

구조 및 내용-기반 멀티미디어 문서검색 시스템의 설계 및 구현

진 두 석[†] · 이 정 재^{**} · 장 재 우^{***}

요 약

최근 멀티미디어 문서의 개수가 기하 급수적으로 증가함에 따라, 사용자가 요구하는 멀티미디어 문서를 보다 효과적으로 저장 및 검색할 수 있는 멀티미디어 문서 검색 시스템을 개발하는 것이 필요하다. 본 논문에서는 XML로 정의된 문서를 문서 구조 및 이미지 내용을 기반으로 보다 효율적으로 검색할 수 있는 시스템을 설계 및 구현한다. 효율적인 구조-기반 검색을 지원하기 위해서 구조 인덱스를 o2store 저장 시스템을 사용하여 구현한다. 아울러 내용-기반 검색을 지원하기 위해서 X-트리에 기반한 효율적인 고차원 색인구조를 구현한다. 마지막으로 구현된 멀티미디어 문서검색 시스템을 검색시간, 저장시간, 부가 저장공간의 관점에서 성능평가를 수행한다.

Design and Implementation of a Structure and Content-based Multimedia Document Retrieval System

Du-Seok Jin[†] · Jeong-Jae Lee^{**} · Jae-Woo Chang^{***}

ABSTRACT

As the number of multimedia documents is dramatically increasing, it is needed to develop a multimedia document retrieval system which effectively store and retrieve multimedia documents required by users. In this paper, we design and implement a document retrieval system which can more efficiently retrieve documents defined by XML, based on both document structure and image content. In order to support efficient, structure-based retrieval, we implement a structured index by using the o2store storage system. For supporting content-based retrieval, we implement an efficient high-dimensional index structure based on X-tree. Finally, we perform the performance evaluation of our multimedia document retrieval system in terms of retrieval time, insertion time, and storage overhead.

1. 서 론

최근 인터넷의 발달로 멀리 떨어진 사이트간에 텍스트 뿐만 아니라 이미지, 오디오, 비디오를 포함하고 있는 멀티미디어 문서를 전달하는 것이 보편화되고 있

다. 아울러, 이러한 멀티미디어 문서가 인터넷을 통해 전달될 때, 문서의 개수가 기하급수적으로 증가함에 따라 사용자가 요구하는 멀티미디어 문서를 보다 효과적으로 검색할 수 있는 검색 기술이 요구된다. 이러한 특징을 잘 표현하고자 W3(World Wide Web Consortium)에서는 차세대 웹 문서의 표준으로 XML(extendible Markup Language)을 1996년에 제안하였으며 현재까지 그 기능이 계속 확장되고 있는 상태이다[1]. XML은 SGML(Standard Generalized Markup Language

※ 본 연구는 정보통신부의 정보통신 우수 지원학교 지원사업에 의하여 수행되었습니다.

† 정희원 경북대학교 대학원 컴퓨터공학과

** 장희원 케이원 시스템(주)

*** 종신희원 경북대학교 컴퓨터공학과 교수

논문접수: 1999년 8월 27일, 심사완료: 2000년 10월 18일

age)이 가지고 있는 다양한 기능들과 구조적인 표현 능력, 그리고 HTML이 가지는 장점인 사용하기 편리하다는 점을 가지고 있다. 최근의 인터넷 브라우저인 익스플로러 5.0에서는 XML과 이에 대한 표현양식을 기술한 XSL(eXtensible Style Language)을 지원하고 있어, 인터넷 상에서의 XML문서 활용은 더욱 증가할 것이다. 이에 XML문서의 특성을 반영한 문서 저장 및 검색 시스템에 대한 요구도 시급한 상태이다.

기존의 문서검색 시스템의 경우 내용정보와 구조정보 모두를 효과적으로 검색하지는 못하고 있다[2]. 즉 내용 정보 혹은 구조 정보만을 이용한 검색을 지원하고 있어, 내용 정보와 구조 정보가 결합한 형태의 검색을 처리하고 있지 못한 실정이다. 따라서, 본 논문에서는 XML문서의 구조 정보, 내용 정보를 효과적으로 검색할 수 있는 XML문서 검색 시스템을 설계 및 구현한다. 제안하는 시스템은 구조-기반 검색을 위해 o2store상에 구조 인덱스를 구축하고, 내용-기반 검색을 위해 고차원 색인 구조인 X-트리를 사용한다.

본 논문의 구성은 다음과 같다. 2장에서는 구조 및 내용기반 검색 시스템의 관련연구를 소개하며, 3장에서는 본 논문에서 제안한 구조 및 내용-기반 멀티미디어 문서검색 시스템의 설계에 대해 설명한다. 아울러, 4장에서는 시스템 구현 및 성능평가를 제시하며 마지막으로 5장에서는 결론 및 향후 연구 방향을 제시한다.

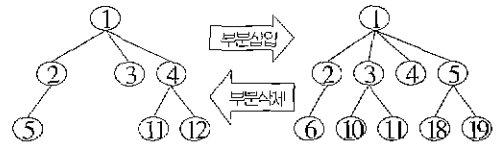
2. 관련 연구

2.1 구조·기반 검색

구조적인 정보를 지닌 문서의 검색을 위해서는 구조·기반 검색이 필요하며 관련 연구로는 SGML 정보 검색을 꼽을 수 있다. SGML 정보검색에서는 인덱스 설계에 있어 크게 세가지 접근방법이 있으며[3] 각각 K-ary Complete 트리구조, 문서 단위의 문서 구문 트리 구조, 엘리먼트 단위의 문서 구문 트리 구조가 있다.

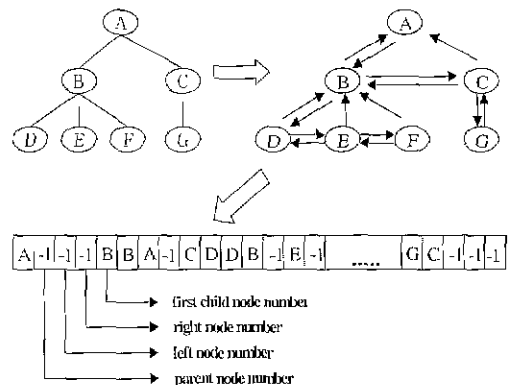
먼저, K-ary Complete 트리 구조를 보면 이 방법은 엘리먼트 단위의 설계 방법으로, 문서를 K-ary Complete 트리로 구성하여 각 트리를 구성하는 노드에 엘리먼트를 대응시키는 방법이다. 따라서 하나의 특정 엘리먼트와 다른 엘리먼트들 간의 관계를 계산할 수 있는 전역적인 접근 방식이다. 이 방법은 계산에 의해서 논리적 포함관계인 엘리먼트를 빠르게 찾을 수 있는 장점이 있으나, 노드의 깊이가 깊어질수록 노

드번호가 지수적으로 증가하고 사용하지 않은 노드번호가 많아지므로 데이터량이 커지는 단점이 있다. 또한 문서 구조의 부분 삽입과 부분 삭제 시 (그림 1)처럼 노드번호가 대부분 바뀌므로 연산 오버헤드가 크다



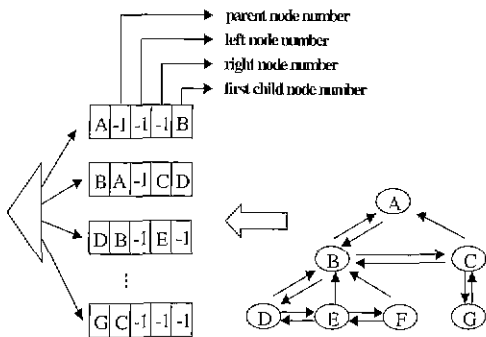
(그림 1) 부분삽입 및 부분삭제

들께, 문서 단위의 문서 구문 트리 구조는 SGML 파서를 통해 나온 문서 구조 정보를 하나의 레코드에 반영하는 광범위이다 (그림 2)는 이 과정을 나타낸다. 각 엘리먼트의 노드는 부모, 오른쪽 첫 번째 형제, 왼쪽 첫 번째 형제 및 첫 번째 자식을 가리키는 포인터를 구성한 후, 실제 레코드에 저장 시에는 포인터에 해당하는 위치의 노드번호를 저장하게 된다. 문서의 구조 정보를 하나의 레코드가 유지하기 때문에, 그 크기가 커서 레코드 안에 디렉토리를 유지하여 커다란 정보를 유지할 수 있는 Long Data 타입 레코드로 문서의 구조 정보를 관리하며 이는 레코드 접근하는데 많은 시간이 필요하다. 또한, 문서의 특정 구조 정보를 접근하더라도 전체 문서 구조 정보를 읽어야 하는 단점을 가지게 된다. 그러나, 구조 정보를 부분 삽입, 부분 삭제하는데 있어서는 특정 엘리먼트에만 갱신되어지고, 그 외의 엘리먼트에 대해서는 영향을 주지 않으므로 부분 삽입 및 부분 삭제가 용이하다는 장점이 있다.



