

# Ventanas deslizantes por bloques para la implementación online de la transformada wavelet discreta

Antonio Cedeño Pozo<sup>1</sup> y Rafael Trujillo Codorníu<sup>2</sup>

<sup>1</sup> Universidad de las Ciencias Informáticas, Centro de Informática Industrial, Cuba

<sup>2</sup> Instituto Superior Minero Metalúrgico de Moa, Departamento de Matemática, Cuba

acedeno@uci.cu, rtrujillo@ismm.edu.cu

**Resumen.** En este trabajo se propone un esquema para la implementación *online* de la Transformada Wavelet Discreta. Se introducen mejoras en cuanto a tiempo de ejecución respecto al método de ventanas deslizantes tradicional. En la propuesta se realiza una modificación a la definición de la ventana de datos propuesta en el esquema original. Las pruebas realizadas muestran que el algoritmo propuesto es más rápido que el de ventanas deslizantes tradicionales.

**Palabras clave.** Wavelet, online, ventanas deslizantes, programación dinámica.

## Sliding Windows by Blocks for Online Wavelet Discrete Transform Implementation

**Abstract.** In this paper we propose an online Wavelet Discrete Transform implementation scheme. Our proposal improves the execution time compared to the traditional sliding window method. Also, we modify the definition of the data window concept given in the original scheme. The experiments we performed show that the runtime cost of the proposed algorithm is better than that of the traditional sliding window method.

**Keywords.** Wavelet, online, sliding window, dynamic programming.

## 1. Introducción

La Transformada Wavelet Discreta (WDT) es una herramienta matemática relativamente reciente, entre sus múltiples utilidades sobresale en áreas como la reducción de ruido, la compresión de datos y el reconocimiento de

patrones. El concepto clave de la WDT es la adaptación del análisis en tiempo y frecuencia de forma simultánea, ésta característica la hace más factible para el tratamiento de señales no estacionarias que otras transformadas, como la Transformada de Fourier [1]. De forma natural esta transformada se utiliza para el tratamiento de señales una vez que han sido almacenadas, en ese sentido se han publicado un número importante de trabajos que ponen de manifiesto la eficacia de la WDT en este tipo de tratamiento [2, 3, 4], que en este material denominaremos *offline*. Estos buenos resultados para el tratamiento *offline* de señales han motivado a que algunos investigadores utilicen la WDT para el tratamiento *online*. Entiéndase por *online* el tratamiento sobre la señal a medida que se van adquiriendo las muestras. La utilización de la WDT presupone el aprovechamiento de un conjunto de propiedades que caracterizan a las señales en el espacio de la transformada, que en este caso muchos autores llaman *coeficientes wavelets*, por tanto los esquemas de aplicación constan de tres pasos fundamentales:

1. Descomposición de la señal en *coeficientes wavelets* mediante la WDT.
2. Tratamiento de los coeficientes según el área específica de aplicación.
3. Reconstrucción de la señal mediante la WDT inversa.

En el caso *online* las muestras se deben tratar por separado a medida que van siendo adquiridas, para evitar retardos en la salida los pasos mencionados anteriormente se deben

realizar sobre un histórico de datos que incluye a la muestra más reciente o muestra actual. En [5] se propone un esquema de este tipo, el mismo está basado en un mecanismo de ventanas deslizantes, esta propuesta ha sido aplicada con buenos resultados en [5, 6, 7] para la reducción de ruido en diferentes tipos de señales. El principal inconveniente que tiene este enfoque es el tiempo de ejecución de los algoritmos, pues para cada muestra se deben recalcular los *coeficientes wavelets*, elevando los tiempos de respuesta y comprometiendo la utilización de esta técnica en algunas áreas de aplicación con restricciones de tiempo importantes. En ese sentido en este material se realizan modificaciones al esquema de ventanas deslizantes (MW), y se propone un esquema que mejora el costo computacional, la nueva propuesta se denomina ventana deslizante por bloques (MWB).

## 2. Desarrollo

### 2.1. Implementación de la WDT: esquema lifting

Una de las principales dificultades que enfrentan los ingenieros a la hora de trabajar con transformadas es la complejidad que esto implica, algunas implementaciones de la WDT presuponen un amplio conocimiento matemático, por ejemplo en [8,9] se emplea la teoría de Fourier como instrumento para las construcciones *wavelet*. El esquema *lifting* permite implementar la WDT de una forma sencilla y eficiente, es un esquema natural, con todas las ventajas de los métodos tradicionales y con menor costo computacional. En el caso más simple consta de tres fases: *división*, *predicción* y *actualización*. El esquema parte de un conjunto de datos muestreados dado por  $X_0 = \{x_0, x_1, \dots, x_{N-1}\}$ , en la primera fase este conjunto se divide en dos subconjuntos  $C_{-1}$  (*coeficientes wavelets*) y  $A_{-1}$  (*aproximaciones*), formados por los elementos de subíndice impar y par de  $X_0$  respectivamente. En la fase de predicción se trata de predecir los elementos de  $C_{-1}$  a partir de  $A_{-1}$  usando algún tipo de correlación presente en los datos originales.

