Fourth LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCET'2006)
"Breaking Frontiers and Barriers in Engineering: Education, Research and Practice"
21-23 June 2006, Mayagüez, Puerto Rico.

Web-based Data Processing for Hydro-Ecological Applications

Lola Xiomara Bautista Rozo
MSc Student in Computer Engineering
University of Puerto Rico at Mayagüez Campus
Mayagüez, Puerto Rico
Lola.Bautista@ece.uprm.edu

Domingo Rodríguez
Professor at Department of Electrical and Computer Engineering
University of Puerto Rico at Mayagüez Campus
Mayagüez, Puerto Rico
Domingo@ece.uprm.edu

Abstract

In this work we describe the design and devolpment of a Java-based environment, called J-CID (Java Computational Image Developer), for the treatment of remote sensing data information. The treatment of the data images is performed through the use of image operators and other image processing functions. An image operator takes an image and performs certain tasks such as noise reduction, edge detection, feature enhancement, spectral analysis, format translation, etc. Especial attention is given to the development of a computational signal algebra framework for the modeling of digital image interferometry operations for environmental surveillance monitoring, using correlation techniques to process the phase information of a digital image.

For the implementation of some image processing algorithms, an interface between Java and MATLAB® was used in order to take advantage of the existing operators of the Image Processing Toolbox of MATLAB®. Java-based image processing operators were built using the Java Advanced Imaging API (Application Programming Interface), which takes advantage of supporting different types of format files, be platform-independent, and work over a network architecture.

This system has the option of allowing end-users to add their own customized algorithms as encapsulated operators. The distributed access property of the application is given by the use of web services technology, which provides interoperability between various software applications running on diverse platforms, allowing the reuse of services and components.

Keywords
Image Processing, Web-Based Processing, Web Application Services, Signal Algebra Operators.

1. Introduction

Environmental management has become one of the issues considered mos important by concerned governmental agencies throughout many countries since its impact on their economic development,

public health, and national security has benn amply documented. Environmental surveillance and monitoring is an important activity in environmental management. It deals with the gathering and processing of appropiate environmental information to aid in the process of effective decision making.

Among the techniques used to obtain information about the environment is the remote sensing, which consists in the utilization of a device or sensor at a relative distance without direct contact with the object of interest. Most of the times the devices are transported aboard of an aircraft , spacecraft or satellite. In this work we are interested in the images acquired by Synthetic Aperture Radar (SAR), because it has the capability of acquire images in extremely weather conditions at day or night. A SAR image represents the back-scattered radiation from the target area. Some of the most known SAR instruments are ERS-1 and ERS-2 of the European Space Agency, JERS-1 of the National Space Development Agency of Japan, RADARSAT-1 of the Canadian Space Agency and SIR-C/X-SAR of the United States, German and Italian space agencies (Rosen, 2000).

In this project we are also investigating the use of SAR interferometry, since it is an imaging technique to detect changes over time of a determined surface, which permits to monitor geographic processes, such as sedimentation, floodplains, earthquakes, landslides, etc. An interferogram is obtained by averaging the product of one complex SAR image and the complex conjugate of another image.

Due to the discrete nature of digital images, it makes necessary the use of Digital Image Processing techniques to manipulate the images. In this work we are taking advantage of the Image Processing Toolbox of MATLAB® (Kennedy, 2005) and the Java Advanced Imaging (JAI) API of Java to implement the basic image operators such as noise reduction, edge detection, feature enhancement, spectral analysis, format translation, etc., and advanced image operators such as cyclic convolution, cyclic correlation, Hadamard product, shift operator and conjugate operation to obtain the interferogram of two images.

Generally, this type of processing is done by stand-alone applications dedicated for a specific purpose, but, with the increasing of available information related to geophysical studies, important agencies and research groups of educative institutions have decided to storage their data in distributed databases available for scientists working in that types of applications. Thanks to the use of the service-oriented architectures, we considered the implementation of web services to allowing easy accessibility to heterogeneous applications, such as Java, MATLAB® and Geographical Information Systems (GIS) over the Internet.

Next sections are dedicated to make a deep explanation about some of the mathematical considerations that were used to build the environment as in section 2, the architecture of the proposed environment in Section 3, the description of how the environment has been implemented in Section 4 and the presentation of some results is in Section 4.

