GeoNet : Web-based Remotely Sensed Image Processing System

Jong-Yoon Yang, Chung-Hyun Ahn, Kyoung-Ok Kim

Remote Sensing Team, Image Processing Department
ETRI-Computer and Software Technology Laboratory
(305-350) 161 Kajong-dong Yusong-Gu Taejon, Korea
Tel : +82-42-860-5374,Fax : +82-42-860-4844
E-mail : yangjy@etri.re.kr

Abstract.

Previous technology of remote sensing was focused on analyzing raster image and gaining information through image processing. But now it has extended to diverse fields like automatic map generation, material exploitation or monitoring environmental changes with effort to utilizing practical usage. And with rapid expansion of information exchange on Internet and high-speed network, the demand of public which want to utilize remotely sensed image has been increased. This makes growth of service on acquisition and processing remotely sensed image.

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

Remotely sensed image processing software made by Java programming language can suggest alternatives to meet readily demand on remotely sensed image processing in proportion to increase of remotely sensed data.

In this paper, we introduce GeoNet and explain its architecture.

Keywords : Remotely sensed image processing software

1 Introduction

Satellites have acquired information easily with wide area. It is periodical and verifiable by standard procedure in regardless of processor's experience or knowledge. With start of launch of LANDSAT-1 in 1972, technology acquiring and extracting remotely sensed satellite image have been developed remarkably. Recently a group of satellites have capability of 10m/pixel spatial resolution and multi-spectral sensor. It makes us get more information from land cover on earth.

Previous technology of remote sensing was focused on analyzing raster image and gaining information through image processing. But now it has extended to diverse fields like automatic map generation, material exploitation or monitoring environmental changes with effort to utilizing practical usage. In addition to that, building environmental information database of earth using remotely sensed satellite image has being performed actively.

High-quality remotely sensed image with which is made up of hyper-spectral sensor, high spatial resolution sensor is expected to be a couple of hundreds of MB/scene. Most of commercial remotely sensed image processing software does not deal with many bands with hyper-spectral sensor image and usually manipulates data after loading into memory. Although increase of capacity of memory and disk space, it seems to have a limitation to handle large remotely sensed image efficiently.

With rapid expansion of information exchange on Internet and high-speed network, the demand of public that wants to utilize remotely sensed image has been increased. This makes growth of service on acquisition and processing remotely sensed image.

It requires active system which users can get into the service and find resources. It is also essential to process data and to transfer the result into own system instead of passive system that can only get the information processed before by others. It is indispensable to develop appropriate software system. It should provide us with detail history information about procedure of individual image processing. It makes it possible to take a track of history of processed image.

In this paper, GeoNet is a Java-based remotely sensed image processing software system to meet these demands. The purpose of this document is to introduce this system and explain its architecture.

2 System Architecture

GeoNet is executed in any operating system with Java virtual machine and is based on object oriented paradigm. It is so much flexible to be used as remote sense image processing API.

In order to manipulate high volume satellite image GeoNet is consisted of optimized file i/o structure. For instance, it has three viewing windows to avoid memory swapping and deals with image line by line. Figure 1 is a overall system architecture. Basics are display module of vector and raster image and image processing engine.

Core package is a collection of classes comprising of kernel in GeoNet. It includes Cube class which stands for satellite image and GeoVec class which representing vector. Classes in IO package take charge of image input and output processing. Utility packages have common classes in which use in general all over system. It is such as file header parsing, converting coordination.

Process package is made up of classes that implement image processing algorithms. Each algorithm makes a pair of Supervisor class and Operator class. Classes in raster image package carry out color palette controls. Vector classes deals with AOI operation with many classes. These module is designed with object oriented perspective to expand more easily and to minimize the dependency among modules.

Figure 2 represents relationship between Supervisor and Operator class operating with threading model in process package.

