THE STRUCTURE OF THE COMPUTATIONAL SIGNAL ALGEBRA AND ITS APPLICATION IN DIGITAL IMAGE PROCESSING

LA ESTRUCTURA DEL ÁLGEBRA DE SEÑALES COMPUTACIONAL Y SUS APLICACIONES EN PROCESAMIENTO DE IMÁGENES DIGITALES

MARLIO PAREDES
Professor, School of Science and Technology, Universidad del Turabo, Puerto Rico, USA, maparedes@suagm.edu

JORGE VILLAMIZAR MORALES
Escuela de Matemáticas, Universidad Industrial de Santander, jorge@uis.edu.co

LOLA XIOMARA BAUTISTA ROZO
Automated Information Processing Laboratory, University of Puerto Rico at Mayaguez, lola.bautista@ece.uprm.edu

DOMINGO RODRIGUEZ
Professor, ECE Department University of Puerto Rico at Mayaguez, Puerto Rico, USA, domingo@ece.uprm.edu

Received for review August 10 th, 2009, accepted October 13th, 2010, final version October, 22 th, 2010

ABSTRACT: This work starts from the knowledge of the mathematical structure of the signals space used in signal processing and provides the development of a computational theoretical framework of signal algebra for modeling and processing applications using digital images. The mathematical structures were implemented over computational structures using the Java programming language as a tool for the coding of the algorithms. The implemented tool was called JCID (Java Computational Image Developer), which allows for one to implement several of the operators from the signal algebra for one-dimensional and two-dimensional signals, and the creation of new entrants through the composition of the basic operators.

KEYWORDS: Signals space, digital images, signal algebra, mathematical structure, digital signal processing.

RESUMEN: Este trabajo se inicia a partir del conocimiento de la estructura matemática del espacio de señales usado en el procesamiento de señales y provee el desarrollo de un marco teórico computacional de álgebra de señales para el modelamiento y procesamiento de aplicaciones usando imágenes digitales. Las estructuras matemáticas fueron implementadas sobre estructuras computacionales usando el lenguaje de programación Java como una herramienta para la codificación de los algoritmos. La herramienta implementada fue llamada JCID (Java Computational Image Developer), la cual permite implementar varios de los operadores del algebra de señales para señales de dimensión uno y dimensión dos, y la creación de nuevas entradas a través de la composición de los operadores básicos.

PALABRAS CLAVE: Espacio de señales, imágenes digitales, algebra de señales, estructura matemática, procesamiento de señales digitales.

1. INTRODUCTION

Digital signal processing and image processing are techniques used in different fields such as hydrology and medicine. Although the conditions of acquisition of data are completely different, most of the final outputs displayed to a user are relatively seamless. Also, most of the techniques used to do processing in one of them is also used in the other. For example, applying filters for image enhancement is a common tool that is available in specialized software for either hydrology or medicine.

The advantage of creating a Computational Signal Processing System that deals with the algorithmic treatment of signal-based data is that, independent of the kind of data or its source, the data can be processed by standard methods established by a formal mathematical notation which can be translated to any programming language. In this work, Digital Signal Processing (DSP) techniques are used to represent the signals mathematically and get the information carried by them. The mathematical basis of DSP algorithms lies in the set theory, group theory, and algebraic structures concepts which can be applied to the fields of real numbers ($\mathbb{R}$) and complex numbers ($\mathbb{C}$).

Signal algebra is constructed by using binary two-dimensional cyclic convolution as the product operation that turns a linear space of two-dimensional finite discrete images into linear algebra. Matrix representations of two-dimensional cyclic convolution operations are represented as block circulant matrices with circulant blocks [1]. This is accomplished when finite discrete image object arrays, serving as inputs in the matrix-vector computation of the two-dimensional cyclic convolution operation, are transformed into one-dimensional column vectors by using both lexicographic and anti-lexicographic ordering. Special attention is given to the algebra of cyclic correlations which is related to the algebra of cyclic convolutions through the index reversal or reflection operator. Some basic DSP concepts such as the definition of one-dimensional signal space, its algebraic structure, and the operators that act over this space are presented [2]. All these concepts are generalized later to two-dimensional signal space.

2. SIGNAL ALGEBRA

The mathematical concepts and results presented in sections 2, 3, and 4, were studied in detail in [3].

Definition 1: A function whose domain is a discrete set will be called a discrete function. Here, every discrete function will be called a **discrete signal**.

Definition 2: A function whose codomain is a discrete finite set is called a **digital signal**. If the codomain is the set of complex numbers, it is said that the signal is a complex discrete of dimension N.

Definition 3: Let F be a field and let V and W be two algebras over F. An **isomorphism** between the algebras V and W is a bijective function $T\colon\ V \rightarrow W$ such that

1. $T(\alpha_1 v_1 + \alpha_2 v_2) = \alpha_1 T(v_1) + \alpha_2 T(v_2)$,

2. $T(v_1 \cdot v_2) = T(v_1) \cdot T(v_2)$,

for all $v_1, v_2 \in V$ and all $\alpha_1, \alpha_2 \in F$. Notice that the first condition establishes that an isomorphism of algebras is an isomorphism of vector spaces which, in addition, preserves the products.

Definition 4: A circulant matrix of order N is a square matrix of order $N \times N$ of the form

$$C_N = \begin{bmatrix} C_0 & C_1 & \cdots & C_{N-1} \\ C_1 & C_2 & \cdots & C_0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{N-2} & C_{N-1} & \cdots & C_{N-3} \\ C_{N-1} & C_0 & \cdots & C_{N-2} \end{bmatrix}$$

The elements of each column (row) of C_N are identical to those of the previous column (row), but they are moved one position up (left) and wrapped around. So the whole matrix is determined by the first column (or row). Also, it may occur that the entries move going down, and in this case the matrix has the form

$$H_N = \begin{bmatrix} H_0 & H_{N-1} & \cdots & H_1 \\ H_1 & H_0 & \cdots & H_2 \\ \vdots & \vdots & \ddots & \vdots \\ H_{N-2} & H_{N-3} & \cdots & H_{N-1} \\ H_{N-1} & H_{N-2} & \cdots & H_0 \end{bmatrix}$$

Let $C_{[0]}, C_{[1]}, \ldots, C_{[N-2]}, C_{[N-1]}$ be square matrices each of the order N. By a block circulant matrix of type (m, n) (and of the order mn) is meant an array of order mn which has the form

$$C_N = \begin{bmatrix} C_{[0]} & C_{[1]} & \ldots & C_{[N-1]} \\ C_{[1]} & C_{[2]} & \ldots & C_{[0]} \\ \vdots & \vdots & \ddots & \vdots \\ C_{[N-2]} & C_{[N-1]} & \ldots & C_{[N-3]} \\ C_{[N-1]} & C_{[0]} & \ldots & C_{[N-2]} \end{bmatrix},$$

where each entry $C_{[i]}$ is a circulant matrix.

Definition 5: The simplest relation between a finite-length sequence $x[n]$, defined for $0 \leq n \leq N-1$, and its **discrete-time Fourier transform** (DTFT) $x\hat{}(e^{jw})$ is obtained by uniformly sampling $x\hat{}(e^{jw})$ on the w-axis between $0 \leq w \leq 2\pi$ at $w_k = 2\pi k/N$, $0 \leq k \leq N-1$ and it is usually written:

$$x\hat{}[k] = x\hat{}(e^{jw})|_{w=2\pi k/N} = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N},$$

where $0 \leq k \leq N-1$. Note that $x\hat{}[k]$ is also a finite-length sequence in the frequency domain of length N . The sequence $x\hat{}[k]$ is called the discrete Fourier transform (DFT) of the sequence $x[n]$. According to the commonly used notation $W_N = e^{-j2\pi/N}$, we can rewrite the DFT of $x[n]$ as

$$x\hat{}[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad 0 \leq k \leq N-1.$$

The inverse discrete Fourier transform (IDFT) of $y[n]$ is given by

$$y^{\vee}[n] = \frac{1}{N}\sum_{k=0}^{N-1} y[k]W_N^{-kn}, \quad 0 \le n \le N-1.$$

3. THE SPACE OF ONE-DIMENSIONAL SIGNALS

Definition 6: A **one-dimensional signal** is a function

$$\begin{array}{cccc} x: & \mathbb{Z}_N & \longrightarrow & \mathbb{C} \\ & n & \longmapsto & x[n] \end{array}$$

An equivalent form to denote a signal is as a sequence of complex numbers $x[0], x[1], \ldots, x[N-1]$. It is common to write a signal as a column vector

$$x = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

The space of one-dimensional signals is denoted by $l^2(\mathbb{Z}_N)$. This space has a vector space structure over the complex numbers with the sum of functions and the multiplication by a scalar [3]. A basis for this signal space is the set$\Delta_N = \{\delta_{\{k\}} : k = 0, 1, \ldots, N-1\}$,where the functions $\delta_{\{k\}}$ are defined by

$$\delta_{\{k\}}[j] = \begin{cases} 1, & \text{si } j = k, \\ 0, & \text{si } j \neq k. \end{cases}$$

This is the standard basis for $l^2(\mathbb{Z}_N)$.

