

# Joint querying and relevance feedback scheme for an on-line image retrieval system

WICHIAN PRAMCHAUSAWADI, ANUCHA TUNGKASTHAN

Graduate School of Information Technology in Business

Siam University

235 Petkasem Rd., Phasicharoen, Bangkok, 10160

THAILAND

[wichian@siam.edu](mailto:wichian@siam.edu), [aimdala@hotmail.com](mailto:aimdala@hotmail.com)

*Abstract:* - This paper presents a joint querying and relevance feedback scheme based on both high-level and low-level features of images for an on-line content-based image retrieval system. In a high-level semantic retrieval system, 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. 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

## 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. 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).



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  $O(m2d)$  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.

## 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

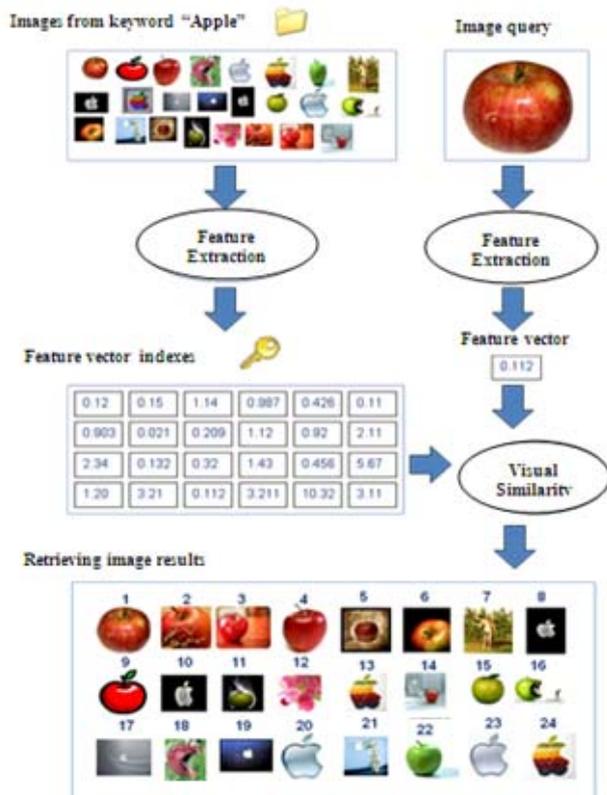


Fig. 2. A Basic principle of an on-line CBIR system

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. 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. 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 [20] 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. The thread control and the tasks inside of a thread for retrieving images are presented in Fig. 4.

