Content Based Web Image Retrieval System Using Both MPEG-7 Visual Descriptors and Textual Information

Joohyoun Park and Jongho Nang

Dept. of Computer Science and Engineering, Sogang University, 1, ShinsuDong, MapoGu, Seoul 121-742, Korea
{parkjh, jhnang}@sogang.ac.kr

Abstract. This paper introduces a complete content based web image retrieval system by which images on WWW are automatically collected, searched and browsed using both visual and textual features. To improve the quality of search results and the speed of retrieval, we propose two new algorithms such as a keyword selection algorithm using visual features as well as the layout of web page, and a k-NN search algorithm based on the hierarchical bitmap index [17] using multiple features with dynamically updated weights. Moreover, these algorithms are adjusted for the MPEG-7 visual descriptors [14] that are used to represent the visual features of image in our system. Experimental results of keyword selection and image retrieval show the superiority of proposed algorithms and a couple of visual interfaces of the system are presented to help understanding some retrieval cases.

Keywords: Content based image retrieval, auto-annotation.

1 Introduction

Advent of new technologies in WWW (World Wide Web) and personal devices such as digital camera and mobile phone lead to increase the number of images on the WWW dramatically. Consequently, the needs of efficient searching by example or keyword have been increased as well. To fulfill these needs, there are three main issues should be considered carefully.

The first issue is how to annotate images collected from WWW automatically. There were some researches [1-3] which describe the problem of the image auto-annotation as a supervised or an unsupervised learning problem which builds up the relationship between visual features and concepts (textual features). Unfortunately, the annotations which generated by this approach would not describe the image content accurately because of the problem called "Semantic Gap [4]". Even though the images in web pages can be annotated and assigned to the images automatically by analyzing the layout on web pages where the descriptive texts are staying close to the images [5-8], it would produce many irrelevant annotations as well as relevant

T.-J. Cham et al. (Eds.): MMM 2007, LNCS 4351, Part I, pp. 659 – 669, 2007.

ones because of the lack of measures which could evaluates the degree of relevance between the surrounding texts and the images. Another issue would by the way to define the similarity of images, which is the basis of CBIR (Content-Based Image Retrieval). This issue may include which features are used – in broad sense, features may include both textual and visual features – and how to calculate the distance between images. Several studies [9-12], which proposed their own visual features and similarity measures, have been made on CBIR. Final issue is how to reduce the search time which is incurred by the high dimensionality of features. To make the system scalable to large set of images, the use of efficient high dimensional indexing method needs to be considered seriously.

In this paper, the content based web image retrieval system using both MPEG-7 visual descriptors [14] and textual information with sufficient consideration for the above three issues will be introduced. There are three main components in the system such as the web image miner, the search server, and the search client. The web image miner periodically collects images on the WWW and extracts the visual and textual features from those images. The textual features are selected using both the visual features and the layout of web pages in order to improve the correctness of keyword selection. The collected images and the features extracted from those images are delivered to the database manager in the search server, which manages the three databases such as an image database, a keyword database, and a visual feature database. For efficient retrieval by combining visual and textual features, they are indexed together by the HBI (Hierarchical Bitmap Indexing) [18], an efficient high dimensional indexing method. Since all features must be represented as vector form to index it, the way to convert each feature to vector form should be considered. Based on theses databases, every image in the image database is ranked by the search engine according to the query object which is generated by the search client.

2 System Architecture

The system consists of three major components as shown in <Fig. 1>. The first component is the web image miner consists of three tools such as an image collector and a keyword extractor. The image collector periodically crawls in the WWW and collects image and the words around that image. Then MPEG-7 visual descriptors [14] would be extracted from the images and some keywords for the images are selected by the keyword extractor. The second component is the search server which consists of a database manager and a search engine. The database manager manages visual features, textual features, and images and indexes them for efficient retrieval. Based on these databases, the images in the Image database are ranked by the search engine according to the visual or textual query which is sent from the search client. The third component is the search client which generates a query object and helps to browse the image from the results.