Definition 7: The **cyclic convolution** operation over the space of one-dimensional signals is defined as follows:

$$\begin{array}{cccc} \circledast_N: & l^2(\mathbb{Z}_N) \times l^2(\mathbb{Z}_N) & \longrightarrow & l^2(\mathbb{Z}_N) \\ & (x, y) & \longmapsto & x \circledast_N y \end{array},$$

where $x \circledast_N y$ is defined by

$$(x \circledast_N y)[n] = \sum_{k=0}^{N-1} x[k]y[\langle n-k \rangle_N].$$

The cyclic convolution has the following properties for $x, y, z \in l^2(\mathbb{Z}_N)$ and $\alpha \in \mathbb{C}$:

1. $x \circledast_N y = y \circledast_N x$.
2. $(x \circledast_N y) \circledast_N z = x \circledast_N (y \circledast_N z)$.
3. $x \circledast_N (y + z) = x \circledast_N y + x \circledast_N z$
4. $\alpha(x \circledast_N y) = (\alpha x) \circledast_N y = x \circledast_N (\alpha y)$.

With all the previous properties, the following result can be obtained.

Theorem 1: The space of one-dimensional signals $l^2(\mathbb{Z}_N)$ is a commutative algebra using the cyclic convolution as multiplication.

Now, let's represent the cyclic convolution as an operator over the signal space $l^2(\mathbb{Z}_N)$. Let $x, h \in l^2(\mathbb{Z}_N)$ and $y = x \circledast h$, then we obtain the following system

$$y[n] = (x \circledast_N h)[n] = \sum_{k=0}^{N-1} x[k]h[\langle n-k \rangle_N],$$

where $0 \le n \le N-1$. This system can be written as $Y_N = H_N X_N$, where

$$Y_N = \begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix}, \quad X_N = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \quad \text{and}$$

$$H_N = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{bmatrix}$$

Fixing h, the cyclic convolution operation becomes a linear operator acting over the signals space which is called the cyclic convolution operator. This operator is denoted by $\circledast_N^h$ and defined as

$$\begin{array}{cccc} \circledast_N^h: & l^2(\mathbb{Z}_N) & \longrightarrow & l^2(\mathbb{Z}_N) \\ & x & \longmapsto & \circledast_N^h\{x\} = x \circledast_N h \end{array}$$

The matrix of this operator is H_N which is a circulant matrix.

Definition 8: **The cyclic correlation** operation over the space of one-dimensional signals is defined as

$$\begin{array}{cccc} \diamondsuit_N: & l^2(\mathbb{Z}_N) \times l^2(\mathbb{Z}_N) & \longrightarrow & l^2(\mathbb{Z}_N) \\ & (x, y) & \longmapsto & x \diamondsuit_N y \end{array},$$

where

$x \circledcirc_N y$ is defined by

$$(x \circledcirc_N y)[n] = \sum_{k=0}^{N-1} x[k]y[\langle n+k\rangle_N] \,.$$

The following properties of the cyclic correlation can be proved easily. For every $x, y, z \in l^2(\mathbb{Z}_N)$ and $\alpha \in \mathbb{C}$,

1. $(x \circledcirc_N y) \circledcirc_N z = x \circledcirc_N (y \circledcirc_N z)$.

2. $x \circledcirc_N (y+z) = x \circledcirc_N y + x \circledcirc_N z$.

3. $\alpha(x \circledcirc_N y) = (\alpha x) \circledcirc_N y = x \circledcirc_N (\alpha y)$.

These properties provide the following result:

Theorem 2: The space of the one-dimensional signals $l^2(\mathbb{Z}_N)$ is a non-commutative algebra when the cyclic correlation is considered to be the multiplication operation.

The cyclic correlation can be expressed as an operator over the space of one-dimensional signals $l^2(\mathbb{Z}_N)$. Indeed, fixing the signal $g \in l^2(\mathbb{Z}_N)$ we define the cyclic correlation operator as

$$\begin{array}{rccc} \circledcirc_N^g : & l^2(\mathbb{Z}_N) & \longrightarrow & l^2(\mathbb{Z}_N) \\ & x & \longmapsto & \circledcirc_N^g\{x\} = x \circledcirc_N g \end{array}$$ .Using the standard basis of $l^2(\mathbb{Z}_N)$ it is easy to calculate the matrix of the cyclic correlation

$$C_N = \begin{bmatrix} g[0] & g[1] & g[2] & \cdots & g[N-1] \\ g[1] & g[2] & g[3] & \cdots & g[0] \\ g[2] & g[3] & g[4] & \cdots & g[1] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g[N-2] & g[N-1] & g[0] & \cdots & g[N-3] \\ g[N-1] & g[0] & g[1] & \cdots & g[N-2] \end{bmatrix}$$

Notice that, like the cyclic convolution operator, the matrix of the cyclic correlation operator is also a circulant matrix.

Definition 9: **The reflexive operator** over the space of one-dimensional signals is defined by

$$\begin{array}{rccc} \mathfrak{R}_N : & l^2(\mathbb{Z}_N) & \longrightarrow & l^2(\mathbb{Z}_N) \\ & x & \mapsto & \mathfrak{R}_N\{x\} = x^{(-)}, \end{array}$$ where

$$(\mathfrak{R}_N\{x\})[k] = x^{(-)}[k] = x[\langle N-k\rangle_N] = x[\langle -k\rangle_N].$$

It is easy to see that the matrix of the reflexive operator is given by

$$R_N = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix},$$

which is again a circulant matrix. The following theorem establishes an interesting property between the cyclic convolution operator, the cyclic correlation operator, and the reflexive operator.

Theorem 3: Let $x, y \in l^2(\mathbb{Z}_N)$. Then

$$x \circledcirc_N y = (\mathfrak{R}_N\{x\}) \circledast_N \,.$$

Definition 10: **The cyclic shift operator** over the space of one-dimensional signals is defined by

$$\begin{array}{rccc} \mathcal{S}_N : & l^2(\mathbb{Z}_N) & \longrightarrow & l^2(\mathbb{Z}_N) \\ & x & \mapsto & \mathcal{S}_N\{x\} \end{array},$$

where $(\mathcal{S}_N\{x\})[n] = x[\langle n+1\rangle_N]$.

The matrix representation of this operator with respect to the standard basis is

$$S_N = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

Definition 11: **The Hadamard product** over the space $l^2(\mathbb{Z}_N)$ of one-dimensional signals is defined as

$$\begin{array}{rccc} \odot_N : & l^2(\mathbb{Z}_N) \times l^2(\mathbb{Z}_N) & \longrightarrow & l^2(\mathbb{Z}_N) \\ & (x, y) & \longmapsto & x \odot_N y \end{array},$$

where $(x \odot_N y)[n] = x[n]\, y[n]$.

The Hadamard product satisfies the following properties, for all $x, y, z \in l^2(\mathbb{Z}_N)$ and $\alpha \in \mathbb{C}$,

1. $x \odot_N y = y \odot_N x$

2. $x \odot_N (y \odot_N z) = (x \odot_N y) \odot_N z$

3. $x \odot_N (y+z) = x \odot_N y + x \odot_N z$

4. $\alpha(x \odot_N y) = (\alpha x) \odot_N y = x \odot_N (\alpha y)$

Theorem 4: The space $l^2(\mathbb{Z}_N)$ of one-dimensional signals is a commutative algebra with the Hadamard product as multiplication.

The following theorem relates the cyclic convolution and the Hadamard product through the Discrete Fourier Transform (see *Definition 5*).

Theorem 5: If S_0 and S_1 are one-dimensional signals, then

$$(S_0 \circledast_N S_1)^\frown = (S_0)^\frown \odot_N (S_1)^\frown.$$

Corollary 1: If $S_0, S_1 \in l^2(\mathbb{Z}_N)$ then $S_0 \circledast_N S_1 = [(S_0)^\frown \odot_N (S_1)^\frown]^\vee$.

