Developing an XML Document Retrieval System for a Digital Museum

Jae-Woo Chang

Dept. of Computer Engineering, Research Center for Advanced LBS Technology,
Chonbuk National University, Chonju,
Chonbuk 561-756, South Korea
jwchang@chonbuk.ac.kr

Abstract. In this paper, we develop an XML document retrieval system for a digital museum. It can support unified retrieval on XML documents based on both document structure and image content. To achieve it, we perform the indexing of XML documents describing Korean porcelains used for a digital museum, based on not only their basic unit of element but also their image color and shape features. In addition, we provide a similarity measure for a unified retrieval to a composite query, based on both document structure and image content. Finally, we implement our XML document retrieval system designed for a digital museum and analyze its performance in terms of retrieval time, insertion time, storage overhead, as well as recall and precision measure.

1 Introduction

Recently, it has been very common for users to meet a variety of XML documents through a Web browser since the XML (eXtensible Markup Language) has become a standard markup language to represent Web documents [1]. To develop a digital information retrieval system which provides services in the Web, it is necessary to efficiently retrieve XML documents required by users. An XML document not only has a logical and hierarchical structure, but also contains its multimedia data, such as image and video. As a result, it is essential to develop an XML document retrieval system that can support both the retrieval based on both document structure and image content.

In this paper, we develop an XML document retrieval system used for a digital museum. It can support unified retrieval on XML documents based on both document structure and image content. In order to support retrieval based on document structure, we perform the indexing of XML documents describing Korean porcelains used for a digital museum, based on their basic unit of elements. Using this, we design four index structures, i.e., keyword, structure, element and attribute index structure. For supporting retrieval based on image content, we also do the indexing of the documents describing Korean porcelains, based on color and shape features of their images. This results in the design of a high-dimensional index structure using the CBF method. Finally, we provide a similarity measure for a unified retrieval to a composite query, based on both document structure and image content.

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3480, pp. 77 – 86, 2005.

This paper is organized as follows. In Section 2, we introduce related work in the area of retrieval based on document structure and image content. In Section 3, we design an XML document retrieval system for a digital museum. In Section 4, we present the implementation of our XML document retrieval system designed for a digital museum and analyze its performance. Finally, we draw conclusions and provide future work in Section 5.

2 Related Work

Because an element is a basic unit that constitutes a structured (i.e., SGML or XML) document, it is essential to support not only retrieval based on element units but also retrieval based on logical inclusion relationships among elements. First, RMIT in Australia proposed a *subtree model* which indexes all the elements in a document and stores all the terms which appear in the elements [2] so as to support five query types for structure-based retrieval in SGML documents. Secondly, SERI in South Korea proposed a *K-ary Complete Tree Structure* which represents a SGML document as a K-ary complete tree [3]. In this method, a relationship between two elements can be acquired by calculation because each element corresponds to a node in a K-ary tree. Thirdly, Univ. of Wisconsin in Madison proposed a new technique to use the position and depth of a tree node for indexing each occurrence of XML elements [4]. For this, the inverted index was used to enable ancestor queries to be answered in constant time. Fourthly, IBM T.J. Watson research center in Hawthorne proposed ViST, a novel index structure for searching XML documents [5]. The ViST made use of tree structures as the basic unit of query to avoid expensive join operations and provided a unified index on both text content and structure of XML documents. However, these four indexing techniques were supposed to handle tree data. Finally, Univ. of Singapore proposed D(k)-Index, a structural summary for general graph structured documents [6]. The D(k) index possesses the adaptive ability to adjust its structure according to the current query load, thus facilitating efficient update algorithms

There have been a lot of studies on content-based retrieval techniques for multimedia or XML documents. First, the *QBIC(Query By Image Content) project* of IBM Almaden research center studied content-based image retrieval on a large on-line multimedia database [7]. The study supported various query types based on the visual image features such as color, texture, and shape. Secondly, the VisualSEEk project of Colombia University in the USA developed a system for content-based retrieval and browsing [8]. Its purpose was an implementation of CBVQ(Content-Based Visual Query) that combines spatial locations of image objects and their colors. Thirdly, the Chonbuk National Univ. in South Korea developed an XML document retrieval system that can support a unified retrieval based on both image content and document structure [9]. Fourthly, the Pennsylvania State Univ. presented a comprehensive survey on the use of pattern recognition methods for content-based retrieval on image and video information [10]. Finally, the Chinese Univ. of Hong Kong presented a multi-lingual digital video content management system, called iVIEW, for intelligent searching and access of English and Chinese video contents [11]. The iVIEW system allows full content indexing and retrieval of multi-lingual text, audio and video materials in XML documents.

