JDBC를 이용한 웹기반 사용자 질의 영상 검색

차상환*. 이상열*,황병곤*
*대구대학교 정보통신공학부

Web based User Query Image Retrieval using JDBC

Sang-hawn Cha*, Sang-Youl Lee*, Byung-Kon Hwang*
*School of Information & communication Engineering, Daegu University

요 약

본 논문에서는 웹 에이전트를 이용하여 웹상에서 멀티미디어 정보를 검색하는 것으로 HTML문서에 나타나는 텍스트 중 영상 이름이나 링크에 붙어 있는 텍스트를 추출하여 멀티미디어 자료를 JDBC를 이용하여 데이터베이스화하였다. 이 데이터베이스에 저장된 영상 자료는 웹 브라우저에서 질의자의 스케치에 의한 검색과 그리고 예제 영상 질의로 검색하는 방법을 제시하여 질의 효율성을 개선하였다.

1. 서론

컴퓨터 기술과 통신망 기술의 발전으로 인하여 정지화상(Image), 음성 자료(Audio data), 동화상 자료(Video data) 등의 대용량의 멀티미디어 자료가 대량으로 발생되고 있다. 이에 따라 전자 신문, 홈 쇼핑 등과 같은 새로운 미디어 정보 서비스가 인터넷을 통하여 확산되고 있다. 특히 이러한 멀티미디어 데이터 중 영상 데이터의 급속한 팽창으로 인해 이들의 효율적으로 관리하고 활용할 수 있는 기술이 요구되고 있다. 초기 영상 데이터 검색은 기존의 데이터베이스 검색과 동일한 방식으로 관리하기 위한 키워드 방식이었으나 점차 영상 데이터베이스의 크기가 증가함에 따라 키워드의 중복 및 사용자가 기억할 수 없게 되는 등 이 방식의 한계를 극복하기 위한 영상시각적 내용을 파악하여 검색하는 내용기반 방식이 검색에 이용하게 되었다.

본 논문은 웹 서버에서 웹 문서상의 영상을 설명하는 텍스트를 자동으로 추출하고, 추출된 텍스트를 이용하여 자동으로 분류하여 JDBC를 이용하여 데이터베이스에 저장한다. 영상 설명 텍스트뿐만 아니라 파일명에 기술된 확장자를 이용하여 검색한다. 한편 일차적으로 웹 에이전트로 검색된 영상은 질의 예제 영상의 색상 히스토그램 특징과 그리고 웹 브라우저 상에서 사용자가 직접 주어진 캔버스에 그린 영상의 색상 및 모양 특징으로 유사한 영상을 검색하는 방법을 제안한다. 논문의 구성으로는 2장에서는 관련연구로서 웹 영상의 기존의 색인방법에 대하여 설명하고, 3장에서는 시스템구조에 대하여 설명한다. 4장에서 실험 결과에 대하여 설명하고 마지막으로 5장에서 본 논문의 결론과 개선점, 향후 연구 과제를 제시한다.

2. 관련 연구

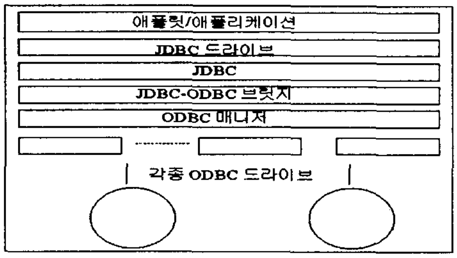
웹상의 영상을 검색하는 시스템이 최근 다양하게 연구되어 왔으며 Yahoo의 Image sufer는 키워드로 분류 트리(Category tree)를 생성하고 이를 이용하여 웹에서 영상을 검색한다. 이 시스템에서 주제 인덱스는 웹의 하이퍼 텍스트 구조를 탐색하지 않고 주제에 따라 인덱스를 분류하였다[1,2]. Lycos의 미디어 검색 툴은 영상 URL과 영상이 포함된 웹 문서에서 키워드를 자동으로 추출하였으나 CERN의 libwww 라이브러리를 사용하였다[3]. WebSeek 시스템은 웹 영상 내용을 분석하여 영상 헤더, 파일 종류, 크기, 날짜 등과 같은 영상에 연결된 텍스트 정보를 검색 키워드로 사용한다. 또한 이 시스템은 사람의 얼굴이나 수평선 등의 객체를 영상에서 자동으로 인식하고, 색상과 질감 등의 정보를 사용하여 하나의 영상을 여러 조각으로 분할한다[4]. ImageRover는 영상 수집, 해석을 담당하는 영상 수집 시스템과, 질의 서버와 사용자 인터페이스로 이루어진 영상 질의 서버시스템으로 이루어진다[5]. WISE는 영상 설명 텍스트와 영상 색상 히스토그램을 이용하여 영상 검색을 지원하는 시스템이다.[6] 그리고 WISE는 웹상의 영상을 검색하기