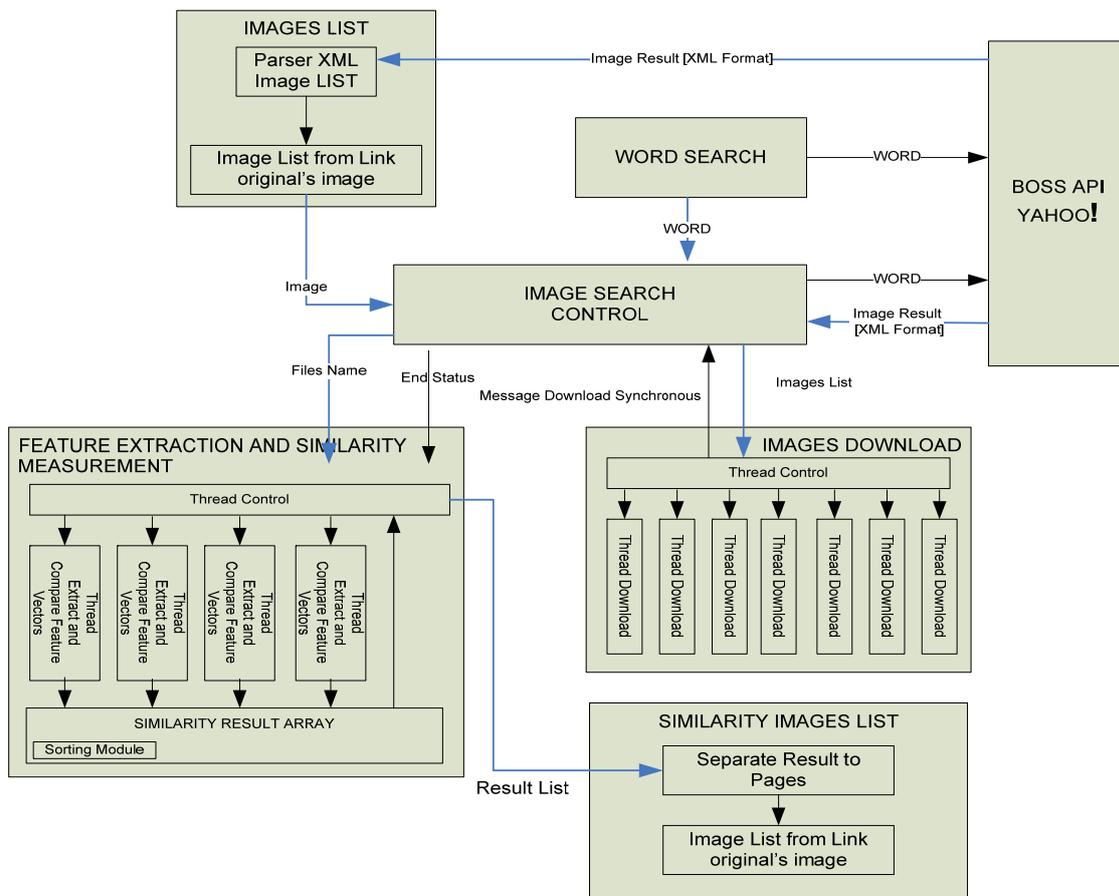


Fig. 3 The framework of multi-threaded processing for an on-line CBIR application

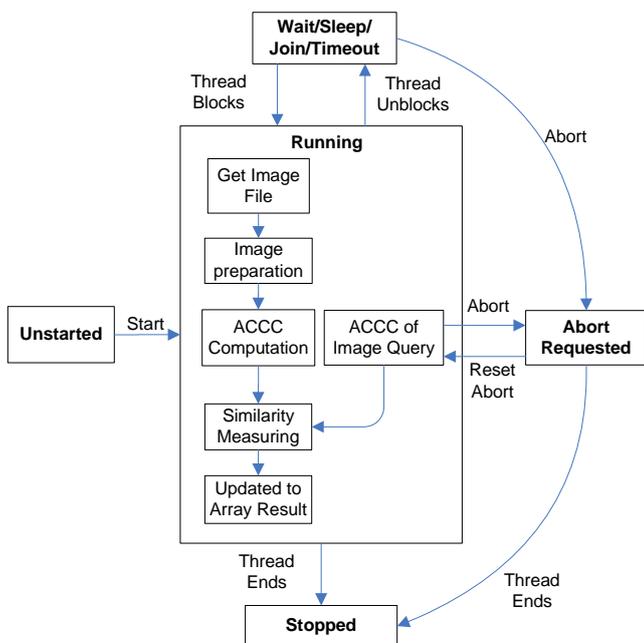


Fig. 4 The thread control and the tasks insight of a thread for retrieving images.

The image list control receives the xml files that are returned from Yahoo BOSS API. 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 [20]. The lists of URL can be 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. It is controlled by an image search control module. The image search control module performs a very important function for thread 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.

### 3 Feature Computation

#### 3.1 Feature Extraction

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. It is a fast and robust algorithm for spatial color feature extraction for image indexing. It does not only capture the spatial correlation between identical colors but also computes the local spatial correlation between colors. Using this proposed technique, the correlation values of colors for each color distribution of an image are computed more efficiently. The Auto Color Correlogram and Correlation is defined as follows:

$$ACCC(I, j, k) = \left\{ \gamma_{c_i}^{(k)}(I), Avg \gamma_{c_i, v_{c_j}}^{(k)}(I) \right\} \quad (1)$$

