ROI) in the query image to express his/her intentions. The similarity is evaluated by inspecting each region in the candidate images in turn, to find the best matching region with the query region. To let the system concentrate its effort on promising images, our two-stage retrieval strategy is applied. At the first stage of retrieval, the threshold-based pruning (TBP) serves as a filter to remove those candidates with widely distinct features. At the second stage, a more detailed feature comparison (DFC) is conducted over the remaining candidates.

A system framework is developed for realizing the proposed two-stage approach. It shows advantage over other RBIR systems in the following three aspects. First, during the retrieval process, color features and texture features are used in a two-stage way. To better combine these features, a friendly system user interface (UI) is provided to allow weight assignment for each individual feature. Second, the TBP mechanism employs an effective pruning algorithm to obtain potential candidate set, based on low-dimensional color and texture features of images. Last, a perception-dependent query strategy is proposed to support implicit relevance feedback. Instead of enforcing users to make explicit judgment on the results, the system UI simply lets the user select different strategies, adjust weights, and browse the results during the retrieval process. The experimental results show that the proposed framework is general, efficient and effective.

The remainder of this paper is organized as follows. The next section introduces the system framework for realizing the proposed two-stage approach. Section 3 discusses related issues about segmentation and region of interest. Section 4 illustrates the feature extraction process for an image or a region. The TBP method, which serves as the first stage of the retrieval, is shown in Sec. 5. Section 6 describes the second stage of the retrieval.



Section 7 explains how the performance is evaluated and Sec. 8 presents experimental results. Finally, some concluding remarks are drawn in Sec. 9.

2. The Proposed System Framework

To alleviate the burden of computation, the retrieval strategy is divided into two stages. The system framework that supports the proposed two-stage approach is illustrated in Figure 1. For the ease of extracting features, all images with different color-space are first converted into the YUV domain before they are forwarded in to our system. Then an image is equally divided into four rectangular regions and one additional central region of the same size. The DCT transform is performed over the Y, U, and V components for a whole image (global features) and five regions (regional features). In the indexing phase, the low-frequency DCT coefficients are all stored into the database.

When a query is submitted, the user can interact with the system by selecting different strategies and setting parameter values, such as ROI, threshold, standard deviation, and feature weights, etc. In the retrieval phase, at the first stage, the threshold-based pruning (TBP) serves as a filter to prune those images whose distances to the query image are beyond a distance threshold so that a smaller set of candidates is achieved. The threshold is obtained in advance by gathering the distances between the query image and database images. At the second stage, the detailed feature comparison (DFC) is performed on those resultant candidates passing through the first stage. This system also gets the user into the retrieval loop so that personalized results could be provided for the specific user.

3. Segmentation and Region of Interest

Image features for CBIR may be either "global" or "regional." A global feature is "coarse-grained," that is, it represents the image in its entirety, such as the overall color distribution. A regional feature is a finer-grained representation achieved through segmentation of the image into smaller regions. Both global and regional features offer advantages in processing and querying[13]. Global features offer advantages in terms of low computation complexity of feature extraction and pattern matching algorithms and can be used when queries deal with single entities. Regional features can be used to identify (or locate) objects within region of interest and to extract detailed information.

Object localization is an important task for the automatic understanding of images, which decides an object is present in an image or not, and even where exactly in the image the object is located[14]. There are a number of problems that generally hinder object localization, such as background noise, varying angles of view and object occlusion, and variations in image resolution and lighting conditions. We should note that though noise present in images interferes with image processing and interpretation, RBIR can still improve the performance because segmenting images into small regions would degrade the background effects.

To separate a region/object from the background, many different definitions of object localization exist in the literature. Typically, they differ in the form that the location of an object in the image is represented, e.g. by its center point, its contour, a bounding box, or by a pixel-wise segmentation. In the field of object localization with bounding boxes, sliding window approaches have been the method of choice for many years. Because the number of rectangles in an $n \times m$ image is of the order n^2m^2, one cannot check all possible regions exhaustively. Besides, the object is very hard to be identified because an image may contain several objects with varying positions and sizes, and one region may encompass different shades of the same hue, strong or weak textures, etc. In the approaches of the pixel-wise segmentation, the algorithms are complex and computation intensive, and the segmentation results are often not correct. For example, it is very hard to extract a region in complex natural scenes. The main difficulties arise from the intrinsic randomness of natural textures and the high-semblance between the objects and the background[15]. Instead of pursuing sophisticated segmentation methods, we segment each into a fixed 5-region layout (4 equal corner regions and an overlapping center region of the same size) as in the IBM TRECVID video retrieval system[16]. It has been shown that the segmentation is fast and thus suitable for a RBIR system with a large database.