3 Design of XML Document Retrieval System

We design an XML document retrieval system for a digital museum, which is mainly consists of five parts, such as a preprocessing part, an indexing part, a storage manager part, a retrieval part and a user interface part. Figure 1 shows the system architecture of our XML document retrieval system for a digital museum. When an XML document is given, we parse it and perform image segmentation from it through the indexing part, so that we can index its document structure consisting of element units and can obtain the index information of color and shape features of its image. The index information for document structure and that for image content are separately stored into its structure-based and content-based index structures, respectively. Using the index information extracted from a set of XML documents, some documents are retrieved by the retrieval part in order to obtain a unified result to answer user queries. Finally, the unified document result is given to users through a user interface part using a Web browser.

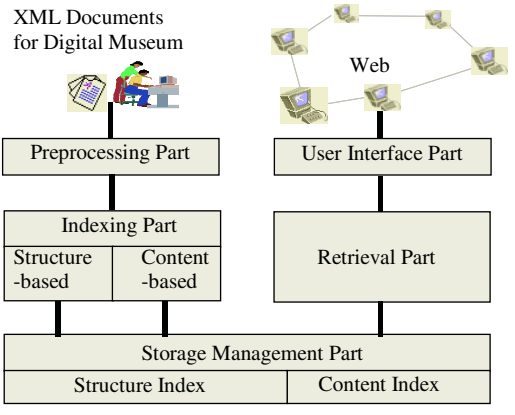

Fig. 1. XML document retrieval system architecture for a digital museum

3.1 Indexing

Because an element is a basic unit for retrieving an XML document, it is necessary to support a query based on a logical inclusion between elements and based on the characteristic value of elements. To achieve it, we construct a document structure tree for XML documents describing Korean porcelains used for a digital museum, after analyzing XML documents based on DTD. Figure 2 depicts a DTD grammar for representing XML documents describing Korean porcelains and an XML document instance following the DTD. The XML document instance contains not only a document structure between elements, but also attribute information. That is, the *porcelain* element has child elements, i.e., *name, year, description, and image* element. The *porcelain* element has an attribute 'TYPE' with a value 'Chung-ja'. To build a document structure tree for XML documents describing Korean porcelains, we parse the XML documents by using sp-1.3 parser [12]. Finally the storage manager extracts document structure information and image content information from the tree

and stores them into a database. Figure 3 shows the document structure tree built from the XML DTD grammar in Figure 2.

```
<! ELEMENT relic(porcelain)*>
<!ELEMENT porcelain
 (name,year,possession,classification,description,image)>
  <!ATTLIST porcelain TYPE CDATA #REQUIRED>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT year (#PCDATA)>
  <!ELEMENT possession(museum|university|personal)>
      <!ELEMENT museum (#PCDATA)>
      <!ELEMENT university (#PCDATA)>
      <!ELEMENT personal (#PCDATA)>
                    •
  <! ELEMENT image(align)*>
  <! ATTLIST image SRC CDATA#REQUIRED>
      <! ELEMENT align (#PCDATA)
                    •
```

```
<relic>
 <porcelain TYPE = "Chung-ja">
   <name>
     Chung-ja kettle
   </name>
   <year>
     A.D 1170
   </year>
          :
   <description>
     Chung-ja kettle is like a shape …
   </description>
   <image SRC = "hc_1">
   </image>
 </porcelain>
</relic>
```

Fig. 2. DTD grammar and its instance for XML documents describing Korean porcelains

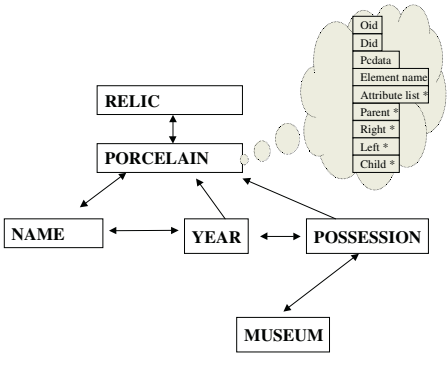

Fig. 3. Document structure tree built from XML DTD grammar

For content-based indexing of images contained in XML documents, we extract images from XML documents and analyze them for obtaining image feature vectors. To achieve it, it is necessary to separate a salient object, i.e., a Korean porcelain, from the background of an image. To extract only a region for the salient object, we use the fuzzy c-mean (FCM) algorithm that is a generally famous clustering one to divide object regions from color images [13]. When we divide an image into two clusters, the FCM algorithm calculates the distance of a pixel from the center point of each cluster and assigns it to a cluster with shorter distance. It has an advantage that the separation of a salient object of an image from its background can be performed well when an image has little noise, as shown in our porcelain images.