위한 시스템을 연구하였어나 영상의 색상 히스토그램 비교로 구한되었다.[7]

3. JDBC 기반 이미지 검색 시스템

1) 데이터 베이스 연결 종류

웹상에서 클라이언트와 서버간에 데이터베이스로 연결하는 것은 여러 가지 방법들이 있다. SQL문을 HTML에 내장시켜서 사용하는 확장 HTML 방법, 데이터베이스에 저장된 테이블과 필드들에 관한 정보를 외부 템플릿 파일에 저장한 다음, 사용자가 요청하면 이 정보를 이용하여 동적 SQL 질의를 생성하는 템플릿 파일 방법, 일련의 SQL 명령어들이 데이터 베이스에 저장한 다음 필요할 때 외부 프로그램에서 호출할 수 있는 데이터베이스 저장 프로시저 기능을 이용하는 데이터 베이스 저장 프로시저 방법, 동일한 데이터배이스 응용 프로그램으로 다양한 데이터베이스를 접근할 수 있도록 산업계 표준 인터페이스인 ODBC (Open DataBase Connectivity)를 이용하는 표준 인터페이스 방법, 특정 데이터 베이스엔진이 제공하는 고유의 API를 이용하는 고유 API 방법, 자바 서블릿(servlets) 접근 방법 등이 있다. 클라이언트와 웹서버간에 HTTP 메시지 처리과정에서 하나의 요구(request)와 하나의 응답(response)이 서로 독립적으로 수행하는 CGI는 사용자가 HTML FORM 필드의 내용을 전송할 때마다 매번 CGI 프로그램을 다시 실행하게 된다. 즉, CGI를 이용한 데이터 베이스와의 연동은 단 방향성 프로토콜이기 때문에 사용자가 질의할 때마다 매번 데이터베이스와 연결하고 해제하는 프로세스가 수행하게 된다. 그러나, JAVA 기반의 연동방법은 서블릿(servlets)을 이용하는 방법이 사용하므로 CGI 접근 방법에서 발생하는 문제를 해결할 수 있다. 서블릿은 서브쪽에서 실행되는 애플릿(applet)으로 자바 서버 API의 서브셋이다. 서블릿은 CGI와 유사하게 HTML 폼(form)이나 클라이언트 애플릿을 통해서 PUT/POST 메소드로 사용자 질의를 전달 받는다. 질의를 전달 받은 서블릿은 JDBC를 통해서 데이터베이스 서버에 연결하고, 실행결과를 반환 받아서 HTML 문서로 포맷팅 한 후 클라인트로 전송한다. <그림 1>은 이러한 연동구조를 나타난 것이다. 서블릿은 자바 멀티 쓰레드를 이용해 실행되기 때문에 CGI 방식처럼 다수의 프로세스 생성으로 인한 성능 저하 현상이 발생하지 않는다. 서블릿을 이용한 자바기반 접근방법은 상태 유지가 가능하고 JDBC를 통한 데이터 베이스 연결이 쉽다는 장점을 가진다. JDBC는 Sun Microsystem 사에서 개발한 SQL 데이터 베이스 접근 인터페이스로서, JDBC API는 데이터 베이스 연결, SQL 명령문, 질의 결과 집합, 메타 데이터 등을 표현 할 수 있다.



<그림 1> JDBC 연동구조

2) JDBC 접속

JDK(JAVA Development Kit) 에는 JDBC를 위한 java.sql.* 클래스들이 있다. JDBC 2.0 표준 외에 근래에 몇 가지 기능이 추가된 Standard Extension API를 담고 있는 javax.sql.* 클래스들이 있는데 이 클래스는 JDBC API의 인터페이스 정의에 대한 것만 담고 있을 뿐, 개별 데이터베이스와는 무관하다. 따라서, Java가 제공하는 JDBC API를 사용하기 위해서는 개별 데이터베이스에 맞게 구현된 드라이버가 있어야만 한다.

