A Java-based Tool for Image Analysis over Distributed Networks

Carlos Huallparimachi Suarez
Advisor: Domingo Rodríguez Rodríguez

Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez Campus
Mayagüez, Puerto Rico 00681-9042
carloshs@ece.uprm.edu

Abstract

This paper describes the design and development of a Java-based distributed system tool-environment for image analysis with special attention given to synthetic aperture radar (SAR) imaging applications. This tool-environment allows a given user to effect important image processing functions such as to visualize, manipulate, improve, filter, detect edges, reduce the noise on SAR images, as well as operations on SAR raw data. Also, this system has the unique option of allowing end-users to add their own customized algorithms as encapsulated operators to act on elements resident on local or remote SAR images-servers on a computer network.

1. Introduction

Image processing is used to analyze a digital image or to transform it into a new image. Different techniques for the manipulation, correction, and enhancement of digital images have been used for years. One of the main objectives of these techniques has been the removal of defects from images obtained through a wide variety of equipments. In general, these images can be obtained from a diverse set of instruments such as a low resolution digital camera to an expensive high resolution medical imaging scanner, or by sensors mounted on satellites or airplanes. Regardless of how an image is acquired, image processing applications typically provide three basic functions for handling images: loading, rendering, and manipulating [Rodrigues01].

The processing and analysis of images require a great amount of computational power. At the present time it is very important to develop tools that allow us to make our work with image processing faster and more efficient. Also, large amount of diverse data is now available, such as very high resolution images and images from spectrometers. Some expert and data fusion systems offer a facility to integrate various data through various approaches and techniques. The results of such approaches allow access to diverse data (vegetation, disturbed areas, burned zones, mixed zones)[Voirin02].

The internet has been the breeding ground for many exciting new technologies in the recent years; one of the most important of these technologies is the Java programming language [SUN99]. Java is a language that seems particularly well-suited for the development of software in network environments. Within a short period of time, server-side computing has already embraced Java. Even though the imaging industry has been slow to react to Java, many industry leaders have realized its merits and are moving towards this it.

In the sections that follow we provide a step by step description of the work presented in this article. We start with a basic introduction to synthetic aperture radars. Next, comes a description of the developed environment, named JSIM. We talk about the prototype itself,

the network architecture behind it, its special features and how it relates to its special operators, and metadata. Finally, we provide some conclusions and future works.

2. Synthetic Aperture Radar Systems

Synthetic aperture radar are sensor-based system mounted on space borne or airborne platforms which have as main function the active, low-power, microwave illumination of the Earth's surface in order to obtain imaging information for geosciences applications. Its active sensor system nature allows SAR to work all the time, day and night, under any climatic condition and to obtain images of higher spatial resolution than those generated with conventional instruments. These characteristics make SAR a very attractive system in remote sensing applications[Rueda99].

The synthetic aperture radar (SAR) technology, in general, has been one of the most exciting and progressive fields in remote sensing in the last years. The SAR images generated by the SAR systems can be used to solve many problems in land and oceanographic remote sensing. The SAR technology has resulted in marked improvements in spatial resolution imaging operations when observing a ground scene from aircrafts or satellites and it can be used to estimate also parameters and features such as the dampness of soils, wetlands, fluvial structures, metallic surface structures, thickness of the forests, or the roughness of the sea.

3. JSIM Prototype

JSIM is an acronym for Java-based SAR image analysis tool environment. JSIM was designed and developed using the Java programming language. Considering advances in information technologies, we envisioned to offer an open-source application, platform independent tool environment, for analyzing and manipulating SAR images. The resulting application allows access to local and remote image-data in a variety of file formats.
The tool-environment presented here provides friendly user interfaces, where it is possible to search, manipulate and view images using different image analysis operators such as *filtering, detection edges, enhancement, convolution, correlation, FFT* (Fourier Fast Transform), as well as basic operators such as *load, save, zoom in, zoom out, scale, rotate and flip*. It allows also multiple files to be opened simultaneously for both viewing and editing.

We are also providing this java-based image analysis tool environment a degree of portability permitted in a given workstation network configuration. The tool runs on various platforms and can be used through the Internet. The Figure 1 below shows the main graphics user interface of JSIM.

