On-line Content-Based Image Retrieval System using Joint Querying and Relevance Feedback Scheme

WICHIAN PREMCHAISWADI, ANUCHA TUNGKATSATHAN
Graduate School of Information Technology in Business
Siam University
235 Petkasem Rd., Phasicharoen, Bangkok, 10160
THAILAND
wichian@siam.edu, aimdala@hotmail.com

Abstract: - In a high-level semantic retrieval process, we utilize the search engine to retrieve a large number of images using a given text-based query. In a low-level image retrieval process, the system provides a similar image search function for the user to update the input query for image similarity characterization. This paper presents an On-line Content-Based Image Retrieval System using joint querying and relevance feedback scheme based on both high-level and low-level features. We also introduce fast and efficient color feature extraction namely auto color correlogram and correlation (ACCC) based on color correlogram (CC) and autocorrelogram (AC) algorithms, for extracting and indexing low-level features of images. To incorporate an image analysis algorithm into the text-based image search engines without degrading their response time, the framework of multi-threaded processing is proposed. The experimental evaluations based on coverage ratio measure show that our scheme significantly improves the retrieval performance of existing image search engine.

Key-Words: - joint querying, image retrieval, ACCC, on-line CBIR, CBIR, relevance feedback

1 Introduction

Content-Based Image Retrieval (CBIR) has been an ongoing area of research for decades but is still not appearing in the mainstream. Many applications like Qbic [1], VisualSeek [2], Blobworld [3], and MetaSEEk [4] are attracting attention, but they are still not very common. Most current CBIR systems presented in academic and research papers tend to focus on new technical issues and algorithms, and frequently ignore the end user point of view. Some limitations of current CBIR systems were presented by Theo Pavlidis [5]. For instance, "The papers describing the methodologies presented, what seemed to be, impressive results but soon I realized that the paper do not deal with realistic situations". It is not only him who thinks that way, but also the lead editorial of the IEEE Proceedings on multimedia retrieval who presented the titled "The Holy Grail of Multimedia Information Retrieval: So Close or Yet So Far Away?" [6].

As an interdisciplinary research field, image representation schemes designed for image retrieval systems can be categorized into three classes including textual (keyword) feature, visual feature, and their combinations [7]. The very first attempts for image retrieval were keyword-based approach, which requires a previous annotation of the database images. It is relatively simple and computationally efficient and more direct mapping toward high-level semantics than low-level visual features. However, there are two main drawbacks. First, manual annotation is a tedious and expensive process when the size of image database is large [7]. Second, building keyword models from visual features of a set of images labeled with keywords is still critical [7]. In the area of CBIR, it overcomes the difficulties of manual annotations by using visual feature-based representations, such as color, texture, shape, etc. However, after over a decade of intensified. The major bottleneck of this approach is the gap between visual feature representations and semantic concepts of images. Low-level contents often don't describe the high-level semantic concepts in users' minds. Some researcher considered To improve this burden, one promising direction towards semantic retrieval is the adoption of relevance feedback mechanism [8]. Many researchers focus on these relevance techniques because they are important in achieving a better precision rate [9][10][11]. The technique is a variation of "query by example" that involves multiple interactions with a user at search time [6]. It refers to the feedback from a user on specific items regarding their relevance to a target image, in each iteration, the refined query is re-evaluated. Some papers considered optimizing relevance feedback search by query prediction [8]. Even popular commercial search engines such as Google and Yahoo have developed many powerful and flexible functions to search for images that exactly match the individual user's needs. For example, the user can specify the size of images or types of images such as news contents, faces, clip art, line drawings, and photo contents or even choose the color shade of images through a specific color palate. Google has released their latest image search function, Google Similar [12], which allows a user to search for images using pictures rather than words. A user can click on the "Similar images" link under an image to find other images that are similar to it. It is an experimental service from Google Labs [13] that lets a user find images that are similar to an image he or she has selected. This function is very similar to our proposed image search scheme. However, not every image retrieved by using this function is relevant because the image indexing is formerly prepared by Google. Using Google Similar Images, the user enters the keyword, such as “Thai temple”, and clicks on the “search images” button, the results of the retrieved images are presented as shown in Fig. 1(a). After that he or she can further click on the similar images’ link under any retrieved images to obtain all relevant images with that image. This process is shown in Fig. 1(b).

(b)

Fig. 1. Similar image function in Google image search

To overcome the drawbacks of the existing keyword-based and content-based retrieval systems, many researchers combined text-based retrieval with content-based retrieval. In [14], the models and techniques were used to merge textual and image features to classify images. Lu [15] proposed the framework of relevance feedback technique to take advantage of the semantic network on top of the keyword association on the images in addition to the low-level features. Chang [16] further improved this framework using the probabilistic output of SVM to perform annotation propagation in order to updating unlabeled images in addition to labeled images. [7] Proposed a unified image retrieval framework based on both keyword annotations and image visual feature. For each keyword, a statistical model is trained by using visual feature of labeled images. Moreover, an effective update keyword models using newly labeled images periodically approach is proposed. However, the common limitation of this framework is the keyword models built from visual feature of a set of images are labeled with semantic keywords.