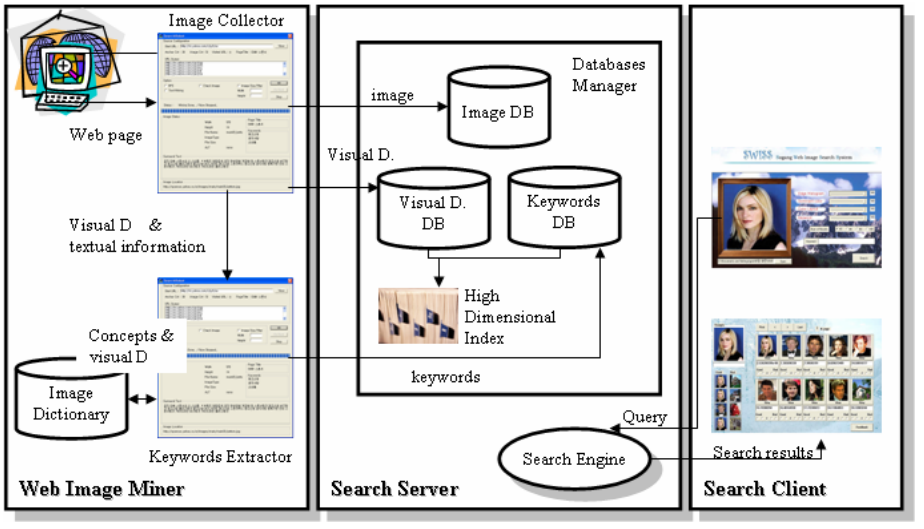


Fig. 1. The architecture of content based web image retrieval system which consists of 3 components such as web image miner, search server, and search client

3 Keyword Selection Algorithm

3.1 The Use of Image Dictionary

The meaning of Image Dictionary is the data structure which represents the relationship between the visual information and the concept (textual information). This relationship could be built up by the following learning process, which is similar to [3].

First, many sample images with manual annotations were collected in order to learn the concepts associated with the visual information. To remove the noises which were incurred by complicated images with multi-objects, each sample image is segmented into 3x3 uniform blocks, which are defined in MPEG-7 visual descriptors [14] such as *dominant color*, *color layout*, and *edge histogram* are extracted from. Based on these features, each block is clustered by *k-means clustering* algorithm with equal weights. Then each cluster has the blocks with similar visual properties and with the words annotated manually at the image preparation step. Finally, the representative keywords of each cluster are selected by the frequency of the words annotated to the blocks in the cluster.

3.2 Keyword Selection Algorithm

All words in the web page may not be evenly relevant to the image content. That is, the words with specific HTML tags could be more relevant than all other words in the web page. For example, according to the weighting scheme in [5], the words closer to the image or appearing with *src*, *alt* fields of the *img* tag, *title*, and *headers* may have higher importance as compared to other words. However, some words with higher

weights may not be relevant to the image content because the weights are evaluated by analyzing the layout of web page not the image content.

<Fig 2> shows the process of the proposed keyword selection algorithm to cope with the above problem. Initially, a HTML document is parsed into an image and its textual information (surrounding texts, pairs of word and its tag). The candidate keyword selector generates the pairs of candidate keyword and its weight from the textual information based on the weighting scheme in [5]. Furthermore, the image concept extractor analyzes the image to find the concepts associated to the image. Finally, the keyword selector with WordNet [16] filters out some irrelevant candidate keywords by comparing with the concepts associated to the image. The detail of the filtering process is as follows;

Assume that the number of the candidate keywords and the number of concept is l and m respectively. For each candidate keyword k_i $(1 \le i \le l)$, its final weight w'_i is calculated as follows;

$$w'_i = (1-\alpha)\cdot w_i + \alpha \cdot s_i \quad , (0 \le \alpha \le 1, 1 \le i \le l)$$
$$where, s_i = \max\left\{ \frac{w_j^c}{d_{i,j}} \middle| 1 \le j \le m \right\} \tag{1}$$

Note that w_i is the weight for the i-th candidate keyword and w_j^c is the weight for the j-th concept. $d_{i,j}$ means the length of the shortest path between k_i and the j-th concept in the word graph of WordNet[16]. Also, α controls relative importance of the visual features compared to the layout of web page. Top 5 words with higher weights will be selected as the final keywords for the image.