In practice, even if a region is correctly detected, its visual appearance is not always specific to a single class of "objects" in a heterogeneous image database. Conversely some semantic "objects" can have very different visual appearance, such as "dog", "cloth", "car", etc. In other words, semantics and visual description do not always have a one-to-one correspondence. Besides, during the query process, it's hard to guess what region/object is the target of users, especially when multiple regions/objects are identified. The most straightforward solution of the problems is to let the user select a ROI while conducting a query. For example, if a query image contains various concepts: "train", "railroad", "sky", "trees", and "mountain", the user has to select the ROI that contains objects of interest. Our system UI allows the user to select any one of the five regions (upper left, upper right, lower left, lower right, and center) as the target region. As far as the target region is selected, the concept in the query image is more focused.

4. Feature Extraction

An image feature vector is a compact representation of how the image populates the feature space. In CBIR, the feature extraction will be invoked very frequently; therefore, too many items in a feature vector will make feature extraction and similarity evaluation become infeasible for an image retrieval application which requires instant response to a query. Generally in a CBIR system images are represented by three main features: color, texture, and shape. Each of these features has its own advantage to characterize a type of image content.

Some transform-based feature extraction techniques have been successfully 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. We use the low-frequency DCT coefficients as the color and texture features of an image. This is due to its strong "energy compaction" property: most of the signal information tends to be concentrated in a few low-frequency DCT coefficients. Some studies have shown the effectiveness of using the low-frequency DCT coefficients as feature vectors in CBIR[17-18].

Although the JPEG standard does not specify any particular color space for standard usage, our approach prefers the use of YUV color space in order to easily extract the features based on the color tones. All images with different color-space are first converted into the YUV domain before they are forwarded in to our system. The DCT is performed over the Y, U, and V components for a whole image (global features) and five regions (regional features). For an $N \times N$ image (or region) represented by pixel values $f(i, j)$, its DCT coefficients $C(u, v)$ can be defined as

$$C(u,v) = \frac{2}{N}\alpha(u)\alpha(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j) \times \cos(\frac{(2i+1)u\pi}{2N})\cos(\frac{(2j+1)v\pi}{2N}), \quad (1)$$

for $i, j, u, v = 0, 1, \ldots, N$-1, where $\alpha(w) = 1/\sqrt{2}$ if $w =$ 0 and 1 otherwise. The feature vector for each image or region is further 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). For a 4×4 upper left corner of a DCT coefficient block, they can be defined as

$V_1 = [C_{00}]$,

$V_2 = [C_{01}, C_{02}, C_{03}, C_{12}, C_{13}]$,

$V_3 = [C_{10}, C_{20}, C_{30}, C_{21}, C_{31}]$, and

$V_4 = [C_{11}, C_{22}, C_{23}, C_{32}, C_{33}]$.

In our experiments, it can be observed that when taking into account directional textures (by setting combination weights to positive values through UI) the improvement in visual relevance of retrieved regions is usually noticeable.

5. The First Stage of Retrieval

The first stage of retrieval is to eliminate the obviously dissimilar candidates via a certain criterion. When an image sample is being queried, the candidates whose distances to the query image are beyond a distance threshold will be pruned at the first stage to narrow the candidate targets. Such a retrieval procedure is coined threshold-based pruning (TBP) in this paper. Note that TBP is conducted from the global view of images, not regional properties. In addition, 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, both the U and V components can be of great help in removing those images with distinct color tones at the first stage of retrieval.

