

# 지도도서관을 위한 GIS자료의 공간검색과 자료전송에 관하여

김창제\*, 장은미\*\*

Chang-Je Kim\*, Eun-Mi Chang\*\*

## 초 록

위성영상의 종류가 많아지고 다양한 종류의 수치지도가 생성되고 관리되기 위해서는 지도도서관이 필요할 것이며, 이는 클리어링하우스와는 달리 전통적인 종이지도까지 포함이 되어야 한다. 상품화된 GIS 툴에서 제공하는 공간검색의 기능을 개발하여 지도도서관에 활용을 한다면, 지명검색보다 다양한 형태의 자료를 검색하여 비교할 수 있으므로, 데이터베이스에 각 자료의 좌상단, 우하단 좌표가 입력 관리되고, 서버에 모든자료가 존재한다는 가정하에 사각검색과 원 검색의 원리를 경우에 따라 분류하고 SQL문으로 만드는 원리를 정리하였다. 공간검색을 통한 GIS자료의 다운로드와 생성된 자료의 업로드 기능을 클라이언트/서버환경에서 JDBC를 활용하여 구현하였다. 국가기관이나 학교에서 지도전문 도서관이 전문한 실정이지만, 앞으로의 가능성을 전제로 하여 고지도로부터 관광지도, 수치지도, 위성영상을 활용한 주제도에 대한 관리방법으로 제안한다.

## 키 워 드

지도도서관, 공간검색, JDBC, 다운로드, 업로드

## 1. 서 론

국가기관에서 생산한 지도와 분석 처리된 영상들과 대학에서 연구를 위하여 축적된 역사지도에 이르는 다양한 지도를 이미지형태나 파일형태로 한 장소에 수집하여 지도도서관을 구성할 수 있다. Library Congress 에서 정한 분류체계와 번호(지역코드 4자리, 주제관련8 자리, 축척관련 6자리)를 준수한 call number대로 분류를 하는 것과는 달리 수치지도의 clearing house 를 위한 메타데이터의 구성은 별도로 진행이 되는 것이 현실이다. 다양한 형태의 지도와 자료수요에 대처하기 위하여 자료의 가공과 생산과정보다는 자료를 수집하여 검색할 수 있는 경우, 다양한 검색방법이 소개되어 질 수 있다. 하나의 인공위성영상 자료를 가지고 목적에 따라 여러 형태의 주제도를 생성하게 될 때, 생성되어진 자료에 대한 메타데이터의 관리가 필수적이며, 이에 따라 자료의 업로드와 다운로드가 되도록 하는 것 또한 필요하다.

지도도서관이 설립되는 경우에 각 지도에 대한 메타데이터의 관리가 주제와 지역에 대한 정보 관리 외에 절대적인 위치좌표를 첨가함으로써 다양한 형태의 지도자료와 GIS 자료, 영상지도를 동시에 비교 검토할 수 있게 된다. 또한 클라이언트가 요구하는 자료를 상용소프트웨어의 설치 없이 유사한 원리의 공간검색이 가능하고 일정한 조건을 만족하면 다운로드 받을 수

\* 삼성 SDS GIS 사업팀 책임

\*\* 삼성 SDS GIS 사업팀 전임

있도록 하는 논리적 구성은 차후에 수치지도 도서판이나 clearinghouse를 구축하고자 할 때에 활용될 수 있다. 특히 국가의 수치지도는 축척에 따른 인덱스를 가지고 정리가 되어 있으므로 문제가 없지만 앞으로 위성자료에서 생성된 부분영상과 다양한 주제도의 생성을 전제로 한다면, 자료의 검색과정에서 사용자의 검색의 용이성과 메타데이터 관리의 용이성을 동시에 고려해야 한다.

축척이 다양한 자료를 동시에 다루게 될 때, 각각의 지도가 어느 지역을 포함하고 있는가에 대한 같은 깊이의 지역명 관리가 불가능하다. 한 영상이 포함하는 지역이 다수일 경우도 그러하며, 몇 퍼센트를 포함해야 그 지역을 포함한다고 할 것인가도 결정해야 한다. 원하는 부분이 정확히 어느 지역에 얼마큼 되는가에 따라 지역명만 넣은 질의에 한계가 있다. 지역명 관리를 할 경우 행정동리를 쓸 것인가 법정동리를 쓸 것인가 결정부터 어느 수준의 행정구역을 사용할 것인가 등의 고려사항을 다 만족하는 일관된 자료의 정리가 어렵다. 위성영상의 헤더파일에서 볼 수 있는 좌상단, 우하단 좌표의 값으로 관리되는 것이 지역명을 대체해서 사용되어 질 수 있다. 이 때의 좌표는 경위도좌표 또는 TM 좌표가 될 수도 있다. 시스템 내에서 변환 프로그램에 의해 일관된 데이터의 정리가 가능하도록 하면 무리가 없이 공간검색을 할 수 있다.