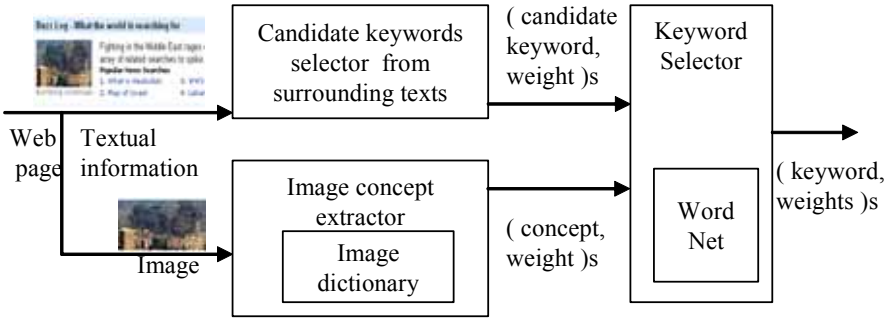


Fig. 2. The process of keyword selection

4 Content Based Image Retrieval

In this section, we will show how to represent textual features and visual features to vector form and how to index these feature vectors. We also discuss how to retrieve images based on the hierarchical bitmap index using multiple features with dynamically updated weights.

4.1 Vector Representation of Textual and Visual Features

4.1.1 Generating Textual Feature Vectors

As the results of the web image miner, each collected image has some keywords with their weights. Then, we can easily create the term matrix A (m x n), of which an element a_{ij} represents the weight of the j-th word in the i-th image. Note that m is the number of collected images and n is the number of words which can be used as keyword. From this matrix, we can use the vector a_i as the textual vector for the i-th image. It works but it can not resolve two problems that different words can be used to express the same concepts and the dimensionality of vector is too high. As the solution of these problems, we use an existing method known *Latent Semantic Analysis* (LSA) [5], which is commonly used in text analysis.

LSA decomposes the matrix A into three matrices U, S, and V by the singular value decomposition (SVD), $A = USV^T$, where $U \in \Re^{m \times k}$, $S \in \Re^{k \times k}$, $V \in \Re^{n \times k}$, and $U^T U = V^T V = I$. This operation reduces the dimension of the term vector by k dimension and captures statistically the semantic association across the terms in the set of terms with size n. Then the vector $u_i \ (1 \le i \le m)$, the i-th row of the matrix U, can be used as the textual vector for the i-th image with k dimension.

4.1.2 Generating Visual Feature Vectors

As Visual features, an image is represented as a subset of 9 visual descriptors which are defined in the visual part of the MPEG-7[14]. According to [15], the best descriptors for these combinations are *dominant color*, *color layout*, *edge histogram*, and *texture browsing* in terms of statistical properties for the judgement of the quality of descriptors such as redundancy, sensitivity, and completeness. *Texture browsing* is excluded from these descriptors because the general usage of it is not comparing of two images but browsing of images with similar perceptual properties. Finally, *dominant color*, *color layout*, and *edge histogram* are used.

In the MPEG-7 visual part of eXperience Model (XM) [13], the special metric of each descriptor is also defined. Therefore, it is necessary to check whether the data space where each descriptor is represented as vector space or not to index it. *Color layout* and *edge histogram* can be indexed without any modification because their metrics are *Euclidean distance* or *Manhattan distance* respectively. However, *dominant color* can not be indexed because its metric do not satisfy the properties of vector space or metric space. Consequently, it has necessitated a slight modification. Even though the definition and the metric function of *dominant color* looks complicate, it could be represented as the form of quantized color histogram with *Euclidean distance* [18].