2. Mathematical Methods

J.1 **Image Representation Fundamentals**

The primary data objects used by J-CID are the set of signals of two dimensions conformed by all the functions of the form:

$$S: \mathbb{Z}_{No} \times \mathbb{Z}_{N1} \rightarrow \mathbb{C}$$
$$(n_o , n_1) \mapsto S[(n_o , n_1)]$$

This notation means that the domain of the function is the set of arrays of N_0 rows by N_1 columns, and the co-domain is the set of complex numbers of the form $a+bi$, where a is the real number and b is the imaginary number (Rodriguez, 2002).

In the J-CID environment, image pixels are grabbed into two-dimensional arrays structures, where an array is used to store the real part of a pixel and another array is used to store the imaginary part. Graphically, this can be seen in the following example (Figure 1): The complex input image of size 3 x 3 is splitte in two arrays, the real and the imaginary. Practical applications deal with images of the order of 500 x 500 and larger.

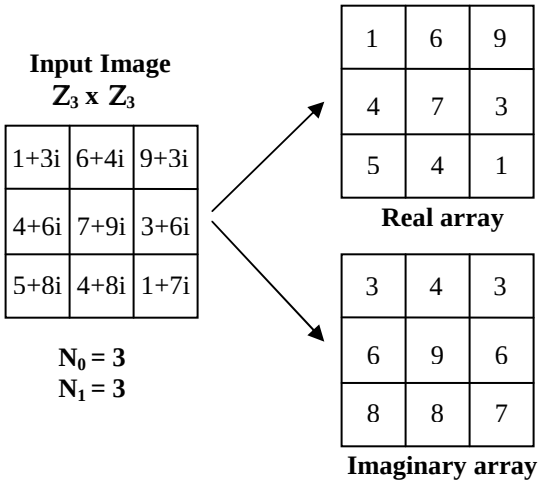

Figure 1: Example of Array Structures for Input Images

From that two arrays is possible obtain the magnitude and phase of each pixel of the image. Magnitude of a complex number, for example $a+bi$ is calculated as $\sqrt{\boldsymbol{a}^2+\boldsymbol{b}^2}$ and phase as $\tan^{-1}\left(\frac{\boldsymbol{b}}{\boldsymbol{a}}\right)$. Then, in the case of the arrays of the example, call $\boldsymbol{R_1}$ the array of real numbers and $\boldsymbol{I_1}$ the array of imaginary numbers of the image $\boldsymbol{G_1}$. If we want to calculate the magnitude of the pixel in the position [0,0], we can do $\sqrt{(\boldsymbol{R}_1[0,0])^2+(\boldsymbol{I}_1[0,0])^2}$ and the phase is $\tan^{-1}\left(\frac{\boldsymbol{I}_1[0,0]}{\boldsymbol{R}_1[0,0]}\right)$.

2.2 Image Interferometry

Interferometry is an imaging technique to detect changes over time of a determined surface, which permits to monitor geographic processes, such as sedimentation, floodplains, earthquakes, landslides, etc. An interferogram is obtained by averaging the product of one complex SAR image and the complex conjugate of another image (Luzi, 2004). The phase of each image pixel in an interferogram is the difference of the phase of the corresponding pixels in the two SAR images. Taking the same example, be $\boldsymbol{R_2}$ the array of real numbers and $\boldsymbol{I_2}$ the array of imaginary numbers of another image $\boldsymbol{G_2}$, it is possible to demonstrate that:

$$\boldsymbol{G}_1[0,0]\boldsymbol{G}_2^*[0,0] = \tan^{-1}\left(\frac{\boldsymbol{I}_1[0,0]}{\boldsymbol{R}_1[0,0]}\right) - \tan^{-1}\left(\frac{\boldsymbol{I}_2[0,0]}{\boldsymbol{R}_2[0,0]}\right),$$

where * denotes conjugate operation. J-CID has implemented the operation for calculate the phase of each pixel of an image, as well as the conjugate. By combining that operations could arrive to the interferogram between two images (Villamizar, 2005).

3. ARCHITECTURE PROPOSED ENVIRONMENT

The architecture presented here has two main attributes: it is internet-centric and it is service-oriented as well. Internet-centric means that the interaction between de user and the components of the system is performed through the Ethernet network, see Figure 2. Access to data stored in Raw Data Servers (RDS) is possible by a secure Java FTP protocol; those servers can be located at any location of the Ethernet network. The Information Technology Services can be also located in a remote equipment or be available in the local setting of the user. It is said that the architecture is service-oriented because the algorithms and tools for image processing are going to be available as web services, which provides interoperability between various software applications running on diverse platforms, allowing the reuse of services and components (Figure 3).