기존의 GIS자료 파일은 GIS자료처리시스템과 같은 호스트에 존재하는 형태로 다른 시스템에 GIS자료 파일에 직접적인 접근을 위해서는 FTP와 같은 파일 전송 프로그램을 이용하여 사용자가 파일을 직접 가져와야 한다. 따라서 자료가 여러 시스템에 분산되어 있는 경우 정보의 관리와 공유가 어려워진다.

본 고에서는 서버에 모든 GIS 파일과 스캐닝된 이미지 파일과 위성 영상이 존재한다고 가정하고, 이 파일들에 대한 공간, 비공간 속성 정보를 메타데이터로 Server의 DB내에 구축하고

사용자가 이 정보를 조회하여 사용자가 원하는 파일을 가져올 수 있도록 한다. 또한 클라이언트에서 가공된 다양한 종류의 GIS자료와 수정된 자료를 서버로 전송할 수 있도록 한다. 이미 GIS자료 파일에 대한 정보가 DB내에 구축되어 있다고 가정하고 공간 검색과 자료 파일 전송방안에 대해 기술하고자 한다.

TCP/IP를 이용한 클라이언트와 서버간의 대표적인 파일 전송 프로토콜 시스템에는 FTP가 있다. 이 프로토콜과 유사한 방법으로 클라이언트의 요청에 따라 서버에 존재하는 GIS자료의 직접적으로 획득하도록 시스템을 구축하기로 한다. 또한 이 응용프로그램과 자료 DB 사이에 인터페이스를 제공함으로써 파일 송수신에 대한 자료 정보를 DB에서 유지할 수 있도록 하는 방식을 채택한다. 파일 전송 응용프로그램 시스템은 TCP/IP를 지원하는 JAVA VIRTUAL MACHINE의 Socket과 관련된 CLASS를 이용하여 구현한다. JAVA로 개발된 응용프로그램은 JAVA 언어의 시스템 독립적인 특성으로 기종에 상관없이 거의 대부분의 시스템에서 운영될 수 있다.

파일 전송 응용프로그램은 클라이언트와 서버 응용프로그램의 두 가지 시스템으로 구성되며, 서버프로그램은 항상 서버 시스템에서 구동되고 있어야만 한다. 클라이언트 응용프로그램은 사용자의 GIS 자료 파일 요청에 따라 Server와 접속하여 요청된 파일을 가져오고 가져온 자료에 대한 정보를 자료 DB에 기록한다.

본 고는 2장에서 GIS자료 검색 시스템의 구성과 공간 검색 방법에 대해 기술한다. 3장에서는 TCP/IP위에서 소켓 프로그램을 이용하여 검색된 GIS자료 파일의 다운로드하는방법, 업로드하는 방안에 대해 기술한다. 끝으로 4장에는 결론과 향후 연구사항에 대해 언급한다.

## 2. GIS자료 검색 시스템의 구성과 공간검색

### 2.1 전체 시스템 구성도

전체 시스템 구성도 그림 2-1과 같다. 검색 시스템을 이용하여 GIS자료 파일에 대한 정보를 찾고 이 정보를 이용하여 서버에 GIS파일을 요청한다. GIS파일에 대한 DB정보는 비공간 속성 정보와 공간 속성 정보로 구성되며 검색 시 이 공간속성정보를 이용하여 자료를 찾는다. 본고에는 공간속성 정보를 이용하여 GIS자료에 대한 정보를 검색하고 자료를 서버로부터 전송 받는 과정에 대해 주로 초점을 두기로 한다. 자료 전송은 검색시스템으로부터 수집된 정보를 이용하여 파일 전송 클라이언트가 파일 전송 서버에게 요청함으로써 이루어진다. 이때 요청된 자료에 대한 이력관리를 위해 DB에 요청된 자료에 대한 기록을 한다.

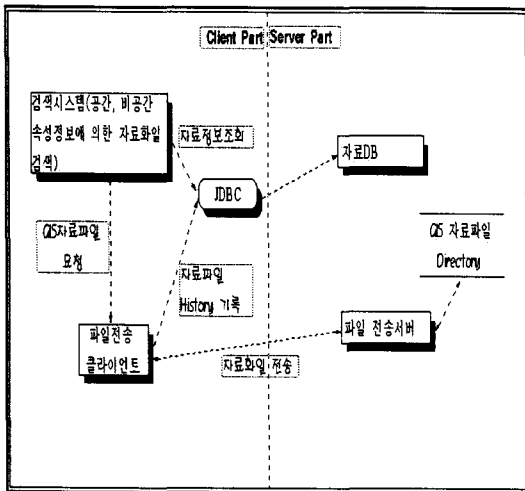


그림1. 전체시스템 구성도

### 2.2 공간 검색

공간 검색은 자료가 가지는 공간 속성정보를 이용하여 특정 위치에 존재하는 자료를 찾는다.