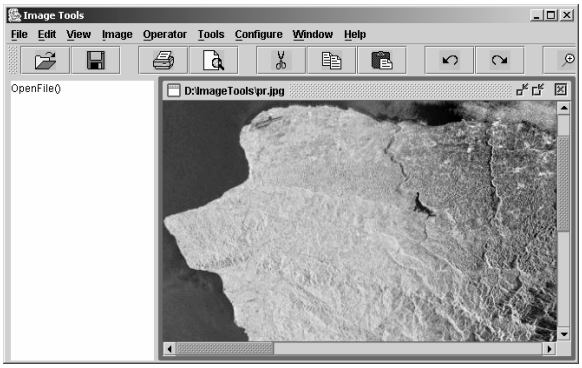

Figure 1. Graphics User Interface of JSIM.

4. JSIM Network Architecture

One of the important advantages of Java is its network friendliness. The Java core contains many features that help us develop network-based applications. The Remote Method Invocation (RMI) is a distributed-computing architecture. It enables communication between Java applications. With the RMI architecture, an object in one application can invoke a method in another application running on a different Java Virtual Machine.

In This work we use the client-server paradigm. Internet-based applications are built in this way, as are many high-end applications [Kim02]. With the client-server approach, multiple clients from a local or a remote machine can access the same application at any time. The clients have the GUI to use the operators provided in this

tool. Figure 2 shows the architecture of the application to be used through Internet.

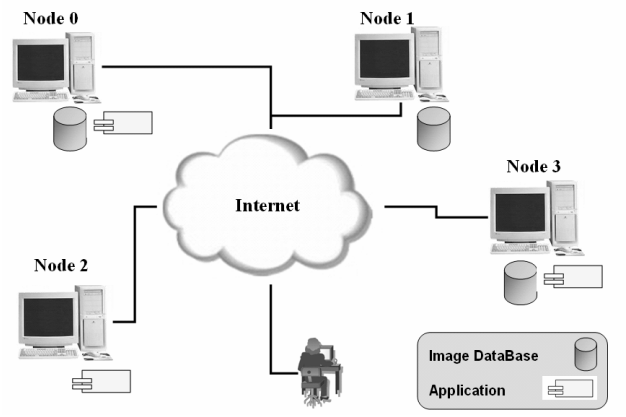

Figure 2. JSIM Architecture.

The architecture of a distributed system offers a wide variety of applications as well as challenges to trying sharing information efficiently over a network of computers.

The distributed systems (DS) is a collection of heterogeneous computers and processors connected via a network to increase the computational power. This collection works closely together to accomplish a common task. A DS consist of a set of processes, executing in different platforms, which communicate via message passing.

5. History List and Operator Encapsulation

The creation of new operators is possible from existing operators which have been initially defined as default. The new operators are encapsulated as a sequential command. The advantage and value added of this option is the economy of time, when reusing a set of operators.

The steps that should be taken to create an operator encapsulated starting from a set of the operators by default are the following: the user should use the operators of the system according to his necessities or the results that he hopes to obtain when combining a set of operators. At the same time, the system will proceed to create a list of the used operators, once the prospective results are obtained the user can store the sequence of operators used as a new operator for his later use. Figure 3 depicts the manner how a set of encapsulated operators are used as a new operator to act on a target image.

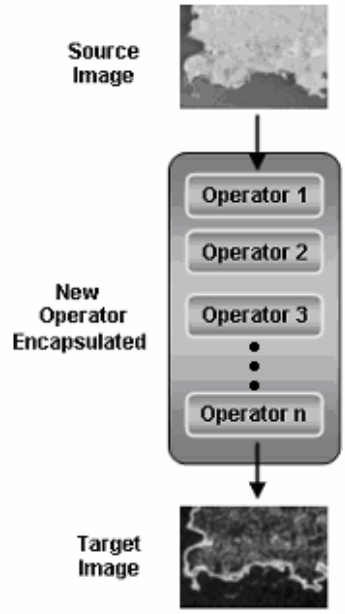

Figure 3. Set of Operators to Encapsulate.