In order to obtain an image feature vector for shape, we obtain a salient object from an image by the preprocessing part and generate a 24-dimensional feature vector. An algorithm for generating a shape feature vector is as follows. First, we calculate the

central point of a salient object using its maximum and minimum value. Secondly, we increase 15 degrees at the central point, starting from the X-axis, and select 24 points met at the edge. Thirdly, we compute the distance between the central point and the 24 edge points. Finally, we generate a 24-dimensional feature vector by normalizing the 24 distances by dividing them by the maximum distance. It can be very efficient for digital museums where an image has only a couple of salient objects. Since the proximity among colors in the RGB color space doesn't mean their similarity among colors, we use HSV color space model. The HSV model provides a uniform distribution of colors and makes color transformation easier. In this model, H means an aggregate of color, ranging from 0 to 360 degree. S means the saturation of color, and V means the brightness of color. An algorithm for generating a color feature vector is as follows. First, we transform all color pixels of an image object in the RGB color space into those in the HSV color space. Secondly, we generate a color histogram by using color histogram generation algorithm. Finally, we generate a 22-dimensional feature vector by normalizing the color histogram by dividing it by the number of the entire pixel.

3.2 Storage Management

The index information for document structure and that for image content are separately stored into structure and content index structures, respectively. The index structures for structure-based retrieval are constructed by indexing XML documents based on an element unit and consist of keyword, structure, element, and attribute index structures. First, the keyword index consists of three files, i.e., keyword index file being composed of keywords extracted from data token element (e.g., PCDATA, CDATA) of XML documents, posting file including the IDs of document and element where keywords appear, and location file containing the location of keyword appearance in elements. Secondly, because the structure index is used for searching an inclusion relationship among elements, it should represent the logical structure of a document and guarantee good performance on both retrieval time and storage overhead. To achieve it, we propose an element unit parse tree structure where an element contains the location of its parent, its left sibling, its right sibling, and its first left child. We can find an inclusion relationship among elements easily because the tree structure represents the hierarchical structure of a document well. The element index structure contains element information and the identifier of a document that an element belongs. Thirdly, the element index structure contains some element information and the identifier of a document that an element belongs. Finally, the attribute index structure contains some attribute information and the identifier of an element that an attribute belongs.

The index structure for content-based retrieval is a high-dimensional index structure based on the CBF method [14], so as to store and retrieve both color and shape feature vectors efficiently. A main focus on managing a large number of XML documents is retrieval performance. As the number of dimensions of feature vectors is increasing, the retrieval performance of the traditional index structures is exponentially increasing. However, the CBF method can achieve good retrieval performance even though the dimension of feature vectors is high. In addition, our CBF-based index structure can support a variety of types of queries, like point, range, and K-NN search.

3.3 Retrieval

Using the stored index information extracted from a set of XML documents, some documents are retrieved by the retrieval part in order to obtain a unified result to answer user queries. There is little research on retrieval models for integrating structure- and content-based information retrieval. To answer a query for retrieval based on document structure, a similarity measure (S_w) between two elements, say *q* and *t*, is computed as the similarity between the term vector of node *q* and that of node *t* by using a cosine measure [15]. Supposed that a document can be represented as D = { E_0, E_1, …., E_{n-1} } where E_i is an element i in a document D. Thus, a similarity measure (D_w) between an element q and a document D is computed as follows.

$$D_w = \text{MAX } \{ COSINE\ (NODE_q, NODE_{E_i}), 0 \le i \le n - 1\}$$

To answer a query for retrieval based on image content, we first extract color or shape feature vectors from a given query image. Next, we compute Euclidean distances between a query color (or shape) feature vector and the stored image color (or shape) vectors by searching the content index structure. A similarity measure, $C_W(q, t)$, between a query image q and a target image t in the database is calculated as the following equation. Here Distc(q, t) and Dists(q, t) are a color vectors distance and a shape vector distance between q and t, respectively. Nc and Ns are the maximum color and the maximum shape distances for normalization, respectively.