각 GIS자료의 공간정보에는 좌상단, 우하단 좌표가 존재한다. Egenhofer 의 공간객체는 내부 (inside), 에지 (edge), 외부 (outside) 로 나누어 관계가 성립된다. 본고에서는 사각 영역내 검색, 원 영역내 검색의 2가지 방법에 따라, 세 가지 관계에 근거하여 논리식을 도출하고자 한다,

#### 2.2.1 사각 영역내 GIS자료 검색

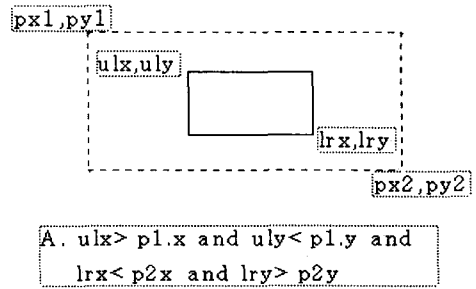


그림 2. 사각영역내 GIS 자료 검색

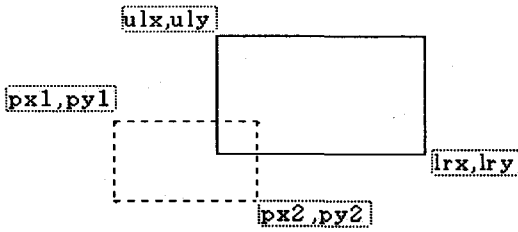
$px1$  (또는  $plx$ )과  $py1$ (또는  $ply$ )은 검색지도에서 클라이언트가 검색영역으로 지정한 사각형의 좌상단의  $x,y$  좌표에 해당되고,  $px2$  (또는  $p2x$ )과  $py2$ (또는  $p2y$ )은 검색지도에서 클라이언트가 검색영역으로 지정한 사각형의 우하단의  $x,y$  좌표에 해당된다.

대상 검색자료가 클라이언트가 지정한 범위를 벗어나지 않는 경우에 해당된다.

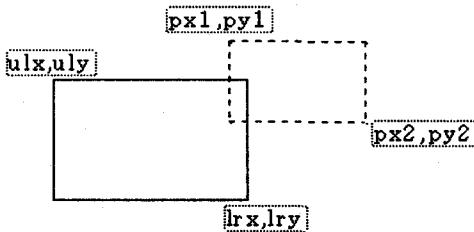
#### 2.2.2 사각검색 영역과 겹친 모든 GIS자료 검색

클라이언트의 검색영역과 검색의 대상이 되는 GIS자료가 일부 중첩이되는 경우이다. 위치에 따라 다음과 같이 세분되어 질 수 있다.

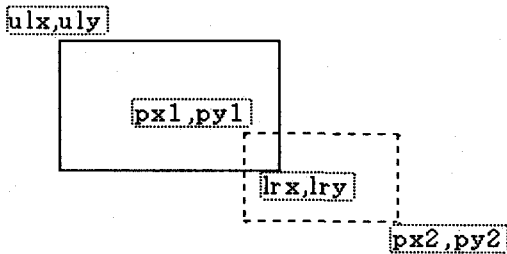
- 1) 사각검색 영역 내에 자료의 꼭지점이 포함되는 경우



B.  $ulx > px1$  and  $ulx < px2$  and  $lry < py1$  and  $lry > py2$



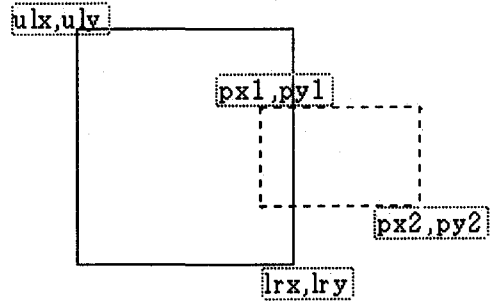
C.  $lrx > px1$  and  $lrx < px2$  and  $uly < py1$  and  $uly > py2$



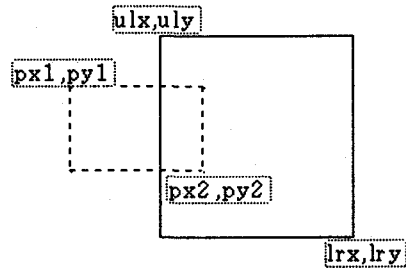
D.  $lrx > px1$  and  $lrx < px2$  and  $lry > py1$  and  $lry > py2$

