

# XML을 이용한 내용기반 이미지 데이터베이스의 설계 및 검색 시스템 구현

## (Design of Content-Based Image Database and Development of Retrieval System using XML)

박 선 영 <sup>†</sup>    용 환 승 <sup>\*\*</sup>

(Sun-Young Park)(Hwan-Seung Yong)

**요약** 내용기반 이미지 검색을 하기 위해서는 이미지에 대한 내용정보가 필요하며, 이러한 내용정보 간에는 상호 연관성이 존재한다. XML(eXtensible Markup Language)은 내용정보간의 상호연관성을 표현하기에 적절하므로 본 논문에서는 이미지 내용정보를 구조화하기 위한 방법으로 XML을 사용하였다.

또한 이미지는 눈에 보이는 객체의 시각적인 특징과 이미지 전체가 내포하는 서술적인 의미도 가지므로, 이러한 이미지의 특성에 따라 내용정보의 구조도 객체의 시각적 특징 중심의 모델링과 의미 중심의 모델링으로 구분하여 XML 문서 구조를 모델링 하였다. 구조화된 모델들 간의 객체지향 특성을 이용하여 XML 데이터 서버인 eXcelon에 통합하고, 이를 XQL(XML Query Language)에 의하여 질의해 낼으로써 검색 구간에 제약을 가하고 이를 통하여 더욱 효과적인 검색을 지원하도록 한다. 검색되어진 XML 문서 구조는 XSL(extensible StyleSheets Language)의 적용을 통하여 쉬운 형태로 웹 브라우저 상에 출력하도록 한다.

마지막으로 본 논문에서 제안한 모델링의 효율성을 검증하기 위하여, 웹 상에서 이미지 내용정보의 상호 연관성에 기반 하여 원하는 이미지를 검색할 수 있는 검색 시스템을 구현하였다.

**Abstract** Content-based image retrieval requires acquiring content descriptions from image, in which interrelationship presents. As for the method to structure content descriptions, this paper chooses XML (eXtensible Markup Language), because XML is appropriate to express the interrelationship between content descriptions.

As image has both visual features of visible objects and semantic features conveyed by the whole image, the structure of XML is modeled for the structure of content description to be divided to visual feature centered modeling and semantic feature centered modeling. Object-Oriented features of the structured model unifies image content description to XML Data Server, eXcelon and executes retrieval by XQL limiting search area. This will lead to the more efficient results. The structure of XML documents resulted from the retrieval is made easy to be printed in the web browser by applying an extensible StyleSheets Language(XSL).

In order to validate the efficiency of the proposed modeling in this paper, a retrieval system is implemented, which retrieves images using interrelationship between image content descriptions on the web.

## 1. 서론

· 이 논문은 2000년도 두뇌한국 21사업에 의하여 지원되었음.

<sup>†</sup> 비 회 원 : LG전자기술원 정보기술연구소 연구원  
ehcom@Lg-Elite.com

<sup>\*\*</sup> 종신회원 : 이화여자대학교 컴퓨터학과 교수  
hsyong@mm.ewha.ac.kr

논문접수 : 2000년 1월 3일

심사완료 : 2000년 10월 10일

이미지는 멀티미디어에서 많이 사용되는 자료형이며 [1] 생물학 정보, 의료 정보, 기상 정보, 예술 정보 등 다양한 분야에서 많은 이미지가 생성되고 있다. 이러한 많은 이미지들 중에서 원하는 이미지를 어떻게 찾을 것인가 하는 풀음이 제기 되었고, 이에 따라 이미지를 효과적으로 검색할 수 있는 여러 가지 연구가 진행되고 있다[1-7].

이미지를 검색하기 위하여 주로 사용되는 방식은 이

미지의 내용에 기초하는 내용기반 이미지 검색 방식이다. 즉, 이미지 분석을 제공하기 위하여 이미지의 내용 정보를 데이터베이스 내에 저장하고, 저장된 내용정보를 통하여 사용자가 원하는 이미지를 질의(query)에 의하여 검색해 낼 수 있다.

이미지의 내용 정보를 추출하는 첫 번째 방식은 자동으로 이미지의 내용 정보를 추출하여 검색에 사용하는 방법이다[4, 6]. 이 방법은 자동으로 색상, 질감, 형태 등과 같이 눈에 보이는 특징을 자동으로 추출해 내는 것으로서, 자동화의 이점은 있지만 알고리즘 개발의 어려움이 있으며, 자동화를 위한 비용도 비싸다[8]. 또한 추출해 내는 정보도 극히 미약하며 단순하다[1, 2].

두 번째 방식은 키워드(keyword), 추상화된 단어나 텍스트 형태의 의미 정보 등과 같이 자동 인식이 어려운 좀더 구체적인 요소를 수동으로 추출해 내는 방식으로, 직접 기술에 의하여 이미지의 내용정보를 구축하는 방법을 사용한다[5, 7]. 이 방법은 서술 정보에 사용되는 어휘가 사용자 질의의 표현 어휘와 정확히 일치하지 않을 수 있을 뿐 아니라 기술해야 하는 의미의 양이 거의 폭발적으로 증가할 수 있다는 단점을 가진다. 그러나 풍부한 의미를 기술할 수 있으며, 저 차원적인 이미지 연산 비용이 너무 비싸므로 많이 사용되는 방식이다[8].

세 번째 방식은 위의 두 가지 방법을 혼용하여 사용하는 방법이다[2, 8, 9]. 이미지 처리 기법을 통하여 몇 가지 특징을 추출해 내고, 자동으로 추출이 불가능한 서술 정보를 수동으로 기술해 주는 방법이다.

앞에서 설명된 이미지 검색 방식은 이미지 내용정보를 어떻게 추출해 내는가에 주로 초점을 맞추고 내용정보간의 상호관계를 어떻게 표현해 줄 것인가에 대해서는 생각하지 않았다. 따라서 내용정보간의 상호관계가 사라져 버리게 되고 이 상호 관계 정보를 검색에 이용할 수 없게 되었다.

본 논문에서는 이러한 단점을 극복하기 위하여 어떠한 방법으로 내용정보간의 관계를 표현할 것인가에 그 초점을 맞추려 한다. 특히 점차 표준화되어 가는 웹상의 표현 방식인 XML을 사용하고 이를 XML 데이터 서버인 eXcelon에 저장함으로써 다음의 효과를 얻을 수 있다.

- 이미지 내용정보간의 상호 연관성을 검색에 사용 가능
- 검색을 위한 연산 비용 절감 효과
- 데이터베이스의 교환과 통합의 용이
- 이미지 내용정보를 설명 정보로써 재사용

- XSL을 이용한 다양한 형태로의 출력

따라서 이미지 내용정보를 XML 형태로 구조화하기 위한 모델링이 필요하며, 이미지 검색을 위한 편리한 인터페이스와 다양한 질의어가 제공되어야 한다. 본 논문에서는 기존의 이미지 검색 방법에서 고려되지 않았던 이미지 내용정보간의 관계를 정의하기 위한 XML 모델을 제시하고, 이 모델의 객체 지향적인 특성을 이용하여 XML 데이터 서버인 eXcelon에 저장하려 한다.

즉, 이미지 내용정보간의 상호 연관성을 기반으로 하여 계층적 구조를 정의하고 이에 맞는 XML 문서 구조를 제시하도록 DTD를 설계한다. 설계된 DTD를 기반으로 한 이미지 내용정보를 eXcelon에 통합하고, 다양한 검색 질의 함수와 편리한 사용자 인터페이스를 지원하도록 한다.