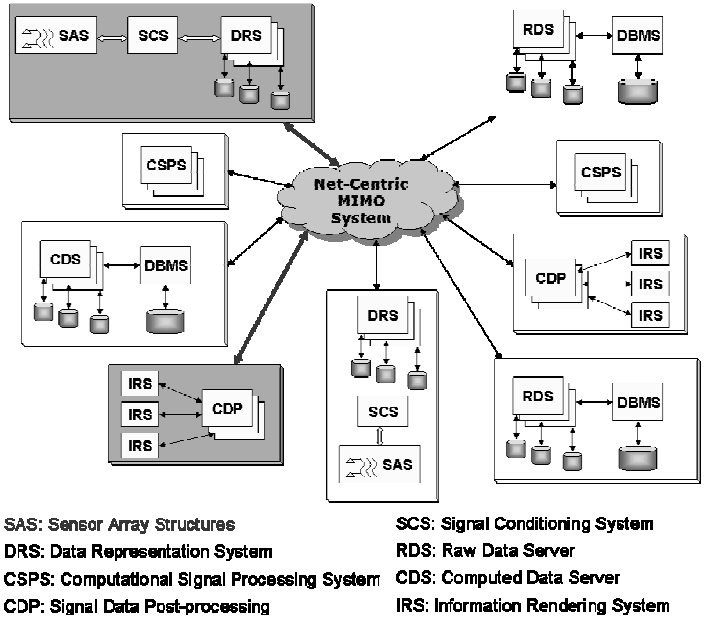

Figure 2: Computing and Information Processing over a Cyber-Infrastructure

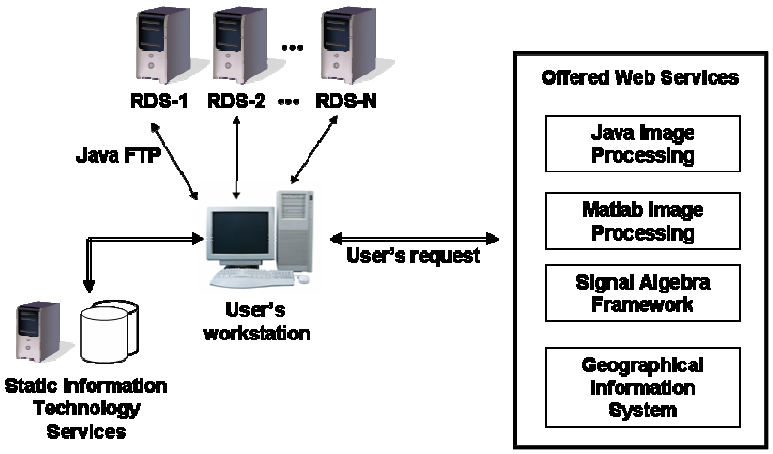

Figure 3: Local User Configuration and Available Web Services

4. IMPLEMENTATION

This section describes in a general way how J-CID has been implemented, starting with the analysis of other environments for image processing. Then appears some details of the implementation and features that J-CID has.

4.1 Related Works

Table 1 shows a comparison of attributes between J-CID and other environments for image processing that are well known in academia. What we want to show is that J-CID was thought as a tool that permits a basic level of interaction for those users that do not have much experience in this area, and in the same way, it is a tool that permits some advanced features that experienced users need. All depends on the requirements and results that users want to get.

Table 1. Comparison of Application Attributes

	J-DSP	ImageJ	J-CID
Applet	✓		
Stand-Alone Application		✓	✓
General Purpose	✓	✓	✓
Educational Purpose	✓		✓
Basic Operators for Image Processing	✓	✓	✓
Specialized Operators for Image Processing			✓
Encapsulation Operators			✓
Service Oriented Architecture			✓
Web Services Offered			✓
Middleware Interface			✓

4.2 System Configuration

The code development of J-CID has been performed with Java language, using Eclipse 3.1 as the Integrated Development Environment (IDE), which was configured with the plug-ins of J2EE (Java Enterprise Edition) technology, Tomcat, and the Visual Editor for the implementation of the user interface. The server configuration is as follows: processor Intel Xeon of 3.6 GHz, 4 GBytes of RAM memory, Linux Cent OS 4.2, JRE 1.5 of Java Sun Microsystems, Apache Server, Tomcat 5.5.16, AXIS 1.3, MATLAB® 7.2, and ArcGIS 9. User equipment only requires to have installed the JRE 1.5 and MATLAB® 7.2 as well.