Connection 객체는 데이터베이스와의 연결(connection)을 담당한다. 연결 부분은 실행될 SQL문, 그리고 연결을 통해서 반환되어지는 결과들을 포함한다. 어플리케이션은 한 개의 데이터베이스와 하나 이상의 연결을 할 수 있거나 또는 많은 서로 다른 데이터베이스에 연결할 수도 있다. 다음은 JDBC를 이용하여 데이터베이스를 연결하는 프로그램이다.

```
import java.sql.*;
import java.lang.*;
public class Test {
  public static void main(String[] args) {
      // 드라이버 로딩
    try {
      Class.forName("myDriver.ClassName");
      } catch (ClassNotFoundException e) {
        System.err.println("Class Not Found : " +
e.getMessage());
         }
      // 데이터베이스 접속
       try {
        String url = "jdbc:myprotocol:mydatabase";
        Connection db =
          DriverManager.getConnection(url, "myid",
              "mypassword");
        } catch (SQLException e) {
         System.err.println("SQL Error : " +
              e.getMessage());
        }
    }
}
```

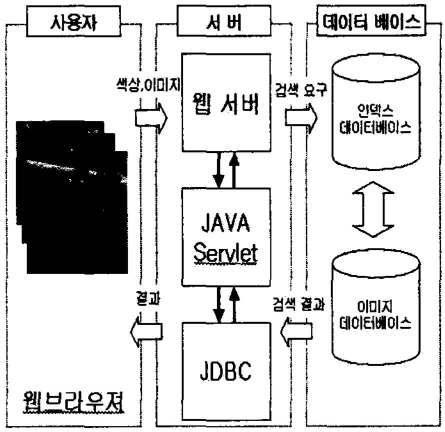
<그림2> JDBC 데이터베이스 연결

4. 내용기반 영상 검색 시스템

만일 사용자가 웹 상에 존재하는 많은 이미지 중 자신이 원하는 이미지를 보고 싶다고 하자. 사용자가 원하는 이미지에 대해 적절히 설명하면 검색 엔진이 이와 가까운 대상부터 보여준다. 사용자는 이중에서 자신의 의도에 맞는 것을 고를 수 있다. 이러한 검색 엔진이 사용자의 질의 요청이 들어온 다음부터 웹사이트들을 하나하나 검색할 수는 없으므로, 사용자의 질의 요청이 없을 때에도 미리 웹사이트에 있는 문서를 돌아다니면서 이미지와 관련 있는 텍스트와 이미지의 색상 히스토그램을 가져온다. 사용자가 원하는 이미지를 설명하는 방법은 이미지 설명 텍스트와, 이미지 특성 중 색상 히스토그램과 모양의 복잡도를 사용한다. 질의 방법은 다음과 같다.

첫째, 검색 엔진에서 제공하는 색상 히스토그램 표를 질의자가 선택하여 검색할 수 있도록 했다.

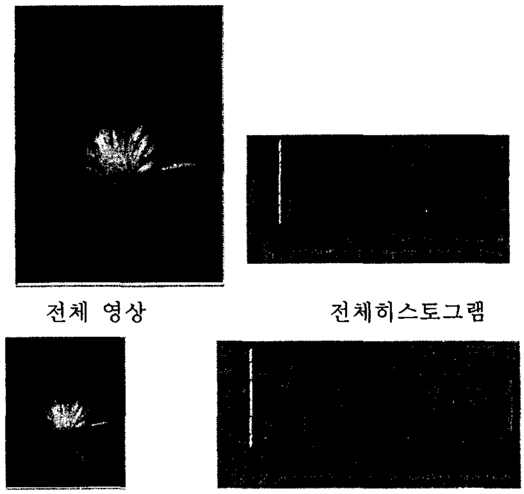
둘째, 사용자가 그림 판을 이용하여 스케치하여 검색하고자하는 특정 이미지를 입력한다. 입력받은 검색 엔진은 데이터베이스에 저장되어 있는 이미지의 색상 히스토그램과 모양의 복잡도를 비교하여 사용자가 입력한 이미지와 유사한 이미지를 출력한다. 이미지 특성만 이용하거나 이미지 설명 텍스트 또는 이미지의 이름을 입력하여 검색할 수 있다. <그림3>은 JDBC를 이용한 이미지 검색 시스템의 전체 구조를 나타내고 있다.