본 논문에서 제안하는 이미지 내용정보의 저장 방식과 질의어의 효율성을 검증하기 위하여 웹 상에서 이미지 데이터를 검색하고, 검색되어진 결과를 XSL의 적용을 통하여 검색된 이미지와 함께 내용정보를 다양한 형태로 웹 브라우저 상에 보여주는 검색 시스템을 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 이미지 검색 방법에 관한 연구와 그 한계점을 XML 기반 방식과 비교하여 소개한다. 3장에서는 XML의 특징에 기반 하여 이미지의 내용정보에 대한 XML 문서 구조 모델을 설계한다. 4장에서는 3장에서 제안된 XML 문서 구조 모델을 통하여 실제 검색 시스템을 구현한다. 마지막으로 5장에서는 본 논문의 결과를 정리하고 앞으로의 연구 방향을 제시한다.

## 2. 기존의 이미지 검색 방식 및 한계점

기존의 내용기반 이미지 검색 방식에 있어서, 이미지 데이터베이스의 검색 능력은 유사성 정도와 검색 시간에 영향을 받는다.

유사성 정도는 이미지의 특성을 비교하기 위하여 사용되는 색상, 질감, 형태, 객체간의 관계 등의 추출이 실제 이미지와 얼마만큼 유사할 것인가를 나타내는 것인데, 이것이 얼마만큼 정확하게 표현되어 있는가 하는 것은 검색 시스템의 성능을 좌우한다. 자동화 기법을 통한 추출은 컴퓨터의 판단이 인간의 판단과 얼마만큼 일치할 것인가와 매우 관련 깊은데, 컴퓨터를 통한 자동화 기법은 완벽하게 인간의 판단과 일치할 수 없다.

다량의 데이터베이스에서는 이미지마다 요구되어지는 검색 시간이 길 경우 효과적이지 않다. 더욱이 검색되어진 이미지로부터 정확히 원하는 이미지를 검색하기 위

해서는 검색 조건을 재정의 해야하고, 이를 통한 검색 시간은 증가할 것이다.

이러한 이미지 데이터베이스의 문제를 풀기 위한 해법으로 많은 검색 기법들과 형태[13], 색상[6], 형태와 색상 인덱스의 조합[3, 14]에 기초한 다양한 인덱싱 방법이 제시 되어왔다.

UC Berkeley에서 개발된 Chabot 프로젝트는 POSTGRES라는 데이터베이스 시스템 위에 B트리와 R 트리 기법을 이용하여 단순 속성, 객체에 대한 설명 정보를 이용한 텍스트 기반 검색과 색상분석 기술을 이용한 내용기반 검색을 지원한다[4].

Columbia 대학에서 개발된 VisualSEEK 시스템은 색상/공간 질의를 제공하기 위한 이미지 데이터베이스 시스템이다. 이미지의 구분은 색상 히스토그램과 같은 전형적인 특성에 의하여 이루어지는데, 특별한 이미지의 비교를 위해서 이미지의 영역과 색상, 크기, 공간적 위치들을 추가적으로 사용하는 방식이다[6].

그러나 이러한 방식들에서는 이미지가 갖는 의미 있는 정보가 사라질 수 있다. 예를 들어 “사과와 딸기가 같은 색상 히스토그램을 갖는다면 같은 이미지로 볼 수 있는가?” 와 같은 물음에서 대답은 “아니오”이다. 이는 자동화 도구의 판단이 인간의 판단과 완전히 일치 할 수 없기 때문이다. 또한 이미지의 내용을 묘사하고 비교하기 위하여 색상, 질감, 형태 정보와 같은 원시적인 특징을 사용하는 내용기반 검색 방식들은 객체간의 상호 관계에 대해서는 고려되지 못하였다[15]. 그림 1을 통하여 이러한 기존의 이미지 검색 방식에 있어서의 한계점과 본 논문에서 제안하는 XML 사용의 장점을 설명하도록 하겠다.

그림 1과 같은 이미지를 통하여 추출해 낼 수 있는 특징들은 이미지에 포함된 객체의 색상, 형태, 윤곽선,



그림 1 상호 연관성을 표현하기 위한 이미지의 예

주제, 키워드 등이다. 기존의 저장 방식에서는 색상 분석 기술[4, 6]이나 윤곽선 추출 기법[8]에 의하여 얻어진 정보를 통한 유사도에 의한 검색 방식을 사용하거나 키워드 추출에 의한 방법[4], 공간 질의를 통한 위치 정보[6] 등을 검색에 사용한다. 이러한 방식에서는 이미지 내용정보간의 상호 관계를 추출해 낼 수 없다.

예를 들어 색상 분석 기술의 경우 이미지에 포함된 대체적인 색상에 대한 색상 질의는 가능 하지만 “노란색 옷을 입은 여자 아이”와 같은 사용자 질의가 요청 될 경우, 그림 1과 같은 이미지 뿐 아니라 “노란색 꽃을 포함한 이미지”, “노랑 우산이 포함된 이미지”와 같이 노란색을 포함한 이미지들도 모두 검색된다. 이 때 사용자는 원하는 이미지를 얻기 위하여 질의를 재정의 하여야 하며, 이 경우 검색 시간은 증가하게 된다.

기존의 자동화를 통하여 추출된 이미지의 특성이 인간의 판단과 정확히 일치 할 수 없고, 추출해 낸 “노란색” 이라는 색상 정보와 “옷” 이라는 객체 내의 특징, 그리고 그러한 특징을 갖는 “여자 아이”라는 객체의 상호 연관성이 검색에 반영되지 못하므로 이러한 문제가 발생하게 된다.

```

예1) retrieve (q.all) from q in photoce_bib
      where q.description ~ “여자아이” and
      MeetsCriteria( “SomeOrange” ,q.histogram)
    
```

표 1은 UC Berkeley에서 개발된 이미지 검색 방법인 Chabot 프로젝트의 스키마(schema)이며, 예1)은 이러한 데이터베이스 스키마로 “노란색 옷을 입은 여자아이”를 찾기 위한 질의문을 표현한 것이다[4].

앞에서 설명한 것과 같이 표 1의 스키마와 예1)과 같은 질의문으로는 “여자아이가 있고 노란색을 포함하는 이미지”를 검색할 수 있다. 그러나 위의 질의 예에서는

표 1 photo\_bib class를 위한 애트리뷰트

Attribute	Type	Description
Abstract	Text	문서의 개요
Title	Text	문서의 제목
..... 중간 생략		
description	Text	이미지에 대한 정보
Color	Char	C(색상) B(감정과 흰색)
orientation	Char	H(수평) V(수직)
histogram	Text	색상 히스토그램
entry_date	Abstime	Db에 들어간 날짜
shoot_date	Abstime	이미지를 찍은 날짜
Oid	Oid	Postgres 객체 ID

이미지 객체와 객체 내의 속성에 관한 상호 연관성을 표현할 수 없으므로, 원하는 이미지 외에 “노란색 우산을 든 여자 아이” 혹은 “노란색 차 앞에서 있는 여자 아이”와 같은 이미지도 같이 검색될 수 있다. 따라서 원하는 이미지를 바로 검색해 낼 수 없고, 정확한 이미지를 선택하기 위한 검색 시간은 증가하게 된다.

```

<이미지>
<객체>
<객체명>남자 아이</객체명>
<객체동작>앉아있다</객체동작>
<속성 부위=" 옷" >
<이름>티셔츠</이름>
<색상>연두색</색상>
</속성>
<속성 부위=" 옷" >
<이름>반바지</이름>
<색상>하늘색</색상>
</속성>
</객체>
<객체>
<객체명>여자 아이</객체명>
<객체 동작>앉아있다</객체동작>
<속성 부위=" 옷" >
<이름>원피스</색상>
<색상>노란색</색상>
</속성>
</객체>
... 중간 생략
</이미지>
    
```

그림 2 XML을 통한 그림 1의 구조화 예제

그림 2는 그림 1을 이미지 내의 객체 정보 중심으로 XML을 이용하여 구조화한 예를 보여 준다. 이 경우 “노란색 옷을 입은 여자 아이”와 같은 질의를 사용자가 요청할 경우 정확한 이미지를 검색해 낼 수 있다. XML의 트리 구조는 상위 노드의 특성이 하위 노드에 상속되고, 태그의 자기 표현(self describing)특성[16]은 각 엘리먼트와 에트리뷰트가 무엇을 의미하는지를 태그 안에 구체적으로 표현함으로써 내용정보간의 상호 관계를 표현하기 적절하기 때문이다. 위의 질의를 XQL 질의를 통하여 표현하면 다음과 같다.

```

//객체[객체명=" 여자 아이" $and$속성[@부위=" 옷" $and$색상=" 노란색" ]]
    
```

표 2는 기존의 이미지 검색 방식의 특징과 문제점을 설명하고 본 논문에서 제안하는 XML 문서 구조 사용의 특징을 보여준다.