Notice that the DFT is a homomorphism of vector spaces, or equivalently, a linear transformation between the one-dimensional space and itself,

$$\begin{array}{cccc} ^\frown: & l^2(\mathbb{Z}_N) & \longrightarrow & l^2(\mathbb{Z}_N) \\ & S & \longmapsto & S^\frown \end{array}.$$

This operator has the following properties. Let $S, S_0, S_1 \in l^2(\mathbb{Z}_N)$ and $\alpha \in \mathbb{C}$. Then,

1. $(S_0 + S_1)^\frown = (S_0)^\frown + (S_1)^\frown$.
2. $(\alpha S)^\frown = \alpha S^\frown$.

In addition, the DFT is an injective function because every signal has its own transform, i.e., $S_0 \neq S_1 \Longrightarrow (S_0)^\frown \neq (S_1)^\frown$. Because of $l^2(\mathbb{Z}_N)$ is a finite dimensional space, we can conclude the DFT is a bijective function. The previous theorem shows the DFT is a homomorphism of algebras between $(l^2(\mathbb{Z}_N), \circledast_N)$, the space of one-dimensional signals with the cyclic convolution operation, and $(l^2(\mathbb{Z}_N), \odot)$, the space of one-dimensional signals with the Hadamard product operation. With all of this established, we can now say that:

Theorem 6: The space of one-dimensional signals with the cyclic convolution operation, $(l^2(\mathbb{Z}_N), \circledast_N)$, is isomorphic as an algebra for the same space of one-dimensional signals with the Hadamard product $(l^2(\mathbb{Z}_N), \odot)$.

Theorem 7: If $S \in l^2(\mathbb{Z}_N)$ then

$$(\Re_N\{S^*\})^\frown = (S^\frown)^*.$$

4. THE SPACE OF TWO-DIMENSIONAL SIGNALS

The space of two-dimensional signals contains all the functions of the form

$$\begin{array}{cccc} S: & \mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1} & \longrightarrow & \mathbb{C} \\ & [n_0, n_1] & \longmapsto & S[n_0, n_1] \end{array}.$$

This space will be denoted by $l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$.

A two-dimensional signal S in $l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$ can be represented as a matrix of size $N_0 \times N_1$,

$$S = \begin{bmatrix} S[0,0] & S[0,1] & \cdots & S[0, N_1-1] \\ S[1,0] & S[1,1] & \cdots & S[1, N_1-1] \\ \vdots & \vdots & \ddots & \vdots \\ S[N_0-1, 0] & S[N_0-1, 1] & \cdots & S[N_0-1, N_1-1]. \end{bmatrix}.$$

Also, this space of signals is a vector space over the complex numbers with the sum of functions and the usual product by a scalar number. The standard basis for this space is denoted by

$$\begin{aligned} \Delta_{N_0 \times N_1} &= \{\delta_{\{k_0,k_1\}} : 0 \le k_0 \le N_0 - 1 \text{ y } 0 \le k_1 \le N_1 - 1\} \\ &= \Delta_N \end{aligned}$$

and its elements are defined by

$$\delta_{\{k_0,k_1\}}[n_0, n_1] = \begin{cases} 1, & \text{if } [n_0, n_1] = [k_0, k_1], \\ 0, & \text{if } [n_0, n_1] \neq [k_0, k_1]. \end{cases}$$

A. The Algebra of Cyclic Convolutions of the order $N_0 \times N_1$ over $\mathbb{C}$

The vector space or signals space $l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$ is turned into a linear algebra, which we call a *Signal Algebra*, by introducing the vector multiplication or signal multiplication operation, denoted by $\circledast_N$ where $N = N_0 \times N_1$, defined as

$$\begin{array}{cccc} \circledast_N : & l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \times l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) & \longrightarrow & l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \\ & (x, h) & \longmapsto & x \circledast_N h, \end{array},$$

where

$$(x \circledast_N h)[n_0, n_1] = \sum_{k_1=0}^{N_1-1} \sum_{k_0=0}^{N_0-1} x[k_0, k_1] h[\langle n_0 - k_0 \rangle_{N_0}, \langle n_1 - k_1 \rangle_{N_1}]$$

As in the one-dimensional case, the cyclic convolution operation is commutative in the two-dimensional case, so that we have the following result:

Theorem 8: The space $l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$ is a commutative algebra with the cyclic convolution as the multiplication operation.

In the two-dimensional case, there is also a cyclic convolution operator. Indeed, fixing a signal $h \in l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$, we define the cyclic convolution operator as

$$\begin{array}{ll}\circledast_N^h : l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \longrightarrow l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \\ \qquad x \qquad \longmapsto \circledast_N^h\{x\} = x \circledast_N h\end{array}.$$

By using the anti-lexicographic order with respect to the standard basis Δ_N, the matrix representation of this operator is obtained as

$$H_N = [\circledast_N^h\{\delta_{\{0,0\}}\}\ \circledast_N^h\{\delta_{\{1,0\}}\} \ldots \circledast_N^h\{\delta_{\{N_0-1,N_1-1\}}\}],$$

where $\delta_{\{i,j\}} \in \Delta_N$. This matrix can be written as the circulant block matrix

$$H_N = \begin{bmatrix} H_{[0]} & H_{[N_1-1]} & \ldots & H_{[1]} \\ H_{[1]} & H_{[0]} & \ldots & H_{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ H_{[N_1-2]} & H_{[N_1-3]} & \ldots & H_{[N_1-1]} \\ H_{[N_1-1]} & H_{[N_1-2]} & \ldots & H_{[0]} \end{bmatrix},$$

where the blocks $H_{[k]}$ are circulant matrices which have the form

$$H_{[k]} = \begin{bmatrix} h[0,k] & h[N_0-1,k] & \ldots & h[1,k] \\ h[1,k] & h[0,k] & \ldots & h[2,k] \\ h[2,k] & h[1,k] & \ldots & h[3,k] \\ \vdots & \vdots & \ddots & \vdots \\ h[N_0-1,k] & h[N_0-2,k] & \ldots & h[0,k] \end{bmatrix}$$

and $k = 0, 1, \ldots, N_1 - 1$

B. The Cyclic Correlation

The cyclic correlation operation, denoted by $\circledcirc_N = \circledcirc_{N_0 \times N_1}$, over the space of two-dimensional signals $l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$ is defined as

$$\begin{array}{ll}\circledcirc_N : l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \times l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \longrightarrow l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \\ \qquad (x, g) \qquad \longmapsto x \circledcirc_N g,\end{array}$$

where $x \circledcirc_N g$ is defined by

$$(x \circledcirc_N g)[n_0, n_1] = \sum_{k_1=0}^{N_1-1} \sum_{k_0=0}^{N_0-1} x[k_0, k_1] g[\langle n_0 + k_0 \rangle_{N_0}, \langle n_1 + k_1 \rangle_{N_1}]$$

The cyclic correlation is not commutative but it satisfies the following properties: For every $x, y, z \in l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$ and $\alpha \in \mathbb{C}$ we have

1. $(x \circledcirc_N y) \circledcirc_N z = x \circledcirc_N (y \circledcirc_N z)$.
2. $x \circledcirc_N (y + z) = x \circledcirc_N y + x \circledcirc_N z$.
3. $\alpha(x \circledcirc_N y) = (\alpha x) \circledcirc_N x = x \circledcirc_N (\alpha y)$.

Similar to the space of one-dimensional signals, the space of two-dimensional signals is a non-commutative algebra with the operation of the cyclic correlation.