<그림3> 자바 기반의 JDBC 연동구조

◆ 객체 정보 인덱싱

일반적으로 검색 대상의 영상은 대부분 사진 영상으로 검색 객체가 영상의 중앙에 위치하고 있다. 이러한 경우 전체 영상의 히스토그램만으로 유사 영상을 검색하는 경우 질의 영상의 배경 부분의 색상과 데이터베이스내의 유사 영상의 배경 부분의 색상의 차가 클 경우, 찾고자 하는 영상을 검색할 수 없다. 이러한 이유로 본 논문에서는 영상의 중앙에 적절한 크기의 중앙 객체 영역을 설정하여 따로 색상 특징을 추출한다.



<그림 4> 영역별 히스토그램

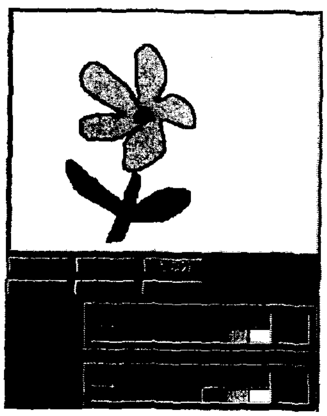
영상 자체를 두 개의 영역으로 분할함으로써 색상 히스토그램 인터섹션의 단점인 지역적 특징을 보안하기 위해서이다. 각각의 픽셀을 읽어 들여 색상 히스토그램을 만드는 과정에서, 중앙 영역은 미리 정의된 일정 크기의 영역 부분만을 가지고 색상

히스토그램을 만드며, 배경 영역은 전체 영상에서 중앙 영역을 제외한 영역의 색상 히스토그램을 얻는다.

여기에서 중앙 영역과 배경 영역의 구분은 입력 영상의 크기를 N × M으로 가정할 경우, 중앙 영역은 각 N, M 의 $\frac{1}{4}$에서 $\frac{3}{4}$의 지점까지를 길이로 정한다.

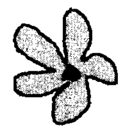
◆ 사용자 스케치에 의한 방법

본 논문에서 질의 인터페이스는 사용자가 목표 영상을 위한 질의 영상을 작성할 때 물체의 윤곽선을 그리고 내부의 색을 지정하며 이러한 물체의 배열을 통해 상대적인 위치관계를 표시한다. 물론 사용자가 그리는 윤곽선은 검색하고자 하는 목표 영상의 객체의 윤곽선과 정확하게 일치하지 않아도 된다. 물체의 중심선간에 상대적인 위치만으로도 유효한 정보가 될 수 있다. 검색 특징으로 각 물체의 색상, 물체의 윤곽선에 의한 모양정보로서의 복잡도 및 물체의 중심점들이 된다. 이들 값의 유사도 계산에 의해 질의 스케치 영상에 의한 데이터베이스 영상에서 유사한 영상들을 검색한다. 그림5는 질의자가 질의할 내용을 스케치한 예이며 그림6은 질의된 그림을 각각의 객체로 추출된 그림이다. 추출된 객체는 2개며 각각 색상, 복잡도 및 무게 중심으로서 벡터값이 데이터베이스에 저장된다.



