

# JDBC를 이용한 분산 환경에서의 이미지 검색 웹 에이전트

차상환, 황병곤  
대구대학교 컴퓨터IT공학부

## JDBC based Distributed Image search Web Agent

Sang-Hwan Cha, Byung-Kon Hwang  
Dept. of Computer & IT, Daegu University

### 요 약

본 논문은 웹상에 존재하는 이미지를 멀티 스레드에 의한 분산 아키텍처를 이용하여 수집 및 검색 시스템으로, 웹문서에 나타나는 텍스트 중 이미지의 이름이나 확장자 그리고 링크에 붙어 있는 텍스트를 추출하여 이미지 자료를 JDBC를 이용하여 데이터베이스화하였다. 이 데이터베이스에 저장된 이미지 자료는 웹 브라우저에서 질의자의 스케치에 의한 검색과 그리고 예제 영상 질의로 검색하는 방법을 제시하여 질의 효율성을 개선하였다. 또한, 멀티 스레드를 이용한 분산 아키텍처를 이용하여, 데이터베이스화 하는 시간에 효율을 개선하였다.

### 1. 서 론

인터넷망과 서버의 규모가 증가함에 따라 사용자들은 텍스트, 이미지, 비디오 등 광범위한 인터넷 정보 속에서 원하는 정보를 정확하고 신속하게 검색하기는 대단히 어려운 일이다. 인터넷망의 성장속도에 비례해서 정보의 양이 늘어나고, 원하는 정보를 검색하기 위한 시간 역시 늘어나게 되었다<sup>[1,2,8,10]</sup>. 이로 인해 텍스트 정보와 함께 이미지 정보에 대한 효과적인 저장, 검색 방법에 관한 연구가 필요하게 되었다<sup>[3,4,5,8]</sup>.

본 논문에서는 멀티 스레드에 의한 분산 아키텍처를 이용한 에이전트를 이용하여 웹

상에서 텍스트 정보를 효과적으로 검색한다. 웹 검색 에이전트가 웹상의 텍스트 정보를 검색하기 위해 URL 목록을 요청/응답 처리할 때는 CPU와 메모리의 사용량이 늘어난다. 이를 멀티 스레드에 의한 분산 처리 시스템으로 효율성을 높일 수 있다. 텍스트 정보는 이미지의 이름이나 확장자 그리고 이미지를 설명하는 텍스트를 자동으로 추출한다. 추출한 텍스트는 멀티미디어 수집기를 통해 JDBC를 이용하여 데이터베이스화 하였다.

이미지를 설명하는 텍스트는 텍스트를 생성하는 과정에서 개인의 주관차이로 인한 오차가 생길 수 있다. 또한, 사람이 직접 작성해야 하는 번거로움으로 인해 생략될 수 있는 문제점을 가지고 있다. 이 같은 문제

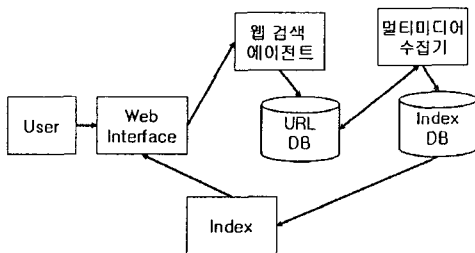
를 해결하기 위해 이미지 자체의 정보를 이용해 유사성을 비교하는 내용기반 이미지 검색 방법이 연구 되었다<sup>[7,9,12]</sup>. 일차적으로 멀티 스톱에 의한 분산 아키텍처를 이용한 웹 에이전트로 검색된 영상은 질의 예제 영상의 색상 히스토그램 특징과 그리고 웹 브라우저 상에서 사용자가 직접 주어진 캔버스에 그린 영상의 색상 및 모양 특징으로 유사한 영상을 검색하는 방법을 제안한다.

## 2. 관련연구

### 2.1 웹 에이전트

에이전트란 원하는 정보를 얻기 위하여 웹상의 문서들을 검색하고, 참조되는 문서들을 재귀적으로 검색하면서 웹의 하이퍼텍스트 구조를 추적하여 정보를 저장해 주는 프로그램으로 자치능력(Autonomous Ability)을 이용하여 사용자를 대신해서 어떤 일을 수행해 주는 컴퓨터 프로그램으로 정의할 수 있다<sup>[6,11]</sup>.

