韓國地形空間情報學會論文集 第8卷 第2號 2000年 6月 pp. 109-116

GeoNet: 웹 기반 위성영상 처리

GeoNet: Web-based Renotely Sensed Image Processing System

안충현*, 김경옥*
Chung-Hyun Ahn, Kyung-Ok Kim

要 旨

자바 언어를 이용하여 구축한 위성 영상 처리 소프트웨어인 GeoNet은 자바 언어의 장점을 그대로 수용하는 cross-platform 대용량 위성 영상처리 API로써의 인터페이스를 제공하며 개발 기간을 단축하는 자바 객체지향 패러다임의 기반에서 구축되었다. 네트워크 환경에서의 자바 확장성을 이용한 클라이언트/서버 이미지 처리의 적합성과 융통성 있는 시스템 구조로의 기반을 가지며 웹브라우저를 통한 실행도 GeoNet의 특징이다.

본 연구에서는 자바 언어를 통한 위성 영상 처리 소프트웨어 GeoNet의 개발을 통해 앞으로 확대될 위성 영상의 보급과 분산 환경에서의 영상 처리 요구에 신속히 대처할 수 있는 대안을 제시한다.

ABSTRACT

GeoNet is java-based remotely sensed image processing system. It is based on java lbject-oriented paradigm and features cross-platform, web-based execution and extensibility to client/server remotely sensed image processing model.

Remotely sensed image processing softwares made by java programming language can suggest alternatives to meet readily demand on remotely sensed image processing in proportion to increasement of remotely sensed data.

Keywords : Remotely sensed image processing software

1. 서 론

위성에 의한 정보 취득은 광역에 대한 정보를 쉽게 취득할 수 있으며, 동일한 센세에 의해 주기적으로 취득되는 정보는 표준화된 자료 처리를 통하여, 자료 처리를 행한 작업자의 경험과 지식과는 관계없이 객관적인 정보의 획득이 가능하다.

1972년 LANDSAT-1호의 발사를 시작으로 인공이성 영상을 이용한 지표면 정보의 취득 및 정보 추출 기술은 괄목할 만한 성장을 해 오고 있다. 현재 위성에 따라서는 10m/pixel 미만의 공간 해상도와 보다 다양한 파장대의 센서로부터 영상을 얻어, 지표면에 대한 많은 정보의 추출이 가능하게 되었다. 종래의 원격탐사기술은 화상처리에 의한 각종 정보 취득기술

* 한국 전자통신연구원
컴퓨터 · 소프트웨어기술연구소RS팀

개발과 래스터 데이터 분석이 중점이었으나, 다양한 위성 정보의 실제적 활용이 증가되면서, 점차 사진에 의한 각종 지도의 작성과 자원 탐사 및 광역의 환경 변화 감시 등 다양한 응용 분야로 그 활용범위가 확대되고 있다. 위성에 의하여 관측된 정보를 이용하여 각각의 목적에 맞게 2차 가공된 데이터는 격자형 지리 정보의 일종으로 취급할 수 있으며, 해당 지역의 토지 이용에 관한 현황이나, 자연 환경, 토지 피복 분류, 식물 및 식생의 활성도 또는 각종 주제도의 갱신에 사용될 정도로 지리정보시스템에 있어서 중요한 정보원의 하나로 이용되고 있고, 근래 위성자료에 의한 지구 환경 정보에 대한 데이터베이스의 구축이 활발하게 진행되고 있다.