또한 XML을 이용한 내용 기반 이미지 검색 방식에서는 다음과 같은 특징을 갖는다.

표 2 이미지 검색 방식 비교

	지원 질의	내용정보 추출 방식	문제점
Chabot	색상 및 텍스트 기반 질의	색상 히스토그램과 수동의 키워드 입력 방식	• 내용정보간의 상호 연관성을 배제한 데이터베이스 구조. • 구조화된 복합질의 지원 어려움.
Visual-SEEK	색상 및 공간 질의	color set과 back-projection을 통한 색상 추출 방식[17]과 자동화된 영역 추출 방식	• 이미지 검색을 위한 원시적 특징과 객체 영역간의 공간정보의 조합을 위한 매칭 알고리즘은 실제로 구현하기 복잡. • 일반적인 이미지 검색 방식에 적용하기 어려움[18]
QBIC	예제 이미지를 통한 유사도 질의, 사용자 스케치에 의한 질의, 색상 및 질감 패턴에 기반 한 질의	database population을 통한 추출 (지동 추출이던 수동 추출이던 미리 추출된 내용정보를 미리 데이터베이스에 저장)	• 유사도 함수는 이 인 간의 판단과 정확히 일치 못함. • 의미 정보를 통한 검색 불가능.
Visual Keyword 사용방식	Visual 키워드 질의	Typed Aggregate Map과 Coded Tam을 통한 Visual 토큰 추출	• 특정 도메인에 국한된 추출 방식.
XML 사용방식	시각적 정보, 의미 정보에 의한 구조화된 질의	객체의 시각적 특징과 의미 특징에 따른 내용정보 추출	• 구조화 질의 제공 및 질의 확장 가능성 제시.

XML은 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이며[16], 다양한 형태로의 필터링(filtering)과 조합을 제공한다. 이를 통하여 여러 장소에서 생성된 문맥기반 이미지 데이터베이스를 표준화된 방식으로 교환이 가능하며, 동적인 태그의 추가와 삭제를 통한 통합이 용이하다.

데이터베이스 내에 저장된 내용정보는 검색된 이미지와 함께 이미지의 설명 정보로써 재사용 가능하다. 즉, 조건에 맞는 이미지뿐만 아니라 검색된 이미지의 내용정보를 동시에 검색 결과로써 보여 줄 수 있다. 이때 이미지 내용정보는 다양한 형태로의 출력이 가능한데, 검색하고자 하는 내용정보를 다양한 XSL의 적용을 통하여 다양한 형태로 출력 할 수 있기 때문이다[22]. 따라서 같은 내용정보를 사용하여 사용자의 요구에 맞는 다양한 검색 형식을 출력해 낼 수 있다.

### 3. 자연사 이미지 데이터 모델링

#### 3.1 XML 문서 구조 설계

이미지를 구성하는 요소 중 가장 핵심이 되는 것은 이미지에 포함되어 있는 객체이다. 따라서 이미지에 포함된 각 객체들의 내용을 중심으로 한 구조화가 필요하며, 이를 위하여 객체의 시각적 특징을 중심으로 한 모델링이 필요하다. 그러나 이미지의 종류에 따라 객체의 특성 뿐 아니라 이미지 전체가 내포하는 의미에 더 초점을 맞추어야 하는 경우도 있다. 따라서 이미지가 내포하는 의미를 표현하기 위한 의미를 중심으로 한 구조화도 필요하다.

본 논문에서는 내용정보가 표현하는 정보의 종류에 따라서 이미지에 포함된 실제 객체에 대한 시각적 특징 정보 구조와 이미지 전체의 서술적 의미를 포함하는 의미 정보 구조로 내용정보를 크게 나누고 이러한 정보에 대하여 각각을 XML을 사용하여 구조화하도록 한다[19, 20]. 검색 문맥 지정이 가능한 XQL의 특성[10, 21]을 이용하여, 사용자가 원하는 범위에서만 검색이 이루어지고, 질의 종류에 따른 빠른 검색 효과를 기대 할 수 있다.

##### 3.1.1 객체의 시각적 특징 중심 모델링

이미지는 여러 개의 객체들로 구성되며, 이미지에 포함된 객체들의 특징에 따라 이미지의 종류와 범위가 결정된다. 따라서 이미지에 포함된 각 객체들을 중심으로 그들의 시각적인 특징을 내용정보로 표현해야 한다.

예2) 그림 3의 이미지 특징을 결정하는 가장 중요한 요소는 이미지에 포함된 “가시고기” 라는 객체이다. 따라서 이 객체의 시각적인 특징을 추출하여 내용정보로 사용하고, 이미지 검색에 이 정보를 이용 할 수 있다.

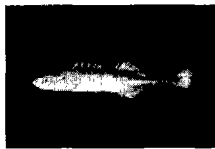


그림 3 가시고기

기본적으로 각 객체들은 색상, 형태, 동작, 객체의 특수한 특질 등의 시각적 특징은 같고, 이러한 속성은 서로 상호 연관적인 관계를 가질 수 있다. 상호 연관적인 객체의 특징을 검색에 반영하기 위해서는 이를 표현할 수 있는 구조적인 모델이 필요하다. 본 논문에서는 XML DTD를 사용하여 각 객체의 시각적인 특징 정보를 구조화하도록 하였다. 그림 4는 객체의 시각적 특징을 중심으로 모델링한 XML DTD 구조이다.

그림 4에서 나타낸 것처럼 객체는 각 부위별로 색상, 형태, 동작, 특정 부위가 갖는 특질의 속성을 갖는다.

```
<!ELEMENT visual-object (object-name, characteristics*)>
<!ELEMENT object-name (#PCDATA)>
<!ELEMENT characteristics (color?, shape?, behavior?,
    object-specialty*)>
<!ATTLIST characteristics part CDATA #IMPLIED>
<!ELEMENT color (#PCDATA)>
<!ELEMENT shape (#PCDATA)>
<!ELEMENT behavior (#PCDATA)>
<!ELEMENT object-specialty (#PCDATA)>
```

그림 4 객체의 시각적 특징을 중심으로 한 DTD 구조

characteristics(속성)는 애트리뷰트로서 part(부위)를 가지므로 color(색상), shape(형태), behavior(동작), object-specialty(특질)는 모두 part에 국한된 것으로 규정되고, characteristics는 visual-object(시각객체)의 하위 엘리먼트이므로 정의하고자 하는 한 객체의 특성으로 규정된다. 이를 통하여 각 객체의 내용정보에 있어서 상호연관성을 보존 할 수 있다. 또한 part 정보를 characteristics의 애트리뷰트로 선언함으로써 part 정보와 나머지 속성 정보간의 상속 관계를 명확하게 표현하였다. 그림 5는 그림 4에서 구조화한 DTD에 의하여 그림 3의 시각적 특징을 XML 문서로 작성한 것이다.

```
<visual-object>
  <object-name>가시고기</object-name>
  <characteristics part="몸통">
    <color>회록색</color>
  </characteristics>
  <characteristics part="등">
    <color>암녹색</color>
  </characteristics>
  <characteristics part="배">
    <color>은백색</color>
  </characteristics>
  . . . . .중간 생략
</visual-object>
```

그림 5 객체의 시각적 특징을 중심으로 한 그림 3에 대한 XML문서

##### 3.1.2 의미 중심 모델링

이미지의 특징을 결정하는 중요한 요소 중 하나는 이미지 전체가 내포하는 의미이다.