The cyclic correlation operator is defined by fixing a signal $g \in l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$

$$\begin{array}{llll}\circledcirc_N^g : & l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) & \longrightarrow & l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \\ & x & \longmapsto & \circledcirc_N^g\{x\} = x \circledcirc_N g\end{array}$$

The matrix representation of this operator, using an anti-lexicographic order with respect to the standard basis, is the following circulant matrix of circulant blocks:

$$C_N = \begin{bmatrix} C_{[0]} & C_{[1]} & \ldots & C_{[N_1-1]} \\ C_{[1]} & C_{[2]} & \ldots & C_{[0]} \\ \vdots & \vdots & \ddots & \vdots \\ C_{[N_1-2]} & C_{[N_1-1]} & \ldots & C_{[N_1-3]} \\ C_{[N_1-1]} & C_{[0]} & \ldots & C_{[N_1-2]} \end{bmatrix},$$

$$C_{[k]} = \begin{bmatrix} g[0,k] & g[1,k] & \ldots & g[N_0-1,k] \\ g[1,k] & g[2,k] & \ldots & g[0,k] \\ g[2,k] & g[3,k] & \ldots & g[1,k] \\ \vdots & \vdots & \ddots & \vdots \\ g[N_0-1,k] & g[0,k] & \ldots & gk[N_0-2,k] \end{bmatrix}$$

and . $k = 0, 1, \ldots, N_1 - 1$

C. The Hadamard Product

The Hadamard product over the space $l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$ is defined by

$$\begin{array}{ll}\odot_N : l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \times l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \longrightarrow l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \\ \qquad (S, T) \qquad \longmapsto S \odot_N T,\end{array}$$

where $(S \odot_N T)[n_0, n_1] = S[n_0, n_1] T[n_0, n_1]$.

Fixing a signal $T \in l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1})$, we define the Hadamard operator as

$$\begin{array}{lll}\odot_N^T : l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) & \longrightarrow & l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \\ \qquad S & \longmapsto & \odot_N^T\{S\} = S \odot_N T.\end{array}$$

D. The Reflexive Operator

The reflexive operator over the space of two-dimensional signals is defined by

$$\begin{array}{lll}\Re_N = \Re_{N_0 \times N_1} : l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) & \longrightarrow & l^2(\mathbb{Z}_{N_0} \times \mathbb{Z}_{N_1}) \\ \qquad x & \longmapsto & \Re_N\{x\},\end{array}$$

where

$$(\Re_N\{x\})[n_0,n_1] = x^{(-)}[n_0,n_1] = x[\langle N_0-n_0\rangle_{N_0}, \langle N_1-n_1\rangle_{N_1}] \\ = x[\langle -n_0\rangle_{N_0}, \langle -n_1\rangle_{N_1}].$$

The matrix representation of the cyclic reflexive operator $\Re_N$, using anti-lexicographic order with respect to the canonic basis Δ_N is the circulant matrix of circulant blocks:

$$\Re_N = \begin{bmatrix} \Re_{[0]} & \Re_{[1]} & \dots & \Re_{[N_1-1]} \\ \Re_{[1]} & \Re_{[2]} & \dots & \Re_{[0]} \\ \vdots & \vdots & \ddots & \vdots \\ \Re_{[N_1-2]} & \Re_{[N_1-1]} & \dots & \Re_{[N_1-3]} \\ \Re_{[N_1-1]} & \Re_{[0]} & \dots & \Re_{[N_1-2]} \end{bmatrix},$$

where the blocks $\Re_{[i]}$ are circulant matrices of the form

$$\Re_{[0]} = \begin{bmatrix} 1_{[0,0]} & 0_{[1,0]} & \dots & 0_{[N_0-1,0]} \\ 0_{[1,0]} & 0_{[2,0]} & \dots & 1_{[0,0]} \\ 0_{[2,0]} & 0_{[3,0]} & \dots & 0_{[1,0]} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{[N_0-1,0]} & 1_{[0,0]} & \dots & 0_{[N_0-2,0]} \end{bmatrix}$$

$$\Re_{[i]} = \begin{bmatrix} 0_{[0,i]} & 0_{[1,i]} & \dots & 0_{[N_0-1,i]} \\ 0_{[1,i]} & 0_{[2,i]} & \dots & 0_{[0,i]} \\ 0_{[2,i]} & 0_{[3,i]} & \dots & 0_{[1,i]} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{[N_0-1,i]} & 0_{[0,i]} & \dots & 0_{[N_0-2,i]} \end{bmatrix}$$

and $i = 1, \dots, N-1$.

E. The Cyclic Shifting Operator

The cyclic shifting operator $\mathcal{S}_N = \mathcal{S}_{N_0\times N_1}$ over the space of two-dimensional signals $l^2(\mathbb{Z}_{N_0}\times\mathbb{Z}_{N_1})$ is defined as

$$\mathcal{S}_N = \mathcal{S}_{N_0\times N_1} : \begin{array}{ccc} l^2(\mathbb{Z}_{N_0}\times\mathbb{Z}_{N_1}) & \longrightarrow & l^2(\mathbb{Z}_{N_0}\times\mathbb{Z}_{N_1}) \\ x & \longmapsto & \mathcal{S}_N\{x\} \end{array}$$

where $\mathcal{S}_N\{x\}[n_0,n_1] = [\langle n_0+1\rangle_{N_0}, \langle n_1+1\rangle_{N_1}]$.

The matrix representation of this operator using an anti-lexicographic order with respect to the standard basis Δ_N of the space $l^2(\mathbb{Z}_{N_0}\times\mathbb{Z}_{N_1})$ is a matrix of circulant blocks with circulant blocks:

$$\mathcal{S}_N = \begin{bmatrix} \mathcal{S}_{[0]} & \mathcal{S}_{[N_1-1]} & \dots & \mathcal{S}_{[1]} \\ \mathcal{S}_{[1]} & \mathcal{S}_{[0]} & \dots & \mathcal{S}_{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{S}_{[N_1-2} & \mathcal{S}_{[N_1-3]} & \dots & \mathcal{S}_{[N_1-1]} \\ \mathcal{S}_{[N_1-1} & \mathcal{S}_{[N_1-2]} & \dots & \mathcal{S}_{[0]} \end{bmatrix},$$

where each block $\mathcal{S}_{[i]}$ is a circulant matrix of the form

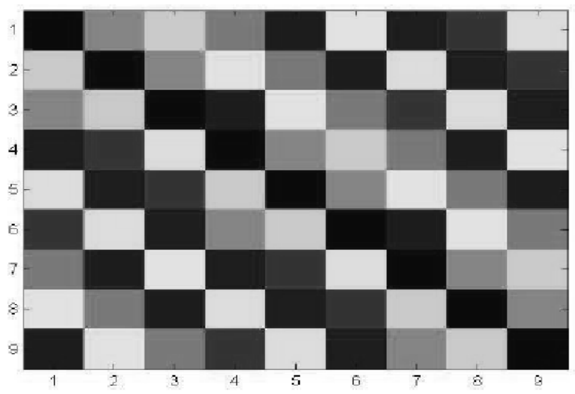


Figure 1. Matrix representation of the cyclic convolution operator

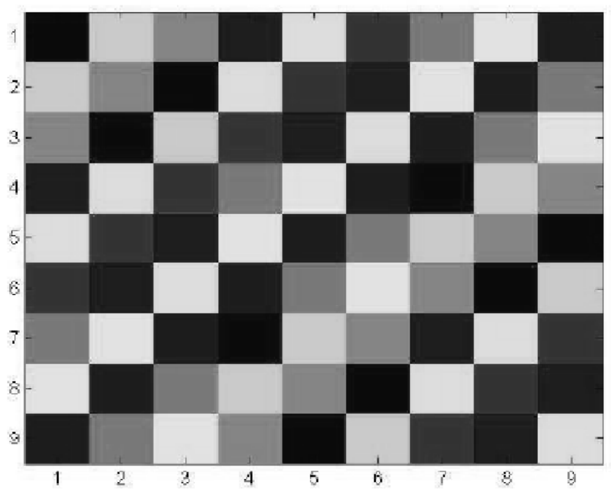


Figure 2. Matrix representation of the cyclic correlation operator

$$\mathcal{S}_{[0]} = \begin{bmatrix} 0_{[0,i]} & 0_{[N_0-1,i]} & \dots & 0_{[1,i]} \\ 0_{[1,i]} & 0_{[0,i]} & \dots & 0_{[2,i]} \\ 0_{[2,i]} & 0_{[1,i]} & \dots & 0_{[3,i]} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{[N_0-1,i]} & 0_{[N_0-2,i]} & \dots & 0_{[0,i]} \end{bmatrix},$$

when $i = 0, 1, \dots, N_i - 1$ and $i \neq 1$, and

$$\mathcal{S}_{[1]} = \begin{bmatrix} 0_{[0,1]} & 0_{[N_0-1,1]} & \dots & 1_{[1,1]} \\ 1_{[1,1]} & 0_{[0,1]} & \dots & 0_{[2,1]} \\ 0_{[2,1]} & 1_{[1,1]} & \dots & 0_{[3,1]} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{[N_0-1,1]} & 0_{[N_0-2,1]} & \dots & 0_{[0,1]} \end{bmatrix}.$$

Using an anti-lexicographic order, it is possible to get the matrix-graphical representation of the cyclic convolution operator [4] as is indicated in Figure 1. This representation helps to visualize the matrix of circulant blocks with circulant blocks.