그림 3. 사각검색 영역 내에 자료의 꼭지점이 포함되는 경우

2) 사각검색 영역이 자료영역의 좌측면 또는 우측면을 포함하는 경우



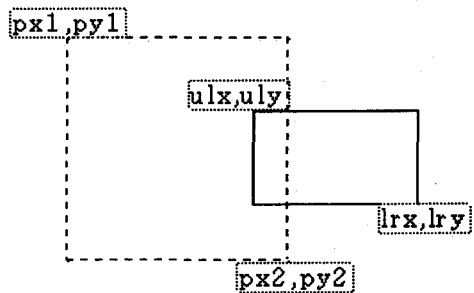
A.  $uly > py1$  and  $lry < py2$  and  $lrx > px1$  and  $lrx < px2$



B.  $uly > py1$  and  $lry < py2$  and  $ulx > px1$  and  $ulx < px2$

그림 4. 사각검색 영역이 자료영역의 좌측면 또는 우측면을 포함하는 경우

3) 자료영역이 사각검색 영역의 좌측면 또는 우측면을 포함하는 경우



A.  $uly < py1$  and  $lry > py2$  and  $ulx < px2$  and  $lrx > px2$

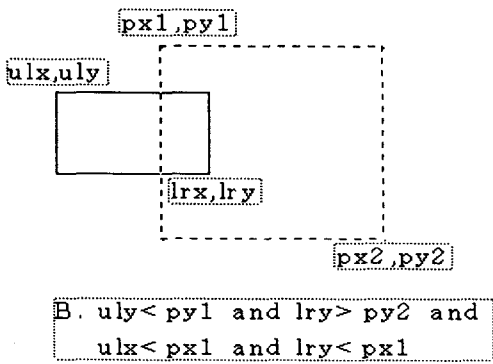
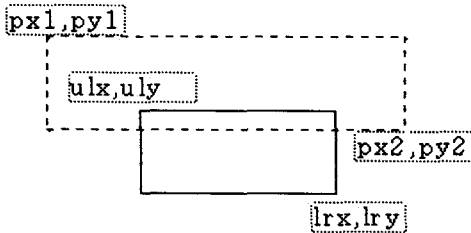


그림 5. 자료영역이 사각검색 영역의 좌측면 또는 우측면을 포함하는 경우

4) 사각검색 영역이 자료영역의 상측면 또는 하측면을 포함하는 경우



A.  $ulx > px1$  and  $lrx < px2$  and  $uly > py2$  and  $lry < py2$

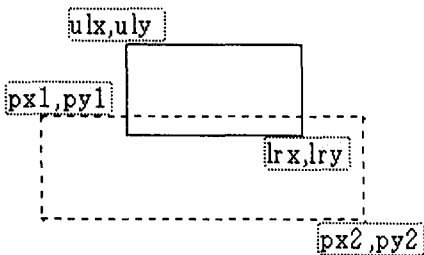
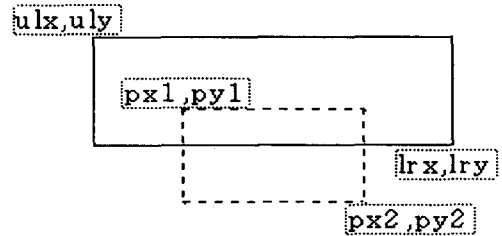
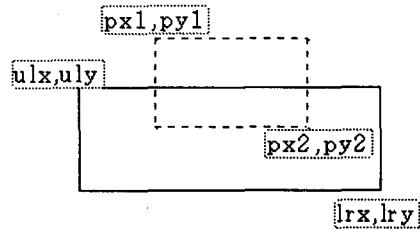


그림 6. 자료영역이 사각검색 영역의 좌측면 또는 우측면을 포함하는 경우

5) 자료영역이 사각검색 영역의 상측면 또는 하측면을 포함하는 경우



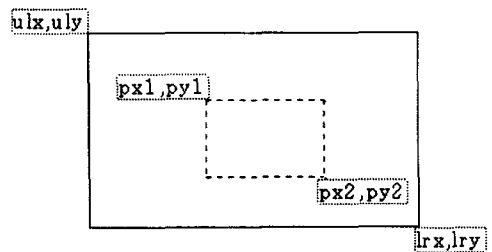
A.  $ulx < px1$  and  $lrx > px2$  and  $lry < py1$  and  $lry > py2$



B.  $ulx < px1$  and  $lrx > px2$  and  $uly < py1$  and  $uly > py2$

그림 7. 자료영역이 사각검색 영역의 상측면 또는 하측면을 포함하는 경우

6) 자료영역이 사각검색 영역을 포함하는 경우

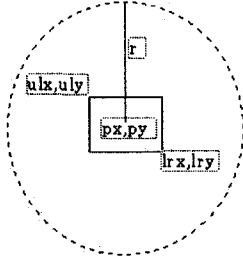


A.  $ulx < px1$  and  $lrx > px2$  and  $uly > py1$  and  $lry < py2$

그림 8. 자료영역이 사각검색 영역의 상측면 또는 하측면을 포함하는 경우

### 2.2.3 원 영역 내 검색

px와py 값이 검색영역의 중심이고, 검색영역의 반지름이 r 인 경우 관계식은 다음과 같다.