예3) 어떤 사용자가 특정 동물의 분류인 “조기강”에 속하는 이미지를 검색하고자 한다. 그림 5와 같이 이미지의 내용정보로써 객체의 시각 정보만을 이용한다면, 이미지가 나타내는 의미 정보인 “조기강”이라는 조건으로 원하는 이미지를 검색 해 낼 수

없다. 즉, 객체의 시각 정보만으로는 사용자가 원하는 이미지를 검색해 낼 수 없다.

예4) “야구 경기” 하는 장면을 담은 이미지는 이미지 안의 각 객체보다는 “야구 경기” 라는 이미지 전체가 내포하는 의미가 더 중요하며, 이러한 의미가 이미지의 종류와 범위를 결정하게 된다.

위의 예3),4)와 같이 이미지에 포함된 객체의 시각적인 특징만으로는 이미지에 대한 정보를 모두 표현할 수 없을 뿐 아니라, 사용자가 원하는 적절한 이미지를 검색해 낼 수 없다. 따라서 본 논문에서는 이미지가 내포하는 주제, 분류, 이미지 특징 등의 의미 정보를 객체의 시각적 특징 정보와 함께 표현할 수 있도록 의미를 중심으로 하여 XML DTD를 설계 하고자 한다. 그림 6은 의미를 중심으로 모델링한 XML DTD 구조를 보여준다.

```
<!ELEMENT semantics (subject, image-property*)>
<!ELEMENT subject (subject-name, class*)>
<!ELEMENT subject-name (#PCDATA)>
<!ELEMENT class (#PCDATA)>
<!ATTLIST class class-name CDATA #IMPLIED>
<!ELEMENT image-property (#PCDATA)>
```

그림 6 의미 정보 중심 DTD 구조

그림 6에서 나타낸 것처럼 이미지는 이미지 전체가 내포하는 의미 정보를 갖는다. subject(주제)와 image-property(이미지특징)는 모두 semantics(의미 정보)의 하위구조로써 표현되고, 이는 전체적인 이미지에 국한된 정보이다. subject의 하위 엘리먼트인 class(분류)정보는 class-name(분류명)을 애트리뷰트로 가짐으로써 이미지에 따라 분류 정보를 다양하게 확장 가능하도록 한다. 그림 7은 그림 6에서 구조화 한 DTD에 의하여 그림 3의 의미 정보를 XML 문서로 작성한 예이다.

```
<semantics>
  <subject>
    <subject-name>어류</subject-name>
    <class class-name="학명">Pungitius sinensis Guichenot
    </class>
    <class class-name="강">조기 강</class>
    <class class-name="목">큰가시고기 목</class>
    <class class-name="과">큰가시고기 과</class>
  </subject>
  <image-property>표본</image-property>
  <image-property>회귀야생동물</image-property>
</semantics>
```

그림 7 의미 정보를 중심으로 한 그림 3에 대한 XML 문서

### 3.1.3 이미지와 내용정보간의 매핑 정보 표현

XML 문서 구조로 표현된 이미지 내용정보만으로는 이 정보가 어떠한 이미지와 매핑 되어야 하는가를 나타낼 수 없다. 그림 8은 이미지와 내용정보간의 매핑 구조를 보여준다.

```
<!ELEMENT imageDB (image*)>
<!ELEMENT image (image-view?,visual-object*, semantics*)>
<!ELEMENT image-view EMPTY>
<!ATTLIST image-view src CDATA #REQUIRED width
  CDATA #IMPLIED height CDATA #IMPLIED alt CDATA
  #IMPLIED>
```

그림 8 이미지와 내용정보간의 매핑 구조

본 논문에서는 매핑정보를 표현하기 위하여 image-view 엘리먼트를 추가하였다. 이 엘리먼트의 애트리뷰트로써 src 속성을 추가하여 실제 이미지가 데이터베이스의 어느 부분에 저장되어 있는가를 표현하고, 이를 통하여 이미지 내용정보와 이미지간의 매핑정보를 XML 문서 구조 내에 표현 할 수 있다. 이 매핑 정보를 통하여 데이터베이스 내에 저장된 이미지를 실제 웹 브라우저에 검색결과로써 보여주게 된다.

## 4. 시스템 구현

본 장에서는 논문에서 제안한 이미지 내용정보간의 상호 연관성을 바탕으로 설계된 DTD 구조를 기반으로 하는 시스템 구조의 효율성을 검증하기 위하여 구현된 프로토타입 시스템에 대해서 기술한다. 이 시스템을 기반으로 웹 환경에서 XML 데이터 서버인 eXcelon에 저장된 이미지의 검색을 제공해 주는 예를 보인다.

### 4.1 시스템 구현 환경

본 논문에서 제안한 프로토타입 시스템의 구현 환경은 표 3과 같다.

내용정보의 상호 연관성을 기반으로 한 이미지 데이터 모델을 구현하기 위한 프로토타입 시스템은 서버 측에는 Windows NT 환경을 기반으로 XML 데이터 서버인 Object Design사의 eXcelon을 사용하였다. eXcelon은 XML 데이터를 위하여 개발된 데이터 서버로써 XML 데이터의 동적인 구조를 저장하고 관리하는데 효율적으로 고안되었다. 존재하는 XML데이터에 어떠한 변형 없이 그 자체로 저장이 가능하고, 꼬임 없는 확장이 가능하다. 이에 따라 구조화된 객체를 행과 열로 바꾸는 매핑 코드가 필요 없다. 이는 실행 도중 성능을 저해하는 조인이 필요 없음을 나타내며, 높은 성능의 향

표 3 시스템 구현 환경

서버 측 운영 체제	Windows NT Server 4.0
XML Data Server	Object Design사의 eXcelon 2.0
웹 서버	Internet Information Server 4 (IIS4)
웹 클라이언트	Microsoft Internet Explorer 5.0 이상
개발 도구 및 언어	XML Query Language (XQL) Extensible Style Sheet Language (XSL) Active Server Page (ASP) JavaScript Microsoft Visual InterDev

상을 보여준다. 또한 캐쉬 포워드(Cache-Forward) 구조는 다 계층(Multi-Tier) 응용 프로그램 개발을 지원한다[23, 24].

사용자로부터 웹 브라우저 상에서 이미지에 대한 검색어를 입력받아 이를 전송해 주는 웹 서버로는 Internet Information Server 4 (IIS 4)를 사용하였으며, 웹 클라이언트로는 XML 문서 구조를 지원하는 마이크로소프트의 익스플로러 5.0 이상을 지원한다. 사용자 질의는 XQL로 변환되고, 이러한 XQL 질의를 데이터 서버에 연결시키기 위하여 COM API로써 ASP 기술을 사용하였다. XQL 질의에 의하여 검색된 결과를 웹 브라우저 상에 보여 주기 위하여 XSL을 이용한 포매팅을 지원하였다[10, 22].