그림 1은 웹 에이전트(Web Agent)의 구성을 나타낸다. 사용자를 대신해 웹 공간을 순회하여 탐색결과를 알려주는 웹 검색 에이전트(Web Search Agent)와 멀티미디어 수집기로 분류된다. 웹 검색 에이전트는 웹 문서에 포함되어 있는 이미지 및 사운드 등의 멀티미디어 파일의 경로를 찾아내고 멀티미디어 수집기는 찾아낸 파일의 확장자를 이용해 멀티미디어를 분류해 데이터베이스에 저장한다.



<그림 1> 웹 에이전트 구성

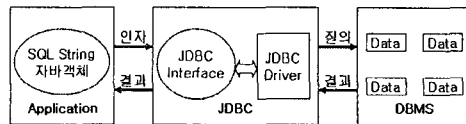
### 2.2 웹 에이전트 동작 알고리즘

문서 탐색 에이전트는 주어진 웹 사이트를 방문하여 이미지 자료를 수집하는 역할을 한다<sup>[9,11,12]</sup>.

- (1) 초기 사용자가 지정한 웹 사이트를 탐색한다.
- (2) URL의 호스트 이름에 따라 robots.txt에 접근한다.
- (3) robots.txt 파일의 내용을 받아 분석하여 그 호스트에 접근가능한 URL을 추출한다.
- (4) 로봇의 접근을 배제하는 사이트가 아니면 방문하여 연결된 문서를 수집한다.
- (5) 수집된 문서를 분석하여 URL을 추출하여 URL저장소에 저장한다.
- (6) 수집한 문서에서 필요한 키워드와 정보를 추출하고 저장한다.
- (7) 연결된 문서에서 사용자가 지정한 카운트를 벗어나지 않는 범위에서 (2) 혹은 (3)부터 반복하여 탐색을 한다.

### 2.3 JDBC의 구조

JDBC는 자바 인터페이스를 이용하여 데이터베이스 연동을 수행하고, 이들 인터페이스들은 각각 서로 다른 개별적인 제공업체들에 의해 구현되어 있다.

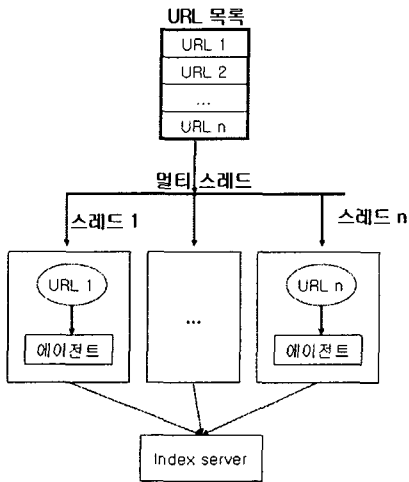


<그림 2> JDBC의 구조

그림 2에서와 같이, JDBC는 SQL 문을 DBMS에 전달하고 결과물을 리턴해 주는 역할을 한다.

## 3. 분산 환경에서의 웹 검색 에이전트

### 3.1 멀티 스톱 아키텍처



<그림 3> 웹 검색 에이전트 아키텍처

그림 3은 웹 검색 에이전트 아키텍처로 웹 검색 에이전트가 웹페이지를 다운받기 위해 URL 서버에 URL 목록을 요청(request)하고 URL 서버는 URL 목록을 에이전트에 응답(response)하는 시간동안은 CPU는 멈춰있다. 이로 인해 CPU 사이클이 낭비되게 된다. 그러나 병행 스레드들의 실행을 멀티 처리하는 멀티 스레드 시스템으로 낭비되는 CPU 사이클을 줄일 수 있다. 즉, 하나의 스레드가 URL 목록을 요청해 기다리는 동안 다른 스레드는 다운로드 받은 URL 목록을 처리할 수 있다. 멀티 스레드는 요청과 응답에 독립적으로 통신하는 환경으로 CPU, 메모리 등의 시스템 자원을 능률적으로 사용하게 한다.

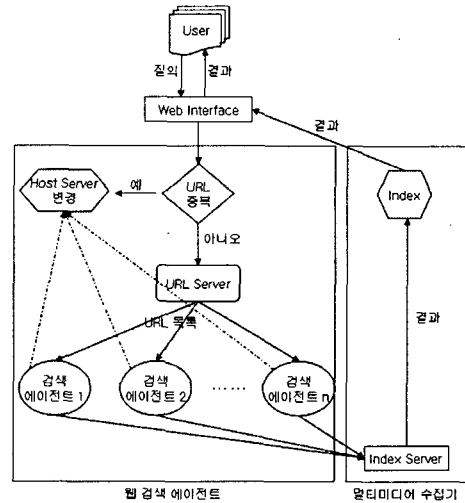
### 3.2 분산 아키텍처

분산 아키텍처를 사용함으로써 PC의 연산 속도를 증가시키고 광범위한 인터넷 정보 속에서 원하는 정보를 신속하게 검색할 수 있다.