5.1 The Similarity Measurement

To exploit the energy preservation property of DCT, the sum of squared differences (*SSD*) is used to measure the distance of two images. Assume that $C_Q(u,v)$ and $C_X(u,v)$ represent the DCT coefficients of the query image *Q* and a candidate *X* in the Y component, respectively. Then the SSD_Y between Q and X under the upper left block of size *n*×*n* of the Y component can be defined as

$$SSD_Y(Q,X,n) = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1}\left(C_Q(u,v) - C_X(u,v)\right)^2. \quad (2)$$

5.2 Threshold-Based Pruning

To realize the threshold-based pruning (TBP), we have to choose the suitable block size of DCT and the threshold *T*. A bigger candidate set after TBP means more probable to include good candidates at the cost of more computation. Decreasing the value of *T* reduces the amount of comparison at the sacrifice of excluding more potential candidates. Therefore, choosing the threshold *T* requires a certain amount of compromise.

In our work, the threshold values are derived by using the Chebyshev rule[19]. From this rule, we know that the percentage of observations contained within distances of *k* standard deviations around the mean must be at least $(1 - 1/k^2) \times 100\%$. For example, 75% is guaranteed within the distances of 2 standard deviations. Assume that T_n represents the threshold for the block size *n*x*n*. To obtain the threshold T_n, the *SSD* between the query image and each candidate under the block size *n*×*n* is gathered. Thus, T_n can be derived by

$$T_n = \mu_n + k \times \sigma_n, \quad (3)$$

where μ_n and σ_n are the mean and standard deviation of *SSD*, respectively. Note that the Y, U, and V components are all used with a block size of 2×2 for the pruning criterion; therefore,□ μ_2 and σ_2 can be obtained by

$$\mu_{2(Y)} = \frac{1}{N_c}\sum_{i=1}^{N_c} SSD_Y(q, C_i, 2), \quad (4)$$

$$\mu_{2(U)} = \frac{1}{N_c}\sum_{i=1}^{N_c} SSD_U(q, C_i, 2), \quad (5)$$

$$\mu_{2(V)} = \frac{1}{N_c}\sum_{i=1}^{N_c} SSD_V(q, C_i, 2), \quad (6)$$

$$\sigma_2 = \frac{1}{N_c}\sum_{i=1}^{N_c}((SSD_Y(q, C_i, 2) - \mu_{2(Y)})^2 + (SSD_U(q, C_i, 2) - \mu_{2(U)})^2 + (SSD_V(q, C_i, 2) - \mu_{2(V)})^2), \quad (7)$$

where N_c is the number of candidates in the database. The pruning process is denoted as TBP(YUV) for better understanding. The reason that Y, U, and V components are all used at TBP is to remove those images with widely distinct textures or color tones such that the return images will have a more consistent visual similarity with the query image whether they are uniform, textured or encompassing different shades of a given color.

To find out the best number of standard deviation, the threshold area provides an interface for the user to specify a threshold value in our system. Intuitively, setting the threshold value low can include more potential candidates. Since test results for CBIR systems are difficult to quantify objectively, a series of sample queries are conducted to access the mean performance. The finding shows that selecting any one of the thresholds did not cause any change in the top-10 list. Therefore, *k*=1 is the best choice because it provides 42.72% of data reduction rate, the highest one among four thresholds. The following experiments are all conducted based on the threshold under one standard deviation.