초미세분광분해능센서(Hyperspectral sensor), 고공간분해능 센서(High spatial resolution sensor) 등으로 향후 획득되는 고품질 위성영상은 최소 수백 MB/scene 정도의 정보량이 될 것으로 예상된다. 그렇지만, 현재 사용화 되어있는 대부분의 위성영상처리 S/W는 초미세분광 영상과 같은 많은 분광대의 자료를 대상으로 하고 있지 않으며, 대상 영상을 전부 메모리에 올려놓고 작업을 하는 방식을 취한 것이 대부분으로, 이러한 방법으로는 비록 RAM용량이 비약적으로 증가되거나 하드디스크의 용량이 비약적으로 증가된다고 하더라도 수십 메가바이트 이상의 영상자료에 대하여 유연한 처리 및 분석에는 한계가 있을 것으로 여겨진다. 또한, 인터넷 및 초고속 정보망을 이용한 일괄적인 위성정보의 가공, 처리, 분석 및 부가가치 정보의 생산에 대한 일련의 서비스에 대한 요구 또한 증가되고 있다. 이러한 요구를 만족하기 위해서는 단순히 정보 제공자로부터 일방적으로 가공되어 준비된 정보를 제공받는 수동적인 형태의 서비스보다는 사용자가 서비스 시스템에 들어와 자기가 원하는 정보를 찾거나, 자신의 정보를 이용하여 또 다른 형태의 부가가치 정보를 가공한 후, 그 결과를 다시 자신의 시스템으로 가져갈 수 있는 능동적인 형태의 서비스 시스템의 개발이 필요해 진다. 이러한 네트웍을 이용한 컴퓨팅이 가능해지기 위해서는 정보망의 자료전송속도 및 연결된 컴퓨터간의 고속분산처리 기능 등에서의 하드웨어적인 성능 향상과 더불어 소프트웨어적으로도 이에 적합한 분석시스템이 개발되어야 한다. 또한 미래의 시스템은 각각의 영상에 적용된 처리과정에 대한 상세한 정보를 제공할 수 있어야 한다. 이를 가능하게 함으로서, 각각의 정보가 어떠한 영상으로부터 어떠한 처리과정을 거쳐서 얻어졌는가에 대한 명확한 추적이 가능하게 된다.

본 논문에서 소개할 GeoNet은 이러한 소프트웨어 개발 필요성에 맞추어 개발된 자바 언어 기반의 시스템이다.

2.시스템 구조

GeoNet은 자바 가상 머신을 가지는 어떠한 시스템에서도 동작하며 객체 지향개념을 바탕으로 설계되었다. 그리고 시스템의 확장이 용이하며, 위성 영상 처리 API로써 사용될 수 있도록 하였다.

GeoNet에서는 대용량 위서 영상을 처리하기 위하여 파일 접근에 있어서 대용량 자료에 최적한 시스템 구조를 가지고 있다. 예를 들어 영사의 화면 출력시 세 개의 화면을 동시에 사용하여 가상메모리에서 발생할 수 있는 스왑핑을 피하고 있으며 영상 처리 모듈은 대부분 라인 단위로 영상을 읽어들여 처리하도록 구현되었다.

Figure 1은 시스템의 전체적인 구조를 나타낸 것이다. 기반이 되는 것은 래스터와 백터 영상의 출력과 위성 영상 처리 엔진 부분이다.

Core 패키지는 GeoNet에서 가장 핵심이 되는 클래스들의 모음이다. 여기에는 영상을 나타내는 Cube 클래스, 베터를 나타내는 GeoVec 클래스 등이 포함된다. IO 패키지는 파일 입출력과 관련된 클래스들의 모음으로 영상의 파일 입출력과 관련된 클래스들을 포함한다. Utility 패키지는 여러 헤더 파일을 읽기 위한 파싱 관련 클래스, 그리고 조표변환 등과 같은 범용적인 역할을 하는 클래스들이 포함된다.

Process 패키지는 실제 위성 영상 처리를 수행하는 여러 알고리즘들이 구현된 클래스들로 구성된다. 그리고 각각의 알고리즘들은 Supervisor 클래스와 Operation 클래스의 쌍으로 이루어진다.