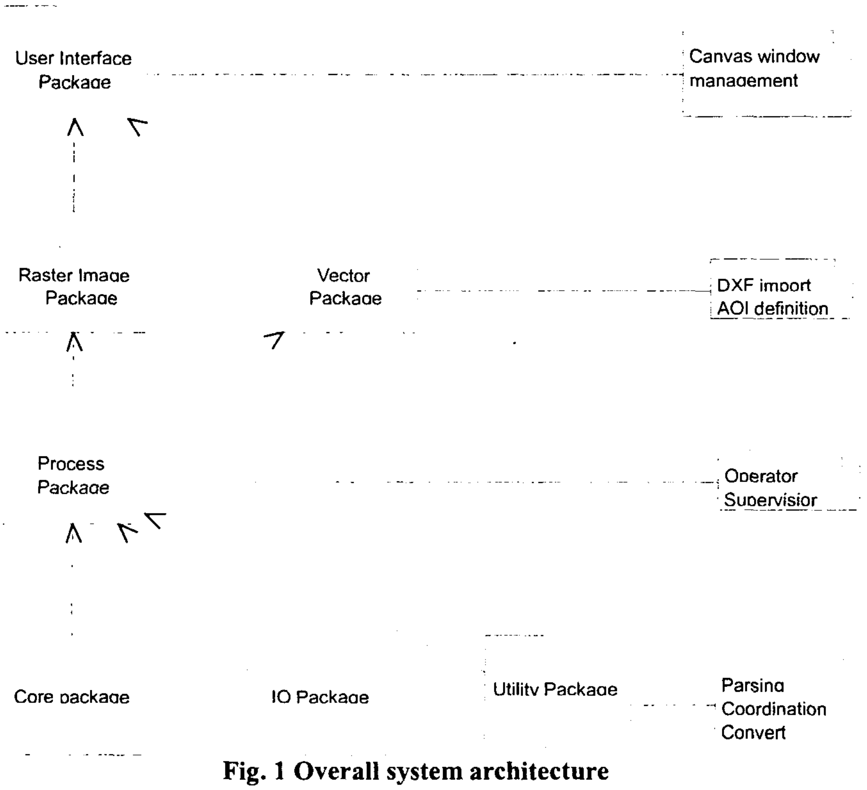


Fig. 1 Overall system architecture

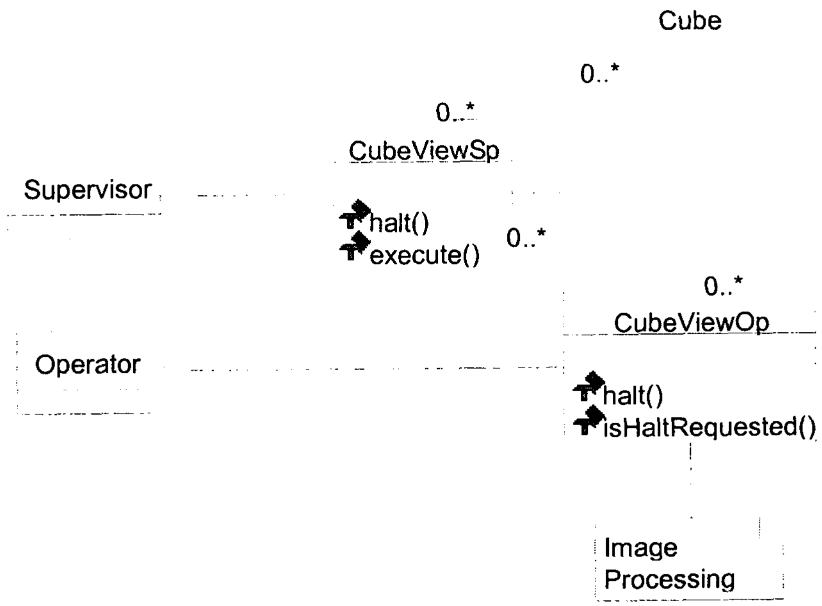


Fig. 2 Relationship between Supervisor and Operator class

Supervisor object controls Operator object that carries out image processing. Besides it takes a role in connecting graphical user interface and Operator object. It displays GUI that related to Operator object and generates additional thread for itself. Created thread generates Operator object and makes it carry out image processing. Derived classes from Supervisor class implements Halt and Execute method. Halt method manipulates interrupt exception and executes method creating Operator object.

3. Implementation

GeoNet is developed under JDK 1.1.7B and pure 100% Java code. Under JDK 1.2 it does not work due to the internal bug of JDK 1.2.

3.1 Image Display

Satellite image are implemented by Cube ,CubeFile and CubeMem class. Cube class is a super class of CubeFile class that stands for file image as CubeMem class that represents memory image.