4.2 Content Based Image Retrieval Using Visual and Textual Feature Vectors

To describe the way to calculate the distance of two images, it is necessary to formalize an image as visual and textual features. Consider an image database

Λ ($\Lambda = \{ o_i \mid 1 \le i \le n \}$, where o_i is the i^{th} image object.) with n image objects. An image object o_i is represented as a combination of feature vector as follows;

$$o_i = [t_i, d_i, c_i, e_i] \tag{2}$$

Note that t_i is a vector of textual feature and d_i, c_i, e_i are the vectors of *dominant color*, *color layout*, and *edge histogram* respectively associated with the image o_i. Then, total distance between the two images o_i and o_j, $D(o_i, o_j)$ is could be defined as follows;

$$\begin{aligned} &D(o_i,o_j) = \sum_{k=t,d,c,e} w_k \cdot GausNorm(D(k_i,k_j)) \quad (w_t + w_d + w_c + w_e = 1) \\ &where,\ D(t_i,t_j) = L_2(t_i,t_j),\ D(d_i,d_j) = L_2(d_i,d_j),\ D(c_i,c_j) = L_2(c_i^Y, c_j^Y) + L_2(c_i^{Cb}, c_j^{Cb}) + L_2(c_i^{Cr}, c_j^{Cr}), \\ &and \quad D(e_i,e_j) = L_1(e_i^l, e_j^l) + 5 \cdot L_1(e_i^g, e_j^g) + L_1(e_i^s, e_j^s) \end{aligned} \tag{3}$$

Note that *GausNorm* means Gaussian Normalization which normalized the distance of each feature within [0, 1]. To keep the original metrics defined in the MPEG-7 visual part of XM, the vectors of *color layout* and *edge histogram* must be split into 3 sub-vectors respectively before the distances are calculated. That is, c_i is split into the DCT coefficients for the luminance c_i^Y, and c_i^{Cb}, c_i^{Cr} for the chrominance. e_i is also split into the local edge histogram e_i^l, the global edge histogram e_i^g, and the semi global histogram e_i^s.

Similarity search problem in Λ can be formulated as a k-NN (Nearest Neighbor) problem because the distance measure between two images is defined. Also, the hierarchical bitmap indexing (HBI) [17] method is applied to solve the problem incurred by high dimensionality of features. With HBI, each feature vector is represented as a compact approximation and it reduce the time to calculate the distance of two images. The most irrelevant images can be filtered out during the process of scanning these approximations.

Let $B_p(\cdot)$ be the approximation of $L_p(\cdot)$ calculated using bitmap index. Then $D'(o_i, o_j)$, the approximation of the distance between the two images o_i and o_j, can be calculated as follows;

$$\begin{aligned} &D'(o_i,o_j) = \sum_{k=t,d,c,e} w_k \cdot GausNorm(D'(k_i,k_j)) \quad (w_t + w_d + w_c + w_e = 1) \\ &where,\ D'(t_i,t_j) = B_2(t_i,t_j),\ D'(d_i,d_j) = B_2(d_i,d_j), \\ &D'(c_i,c_j) = B_2(c_i^Y, c_j^Y) + B_2(c_i^{Cb}, c_j^{Cb}) + B_2(c_i^{Cr}, c_j^{Cr}), \\ &and \quad D'(e_i,e_j) = B_1(e_i^l, e_j^l) + 5 \cdot B_1(e_i^g, e_j^g) + B_1(e_i^s, e_j^s) \end{aligned} \tag{4}$$

According to [17], $L_p(v_1, v_2)$ is always bigger or equal than $B_p(v_1, v_2)$ for any vector v_1, v_2. It implies $D(k_i, k_j) \ge D'(k_i, k_j)$, where k=t, d, c, e. Therefore, it always satisfies the condition $D(o_i, o_j) \ge D'(o_i, o_j)$. From this property, k-NN search algorithm for this CBIR system as shown in <Fig. 3> can be created. In this algorithm, the candidate set could not be generated completely during the filtering process because objects should be selected which distance to the query *relatively*