4.2 시스템 구조

본 논문에서 제안한 시스템의 전체 구조는 그림 9와 같다.

물리적으로는 사용자가 검색하고자 하는 질의 요구와 질의에 대한 결과를 제공하는 웹 클라이언트, 클라이언트의 요청을 받아들이는 웹 서버, 그리고 이미지와 내용 정보를 저장하고 있는 XML 서버의 세 계층으로 구성되어 있다. 웹 서버와 데이터베이스 서버의 연결은 COM 인터페이스를 통하여 가능하며, 이를 통하여 XQL 질의가 전달된다. 사용자가 이미지에 대한 요청을 시작하면, 웹 브라우저는 웹 서버에게 질의 요구를

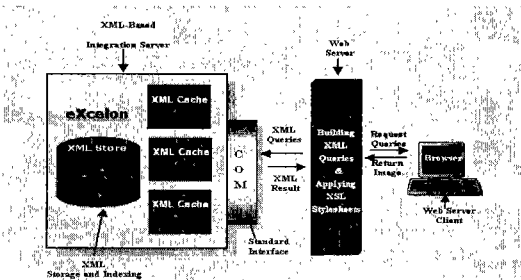


그림 9 시스템 구조

넘겨주고, 웹 서버에서는 이를 XQL 형태로 변환하여 질의어를 구성한다. 이러한 질의는 COM 인터페이스를 통하여 XML 서버인 eXcelon으로 전달된다.

검색되어진 XML 결과는 다시 COM 인터페이스를 통하여 웹 서버로 전달되고, 웹 서버에서는 XSL Style Sheets의 적용을 통하여 사용자가 원하는 형태로 검색 결과를 포맷하여 HTML 문서로 변환한다. 변환된 문서는 웹 브라우저에 의하여 사용자에게 보여짐으로써 원하는 이미지가 검색된다.

4.3 eXcelon XML 데이터 서버

논문에서는 이미지와 이미지의 내용정보를 저장하기 위하여 eXcelon XML 데이터 서버를 사용하였다. 본 절에서는 기존의 다른 RDBMS의 사용과 비교되는 eXcelon 사용의 특징을 설명하도록 하겠다.

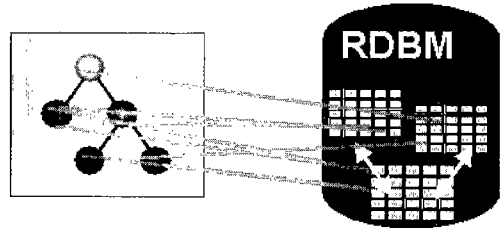


그림 10 RDBMS의 XML 데이터 매핑 방식

XML로 구조화된 이미지 내용정보들을 관계형 데이터베이스에 저장 할 경우 데이터 구조에 대한 상세한 매핑(mapping) 정보가 요구된다. 그림 10은 관계형 데이터베이스에 저장 될 때의 XML 데이터 매핑 방식을

```

<InsuranceClaim>
  <ClaimID> 10023</ClaimID>
  <LossCategory>7</LossCategory>
  <Settlements>
    <Payment>
      <Payee>홍길동</Payee>
      <Date>12-OCT-1998</Date>
      <Amount>200000</Amount>
      <Approver>박문수</Approver>
    </Payment>
  </Settlements>
  <DamageReport>
    아주 큰 <Cause>화재</Cause>
    가 빌딩을 휩쓸어
    <Casualties>12</Casualties>
    명이 죽었다. 경찰 조사에 따르면
    <Motive>방화범</Motive>에 의한 것으로 보고 있다.
  </DamageReport>
</InsuranceClaim>
    
```

그림 11 Oracle 8i의 XML 파일 저장 방식

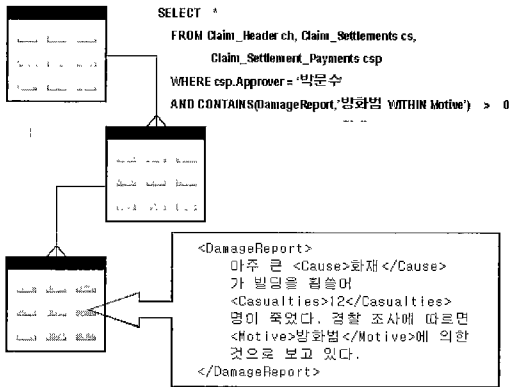


그림 12 Oracle 8i에서의 XML 구조 검색을 위한 질의 방식

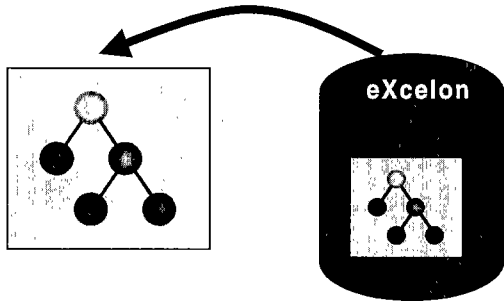


그림 13 eXcelon의 XML 데이터 저장 방식

나타내며[21], 그림 11은 Oracle 8i의 XML 파일 타입 저장 방식을 보여준다[24]. 이처럼 관계형 데이터베이스 사용 시 이미지 내용정보는 여러 테이블에 분할되어 저장되고, 이때 복합 질의를 사용한다면 그림 12와 같이 저장 테이블간의 조인(join)연산이 필요하다. 따라서 조인으로 인한 연산 비용은 증가하게 된다. 그러나 eXcelon을 이용하여 객체를 중심으로 저장할 경우 그림 13과 같이 구조 정보가 그대로 저장된다[21].

그림 14는 그림 12의 조인 연산을 XQL 질의로 변환한 것으로 조인 없이 이미지의 검색이 가능함을 볼 수 있다[21]

또한 XML 데이터 서버를 지원하는 질의어인 XQL에 의하여 XML 정보 구조로의 다양한 접근 지점을 가지

```

/InsuranceClaim[Settlements/Payment/Approver='박문수'
$and$
DamageReport[Motive='방화범' $and$ Cause='화재' $and$
... ]]
    
```

그림 14 XQL을 이용한 질의 예

며[10], 이러한 속성을 이용하여 사용자는 원하는 범위 내에서 원하는 내용을 마음대로 추출해 낼 수 있다. 따라서 조인 없는 복합 질의와 검색 문맥의 선택은 검색 비용의 절감 효과를 가져온다.

4.4 eXcelon에서의 COM Client API

본 논문에서는 eXcelon과 통신하기 위한 방법으로 Active Server Page(ASP) 기술을 사용하여 COM 클라이언트 API를 구현하였다. ASP내에서 eXcelon 서버와 연동하여 어플리케이션(Application)을 작성하기 위해서는 크게 다음과 같은 3단계를 거친다.

- 1 단계 : 세션(Session) 생성
- 2 단계 : Query
- 3 단계 : XSL의 적용

가. 세션 생성

클라이언트 어플리케이션을 위하여 ASP 내에서 eXcelon에 저장된 데이터를 조작하기 위해서는 세션을 획득하여야 한다.

```

Function getSession()
on error resume next
dim sess, fact
set sess = Application("eXcelon")
if (typename(sess) <> "IXInSession") then
session이 생성 되어 있지 않다면
set fact = Server.CreateObject("ODLeXcelon")
'Clinet API를 사용하기 위하여 반드시
처음 생성 해 주어야 하는 Client Object.
set sess = fact.GetSession("",0)
'Session 획득. 첫 번째 변수는 캐쉬를 나타냄 empty인 경우는
default 캐쉬를 나타냄.
set Application("eXcelon") = sess
end if
set getSession = sess
end function
    
```

그림 15 eXcelon의 세션 생성 과정

세션은 XML Storage인 XMLStores로의 클라이언트 접근을 제공하기 위하여 사용된다. 세션을 형성하기 위한 과정은 그림 15와 같다.