Where the original image  $I(x, y)$  is quantized to  $m$  colors  $C_1, C_2, \dots, C_m$  and the distance between two pixels  $k \in [\min\{M, N\}]$  is fixed a priori. Let  $v_{c_j}$  is the RGB value of color  $m$  in an image  $I$ . The ACCC values are computed as follows:

$$\begin{aligned} \overline{r\gamma_{c_i, c_j}^{(k)}}(I) &= \frac{\Gamma_{c_i, v_{c_j}}^{(k)}(I)}{\Gamma_{c_i, c_j}^{(k)}(I)} \mid c_i \neq c_j \\ \overline{g\gamma_{c_i, c_j}^{(k)}}(I) &= \frac{\Gamma_{c_i, v_{c_j}}^{(k)}(I)}{\Gamma_{c_i, c_j}^{(k)}(I)} \mid c_i \neq c_j \\ \overline{b\gamma_{c_i, c_j}^{(k)}}(I) &= \frac{\Gamma_{c_i, v_{c_j}}^{(k)}(I)}{\Gamma_{c_i, c_j}^{(k)}(I)} \mid c_i \neq c_j \end{aligned} \quad (2)$$

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

#### 3.2 Visual Similarity

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  $(\alpha_i, \beta_i)$  in  $I$  and  $(\alpha'_i, \beta'_i)$  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 [21] as follows:

$$d(I, I') = \lambda_1 \sum_{v=1}^m \frac{|\alpha_i - \alpha'_i|}{1 + \alpha_i + \alpha'_i} + \lambda_2 \sum_{v=1}^m \frac{|\beta_i - \beta'_i|}{1 + \beta_i + \beta'_i} \quad (3)$$

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

$$\begin{aligned} \alpha_i &= \gamma_{c_i}^{(k)}(I) \\ \beta_i &= \left\{ \begin{array}{l} Avg \gamma_{c_i, v_{c_j}}^{(k)}(I), Avg \gamma_{c_i, v_{c_j}}^{(k)}(I) \\ Avg \gamma_{c_i, v_{c_j}}^{(k)}(I) \mid c_i \neq c_j \end{array} \right\} \end{aligned} \quad (4)$$

### 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 [22]. 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 [23]. 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  $U \in R$ . 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.

$$Coverage(C_q) = \frac{|R_k|}{|U|} \quad (5)$$

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

$$Avg C_{(Q)} = \frac{1}{N_q} \sum_{i=1}^{N_q} \frac{|R_i|}{|U|} \quad (6)$$

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, nature, and landscape. The results are presented in table 2 and fig. 5.

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.

| Platform   | Number of Threads |             |             |           |                  |                  |                  |                  |
|------------|-------------------|-------------|-------------|-----------|------------------|------------------|------------------|------------------|
|            | 1                 | 5           | 10          | 25        | 50               | 75               | 100              | 200              |
| Multi-Core | 568.5 ±15.7       | 201.7 ±15.1 | 145.9 ±11.7 | 71.0 ±5.7 | <b>62.0 ±7.3</b> | <b>62.1 ±6.2</b> | <b>62.6 ±7.3</b> | <b>62.8 ±8.5</b> |

Table 1 Computational average time in second(s) of the number of threads corresponding in multi-core platform (mean ± std-dev)

| Sample images | Schemes    | Coverage Ratio |       |                 |        |           |
|---------------|------------|----------------|-------|-----------------|--------|-----------|
|               |            | Animal         | Fruit | Sunset/ Sunrise | Nature | Landscape |
| Query 1       | Keyword    | 0.44           | 0.66  | 0.43            | 0.35   | 0.43      |
|               | Our Scheme | 0.71           | 0.79  | 0.62            | 0.64   | 0.69      |
| Query 2       | 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

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. 6.

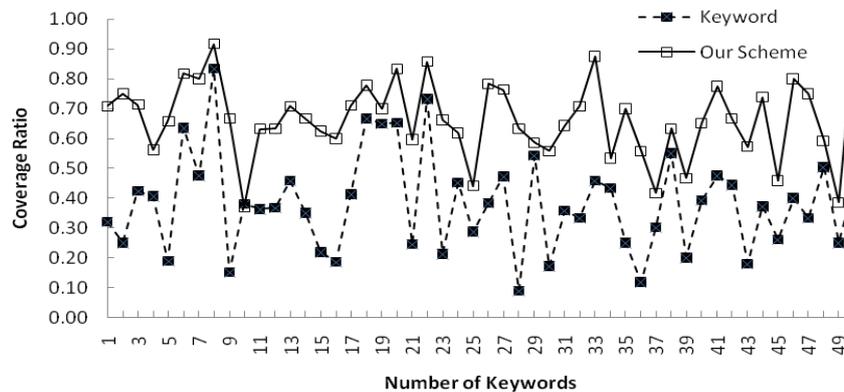


Fig. 5 comparison of the traditional Yahoo text-based search and our scheme with the ACCC algorithm