In this paper, we utilize the search engine to retrieve a large number of images using a given text-based query. In the low-level image retrieval process, the system provides a similar image search function for the user to update the input query for image similarity characterization. The proposed scheme is not the same as the existing framework of unifying keywords and visual content systems. The key word models built from visual feature of a set of images are labeled with keywords. It incorporates an image analysis algorithm into the text-based image search engines. Moreover, it is implemented on real-world image database. A high-level semantic retrieval can be done by using relevance images from Yahoo image search engine. For low-level feature, we introduce a fast and robust color feature extraction technique namely auto color correlogram and correlation (ACCC) based on color correlogram (CC) [17] and autocorrelogram (AC) [17] algorithms, for extracting and indexing low-level features of images. The retrieval performance is satisfactory and higher than the average precision of the retrieved images using autocorrelogram (AC). Moreover, It can reduce computational time from $O(m^2d)$ to $O(md)$ [18]. The framework of multi-threaded processing is proposed to incorporate an image analysis algorithm into the text-based image search engines. It enhances the capability of an application when downloading images, indexing, and comparing the similarity of retrieved images from diverse sources.

Section II Proposed the framework of a joint querying scheme for an on-line image retrieval system with multithreading. Section III discusses the proposed indexing technique for image visual content. Our experimental study is presented in Section IV, and concluding remarks are set out in Section V.

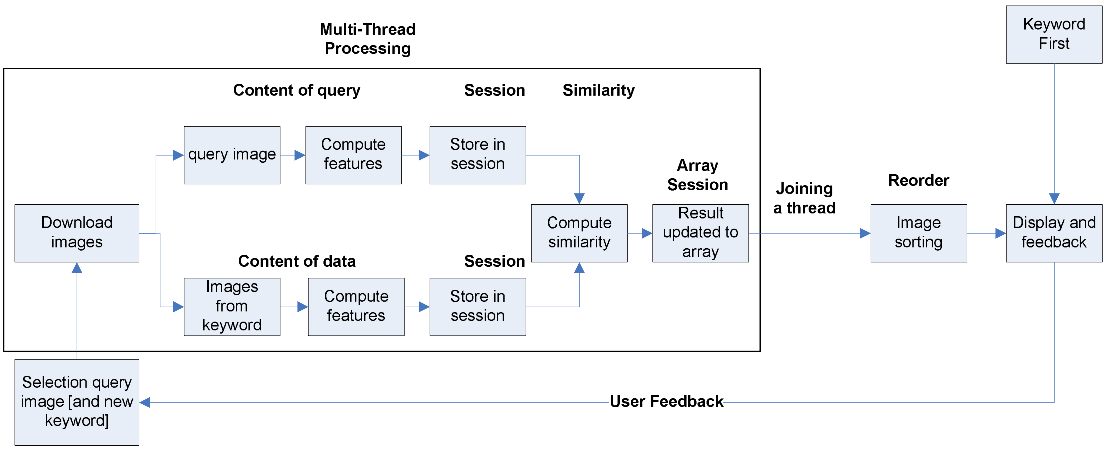

Fig. 2 An overview of an on-line CBIR system

2 The Proposed Framework

2.1 Preliminary

Before introducing our framework of multi-threading for a joint querying image search scheme, we will briefly examine the properties of the queries to be answered. The query modalities require different processing methods and support for user interaction. We can characterize query processing from a system perspective including text-based, content-based, composite, interactive-simple, and interactive-composite [19].Our retrieval model is interactive-composite because it integrates multi-model information (associated text, visual similarity, and user's feedbacks) for providing search results. We have developed a novel framework of real-time processing for an on-line CBIR system, using relevance images from Yahoo images search. Our method uses the following major steps: (a) Yahoo Images is first used to obtain a large number of images that are returned for a given text-based query; (b) The users can select any certain images to perform an update the input query for image similarity characterization; (c) A multi-threaded processing method is used to manage and perform data parallelism or loop-level parallelism such as downloading images, extraction of visual features and computation of visual similarity measures; (d) If necessary, users can also change a keyword before selecting a relevance image for the query; (e) The updated queries are further used to adaptively create a new answer for the next set of returned images according to the users' preferences. The overview of our scheme is shown in Fig. 2

2.2 A Framework Design for Multi-Thread Processing

The image indexation and similarity measure computation of images are complex processes and they are an obstacle for the development of a practical CBIR system. Especially, when it is developed based on a real-time process optimization approach. There are a number of papers concerning parallel computing for image processing [20] [21] [22] , For instance, Yongquan Lu, et al [23] presented a parallel technique to perform feature extraction and a similarity comparison of visual features, developed on cluster architecture. The experiments conducted show that a parallel computing technique can be applied that will significantly improve the performance of a retrieval system. Kao, et al [24] proposed a cluster platform, which supports the implementation of retrieval approaches used in CBIR systems. Their paper introduces the basic principles of image retrieval with dynamic feature extraction using cluster platform architecture. The main focus is workload balancing across the cluster with a scheduling heuristic and execution performance measurements of the implemented prototype. Although, cluster computing is popularly used in images retrieval approaches, it only attacks this problem at the macro level. Fortunately, with the increasing computational power of modern computers, some of the most time-consuming tasks in image indexing and retrieval are easily parallelized, so that the multi-core architecture in modern CPU and multi-threaded processing may be exploited to speed up image processing tasks. It is possible to incorporate an image analysis algorithm into the text-based image search engines such as Google, Yahoo, and Bing without degrading their response time. Multi-threading is not the same as distributed processing. Distributed processing which is sometimes called parallel processing and multi-threading are both techniques used to achieve parallelism (and can be used in combination) [25].