$$C_w = \begin{cases} 1 - \dfrac{Distc(q,t)}{Nc}, & \text{if a query contains only a color feature.} \\ 1 - \dfrac{Dists(q,t)}{Ns}, & \text{if a query contains only a shape feature.} \\ (1 - \dfrac{Distc(q,t)}{Nc}) \times (1 - \dfrac{Dists(q,t)}{Ns}), & \text{if a query contains both color and shape feature.} \end{cases}$$

Finally, when α is the relative weight of retrieval based on document structure over that based on image content, a similarity measure (T_w) for a composite query is calculated as follows. If the weight of the former retrieval is equal to that of the latter retrieval, α equals 0.5.

$$T_w = \begin{cases} C_w \times \alpha + D_w \times (1 - \alpha), & \text{if results are document for user query} \\ C_w \times \alpha + S_w \times (1 - \alpha), & \text{if results are element for user query} \end{cases}$$

4 Implementation and Performance Analysis

We implement our XML document retrieval system for a digital museum, under SUN SPARCstation 20 with GNU CCv2.7 compiler. For this, we make use of O_2-Store v4.6 [15] as a storage system and Sp-1.3 as an XML parser. For constructing a prototype digital museum, we make use of 630 XML documents describing Korean porcelains with a museum XML DTD, as shown in Table 1. They are extracted from several Korean porcelain books published by Korean National Central Museum.

Table 1. XML documents used for a digital museum

Document content	Korean porcelains
The number of documents	630 documents
Document average size	1.20K(text)+42K(image)
The number of elements	10754
The number of keyword	37807

For retrieving XML documents, we can classify user queries into two types, i.e., simple and composite queries. The simple queries can be divided into keyword-, structure-, attribute-, and content-based query. Their examples are as follows.

- Keyword-based: Find documents which contain 'Buddhist image' term.
- Structure-based: Find all children elements of [Porcelain] element.
- Attribute-based: Find documents whose attribute type is 'Chong-ja'.
- Content-based: Find documents whose image contains a specific color or shape.

The composite query is the combination of simple queries, i.e., structure + keyword, structure + attribute, keyword + color and structure + shape query. Figure 4 shows a digital museum interface for retrieving XML documents describing Korean porcelains. By using this interface, a user, for example, generates a composite query to retrieve documents that contain a keyword 'Buddhist image' in all children element of [porcelain] element whose TYPE attribute value is 'chong-ja' and that include an image whose color is 'blue' and whose shape is like 'kettle'. Figure 4 also shows a result for the composite query and its similarity with images in the database.

To evaluate the efficiency of our XML document retrieval system for a digital museum, we measure insertion time, retrieval time, and storage overhead. Our system requires about 6 seconds on the average to insert an XML document into keyword, attribute, structure, and element indexes. It also requires less than 1 second on the average to insert an image content into color and shape index. Figure 5 shows retrieval times for simple queries. The retrieval time for the structure-based query is 6.5 seconds. But the retrieval times for the other queries are less than 2 seconds, respectively. It is shown form the result that the structure-based query requires the largest amounts of time to answer it. We also measure retrieval times for the combinations of two simple queries, such as structure + keyword, structure + attribute, keyword + color, and structure + shape. Figure 6 shows retrieval times for composite queries. The retrieval time for a keyword + color query is less than 3 seconds. However, the retrieval times for structure + keyword, structure + attribute, and structure + shape are 8.7, 8.1, and 7.3 seconds, respectively. This shows that a composite query containing structure-based retrieval requires large amounts of time to answer it. Finally, we measure storage overhead as a ratio of the total size of our index files over that of the original XML documents. Our XML document retrieval system requires about 50% storage overhead.

To evaluate the retrieval effectiveness of our XML retrieval system, we measure recall and precision [16] by a test group of computer engineering graduate students from Chonbuk National University, South Korea. Table 2 shows the precision and recall measure of our XML document retrieval system when we choose K porcelains as