4.3 Computational Implementation of the Image Processing Operators

Each one of the operators is a Java class that implements the Operator interface called Operator or the interface called BinaryOperator (Figure 4). With this abstraction, operators are classified as either unary or binary operators. Unary operators are those that operates over just one image, while binary operators are those that takes two images as operands, producing a third image.

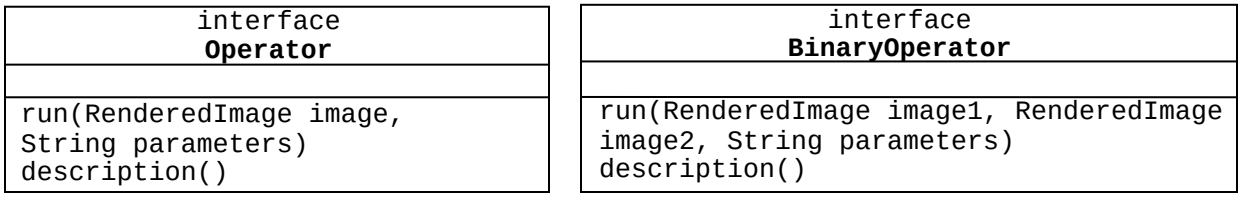

Figure 4: Abstract Interfaces for Unary and Binary Operators using UML Notation

The method called `run` for unary operators receives only one image of type RenderedImage (Java class of the JAI API), while for binary operators receives two images of the same type. Every image is converted to an array as was explained in section 2.1. Each operator has a method called `description()` where is stored the basic information of the operator that is needed for the user.

The J-CID environment can read images of some of this formats: *.bmp, *.jpeg, *.gif and *.png. By using the Java Advanced Imaging API (JAI), it is possible to get the pixel content of the images, in order to perform the image processing operations. At the moment, Dilation and Erotion operators has been implemented using the Image Processing Toolbox of MATLAB; this was possible by an interface build by Ian Toft (Toft, 2002), which permits to open the MatlabEngine to invoke the commands of the toolbox. Figure 5 shows the list code that performs the connection between the Java Virtual Machine and the MatlabEngine.

```
MatlabEngine engine = MatlabEngine.getEngineInstance();
        try {
                engine.openEngine();
                engine.putDoubleMatrix("doubmat", preal);
                engine.evalString("imshow(doubmat);");
                engine.evalString("imwrite(doubmat,'C:/Hadamard.jpg');");
                System.out.println("Going to sleep for 10 seconds...");
                try {
                        Thread.sleep(5000);
                }
                catch(InterruptedException e) {
                        e.printStackTrace();
                }
        }
        catch(MatlabEngineException mle)
        {
                mle.printStackTrace();
                System.out.println(mle);
        }
        finally
        {
                try {
                        engine.closeEngine();
                }
                catch(MatlabEngineException mle)
                {
                        mle.printStackTrace();
                        System.out.println(mle);
                        System.exit(-1);
                }
        }
```

Figure 5: List code of the interface between Java and MATLAB

Table 2 shows a partial list of the operators currently implemented in this moment. They were divided according to the method that they use for the treatment of the pixels (Hualparimachi, 2003).

Table 2. Partial List of Implemented Operators

Point Operators	Arithmetic Operators	Spatial Operators	Morphological Operators	Filter Operators	Cyclic Operators	Complex Operators
Absolute	Addition	Invert	Dilation	Sharpen	Cyclic Convolution	Hadamard product
Clamp	Substraction	Crop	Erotion	Blur	Cyclic Correlation	Phase Calculation
Color Converter	Exponential	Resize		Emboss	Shift Operator	Conjugate Operation
	Logarithm	Rotate		Sobel		Fourier Transform
		Flip		Prewitt		
				Roberts		
				Frei Chen		
				Low Pass		
				High Pass		
				Gaussian		

5. RESULTS AND DISCUSSION

At the present time, J-CID has been developed a stand-alone Java application for image processing. It has a friendly user interface which permits apply different types of operators to images. The environment has the capability to connect remotely to a server through a Java FTP protocol, in order to download data to the local system to perform the processing. Here is an example of how a user can interact with J-CID. In Figure 6 appears an screenshot of J-CID where has been opened an image. In the top of the screen can be seen the menu bar where are located the operators enumered in Table 2. Suppose that a user wants to apply the Dilation operator over the image. Immediately appears an Dialog window where the user can choose the parameters for the operator.