However, the development and implementation of an image search application is a complex process, as it requires an in-depth understanding of both the theory and practice of image processing. There still remains a large gap between theory and practice.

In this paper, a multi-threaded processing method is used to carry out parallel processing of multiple threads for a specific purpose. Multi-threading is a way to let programs do more than one thing at a time, implemented within a single program, and running on a single system. In order to utilize the threads more efficiently, the number of threads should be considered and they must technically be assigned to the correct parts of the program. We utilize the threads in a parallel process for an online CBIR application. The development of functions, classes, and objects in the program should not logically be designed as a sequence of steps. The following steps explain how the threads are applied in this system. First, they help improve the download speed for images from various sources according to the locations specified in the xml file that are returned from Yahoo BOSS API [31] after querying by the keyword. Second, they are used in calculating the image feature extraction. Lastly, they increase the speed of computing the similarity measure of feature vectors. The framework of multi-thread processing for the joint querying and relevance feedback scheme for an on-line image retrieval system is presented in Fig. 3.

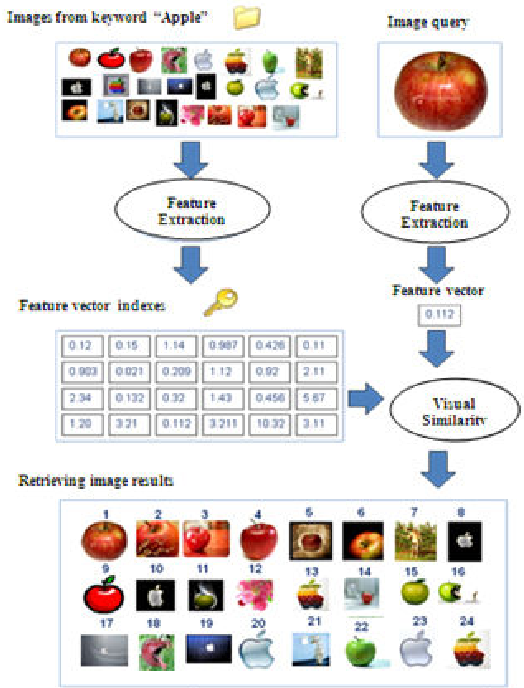

Fig. 3. A basic principle of joint querying for on-line CBIR system

An image list control receives the xml files that are returned from Yahoo BOSS API (See in Fig. 4). BOSS API stands for Build your Own Search Service. It is Yahoo!'s open search web services platform. BOSS gives the users access to Yahoo!'s investments in crawling and indexing, ranking relevancy algorithms, and powerful infrastructure. Lists of URL are obtained from the xml files. They are further displayed and used for downloading images from the hosts. An image download module is designed to work in a multithreaded process for downloading images from diverse sources.


```
<ysearchresponse responsecode="200">
<prevpage>
/ysearch/images/v1/apple?format=xml&count=1&appid=PRG2QiflkY1cA3fLzhgwlVlr48rjlVUmaiWa&start=0
</prevpage>
<nextpage>
/ysearch/images/v1/apple?format=xml&count=1&appid=PRG2QiflkY1cA3fLzhgwlVlr48rjlVUmaiWa&start=2
</nextpage>
<resultset_images count="1" start="1" totalhits="16836085" deephits="16836085">
<result>
<abstract>
Tras muchos años de disputas y acuerdos de concordia Apple Computer y Apple Corps la discografía que
gestiona los derechos de Los Beatles podrían estar a punto de llegar a un acuerdo
</abstract>
<clickurl>
http://lrd.yahooapis.com/
_ylc=X3oDMTRjMHJ2aTlsBF9TAzIwMjMxNTI3MDIEYXBwaWQDUFJHMlFpZklrWTFjQTNmTHpoZ3dsVmxyN
DhyamxWVW1haVdhBGNsaWVudANib3NzBHNlcnZpY2UDQk9TUwRzbGsDdGl0bGUEc3JjcHZpZANCYm43
SjBnZUF1MTdvY3JoeGZKLndIQVpkeTQzU2tvellHVUFDZG5k/SIG=1235rt557/**http%3A//
es.appleweblog.com/wp-content/uploads/2006/11/apple.jpg
</clickurl>
<date>2006/11/27</date>
<filename>apple jpg</filename>
<format>jpeg</format>
<height>330</height>
<mimetype>image/jpeg</mimetype>
<refererclickurl>
http://lrd.yahooapis.com/
_ylc=X3oDMTRjMHJ2aTlsBF9TAzIwMjMxNTI3MDIEYXBwaWQDUFJHMlFpZklrWTFjQTNmTHpoZ3dsVmxyN
DhyamxWVW1haVdhBGNsaWVudANib3NzBHNlcnZpY2UDQk9TUwRzbGsDdGl0bGUEc3JjcHZpZANCYm43
SjBnZUF1MTdvY3JoeGZKLndIQVpkeTQzU2tvellHVUFDZG5k/SIG=12sasnpj5/**http%3A//
es.appleweblog.com/2006/11/27/apple-y-los-beatles-a-punto-de-llegar-a-un-acuerdo
</refererclickurl>
<refererurl>
http://es.appleweblog.com/2006/11/27/apple-y-los-beatles-a-punto-de-llegar-a-un-acuerdo
</refererurl>
<size>19100</size>
<thumbnail_height>102</thumbnail_height>
<thumbnail_url>http://thm-a01.yimg.com/image/2f31a32a15522fac</thumbnail_url>
<thumbnail_width>140</thumbnail_width>
<title>apple jpg</title>
<url>
http://es.appleweblog.com/wp-content/uploads/2006/11/apple.jpg
</url>
<width>450</width>
</result>
</resultset_images>
</ysearchresponse>
```