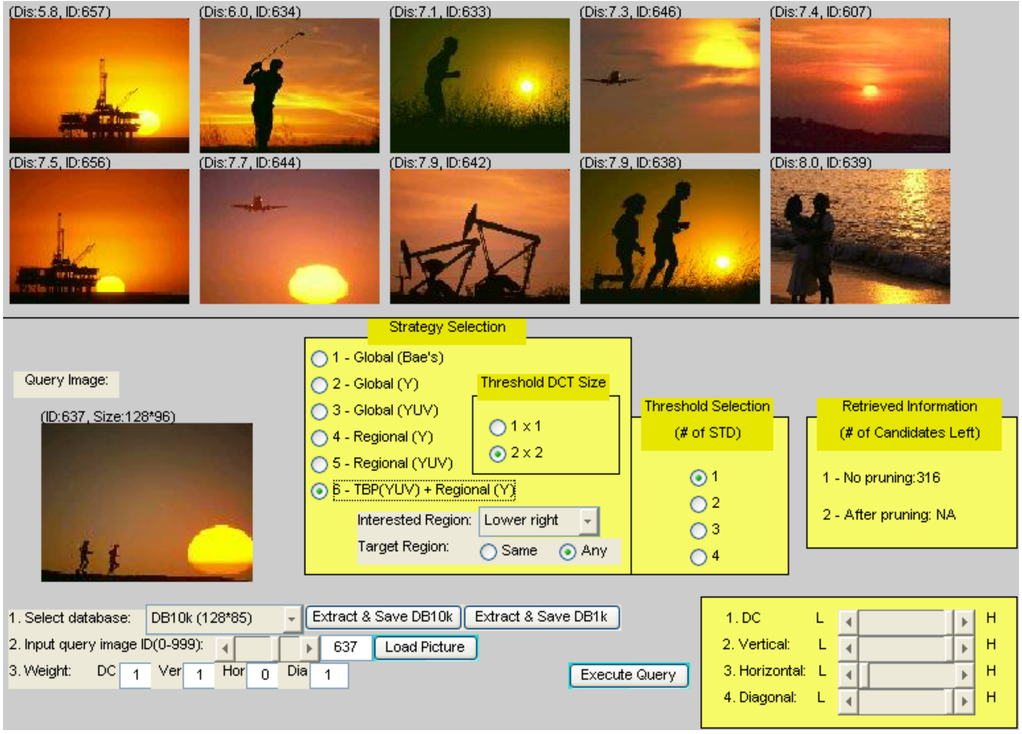
6. The Second Stage of Retrieval

In contrast to global match at the first stage, regional match at the second stage does a complete comparison between the query region and all regions in candidate images. To characterize images in a more detailed sense, a larger block size is involved at the second stage.

6.1 Detailed Feature Comparison

At the second stage, retrieval is performed by exhaustive comparison with query region, i.e., all regions in the remaining candidates that pass through the TBP filter are compared to the query region. The retrieval problem can be formulated as

$$dist(q, X) = \min_{x \in X} dist(q, x), \quad (8)$$



(Dis:5.8, ID:657) (Dis:6.0, ID:634) (Dis:7.1, ID:633) (Dis:7.3, ID:646) (Dis:7.4, ID:607)

(Dis:7.5, ID:656) (Dis:7.7, ID:644) (Dis:7.9, ID:642) (Dis:7.9, ID:638) (Dis:8.0, ID:639)

Query Image:

(ID:637, Size:128*96)

Strategy Selection

○ 1 - Global (Bae's)
○ 2 - Global (Y)
○ 3 - Global (YUV)
○ 4 - Regional (Y)
○ 5 - Regional (YUV)
● 6 - TBP(YUV) + Regional (Y)

Threshold DCT Size
○ 1 x 1
● 2 x 2

Interested Region: Lower right

Target Region: ○ Same ● Any

Threshold Selection
(# of STD)
● 1
○ 2
○ 3
○ 4

Retrieved Information
(# of Candidates Left)
1 - No pruning:316
2 - After pruning: NA

1. Select database: DB10k (128*85)

2. Input query image ID(0-999): 637

3. Weight: DC 1 Ver 1 Hor 0 Dia 1

Figure 2. The main screen of our system, presenting results for a given query (image ID = 637)

where q is a query region and x is a region in a candidate image X. The distance between two regions is evaluated by detailed feature comparison (DFC). In DFC, only the Y components with a block size of 4x4 are used to calculate the distance between two regions. This process is denoted as DFC(Y) in comparison to TBP(YUV). The distance function in DFC(Y) is defined as

$$dist(q,X)=\min_{x\in X} dist(q,x)$$

$$=\min_{x\in X} SSD_Y(q,x,4). \qquad (9)$$

Note that the block size in DFC(Y) is larger than the block size in TBP(YUV), which means DFC(Y) is relatively detailed than TBP(YUV). Because the images with distinct color tones are removed at the first stage, only the Y component needs to be considered at the second stage for the purpose of efficiency.

6.2 Weighted Distance Measurement

In order to express users' semantic concepts with low-level features, the weights of these feature components have to be adjusted by users. This is important as users' interpretation varies with respect to different information needs and perceptual subjectivity. In addition, users tend to learn from the retrieval results to further refine their information priority. It is, therefore, useful to let users describe their perceptions of images by a set of weights, which also serves as a certain level of relevance feedback.