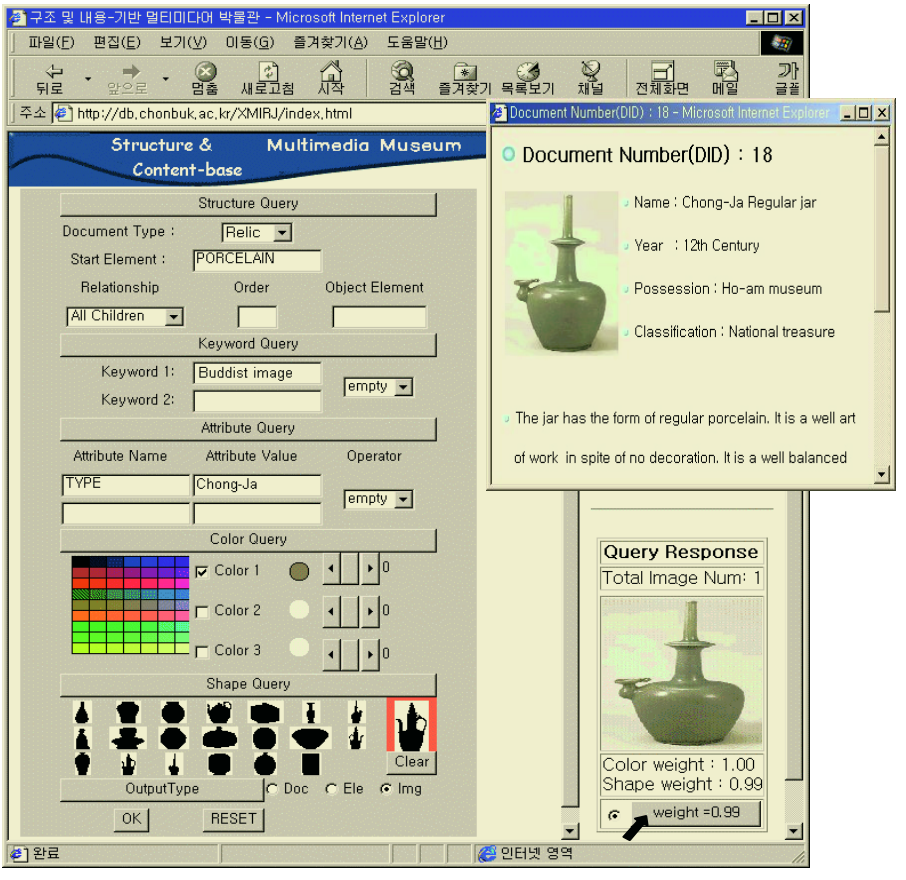

Fig. 4. A digital museum interface

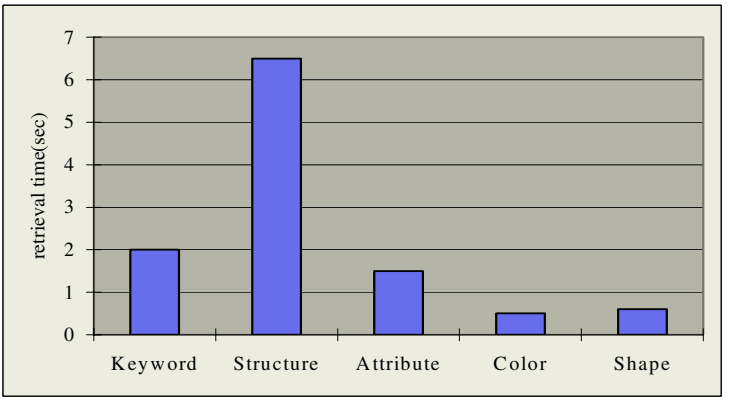

Fig. 5. Retrieval times for simple queries

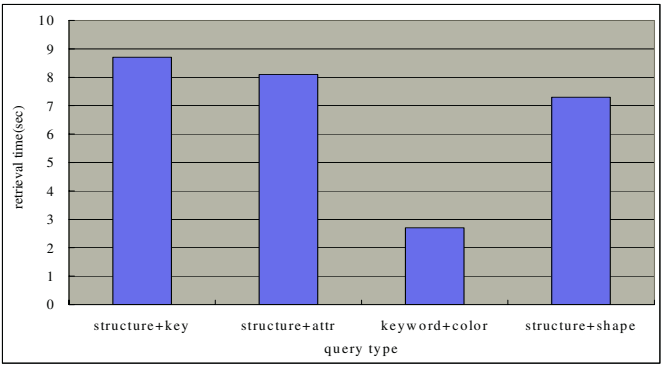

Fig. 6. Retrieval time for complex queries

the most similar ones. As K increases, the precision decreases and the recall increases. When K=10, the precision is about 0.6 and the recall is about 0.4, for both the color- and shape-based query. It is shown from our performance analysis that retrieval based on document structure has a great significance because a structure-based query requires much more time to answer it. Therefore, we compare our XML document retrieval system with the k-ary complete tree system [3] which is known as a promising system for structure-based queries. Our system requires 6.5 seconds for answering a structure-based query and the k-ary tree system requires 6.3 seconds. This shows nearly the same retrieval performance for structure-based queries.