(그림 2) 문서 구문 트리 구조 : 문서 단위

마지막으로 엘리먼트 단위의 문서 구문 트리 구조는 SGML 파서를 통해 나온 구문 트리 정보를 문서단위로 하나의 레코드에 저장하는 것이 아니라, 엘리먼트 단위의 레코드들로 분리시켜 저장한다. (그림 3)은 이 과정을 나타낸다 하나의 문서 구조 정보를 여러 개의 엘리먼트 단위 레코드로 저장하고 이를 빠르게 접근하기 위해 B+-tree를 사용한다. 따라서 저장공간의 오버헤드가 문서 구문 트리의 문서 단위보다 크게 된다. 그러나, 부분 삽입 및 부분 삭제는 용이하며, 문서의 특정 부분 구조를 접근할 때 B+-tree를 통해 빠르게 접근할 수 있다.



(그림 3) 문서 구문 트리 엘리먼트 단위

2.2 내용·기반 검색

내용에 기반을 둔 멀티미디어 문서 검색 시스템에 대한 연구는 국내외의 학교와 연구소에서 많은 연구가 수행되었다. 그 중에서 첫번째로 QBIC(Query By Image Content)은 미국 IBM Almaden 연구소에서 차세대 연구 프로젝트 중 하나로 개발된 시스템이다[4]. 이는 방대한 양의 이미지와 비디오 데이터 베이스를 기반으로 텍스트 키워드, 색상, 모양, 질감, 구도 등과 같은 시각적인 이미지 특징 정보를 통해 사용자-구성 구도나 드로잉 기반 질의(user-constructed sketches and drawings), 사용자 선택 색상 및 질감 질의(selected color and texture patterns), 다른 그래픽 정보 질의(other graphical information)와 같은 다양한 검색 질의를 제공한다. 이미지에서 추출한 특징 패턴을 바탕으로 사용자의 빠른 검색을 위해 색과 질감 특징에 대해 각각 Filtering 기법과 R*-트리를 사용하고 있다. 최근 QBIC 시스템을 이용하여 상업화 된 제품을 발표한 것이 Ultimea라는 내용기반 이미지 검색 시스템이다.

둘째, VisualSEEK는 미국의 콜롬비아 대학에서 이

미시에 대한 검색 및 브라우징을 위한 툴(tool)로서 개발한 이미지 데이터 베이스 시스템으로 이미지의 객체 간의 공간 위치(spatial location)와 색상을 결합한 사용자 질의를 처리할 수 있다[5]. 궁극적으로, VisualSEEK은 이미지 데이터 베이스에 대해 효율적이고 효과적인 검색 기능을 제공하는 CBVQ(Content-Based Visual Query) 시스템을 구현하는 데 그 목적이 있으며 이미지로부터 색상과 부분적인 영역(region) 자동 추출, 색상과 공간 정보를 결합한 사용자 질의, 사용자 질의에 대한 적합성 피드백(relevance feedback)과 텍스트-기반 검색을 위한 다양한 주석 방법(annotation method), 사용자 편의를 위한 길의 인터페이스 툴과 같은 이미지 검색 기능(functionality)을 지원한다.

셋째, Virage[6]는 VIR (Visual Information Retrieval)을 위한 기술의 툴을 개발하고 있으며 디지털 비디오 마켓, 의학 이미지 검색 그리고 인터넷상의 디지털 이미지 검색 등의 응용분야에서 많은 상업용 제품을 개발하고 있다. 또한 Virage 비디오 검색 엔진은 비디오 데이터를 의미 있는 클립(clips)들의 집합으로 변환하여 사용자의 질의에 적합한 장면을 쉽고 빠르게 검색하며 보여준다.

넷째, 멀티미디어 데이터는 다양한 형태의 데이터들로 구성되어 있고 데이터의 크기가 방대하기 때문에 MPEG-7에서는 정확성과 빠른 검색을 위한 멀티미디어 데이터의 내용 검색 표현 방법에 대한 표준화 작업을 수행하고 있다[7]. MPEG-7의 처리 과정은 특징 추출(feature extraction), 기술(description), 그리고 검색 엔진(search engine)으로 이루어져 있다 이 과정 이외에도 데이터베이스 구조 또한 검색 성능 향상에 중요한 역할을 한다.

국내에서는 첫째, 한국 과학 기술원에서 멀티미디어 데이터로부터 내용에 대한 정보를 추출하여 내용에 기반한 질의를 수행하도록 하는 내용 기반 검색 시스템인 COSMIC(Content based Multimedia Information processor)을 설계 및 구현하였다[8]. COSMIC은 대용량 이미지 데이터로부터 자동으로 추출된 시각적 특징 데이터들을 다차원 점 접근 방법(Point Access Method)인 HG(hilbert Grid)-트리를 이용하여 색인하고 멀티미디어 데이터로부터 추출된 다양한 의미 정보를 이용하여 영역지식을 구축하고, 멀티미디어 데이터를 효율적으로 검색하기 위하여 영역지식을 이용한 의미 질의를 제공한다.

둘째, 한양대학교에서는 내용 기반 상표 검색 시스템을 개발하였다[9] 새로운 상표를 등록하고자 할 때

기존에 등록된 상표와의 유사성을 측정할 때 드는 많은 비용을 절감하기 위하여 내용 기반 상표 검색 시스템은 2백만개 이상의 등록상표를 가지고 패턴인식 기술을 사용하여 유사성 검색을 처리함으로써 효율적인 내용 기반 질의처리 시스템을 구현하였다

셋째, 고려대학교의 내용 기반 사진 검색 시스템은 사진이 갖고 있는 색상이나 질감, 형태등의 내용에 기반한 정보 검색 방법으로 사진 영상의 분석을 통한 특징 추출과 특징 정보의 색인화 및 영상 데이터베이스에서의 고속의 검색 기법을 제공한다. 이 시스템에서 사용한 특징 정보는 색상정보와 형태정보 두 가지를 사용하며, 색상정보는 인간의 시각 특성과 유사한 HSV 색상 모델에서 히스토그램을 이용하여 추출하고, 형태 정보는 상대각을 이용하여 보는 각도, 크기, 위치 변환 등에 무관한 정보를 추출한다[10].

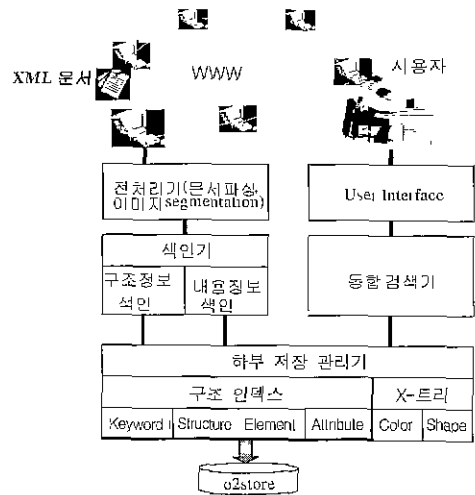
넷째, ICU의 이미지,비디오 내용기반 색인 시스템은 Randon 변환을 이용한 색상과 질감 검색과 객체의 움직임 벡터를 이용한 비디오 검색 시스템을 설계 및 구현하였다[11-12]. 또한 두 특징 정보간의 거리를 Mahalanobis distance를 사용하여 검색 결과에 대한 유사성을 측정할 수 있다

다섯째, 전북대학교의 NoD(News on Demand)시스템은 뉴스 비디오 데이터의 효율적인 저장 및 검색을 위한 하부저장 관리자를 설계하였다[13]. 뉴스 비디오 데이터의 색인을 위해서 SHORE 저장시스템에 키워드 검색을 위한 역화일과 내용기반 검색을 위한 X-tree인덱스를 확장하여 비디오 스트림 데이터와 색인정보를 효율적으로 저장하고 관리한다

