

Java-based Tool for Synthetic Aperture Radar Image Analysis

Carlos Huallparimachi

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

Domingo Rodriguez

Electrical and Computer Engineering
University of Puerto Rico, Mayagüez Campus
Mayagüez, Puerto Rico 00681-9042
domingo@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, as well as to carry out raw data formatting, image formation operations, Kronecker array filtering, and feature extraction on SAR images. 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. [1].

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) [2][3].

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 [4]. 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.

Many commercial products are available to support image processing specialists. These products have many features for loading, manipulating, and rendering images. Companies often conduct research works in image processing to develop new technologies that they commercialize in the software market. The resulting software products tend to be rather expensive for academic settings worldwide, or are usually not widely accessible to most interested users throughout the developing world. This work seeks to contribute to the free availability of image analysis tools for academic purposes.

In the sections that follow we provide a step by step description of the work presented in this article. We start with a description of previous research work followed by a formulation of the main problem addressed in the article and the motivation behind it. We proceed with a basic introduction to synthetic aperture radars. Next, we present 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, template class, servlet model, and metadata. Finally, we provide some conclusions and future works.

2. Previous Research Work

In this section we present previous research works conducted in areas related to our work. Don Murray, Bill

This work was partially supported by NSF Grant No. 99-77071

Hibbard, Tom Whittaker, and James Kelly [5] have developed a tool for visualizing and analyzing remotely sensed data, and examples of image display and analysis tools. They used library VisAD for their implementation. VisAD is an open-source Java library for building interactive and collaborative visualization and analysis tools.

Seng Chuan Tay and Kim Hwa Lim [6] have developed an application for Internet, Web-based processing for remotely sensed data. They adopted the client-server processing technology to process the NOAA data in a distributed manner, thereby reducing the program elapsed time.

The Unidata Program Center offers software and services that enable universities to acquire and use atmospheric and related data on their own computers, usually in real time. Unidata's MetApps is a project to develop Java applications for analyzing and visualizing meteorological and related geosciences data, including remotely sensed data such as satellite and radar data [7].

3. Problem Formulation and Motivation

With current advanced data acquisition and imaging technologies, it is now possible to acquire SAR images in many different ways, from sensors mounted on satellites and/or airplanes platforms. In the recent years there has been a large production of SAR images, which are used by diverse studies in different areas of science and engineering.

The SAR technology allows obtaining terrestrial images of high spatial resolution, even through cloudy skies, to help us improve the comprehension of environmental phenomena in the geosciences. It is necessary to have modern automated tools to manipulate these images and develop new applications.

This stated necessity generates a justifiable motivation for the development of a new set of java-based image analysis tools that will contribute in an integrated manner to the difficult problem of distributed automated information processing.

The researchers and/or students need tools that can be used in their computers or through the Internet. These applications must process the image data available at Internet in an efficient manner, offering modularity, scalability, and robustness. Other special feature might be the availability of operator agents (algorithms), which could be developed for the users of the tool environment in order to conduct studies over several types of images.

4. Synthetic Aperture Radar System

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.

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.

5. 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 also allows 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.

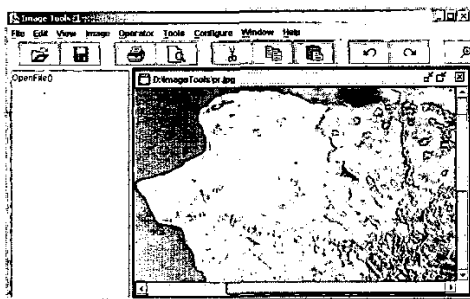


Figure 1. Graphics User Interface of JSIM.

The next table 1 shows the features of JSIM and same time we made a comparison with ImageJ.