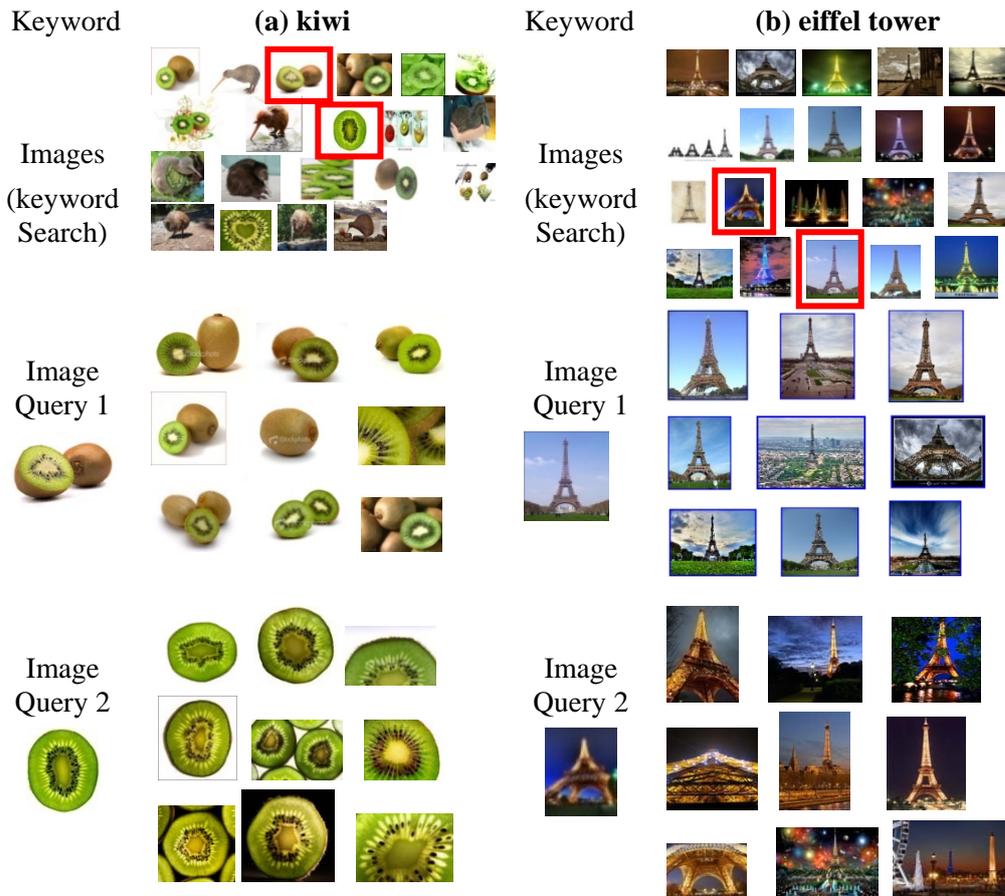


Fig. 6 Images corresponding to two image queries obtained from user's feedback

## 5 Conclusions

In this paper, a joint querying and relevant feedback scheme based on both high-level and low-level features of images for on-line content-based image retrieval system is proposed. 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 Information Retrieval: 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. Datta *et al.*, Image retrieval: Ideas, influences, and trends of the new age, *ACM Computing Surveys*, Vol. 40, No. 2, 2008.
- [20] <http://developer.yahoo.com/search/boss/>
- [21] G. Qiu, Color Image Indexing using BTC, *IEEE Trans. on Image Processing*, Vol. 12, No. 1, pp. 93–101, 2003.
- [22] B.-Y. Ricardo, and R.-N. Berthier, Modern Information Retrieval, *ACM Press Book*, 1999.
- [23] R. K. Robert, Information Storage and Retrieval, *John Wiley & Sons, Inc.* 1993.