본 논문에서 처리하는 URL 리스트는 각각의 에이전트로 보내지고 에이전트는 리스트를 동일한 작업으로 처리하는 구조이다. 즉, URL 리스트는 각각 다른 검색 에이전트의 CPU에서 동일하게 처리되어지며 그

결과를 Index 서버로 보내지게 된다. 따라서 URL목록을 처리하는 시간을 줄일 수 있다.

분산 환경에서의 멀티스레드를 이용한 웹 검색 에이전트의 아키텍처는 그림 4와 같다.



<그림 4> 분산 검색 에이전트

- 서버를 통해 사용자가 질의를 보내면 웹 인터페이스에 의해 지정한 웹사이트 URL의 중복여부를 확인한다.
- 중복되지 않은 URL를 URL 서버에 저장한다.
- URL 서버는 URL 서버에 저장된 목록을 n개의 검색 에이전트에 각각 분배한다.
- 검색 에이전트들은 멀티스레드로 구성되어 있고, 분배 받은 목록들을 검색하고 해당 웹 페이지를 다운받는다.
- 다운받은 웹 페이지의 내용을 분석하여 텍스트나 이미지등 멀티미디어 콘텐츠를 인덱스서버에 저장한다.
- 인덱스 서버는 저장된 콘텐츠의 내용 중 검색에 필요한 정보만을 인덱스로 보낸다.
- 사용자 질의에 대한 응답은 인덱스에서 제공한다.

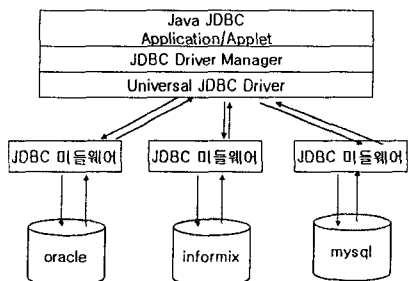
## 4. JDBC 기반 데이터베이스

### 4.1 JDBC 연결

JDBC인터페이스의 DriverManager라는 클래스를 이용하여 각종 드라이버를 자바용 프로그램에 연결시켜 준다. JDBC 드라이버는 DBMS를 만든 업체나 다른 연구단체들에서 만들어지는 DBMS에 의존하기 때문에 매우 다양하다. SUN에서는 JDBC 드라이버를 크게 JDBC-ODBC Bridge, 네이티브 API, 네트워크 연결 드라이버, DBMS 프로토콜로 분류하였다.

본 논문에서는 네트워크 연결 드라이버를 이용하여 JDBC를 구현하였다.

네트워크 연결 드라이버는 순수 자바로 만들어 졌으며, 그림 5는 JDBC 메소드 호출을 DBMS에 독립적인 네트워크 프로토콜로 바꾸어 주는 드라이버이다. 바뀌진 네트워크 프로토콜 호출은 서버에 설치된 미들웨어로 전달되고 이 미들웨어가 서버에 존재하는 DBMS와 통신을 하는 것이다.



<그림 5> 네트워크 연결 드라이버

### 4.2 JDBC 접속

JDK(JAVA Development Kit)에는 JDBC를 위한 java.sql.\* 패키지가 있다. JDBC 2.0 표준 외에 근래에 몇 가지 기능이 추가된 Standard Extension API를 담고 있는 javax.sql.\* 패키지가 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.

```

package imagesagent;
import java.sql.*;

public class DBConnection{
    // 변수를 초기화 한다.
    private Connection conn = null;
    private Statement stmt = null;
    private static DBConnection _Instance = null;

    // DBConnection을 외부에서 생성할 수 없게 하기 위해 private으로 정의한다.
    private DBConnection(){
        try{

Class.forName("org.gjt.mm.mysql.Driver");
            conn = DriverManager.getConnection
("jdbc:mysql://localhost/agent12","root","1111");
            stmt = conn.createStatement();
        }catch(ClassNotFoundException e){
            System.out.println("JDBC드라이버를 찾을 수 없습니다.");
        }catch(SQLException e){
            System.out.println("DataBase에 연결할 수 없습니다.");
        }
    }

    // DBConnection의 인스턴스를 받아오는 함수
    // Connection을 하나만 생성하여 여러곳에서 공유할 수 있다.
    public static DBConnection getInstance(){
        if(_Instance == null) _Instance = new
DBConnection();
        return _Instance;
    }
}
    
```