Table 1. Features of JSIM and ImageJ

Description	ImageJ	JSIM
Runs multiple platforms		
Linux, Solaris, Mac, Windows	Yes	Yes
Data Types		
8 bit grayscale	Yes	Yes
16 bit unsigned integer	Yes	No
32 bit RGB	Yes	No
File Formats		
TIFF, GIF, JPEG, BMP, ASCII	Yes	Yes
DICOM, FITS	Yes	No
Add new components		
Pluggings	Yes	No
Operators	No	Yes
Image Enhancement		
Smoothing, Edge detection, filtering	Yes	Yes
Regions of Interest		
Rectangular	Yes	Yes
Elliptical, irregular regions, transfer an ROI	Yes	No
Geometric Operations		
Crop, scale, resize, rotate	Yes	Yes
Flip vertically/horizontally	Yes	Yes
Analysis		
Measure Area, Mean, Min and Max	Yes	Yes
Histograms	Yes	Yes
Measurements units	Yes	No
Mode of operation		
Local, Internet	Yes	Yes
Distributed task		
over different computers	No	Yes
Image Display		
Zoom, scrolling	Yes	Yes
Editing		
Cut, copy, undo, paste	Yes	Yes
Add text, arrows, rectangles, ellipses, polygons	Yes	No
Metadata		
Local Data	No	Yes
Remote Data	No	No

6. 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 [4]. 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.

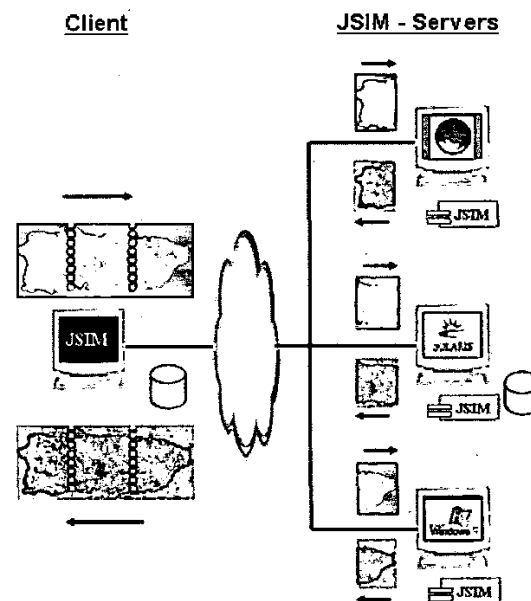


Figure 2. JSIM Architecture.

The components of JSIM are:

1. **JSIM - Server:** set of Java-servlet application.
2. **Client Access:** interacts with user application.
3. **Image Database:** contains the SAR images, these can be distributed on the network.
4. **Operators Agents:** algorithms to act on SAR images.

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. Table 2 presents a summary the type of flexibility that JSIM offers for system configuration. The letter L denotes local and R denotes remote.

Table 2. Properties of JSIM

Description	Application		Data		JSIM Interface	
	L	R	L	R	Lan	Wireless
Fast computation	X		X		X	
High distributed	X	X	X	X	X	X
Security	X		X		X	
Scalability	X	X			X	X
Complexity		X		X	X	X
Probability to failure		X		X		X
Easy to used	X		X	X	X	X
More Setting		X			X	X
Updates	X				X	

7. Operator Agent Environment

The JSIM network architecture presents to user a unique environment with a set of tools for the proper manipulation of object image data in a general setting. We term this environment an *operator agent environment*. Formally, a JSIM operator agent environment is defined as an aggregate of the following components: An object image input data set, an object image output data set, a set of operator agents, a set of composition rules for the operators agents, a set of action rules for the operators agents to act on the object image input data set in order to produce an object image output data set, and a user interface. Figure 3 depicts an abstraction of the interrelation of the components of an operator agent environment in JSIM network architecture.

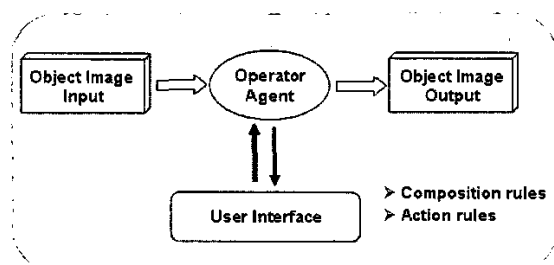


Figure 3. Operator Agent Environment.