Fig. 4 the XML result of query by “apple” from yahoo boss

It is controlled by an image search control module. The image search control module performs a very important function in the management of the system. It fully supports and controls all modules of the online CBIR system. It checks for errors, and the input/output status of each module. Most importantly, it efficiently supports the synchronization of multiple threads that perform image download and similarity measurement by the associated modules. The similarity measurement module performs the computation of the feature vectors and distance metrics of all images that are obtained from the image download module. The image download and similarity measurement modules work concurrently. The query results are recorded into a session of an array in sequential order. The image list object is responsible for the arrangement of all displayed images on the application

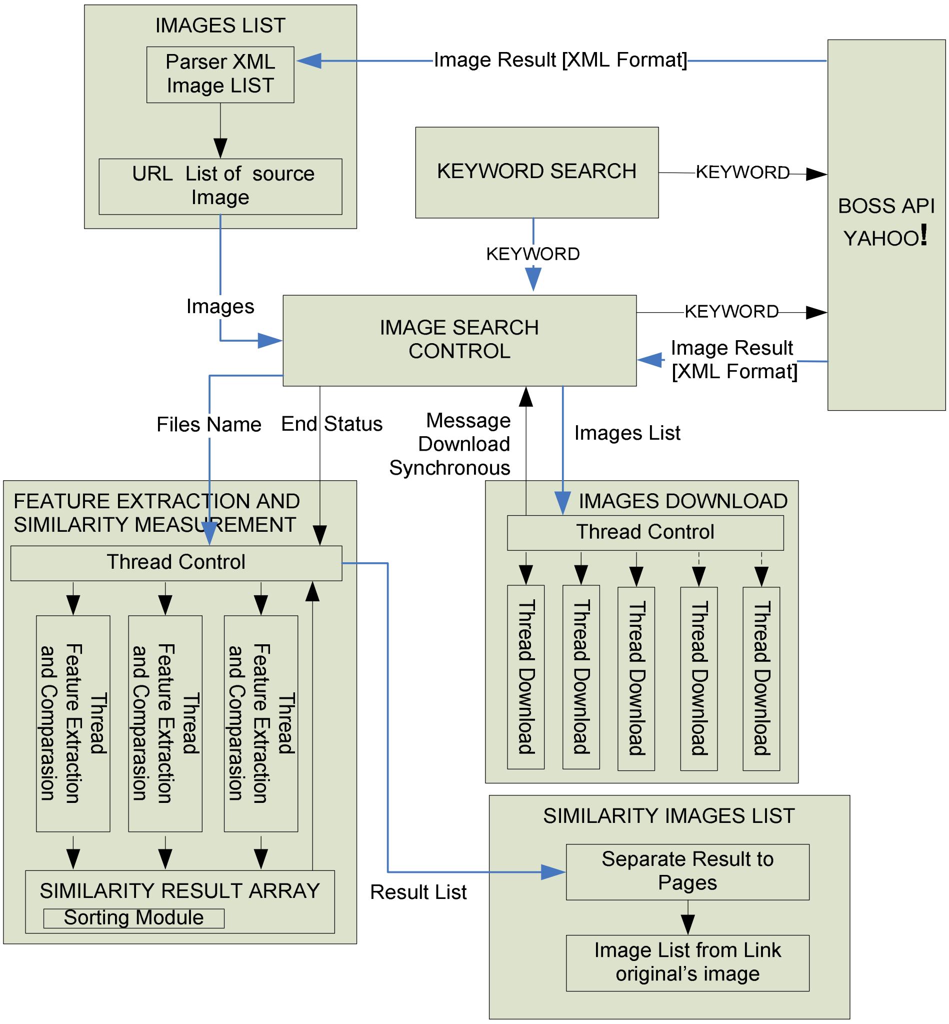

Fig. 5. The framework of multi-threaded processing for an on-line CBIR application

3 Feature Computation

3.1 Feature Extraction

There are various visual descriptors used to extract a low-level feature vector of an image [26][27][28][29][30]. However, in this paper, we used color descriptors for retrieving images. The color correlogram is an efficient feature extraction techniques used in content-based image retrieval (CBIR) systems. The technique, namely color correlogram, is widely used for finding the spatial correlation of each color in an image. It was introduced by Huang J. et al [17]. The technique was implemented and it was found that the retrieval performance of a color correlogram was better than the standard color histogram and the color coherence vector methods. However, the color correlogram is expensive to compute and the computation time of the correlogram is $O(m^2d)$. The authors also present a technique that captures the spatial correlation between identical colors called an autocorrelogram with a computation time of $O(md)$. However, an autocorrelogram only captures the distribution of each color in the image. The disadvantages are: 1) the color correlogram has computation complexity, and 2) the auto correlogram mainly captures the distribution of each color in the images. They mainly capture spatial information of the colors. In this section, we present an efficient color feature extraction algorithm for low-level feature similarity in query process, namely Auto Color Correlogram and Correlation (ACCC) [18], The retrieval performance is satisfactory and higher than the average precision of the retrieved images using autocorrelogram (AC). The ACCC is the integration of Autocorrelogram and Auto Color Correlation techniques [18]. It is a fast and robust algorithm for spatial color feature extraction for image indexing. Using this proposed technique, the correlation values of colors for each color distribution of an image are computed more efficiently. An auto color correlation (ACC) [18] expresses how to compute the mean color of all pixels of color C_j at a distance k-th from a pixel of color C_i in the image. Formally, the ACC of image $\{I(x,y),\ x = 1,2,\dots,M,\ y = 1,2,\dots,N\}$ is defined as