The distance measurements are defined in terms of weighted combinations of all features. Each weight associated to the individual feature is the user's personal interest to that feature. The distance function in DFC(Y) is modified to:

$$SSD_Y(q,x,V_i)=\sum_{C(u,v)\in V_i}(C_q(u,v)-C_x(u,v))^2, \qquad (10)$$

$$SSD_Y(q,x)=\sum_{i=1\text{ to }4} w_i * SSD_Y(q,x,V_i)\text{ , and} \qquad (11)$$

$$dist(q,X)=\min_{x\in X} SSD_Y(q,x), \qquad (12)$$

where q is a query region, x is a region in a candidate image X and w_i is the weight for V_i, indicating the significant level of the *i-th* feature. Note that V_1, V_2,V_3, and V_4 are the average grayness, vertical texture, horizontal texture, and diagonal texture, respectively. They are defined under the upper left block of size 4×4, as described in Sec. 4. The overall similarity is computed on the basis of the assigned weights, modeling what users see when they look at the query image.

7. Performance Evaluation

Generally, searches in CBIR are activities to decide which image is relevant for a certain purpose during the retrieval. For example, users usually search the database 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, some browsing is often involved in a CBIR system and only a short list of images is returned to the user.

Two commonly used performance measures, the *precision* rate and the *recall* rate, in textual information retrieval can be adapted for CBIR. *Precision* measures the ratio with which the relevant images are returned among the best *M* matches; *recall* indicates the portion of relevant images that are returned among the best *M* matches. In practice, when the number of relevant images is greater than the size of the returned list, *recall* is meaningless as a measure of the retrieval quality. To overcome this problem, a measure called *efficacy* (η_M) is introduced[20]:

$$\eta_M = \begin{cases} n_r / N_r \text{ , } if\ N_r \le M \\ n_r / M \text{ , } if\ N_r > M \end{cases} , \qquad (13)$$

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. If $N_r \le M$, η_M becomes the traditional *recall* of information retrieval; if $N_r > M$, η_M is indeed the *precision* of information retrieval. In our system, only the best 10 matches are returned, i.e., $M = 10$.

8. Experimental Results

For the experiments, we used an image database of 1,000 color images downloaded from the WBIIS image database[21]. It mainly consists of scenes of natural, animals, insects, building, people, and so on. No pre-processing was done on the images. Figure 2 shows the main screen of our RBIR system. The lower window on the screen gives the user possibility to edit some parameter values. For instance, the user can select the matching strategies, the position of ROI, the threshold DCT size, the number of standard deviations, or the weights for features. The top 10 matched images are displayed in the upper window, ranked in descending order of similarity to the query image (or region) from the left to the right and then from the top to the bottom. Figure 2 also presents a query example, showing the query results after the user loads a query image (ID = 637), selects the retrieval strategy and the ROI, adjusts the weights, sets other parameter values, and launches the query.

8.1 Evaluation of the YUV Color Space

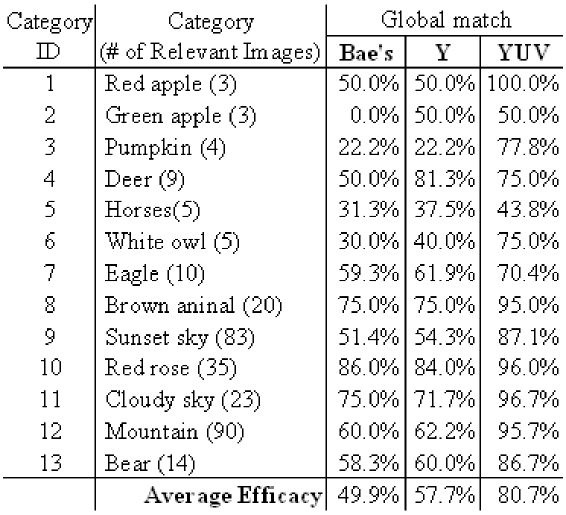
One of the main aspects of color feature extraction is the choice of a color space. The RGB color space provides a useful starting point for representing color features of images. Alternative color spaces can be generated by transforming the RGB color space, such as CMYK (Cyan, Magenta, Yellow, and Black Key), CIE (Centre International d'Eclairage), YUV (Luminance and Chroma channels), etc. Each of the models is specified by a vector of values; each component of that vector being valid on a specified range. In our approach, the RGB images are first transformed to the YUV color space for the purpose of extracting the features based on the color tones more easily.