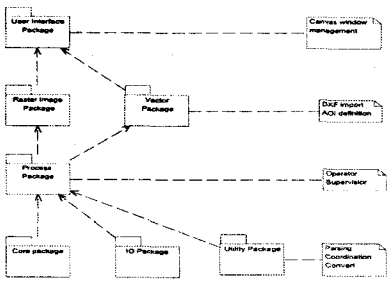

Fig.1. GeoNet System Architecture 1

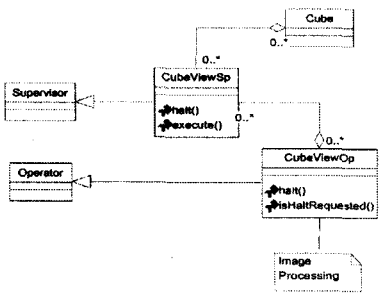

Fig.2. Supervisor and Operator Class Diagram

Raster Image 패키지는 팔레트와 같은 래스터 이미지 처리와 관련된 클래스들을 포함하며 Vector 클래스들은 AOI와 같은 클래스들을 포함한다.

이렇게 구성된 각각의 모듈은 다른 모듈들과의 의존성을 최소화하여 확장이 용이하도록 객체지향적으로 설계된 것이다.

Figure 2는 프로세스 패키지 내부의 쓰레딩 구조를 지원하는 Supervisor와 Operator 클래스와 그 관계를 클래스 다이어그램으로 나타낸 것이다.

Supervisor 객체는 실제 영상 처리를 담당하는 Operation 객체를 관리하며 연관되는 사용사 GUI등과 Operation 객체를 연결하는 역할을 수행한다. Operation 객데와 연관되는 GUI를 생성하고 스스로 새로운 쓰레드를 생성하고 이 쓰레드로 하여금 영상처리를 수행할 Operation 객체를 생성하여 영상처리 작업을 수행하도록 한다.

Supervisor 클래스의 하위클래스들은 모두 halt와 execute 메소드를 구현하며 halt 메소드는 사용자나 혹은 임의의 작업 중지 요청에 의한 처리를 수행하며 execute 메소드는 Operation 객체의 생성을 통해 영상처리 작업을 수행하는 메소드이다.

3.시스템 구현

본 논문엣 구현한 GeoNet의 환경은 자바 가상 머신 1.1.7B 버전에서 개발하였으며 특정 개발 도구에 의존하지 않고 각 도구의 특성에 따라 부분적으로 이용하며 개발하였다. 현 시점에서 이미 공개된 자바 가상 머신 2 플렛폼에는 속도상의 문제가 있어 환경 변경을 하지 않고 있는 실정이다.

3.1. 영상 출력

위성 영상은 Cube 객체, CubeFile 객체, CubeMem 객체로써 표현된다. Cube 객체는 파일의 위성 영상을 나타내는 CubeFile 객체와 메모리의 위성 영상을 나타내는 CubeMem 객체의 상위 클래스이다. CubeFile 객체는 파일의 위성 영상을 가리키는 파일 포인터로 구성된 객체이며 CubeMem 객체는 메로리의 포인터를 가지는 객체이다. 모든 영상을 메로리에 로드한 후 처리를 하게되면 필요한 메로리의 양이 너무 커지게 된다. 이런 이유로 GeoNet에서는 사용자가 필요에 의해 영상을 메모리에 로드하는 경우는 CubeMem 객체를 사용하고 로드할 필요가 없는 경우는 Cubefile 객체를 사용하도록 구분하였다. 작은 용량의 신속한 처리를 위해서 메로리에 저장하는 영상을 사용하는 것이 더욱 효과적이며 모든 명령어는 파일에 저장된 영상뿐만 아니라 메모리에 저장된 영상도 처리할 수 있도록 구현되었다.

이와 함께 GeoNet에서는 대용량 영상의 처리 속도

를 향상시키기 위해서 BIL(Band Interleaved by Line) 구조를 기존 영상 파일 구조로 사용하였다.