In a similar way, it is possible to get the matrix-graphical representation of the cyclic correlation operator, as is shown in the following figure:

F. Multidimensional Signal Processing

Multidimensional Signal Processing deals with the formulation of theoretical methods for the treatment of multidimensional signals in order to extract important information for a user. The treatment of the multidimensional signals is mostly of an algorithmic nature, which means that the treatment is for creating well-defined procedures for solving a problem in a finite number of steps. A multidimensional signal is any signal which admits a mathematical representation as a numeric function of several independent variables. It is important to distinguish between these signals and a true physical signal, which may be conveying information about multiple observable quantities of physical entities at the same time. For example, a true physical signal may be conveying information about the temperature and pressure of a physical entity. However, this information might not be easily translated into a mathematical formulation. A numeric function of several independent variables can be denoted as $f(x, y, z, \ldots)$.

5. DIGITAL IMAGE PROCESSING

Interest in Digital Image Processing (DIP) methods stems from two principal application areas such as the improvement of pictorial information for human interpretation, and the processing of image data for storage, transmission, and representation for autonomous machine interception [6].

DIP refers to processing digital images by means of a digital computer. A digital image may be defined as a 2-D function $f(x, y)$ where x and y are spatial (plane) coordinates, finite and discrete, as well as the amplitude of f at any pair of coordinates (x, y), called the intensity or gray level of the image at that point.

A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, and pixels.

The number of gray levels typically is an integer power of 2, say $L = 2^k$ where k is the number of bits of quantization. When an image can have 2^k gray levels, it is common to refer to the image as a "k-bit image." For example, an image with 256 possible gray-level values is called an 8-bit image.

The information contained in images can be represented in different ways, with the spatial and the frequency representation being the best known. Spatial representation refers to the representation of images as two dimensional (2D) arrays, in which each element of the matrix is called a **pixel**. For this representation, the matrix notation is commonly used. Figure 3 shows an example of a spatial representation of a very simple image of a unitary impulse. Black pixels are represented by zeros and the white pixels by ones. In the lower left side appears the 3D representation of the unitary impulse. If the image contains $M \times N$ pixels, it is represented by an $M \times N$ matrix. The representation in the frequency domain of a digital image is obtained by applying the Discrete Fourier Transform (DFT) to the matrix representing the image.

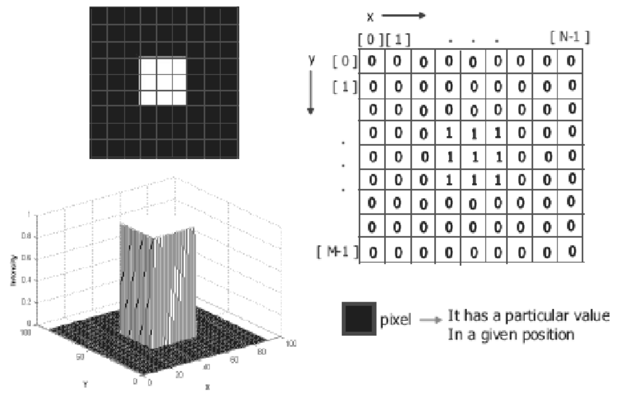


Figure 3. Spatial Representation of Images

Gray scale images were used in this research. In this type of images, intensities of pixels are shades of gray ranging from black to white. This can be viewed clearly in Figure 4. One of the problems confronted is to find the appropriated range of gray values to graphically represent the output from the signal operators, especially if those outputs are negative numbers. Jahne [5] gives an approach to solve this problem as follows: Each pixel in images of 256 gray values occupies 8 bits, which causes negative pixels to take large positive values. "In an 8-bit representation, it is possible to convert unsigned numbers into signed numbers by subtracting 128:

$$q' = (q - 128) \bmod 256, \quad 0 \leq q' \leq 256.$$

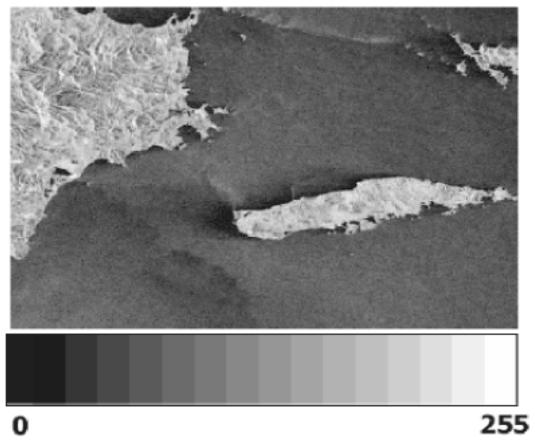


Figure 4. Grayscale Image

Essentially, gray values are regarded in this representation as a deviation from a mean value. In order to have a better display on the screen, the gray values must be converted again to unsigned values by the inverse point operation":

$$q = (q' + 128) \bmod 256, \quad -128 \le q' \le 128^{\cdot}$$

At this point, we will present a list of the implemented operators in JCID (Java Computational Image Developer). The classification was done according to the functionality of the operators.

1) Point Operators: These operators are used to modify the gray values at specific pixels. They can be applied to correct the image illumination and contrast enhancement: Absolute Value, Clamp, Color Convert.

2) Arithmetic Operators: The basic operations that could be done over images include sum, subtraction, multiplication, and division.

3) Spatial Operators: The function of these operators is to modify the position of the pixels:

- Crop. The operator cuts a portion of the image given the width and height of the section to be cropped.
- Resizing. The operator transforms the image by adding or subtracting pixels.
- Flipping. The operator makes a mirror of the image in a vertical or a horizontal direction.

4) Convolution Operators: These operators are based on the convolution operation of image processing: Sharpening, Blurring, Embossing, and Edge Detection.

5) Filtering Operations: These operators are also based on the convolution operation and are mostly used for image enhancement, highlighting, or hiding features of the image: Low-pass, High-pass, Laplacian, Gaussian, and 2-D Fourier Transform.

6) Complex Operators: These operators have been implemented especially for those images whose pixels are complex numbers: Cyclic Convolution, Cyclic Correlation, Conjugate, Phase, Hadamard Product, and Shifting.

6. WEB-BASED COMPUTATIONAL IMAGE PROCESSING

A. Software Tools used for Implementation

The implementation of all the components of the system was done using the programming language Java. It was chosen because of the following features [6]:

It is object-oriented, which means that the programming of the state and behavior of a system is done by programming the state and behavior of the objects that compound the system.

It is platform-independent, or capable for running on different platforms such as Windows, Linux, or Macintosh.

It is the most common programming language for web applications because it is the foundation of many developing frameworks.

Between the several development projects of the Java community, there are two important API's specialized for the treatment of images. The first one is the **Java Advanced Imaging API**, a set of interfaces for manipulating images, and **Java Image I/O API** which provides the management of image files stored in a local file system or distributed across the network [6]. Considering that it was assumed that the input data are complex signals, meaning that each sample is a complex number (formed by a real number and an imaginary number), an open source library of operations that supports complex numbers was acquired. This library is called Flanagan's Java Library (flanagan.jar), created by Dr. Michael Thomas Flanagan from the University College London. It includes basic arithmetic operations of complex numbers (addition, subtraction, multiplication, and division), trigonometric operations, and special mathematical functions.

For the implementation of the web environment, **Java Server Faces**, commonly known as *JSF*, was chosen. JSF is a project from the Java Community Process [7] whose applications are standard Java Web Applications, for which JSF defines three layers: a component architecture which defines a common way to build user interface (UI) widgets, a standard set of UI widgets (such as text boxes, list boxes, tabbed panes, and data grids), and an application infrastructure which uses an HTTP protocol for communication via the Servlet API, it uses JavaServer Pages (JSP) as its display technology, and it uses JavaBeans for exposing properties and event handling [8].

Many distributions of JSF from different companies such as Oracle, IBM, and Apache Project, exist. In this paper, the open source implementation of JSF developed by Apache Project, called **MyFaces**, was used. **Apache Tomcat** was used as the servlet container which is used in the "official Reference Implementation for the Java Servlet and JavaServer Pages technologies, providing an environment for Java code to run in cooperation with a web server." Tomcat is cross-platform, running on any operating system that has a Java Runtime Environment.