CubeFile class is a pointer to file image and CubeMem class is address of memory. When entire image is loaded into memory, system is required to have so much memory. With this reason GeoNet uses CubeFile class in usual case and uses CubeMem class in case users need to load all images into memory. It is more efficient to use memory image when it need rapid processing. All of command in GeoNet can be applied to both images.

In addition to that GeoNet adopted BIL(Band Interleaved by Line) image format as default format to accelerate speed of processing large-sized image.

3.2 Image Management

In GeoNet, ImageRGBManager class and ImageBWManager class do image management. ImageRGBManager class manages display and image in color model and ImageBWManager class does in black/white model.

ImageLoader class carries out loading Cube object into memory. ImageLoader class checks the type of Cube object if it is file or memory type images and processes it with proper IO operation. It is ImageMaker class that converts loaded image into Java Image object to display image. Figure 3 represent flow of image management.

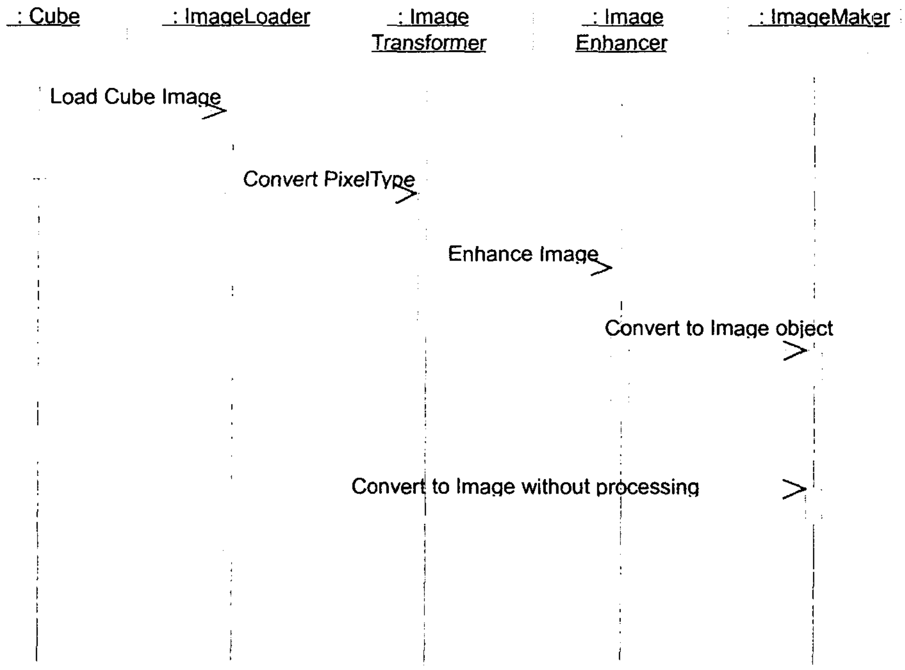


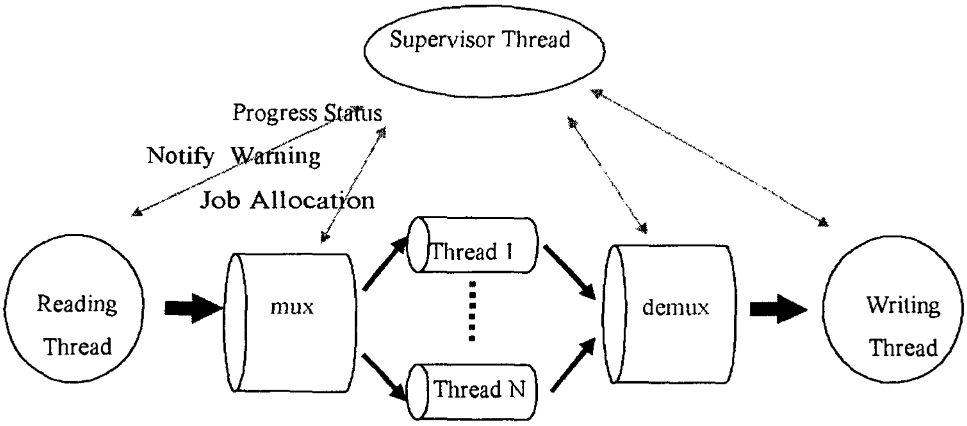
Fig. 3 Flow of image management

3.4 Multi-threaded Data Operation

Nowadays, many personal computer and workstation mount dual CPU and can do more. It seems these multi-CPU computers are becoming popular. But, previous programming language like C, C++ does not support multi-threaded programming more easily for programmers to develop. For this reason, a lot of remote sensing software has not made a best use of multi-processing environment. This has lead to 10% to 20% of utilization rate under multi-process platform. GeoNet have a structure for processing that combining several threads to carry out one procedure. It makes best use of CPU utilization (Figure 5).