$$ACC(i,j,k) = MC_j\gamma^{(k)}_{C_i,C_j}(I) = \left\{r_{mc_j}\,\gamma^{(k)}_{C_i,C_j}(I),\ g_{mc_j}\gamma^{(k)}_{C_i,C_j}(I),\ b_{mc_j}\,\gamma^{(k)}_{C_i,C_j}(I) \mid c_i \neq c_j\right\} \quad (1)$$

Where the original image $I(x,y)$ is quantized to J colors $C_1, C_2, \dots, C_J$ and the distance between two pixels d ∈ [min\{M,N\}] is fixed a priori. Let MC_j is the color mean of color C_j from color C_i at distance d in an image I. The mean colors are computed as follows:

$$r_{mc_j}\gamma^{(d)}_{C_i,C_j} = \left(\prod_{i=1}^{N}\Gamma^{(d)}_{C_i,rC_j}(I)\right)^{1/N} \mid c_i \neq c_j$$

$$g_{mc_j}\gamma^{(d)}_{C_i,C_j} = \left(\prod_{i=1}^{N}\Gamma^{(d)}_{C_i,gC_j}(I)\right)^{1/N} \mid c_i \neq c_j$$

$$b_{mc_j}\gamma^{(d)}_{C_i,C_j} = \left(\prod_{i=1}^{N}\Gamma^{(d)}_{C_i,bC_j}(I)\right)^{1/N} \mid c_i \neq c_j \quad (2)$$

where $C_j \neq 0$ and N is the number of accounting color Cj from color Ci at distance k , defined by:

$$N = \Gamma^{(k)}_{C_i,C_j}(I) = \left\{\begin{matrix} P(x_1,y_1) \in C_i \mid P(x_2,y_2) \in C_i\,; \\ k = \min\{|x_1 - x_2|, |y_1 - y_2|\end{matrix}\right\} \quad (3)$$

Although ACC is able to find the local spatial correlation between colors by reducing the size of color correlogram from $O(m^2d)$ to $O(md)$, it does not consider the color distribution values of each color in an image. Autocorrelogram is an efficient algorithm to solve this problem. Thus, we propose an extended technique of ACC based on the autocorrelogram, namely Auto Color Correlogram and Correlation (ACCC). It not only captures the spatial correlation between identical colors but also computes the local spatial correlation between colors. The Auto Color Correlogram and Correlation is defined as

$$ACCC(i,j,k) = \left\{\gamma^{(k)}_{C_i}(I),\ MC_j\,\gamma^{(k)}_{C_i,VC_j}(I)\right\} \quad (4)$$

In order to gain a deeper understanding of the ACCC's computational procedure, it is described as follows.

```
Algorithm: Auto Color Correlation
For every K distance {
For every X position
For every Y position {
Ci ← current pixel
While (Cj ← Get neighbor pixel of Ci at distance K) {
For every color Cm {
If (Cm = Ci)
If ( Ci ≠ Cj){
colorCount[k][Ci]++
colorR[k][Ci] = colorR[k][Ci] + colorRCj
 colorG[k][Ci] = colorG[k][Ci]+ colorGCj
 colorB[k][Ci] = colorB[k][Ci] + colorBCj

ACC[k][Ci] :=
{ colorR[k][Ci]/(colorCount[k][Ci] * k * 8)
{ colorG[k][Ci]/(colorCount[k][Ci] * k * 8)
{ colorB[k][Ci]/(colorCount[k][Ci] * k * 8)
}else{
colorCount[k][Ci]++
color[k][Ci] = color[k][Ci] + 1
CC[k][Ci] = color[k][Ci] / (colorCount[k][Ci] * k*8)
 }}}}}
```

The descriptions of ACCC algorithm are presented in [18].

3.2 Similarity Measurement

The type of similarity measure to be considered depends on the technique used for feature extraction. The current section presents similarity measures for the proposed color descriptor in the case of image retrieval. Let the ACCC pairs for the m color bin be (α_i, β_i) in I and (α'_i, β'_i) in I'. The similarity of the images is measured as the distances between the AC's and ACC's $d(I, I')$ and is applied from [32] as follows:

$$d(I,I') = \lambda_1 \sum_{\forall i} \frac{|\alpha_i - \alpha'_i|}{1+\alpha_i+\alpha'_i} + \lambda_2 \sum_{\forall i} \frac{|\beta_i - \beta'_i|}{1+\beta_i+\beta'_i} \quad (5)$$