8. 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 adding of this option is the economy of time, when reusing a set of operators.

The steps that should be taken to create an encapsulated operator, 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 its later use. Figure 4 depicts the manner of how a set of encapsulated operators are used as a new operator to act on a target image.

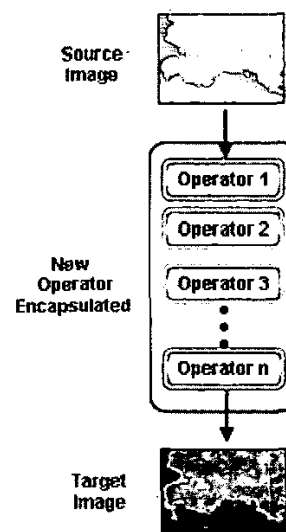


Figure 4. Set of Operators to Encapsulate.

9. 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 found when 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 5, shows the design of the pattern to add new operators.

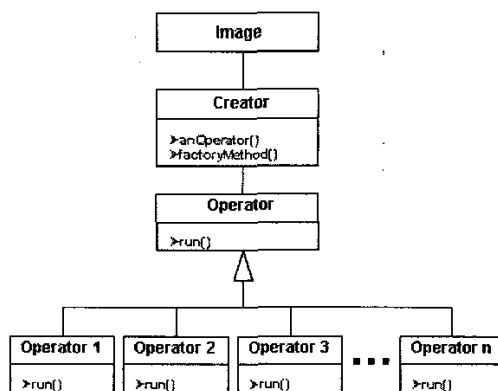


Figure 5. Methodology Pattern for Operator Addition.

10. Template class operator design

For the unique option or feature of adding new operators to the tool-environment, a template class was designed as shown in Figure 8. Using this template class the users can create their own algorithms to analyze SAR images. The new operators can be added to the system as part of the library of operators. When the users want to use their operators the system will invoke them dynamically once they have been compiled and attached to the system. The system provides for the interchange of operators among users.

This template class receives an image that has been loaded by the program and transformed into a matrix. Users then should proceed to add their developed algorithms. In this manner, it is simpler for a user to add new algorithms to the system without worrying about additional details.

11. JSIM Servlet Model

The use JSIM through the Internet will be through the technology of the servlets provided by Java. A servlet is a Java program that runs on the server side and is considered an extension of the Web server [1]. The advantage of using servlets is that the client Web browser does not need to support the version of Java or the extensions that a servlet requires.

The model that we propose to use for JSIM encompasses a server which has the application with a set of basic operators to analyze an image, with an option to use operators provided by the users, and the clients that have SAR images and their own operators. The process to use JSIM is structured in four fundamental steps. The first step is the initial state where the server is awaiting orders to process and the client has SAR images that a user wants to analyze. In the second step, the client makes a request to process a SAR image and the client sends the image and the operator (optional). In the third step, the program processes the request in the server. Once finished, in the last step, the result is sent to the client. Figure 6 shows a scenario of how the application works using servlets.

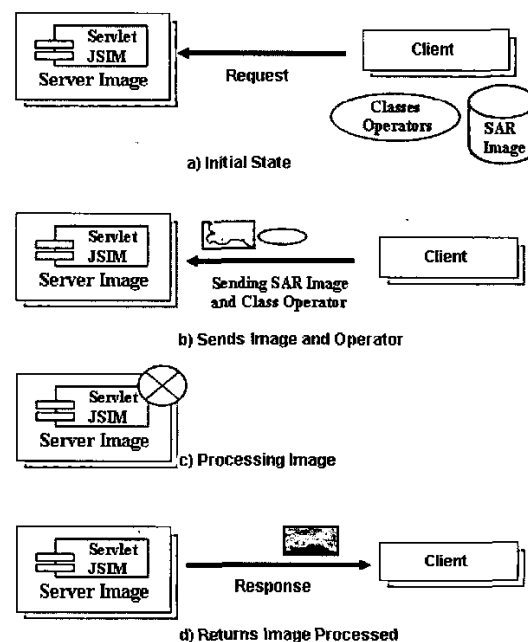


Figure 6. JSIM Using Servlets.