3. 구조 및 내용-기반 멀티미디어 문서 검색 시스템의 설계

본 논문에서 설계한 문서 검색 시스템은 크게 5부분으로 구성되는데 첫째, 멀티미디어 문서의 파싱과 이미지 객체분리(segmentation) 과정을 처리하는 전처리기 부분, 둘째, 저장 및 질의를 위한 색인 키를 생성하는 색인기 부분, 셋째, 색인된 정보를 특정 데이터베이스에 저장하는 하부저장 관리기 부분, 넷째, 웹 브라우저를 통한 사용자 검색 질의를 처리하는 사용자 인터페이스 부분, 마지막으로, 검색된 결과를 적절한 형태로 가공하는 통합검색기 부분으로 구성되어 있다. 이에 대한 전체 시스템의 구조는 (그림 4)와 같다. 전체 시스템의 흐름은 먼저, 입력된 XML문서를 전처리 과

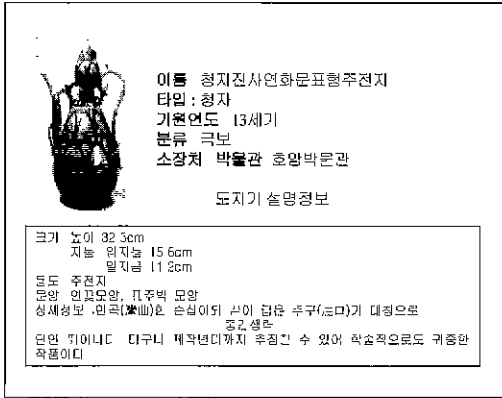
정을 통해 파싱된 이미지 세그멘테이션을 수행하고, 각각 파싱된 XML문서정보와 이미지 정보는 엘리먼트 단위의 문서 구조 색인을 위해 문서 구조 색인기에 전달되고 이미지의 색상과 형태에 대한 색인 정보를 구하기 위해서 이미지 내용 정보 색인기에 전달된다. 색인기를 통해 얻어진 구조 정보와 내용 정보는 하부 저장 관리기에서 구조 인덱스와 X-트리 인덱스에 저장된다[14]. 이렇게 적장된 XML 문서의 정보는 사용자 질의에 만족하는 문서를 통합 검색기에서 추출하고, 웹 브라우저를 통해 사용자가 원하는 적절한 형태로 출력한다.



(그림 4) 전체적인 시스템 구조

3.1 문서 구조 정보 색인기

XML 문서를 구성하는 기본 단위는 엘리먼트이다. 따라서, 기존의 정보 검색 시스템에서 사용한 문서 단위를 위주로 한 검색 이외에 인의 깊이에서 나타나는 엘리먼트 단위의 검색이 가능해야 한다. 아울러, 엘리먼트간의 논리적인 포함관계 및 엘리먼트의 특성 값에 대한 질의도 지원되어야 한다. 이를 위해, 본 논문에서 설계한 엘리먼트 기반 정보 검색 시스템은 DTD를 이용한 구문 분석 후 문서의 구조를 지시하는 마크업(Mark-up)에 대해 인덱스를 구성함으로써 효율적인 구조 질의를 가능하게 한다. 본 논문에서 설계한 시스템에 사용되는 데이터는 (그림 5)와 같은 멀티미디어 문서이다. 이러한 문서를 표현하기 위한 DTD와 XML 문서 (그림 6)는 다음과 같다.



(그림 5) 멀티미디어 문서 예

(그림 6)의 XML문서는 각 엘리먼트 사이의 구조 정보와 애트리뷰트 정보를 모두 가지고 있다. 예를 들면, porcelain과 name은 parent와 children의 관계이고 decoration과 detail은 sibling 관계를 나타내며, porcelain의 애트리뷰트 이름은 TYPE 이고 값은 청자인 경우를 나타낸다.

위의 같은 DTD에 맞게 작성된 멀티미디어 문서(XML)를 구조 트리를 만들기 위해서는 먼저 XML 파서인 sp-1.3 [15]를 사용하여 XML문서를 파싱한다. 파싱된 결과를 가지고 트리를 구성하는 알고리즘에 따라 구조 트리를 만든다 구성된 트리 정보는 하부저장 관리기에 전달한다. 하부저장 관리기에서 각각 구조정보와 내용성

보를 추출하여 별도의 데이터 베이스에 저장한다. (그림 7)은구조 트리를 만드는 과정을 보여주며, (그림 8)은 구성된 구조 트리를 나타낸다.

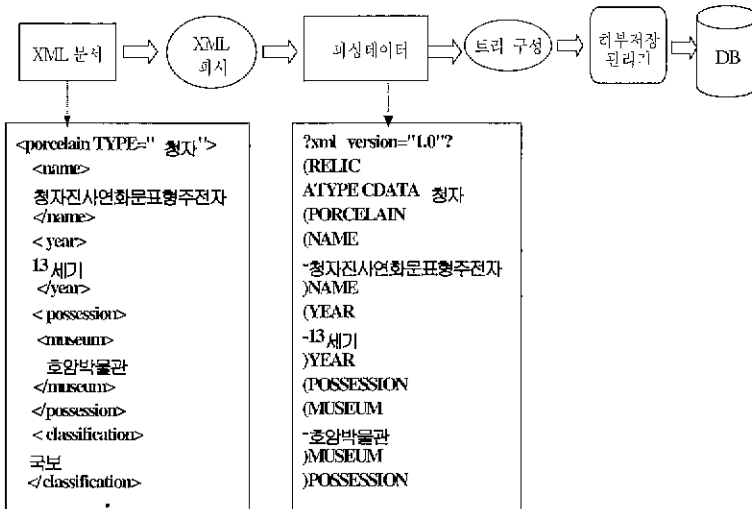
```
<ELEMENT relic (porcelain)*>
<ELEMENT porcelain
(name, year, possessor classification, description, image)>
<!ATTLIST porcelain TYPE CDATA #REQUIRED>
<ELEMENT name (#PCDATA)>
<ELEMENT year (#PCDATA)>
<ELEMENT possessor
(museum/university/personal)>
<ELEMENT museum (#PCDATA)>
<ELEMENT university (#PCDATA)>
<ELEMENT personal (#PCDATA)>

<ELEMENT image (align)*>
<!ATTLIST image SRC CDATA #REQUIRED>
<ELEMENT align (#PCDATA)>
```

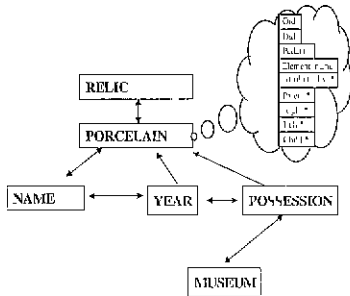
```
<relic>
<porcelain TYPE="청자">
<name>
청자진사연화문표형주전지
</name>
<decoration>
연꽃도임
</decoration>

<detail>
표주박 모양을 본떠 만든 주전기로, 민국(靑州)원
손잡이와 같이 굵은 주구(口口)가 대칭으로 달려 있다.
</detail>
<description>
<image SRC="hc_1">
</image>
</porcelain>
</relic>
```

(그림 6) 도자기 문서를 위한 DTD 및 XML 문서



(그림 7) XML 문서의 파싱과 트리 구성 과정



(그림 8) 구성된 트리의 각 엘리먼트의 구조정보

3.2 이미지 내용 정보 색인기

이미지 객체에 대한 내용-기반 검색을 위해서는 먼저 이미지 객체가 가지는 특징을 분석하고, 이를 기반으로 특징 벡터를 추출해야 한다. 특징 벡터를 추출하기 전에 수행해야 할 중요한 과정은 이미지 내의 배경을 제거하고 의미 있는 객체 영역만을 분리하는 과정이다.

3.2.1 객체 영역 추출 단계

본 논문에서는 컬러 이미지로부터 객체를 분할하기 위한 방법으로 퍼지 이론을 이용한 클러스터링 방법 중에서 가장 많이 알려진 Fuzzv c mean(FCM) 알고리즘[16]을 사용한다. 이 방법은 각 화소의 클래스에 관한 소속 함수 값이 1이 되는 확률적 제약 조건(probabilistic constrain)을 이용하고 있다. 예를 들면, 이미지를 2개의 클러스터로 분할할 경우, 어떤 입자의 화소에 대해서 두 클러스터의 중심으로부터의 거리를 계산하여 거리가 짧은 쪽으로 포함시키게 된다. 이 방법은 이미지 내에 큰 잡음만 없다면 이미지 객체 분할 성능이 좋은 편이다. 다만, 두 클러스터 중심간의 상대적인 거리차를 이용하기 때문에 중심간의 거리차가 다른 두 화소가 같은 클러스터에 속하게 될 때 중심간의 거리차가 다를지라도 실제 잡음과 데이터와의 구분이 불가능하다는 문제점이 있다. 그러니 본 논문에서 사용하는 대부분의 도자기 이미지는 거의 잡음이 없으며, 배경과 객체와의 색상 차이가 크기 때문에 대체로 좋은 이미지 분할 성능을 보인다.