Where λ_1 and λ_2 are the similarity weighting constants of autocorrelogram and auto color correlation, respectively. In the experiments conducted, $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$. α_1 and α_1 are defined as follows:

$$\alpha_i = \gamma^{(k)}_{C_i}(I)$$

$$\beta_i = \left\{\begin{matrix} Avg\ \gamma^{(k)}_{C_i,VC_{jr}}(I), Avg\ \gamma^{(k)}_{C_i,VC_{jg}}(I), \\ Avg\ \gamma^{(k)}_{C_i,VC_{jb}}(I)\ |c_i \neq c_j \end{matrix}\right\} \quad (6)$$

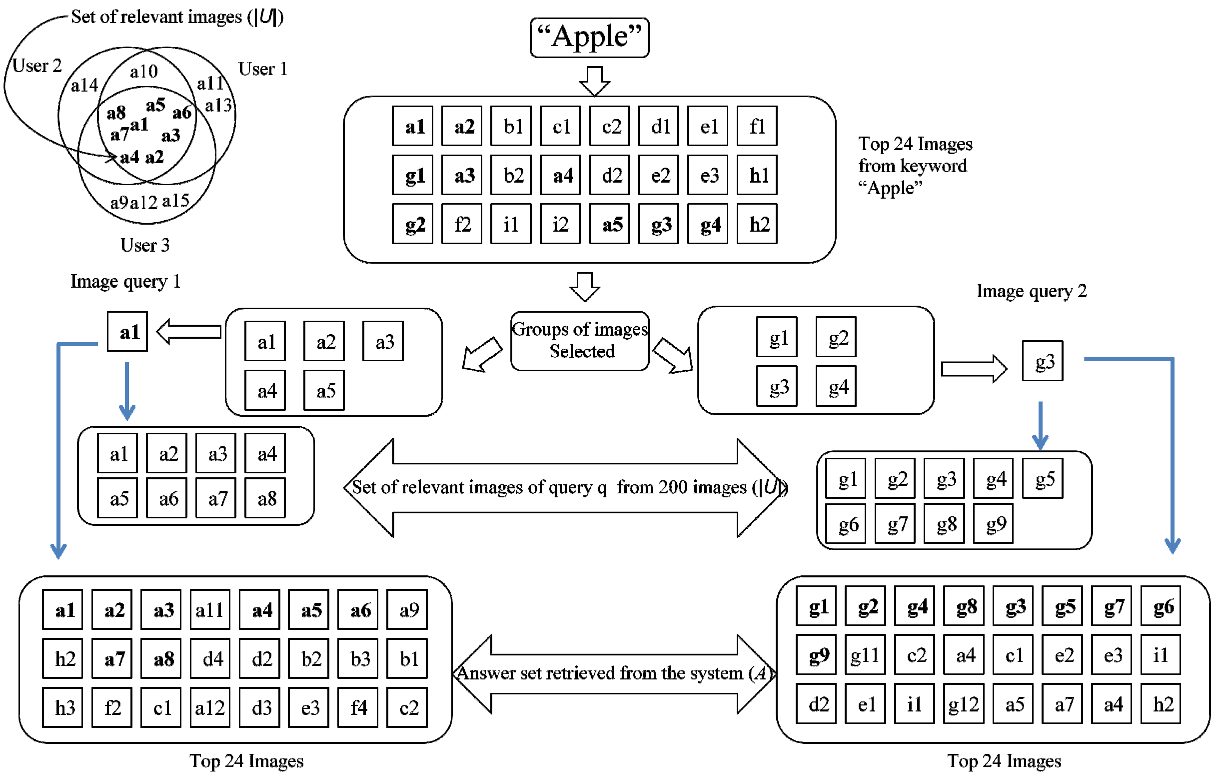

Fig. 6 The procedures of system evaluation

4 Experimental Result

We have implemented a joint querying image search scheme using the Yahoo image database based on the Yahoo BOSS' API. The application are developed by using Microsoft .NET and implemented on Quad-Core Intel Xeon processor E5310 1.60 GHz, 1066 MHz FSB 1 GB (2 x 512 MB) PC2-5300 DDR2, and tested on the Windows NT environment. The goal of this experiment is to show that relevant images can be found after a small number of iterations, the first round was used in this experiment. From the viewpoint of user interface design, precision and recall measures are less appropriate for assessing an interactive system [33]. To evaluate the performance of the system in terms of user feedback, user-orientation measures are used. There have been other design factors proposed such as relative recall, recall effort, coverage ratio, and novelty ratio [34]. In this experiment the coverage ratio measure is selected. Let R be the set of relevant images of query q and A be the answer set retrieved. Let |U| be the number of relevant images which are known to the user, where U∈R. The coverage ratio is the intersection of set A and U, $|R_k|$ be the number of images in this set. It is defined by equation 5.

$$Coverage\ (C_q) = \frac{|R_k|}{|U|} \qquad (7)$$

Let N_q is the number of keyword used. The average of coverage ratio is defined by equation 6.

$$Avg\ C_{(p)}\ \frac{1}{N_q}\sum_{i=1}^{N_q}\frac{|R_k|}{|U|} \qquad (8)$$

To conduct this experiment, Yahoo images search was first executed to obtain a large number of images returned by a given text-based query. The user selected a relevant image, specific to only one interaction with the user. Those images that were most similar to the new query image were returned. The retrieval performance in terms of coverage ratio of the proposed system was compared to the traditional Yahoo text-based search results. The average coverage ratio was generated based on the ACCC algorithms using over forty-nine test keywords in five categories including animal, fruit, sunset/ sunrise, nature, and landscape. The results are presented in table 2 and fig. 7.