A sample image is purposely selected to illustrate the RGB and YUV color space, which contains three overlapped circles (see Figure 3). Each circle has one dominant color: red, green or blue. In this figure, for example, if we need to verify which part of a circle mainly contains a red color tone, it makes vain attempt to analyze the R component of the RGB image because most of the energy of the R component contributes to the luminance of the image; on the other hand, just like the philosophy lies in the orthogonal theorem, it is more effective to analyze the V component of the image, which eliminates the component that constitutes the luminance of the image. This vindicates the appropriateness of using the YUV color space in our approach.

8.2 Evaluation of Global Matching Strategies

Three global-based retrieval (or matching) strategies are examined in the following experiments: Global(Bae's)[22], Global(Y), and Global(YUV). Note that the letters in each parenthesis are the algorithm or the color components used in that strategy. The experimental results are summarized in Table 1, where efficacy is obtained by taking average over the retrieval results of query examples in each category. It clearly indicates that the efficacy of the Global(Bae's) approach is the lowest because it uses the RGB color space and does not sufficiently capture color information in the images. It can also be seen that Global(Y) is better than Global(Bae's) but still not good as Global(YUV), without the help of the U and V components. The finding suggests that Global(YUV) can serve as the criterion of TBP in our two-stage approach, removing the images with far distinct color or texture features from the candidates before getting into regional detailed feature comparison (DFC).

Table 1. The performance comparison of different global matching strategies.



Category ID	Category (# of Relevant Images)	Global match		
		Bae's	**Y**	**YUV**
1	Red apple (3)	50.0%	50.0%	100.0%
2	Green apple (3)	0.0%	50.0%	50.0%
3	Pumpkin (4)	22.2%	22.2%	77.8%
4	Deer (9)	50.0%	81.3%	75.0%
5	Horses(5)	31.3%	37.5%	43.8%
6	White owl (5)	30.0%	40.0%	75.0%
7	Eagle (10)	59.3%	61.9%	70.4%
8	Brown aninal (20)	75.0%	75.0%	95.0%
9	Sunset sky (83)	51.4%	54.3%	87.1%
10	Red rose (35)	86.0%	84.0%	96.0%
11	Cloudy sky (23)	75.0%	71.7%	96.7%
12	Mountain (90)	60.0%	62.2%	95.7%
13	Bear (14)	58.3%	60.0%	86.7%
	Average Efficacy	49.9%	57.7%	80.7%

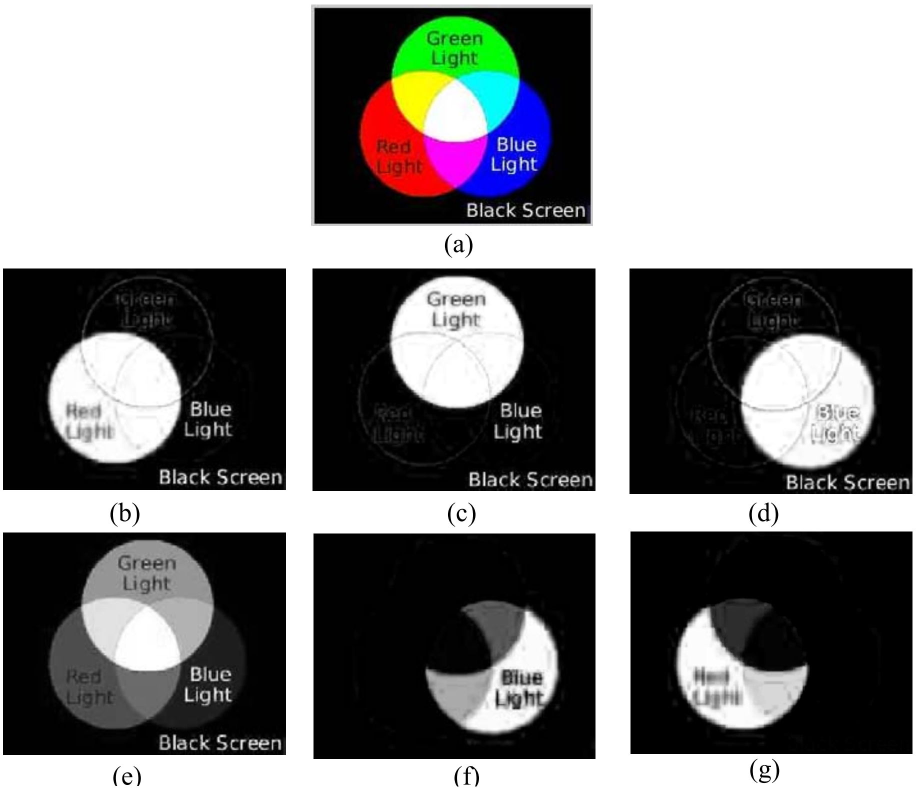


Figure 3. Illustration of RGB and YUV color spaces: (a) the sample image and the (b) R component, (c) G component, (d) B component, (e) Y component, (f) U component, and (g) V component of the image.