<그림 6> JDBC 데이터베이스 연결

Class.forName()을 사용하면 프로그램 내부에서 원하는 드라이버를 직접적으로 로드할 수 있다.

class.forName("org.gjt.mm.mysql.Driver");  
- 그림6의 14라인은 - MySQL 드라이버를 로드한다. 이 경우 자바 런타임 시스템은 CLASSPATH에 지정된 환경 변수를 이용하여 애플릿 클래스 로더나 클래스 로더를 사용하여 해당 드라이버를 로드한다.

Connection interface는 데이터베이스와의 연결(connection)을 담당한다. 연결 부분은 실행될 SQL문, 그리고 연결을 통해서 반환되어지는 결과들을 포함한다. 어플리케이션은 한 개의 데이터베이스와 하나 이상

의 연결을 할 수 있거나 또는 많은 서로 다른 데이터베이스에 연결할 수도 있다.

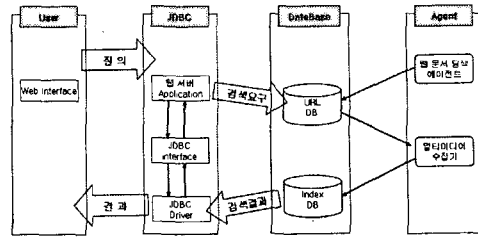
### 5. 내용기반 이미지 검색 시스템

내용기반 이미지 검색 시스템이란 색상, 형태 및 질감 등의 저수준 특징정보를 이용하여 이미지 데이터베이스를 구축하고, 이미지에 대한 검색요구가 발생했을 때 사용자가 찾고자 하는 이미지와 유사한 이미지를 제공하는 시스템으로 정의 된다.

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 이 중에서 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.

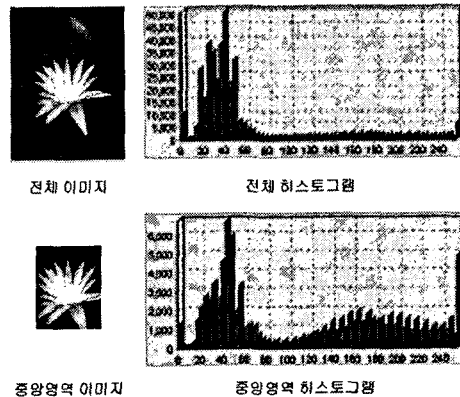
둘째, 사용자가 그림판을 이용하여 스케치하여 검색하고자하는 특정 이미지를 입력한다. 입력받은 검색 엔진은 데이터베이스에 저장되어 있는 이미지의 색상 히스토그램과 모양의 복잡도를 비교하여 사용자가 입력한 이미지와 유사한 이미지를 출력한다. 이미지 특성만 이용하거나 이미지 설명 텍스트 또는 이미지의 이름을 입력하여 검색할 수 있다. 그림 7는 JDBC를 이용한 이미지 검색 시스템의 전체 구조를 나타내고 있다.



<그림 7> 시스템 연동구조

#### 5.1 객체 정보 인덱싱

일반적으로 검색 대상의 이미지는 대부분 사진 이미지로 검색 객체가 이미지의 중앙에 위치하고 있다. 이러한 경우 전체 이미지의 히스토그램만으로 유사 이미지를 검색하는 경우 질의 이미지의 배경 부분의 색상과 데이터베이스내의 유사 이미지의 배경 부분의 색상의 차가 클 경우, 찾고자 하는 이미지를 검색할 수 없다. 이러한 이유로 본 논문에서는 그림 8에서와 같이 이미지의 중앙에 적절한 크기의 중앙 객체 영역을 설정하여 따로 색상 특징을 추출한다.



<그림 8> 영역별 히스토그램

이미지 자체를 두 개의 영역으로 분할함으로써 색상 히스토그램 인덱싱의 단점인 지역적 특징을 보완하기 위해서이다. 각각의 픽셀을 읽어 들여 색상 히스토그램을 만드는 과정에서, 중앙 영역은 미리 정의된

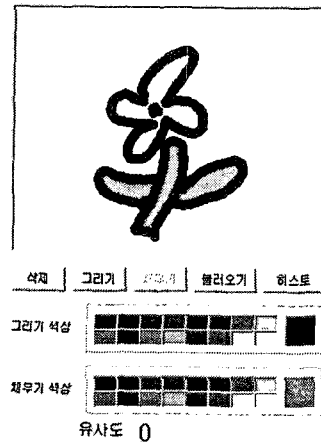
일정 크기의 영역 부분만을 가지고 색상 히스토그램을 만들며, 배경 영역은 전체 이미지에서 중앙 영역을 제외한 영역의 색상 히스토그램을 얻는다.

