SARCSPE: SYNTHETIC APERTURE RADAR COMPUTATIONAL SIGNAL PROCESSING ENVIRONMENT

Dilia Beatriz Rueda S. *Domingo Rodríguez*
dilia@ ece.upm.edu *domingo@ ece.upm.edu*

University of Puerto Rico - Mayagüez Campus
Department of Electrical and Computer Engineering
Computational **S**ignal **P**rocessing **G**roup *(cspg@ ece.upm.edu)*
Mayagüez, Puerto Rico, 00681-9042

ABSTRACT

In this paper, a MATLAB® v. 5.3 environment is presented. The environment is called SARCSPE (Synthetic Aperture Radar Computational Signal Processing Environment) because it includes all operations necessary for performing SAR (Synthetic Aperture Radar) data processing. Operators based on fast transformations such as the Fourier transform have been incorporated. Kronecker products have been used to optimize the algorithms. Special care has been put on the simulation and development of image from raw data recovery algorithms.

1. INTRODUCTION

Synthetic Aperture Radar (SAR) is a remote sensing technique that allows not only to illuminate a scene at any moment independently on weather conditions, but also to improve azimuth resolution taking advantage of the sensor motion to simulate a big antenna. With such an antenna, a sensor working on the microwave frequency range is able to acquire great amounts of raw data of very high resolution but useless without the appropriate processing. Data processing can be performed if fast and accurate algorithms are developed for SAR data processing, and they can be improved if those processing tools are user friendly, easy to use, and available at low cost.

Taking advantage of powerful signal processing tools such as the fast Fourier transform (FFT) and the cyclic convolution, an environment for SAR processing has been developed and constitutes the work reported in this article. The environment includes implementations of SAR processing operations and simulation algorithms. Many of them have been improved with the use of Kronecker products and have been implemented as matrix vector product based operations [4]. For this reason, MATLAB® v. 5.3 has been used as a development tool.

This paper is organized as follows: First, SAR imaging concepts are explained; second, matrix and imaging array algebra concepts are given; third, image processing aspects are treated; fourth, some important operations implemented are presented; fifth, the conclusions are made and, finally, future work is proposed.

2. SAR IMAGING SYSTEM

In SAR (Synthetic Aperture Radar), there is a sensor illuminating an area while traveling in a platform to simulate a big antenna and improve in this way the azimuth resolution.

SAR operates in the microwave region of the spectrum of frequencies, for this reason, data acquisition can be made at almost any moment. The response of each point in the scene provides information about characteristics of the illuminated area such as the system impulse response and the reflectivity and it's used by the sensor in the process of raw data formation [1]. In the same way, raw data and the system impulse response are useful for computing an estimate of the reflectivity function [2] as shown in Figure 1, where the system transfer function is multiplied by the Fourier transform of the reflectivity function, then some corrections are applied to this result for obtaining the image correspondent to the raw data.

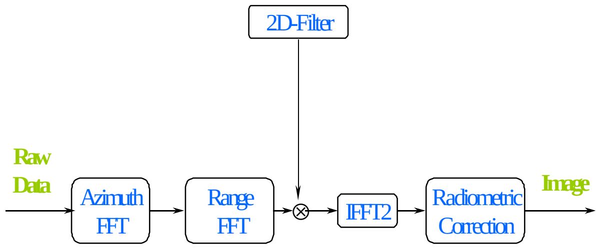

Figure 1 Image Formation Algorithm

Raw data is computed as the convolution between the system impulse response and the scene reflectivity function:

$$h(x,r) = g(x,r) \circledast g(x,r) \quad (1)$$

Taking the Fourier transform to (1) results in

$$H(x,h) = \Gamma(x,h) \odot G(x,h) \quad (2)$$

Factorizing the terms in (2), we obtain that

$$\Gamma(x,h) = H(x,h) \odot G^{-1}(x,h) \qquad (3)$$

The block diagram shown in Figure 1 is a representation of the results obtained in (3) plus the radiometric correction. More variants of the algorithm can be obtained if more aspects are considered.

3. SARCSPE OPERATION

For an individual interested in SAR, processing it is of great value to have a toolbox easy to use with the computational tools necessary to perform necessary data manipulations. This is one of the reasons for developing a computational environment.

One environment is an entity with *operators* organized as *operation rules* in order to apply *action rules* over *input sets* for producing *output sets*. One *user interface* facilitates makes it easy for an individual to access those tools. One environment is said to be a computational environment if fast and accurate algorithms are included in it.

Considering the concept of environment, we have programmed SARCSPE (Synthetic Aperture Radar Computational Signal Processing Environment). The programming language used was MATLAB® [5]. As an example, let's take the case of image formation. Raw data and the system impulse response are *input sets*. Fourier transform, Hadamard product and the inverse Fourier transform are *operators*. The sequence of operations, first the Fourier transform of each input set, then the Hadamard product, and then the inverse Fourier transform, are *operation rules*. Fourier transform is an *action rule*. The resultant image is the output set.