small. It forces us to keep a set of potential nearest objects, and the real distance of an image object is calculated only when its approximation of distance is less than the largest real distance among the distances of image objects in this set. If its real distance is less than the currently largest one, it is inserted and the image object whose real distance is the largest among the image objects in the set is deleted.

```
// o_q : the query image object   // w : the vector of weights associated with features
// o_i : the i-th image objects in the database Λ
// C_kNN−search : a set of candidate image objects for k-NN search
// kNNDist: the maximum distance between the query and the objects in C_kNN−search
// SelectMaxObject( C_kNN−search ) : a function that selects the image object from C_kNN−search
//                  that has the maximum distance to query image object
// FindMaxDist( C_kNN−search ) : a function that find the maximum distance between the query
//                  image object and the objects in C_kNN−search
Procedure k-NN Search( o_q , k,  w ){   //  k  is the number of nearest objects to find
   C_kNN−search = {}; kNNDist = MaxDist;
   for  ∀o_i  (1 ≤ i ≤ n)  do {
      if ( |C_kNN−search| < k ) { // if the number of candidate objects is less than k,
         C_kNN−search  =  C_kNN−search ∪ {o_i}  ; // insert  o_i  into the candidate set
      }
      else{
         apxDist = D'(o_i, o_q);
   // Filtering Process ; Compute real distance D(o_i, o_q)  only when  D'(o_i, o_q) < kNNDist
         if ( apxDist < kNNDist) {
            realDist = D(o_i, o_q);
            if ( realDist < kNNDist) {
               o_max = SelectMaxObject( C_kNN−search ) ;
               C_kNN−search  =  C_kNN−search − {o_max} ∪ {o_i}  ;  // replace  o_max  with  o_i
               kNNDist = FindMaxDist( C_kNN−search ) ;
            }
         }
      }
   }
}
```

Fig. 3. A *k*-NN search algorithm with HBI

5 System Implementation and Experiments

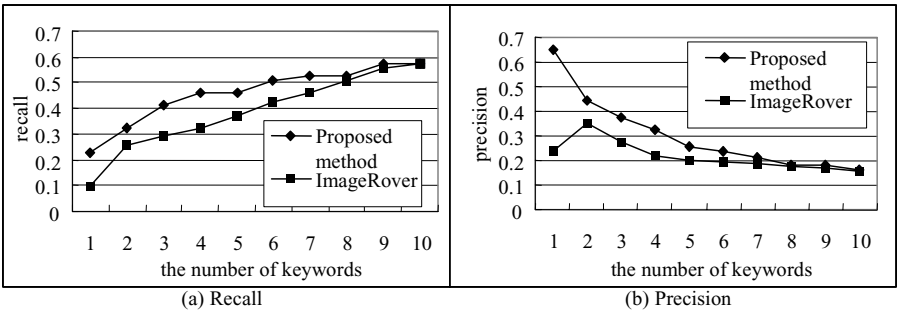
A fully functional web image retrieval system were implemented and tested based on the proposed algorithms. Every component of the system is tested under Windows XP on a Pentium 4 (3.0GHz) with 1GB memory.

5.1 Web Image Miner

In the web image miner, once the image collector starts to find images on the site specified by user, it continuously visits the web pages which are hyperlinked from the current page by breadth first search (BFS). If a visited page includes an image file, it downloads the image and passes it to the MPEG-7 visual descriptor extractor module which is programmed based on XM codes [13]. Three visual descriptors such as *dominant color* with 5 colors, *color layout* with 18 coefficients (6 for both luminance and chrominance), and *edge histogram* with 80 bins will be extracted from the image. After that, these visual descriptors and the HTML code of web page are passed to the keyword extractor module to extract keywords associated with the image by the proposed keyword selection algorithm.

To show the superiority of the proposed keyword selection algorithm compared to ones without use of visual features, experimental results were evaluated using precision and recall. To build up the image dictionary, 500 images are labeled manually and collected with 50 concepts such as landscape, animals, vehicles, and so on. And also the number of labels were restricted manually annotated for each image to 2~6 and set the number of clusters to 10.