A.  $\text{power}(px - ulx, 2) + \text{power}(py - uly, 2) < \text{power}(r, 2)$  and  $\text{power}(px - lr_x, 2) + \text{power}(py - lry, 2) > \text{power}(r, 2)$  and  $\text{power}(px - lr_x, 2) + \text{power}(py - uly, 2) < \text{power}(r, 2)$  and  $\text{power}(px - ulx, 2) + \text{power}(py - lry, 2) > \text{power}(r, 2)$

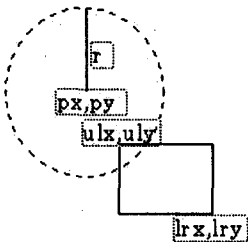
그림 9. 원 영역 내 검색

### 2.2.4 원 영역과 겹친 모든 GIS

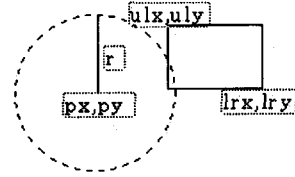
자료 검색

1) 원검색 영역내에 자료영역의 꼭지점이 포함된 경우

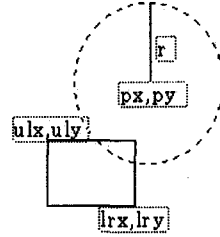
클라이언트가 원하는 검색영역을 원으로 선택할 경우 원의 영역과 검색자료의 영역이 일부 겹치게 되는 경우를 대분 하면 그림9와 같다. px,py, r에 대한 설명은 그림 9에서와 동일하다. 원의 위치에 따라 상이한 4가지의 경우가 발생한다.



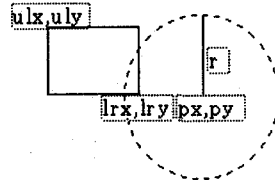
A.  $\text{power}(ulx - px, 2) + \text{power}(uly - py, 2) < r$



B.  $\text{power}(ulx - px, 2) + \text{power}(lry - py, 2) < r$



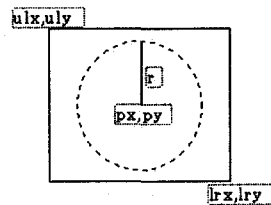
C.  $\text{power}(lrx - px, 2) + \text{power}(uly - py, 2) < r$



D.  $\text{power}(lrx - px, 2) + \text{power}(lry - py, 2) < r$

그림 9. 원검색 영역내에 자료영역의 꼭지점이 포함된 경우

2) 자료영역이 원검색 영역을 포함하는 경우



A.  $(uly > py \text{ and } lry < py) \text{ and } (\text{power}(px - ulx, 2) + \text{power}(py - uly, 2)) > \text{power}(r, 2) \text{ and } (\text{power}(px - lr_x, 2) + \text{power}(py - lry, 2)) > \text{power}(r, 2) \text{ and } (\text{power}(px - lr_x, 2) + \text{power}(py - uly, 2)) > \text{power}(r, 2) \text{ and } (\text{power}(px - ulx, 2) + \text{power}(py - lry, 2)) > \text{power}(r, 2)$

그림10. 자료영역이 원검색 영역을 포함하는 경우

## 2.3 공간 검색 식의 SQL 표현

2.2의 각 항목의 수식은 SQL로 조건절에 표현되며 항목과 항목사이는 OR로 구성된다. 예를 들어, 사각검색 영역과 겹친 모든 GIS자료 검색의 조건절에 표현되는 SQL식은 다음과 같은 함수 AreaIntercross()에 의해 만들어진다.

```
protected String AreaIntercross(Point p1, Point p2) {
```

```
    return //검색영역 내에 자료 영역의 꼭지점이 포함 되는 경우(1)
```

```
        "((ulx>"+p1.x+" and ulx<"+p2.x+")"
        "or (lrx>"+p1.x+" and lrx<"+p2.x+")) "+
        "and ((uly<"+p1.y+" and uly>"+p2.y+")"
        "or (lry>"+p1.y+" and lry>"+p2.y+")) "+
        "or " +
```

```
        //사각검색 영역이 자료영역의 좌측면 또는 우측면을 포함하는 경우(2)
```

```
        "((uly>"+p1.y+" and lry<"+p2.y+")"
        "and ((lrx>"+p1.x+" and lrx<"+p2.x+")"
        "or (ulx>"+p1.x+" and ulx<"+p2.x+"))"
        " or "+
```

```
        //자료영역이 사각검색 영역의 좌측면 또는 우측면을 포함하는 경우(3)
```

```
        "((uly<"+p1.y+" and lry>"+p2.y+")"
        "and((ulx<"+p1.x+"and lrx>"+p1.x+")"
        "or (ulx<"+p1.x+"and lrx>"+p2.x+"))) "+
```

```
        :
```

```
    }
```

## 3. GIS파일전송

### 3.1 인터넷 프로토콜

자료전송을 위한 프로토콜은 TCP/IP를 이용한다.