For example, like Figure 4 Reading thread reads input file image and transfers it to Operation thread. Operation thread processes an image and Writing thread write a result. Absolutely, Operation thread could be separated into sub thread according to attribute of operation.

Each thread should report to Supervisor thread about operation states periodically or in case of special event. Supervisor thread monitors each thread and halts the thread in case of problem. After that it notify users. The



Thread for allocated processing

Fig. 4 Thread architecture of GeoNet

thread that terminate its own job is disposed in memory on stopping and the thread waiting for data input, not starting operation, could be loaded to memory when Supervisor thread instruct it to enter into the operation. This has memory of thread minimize although they are divided with many threads.

4 Prototype and Experiments

4.1 Performance testing with threaded architecture

This is experiment on performance of multi-threaded remotely sensed image processing operation. To evaluate we compare with single-threaded and multi-threaded sobel edge detection processing with diverse image size and CPU number. As Figure looks, it seems have no difference to CPU number in single-threaded processing.

On the other hand, in multi threaded architecture one CPU case has somewhat lower performance due to communication between threads compared to sequential architecture. But performance increase as the number of CPU is added up especially in large image data. We will see that it is 3.7 times efficient than sequential architecture in 5 CPU.

As this result, we see that prevailing multi CPU architecture and a trend of large-scale satellite image will need the system with multi-threaded architecture to operate easily and efficiently. Figure 5 shows result of performance testing.

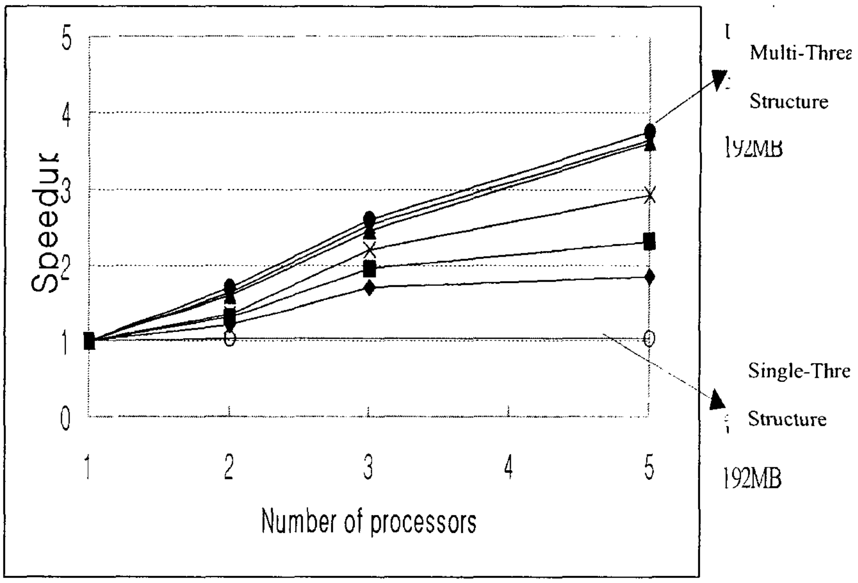


Fig. 5 Result of performance testing

4.2 Web-based processing

GeoNet is designed to execute on applet form in web browser. Java plug-in is downloaded automatically when user accesses the web page, which makes GeoNet is executed in same situation. Permitted users can use the system in a way of certification.

With the certification, users could operate processing in anywhere, anytime by internet. Figure 6 shows capture screen when GeoNet is executed on Internet. Site is http://rsnt1.etri.re.kr/geonet/index.html.

GeoNet applet could obtain information of local system by validating the certificate for execution. Client installs plug-in and requests server for certificate. Server side checks if it is valid user and transfers the certificate to the client to have certificate import for system to access local resources.