80 web pages were collected to evaluate the proposed method where the page includes images associated with the concepts used in the learning stage. As shown in <Fig. 4>, both recall and precision of the proposed method are higher than those of ImageRover [5] and the difference of recall and precision between the two methods are decreased as the number of keywords increase. It implies that more relevant words to the image content get higher weights by the proposed algorithm.



(a) Recall (b) Precision

Fig. 4. The recall and precision of the proposed method compared to that of ImageRover[5] as a function of the number of keywords

5.2 Search Server and Search Client

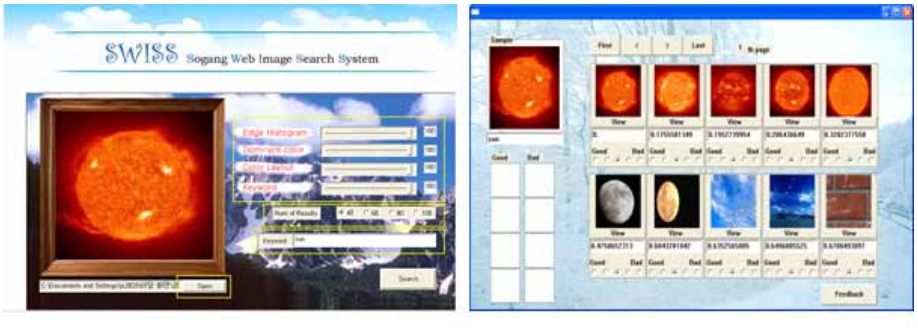
In our system, the search server consists of a database manager and a search engine. Whenever the database manager accepts an image and its features from the web image miner, those image and features will be saved to the temporary folder before inserting them into the database. The reason for it is that the vector representation and creation of index are CPU-consuming jobs. Therefore, the database manager is

designed to automatically trigger the insertion process when the number of collected images come up to the threshold user specifies (this threshold is set to 500 in our experiments).

Once the insertion process has triggered, all features are transformed to vector from and the unique identifier will be assigned to each image, which is used as the linker between an image and its index. Based on these identifiers, an index file per each feature is created respectively. Consequently, the system will have 8 index files for textual vectors, *dominant color* vectors, *Y* , *Cb*, *Cr* coefficient vectors of *color layout*, and local, global, semi global *edge histogram* vectors respectively.

Based on these index files, the search engine ranks the images in the database with regard to the query with the weights of features from the search client by the *k*-NN algorithm as mentioned in section 4.2 as the search results. To show the efficiency of the proposed *k*-NN algorithm using HBI, after 100,000 images were collected on the WWW and inserted into the search server, the total search time for 100 randomly generated query objects were evaluated. The meaning of total search time is that the time only for images ranked in the search server. According to our experiments, the total search time of the *k*-NN search using the proposed algorithm takes 960 ms, while the brute force search is about 2,500 ms on average. It implies the proposed *k*-NN search method is about 2.5 times faster than the brute force search. The detail of the performance of HBI, please refer to [17].

The search client provides convenient way of querying, browsing, and feedback. <Fig. 5>-(a) shows the querying interface of search client that it supports both query by example and keywords and also weight of importance could be specified by user. As the start of search, visual features extracted from example image and query for keywords will be sent to the search server and the search results will be shown as <Fig.5>-(b).



(a) Querying interface (b) Browsing interface

Fig. 5. Querying and browsing interface of the search client

<Fig. 6> is a good example of retrieval by combined visual and texture features. In <Fig. 6>-(a), both images of “star” and “Hollywood starts” are shown because only the textual features are used with the query string “star”. On the contrary, some odd

images are shown in <Fig. 6>-(b) because the images are retrieved by only visual features. Finally, the Hollywood star images could be retrieved by combination of visual and textual features as shown in <Fig. 6>-(c).