나. Query

ASP 내에서의 질의를 전달하기 위해서는 IXInSession.Query 메소드(method)를 호출해야 한다.

```

dim sess, datares
set sess = getSession()
set datares = sess.Query( bio/bio_eng.xml , QueryString)
'eXcelon의 XMLStore중 bio의 bio_eng.xml 이라는 XML 문서에
질의를 요청함.
'QueryString 은 이전에 생성된 XQL query 문자열을 의미.
    
```

그림 16 Query 전달 방식



```
Dim styleses
set styleses = sess.GetFileData(bio/bio_view.xml)
'eXcelon의 XMLStore중 bio에 저장된 XSL 문서인
bio_view.xml을 세션 안으로 불러들임'
```

그림 17 XSL의 적용

그림 16은 Query 전달 방법을 나타낸 것으로 그림 15에서 생성된 세션을 넘겨받은 후, IXinSession을 초기화하고, sess.Query를 호출한다. 이를 통하여 XQL 질의는 eXcelon 서버에 전달된다.

다. XSL의 적용

XQL 질의를 통하여 검색된 XML 문서 구조를 HTML 형태로 변환하여 사용자에게 보여주기 위해서는 XSL StyleSheet을 적용해야한다. XSL의 적용 예는 그림 17과 같다.

4.5 XSL을 이용한 검색 결과의 변환

XQL 질의를 통하여 검색된 결과는 XML 문서 구조 자체이다. 따라서 이를 사용자가 원하는 형태로 웹 브라우저 상에 보여주기 위해서는 이미지와 이미지 내용을 HTML 문서 형태로 변환하여야 한다. XSL은 XML 문서 구조를 HTML 형태로의 변환 기능을 제공

하므로, 본 논문에서는 bio\_view.xml 과 bio\_detail.xml 을 사용하여 두 가지 변환을 기능을 제공한다.

bio\_view.xml은 기본적으로 사용되는 XSL Syle Sheets이다. 사용자의 질의 요청이 시작되면, 이에 따른 검색이 이루어지고, 넘겨받은 XML 형태의 검색 결과는 bio\_view.xml을 통하여 웹 브라우저 상에 변환되어 보여 진다. 이때 이미지에 포함된 주요 객체와 이미지 종류를 이미지와 같이 사용자에게 보여준다. 그림 17은 bio\_view.xml의 소스 코드이다. bio\_view.xml은 "조기 강이고 보호야생동물을 포함한 이미지를 검색하여야"라는 XQL 질의에 대하여 검색되어진 XML 문서 구조를 그림 18과 같이 HTML 형태로 변환하여 사용자에게 결과를 보여 준다.

bio\_detail.xml은 bio\_view.xml에 의하여 보여진 내용 외에 더 자세한 이미지 정보를 사용자에게 보여주기 위하여 사용되는 것으로 자세히 보기 옵션이 눌러 졌을 때 사용된다.

4.6 구현된 시스템을 이용한 예제 프로그램

본 논문에서는 이미지 내용정보의 상호 연관성에 따라 모델링 되어진 XML 데이터베이스를 바탕으로 다양한 질의어를 이용하여 검색되어지는 예제를 보이기 위

```
<?xml version="1.0" encoding="UTF-8"?>
<DIV xmlns:xsl="http://www.w3.org/TR/W3C-xsl" STYLE="padding:.3in .1in .3in .3in;">
  <table STYLE="font:10pt. Verdana" border="0" cellpadding="2" cellspacing="3">
    <xsl:for-each select="//image">
      <tr><td align="center">
        <img>
          <xsl:attribute name="src">../Scripts/xlnisapi.dll/bio/images/
          <xsl:value-of select="image-view/@src"/>
          </xsl:attribute>
        </img>
      </td><td>
        <table cellpadding="4" cellspacing="5" bgcolor="#D9E8F9"
          align="center">
          <tr><td bgcolor="#234D86">
            <font color="white"> 주요 객체 </font>
          </td><td>
            <xsl:value-of select="visual-object/object-name"/>
          </td>
          </tr><tr><td bgcolor="#234D86">
            <font color="white"> 이미지 종류 </font>
          </td><td>
            <xsl:value-of select="semantics/subject/subject-name"/>
          </td>
          </tr><tr><td colspan="2">
            <a><xsl:attribute name="href">bio_details.asp?image_number=<xsl:value-of
              select="image-number"/>
            </xsl:attribute> 자세히 보기 </a>
          </td></tr>
        </table></td></tr>
      </xsl:for-each>
    </table>
  </DIV>
```

그림 18 bio\_view.xml

하여, 생물 데이터베이스를 예제로 사용하였다.

어류, 조류, 양서류, 파충류, 포유류의 이미지를 바탕으로 각 이미지가 포함된 객체의 시각적인 특징과 이미지 전체의 의미를 중심으로 내용정보를 기술하였다. 생물 이미지는 특히 이미지 내용정보간의 연관성을 잘 나타낼 수 있으며, 의미 정보 또한 트리 구조에 맞추어 기술하기 적절하다.

데이터베이스로는 XML 데이터 서버인 eXcelon을 이용하고, 사용자 인터페이스를 위하여 COM API를 사용하였다.

사용자가 질의어를 선택하고 검색 버튼을 누르면, 이는 웹 서버로 전달되어 XQL 질의어로 변환되고, 이는 COM API를 통하여 eXcelon으로 전달된다. 실행된 결과는 사용자의 웹 브라우저로 전달되어진다. 이때 검색된 이미지들과 이미지 내용정보는 XSL StyleSheets에 의하여 사용자가 원하는 형태로 포맷되어 간단한 이미지 내용정보와 이미지가 보여진다. 이때 자세한 정보를 원하는 사용자를 위하여 자세히 보기 메뉴를 추가하여 이를 선택한 사용자에게는 선택한 이미지와 자세한 이미지 정보를 웹 브라우저 상에 보여준다.

사용자 질의는 이미지에 대하여 표현된 정보와 가장 근접한 내용을 찾기 위하여 사용되는 것으로써 사용자가 이미지 데이터베이스를 통하여 사용할 수 있는 정보는 앞에서 제시한 이미지 내용정보의 상호 연관성에 따라 다음과 같이 크게 3가지 질의어로 분류하여 검색을 지원하도록 한다.

- 객체의 시각적 특징을 중심으로 한 단순 질의어
- 의미 정보에 대한 단순 질의어
- 복합 질의어

단순 질의어는 선택된 조건에 따라 전체 이미지 내용 정보 중 원하는 부분만 골라 검색 할 수 있도록 설계되었다. 선택된 구조 내에 포함된 모든 엘리먼트들 중에서, 검색하고자 하는 엘리먼트에 대한 값과 사용자 검색어가 정확히 일치하는 이미지를 검색한다.