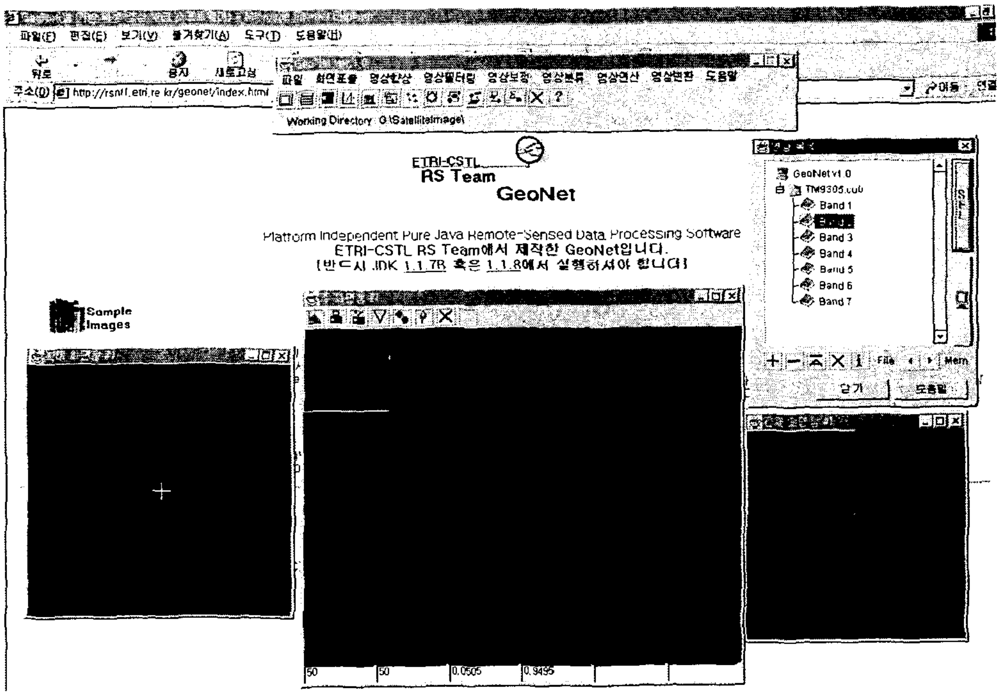
If the user accesses the site having applet by web browser after this process, GeoNet applet checked if it is valid local system and could access to local resources to get information needed. Finally applet executes in valid user's local system.

5 Further Study

GeoNet need certificate in order to execute applet under web browser. But certify process is so complex and tedious that it is better to introduce agent concept. Certification agent should carry out all process of certification and transferring certificate.

Building web-based system is a part of Java network functional advantage. It could be fully achieved when image object is transferred and processed in distributed environment.

Distributed remotely sensed image processing system should be studied to make a ultimate utilization of resources.



Platform Independent Pure Java Remote-Sensed Data Processing Software
ETRI-CSTL RS Team에서 제작한 GeoNet입니다.
[반드시 JDK 1.1.7B 혹은 1.1.8에서 실행하셔야 합니다]

Fig. 6 Snapshot of GeoNet with web browser

5 Conclusion

GeoNet have advantage like this.

(1) Interface for cross-platform large-sized remote sensed image processing API
(2) Minimize period of developing with Java object-oriented paradigm
(3) Client/Server remote sensed image processing using Java network extensibility with component technology like Java RMI, JavaBeans
(4) Flexible system architecture

Development of remote sensing image processing system, GeoNet, with Java language suggest alternative to meet rapidly the demand of image processing in proportion to increment of image provision and distributed environment.

References

1. Rodney A. Brooks, "Symbolic Reasoning Among 3-D Models and 2-D Images", Artificial Intelligence 17, pp. 285-348, 1981
2. Randy Crane, "Simplified approach to Image Processing", Prentice-Hall. 1997
3. Takashi Matsuyama, "Expert Systems for Image Processing: Knowledge-Based Composition of Image Analysis Processes", CVGIP 48, pp. 22-49, 1989
4. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing,", Addison-Welsey.1992
5. John A. Richards, "Remote Sensing Digital Image Analysis,", Springer-Verlag, 1992
6. John R. Schott, "Remote Sensing, The Image Chain Approach", Oxford University Press, 1997
7. Arthur Cracknell and Ladson Hayes, "Introduction to Remote Sensing", Taylor & Francis, 1993