3.2.2 형태 특징 벡터 추출

본 논문에서는 (1)의 과정을 통해 얻어진 이진 영상 객체를 바탕으로 이미지 객체의 위치와 크기에 관계없

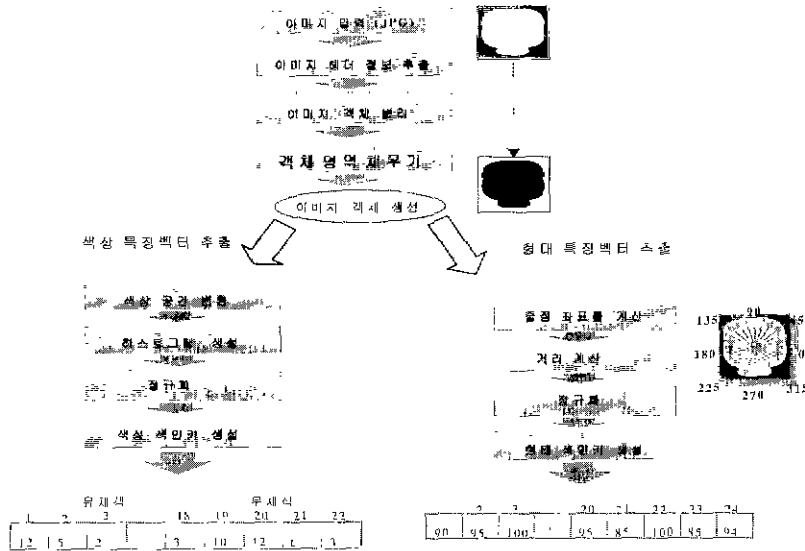
이 검색이 이루어 질 수 있도록 이미지 객체의 중점과 에지 픽셀간의 거리 정보를 이미지의 형태 특징 벡터로 이용한다. 형태 특징 벡터는 15도를 이용하는 방법을 사용하며, 형태 특징 벡터 생성 알고리즘은 다음과 같다[17]

1. 객체 안의 픽셀을 수평과 수직으로 각각 정렬시킨 후 최대값과 최소값을 구해서 객체의 중점 계산
2. 중점으로부터 15도씩 회전해가면서 객체의 에지와 인접하는 24개의 픽셀 좌표값을 구함
3. 중점과 24개의 에지 픽셀들간의 거리를 계산
4. 24차원에서 각 차원의 최대 거리로 나누어서 거리를 정규화 시킴
5. 24차원의 형태 특징 벡터 색인 키 생성

(그림 9)는 실제 이미지로부터 객체의 에지 픽셀들을 구한 뒤에, 각각 24차원의 형태 특징 벡터 색인 키를 구한 예이다. 아울러, 제안하는 형태 특징 벡터 추출 방법은 도자기나 박물관 문화재와 같이 한 이미지 내에 하나의 중심 객체가 존재하는 응용에 직접 적용이 가능하다. 그러나 증강화 이미지와 같이 여러 개의 중심 객체가 포함되어 있는 응용에는 직접 적용이 불가능하다.

3.2.3 색상 특징 벡터 추출

이미지를 구별하는 시각적인 척도 중에서 가장 많이 사용하는 것은 이미지의 색상 특징이다. 일반적으로 RGB 색상 공간은 색상들간의 의존성이 매우 높고 색상들간의 근접성(proximity)이 색상의 유사성(similarity)을 나타내지 못한다는 단점을 지닌다. 이러한 단점을 보완하기 위한 방법으로 HHSV(Hue, Saturation, Value) 색상 공간 모델은 균일한 분포(uniform)와 비선형적인(non-linear) 특성을 가지며, 색상 변환(transform)이 까다롭지 않은 장점을 가진다. HSV모델에서 H값은 색상의 집합을 나타내면서 0도에서 360까지의 각도를 가진다. S값은 H값의 각도에 대한 색상의 세기(채도)를 나타내며, V값은 색상의 밝기 즉, 명도로서 V값이 클수록 색상은 흰색에 가까워지고, 작을수록 검정색에 가까워진다. 본 논문에서는 색상 특징 벡터 차원의 수를 22차원으로 하여 색상 특징 벡터(색인 키)를 추출하며, 알고리즘은 다음과 같다[17]



(그림 9) 색상 및 형태 특징벡터 추출 과정

- 1 배경이 제거된 이미지 객체의 모든 픽셀들을 HSV 색상 공간으로 변환
- 2 색상 히스토그램 알고리즘을 적용하여 색상 히스토그램 생성
- 3 색상 히스토그램 정규화(객체 안의 전체 픽셀 수로 나눔)
4. 22차원의 색상 특징 벡터 색인 키 생성

(그림 9)는 실제 이미지로부터 추출한 22차원의 색상 히스토그램 색인 키의 예를 보여준다.

3.3 하부저장 관리기

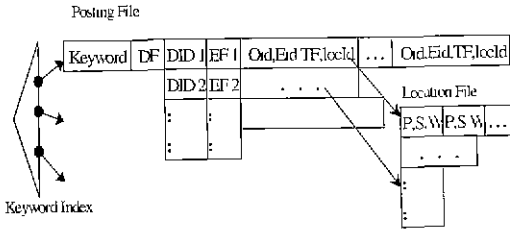
구조 및 내용-기반 멀티미디어 정보 검색 시스템을 위한 하부 저장 구조는 크게 두 가지 유형으로 설계되어진다 첫째는 구조정보 및 단순속성 정보 즉, 텍스트 설명 정보와 엘리먼트간의 관계정보를 저장하기 위한 하부 저장 구조와 이미지 특정 정보에 대해서 각각 그에 적합한 접근 기법을 가지는 형태이다 이러한 구조는 각각의 접근 기법을 제어할 수 있는 별도의 제어 모듈이 필요하다는 단점이 있는 반면에, 현재까지 연구되어진 대부분의 기술로 구현이 가능하다는 장점이 있다. 둘째는 멀티미디어 구성 요소들이 가지는 각각의 특징들을 하나의 차원으로 간주해서 고차원 특징 벡터를 생성하여 그에 적합한 고차원 접근 기법을 적

용시키는 것이다. 그러나 이러한 형태는 그 차원의 수가 최소 수십에서 수백 차원에 이르기 때문에, 현재까지는 구현상에 여러가지 어려운 점이 있다. 본 논문에서 설계하는 하부 저장 구조는 전자의 방식을 택한다 즉, 구조 정보와 단순 속성 정보 그리고 이미지 특정 색인 키에 대해서 각각 그에 맞는 접근 기법을 갖도록 설계한다. 구조 정보 검색 인덱스 관리자는 Keyword 인덱스, 구조 인덱스, 엘리먼트 인덱스와 애트리뷰트 인덱스로 구성되며, XML의 기본 표현 단위인 엘리먼트 단위로 색인하여 인덱스를 구성한다. 아울러 이미지 내용검색 인덱스 관리자는 색상 및 형태 특징 벡터에 대해서 X-트리 인덱스를 구성한다.

3.3.1 Keyword 인덱스

Keyword 인덱스는 XML 문서의 데이터 토큰(PC-DATA, CDATA) 엘리먼트로부터 추출된 색인어들로 구성된 단어 색인 파일(Keyword Index File), 색인어가 출현한 문서와 엘리먼트의 정보를 나타내는 포스팅 파일(Posting File), 엘리먼트에서 색인어의 출현 위치에 대한 정보를 포함하는 위치 정보 파일(Location File)로 구성된다 (그림 10)은 Keyword 인덱스의 구조를 나타내며, 기존의 정보 검색 인덱스에 엘리먼트에 대한 정보가 삽입되어 한 단계 확장된 형태를 가진다.

또한, 위치 정보 파일은 압축된 형태로 저장된다.