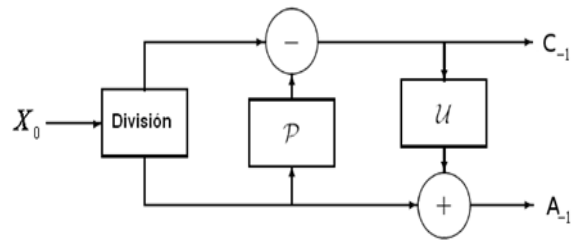


Fig. 1. Esquema lifting

Si se pudiera encontrar un operador de predicción  $P$ , que garantice que  $C_{-1} = P(A_{-1})$ , entonces se pudiera reemplazar el conjunto  $X_0$  por  $A_{-1}$  pues se podría predecir la parte faltante para reconstruir  $X_0$ . En la práctica es imposible encontrar una predicción exacta que sea independiente de los datos iniciales, pero si  $P(A_{-1})$  es bastante cercano a  $C_{-1}$ , entonces se podría reemplazar cada elemento de  $C_{-1}$  por la diferencia entre su valor y la predicción, es decir  $C_{-1} = C_{-1} - P(A_{-1})$ . Es de suponer que si la predicción es razonable entonces esta diferencia contiene mucha menos información que el conjunto original. En la fase de actualización se tratan de conservar en  $A_{-1}$  algunas propiedades globales de los datos del conjunto original mediante un operador  $U$  de la siguiente forma:  $A_{-1} = A_{-1} + U(C_{-1})$  (ver Figura 1).

El esquema puede ser iterado, se divide  $A_{-1}$  en dos subconjuntos  $A_{-2}$  y  $C_{-2}$ , y se aplican nuevamente los operadores  $P$  y  $U$ . Luego de  $n$  pasos el conjunto original puede ser reemplazado por su representación *wavelet*:

$$\{A_{-n}, C_{-n}, C_{-n+1}, \dots, C_{-1}\} \quad (1)$$

Si el modelo de predicción es exitoso entonces la energía del vector obtenido se concentra en  $A_{-n}$  y los coeficientes deben ser pequeños. La transformada inversa se obtiene invirtiendo las operaciones del esquema.

### 2.2. Ventanas deslizantes

Se ha demostrado estadísticamente en muchas investigaciones que las técnicas de

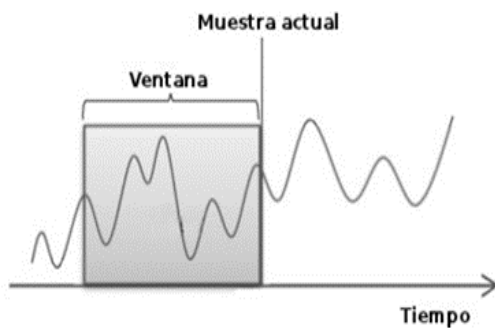


Fig. 2. Ventana de datos en MW

procesamiento de señales *offline* basadas en WDT pueden ser muy eficientes, sin embargo, dichas técnicas no pueden ser aplicadas tal cual para realizar procesamiento *online*, necesario para aplicaciones con requerimientos de algún tipo de tiempo real. Debido a esa razón se han propuesto algunos esquemas para adaptar las técnicas *offline* para el realizar procesamiento *online*, aprovechando el rendimiento de las primeras. Para lograr ese objetivo uno de los puntos críticos es la utilización de la WDT de forma efectiva para minimizar los retardos en las salidas. Una de las técnicas utilizadas es MV [5, 6, 7], que propone la utilización de una ventana de datos sobre la que se realiza de manera *online* la WDT y se actualiza (desplazando o deslizado) ante el arribo de una nueva muestra (ver Figura 2).

La ventana alrededor de la muestra actual  $x(i)$  se puede definir como:

$$W = \begin{cases} \{\text{Vacío}\}, & i < 1 \\ \{x(i-1), x(i-1+1), \dots, x(i)\}, & i \geq 1 \end{cases} \quad (2)$$

donde  $l$  es la longitud de la ventana. De la ecuación (2) se deduce que se necesitan  $l$  datos para formar la ventana, lo que es una precondition para comenzar a aplicar la WDT sobre los datos. Para atenuar algunos efectos indeseados que pueden ocurrir en los extremos de la señal, sobre todo alrededor de la muestra actual  $x(i)$ , se utilizan técnicas para prolongar la ventana, por ejemplo mediante una simetría (o espejo) como se observa en la Figura 3.