8.3 Evaluation of the Two-Stage Regional Matching Strategy

As there is no feature capable of covering all types of images, both global and regional features offer advantages in processing and querying. For those query images without clear objects, the user can select Global(YUV) to get the best results. If the user is only interested in a region that contains a target object, he/she can select regional strategies to conduct a more meaningful query.

The following experiments verified the proposed two-stage regional retrieval (or matching) strategy. We purposely choose ten queries with poor performance on the global match, compared with the same queries on the regional match. Table 2(a) shows the retrieval results for global match, where the queries use the whole image for feature construction without performing ROI selection. In the regional match, where the query is a region in an image, three regional matching strategies are studied and the results are summarized in Table 2(b): Regional(Y), Regional(YUV), and TBP(YUV)+Regional(Y). Note that the right most column shows the ROI, weights, and the number of relevant images. For convenience, the weights for four feature vectors are represented in the form of (w_1,w_2,w_3,w_4). We can see that Regional(Y) gives better performance than Global(YUV). It is reasonable because ROI would degrade the background effect even only the Y component is used. In comparison with Regional(Y), Regional(YUV) makes great improvement on the help of U and V components, raising efficacy from 41% to 77%. The improvement by the proposed approach, TBP(YUV)+Regional(Y), may not be very significant, but it decreases the running time because the total items of the feature vector in TBP(YUV)+Regional(Y) are less than those in Regional(YUV). It is also observed that instead of using the default weights (1,1,1,1), adjusting the weights will favor some queries. Users can adjust the weights through a series of experiments, with the goal of learning which features are most likely to contribute to the query.

9. Conclusions

In RBIR, computing similarity between two images is equivalent to region matching. Designing a RBIR system remains a challenging and open problem: automatic region segmentation is a hard task and its high complexity is generally a strong limitation due to the huge number of regions. In many cases, pixels which lie between segments or in high frequency areas of an image cannot be easily categorized as belonging to any particular segments. In this paper, we have simplified the problem in two aspects: 1) the number of representative regions for each image is only five, and 2) threshold-based pruning (TBP) reduces the matching to a smaller set of images. As can be seen in the experiments, segmenting images into small regions would degrade the background effects if the user is only interested in a portion of the query image. It is also shown that efficiency can be obtained while maintaining and sometimes improving the accuracy, by using TBP to prune those obviously unqualified candidates at earlier stages.

Table 2. The performance study of regional matching strategies.

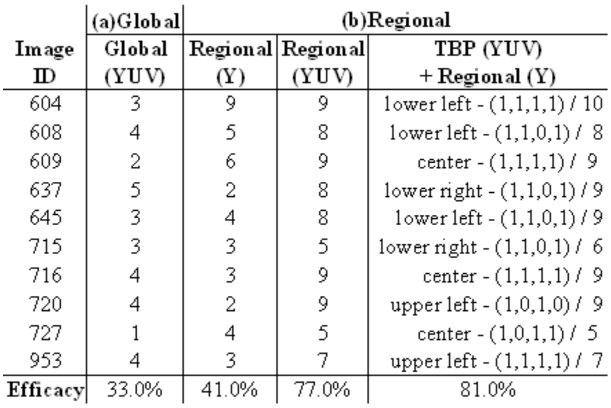


Image ID	(a)Global Global (YUV)	(b)Regional Regional (Y)	Regional (YUV)	TBP (YUV) + Regional (Y)
604	3	9	9	lower left - (1,1,1,1) / 10
608	4	5	8	lower left - (1,1,0,1) / 8
609	2	6	9	center - (1,1,1,1) / 9
637	5	2	8	lower right - (1,1,0,1) / 9
645	3	4	8	lower left - (1,1,0,1) / 9
715	3	3	5	lower right - (1,1,0,1) / 6
716	4	3	9	center - (1,1,1,1) / 9
720	4	2	9	upper left - (1,0,1,0) / 9
727	1	4	5	center - (1,0,1,1) / 5
953	4	3	7	upper left - (1,1,1,1) / 7
Efficacy	33.0%	41.0%	77.0%	81.0%