3.2. 영상 관리

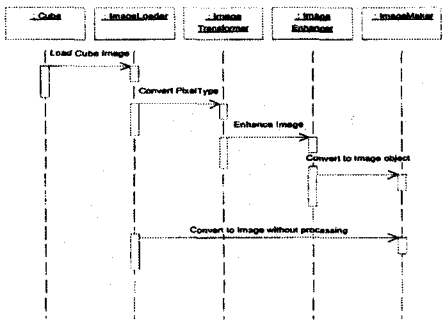

Fig.3. GeoNet Image Foow Sequence Diagram

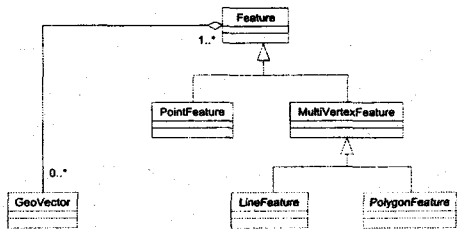

Fig.4. Class Diagram of GeoVector

GeoNet에서의 GUI와 관련된 이미지 관리는 크게 ImageRGBManager 객체와 ImageBWManger 객체가 관리한다. ImageRGBManger 객체는 컬러 모델에서의 화면과 이미지 관리를, ImageBWManager 객체는 흑백 모델에서의 역할을 수행한다. 실제 Cube 객체를 메모리로 로드하는 것은 ImageLoader 객체가 담당한다. ImageLoader 객체는 사용자가 요구한 Cube 객체의 타입이 파일형인지 메모리형인지를 구분하여 각 타입에 적합한 IO 연산을 통해 메모리에 영상을 로드한다. 그리고 로드된 영상을 실제 자바에서 이미지 처리를 위한 Image 객체로 변환시키는 역할을 하는 객체가 ImgeMaker 객체이다. Figure 3은 이미지 관리 구조를 나타낸 것이다.

ImageTransformer 객체는 여러 data type을 모두 byte형으로 변환하는 역할을 수행한다. DEM 영상의 경우는 그 영상의 픽셀 종류가 double 형으로 되기 때문에 이러한 영상을 처리하기 위해서 내부적으로 사용되는 객체가 ImageTransformer 객체이다.

ImageEnhancer 객체는 ImageTransformer를 거친 byte 영상을 픽셀에 적용하는 영상처리 알고리즘을 수행하여 변경한다. ImageEnhancer에서 지원하는 알고리즘에는 히스토그램 평활화나 가우시안 평활화 등이 포함된다.

3.3. 벡터 프로세싱

GeoNet에서는 DXF파일을 읽어들여 자체의 파일 형태인 GeoVec형식으로 변경한뒤 처리한다. GeoNet에서는 Feature들을 Point, Line, Polygon으로서 나타내며 Line과 Polygon의 경우 MultiVertex Feature에서 상속받는 구조로 되어있다. 이 구조는 현재까지 일반적으로 사용되는 벡터 표현 형태로 볼 수 있다. Figure 4는 이것을 클래스 다이어그램으로 나타낸 것이다.

3.4. 다중 쓰레드 자료 처리 개념

현재 워크스테이션급 플랫폼은 2개 이상의 CPU를 장착하는 것이 가능하며, 향후 이와 같은 다중 프로세서 시스템은 점차 보편화 될 것으로 예상된다. 그러나 기존의 프로그램 언어, 예를 들면 C 나 C++로 다중 쓰레드 프로그램을 작성하는 작업이 매우 복잡하여 대부분의 위성 영상 처리 프로그램은 다중 프로세서 환경을 충분하게 활용하지 못하고 있다. 그 결과 다중 프로세서 시스템에서 CPU의 사용률이 10%~20% 정도에 불과하다. GeoNet에서는 이것을 극복하여 CPU의 활용을 최대로 할 수 있도록 일련의

쓰레드들이 연결되어 하나의 작업을 수행하는 구조를 갖는다.(Figure 5).