12. 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.

13. 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. We believe that this tool will assist scientists and engineers in their SAR image analysis efforts.

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. A usability test will be conducted on the developed tool to test ease of use by scientists and engineers. JSIM will be used to assist in a comprehensive study of wetland regions in the Greater Antilles. The tool will be evaluated for wireless applications, and a website will be created and maintained for JSIM source code downloads and related issues.

References

- [1] Lawrence H. Rodrigues, Building Imaging Applications with Java, Addison-Wesley, USA, 2001.
- [2] 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.
- [3] Hellwich O, Heipke C. Wessel B., Sensor and data fusion contest: information for mapping from airborne SAR and optical imagery, Geoscience and Remote Sensing Symposium, IGARSS 01, 2001.
- [4] Java Advanced Imaging JAI, SUN Microsystems, <http://java.sun.com/products/java-media/jai>, USA, November 1999.
- [5] Don Murray, Bill Hibbard, Tom Whittaker, and James Kelly, Using VisAD to Build Tools for Visualizing and Analyzing Remotely Sensed Data, IEEE 2001 International Geoscience and Remote Sensing Symposium, Sydney, Australia, July 2001.
- [6] Seng Chuan, Tay and Kim Hwa, Lim, Web-Based Processing for Remotely Sensed Data, 2002 IEEE International Geoscience and Remote Sensing Symposium and the 24th Canadian Symposium on Remote Sensing, Toronto, Canada, June 2002.
- [7] Don Murray, Russ Rew, Steven Emmerson, Jhon Caron, Stuart Wier, and David Fulker, Unidata's MetApps Project New ways, in Java, of Exploring Remotely Sensed Data, IEEE 2001 International Geoscience and Remote Sensing Symposium, Sydney, Australia, July 2001.
- [8] 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.
- [9] Environmental Systems Research Institute ESRI, GIS and mapping software, <http://www.esri.com>, USA, 2002.
- [10] PCI Geomatics, <http://www.pcigeomatics.com>, Canada, 2002.
- [11] Xu Jiang, Zhu Minhui, Wu Yirong, Peng Hailiang, Parallel programming in SAR image processing, Geoscience and Remote Sensing Symposium 1999 IGARSS 99, Proceedings I 1999 International, 1999.
- [12] H. Izumi, K. Sasaki, K. Nakajima, H. Sato, An efficient technique for corner-turn in SAR image reconstruction to improving cache access, Parallel and Distributed Processing Symposium, Proceeding International, IPD, 2002.
- [13] Li Wang, Deng-Che He, A. Fabbri, Textural filtering for SAR image processing, Geoscience and Remote Sensing, IEEE Transactions, 2000.
- [14] D. Paudyal, A. Eiumnoh, J. Aschbacher, Textural information in SAR images for land-cover applications, Geoscience and Remote Sensing Symposium, 1995 IGARSS 95. Quantitative Sensing for Science and Applications, July 1995.
- [15] J. Le Caillec, R. Garello, B. Chapron, Detection of nonlinearity in sea surface SAR imaging process using bispectrum method estimation, Acoustics, Speech, and Signal Processing, 1995. ICASSP-95, May 1995.
- [16] R. Fjortoft, A. Lopes, P. Marthon, E. Cubero-Castan, Different approaches to multiedge detection in SAR images, Geoscience and Remote Sensing, 1997. IGARSS 97. Remote Sensing A Science Vision for Sustainable Development, 1997.
- [17] Zhang Jun, Cheng Xueguang, Liu Jian, A speckle reduction algorithm by soft-thresholding based on wavelet filters for SAR images, Signal Processing Proceedings, ICSP 98, 1998.
- [18] D.D. Giusto, G. Vernazza, On the use of virtual information in SAR image analysis, Multidimensional Signal Processing Workshop, Sep 1989.
- [19] P. Bolon, J. Chanussot, I. Issa, P. Lambert, Comparison of prefiltering operators for road network extraction in SAR images Image Processing 1999, ICIP 99. International Conference, 1999.