In addition to fast segmentation and efficient matching, our approach also explores the user's presence in the retrieval, allowing the user to select the retrieval strategy, the point of interest and adjust the weights of features. With our friendly system UI, users can easily engage themselves in incremental query refinement and iterative retrieval by selecting different strategies and setting parameter values with the goal of exploring more interesting images. The experimental results show that our approach is general, efficient and effective. This evidence also suggests the potential application of the combination of fixed segmentation, flexible weight assignment, and two-stage retrieval for handling RBIR problems.

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. Vasconcelos N. From Pixels to Semantic Spaces: Advances in Content-Based Image Retrieval, IEEE Computer, 2007; 40(7); 20-26.
2. 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. on Pattern Analysis and Machine Intelligence 2000; 22(12); 1349-1380.
3. Datta R, Li J, Wang Z. Content-Based Image Retrieval - Approaches and Trends of the New Age. Proc. of Int. Workshop on Multimedia Information Retrieval, ACM 2005; 253-262.
4. 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.
5. INOUE M. Image Retrieval: Research and Use in the Information Explosion, Progress in Informatics, Special Issue: Leading ICT Technologies in the Information Explosion 2009; 6; 3-14.
6. Li W-J, Yeung D-Y. Localized Content-Based Image Retrieval Through Evidence Region Identification, Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2009.
7. Islam M M, Zhang D, Lu G. Comparison of Retrieval Effectiveness of Different Region Based Image Representations, Proc. of the 6th Int. Conf. on Information, Communications & Signal Processing 2007; 1-4.
8. Nascimento M A, Sridhar V, Li X. Effective and Efficient Region-Based Image Retrieval, Journal of Visual Languages and Computing 2004; 14; 151-179.
9. Carson C, Thomas M, Belongie S, Hellerstein J M, Malik J. Blobworld: A System for Region-Based Image Indexing and Retrieval, Proc. of the Third Int. Conf. on Visual Information Systems 1999; 509-516.
10. Ma W Y, Manjunath B S. Netra: A Toolbox for Navigating Large Image Databases, Multimedia Systems 1999; 7 (3); 184-198.
11. 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.
12. 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.
13. Al-Khatib W, Day Y F, Ghafoor A, Berra P B. Semantic Modeling and Knowledge Representation in Multimedia Databases, IEEE Trans. on Knowledge and Data Engineering 1999; 11(1); 64-80.
14. Lampert C H, Blaschko M B; Hofmann T. Efficient Subwindow Search: A Branch and Bound Framework for Object Localization, IEEE Trans. on Pattern Analysis and Machine Intelligence 2009; 31(12); 2129-2142.
15. Ding J, Shen J , Pang H, Chen S, Yang J. Exploiting Intensity Inhomogeneity to Extract Textured Objects from Natural Scenes, Lecture Notes in Computer Science 2010; 5996; 1-10.
16. 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.
17. Lu Z-M, Li S-Z, and Burkhardt H. A Content-Based Image Retrieval Scheme in JPEG Compressed Domain, Int. Journal of Innovative Computing, Information and Control 2006; 2(4); 831-839.
18. Tsai T, Huang Y-P, and Chiang T-W. Fast Image Retrieval Using Low Frequency DCT Coefficients, Proc. of the 10th Conf. on Artificial Intelligence and Applications-International Track 2005.
19. R.E. Walpole, R.H. Myers, S.L. Myers, and K. Ye, Probability and Statistics for Engineers and Scientists, 7th Ed., Upper Saddle River, N.J., Prentice Hall, 2002.
20. 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.
21. Wang J Z. Content Based Image Search Demo Page. Available at http://bergman.stanford.edu/~zwang/ project /imsearch/WBIIS.html 1996.
22. Bae H-J, Jung S-H. Image Retrieval Using Texture Based on DCT, Proc. of Int. Conf. on Information, Communications and Signal Processing 1997; 1065-1068.

9/15/2010