6. New Operators Customized

The broad field of image processing involves the manipulation and analysis of visual data; fundamental mathematical procedures (known as algorithms) are applied to the data, enabling them to be processed by computers or hardware designed for this purpose. New algorithms are constantly being developed for procedures such as image encoding, compression, feature extraction, image analysis, and spatial filtering.

JSIM allows writing new operator agents (algorithms), implementing new Java classes that are integrated to the existing tool. End-users can add their own customize operations to be used with the SAR images.

One of the many problems that object-oriented software development is meant to solve is the difficulty with creating dynamic applications that change their behavior at runtime. We use design patterns to solve this problem. The factory method pattern is a model that allows us to dynamically add classes without modifying the program. This pattern helps address this

problem of "dynamic creation" by providing a design solution that allows the application to create different objects based on the current system state. Figure 4, shows the design of the pattern to add new operators.

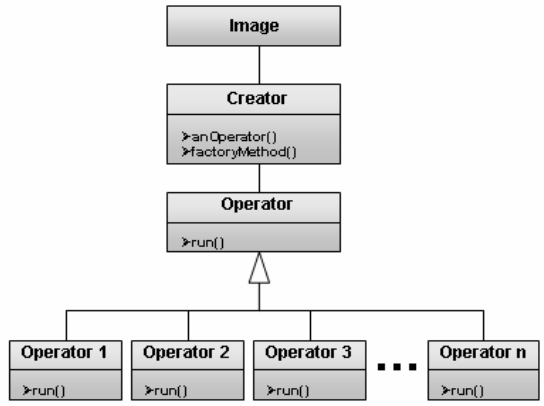

Figure 4. Methodology Pattern for Operator Addition.

7. Metadata

Metadata can be defined literally as "data about data," but the term is normally understood to mean structured data about digital and non-digital resources that can be used to help support a wide range of operations. These might include, for example, resource description and discovery, the information about data management and preservation and as well as continuing access.

We will use the metadata concept since it will help to associate complementary information to SAR images. This information helps to have full advanced knowledge of data characteristics and attributes. Also, the metadata will help the end user to organize, find, identify, select, obtain and interpret a SAR image. The metadata will be stored as records in a database. To share this information we will use XML due to the flexibility that it offers and its platform-independent syntax.

Each SAR image has its own metadata and can include information such as the location of the image, date, resolution, latitude, longitude, format, source, coverage, description, type and other information related. This option will be enabled in local mode only. The end users could manipulate metadata if they so desire.

8. Conclusions and Future Works

In this work we have presented a tool environment for image analysis of synthetic aperture radar data. The tool environment was designed and development using Java programming language. We have proposed to use The Java™ Technology to guarantee protocol and platform independence; thus, our basic prototype runs over a heterogeneous environment without inconveniences at the present time.

As a future work, we expect to implement a set of algorithms to analyze other types of images in addition to SAR images. These algorithms will be added as operators to the JSIM network library.

References

[Rodrigues01] Lawrence H. Rodrigues, "Building Imaging Applications with Java", Addison-Wesley, USA, 2001.

[SUN99] Java Advanced Imaging JAI, SUN Microsystems, http://java.sun.com/products/java-media/jai, USA, November 1999.

[Voirin02] Yves Voirin, Goze B. Benie, Dong Chen He, Ko Fung, and Kalifa Goita, "A Forest Map Updating Expert System based on the Integration of Low Level Image Analysis and Photointerpretation Techniques", 2002 IEEE International Geoscience and Remote Sensing Symposium and the 24th Canadian Symposium on Remote Sensing, Toronto, Canada, June 2002.

[Kim02] Do-Hyun Kim, and Min-Soo Kim, "Web GIS Service Component Based On Open Environment", 2002 IEEE International Geoscience and Remote Sensing Symposium and the 24th Canadian Symposium on Remote Sensing, Toronto, Canada, June 2002.

[Rueda99] Dilia Beatriz Rueda, Domingo Rodriguez, "A Sar Algorithm Development Environment Using Matlab", CRC 99 Computing Research Conference, Mayaguez, Puerto Rico, December 1999.