**질의 예 1: 객체의 시각적 특징에 대한 단순 질의어**  
 “회록색 몸통을 가진 금강모치를 포함한 이미지를 검색하여라”

**변환되어진 XQL 질의:**

**Q1://image [visual-object [object-name='금강모치' \$and\$ characteristics [@part='몸통' \$and\$**

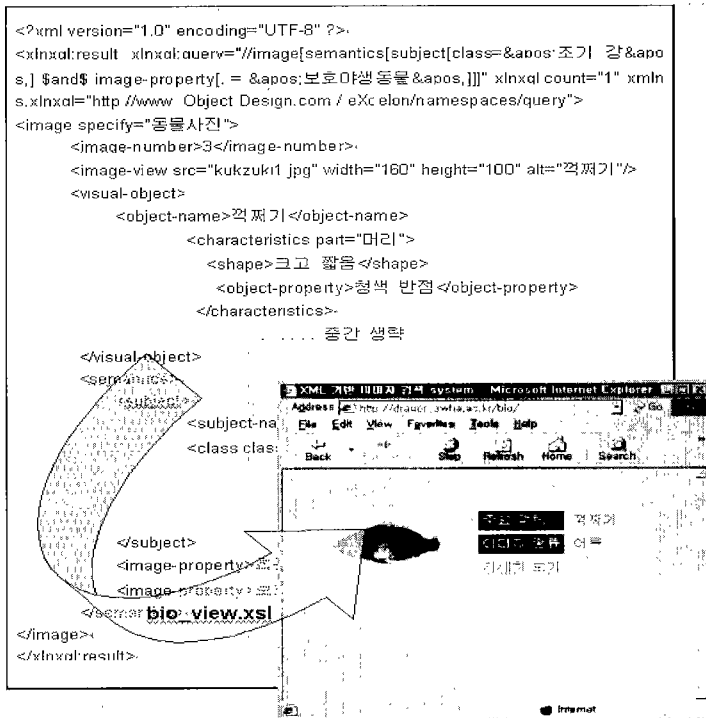


그림 19 XSL 적용 결과

*color='회록색']]]*

질의 Q1은 객체의 시각적인 특징에 대한 단순 질의로써 객체의 시각적인 특징에 해당하는 구조만을 검색하여 사용자에게 결과를 돌려준다. XML의 구조적인 특성을 이용하여 XQL 질의는 검색 영역을 필터링(filtering)하고, 이를 통하여 검색 구간은 객체 명이 “금강모치”인 구간으로 제한된다. 이 구간 안에서 “몸통 색상이 회록색”인 이미지를 검색하게 된다. 그림 20은 객체의 시각적 특징을 중심으로 한 단순 질의어 사용자 입력 화면을 보여준다.

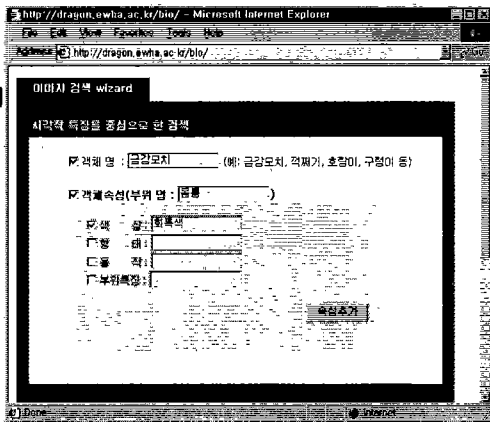


그림 20 객체의 시각적 특징을 중심으로 한 단순질의어 사용자 입력화면

**질의 예 2 : 의미 정보에 대한 단순 질의어**

“조기 강이면서 희귀야생동물인 이미지를 검색하여라”

**변환되어진 XQL 질의**

**Q2: //image[semantics[subject[class='조기 강'] \$and\$ image-property[ = '희귀야생동물']]]**

질의 Q2은 의미 정보에 대한 단순 질의로써 의미 정보에 해당하는 구조만을 검색하여 사용자에게 결과를 돌려준다. 이 질의는 의미 정보를 포함하는 엘리먼트들 중에서 분류명이 “강”인 엘리먼트의 값이 “조기 강”이고 이미지특징이 “희귀야생동물”인 이미지를 검색해 준다. 그림 21은 의미 중심의 단순 질의어 사용자 입력 화면을 보여준다

복합 질의어는 객체의 시각적 특징 구조와 의미 중심 구조 각각에 대하여 하나 이상의 엘리먼트와 사용자가 검색어가 정확히 일치하는 이미지를 검색한다.

**질의 예 3 : 복합 질의어**

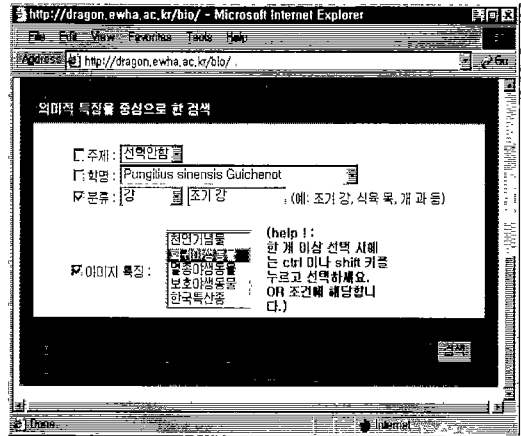


그림 21 의미 중심의 단순 질의어 사용자 입력 화면

“회록색 몸통을 가진 희귀야생동물인 어류를 포함한 이미지를 검색하여라”

**변환되어진 XQL 질의**

**Q3: //image[visual-object[characteristics[@part='몸통' \$and\$ color='회록색'] \$and\$ semantics[subject[subject-name='어류'] \$and\$ image-property[ = '희귀야생동물']]]**

질의 Q3은 객체의 시각적 특징 구조와 의미 중심 구조를 모두 검색하여 사용자에게 검색 결과를 돌려준다. 사용자는 이 질의를 통하여 객체의 시각적 특징 정보와 의미 정보를 동시에 검색 조건에 사용할 수 있게 된다. 그림 22는 복합 질의어 사용자 입력 화면을 보여 준다. 질의 예 2(“조기 강이면서 희귀야생동물인 이미지를

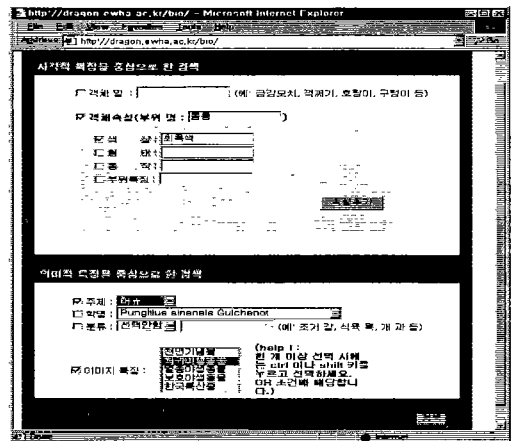


그림 22 복합 질의어 사용자 입력 화면

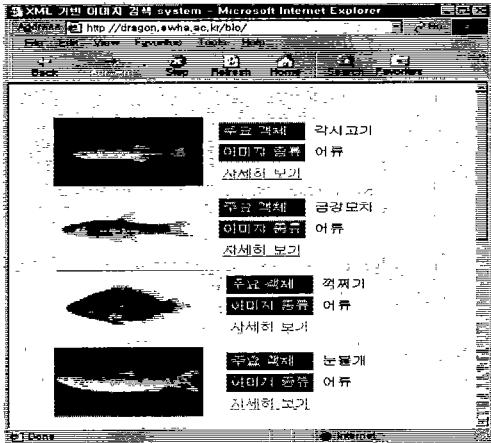


그림 23 질의 예2)의 검색 결과 화면

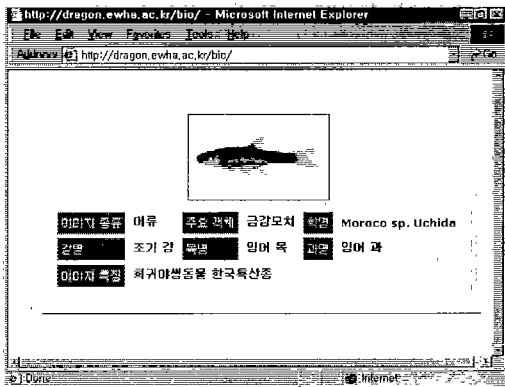


그림 24 자세히 보기 선택 예

검색하여야" )에 대한 실행 결과는 그림 23과 같다. 검색 결과는 사용자가 요청한 이미지와 이미지에 대한 간단한 설명 정보이다. 이미지에 대한 더 자세한 정보를 원하는 사용자를 위하여 자세히 보기 메뉴 기능을 추가하였다. 그림 24는 자세히 보기를 선택한 예이다.