여기에서 중앙 영역과 배경 영역의 구분은 입력 이미지의 크기를  $N \times M$ 으로 가정할 경우, 중앙 영역은 각  $N, M$ 의  $1/4$ 에서  $3/4$ 의 지점까지를 길이로 정한다.

### 5.2 사용자 스케치에 의한 질의

사용자 스케치에 의한 질의 방식은 사용자가 직접 그리기 도구를 이용하여 목표 이미지와 유사한 특성을 갖는 질의 이미지를 스케치하여 가장 유사한 이미지를 찾는 방법이다. 사용자가 원하는 이미지의 특성을 자유롭게 표현할 수 있는 장점이 있다. 다만, 사용자는 질의 이미지에 대한 사전정보 부족 등으로 스케치 과정에서의 작성 오류로 인하여 목표 이미지와의 많은 차이를 보일 수 있다.

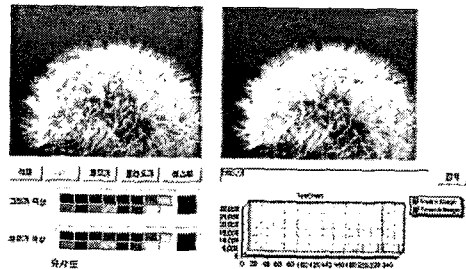
질의 인터페이스는 사용자가 목표 이미지를 위한 질의 이미지를 작성할 때 물체의 윤곽선을 그리고 내부의 색을 지정하며 이러한 물체의 배열을 통해 상대적인 위치관계를 표시한다. 물론 사용자가 그리는 윤곽선은 검색하고자 하는 목표 이미지의 객체의 윤곽선과 정확하게 일치하지 않아도 된다. 물체의 중심선간에 상대적인 위치만으로도 유효한 정보가 될 수 있다. 검색 특징으로 각 물체의 색상, 물체의 윤곽선에 의한 모양정보로서의 복잡도 및 물체의 중심점들이 된다. 이들 값의 유사도 계산에 의해 질의 스케치 이미지에 의한 데이터베이스 이미지에서 유사한 이미지들을 검색한다. 그림 9은 질의자가 질의할 내용을 스케치한 예이다. 추출된 객체는 2개며 각각 색상, 복잡도 및 무게 중심으로서 벡터값이 데이터베이스에 저장된다.



<그림 9> 사용자 질의 이미지

### 5.3 예제 이미지에 의한 질의

예제 이미지에 의한 질의 방식은 사용자가 질의 화면에 예시로 주어지는 이미지를 표시하면 시스템이 표시된 이미지와 가장 유사한 이미지를 선택하는 방법이다. 예제로 주어지는 이미지들은 인덱스 데이터베이스에 저장되어 있는 이미지를 불러오는 방식으로 사용자가 질의 이미지를 선택하면 별도의 특성 추출과정을 거치지 않으므로 사용자 및 시스템의 입장에서는 효율적이다. 다만, 예제이미지만을 사용하기 때문에 선택한 이미지는 원하는 이미지의 구체적인 특성을 사용할 수 없다. 따라서 사용자가 원하는 이미지와 예제이미지와 차이가 클 경우 한번에 원하는 결과를 얻기 힘들고 계속 다른 예제이미지를 제공받아야 한다.



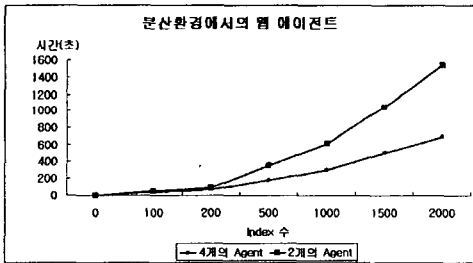
<그림 10> 예제 이미지에 의한 질의

## 6. 결과

본 논문에서 제안된 시스템의 구현 환경은 Windows 2000 Server를 운영체제로 하는 P4 1.6MHz 상의 PC로 실험하였다. 또한, 분산 환경에서의 이미지 검색 웹 에이전트의 성능평가를 위해 로봇의 문서 수집시 발생하는 부가적인 웹서버와 네트워크 부하는 무시하였고, 웹 에이전트 실행을 위한 PC의 성능은 동일하게 하였다.