The performance of a computational environment depends greatly on the implementation of the algorithms. In the case of SARCSPE, the optimization has been done in three different ways. The first, using fast transformations such as the fast Fourier transform that allow fast implementations. The second, using matrix vector representations, easy to optimize with the use of MATLAB®. The last way, using Kronecker products, a mathematical tool that allows reducing the number of redundant and unnecessary operations [3].

In SARCSPE, the components are organized according to their roll. Different alternatives have been presented to the user; for example, different computation methods for executing the same task trying always to minimize the number of computations. Important functions included were ambiguity function, fast Fourier transform, cyclic correlation operation, and cyclic convolution operation.

Figure 4 and Figure 5 illustrate an example of how the environment has been conceived. Figure 4 shows the computation of the point spread function. Figure 5 shows an example of one algorithm of image formation applied to data received at the University of Puerto Rico SAR station. In this case, radiometric corrections have not been considered.

The steps followed for computing the result were: first, apply the cyclic reflection operation to image B, and second compute cyclic convolution between the result and image A. The same result could be obtained by applying any of the available methods provided in the GUI menu for computing cyclic correlation.

As it can be appreciated in the GUI (Graphical User Interface) in Figure 4, the environment is organized following the concepts explained above. Different computational operations such as filtering operations, transformations, arithmetic operations and cyclic convolution operations have been included, and can be located at the left side of the GUI. Some applications using computational functions have been included here, they can be found in the upper right side of the GUI. Visual options can be seen at the lower right side of the GUI.

Figure 2 and Figure 3 illustrate more examples of operations that can be executed with SARCSPE: the computation of the point spread function of a pulse and the point spread function of a chirp signal.

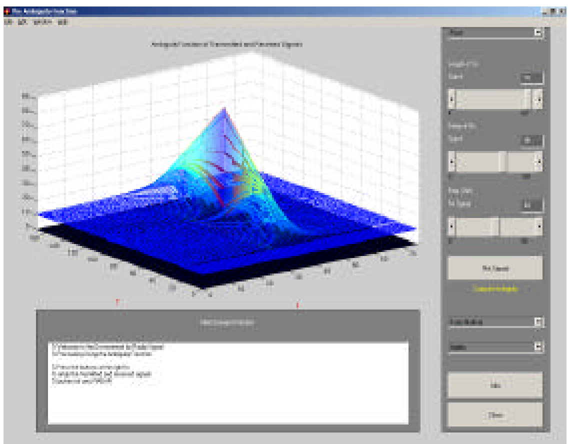

Figure 2 Point Spread Function Simulation of a Pulse Signal.

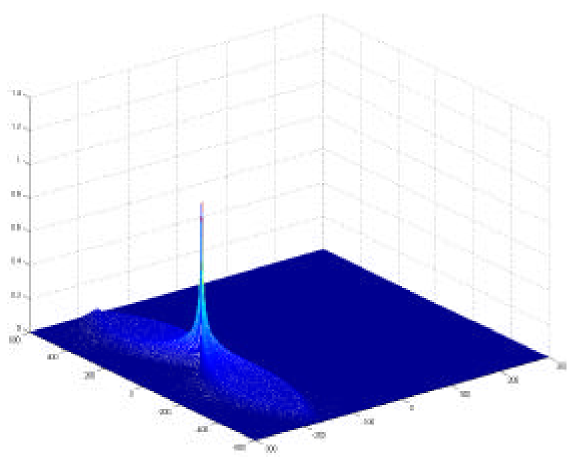

Figure 3 Point Spread Function simulation of a Chirp Signal.

4. CONCLUSIONS

The concept of environment was explained. Following this conceptualization, a computational environment has been developed with all the tools necessary to perform synthetic Aperture Radar signal processing. An image formation approach was presented and explained in the context of environment. Algorithm improvement was also considered and included.

5. FUTURE DIRECTION

Different approaches of image formation algorithms must be considered in different hardware architectures with different software platforms. Real time data acquisition implementations must be considered also.

6. REFERENCES

[1] R. E. Blahut, W. Miller Jr., C.H. Wilcox, "Radar and Sonar", *The IMA volumes in Mathematica and its Applications*, part 1, vol. 32.

[2] G. Franceschetti, R. Lanari, E.S. Marzouc, "Efficient and High Precision Space-Variant Processing of SAR Data", *IEEE Transactions on Aerospace and Electronics Systems*, Vol. 31, No. 1, January 1995.

[3] J. R. Johnson, R. W. Johnson, D. Rodríguez, R. Tolimieri, "A Methodology for Designing, Modifying, and Implementing Fourier Transform Algorithms on Various Architectures", IEEE *Transactions on Circuits, Systems, and Signal Processing*, Vol. 9, No. 4, 1990.

[4] D. Rodríguez, "Computational Kronecker Products for Multidimensional SAR Signal Processing Algorithms", *IGARSS*, 1998.