<그림 5> 사용자 질의 영상

색상 : 3 무게중심 : 159 451

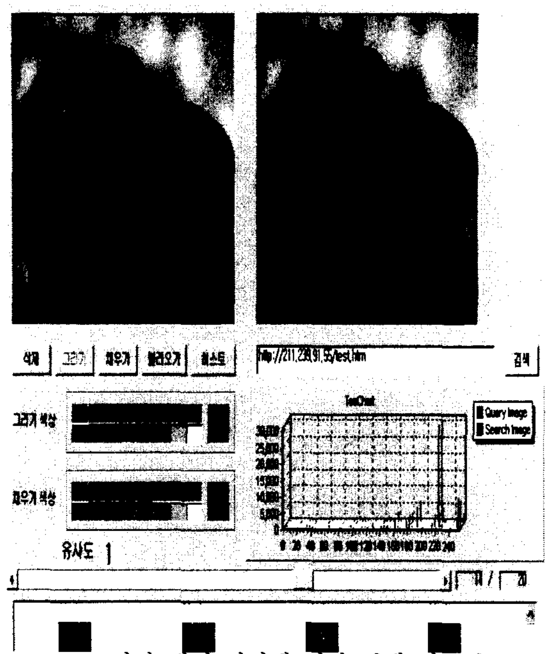


색상 : 15 무게중심 : 168 245

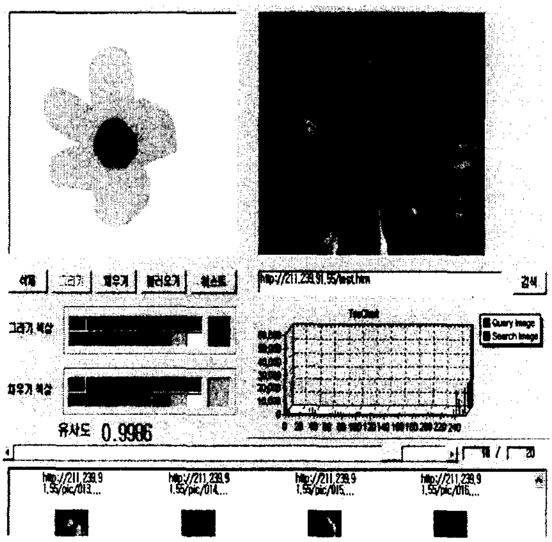
<그림 6> 추출된 객체

5. 실험결과

본 논문에서 제안된 시스템의 구현 환경은 한글 윈도우즈 98을 운영체제로 하는 펜티움 600MHz 상에서 실험하였다. 실험 영상은 다양한 색과 영역의 부분 또는 전체를 차지하여 영역별 변화가 많은 꽃 영상을 사용하였다. 꽃 영상의 경우 내부영역을 차지하는 경우 전체 영역으로 퍼진 경우 등 다양한 데이터를 볼 수 있다. 사용자는 왼쪽 상단의 질의 화면을 이용하여 검색하고자 하는 그림을 그려서 질의 하는 방법과 질의 예제로 준비된 영상을 이용하여 질의 하는 방법모두를 처리할 수 있다. 검색된 그림 중 가장 유사도가 높은 것은 오른쪽에 큰 그림을 한개 보여주고 아래쪽 화면에는 다음 유사도가 높은 것부터 낮은 것 순으로 4개씩 작은 그림으로 나타내었다. 아울러 화면 밑에 작은 그림을 클릭하면 오른쪽 위부분에 있는 큰 화면으로 그림이 나타나도록 했다. 한편 질의자가 선택한 그림을 다운 받을 때 카운트를 하여 가장 많은 카운트를 받은 영상 자료를 화면에 먼저 나타나도록 하였다.



A. 질의 예제 영상에 의한 검색 시스템



B. 질의자가 스케치에 의한 검색새스템
<그림 7> 멀티미디어 검색 시스템

6. 결론

현재 영상 검색 시스템들은 대부분이 텍스트에 의한 검색만을 지원하고 있는 실정이다. 이는 사용자들이 그 동안 텍스트를 이용한 검색 방법에 익숙해져 있기 때문으로 볼 수 있지만 아직 내용 기반에 의한 검색 기술이 실용적으로 쓰일 만큼 높은 성능을 보이고 있지 않기 때문이다. 본 논문은 웹상의 영상을 텍스트웹 기반 에이전트의해 영상을 수집한 후, 질의 예제에 의한 영상 검색 및 사용자가 질의할 내용을 직접 스케치하여 영상을 검색하는 기법을 제안하였다. 그러나 대량의 영상 정보를 저장하게 되는 경우 많은 자료를 신속하게 검색하기 위한 인덱스 설정문제와 데이터베이스의 저장 공간 문제와 더불어 더 많은 영상 특징 연구가 필요하다.

참고문헌

[1]야후의 영상 검색. http://ipix.yahoo.com

[2]J.R. Sm. S.F. Chang, "An Image and Video Engine for the Word-Wide Web", Symposium on Electronic Image: Science and Technology Storage & Retrieval for Image and Video Database V. San Jose, CA, February 1997.

[3] Linda Bertland, "Searching the Internet: Subject Indexes and Search Engines," http://www.voicenet.com/~bertland/search.html.

[4]Charles Frankel, Michael J. Swain, Vassilis, "Webseer: An Image Search Engine for the World-Wide Web", TR 96-14, U. Chicago, 1996.

[5]Stan Sclaroff, Leniod Tayche, Macro La Cascia, "ImageRover: A Content-Base Image Browser for the World Wide Web", Proc. IEEE Workshop on Content-base Access of image and Video Labraries, 1997.

[6] 박명선,이석호."지능형 웹 영상 검색 엔진의 설계", 한국정보과학회 가을 학술 발표논문집, Vol.26, No.2, 1999.

[7]박명선, "WISE: Design and Implementation of WWW Image Search Engine,". 서울대학교 석사논문, 1997.