## 5. 결론 및 향후과제

이미지 내용정보가 갖는 특징 중 하나는 내용정보간에 상호 연관성을 갖는다는 점이다. 따라서 이를 표현해 주기 위하여 XML을 이용하여 내용정보를 구조화하고, 구조화된 정보의 특징을 가장 잘 반영할 수 있는 데이터베이스에 저장하여야 한다. 편리한 사용자 인터페이스를 통하여 생성된 효율적인 질의어는 데이터베이스에서 원하는 이미지를 검색해 낼 수 있도록 한다.

본 논문에서는 상호 연관성을 지닌 구조로 이미지 내

용정보를 구조화 하기 위하여 내용정보의 특성에 따라서 객체의 시각적 특징 중심 모델링과 의미중심 모델링으로 나누었다. 또한 모델링한 내용정보를 XML 데이터 서버인 eXcelon에 저장하고, 저장된 이미지를 검색하기 위한 XQL 질의어 생성을 제공한다. 질의어를 통하여 검색된 결과는 XSL을 통하여 웹브라우저 상에서 인식할 수 있는 HTML 형태로 변형되어 사용자에게 보여진다.

본 논문에서 제안한 내용 기반 이미지 검색 방식은 기존의 다른 연구들에 비하여 다음과 같은 특징을 갖는다.

- XML을 이용한 이미지 내용정보의 구조화를 통하여 내용정보들의 상호 연관성을 보존하고 이를 검색에 이용할 수 있다.
- 구조화 된 XML 문서를 쪼갬 없이 그대로 XML 데이터 서버인 eXcelon에 저장함으로써 조인 없는 검색이 가능하며, 이를 통한 연산 비용을 줄일 수 있다.
- XML은 웹 상에서 구조화된 문서를 전송 가능하도록 설계된 표준화된 텍스트 형식이며[16], 다양한 형태의 필터링(filtering)과 조합을 제공한다. 따라서 여러 장소에서 생성된 내용기반 이미지 데이터베이스의 교환과 통합이 용이하다.
- 검색된 이미지를 결과로써 제공할 뿐 아니라, 데이터베이스에 저장된 이미지 내용정보를 설명 정보로써 재사용 할 수 있다. 이를 통하여 사용자는 이미지에 대한 개략적인 내용정보를 확인할 수 있다.
- XSL의 지원을 통하여 사용자가 원하는 형태의 결과를 출력할 수 있다.

향후 연구 과제로는 본 연구에서 제안한 XML 구조 사용 기법을 좀 더 다양한 이미지 분야로 확장하고, 유사성 기반 검색을 위한 XML 구조화 모델을 제시함으로써 좀더 다양한 내용 기반 검색을 제공하는 것이다.

## 참고 문헌

- [1] Y. Alp Aslandogan, Chuck Their, Clement T. Yu, Jon Zou and Naphtali Raishe, "Using Semantic Contents and WordNetTM in Image Retrieval," *SIGIR 97 Philadelphia PA, USA*, pp.286-295.
- [2] 방난효, 엄기현, "내용 기반 검색을 위한 이미지 데이터베이스 모델", 1997년도 한국 정보과학회 봄 학술발표 논문집, vol.24, No.1, pp.11-14.
- [3] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, Peter Yanker,

- "Query by Image and Video Content: The QBIC System," *IEEE computer*, Vol.28, No.9, pp.23-32, September 1995.
- [4] Virginia E. Ogle and Michael Stonebraker, "Chabot: Retrieval from a Relational Database of Images," *IEEE Computer*, Vol.28, No.9, pp.40-48, September 1995.
- [5] Rohini K. Srihari, "Automatic Indexing and Content-Based Retrieval of Captioned Images," *IEEE Computer*, Vol.28, No.9, September 1995.
- [6] John R. Smith, Shin-Fu Chang, "VisualSEEK: a fully automated content-based image query system," *ACM Multimedia 96*, 1996.
- [7] Akan F. Smeaton and Ian Quigley, "Experiments on Using Semantic Distances Between Words in Image Caption Retrieval," In Proceedings of ACM SIGIR Conference, pp.174-180, 1996.
- [8] A.pentland, R.W. Picard, S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," *SPIE Storage and Retrieval Image and Video Databases II*, Vol.2 185, pp.34-47, 1994.
- [9] Guang-Ho Cha, Chin-Wan Chung, "Object-Oriented Retrieval Mechanism for Semistructured Image Collection," *ACM Multimedia '98 Bristol*, UK, pp.323-331.
- [10] Object Design, Inc. eXcelon User Guide. 1999.
- [11] John Robie, Joe Lapp, David Schach, <http://www.w3.org/TandS/QL/QL98/pp/xql.html>, XML Query Language (XQL).
- [12] <http://www.w3.org/TR/1998/REC-xml-19980210>, Extensible Markup Language (XML) 1.0, REC-xml-19980210.
- [13] H. V. Jagadish, "A retrieval technique for similar shapes," *Proceedings of the 1991 ACM SIGMOD international conference on Management of data*, pp.208-217, 1991.
- [14] S. Smoliar, and H. Zhang, "Content-Based Video Indexing and Retrieval," *IEEE Multimedia Magazine*, Vol.1, No.2, pp.62-72, 1994.
- [15] joo-Hwee Lim, "Learning Visual Keywords for Content-Based Retrieval," *Proceedings of the IEEE Multimedia Systems '99*, Vol.2, pp.169-173.
- [16] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, "Extensible Markup Language(XML) 1.0", <http://www.w3.org/TR/REC-xml>, REC-xml-19980210, 1998.
- [17] J.R. Smith, Shin-Fu Chang, "Tools and techniques for color image retrieval," In *Symposium on Electronic Imaging: Science and Technology Storage & Retrieval for Image and Video Databases IV*, Vol.2670, San Jose, CA, February 1996.
- [18] Ratan, A.L. and Grimson, W.E.L, "Training templates for scene classification using a few examples," *IEEE Workshop on Content-Based Analysis of Images and Video Libraries*, pp.90-97, 1997.
- [19] Frank Boumphrey, Olivia Drenzo, Jon Duckett, Joe Graf, Dave Hollander, Paul Houle, Trevor Jenkins, Peter Jones, Adrian Kingsley-Hughes, Kathy Kingsley-Hughes, Craig McQueen and Stephen Mohr, "XML Applications," Wrox Press, 1998.
- [20] William J. Pardi, "XML in Action Web Technology," Microsoft Press, 1999.
- [21] Vittorio Viarengo, "eXcelon XML Data Server Technical Overview," Object Exchange 98, Object Design User Conference, 1998.
- [22] <http://www.w3.org/TR/1999/WD-xsl-19990421/>, Extensible Stylesheet Language (XSL) Specification, W3C Working Draft 21 Apr 1999.
- [23] <http://www.23.co.kr/News/xmlom.html>.
- [24] iDevelop' 99, ORACLE, 1999.
- [25] <http://www.texcel.no/whitepapers/xql-design.html>.



박 선 영

1998년 이화여자대학교 전자계산학과 학사, 2000년 이화여대 과학기술대학원 컴퓨터학과 공학석사. 2000년 ~ 현재 LG 전자기술원 정보기술연구소 연구원.

용 환 승

정보과학회논문지 : 데이터베이스 제 27 권 제 1 호 참조