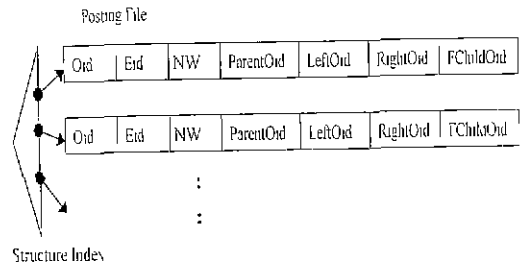
(그림 10) Keyword 인덱스의 구조

DF(Document Frequency)는 문서 출현 횟수로서, 색인어가 출현한 문서의 개수를 나타내며, DID(Document ID)는 각각의 문서가 가지는 고유한 문서 식별자를 나타낸다. EF(Element Frequency)는 문서에서 색인어가 출현한 엘리먼트 개수를 나타내며, 각각의 색인어는 하나의 문서를 구성하는 여러 엘리먼트에서 출현할 수 있으므로 여러 개의 엘리먼트 정보를 가지며, 각 문서마다 서로 다른 EF 값을 가지게 된다. OID(Object ID)는 엘리먼트마다 가지는 고유한 식별자이며, EID(Element name ID)는 각각의 엘리먼트 이름에 대한 식별자이다. TF(Term Frequency)는 하나의 엘리먼트 안에서 출현한 색인어의 개수를 나타내며, 각각의 위치 정보 (P,S,W)을 위치 정보 파일에 저장한 후, locId(Location ID)로 연결한다. 아울러, Keyword 인덱스의 저장공간을 줄이기 위해 위치 정보 파일의 압축 방법을 수행한다 위치 정보는 문서에 대해서 색인어가 출현한 위치를 나타내는 것으로 P(Paragraph), S(Sentence), W(Word)의 세 가지 값으로 표현된다 하나의 엘리먼트 안에서 색인어는 여러 번 나올 수 있으므로 위치 정보 파일은 (P,S,W)의 연속된 값을 가진다. 하나의 엘리먼트 안에서 출현한 (P,S,W) 정보들은 P 또는 S 값의 중복이 종종 일어난다 따라서, 압축 일고리즘에서는 (P,S,W) 중에서 중복된 값은 저장하지 않고, 변경된 값의 증가분(Increment값)만을 저장하여 저장 공간을 줄인다.

3.3.2 구조 인덱스

구조 인덱스는 문서를 구성하는 엘리먼트들 간의 포함 관계를 검색하기 위한 것으로, 문서의 논리적인 구조를 정보의 손실 없이 표현할 수 있어야 하며, 효율적인

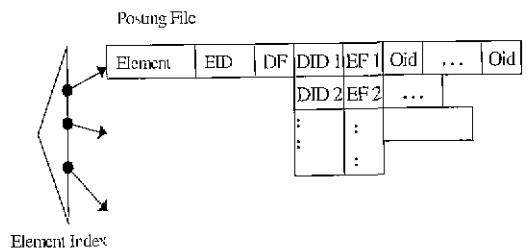
저장 공간과 탐색 시간을 보장해야 한다. (그림 11)은 구조 인덱스의 구조를 나타낸다. 구조 인덱스는 엘리먼트마다 고유한 식별자 값인 OID값으로 인덱스가 구성되며, 엘리먼트 간의 포함 관계에 대한 빠른 탐색을 위해 부모 노드의 OID값(ParentOid), 왼쪽 형제 노드의 OID값(LeftOid), 오른쪽 형제 노드의 OID값(RightOid), 그리고 첫번째 지식 노드의 OID값(FChildOid)에 대한 정보를 가지고 있다 또한, 부모 노드와 자식 노드 간의 중요도를 나타내기 위해 가중치 NW(Node Weight)에 대한 정보를 포함한다. NW는 엘리먼트 내의 색인어(term)를 벡터로 표현하여 엘리먼트 term 벡터간의 유사도를 나타낸다.



(그림 11) 구조 인덱스의 구조

3.3.3 엘리먼트 인덱스

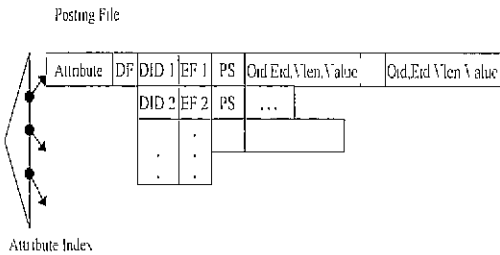
엘리먼트 인덱스는 구조 검색의 경우 시작 엘리먼트를 검색하기 위해서 사용되며, 또한 Keyword 인덱스나 에트리뷰트 인덱스에서 얻은 엘리먼트 이름 식별자(EID)와 엘리먼트 이름(Element Name)을 매핑시켜 주는 역할을 수행한다. (그림 12)은 엘리먼트 인덱스의 구조를 나타내며, 각 필드의 내용은 Keyword 인덱스 구조와 동일한 의미를 지닌다.



(그림 12) 엘리먼트 인덱스의 구조

3.3.4 애트리뷰트 인덱스

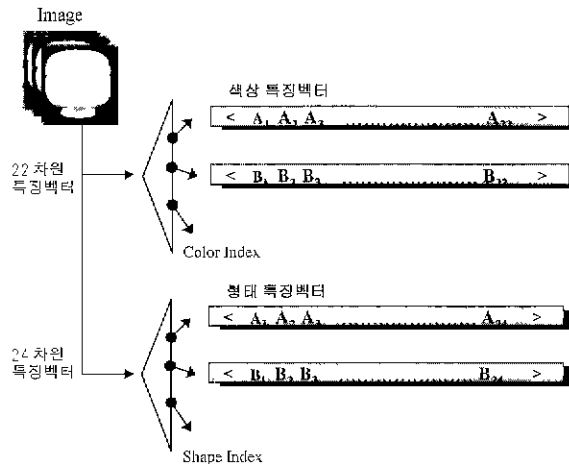
애트리뷰트 인덱스는 엘리먼트가 가지는 애트리뷰트 이름(Attribute Name)과 그 특성 값(Attribute Value)에 대한 검색을 수행한다. 애트리뷰트가 가지는 값은 여러 형태로 존재할 수 있으므로, 설계하는 인덱스 구조에서는 애트리뷰트의 특성 값을 실제 문서에서 나타나는 문자열의 형태로 저장한다. (그림 13)는 애트리뷰트 인덱스의 구조를 나타낸다. 애트리뷰트가 지니는 특성 값의 길이는 가변이므로 특성값의 길이를 나타내는 Vlen(Value Length)을 실제 특성값(Value)과 함께 저장한다. 또한, 검색시 다음 문서의 포스팅 정보를 빠르게 접근하기 위해서, 문서가 가지는 포스팅 정보의 길이를 PS(Post Size)에 저장 한다. 엘리먼트 이름과 애트리뷰트 이름은 문서의 해당 XML DTD에 따라 고정된 개수로 정해지며, 그 수가 Keyword 인덱스의 색인에 비해 상당히 직다는 특징을 갖는다. 따라서, 엘리먼트 인덱스나 애트리뷰트 인덱스는 빠른 탐색을 위해 인덱스 초기화 시 메모리에 상주시켜 사용할수 있다.



(그림 13) 애트리뷰트 인덱스의 구조

3.3.5 X-트리 인덱스

대용량의 멀티미디어 데이터베이스를 다루는데 있어서 주된 관심사는 검색 효율성이다. R*-트리를 비롯한 여러 색인기법들은 차원의 수(특징의 수)가 증가함에 따라, 시간 또는 공간 요구량이 지수적으로 증가되어 고차원의 색인 구조로서 효율성을 상실하게 된다. 이러한 문제를 해결하기 위해 최근에 고차원 특징 벡터를 수용할 수 있는 다수의 색인 기법들이 연구되었다 [18]. 이 중에서 X-트리는 기존의 색인구조들이 차원이 증가함에 따라 검색 영역이 증가하여 검색 성능이 현저히 저하되는 문제점을 방지하기 위해 제안된 고차원 색인 구조이다. X-트리구조는 분할시 검색 영역이 최소가 되지 못할 때는 분할하지 않고 노드의 크기가 가변적으로 확장될 수 있는 슈퍼노드를 사용한다. X-트리 구조는 저차원에서는 계층적인 디렉토리 구조를 사용하고, 고차원으로 갈수록 검색 영역이 증가되기 때문에 기억 공간이 절약되고 빠른 접근이 가능한 선형적인 디렉토리 구조를 사용한다. X-트리는 고차원 특징벡터 공간에서 k-최근접 탐색 질의와 같이 유사성에 기반한 검색을 효율적으로 지원하기 위한 적절한 색인 구조의 하나로 평가된다. 따라서 본 논문에서는 색상 및 형태의 고차원 특징 벡터를 효율적으로 검색하고 완전 일치 질의(exact match query), k-최근접 탐색 질의(k-nearest neighbor), 범위 질의(range query) 등과 같은 다양한 검색 질의를 지원하기 위해 X-트리를 사용한다. (그림 14)는 색상 및 형태 특징벡터에 대한 X-트리 인덱스 구조를 보여준다.