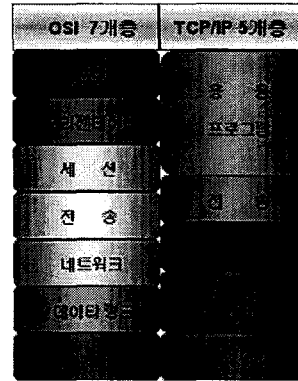


그림11. OSI 7 layer와 TCP/IP layer 비교

인터넷에서 사용하는 TCP/IP는 정확하게 보면 TCP는 OSI의 4번째 계층인 트랜스포트계층에 포함 되고, IP는 Layer 3 네트워크 계층에 각각 해당된다.

### 3.2 파일 전송 클라이언트 시스템의 구성

#### 3.2.1 검색시스템과 자료전송 클라이언트와 I/F

검색 시스템은 자료 전송 클라이언트에 수신과 송신 요청 자료에 대한 정보로 {자료명, 자료 서버 PATH, GIS자료 파일명}로 보내 준다. 여기서 파일명은 생략될 수 있으며, 이 경우는 해당되는 자료의 PATH에 존재하는 모든 파일을 서버에 요청한다.

#### 3.2.2 클라이언트 Thread

1) 클라이언트의 서버 접속

```
public class Transceiver extends Thread {
    public static final
        //서버의 port number
        int DEFAULT_PORT = 6789;
        //서버의 주소
        String hostaddress = "141.55.1.11";
        Socket socket;
```

```

boolean connected=false;
public Transceiver(Socket income) {
    this.Connect();
}

private void Connect() {
    InetAddress IPaddress;
    int port = DEFAULT_PORT;

    try {
        Ipaddress=
        InetAddress.getByAddress(hostaddress);
        initializeConnection(IPaddress, port);
        connected=true;
    }
    catch(Exception e) { //접속실패
        System.out.println("Not connected.");
    }
}

private void initializeConnection
    (InetAddress address, int port)
        throws IOException {
        //서버에 접속
    clientsocket = new Socket(address, port);
}

public void run() {
    try {
        in=new BufferedReader(
            new InputStreamReader(
                clientsocket.getInputStream()));
        out=new PrintWriter(
            clientsocket.getOutputStream(),true);
        :
    }
    catch (Exception e) {

```

```

}
:
}

```

클라이언트는 서버의 Address와 Port를 가지고 Socket을 이용하여 서버에 접속을 시도한다. 이때 접속에 실패한 경우에는 boolean 변수 connected는 false를 가지며, 성공한 경우에는 true를 갖게 된다. 클라이언트 Thread는 클라이언트와 접속된 Socket의 inputStream과 outputStream을 얻어 IO stream 인스턴스 in과 out에 할당한다. 이제부터 클라이언트와 서버의 모든 protocol 교환은 이 IO instance variable에 의존하여 이루어진다.

### 3.2.3 자료 파일 수신(downloading)

```

{자료명1, Server_PATH1, filename1}
{자료명2, Server_PATH2, ""
:
{자료명n, Server_PATHn, filenamen}

```

파일 downloading은 여러 개의 파일을 동시에 요청 가능하게 구성하였으며, 또한 요청 파일 없이 server 경로만 존재하는 경우는 해당 경로에 존재하는 모든 파일을 찾아 자료 요청 list를 다음과 같이 재구성한다. 이 때 특정 경로의 파일을 찾기 위해 LIST명령어를 이용하여 파일list를 서버로부터 가져온다.

```

private void LISTwith(String directory) { ....
}

{자료명1, Server_PATH1, filename1}
{자료명2, Server_PATH2, filename2}
{자료명2, Server_PATH2, filename3}
:
{자료명n, Server_PATHn, filenamen}

```



파일 요청 list에 대한 구성이 최종적으로 이루어지고 나면 STOR function을 이용하여 서버에 하나씩 자료를 요청하여 클라이언트시스템으로 가져온다.

```
private void getFile() {
    gatherRemoteFiles();
    for(int i=0;i<n;i++)
        copiedfile+=STORwith(Client_path,
            server_path[i], filename[i]);
}
```

### 3.2.4 자료 파일 송신(uploading)

```
{자료명1,Server_PATH1,
    Client_PATH1, filename1}
{자료명2, Server_PATH2,
    Client_PATH2, filename2}
:
{자료명n, Server_PATHn,
    Client_PATH4, filenamen}
```

업로딩파일 목록으로부터 RETR function을 이용하여 클라이언트 시스템으로부터 서버에 하나씩 자료를 전송한다.

```
private void putFile() {
    for(int i=0;i<n;i++)
        copiedfile+=RETRwith(client_path[i],
            server_path[i], filename[i]);
}
```

### 3.2.5 자료 전송 History기록

자료의 uploading과 downloading이후 전송된 파일에 대한 정보를 자료 DB table에 보관한다. 테이블 속성은 {자료명, 파일명, 일자, 송수신 상태(fail/success), downloading/uploading }로 구성되며, 클라이언트는 JDBC driver를 이용하

여 DB에 접근하여 송수신 정보를 기록한다.