Table 2. Precision and recall

	Color		Shape	
	Precision	Recall	Precision	Recall
K = 7	0.64	0.33	0.57	0.23
K = 10	0.60	0.37	0.55	0.32

5 Conclusions and Future Work

In this paper, we developed an XML document retrieval system for a digital museum. It can support efficient retrieval on XML documents for both structure- and content-based queries. In order to support structure-based queries, we performed the indexing of XML documents describing Korean porcelains for a digital museum, based on their basic unit of element. This resulted in designing four efficient index structures, i.e., keyword, structure, element, and attribute. In order to support content-based queries, we also performed the indexing of the XML documents based on both color and shape features of their images. This resulted in designing a high-dimensional index structure based on the CBF method. We also provided a similarity measure for a unified retrieval to a composite query, based on both document structure and image content. Our system for a digital museum requires about 6 seconds for answering a structure-based query and requires less than 2 seconds for the remaining queries. Our

system spends about 6 seconds on the average for inserting an XML document describing a Korean porcelain and requires about 50 % storage overhead. Future work can be studied on new information retrieval models for integrating preliminary results acquired from both structure- and content-based queries. This can be achieved ultimately by trying to handle MPEG-7 compliant XML documents [17].

References

[1] eXtensible Markup Language(XML), http :// www.w3.org/TR/PR-xml-971208.
[2] B. Lowe, J. Zobel and R. Sacks-Davis, “A Formal Model for Databases of Structured Text,” In Proc. Database Systems for Advanced Applications, pp 449-456, 1995.
[3] S.G. Han, et. al., “Design and Implementation of a Structured Information Retrieval System for SGML Documents,” In Proc. Database Systems for Advanced Applications, pp 81-88, 1999.
[4] C. Zhang, J. Naughton, D. DeWitt, Q. Luo, and G. Lohman, “On Supporting Containment Queries in Relational Database Management Systems,” In Proc. ACM SIGMOD. pp 425-436, 2001.
[5] H. Wang, S. Park, W. Fan, and P.S. Yu, “ViST: A Dynamic Index Method for Querying XML Data by Tree Structures,” In Proc. ACM SIGMOD. pp 110-121, 2003.
[6] Q. Chen, A. Lim, and K.W. Ong, “D(k)-Index: An Adaptive Structural Summary for Graph-structured Data,” In Proc. ACM SIGMOD. pp 134-144, 2003.
[7] M. Flickner, et. al., “Query by Image and Video Content: The QBIC System,” IEEE Computer, Vol. 28, No.9, pp. 23-32, 1995.
[8] J. R. Smith and S. F. Chang, “VisualSEEk: a Fully Automated Content-Based Image Query System,” In Proc. ACM Int’l Conf. on Multimedia, pp 87-98, 1996.
[9] K. Jin and J. Chang, “An Efficient Storage Manager for Content-based Multimedia Information Retrieval in NoD Applications,” In Proc. the 3rd Int’l Conf. of Asia Digital Library, pp 275-281, 2000.
[10] S. Antani, R. Kasturi, and R. Jain, “A Survey on the Use of Pattern Recognition Methods for Abstraction, Indexing and Retrieval of Images and Video,” Pattern Recognition. Vol. 35, No. 4, pp 945-965. (2002).
[11] M.R. Lyu, E. Yau, and S. Sze, “A Multilingual, Multimodal Digital Video Library System,” In Proc. ACM/IEEE-CS Joint Conf. on Digital Libraries, pp 145-153, 2002.
[12] http://www.jclark.com/sp.
[13] J.C. Bezdek and M.M. Triedi, "Low Level Segmentation of Aerial Image with Fuzzy Clustering," IEEE Trans. on SMC, Vol. 16, pp 589-598, 1986.
[14] S.G. Han and J.W. Chang, “A New High-Dimensional Index Structure using a Cell-based Filtering Technique,” Lecture Notes in Computer Science, Vol., pp 79-92, 2000.
[15] O. Deux et al. “The O_2 System,” Communication of the ACM, Vol. 34, No. 10, pp 34-48, 1991.
[16] G. Salton and M. McGill, “An Introduction to Modern Information Retrieval,” McGraw-Hill, 1983.
[17] U. Westermann and W. Klas, “An Analysis of XML Database Solutions for the Management of MPEG-7 Media Descriptions,” ACM Computing Surveys. Vol. 35, No. 4, pp 331-373, 2003.