(그림 14) X-트리를 이용한 색상 및 형태 인덱스

3.4 통합 검색기

기존 텍스트-기반 정보검색 시스템에 대해서는 다양한 검색 모델이 연구되었으나, 멀티미디어 객체를 위한 구조 및 내용-기반 검색 모델에 관한 연구는 많지 않다. 대부분이 시각화된 다양한 길의 정보를 토대로 유사성에 기반하여 유사성이 높은 순으로 브라우징하는 방법이 일반적이다. 본 논문에서는 5가지 타입의 사용자 검색 질의, 즉 구조 질의, 키워드 질의, 애트리뷰트 질의, 내용 질의, 복합 질의 등을 통합적으로 처리한다

첫째, 구조 질의 처리는 사용자 인터페이스를 통해 입력된 질의에 대해서 그에 맞는 구조 인덱스를 탐색하여 질의를 만족하는 결과를 사용자에게 제공한다 또한, 구조 질의의 엘리먼트 term 벡터간의 유사성은 다음과 같은 식을 이용하여 계산한다[19].

$$S_W = \text{COSINE}(NODE_q, NODE_t) = \frac{\sum_{j=1}^m (TERM_{qj} \cdot TERM_{tj})}{\sqrt{\sum_{k=1}^m (TERM_{qk})^2 \cdot \sum_{l=1}^m (TERM_{tl})^2}}$$

구조 질의에서 노드q의 구조 질의 유사성은 노드l의 term벡터와 노드q의 term벡터 간의

$$D_W = \text{MAX}\{ \text{COSINE}(NODE_q, NODE_E), 0 \leq i \leq n-1 \}$$

유사성(S_w)을 나타낸다. 만약 사용자 질의 결과가 문서이면 이문서는 D = {E₀, E₁, ..., E_{n-1}}로 표현한다 이때 E_i는 문서 D에 속한 엘리먼트의 유사성을 나타낸다. 또한, 엘리먼트 q의 구조 질의 유사성은 D_w로 나타내며 다음과 같이 표현한다

둘째, 키워드 질의 처리는 사용자 인터페이스를 통해 입력된 키워드 질의 정보를 받아서 불리안 모델에 근거하여 키워드 인덱스를 탐색한 후, 검색단어를 포함하고 있는 oid들을 구한후 oid에 해당하는 엘리먼트나 문서를 검색한다

셋째, 애트리뷰트 질의 처리는 사용자 인터페이스를 통해 입력된 애트리뷰트 이름과 값을 가지고 애트리뷰트 인덱스를 탐색하여 검색한다.

넷째, 이미지 특징 벡터에 기반한 질의 처리는 구조

질의, 키워드 질의, 애트리뷰트질의 처리와는 달리, 사용자 인터페이스를 통해 선택된 이미지 특징벡터를 분석하여 색인키를 생성하고 X-트리에 저장된 특징벡터와의 유클리디언 거리(Euclidean-Distance)를 이용하여 유사성이 높은 순서로 검색한다. 이미지에 대한 특징 벡터 질의는 색상이나 형태 중에서 하나만 주어지는 경우와 모두 주어지는 경우에 대해 유사성 계산이 다르다. 후자의 경우는 각각에 대해서 유사성을 계산한 후에 색상과 형태를 병합할 때, 색상과 형태의 유사성의 곱으로 나타낸다. 선택되지 않은 검색에 대한 유사성은 1로 하고 전체 유사성을 구한다. 내용질의의 유사성은 다음과 같은 식을 이용하여 계산한다.

$$C_W = \begin{cases} \frac{\text{Distc}(q, t)}{Nc}, & \text{if a shape query is empty} \\ \frac{\text{Dists}(q, t)}{Ns}, & \text{if a color query is empty} \\ \frac{\text{Distc}(q, t)}{Nc} \times \frac{\text{Dists}(q, t)}{Ns} \end{cases}$$

위 식에서 질의 이미지 q와 데이터 베이스 내의 대상 이미지 t사이의 유사도 C_w(q, t)는 사용자 질의 이미지에 대한 하나의 검색된 이미지 사이의 가중치(Weight) 값을 나타내며, 이 값은 실제로 검색된 이미지와 함께 브리우징 된다 여기서 Distc(q, t)와 Dists(q, t)는 각각 질의 이미지 q와 대상 이미지 사이의 색상과 형태에 대한 거리를 계산한 값이다. Nc와 Ns는 각각 색상과 형태의 유사성을 정규화 하기위한 값이다.

마지막으로, 구조 및 내용에 관한 복합 질의의 유사성(T_w)은 다음과 같은 식을 이용하여 계산한다

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

복합 질의에서 만약 사용자가 원하는 결과가 문서 검색 질의인 경우 내용질의와 구조질의 결과로 찾아진 문서들의 구조 유사성(D_w)은 각 문서에 속한 엘리먼트의 유사성(d_i)중 가장 큰 값이 그 문서의 유사성(D_w)을 대표한다. 따라서, 문서의 구조 질의 유사성의 대표 값과 내용질의 유사성의 값에 α와 (1-α)를 곱한 값을 합한 것으로 통합 검색 유사성(T_w)을 나타낸다. 그리고 사용자가 원하는 결과가 엘리먼트 검색인 경우 통합 검색 유사성 또한 검색된 각각 엘리먼트의 유사성

(S_w)과 내용 길의 유사성(C_w)에 α 와(1- α)를 곱한 값을 합한 것으로 나타낸다.

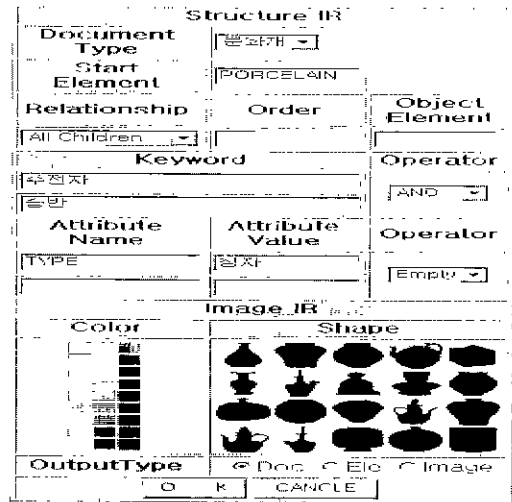
3.5 사용자 인터페이스

본 논문에서 사용한 XML문서는 국립중앙박물관의 청자, 백자, 분청사기 도감에서 추출한 문서로서 XML DTD형식에 맞게 구성하여 사용하였다. 아울러 초고속 이동 통신망에서 사용자는 원하는 XML문서를 검색하기 위해 구조적인 정보와 내용 정보에 관하여 검색 질의를 수행한다. 사용자 인터페이스에서 처리할 수 있는 질의의 유형을 분류하면 단순질의, 복합질의로 나눌 수 있으며, 단순질의에는 키워드, 구조, 엘리먼트, 애트리뷰트, 이미지 색상, 이미지 형태 질의가 있다. 먼저 단순질의의 예를 들면 다음과 같다.