## 3.3 자료 전송 서버 시스템의 구성

### 3.3.1 클라이언트의 Request 감지

서버는 ServerSocket 클래스를 가지고 Server 시스템의 포트번호 6789를 점유하는 ServerSocket을 생성한다. 클라이언트의 request를 감지하기 위해 이 ServerSocket은 Method accept()를 이용하며 클라이언트의 request가 들어왔을 때 서버는 새로운 소켓을 생성하고 이 소켓과 data를 주고 받기 위한 Thread를 생성한다.

```
try {
    ServerSocket
        s = new ServerSocket(6789);
    for(;;) {
        Socket incoming = s.accept();
        newServer(incoming,i).start();
        InetAddress
            Inet=incoming.getInetAddress();
        String address=Inet.getHostAddress();
        System.out.println(address
            + " is connected.");
    }
}
catch(Exception e){
}
```

### 3.3.2 Server Thread

Server Thread는 클라이언트와 접속된 Socket의 inputStream과 outputstream을 얻어 IO stream 인스턴스 in과 out에 할당한다. 이제 부터 클라이언트와 서버의 모든 protocol 교환은 이 IO instance variable에 의존하여 이루어진다.

```

public class Server extends Thread {
public Server(Socket income, int c) {
    incoming = income;
    counter = c;
}

```

```

public void run() {
try {
in=new BufferedReader(
    new InputStreamReader(
        incoming.getInputStream()));
out=new PrintWriter(
    incoming.getOutputStream(),true);
boolean done = false;

while(!done) {
String set=in.readLine();
// 클라이언트의 파일 요청
if(str.startsWith("RETR"))
    RETERwith(set);
//클라이언트에서
    서버로 파일전송
if(str.startsWith("STOR"))
    STORwith(set);
// 접속 종료
if(str.startsWith("QUIT"))
    done=QUITwith(set);
// 포트설정
if(str.startsWith("PORT"))
    ip=PORTwith(set);
//클라이언트의 특정
    서버 dir 파일 목록 요청
if(str.startsWith("LIST"))
    LISTwith(set);
}
incoming.close();
}
catch (Exception e) {

```

```

}
}
private Socket incoming;
}

```

### 1) 파일 목록 검색

클라이언트의 파일 목록 검색 요청은 검색을 위한 검색 명령어와 directory명 가 하나의 set 으로 서버쪽에 전달되면 이를 이용하여 이루어진다.

LIST directory\_name

서버는 이 클라이언트 요청을 in instance로부터 읽어와서 식별자 List가 file 목록 조회임을 인지한다. 따라서 File class를 이용하여 directory\_name file 목록을 감지하고 클라이언트에 이 목록을 전달한다.

### 2) 클라이언트의 서버 파일 요청

클라이언트의 서버 파일 요청은 검색 명령어와 파일명이 하나의 set로 Server쪽에 전달되면 이를 이용하여 이루어진다.

RETER [directory]file\_name

서버는 식별자 RETER가 클라이언트의 파일 요청임을 감지하고 해당 파일을 찾아 클라이언트에 전달한다. 이때 파일이 존재하지 않거나 접근이 불가능과 같은 에러를 만났을 때는 ostream에 에러 메시지를 전달한다. 파일 전송의 단위는 한번에 1024byte로 정한다.

### 3) 클라이언트 파일을 서버에 전송

클라이언트 파일의 서버 전송은 검색 명령어와 파일명이 하나의 set로 Server쪽에 전달되면 이를 이용하여 이루어진다.

STOR [directory]file\_name

서버는 식별자 STOR가 클라이언트의 전송 임을 감지고 해당 파일을 서버의 특정 directory에 writing한다. 이때 파일이 쓰기에 실패하나 접근이 불가능과 같은 에러를 만났을 때는 out stream에 에러 메시지를 전달한다. 파일 전송의 단위는 한번에 1024byte로 정한다. 파일 directory가 존재하지 않는 경우는 directory 생성 클래스 이용하여 새로운 directory 생성한다.