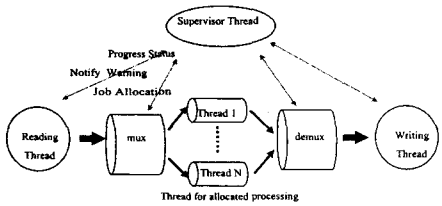

Fig.5. Concepts of Pipeline Data Input/Output and Multi-Thread Image Processing

예를 들어 설명하면, Figure 5에 나타낸 바와 같이, 읽기 전용 쓰레드가 입력 파일을 조금씩 읽어서 자료 처리 쓰레드에 넘겨주면 자료 처리 쓰레드는 담당하고 있는 처리 방법에 다라 이를 처리하여, 다음 단계의 쓰레드에게 넘겨 주고, 최종적으로 쓰기 전용 쓰레드가 그 결과를 출력 파일에 기록하는 구조를 취하는 것이다. 물론 자료 처리 쓰레드는 작업의 성질에 따라 여러 개의 기본적인 작업을 수행하는 하위 쓰레드로 분리될 수 있다. 각 쓰레드는 주기적으로 또는 특정한 이벤트가 발생하였을 경우, 작업 감독 쓰레드에게 자신의 작업 상태에 대한 보고를 하도록 되어 있다. 작업 감독 쓰레드는 각각의 쓰레드들을 감시하고, 문제가 발생할 경우 쓰레드들의 작업을 중지시키고, 사용자에게 이를 통보하도록 한다. 또 한 자신에게 할당된 작업이 완료된 쓰레드는 작업 종료와 동시에 메로리에서 제거되며, 자신에게 할당된 작업이 아직 시작되지 않은, 즉 자료의 입력을 기다리고 있는 쓰레드는 작업 감독 쓰레드로부터 작업을 시작하라는 지시를 받아야 메로리에 올라와 작업을 수행한다. 이로써비록 많은 쓰레드들로 프로그램이 분할 되어 있더라도 각 쓰레드가 차지하는 메모리 영역을 최소화 할 수 있다.

4. 실험 및 결과

4.1. 쓰레드를 통한 성능 평가

앞에서 설명한 다중 쓰레드와 객체파이프 자료 입출력 개념에 의한 위성영상처리의 성능을 평가하기 위하여 처리 대상 영상 파일의 크기와 CPU의 수를 다르게 하여 기존의 순차적 프로그래밍 구조를 갖는 Sobel엣지 추출과 다중 쓰레드에 의한 sobel엣지 출출 결과를 비교하여 보았다(Figure 6). 그림에서 보이는 바와 같이 순차 구조인 경우 CPU의 증가에 따라 별다른 성능의 향상은 기대되지 않는다. 반면, 다중 쓰레드 구조의 경우 CPU 1개의 경우 쓰레드간의 통신으로 인하여 다소 순차구조에 비해 성능은 떨어지지만 CPU가 증가할수록 순차 구조와 비교하여 성능이 향상되며, 이는 영상 파일이 클수록 뚜렷하게 나타나, CPU 5의 경우 기존의 순차구조에 의한 결과보다 약 3.7배 정도성능이 향상됨을 알 수 있다. 따라서, 앞으로 다중 CPU를 탑재한 컴퓨터의 일반화 경향과 점차 대용량화 되는 위성영상의 신속하고 효율적인 처리를 위하여는 다중 쓰레드 구조를 갖는 시스템이 필요함을 알 수 있다.

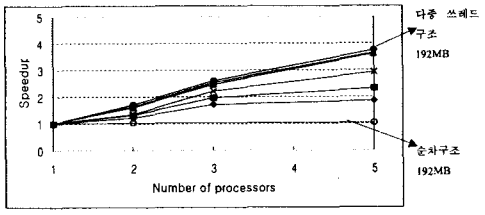

Fig.6. Evaluation result of Multithread data processing

4.2. 위성 영상처리 API로서의 GeoNet

위성 영상처리를 위한 API로써 GeoNet은 크게 3부분으로 이루어진다. 시스템의 core 부분과 I/O 부분, 그리고 여러 영상처리 알고리즘 API이다. 일반적인 언어의 API와는 달리 자바 언어는 객체 기반의 API 구성을 가능하게 한다. 이것은 API를 사용하는 입장에서 보았을 때 보다 쉬운 사용을 제공하는 것이다. Table 1은 현재 본 연구소에서 기술이전 중인 업체가 GeoNet API를 이용하여 간단히 만든 Cube 이미지 파일을 보여주는 프로그램이다.