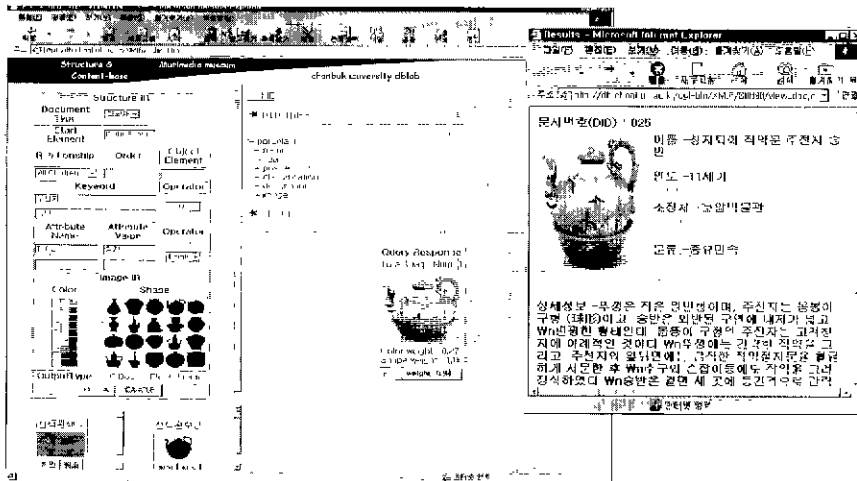
- 키워드질의 : '귀현사기'가 있는 문서를 찾아라.
- 구조질의 : [도자기]엘리먼트의 자식 엘리먼트를 찾아라
- 엘리먼트질의 [museum]엘리먼트가 있는 문서를 찾아라
- 애트리뷰트질의 : type이 백자인 문서를 찾아라.
- 이미지질의 : 특정 색상 또는 특정 형태를 가진 이미지를 찾아라.

또한 복합질의는 단순질의의 복합 형태로서 그 예는 다음과 같다.

- 구조 + 키워드질의 : [detail]엘리먼트에 '십선십선'이 있는 문서를 찾아라
- 구조 + 애트리뷰트질의 [도자기]엘리먼트의 자식 엘리먼트가 있는 문서 중 type이 청자인 문서를 찾아라
- 이미지-키워드질의 : 파란색 이미지를 가진 문서 중 '귀현사기'가 있는 문서를 찾아라.
- 이미지+구조질의 : 이미지가 청산 형태를 가진 문서 중 [도자기]엘리먼트의 자식 엘리먼트를 찾아라.



(그림 15) 사용자 질의 인터페이스 예



(그림 16) 구조 및 색상 질의 예

다음 (그림 15)는 사용자 질의 인터페이스를 나타내며, 질의 입력부분은 키워드, 애트리뷰트, 구조, 이미지 색상, 그리고 이미지 형태에 대한 질의 중 필요한 정보를 입력 받는다. 예제에서는 PORCELAN 엘리먼트의 All Children 엘리먼트 중에서 '주전자'와 '승반' 키워드를 포함하고 TYPE 애트리뷰트의 내용이 청자이며, 색상은 청색이고 주전자 형태의 이미지를 포함한 문서 검색에 대한 질의를 나타낸다. (그림 16)은 (그림 15)의 구조 및 내용 질의에 대한 결과를 나타내며, 보다 상세한 정보를 원하는 문서에 대해서는 별도의 창을 통해 브라우징 해준다.

4. 구현 및 성능평가

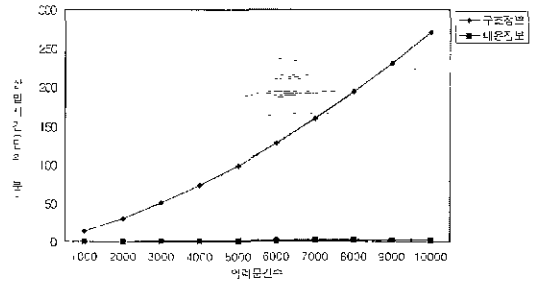
본 논문에서 구현한 내용 및 구조-기반 멀티미디어 문서검색 시스템은 GNU CCv2.7 컴파일러를 사용하여 SUN Sparc20 에서 개발되었다. 하부 저장 기반 시스템으로는 O2 OODBMS v4.6을 사용하였으며[20], XML 파서로는 Sp-1.3을 사용하였다 <표 1>은 성능평가를 위해 사용한 데이터 자료를 나타낸다. 먼저, 시스템 Efficiency에 대한 성능평가를 위해 검색시간, 삽입시간, 부가 저장공간을 측정하였고, 시스템 Effectiveness에 대한 성능평가를 위해 대학원생 10명으로 구성된 테스트 집단을 통해 정확율과 재현율을 측정하였다

<표 1> 테스트 데이터 자료

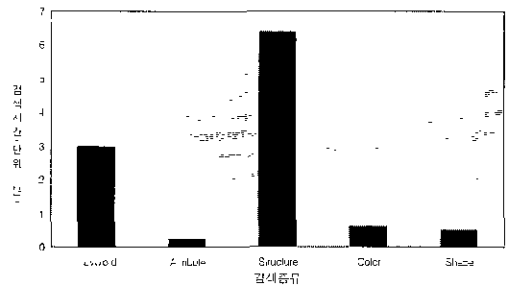
문서 갯수	10000건의 XML 문서
문서평균크기	1.29K + 40.7K (image)
엘리먼트갯수	165,800개

멀티미디어 문서의 삽입 시간은 (그림 17)과 같다. 구조 정보 삽입시간은 키워드, 애트리뷰트, 구조 엘리먼트 인덱스에 각각 정보를 추가하는 시간을 의미하며, 내용 정보 삽입시간은 X-트리 인덱스에 색상 및 형태 질의 정보를 추가하는 시간을 의미한다. 아울러 구조 정보의 경우 XML문서당 삽입시간은 약1.6초 정도가 걸리며, 내용정보 삽입시간은 0.02초 미만이 소요된다. (그림 18)은 키워드, 애트리뷰트, 구조, 색상 그리고 형태 질의에 대한 각각의 실시간 검색시간을 나타낸다. 애트리뷰트, 색상, 형태 질의의 검색시간은 1초 이하이며, 키워드 질의의 검색시간은 3초, 구조 질의 검색시

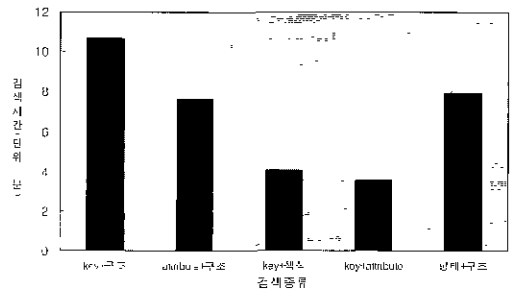
간은 6.5초이다. 따라서, 구조질의 경우가 가장 많은



(그림 17) 문서 삽입시간



(그림 18) 단순질의의 검색시간



(그림 19) 복합질의의 검색시간

시간이 소요됨을 알 수 있다 또한 (그림 19)은 복합질의의 경우에 대한 검색 결과를 나타내며 각 복합질의에 대한 결과를 보여주고 있다. '구조+키워드' 질의는 10.5초, '구조+애트리뷰트' 질의는 7.6초, '색상+키워드' 질의는 1초, 키워드-애트리뷰트 질의는 3.5초 그리고 구조+형태 질의는 약 8초가 걸린다. 복합질의에서도 구조질의가 포함된 질의가 시간이 많이 소요됨을 알 수 있다

부가 저장공간은 <표 2>에 나타나며, 원 데이터의 약 1000%정도가 소요된다

<표 2> 부가저장공간

원 데이터 크기	17056K
인덱스 파일 크기	구조정보(183562K) + 내용정보(6568K)
부가저장공간	1015%

시스템 Effectiveness의 평가 기준인 정확율과 재현율은 다음 식으로 계산된다[19].

$$\text{정확율} = \frac{\text{검색된 이미지들 중에서 테스트집단이 선택한 것과 같은 관련된 이미지 수}}{\text{시스템에 의해 검색된 이미지 수}}$$

$$\text{재현율} = \frac{\text{검색된 이미지들 중에서 테스트집단이 선택한 것과 같은 관련된 이미지 수}}{\text{테스트 집단에 의해 선택된 관련 이미지 수}}$$

본 논문에서 제안한 시스템의 정확율과 재현율은 <표 3>과 같다. k값을 증가시키면 정확율은 낮아지는 반면 재현율은 높아지는 결과를 보인다. 즉 많은 수의 질의 결과를 추출하도록 시스템을 설계하면 재현율이 높아진다. 그러나 너무 많은 결과를 추출할 경우 정확율이 상당히 낮아지는 경우가 발생하기 때문에 적절한 값을 사용한다. 본 논문에서는 k값을 10으로 사용한다. 한편 k가 10일때, 직상에 대한 정확율과 재현율은 약 0.6, 0.4이며, 형태에 대해서는 약 0.6, 0.3으로 둘다 거의 유사한 결과를 나타내고 있다. 단 k값은 k nearest neighbor 검색의 k값을 나타낸다.