[5] The MathWorks Inc., "MATLAB® the Language of Technical Computing". http://www.mathworks.com

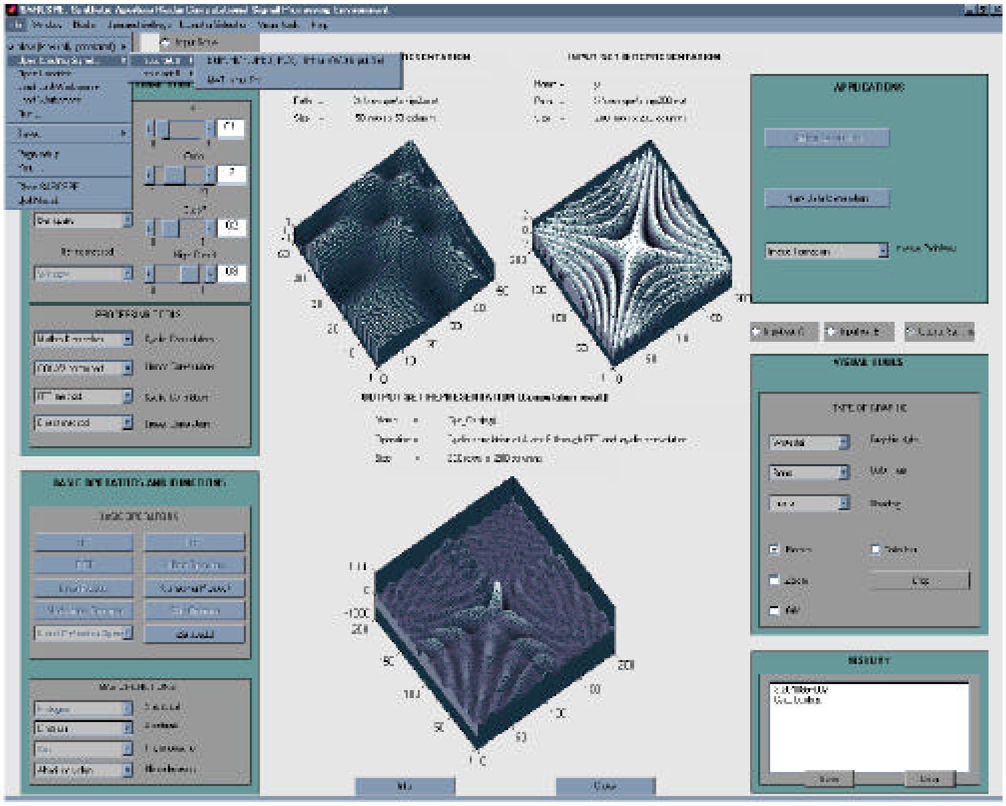

Figure 4 Signal Generation and Point Spread Function with SARCSPE.

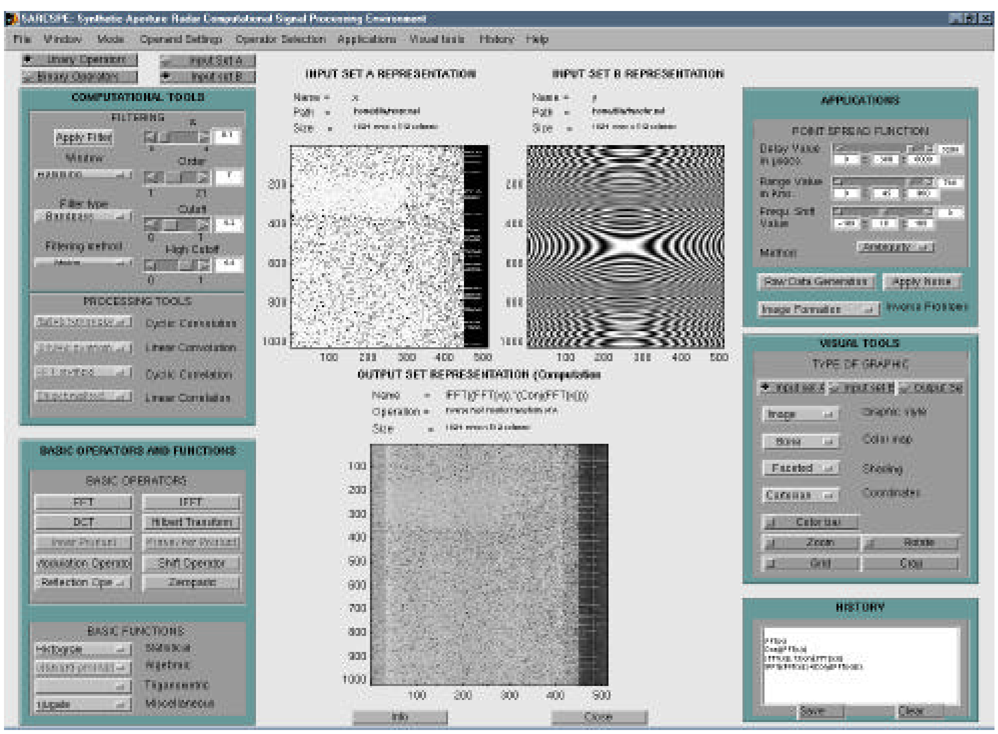

Figure 5 Example of Image Formation 1