#### 4. 결론

본 고에서는 자료의 공간 검색 방법, 검색된 자료의 downloading 방법, 자료의 uploading 방법에 대해 기술하였다. 자료의 검색은 자료에 대한 공간 정보가 이미 DB내에 존재하는 것으로 가정하고 사각 검색과 원형 검색 방법에 대해 설명하였다. 공간 검색은 자료가 사각형의 영역 좌표(좌상단, 우하단)를 가지고 있고, 이 자료에 대하여 원형 검색과, 사각영역 검색을 겹침과 포함으로 구분하여 공간 검색을 할 수 있는 방안을 제시하였다. 또한 검색된 자료의 list를 이용하여 서버로부터 파일을 다운로드할 수 있도록 하는 클라이언트와 서버의 시스템 구성에 대해 설명하였다. 본 고의 모든 내용은 JDK Version 1.1.5를 통해 구현되었으며 DB는 ORACLE를 사용하였다.

향후 연구 사항으로는 공간 검색 방법으로 자료 영역과 검색 영역이 사각형나 원으로 한정되는 범위를 넘어 다양한 유형의 다각형으로 존재한다 하더라도 검색이 가능하도록 하는 방법에 대한 연구가 필요하다.

#### 참고문헌

- [1] 구호본 외 1995 "산사태 데이터베이스 시스템의 GIS 이용" 한국 GIS 학회지 문상호 외 1999 " ISO/TC211 지리공간/지리정보 서비스의 표준 (I) 개방형 GIS 연구회지 1:1 pp48-54
- [2] 문상호 외 1999 " ISO/TC211 지리공간/지리정보 서비스의 표준 (I) 개방형 GIS 연구회지 1:1 pp48-54
- [3] Tryfona, N. and Egenhofer, M. 1996 Multi-Resolution Spatial Databases: Consistency Among Networks. International Workshop of FMLDO (Foundations of Models and Languages for Data and Objects ): 119-132
- [4] 김은형, 1999 "지리정보시스템의 현황과 미래" 개방형 GIS 연구회지1:1 23-3

#### 장은미

서울대 지리학과 학사 1986  
 서울대 대학원 석사 1988  
 캔사스 주립대 박사 1997  
 현재 삼성 SDS GIS 사업팀 책임

#### 김창제

충북대 컴퓨터공학과 학사 1994  
 충북대 컴퓨터공학과 석사 1996  
 현재 삼성 SDS GIS 사업팀 전임

## 부록 : 미국 대학 지도도서관의 특성정리

1. 미네소타주립대학, Borchert Map library, 27 만 건의 지도와 지도첩 8천여 권 소장, 분류체계는 1차로 지도제작기관중심 분류, 미국지도 (지형도, 기관별 지도 분류, 기타주제도), 주정부관련 지도, 외국지도 ( CIA, DMA 지도) 인도 및 동남아시아 지도, 세계지도 (ONC,TPC,JOG) , 순수하게 종이지도에 국한하여 정리
2. 서 일리노이주립대학, 지도도서관, 제작기관별 분류체계로 일관하고 있음, CD-ROMs 형태의 수치지도도 함께소장, 지도첩 1400 여개 소장
3. 콜로라도 광업대학의 지도도서관, 지형도와 지질도 검색을 위한 web service 수행이 중이나, 위치와 sheet의 명칭을 알고 있어야 질의가 가능함
4. 캘리포니아 버클리대학, 지도제작기관별 분류, chart, 교통도, 도시, 주제도로 대분, 지도의 상태 (지명록,지도첩, 낱장, 말린지도, 접힌지도 등의 형태분류체계) 정보가 많다.
5. 텍사스 주립대학, Perry Castaneda library, 대륙별, 국가별, 지역별 지도분류, online mappin service 로 미국 CIA 에서 발행한 종이지도를 jpg 형태로 스캐닝하여 web service 함
6. 캘리포니아 산타바바라 대학, Alexandra Library, 기존지도의 수치지도화에 가장 투자를 많이 하는 지도도서관, AVHRR 부터 항공사진에 이르는 영상자료를 지질국지도와 함께 소장하면서 대학 주변지역에 관한 응용예시를 보여줌, 메타데이터와 관련된 참고문헌에 대한 정리가 제공됨
7. 캔사스 주립대학교, 정부 간행물 및 지도도서관, Library of Congress 의 원리에 따른 지도분류체계를 유지하면서, TIGER file 을 편집하여 맞춤지도 제작을 할 수 있음. 한국과 베트남 전 관련 지도가 많고 EPA (미 환경국) 자료와 연동하여 TMS 자료를 연동하여 모델링 작업 까지 도서관에서 수행가능함
8. 컬럼비아 대학, 프랑스 혁명 시 파리의 지도 100 여장을 jpg로 스캐닝하여 서비스함
9. 조지아 주립대학, Hargrett library, 회귀본 역사지도를 스캐닝하여 서비스함