The JFree Chart is a free Java chart library that makes it easy for developers to display professional quality charts in their applications. This library consists of an API that supports a wide range of chart types, a flexible design that is easy to extend, and targets both server-side and client-side applications. It is distributed under the terms of the GNU Lesser General Public License (LGPL), which permits its use in proprietary applications.

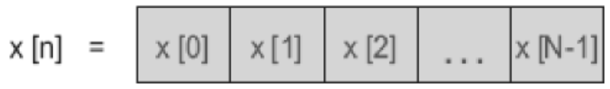
B. Object-Oriented_based Signal Operator Approach

Object-Oriented programming (OOP) is a programming paradigm that uses "objects" to design applications. "An object is a kind of self-sufficient entity that has an internal *state* (the data it contains) and that can respond to *messages* (calls to its subroutines). The OOP approach to software engineering is to start by identifying the objects involved in a problem and the messages that those objects should respond to. The program that comes out of it is a collection of objects, each with its own data and its own set of responsibilities. "The objects interact by sending messages to each other" [9]. The Object-Oriented-Based Signal Operator Approach (OOSO) is a methodology used for modeling aspects of the Computational Information Processing (CIP) Framework, which comprises the set of inputs, the set of outputs, and the rules of composition.

1) Data Input Structures: The CSP system has two basic structures of data input. The one-dimensional (1-D) signals of dimension N and the two-dimensional (2-D) signals of dimension $M \times N$. A 1-D signal is represented mathematically as a sequence of samples like this:

$$x[n] = \{x[0], x[1], x[2], \ldots, x[N-1]\}.$$

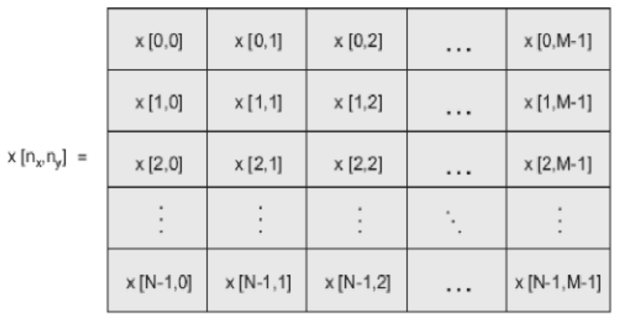
Graphically, this sequence can be seen as a 1-D array, or as a *vector* data structure:



$$n = 0, 1, 2, \ldots, N-1, \; n \in \mathbb{Z}_N$$

Figure 5. A 1-D signal represented as vector

A (2-D) signal is represented mathematically as an $M \times N$ matrix. Graphically, this can be seen as a 2-D array data structure, as presented in Figure 6.



$$n_x = 0, 1, 2, \ldots, N-1, \; n_x \in \mathbb{Z}_N;$$
$$n_y = 0, 1, 2, \ldots, M-1, n_y \in \mathbb{Z}_M$$

Figure 6. A 2-D signal represented as matrix

Making an extension to OOP, both the 1-D and the 2-D signals each represent a class with its own attributes and methods. Figure 7 illustrates the attributes that identify each class.

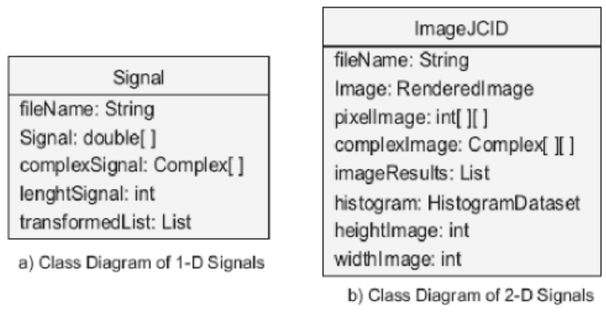


Figure 7. Class Diagrams of 1-D and 2-D signals

For 1-D signals, the class has been called **Signal**. The attribute *complexSignal* of *Complex*[ ] type is a vector that contains the complex samples read from the data file. If the data are just real numbers, the imaginary part of each sample of the complex array is set to be zero. Each element of the array is an object of the class *Complex* from Flanagan's jar library. Figure 8 shows the class diagram of the *Complex* class. For each loaded signal there is a *List* of all the transformations resulting from the action of the operators over the signal. This list is represented by the attribute *transformedList* of the *Signal* class.

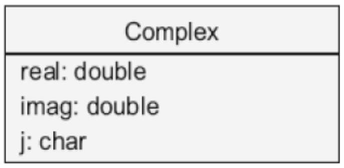


Figure 8. Class Diagrams of the Complex class

The class of 2-D signals has been called *ImageJCID*. The attribute of *RenderedImage* type represents the image as a grid of pixels. Images in .jpg, .gif, .bmp, and .tiff formats are easily loaded in this type of data. With this representation, it is possible to obtain the values of the pixels as 2-D arrays, with *int* data type or as a matrix of complex numbers of type *Complex*[ ][ ]. As in the 1-D case, each image has its corresponding *List* of images that come from the action of the operators.

2) Operator Structures: Operators are systems able to transform an input signal to produce an output signal. For the CIP, they were classified in unary operators and binary operators. Unary operators take only one signal on the input and produce one signal on the output. Figure 9 shows an example of a unary operator such as the Discrete Fourier Transform, and an example of a binary operator such as addition.

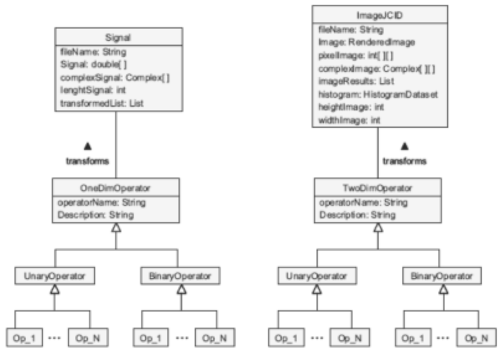


Figure 9. Example of Unary and Binary Operators

Such a distinction of the unary and binary operators apply to both 1-D signals and 2-D signals. Figure 10 is the class diagram for both kinds of operators. The set of implemented operators is divided into two groups: the *OneDimOperators* used to transform 1-D signals, and the *TwoDimOperators* used to transform 2-D signals. Each group is classified as *UnaryOperator* or *BinaryOperator*, which in turn, are composed of all the different operators that fit into to each category.

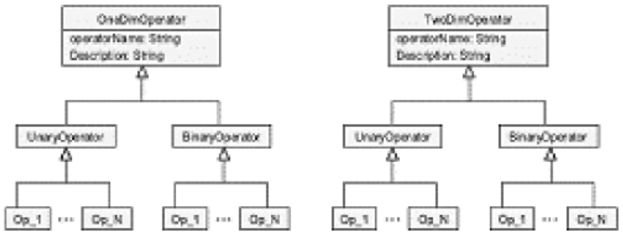


Figure 10. Class Diagrams for Operators

So, the relationship of the transformation of the Operator classes on the Signal classes can be seen in Figure 11.

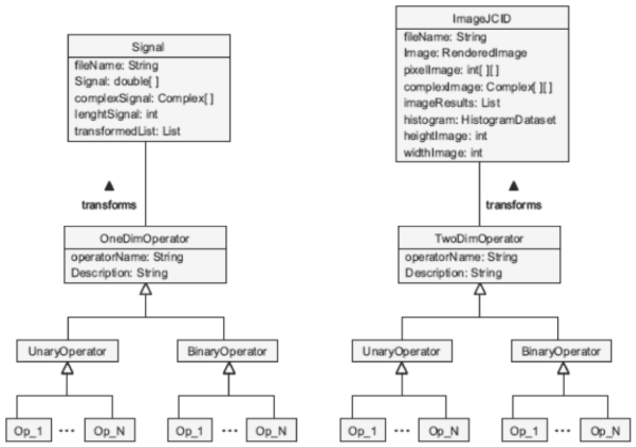


Figure 11. Class Diagram of the Action of Operators on Signals

C. Web Application Architecture

A web application could be defined as a web system where user input (navigation and data input) affects the state of the logic of the system. The basic architecture of a web application includes browsers, a network, and a web server. In [10] Conallen also states that "a web application uses a web site as the front end to a more typical application." Figure 12 shows the relationship between the main components of a web application.