(a) The results of the query by keyword "star"

(b) The results of the query by example

(c) The results of the combined query by example and keyword "star"

Fig. 6. Comparison results of query by keyword and query by example

6 Conclusion

Our content based web image retrieval system was designed and implemented using both textual and visual features. To improve both the quality of results and the speed of retrieval, a new keyword selection algorithm based on both the visual features extracted from images and the layout of web pages, and an efficient k-NN search algorithm based on the hierarchical bitmap index using multiple features with dynamically updated weights was proposed. Also, these algorithms are adjusted to be well-suited with MPEG-7 visual descriptors such as dominant color, color layout, and edge histogram. Based on these algorithms, we built up a complete image retrieval system which provides the functionality for collection, management, searching, and browsing for images effectively. Upon experimental results, recall and precision of the proposed keyword selection algorithm were ranked higher than the existing algorithms. And it also shows that some examples of retrieval were enhanced by combination of visual and textual features. In terms of the efficiency of the system, the proposed *k*-NN search algorithm using HBI was about 2.5 times faster than brute force search when 100,000 images were stored in the server.

References

1. S. Rui, W. Jin, and T. Shua, "A Novel Approach to Auto Image Annotation Based on Pair-wise Constrained Clustering and Semi-naïve Bayesian Model," *Proc. of IEEE Int. Conf. on Multimedia Modeling*, pp.322-327, 2005.
2. L. Wang, L. Liu, and L. Khan, "Automatic Image Annotation and Retrieval using Subspace Clustering Algorithm," *Proceedings of the ACM international workshop on Multimedia Databases*, 2004.
3. Y. Mori, H. Takahashi, and R.Oka, "Image-To-Word Transformation based on Dividing and Vector Quantizing Images with Words," *Proc. of Int. Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

4. R. Yates and B. Neto, *Modern Information Retrieval*, Addison Wesley, pp. 74-84, 1999.
5. M. Cascia, S. Sclaroff, and L. Taycher, "Combining Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web," *Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries*, pp. 24-, 28, 1998.
6. J. Smith and S. Chang, "WebSeek: An Image and Video Search Engine for the World Wide Web," *IS&T/SPIE Proc. of Storage and Retrieval for Image and Video Database V*, 1997.
7. C. Frankel, M. Swain, and V.Athitsos, "WebSeer: An Image Search Engine for the World Wide Web," *Technical Report 96-14*, University of Chicago Computer Science Department, 1996.
8. N. Rowe and B. Frew, "Automatic Caption Localization for Photographs on World Wide Web Pages," *Information Processing and Management*, Vol.34, No.1, 1998.
9. M. Flickner, et.al., "Query by Image and Video Content : the QBIC System," *IEEE Computer,* Vol.28, pp.23-32, 1995.
10. J. Smith and S. Chang, "VisualSeek : A Fully Automated Content Based Image Query System," *Proceedings of ACM Multimedia 96*, pp.87-98, 1996.
11. J. Bach, et.al., "The Virage Image Search Engine: An Open Framework for Image Managemant," *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, pp.76-87, 1996.
12. Y. Rui, T. Huang, and S. Mehrota, "Content based Image Retrieval with Relevance Feedback in MARS," *Proceedings of International Conference on Image Processing*, pp.815-818, 1997.
13. ISO/IEC JTC1/SC29/WG11 *MPEG-7 Visual part of eXperience Model Version 11.0*, 2001.
14. ISO/IEC JTC1/SC29/WG11 *Information Technology Multimedia Content Description Interface-Part3: Visual*, 2001.
15. H. Eidenberger, "Statistical Analysis of Content-based MPEG-7 Descriptors for Image Retrieval," *ACM Multimedia Systems Journal*, Vol.10, No.2, 2004.
16. C. Fellbaum, *WordNet: An Electronic Lexical Database*, MIT Press, pp.265~283, 1998.
17. J. Park and J. Nang, "A Hierarchical Bitmap Indexing Method for Content Based Multimedia Retrieval," *Proceedings of the IASTED International Conference on Internet, Multimedia systems, and Application*, pp.223-228. 2006.
18. J. Park and J. Nang, "Analysis of MPEG-7 Visual Descriptors for Data Indexing," *Proceedings of the Korean Information Science Society Conference*, pp. 175-177, 2005.