The number of threads versus time on the proposed framework for an image retrieval process that includes image downloading, feature extraction and image comparison are shown in Table 1. The most suitable number of threads was in between 50 and 200 for overall processes in this experiment.

The data in table 2 shows that our scheme can increase the efficiency of image retrieval from the Yahoo image database. Using the combination of text and a user's feedback for an image search, the images that do not correspond with the category are filtered out. It also decreases the opportunity of the images in other categories to be retrieved. In the experiment, we used two sample images obtained from the keyword search to test querying images for evaluating the performance of the system. It is shown in fig. 8.

Platform	Number of Threads						
	1	10	25	50	75	100	200
Multi-Core	568.5 ±15.7	145.9 ±11.7	71.0 ±5.7	62.0 ±7.3	62.1 ±6.2	62.6 ±7.3	62.8 ±8.5

Table 1. Computational average time in second of the number of threads corresponding in image retrieval process (mean ± std-dev)

Sample images	Schemes	Coverage Ratio				
		Animal	Fruit	Sunset/ Sunrise	Nature	Landscape
Q1	Keyword	0.44	0.66	0.43	0.35	0.43
	Our Scheme	0.71	0.79	0.62	0.64	0.69
Q2	Keyword	0.31	0.46	0.22	0.33	0.30
	Our Scheme	0.65	0.71	0.65	0.59	0.65
Avg.	Keyword	0.38	0.56	0.33	0.34	0.37
	Our Scheme	0.68	0.75	0.63	0.62	0.67

Table 2. Coverage ratio average of the top 24 of 200 retrieved images

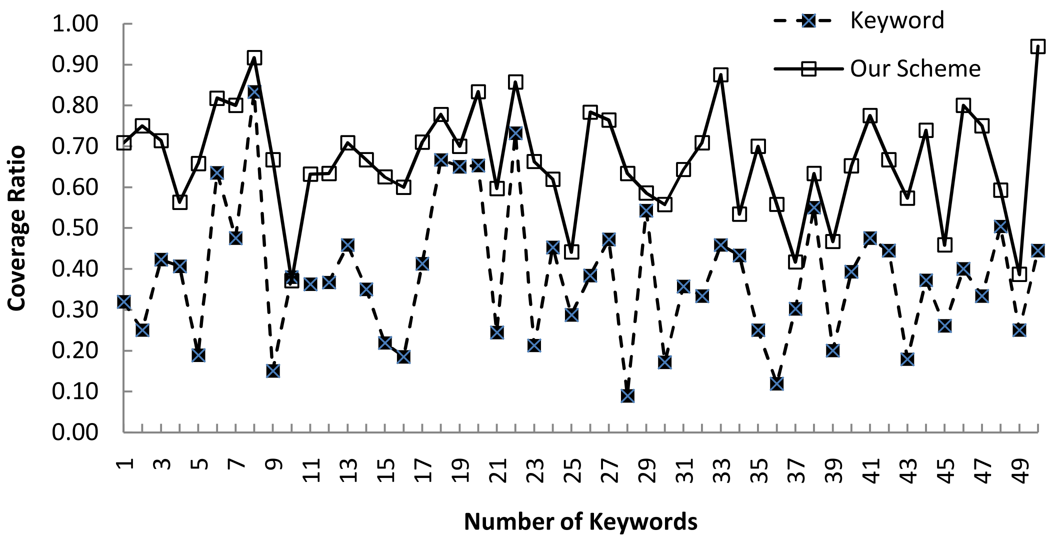

Fig. 7. Comparison of the traditional Yahoo text-based search and our scheme with the ACCC algorithm

5 Conclusions

This paper proposed an on-line content-based image retrieval system using joint querying and relevance feedback scheme. The proposed framework can be efficiently merged textual and image features for image retrieval systems. To incorporate an image analysis algorithm into the text-based image search engines without degrading their response time, the framework of multi-threaded processing is developed. In a high-level semantic retrieval system, we utilized the search engine to retrieve a large number of images using a given text-based query. In low-level image retrieval process, the system provides a similar image search function for the user to update the input query for image similarity characterization. We also introduced fast and efficient color feature extraction namely auto color correlogram and correlation (ACCC) for efficiently extracting and indexing low-level features of images. We investigated the performance of our retrieval scheme. The results clearly show that our retrieval procedure is effective for existing on-line image search engine. Moreover, it can overcome the common limitation of the existing keyword-based and content-based of image retrieval systems.

In a future work based on this study, we will determine the suitable number of threads tested in different processor architectures or platforms in parallel computing for an image retrieval task and improve the performance in term of speed and accuracy of color feature extraction based on ACCC algorithm.

References:

[1] J. Ashley *et al.*, The query by image content (QBIC) system, *in proceeding of the ACM SIGMOD.* 1995.

[2] J. R. Smith, S. Chang, VisualSEEk: a fully automated content-based image query system, in proceeding of the ACM on Multimedia, 1997, pp. 87-98.

[3] C. Carson et al., Blobworld: A System for Region-Based Image Indexing and Retrieval, Springer LNCS. Vol. 1614/1999, pp. 660, 1999.