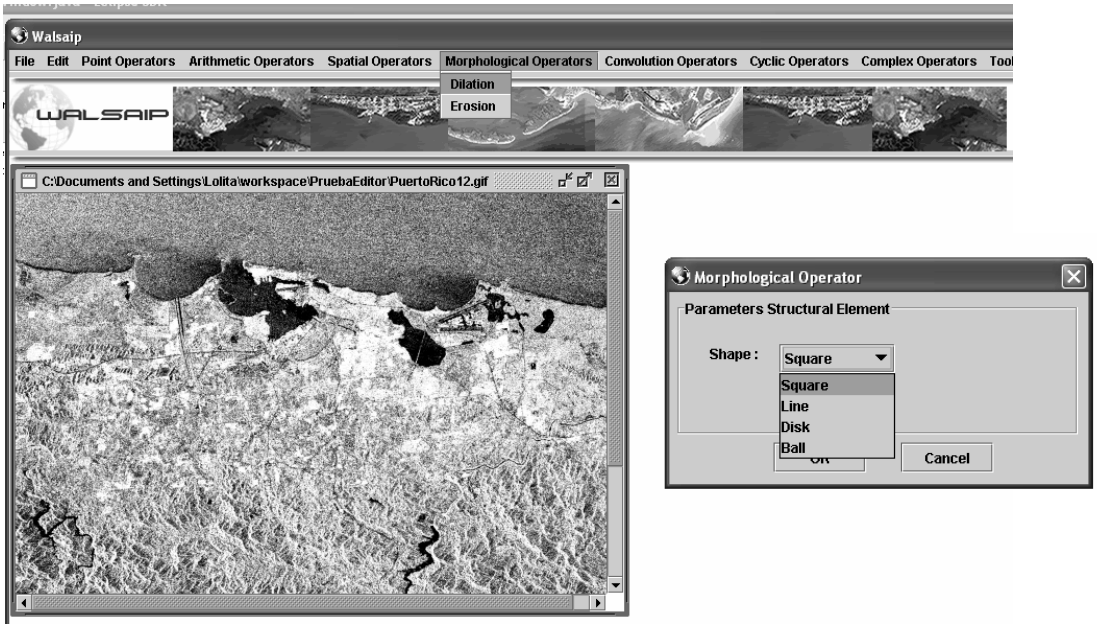

Figure 6. Screenshot of J-CID in Execution

After the user selects the parameters, the system starts to execute the operator. In this case, the operator is related with the Image Processing Toolbox of MATLAB®, for that reason emerges a window of the command window of MATLAB® until it finishes its execution. After that, the produced image is also displayed in J-CID, see Figure 7.