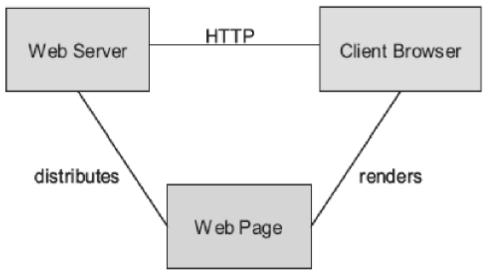


Figure 12. Components of a Web Application

The connection between the client and server only exists during a page request. Once the request is complete, the connection is broken. All the activity on the server occurs during the page request [10].

The client/server architecture has been one of the most used network architectures to build web applications. In this architecture, the client is the requester of services and the server is the provider of such services. The client could be interpreted as a desktop application installed on several numbers of workstations, which make requests to a single application installed on the server. This kind of interaction is known as *two-tiered architecture*.

D. Multired Applications

Though many variations are possible, a web application is commonly structured as a three-tiered application, which is the most common example of multi-tier architecture. In its most common form, a web browser is the first tier, an engine created using some dynamic web content technology (e.g., CGI, PHP, Java Servlets, or Active Server Pages) is the middle tier, and a database is the third tier. The web browser sends requests to the middle tier, which services them by making queries and updates on the database and generating a user interface. Figure 13 shows the interaction between the three elements of this architecture.

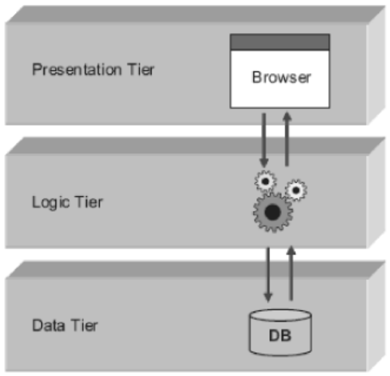


Figure 13. Three-Tier Architecture

In three-tier applications, the Presentation tier never communicates directly with the Data tier. Additional to this configuration, another component has been added by the **Model-View-Controller (MVC)** design pattern. This pattern inserts an intermediate component called the **controller**, which is located between the Presentation tier and the Logic tier. The function of the controller is to process and respond to events, typically user actions, and it invokes changes on the model [11]. Figure 14 illustrates the standard Model-View-Controller Architecture.

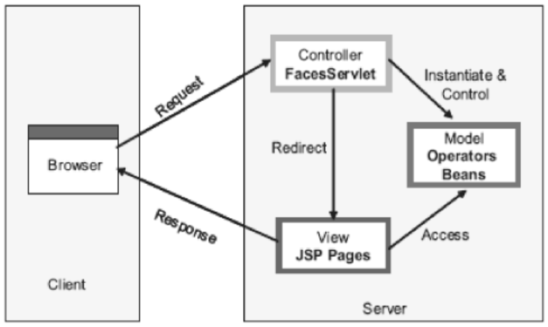


Figure 14. Model-View-Controller Architecture

In section 5, it was explained that JavaServer Faces was chosen to implement this architecture for the web-based computational signal processing. This development framework also uses the **J2EE (Java Platform, Enterprise Edition)** framework of Java Sun Microsystems. “The J2EE platform offers a multi-red distributed application model, reusable components, a unified security model, flexible transaction control, and web services support through integrated data interchange on Extensible Markup Language (XML)-based open standards and protocols” [11]. In the context of the web-based computational signal processing system, the basic structures of the set of multidimensional signals and the set of implemented operators reside in the Model tier of the MVC model. They are invoked by the *FacesServeletContext* instance of the Controller tier. The next section explains in more detail how each one of the modules that compose the environment was developed.

7. JCID: JAVA COMPUTATIONAL IMAGE DEVELOPER

In order to demonstrate the functionality of the theory presented in section 1, it was necessary to implement an application that could help scientists that work with multidimensional signals [12]. This application was named **JCID: Java Computational Image Developer**. At first, it was thought that this tool would only process 2-D signals, specifically signals that could be represented as images, but in an advanced stage of the development of the project, the alternative of also processing 1-D signals was considered, since some raw data in this dimension was available to the research group.

This application is part of the general project "**WALSAIP: Wide Area Large Scale Automated Information Processing**" of the Institute for Computing and Informatics Studies of the Department of Electrical and Computer Engineering. The following research groups participate in this project:

The Advanced Data Management Group (ADMG)

The Automated Information Processing Group (AIPG)

The Human Computer Interfaces Group (HCIG)

The Hydro-Ecological Research Group (HERG)

The Network Communication Infrastructure Group (NCIG)

The Parallel and Distributed Computing Group (PDCG)

The technological infrastructure of this project is composed by a set of 7 servers, each one with 28 disks of 300 Gigabytes, and each research group has at least one of the servers. This infrastructure belongs to the network core of the INEL/ICOM department which acts as an intermediate between the internal network and the exterior world (the Internet and another dependencies of the UPRM). Figure 15 represents a scheme of this infrastructure.

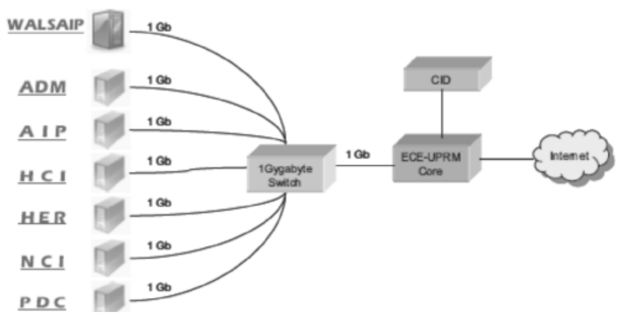


Figure 15. Infrastructure of WALSAIP Project

The WALSAIP project has an agreement with the Jobos Bay Reserve, a dependency of NOAA located in the municipality of Salinas in Puerto Rico, dedicated to the monitoring of the hydrological and ecological factors of the coastal zone. In that reserve some environmental sensors have been set which are collecting information which is being stored in a database. Figure 16 shows a picture of the resources shared by the Jobos Bay Reserve and the UPRM.

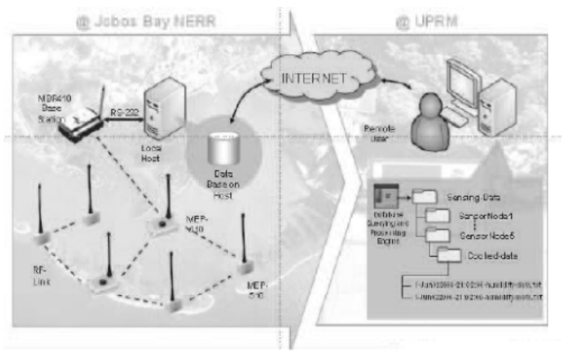


Figure 16. Jobos Bay Reserve

The development of the JCID application was done in two phases. In the first phase, a stand-alone version was developed in order to make a "proof of concept" of the implemented operators and the integration in a graphical user interface. The second phase was the implementation as a web application, making use of the network infrastructure described above.

In the design of the application some rules of Usability Engineering and Human Computer Interaction were taken into account in order to provide a user friendly interface that facilitates the work of the scientists. Figure 17 is a screenshot of the graphical user interface of JCID.

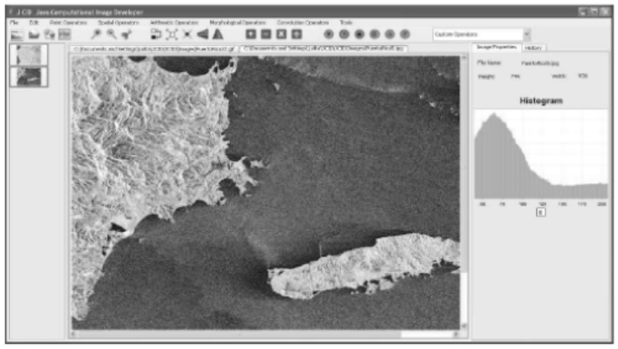


Figure 17. Graphical User Interface of JCID

In the menu bar of JCID appear the operators available to process the signals. The operators were classified by the type of functionality, for instance, performing arithmetic operations over the signals. Also, there is a menu of arithmetic operators, like addition, subtraction, multiplication, division, exponential, and logarithmic. If a user is working with an image and wants to enhance it, there is a menu of filtering operators, like edges detection, low pass filter, high pass filter, etc. The user can apply a sequence of operators over the signal, and once an optimal result is obtained, the sequence can be encapsulated in a new operator in order to avoid the repetition of steps in another session which helps the user to make a repository of his/her own operators. Figure 18 illustrates how the sequence of actions is presented to the user.