4.3. 웹 기반 프로세싱

GeoNet은 자바 언어의 특징을 살려 웹에서 브라우저를 통해 애플릿으로 수행 가능하도록 설계되었다. GeoNet의 개발환경과 동일한 환경에서 수행될 수 있도록 자바 Plug-in을 자동으로 웹브라우저를 통해 내려받아 설치하도록 하였으며 사용자가 로드 된 애플릿을 통해 직접 로컬 시스템의 위성영상을 처리할 수 있도록 공개키 방식의 인증을 통해 허가된 사용자만이 사용할 수 있도록 하였다. 사용자는 인증서만 있으면 장소의 제약을 받지 않고 어느 곳에서든지 웹을 통해 위성영상 처리 업무를 수행할 수 있는 것이다.

```
...
import geo.core.*;
import geo.process.*;
....
  private Image imgBuf = null;

  public ImageLoaderTest() {
       int _bands, int _lines, _samples ;
       String outFile = new String("test.bmp");
      ImageLoader rgbImger = null;
      ImageMaker maker = null;
       CubeFile cubFile = new CubeFile("img1000.cub");
      CubeMem rImg = new CubeMem("r");
      CubeMem gImg = new CubeMem("g");
      CubeMem bImg = new CubeMem("b");
      cubFile.hd.read(cubFile.getHDFile());
      _samples = cubFile.hd.samples; _lines = cubFile.hd.lines; _bands = cubFile.hd.bands;
      rgbImger = new ImageLoader();
      maker = new ImageMaker();
      rImg = rgbImger.load(cubFile, 2, rImg, 1);
      gImg = rgbImger.load(cubFile, 1, gImg, 1);
      bImg = rgbImger.load(cubFile, 0, bImg, 1);
      imgBuf = maker.getImage(rImg, gImg, bImg);

  }
}
```

Table.1. Sample Program Using GetNet API

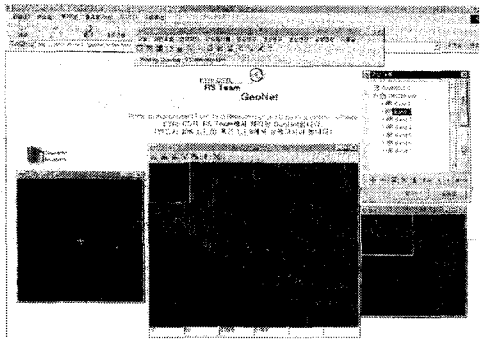

Fig.7. HTML interface to GeoNet Server

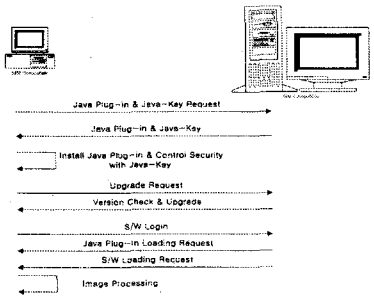

Fig.8. GeoNet Network Operation Flow

Figure 7은 웹브라우저를 통해 GeoNet을 구동시킨 화면으로 <http://rsn시.etri.re.kr/geonet/index.html> URL을 통해 수행된다. 수행을 위해서는 인증서를 받아 시스템에 인증시켜야 GeoNet 애플릿이 수행될 로컬 시스템의 정보를 파악하여 로드될 수 있다.

Figure 8은 인증과 관련된 네트워크 상에서의 GeoNet의 동작에 대하여 설명하는 그림이다. 클라이언트는 플러그인을 설치하고 인증서를 서버측에 요구한다. 서버측은 인증된 사용자인 것을 확인한 후 인증서를 클라이언트 측에 전달하면 사용자는 전달받은 인증서를 통해 시스템의 자원을 애플릿이 접근할 수 있도록 인증한다. 이렇게 인증된 시스템에서 웹브라우져로 애플릿을 연결하면 GeoNet 애플릿은 인증된 시스템인지를 확인한 후 로컬 시스템의 자원을 접근하여 필요한 정보를 얻은 후 로드되어 실행된다.