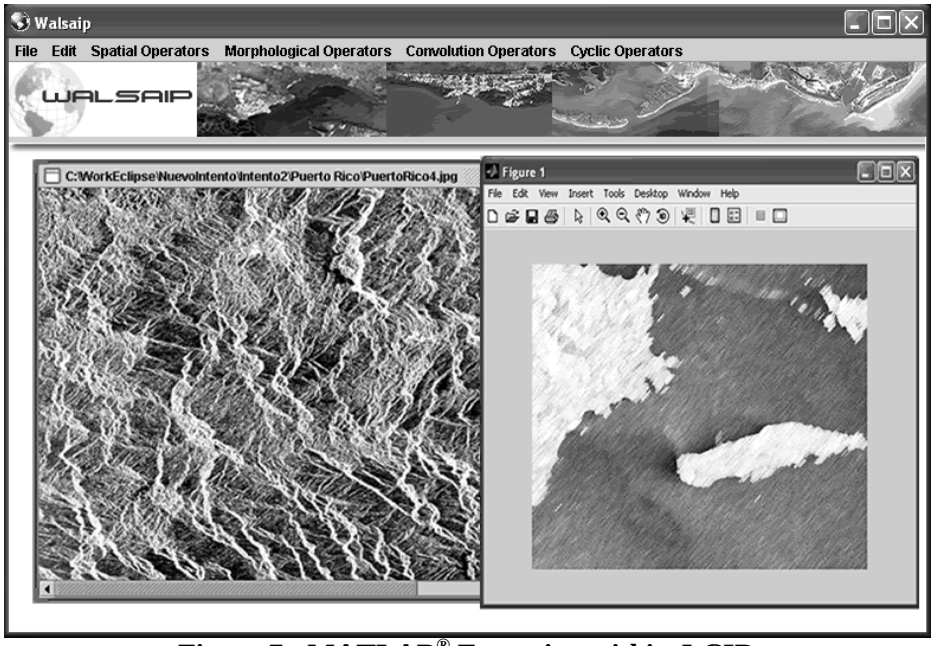

Figure 7: MATLAB® Execution within J-CID

J-CID also includes the facility to encapsulate a sequence of operators used over an image, in order to build a customized operator that could be used in the future. Figure 8 shows the example of three operators that were applied to an image and that could be integrated into a new one. Introducing the name for the new operator into de "Command Operator" and clicking on the "Execute" button, the new operator is added into the apropiate set of existing operators of the environment. With this feature of J-CID we want to increment the number of operators and permits users to make their contributions to the field of image processing and SAR research.

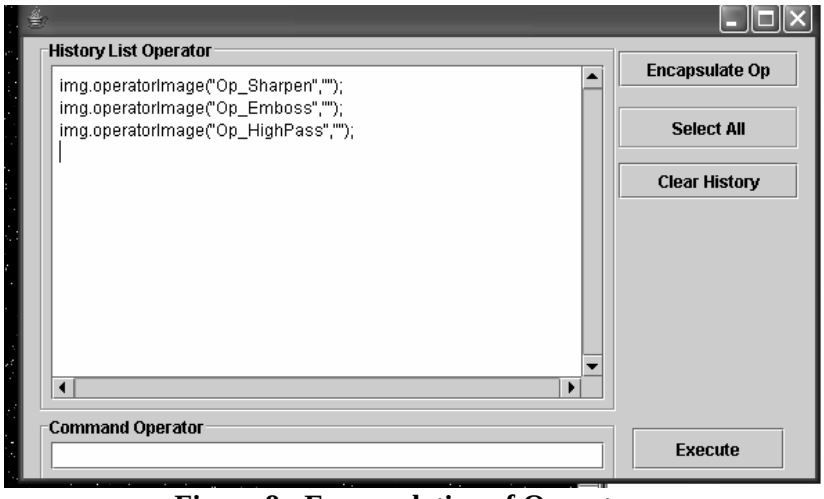

Figure 8: Encapsulation of Operators

6. Conclusions

The developing environment is being designed to be a useful tool for geologists, hydrologists, and scientists in related areas. It provides a friendly access to basic and advanced operators, with encapsulation capabilities. One of the most important features of the environment is the portability offered by Java, that also facilitates the availability of the source code for potential users. Distributed services features will enhance the capability of this tool.

7. References

Huallparimachi, C. and Rodriguez, D. (2003) "Java-based Tool for Synthetic Aperture Radar Image Analysis". Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis.

Kennedy, K. Broom, B. Chauhan, A. Fowler, R.J. Garvin, J. Koelbel, C. McCosh, C. Mellor-Crummey, J. (2005) "Telescoping languages: a system for automatic generation of domain languages". Invited paper Proceedings IEEE Vol 93, No 2.

Luzi, G., Pieraccini, M., Mecatti, D., Noferini, L., Guidi, G., Moia, F., Atzeni, C. (2004) "Ground-Based Radar Interferometry for Landslides Monitoring: Atmospheric and Instrumental Decorrelation Sources on Experimental Data". IEEE Transactions on Geoscience and Remote Sensing. Vol 42, No. 11.

Rodriguez, D. (2002). "Computational Signal Processing and Sensor Array Signal Algebra: A Representation Development Approach". University of Puerto Rico at Mayaguez.

Rosen, P., Hensley, S., Joughin, I., Li, F., Madsen, S., Rodriguez, E., Goldstein, R. (2000). "Synthetic Aperture Radar Interferometry". Invited Paper Proceedings of IEEE Volume 88 Issue 3.

Toft, I. "JLab – Connecting Java and MATLAB". www2.cmp.uea.ac.uk/~it/code.html.

Villamizar, J., Bautista, L., and D. Rodriguez (2005) "A Computational Signal Processing System for Correlated Digital Interferometry". IEEE Transactions on Circuits and Systems.

8. Acknowledgments

This project is partially supported by NSF Grant No. CNS0424546.

Authorization and Disclaimer