<표 3> 정확율과 재현율

	Color		Shape	
	정확율	재현율	정확율	재현율
K = 7	0.64	0.33	0.57	0.23
K = 10	0.60	0.37	0.55	0.32

마지막으로 본 논문에서 구현한 시스템과 타 문서검색 시스템을 비교하기 위해 각 시스템에서 지원하는 질의 타입과 검색시간 관점에서 비교를 수행한다. <표 4>는 질의타입 비교 결과를 나타낸다.

타 시스템의 경우 구조기반질의 또는 내용기반질의만을 지원하고 있어서 '구조 + 내용기반' 형태의 복합질의에 대한 검색을 지원하지 못한다. 반면 본 논문에서 제안한 시스템은 구조 및 내용기반 질의를 모두 지원하므로, '구조 + 내용기반' 형태의 복합질의도 훌륭히

<표 4> 본 논문에서 제안한 시스템과 타 시스템의 질의타입 비교

	Our System	StIR[3]	COSMIC[8], NoD[13]	CBIR [9]
에트리뷰트질의	가능	가능	가능	없음
구조기반질의	가능	가능	없음	없음
내용기반질의	가능	없음	가능	가능
복합질의	가능	없음	없음	없음

지원하고 있다. 또한 <표 5>는 시스템간의 검색시간 비교 결과를 나타낸다. 본 논문에서 제안하는 시스템과 비교할 수 있는 구조 및 내용기반 검색 시스템이 존재하지 않기 때문에, 구조기반검색과 내용기반검색을 각각 StIR, NoD 시스템과 비교하였다. COSMIC의 경우 검색시간을 측정하고 있지 않기 때문에 비교대상에서 제외하였다. 구조기반질의의 경우 검색시간은 약 6초로서 StIR 시스템과 유사한 성능을 나타내고 있으며, 내용기반질의의 경우 본 논문에서 제안한 시스템은 검색시간이 1초 이내로 NoD 시스템보다 우수한 성능을 나타내고 있다.

<표 5> 본 논문에서 제안한 시스템과 타 시스템의 평균 검색시간 비교

	Our System	StIR[3]	NoD[13]
구조기반질의	65초	63초	N/A
내용기반질의	0.6초	N/A	3.5초

(N/A Not Acceptable)

5. 결론 및 향후 연구

본 연구에서는 다양한 멀티미디어 문서에 대한 구조 및 내용-기반 검색을 위해 XML문서를 이용한 문서검색 시스템을 설계 및 구현하였다. XML문서의 키워드 및 구조 정보의 검색을 위해서 엘리먼트 단위의 구문 트리 저장 구조를 o2store 하부저장 시스템을 사용하여 구축하였고, 내용에 기반한 이미지 정보의 지장을 위해서 고차원 색인 기법인 X-트리를 사용하였다. 따라서 XML문서의 복잡한 구조정보와 내용에 기반한 이미지 정보를 효율적으로 저장하고 검색할 수 있는 시스템을 구축하였다. 본 논문에서 구현한 구조 및 내용-기반 멀티미디어 문서검색 시스템의 한 개의 문서 삽입시간은 평균1.6초이며, 검색시간은 복잡한

구조질의가 포함된 경우 평균 9초가 구조질의가 포함되지 않는 질의는 4초 이하의 시간이 소요된다. 또한 이미지 색상 및 형태 질의에 대한 내용-기반 검색의 정확율 및 재현율은 약 0.6, 0.4 정도의 결과를 보였다. 향후 연구과제로는 하부저장구조로 사용된 o2store를 Public 저장 시스템인 Shore로 대체하고, 다양한 DTD에 대해 질의 처리가 가능한 시스템을 설계 및 구현하는 것이다.

참 고 문 헌

[1] Extensible Markup Language(XML), <http://www.w3.org/TR/PR-xml-971208>.

[2] R Sack-Davis, T. Arnold-Moore and J Zobel, Database Systems for Structured Documents, Informational Symposium on Advanced Database Technologies and Their Integration, 1994.

[3] 손정환, 한성근, 장재우, 주종천, SGML 경로 검색 인덱스 설계를 위한 K-ary트리, 문서 단위 구문 트리와 엘리먼트 단위 구문 트리의 비교, '98한국정보과학회 가을 학술 논문집 Vol.25, No 2, pp 383-385, 1998.

[4] W. Niblack, et al, The QBIC project : Querying by Image Content Using Color, Texture, and Shape, Proc. of SPIE Storage and Retrieval for Image and Video Databases, pp.173-187, 1993.

[5] J R Smith, S F. Chang, VisualSEEK : a Fully Automated Content-Based Image Query System, ACM Multimedia Systems, Nov 1996

[6] Virage, <http://www.virage.com>

[7] Mpeg-7, <http://drogo.csell.it/mpeg/standards/mpeg-7/mpeg-7.htm>.

[8] 김덕환, 김시우, 박광순, 이병구, 차필호, 정진완, COSMIC 영역지식과 시각정보를 이용한 내용기반 멀티미디어 검색 시스템의 설계 및 구현, 정보과학회 논문지 제5권 제1호, 1999.

[9] Youngsung Kim, Whoyul Kim, Myoungjoon Kim, Development of Content-based Trademark Retrieval System on the World Wide Web, ETRI Journal, March, 1999

[10] 황분우, 노영기, 이성환, 색상 및 형태 정보를 이용한 내용기반 영상 검색 시스템의 Web 상에서의 구현, 한국정보과학회 봄 학술 발표 논문집, 제25권 제1호,

pp.607-609.1998

[11] Youngman Ro, Texture featuring and indexing using matching pursuit in Radon space, IEEE ICIP.1999.

[12] Soveon Kim, Youngman Ro, Fast content-based MPEG video indexing using object motion. Proceeding of IEEE TENCON, pp.1509, 1999.

[13] 진기성, 경세욱, 장재우, NoD서비스용 멀티미디어 데이터의 효율적인 저장 및 검색을 위한 하부저장 관리자의 설계 및 구현, 한국정보처리학회 논문지, 제7권 제4호, 2000

[14] S. Berchtold, D. A. Keim, H-P Kriegel, The X-tree . An Index Structure for High-Dimensional Data, Proceeding of the 22nd VLDB Conference, pp.28-39,1996

[15] <http://www.jclark.com/sp>.

[16] J. C. Bezdek, M M Triedi, Low level segmentation of aerial image with fuzzy clustering, In IEEE Trans. SMC. SMC-16, pp.589-598, 1986.

[17] Choon-Bo Sim, Kwang-Taek Song, Jae-Woo Chang, Joon-Whoan Lee, and Jae-Dong Yang, Design and Implementation of a Content-Based Multimedia IR System for Cyber Museums, SPIE Electronic Imaging and Multimedia Systems II, pp 86-93, 1998

[18] Stefan Berchtold, Daniel A. Keim, High-Dimensional Index Structures, Database Support for Next Decades Applications (Tutorial), SIGMOD Conference, 1998.

[19] Salton, G., and M McGill, An introduction to Modern Information Retrieval, McGraw-Hill, 1983

[20] O. Deux et al. The O2 System, IEEE Communication of the ACM. Vol 34, No.10, 1991.



진 두 석

e-mail : dsjin@dblab.chonbuk.ac.kr
 1999년 전북대학교 컴퓨터공학과 (공학사)
 1999년~현재 전북대학교 컴퓨터 공학과 석사과정
 관심분야 : 멀티미디어 데이터베이스
 멀티미디어 정보검색



이 정 재

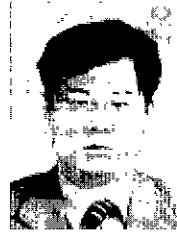
e-mail jlee@dblab.chonbuk.ac.kr

1997년 전북대학교 컴퓨터공학과
(공학사)

2000년 전북대학교 컴퓨터공학과
(공학석사)

2000년~현재 케이원 시스템㈜ 근무

관심분야 : 멀티미디어 데이터베이스, 멀티미디어 정보 검색



장 재 우

e-mail : jwchang@dblab.chonbuk.ac.kr

1984년 서울대학교 전자계산기공
학과(공학사)

1986년 한국과학기술원 전산학과
(공학석사)

1991년 한국과학기술원 전산학과
(공학박사)

1996년~1997년 Univ. of Minnesota, Visiting Scholar.

1991년~현재 전북대학교 컴퓨터공학과 부교수

관심분야 : 멀티미디어 데이터베이스, 멀티미디어 정보
검색, 히부저장구조