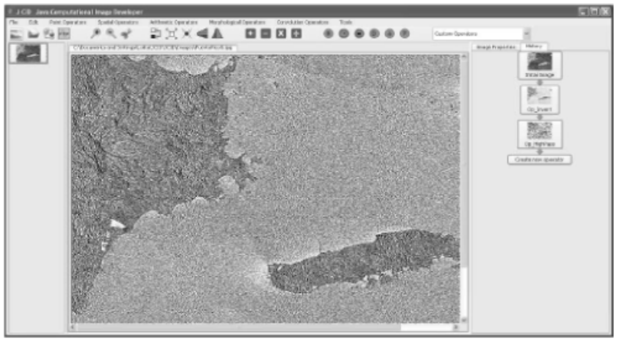
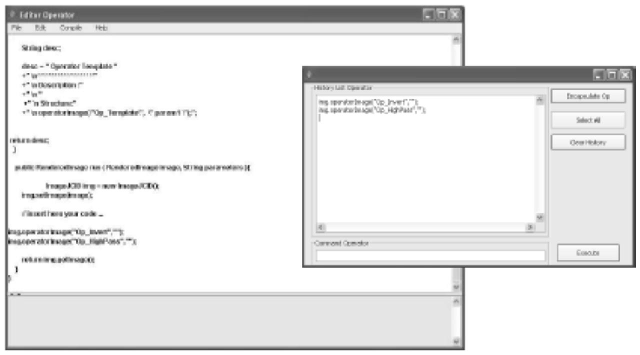


Figure 18. Encapsulation of Operators

In the right panel of the figure the sequence of operators appears represented as thumbnails of the obtained results. At the end of the sequence, there is a command button to execute the encapsulation of the sequence.

In Figure 19 the dialog windows where the list of operators appears which a user may want to encapsulate, and a small editor can be seen, if the user wants to add more functionalities to the new operator. The operators Dilation and Erosion that appear in the Morphological Operators menu were implemented using the connection between MatLab and Java using the JLab library, described in section 3.2. The operator of the Short Time Fourier Transform (STFT) was also deployed with the Javabuilder toolbox, and it was made by using a trial version of the toolbox.

Between other functionalities of JCID, there is an interface to connect to the database that collects the data from the sensors. This interface helps to make queries according to the time interval the user wants to analyze. Results of the queries are stored in files of data and metadata. Data files contain the measurements of the sensors, and metadata files contain the information related to the sensor, its location, and the settings for the sampling of the data. Figure 20 shows a screenshot of the interface.



```
String desc;

desc = " Operator Template "
+" \n*****************"
+" \n Description :"
+" \n "
+" \n Structure:"
+" \n operatorImage(\"Op_Template\", \" param1 \");";

return desc;
}

public RenderedImage run ( RenderedImage image, String parameters ){

        ImageJCID img = new ImageJCID();
    img.setImage(image);

    //Insert here your code ...

img.operatorImage("Op_Invert","");
img.operatorImage("Op_HighPass","");

    return img.getImage();
  }
}
```

```
img.operatorImage("Op_Invert","");
img.operatorImage("Op_HighPass","");
```

Figure 19. Edition and Compilation of new Operators

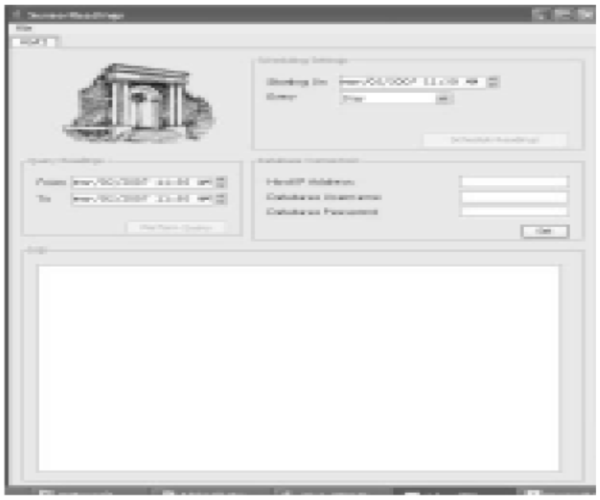


Figure 20. Connection with the sensors Database

Since it was considered that a user could have data files in other servers o workstations in the network, an open source application was integrated into JCID that allows one to open an FTP connection. This application is called **J-FTP**, developed by *JMethods Inc*. Figure 21 is a screenshot of J-FTP.

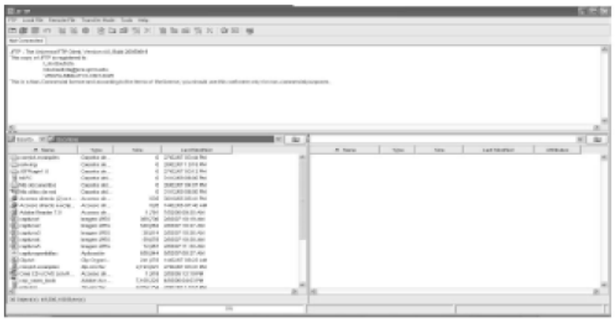


Figure 21. FTP connection using J-FTP

The web version of JCID has been deployed in the WALSAIP server of the project. It contains the same functionalities that were developed in the stand-alone version. In order to conserve the same graphical configuration of the stand-alone, the user interface of the web version was coded with JavaServer Faces using the tag libraries released in the MyFaces distribution of Apache Project. Figure 22 is a display of the web version.

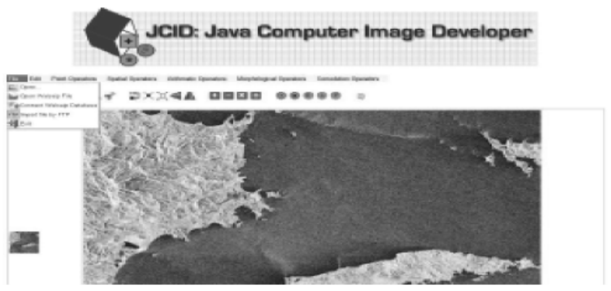


Figure 22. Graphical User Interface of Web-JCID

8. CONCLUSIONS

This research fits into the development of a computational signal algebra framework for modeling, simulation, and processing applications using digital images. The Computational Signal Processing Environment has been designed to be a useful tool for geologists, hydrologists, and scientists in related areas. It provides a friendly access to basic and advance operators, with encapsulation capabilities. One of the most important features of the environment is the portability offered by Java implementation, which also facilitates the availability of deploying the source code for potential users and developers. Finally, the integration of Java with scripting languages like MatLab diminishes the time consumed in coding and provides a better tool for scientists who do not have experience in programming languages.

REFERENCES

[1] DAVIS, P. J., Circulant Matrices, John Wiley and Sons, 1979.

[2] MITRA, S. K., Digital Signal Processing: A Computer-Based Approach, Mc Graw Hill, 2001.

[3] PAREDES, M., RODRÍGUEZ, D. and VILLAMIZAR, J., La estructura algebraica del espacio de señales unidimensionales, Revista Integración, vol. 23, pp. 15-39, 2005.

[4] VILLAMIZAR, J., A Computational Signal Algebra Framework for Image Interferometry Processing Applications, Master Thesis, Electrical and Computer Engineering Department; University of Puerto Rico at Mayaguez, 2006.

[5] JÄHNE B., Digital Image Processing, Springer, 2005.

[6] Java Technology, http://www.sun.com

[7] Java community process, http://jcp.org/en/home/index.html

[8] MANN K. D., JavaServer Faces in Action, Manning Publications Co., 2005, pp. 16-37.

[9] ECK D. J., Introduction to programming using java, Hobart and William Smith Colleges, http://math.hws.edu/javanotes, 2006.

[10] CONALLEN J., Modeling Web application architectures with UML, Communications of the ACM, vol. 42, no. 10, pp. 63-70, 1999.

[11] BODOFF, S., ARMSTRONG, E., BALL, J., BODE, D., EVANS I., GREEN, D., HAASE, K. and JENDROCK, E. The J2EE Tutorial, Addison Wesley, 2004, pp. 2-10.

[12] BAUTISTA, L. X., Web-based Data Processing for Environmental Surveillance Monitoring Applications, Master Thesis, Electrical and Computer Engineering Department, University of Puerto Rico at Mayaguez, 2007.