5. 개선점 및 향후 계획

GeoNet을 웹 브라우저에서 애플릿으로 실행시키기 위해서는 서버에서 인증한 인증서를 필요로 한다. 인증 된 인증서로서 사용자는 로컬 시스템에 있는 위성 영상들을 처리하기 위해 인증서를 인증 목록에 넣어 주어야 하는데 이 과정을 보다 쉽게 처리하기 위한 인증 에이전트의 도입이 필요하다. 인증 에이전트는 인증 된 사용자와 서버와의 연결 및 인증 과정 및 인증서 전달에 대한 모든 역할을 수행해야 할 것이다.

웹을 기반으로 한 시스템의 구축은 자바의 네트워크 프로그래밍의 장점 중 일부이다. 자바 네트워크 기능을 충분히 활용하기 위해서는 객체의 전송을 통한 진정한 의미의 분산 위성 영상 처리 개념이 충족될 때에야 비로써 이루어진다고 할 수 있다. 분산 위성 영상 처리 시스템은 제한된 자원의 최대 활용을 위해서는 필수적인 요소이기에 앞으로 구축해야 할 과제인 것이다.

6. 결론

자바 언어를 이용하여 구축한 위성 영상 처리 소프트웨어인 GeoNet은 Java 언어의 장점을 그대로 수용하는 다음과 같은 장점을 가진다.

(1)cross-platform 대용량 위성 영상처리 API로써의 interface를 제공

(2)개발 기간을 단축하는 Java object-oriented paradigm

(3)Java RMI 와 JavaBians와 같은 자바 네트워크/컴포넌트 기술을 기반으로 하여 네트워크 환경에서의 자바 확장서을 이용한 클라이언트/서버 위성 영상처리에 적합

(4)융통성 있는 구조의 시스템 적용

GeoNet의 구현을 통해 자바 언어를 통한 위성 영상 처리 소프트웨어 개발은 앞으로 확대될 위성 영상의 보급과 분산 환경에서의 영상 처리 요구에 신속히 대처할 수 있는 대안을 제시할 수 있음을 보였다.

7. 참고문헌

1) 시스템공학연구소, 1987, 마이크로 컴퓨터 Image Processing System 개발 연구(II), 연구 보고서, 233p.
2) 시스템공학 연구소, 1991, 위성 영상 및 항공기 탑재 리모트센싱 자료 분석을 위한 마이크로 컴퓨터 영상 처리 시스템 개발 연구(III), 연구 보고서, 134p.
3) 한국 자원 연구소, 1994, PC VGA용 화상처리 소프트웨어 및 응용 기술 개발, 연구 보고서, 48p.
4) Arnold, K and Gosling, j., 1996, The Java Programming Language, Addition-Wesley, 333p
5) Ayers, L. F., 1995, Data vendor + Software vendor = Successful solution for the remote sensing user, Land Satellite Information in the next decade,III-26 ~ III-29
6) Chan, P. and R. Lee, 1996, The JavaTM Class Libraries An Anotated Referencd, Addison-Wesley, 1660pp.
7) Lee Chan, The java Class Libraries-2nd ed., Vol.2, Addison Wesley, pp.795-844.
8) Fritz, L. W. 1996, The Era of commercial earth observation satellites, Photogrammetry Engineering & Remote Sensing, 39 ~ 44.
9) Gosling, J. F. Yellin and The Java Team, 1996a, The JavaTM Application Programming Interface, Volumne I, Addison-Wesley Publishing Company, 494
10) Gosling, J. F. Yellin and The java Team, 1996b, The JavaTM Application Programming Interface, Volumne II, Addison-Wesley Publishing Company, 406
11) Nelson, L. J., 1997, Space Imaging, Earthwatch, and the satellite imaging picture in 1997, Advanced Imaging, 61 ~ 63
12) Joo, S.H., M. Hion, H. Y. Choi and D. H. Jung, 1995, The prospect of commercial space remote sensing business in Asia-Pacific region.
13) Sinclair, S., 1996, Expanding role of GIS satellites in Asia Pacific, GIS ASIA PACIFIC, 2, 3, p.27 ~ 32.
14) Thibault, D., 1995, Land Satellite Information in the future, Land Satellite Information in the next decade, III-14 ~ III-21