[4] M. Beigi et al, MetaSEEk: A Content-Based Meta-Search Engine for Images, in proceeding of the SPIE Storage and Retrieval for Image and Video Databases, 1997.

[5] T. Pavlidis, Limitations of content-based Image Retrieval. in proceeding of the ICPR, 2008.

[6] A. Hanjalic et al., The Holy Grail of Multimedia InformationRetrieval: SoClose or Yet So Far Away?, In proceeding of the IEEE multimedia, Vol. 96, No. 4, April 2008.

[7] F. Jing et al., A Unified Framework for Image Retrieval Using Keyword and Visual Features, IEEE Trans. on Image Processing, Vol. 14, No. 7, pp. 979–989, July 2005.

[8] H. T. Shen., Speed up interactive image retrieval, The VLDB Journal, Springer, Vol. 18, No. 1, January 2009.

[9] G. Aggarwal, T. V. Ashwin, S. Ghosal, An Image Retrieval System with Automatic Query Modification. IEEE. Trans. on Multimedia. Vol. 4, No. 2, pp. 201–214, 2002.

[10] E. Cheng el al., Search Result Clustering Based Relevance Feedback for Web Image Retrieval, in Proceeding of the ICASSP, pp. I-961-I-964, 2007.

[11] F. Marin, and B. Nozha, Interactive Remote-Sensing Image Retrieval Using Active Relevance Feedback. IEEE Trans. on Geoscience and Remote Sensing, Vol. 45, No. 4, pp. 818–826, 2007.

[12] http://similar-images.googlelabs.com/

[13] http://www.googlelabs.com/

[14] T. Gevers , F. Aldershoff and J. M.Geusebroek, Integrating Visual and Textual Cues for Image Classification, Springer LNCS. Vol. 1929/2000, pp. 133, 147, 2000.

[15] Y. Lu et al., A unified framework for semantics and feature based relevance feedback in image retrieval systems, in proceeding of the ACM on multimedia, pp. 31-38, 2000.

[16] E. Chang et al., CBSA: content-based soft annotation for multimodal image retrieval using Bayes point machines, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 13, No. 1, pp. 26-38, January 2003.

[17] J. Huang et al, Spatial Color Indexing and Applications, in proceeding of Sixth International Conference on Computer Vision, pp. 606 – 607, 1998.

[18] T. Anucha el al., Spatial Color Indexing using ACC Algorithms, in Proceeding of the ICT&KE, pp. 113-117, 2009.

[19] R Dattaet al., Image retrieval: Ideas, influences, and trends of the new age, ACM Computing Surveys, Vol. 40, No. 2, 2008.

[20] T. Anucha, J. Niapt, P. Wichian, Using Web Portal for 3D Grid-Based Rendering, Proceeding of the ICT&KM, pp. 78-86, 2008.

[21] A. Chong, A. Sourin, K. Levinski, Grid-Based Computer Animation Rendering, Proceeding of the GRAPHITE, pp. 39-47, 2006.

[22] L. Guoqing, M. Yan, W. Jian, L. Dingsheng, Preliminary Through-Out Research on Parallel-Based Remote Sensing Image Processing, Springer LNCS, Computer Science, pp. 880-883 2006.

[23] L. Yongquan, G. Pengdong, L. Rui, S. Zhiwu, Y. Wenhua, Study of content-based image retrieval using parallel computing technique, Proceedings of the ATIP's, pp. 186–191, 2007.

[24] O. Kao, G. Steinert, F. Drews, Scheduling aspects for image retrieval in cluster-based image databases, Proceedings of First IEEE/ACM. Cluster Computing and the Grid, pp. 329 – 336, 2001.

[25] Multi-Threading in IDL. : ITT Visual Information Solutions , 2006. www.ittvis.com

[26] L. Zhang, L. Xi and B. Zhou, Image Retrieval Method Based on Entropy and Fractal Coding, WSEAS Transactions on Systems, Vol. 7, Issue 4, April 2008.

[27] K.-A. Wang et al., Implementation of an Image Retrieval System Using Wavelet Decomposition and Gradient Variation, WSEAS Transactions on Systems, Vol. 7, Issue 6, June 2008.

[28] A. Khaparde, M. M. Vilatha, M. B. L. Manasa, P. A Babu, S. P. Kumar, FastICA Algorithm for

[29] the Separation of Mixed Images, WSEAS Transactions Signal Processing, Vol. 4, Issue 5, May 2008.

[30] Z. Ye, H. Mohamadian, Y. Ye, Adaptive Approach on Trimulus Color Image Enhancement and Information Theory Based Quantitative Measuring, WSEAS Transactions Signal Processing, Vol. 4, Issue 1, January 2008.

[31] A. Kulkarni, H. Gunturu, S. Datla, Association-Based Image Retrieval, WSEAS Transactions Signal Processing, Vol. 4, Issue 4, April 2008.

[32] http://developer.yahoo.com/search/boss/

[33] G. Qiu, Color Image Indexing using BTC, IEEE Trans. on Image Processing, Vol. 12, No. 1, pp. 93–101, 2003.

[34] B.-Y. Ricardo, and R.-N. Berthier, Modern Information Retrieval, ACM Press Book, 1999.

[35] R. K. Robert, Information Storage and Retrieval, John Wiley & Sons, Inc. 1993.

Fig. 8. Images corresponding to two image queries obtained from the keyword search