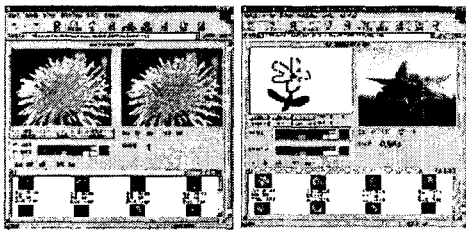
실험 이미지는 다양한 색과 영역의 부분 또는 전체를 차지하여 영역별 변화가 많은 꽃 이미지를 사용하였다. 꽃 이미지의 경우 내부영역을 차지하는 경우 전체 영역으로 퍼진 경우 등 다양한 데이터를 볼 수 있다.

그림 11은 검색 에이전트 아키텍처를 2개와 4대의 PC로 분산처리를 했을 때 Index 저장 시간을 나타내었다.



<그림 11> 분산 환경에서의 웹 에이전트

분산 환경에서의 멀티 스레드를 이용한 검색 에이전트는 PC의 CPU사용률을 높여 URL 목록 정보를 빠르게 검색하여 Index Server로 보낸다.



a. 꽃의 예제 영상에 의한 검색 시스템      b. 꽃의 스케치에 의한 검색 시스템

<그림 12> 이미지 검색시스템

사용자는 왼쪽 상단의 질의 화면을 이용하여 검색하고자 하는 그림을 그려서 질의 하는 방법과 질의 예제로 준비된 이미지를 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 이미지 자료를 화면에 먼저 나타나도록 하였다.

## 7. 결론

현재 이미지 검색 시스템들은 대부분이 텍스트에 의한 검색만을 지원하고 있는 실정이다. 이는 사용자들이 그동안 텍스트를 이용한 검색 방법에 익숙해져 있기 때문에 볼 수 있지만 아직 내용 기반에 의한 검색 기술이 실용적으로 쓰일 만큼 높은 성능을 보이고 있지 않기 때문이다. 본 논문은 웹상의 이미지를 멀티 스레드를 이용한 분산 아키텍처를 이용해 이미지를 수집한 후, 질의 예제에 의한 이미지 검색 및 사용자가 질의할 내용을 직접 스케치하여 이미지를 검색하는 기법을 제안하였다.

문서 수집 시 발생할 수 있는 일부 서버가 다운되거나 네트워크 상태가 갑자기 좋지 않게 된 경우의 로봇 동작에 대한 연구 및 몇 개의 에이전트를 가동 했을 때 Index 저장에 가장 효율적인지에 대한 연구가 요구된다. 또한, 대량의 이미지 정보를 저장하게 되는 경우 많은 데이터를 신속하게 검색하기 위한 인덱스 설정문제와 데이터베이스의 저장 공간 문제와 더불어 더 많은 이미지 특정 연구가 필요하다.

## 8. 참고문헌

- [1] Y.Alp Aslandogan and Clement T. Yu. Multiple Evidence Combination in Image Retrieval: Diogenes Searches for People on Web. In Proceedings of ACMSIGIR 2000, Athens, Greece, July 2000.
- [2] Content-Based Image Retrieval Systems: A survey. Remco C. Veltkamp, Mirela Tanase, Department of Computing Science, Utrecht University.
- [3] Searching for Images and Videos on the World-Wide Web. John R. Smith and Shih-Fu Chang.
- [4] Henry A. Rowley, Shumeet Baluja, and Takek Kanade. Neural Network Based Face Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Jan 1998.
- [5] Performance evaluation of a Distributed Architecture for Information Retrieval, Brandon Cahoon, Kathryn S. McKinley, Dept of CS, UMass, Amherst.
- [6] Scalable Distributed Architecture for Information Retrieval, Zhihong Lu, University of Massachusetts, Amherst.
- [7] 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.
- [8] Linda Bertland, "Searching the Internet: Subject Indexes and Search Engines.
- [9] Charles Frankel, Michael J. Swain, Vassilis, "Webseer: An Image Search Engine for the World-Wide Web", TR 96-14, U. Chicago, 1996.
- [10] 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.
- [11] 박명선, 이석호, "지능형 웹 영상 검색 엔진의 설계", 한국정보과학회 가을 학술 발표논문집, Vol.26, No.2, 1999.
- [12] 박명선, "WISE: Design and Implementation of WWW Image Search Engine,". 서울대학교 석사논문, 1997.