La cantidad de elementos de la prolongación  $p$  aumenta el tamaño de la ventana a  $l + p$ , y ésta

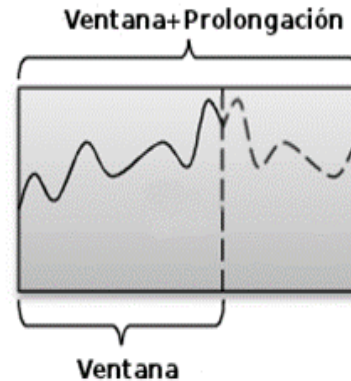


Fig. 3. Simetría o espejo

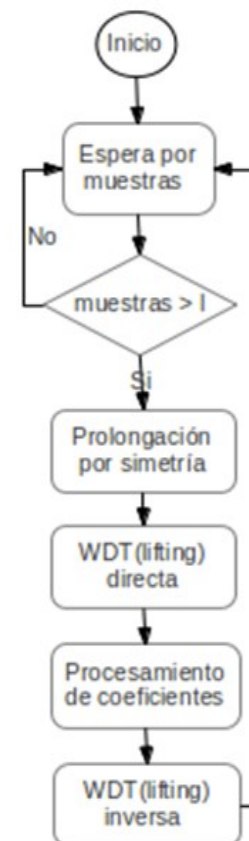


Fig. 4. Esquema MW

se expresa entonces mediante:

$$W = \left\{ \begin{array}{l} \{ \text{Vacío} \}, \quad i < l \\ \left\{ x(i-l), \dots, x(i), \overset{\text{Prolongación}}{\tilde{x}(i)}, \dots, x(i-p) \right\}, \quad i \geq l \end{array} \right\} \quad (3)$$

De esta forma, el algoritmo para el procesamiento de señales *online* aplicando la técnica MW se muestra en la Figura 4.

En esta técnica ante el arribo de una nueva muestra la ventana se desliza en una unidad de datos, y es necesario en cada paso realizar la transformadas wavelet directa e inversa, lo que implica un procesamiento computacional importante, impidiendo el empleo del esquema en algunos entornos.

### 2.3 Ventanas deslizantes por bloques

La idea de la propuesta de este material, esquema que denominamos MWB, se basa en principios de la programación dinámica, siguiendo el patrón de MW se tratan de reutilizar los *coeficientes wavelets* tanto como sea posible, evitar que todo el tiempo se tenga que recalcular la WDT ante la llegada de cada una de las muestras de la señal.

En MW se utiliza una prolongación de tamaño fijo, en el esquema que se propone la prolongación  $p$  varía desde  $p_{max}$  hasta  $p_{min}$ , que son variables que representan la máxima y mínima longitud respectivamente, que puede tomar la prolongación. De esa forma la nueva ventana  $W$  queda definida por:

$$\left\{ \begin{array}{l} \{ \text{Vacío} \}, \quad i < l \\ \left\{ x(i-l-m), \dots, x(i), \overset{p}{\tilde{x}(i)}, \dots, x(i-p_{max}) \right\}, \quad i \geq l \end{array} \right\} \quad (4)$$

siendo  $m = p_{max} - p$ .

Una vez definida la ventana de esta forma, el deslizamiento se realiza por bloques, de ahí proviene el nombre asignado al esquema. Cuando  $p$  alcanza el valor de  $p_{min}$ , la ventana se desplaza  $p_{max} - p_{min}$  elementos, que es el tamaño del bloque, y se restablecen los valores iniciales. De esta manera se pueden reutilizar los

coeficientes calculados para las  $l$  primeras muestras de la ventana en la primera iteración para el intervalo  $[p_{max} > p > p_{min}]$ , permitiendo reutilizar una parte importante de los coeficientes en cada bloque.

El esquema MWB está originalmente diseñado para el funcionamiento con la *wavelet Haar* (ver Figura 5), que por definición es la más sencilla y de menor complejidad computacional en su implementación *lifting*, cuyos operadores  $P$  y  $U$  se definen a partir de un término adyacente.

Para el correcto funcionamiento de MWB los parámetros deben seleccionarse teniendo en cuenta las consideraciones que se listan a continuación:

- $l$ : Este parámetro tiene que ver con la *wavelet* que se va a utilizar, en este caso la *Haar*, puede tomar valores en correspondencia con  $l = j \cdot 2^k$ ,  $j$  es un número entero cualquiera y  $k$  es la cantidad de niveles de descomposición *wavelets* deseados.
- $p_{max}$ : Se define de igual forma que  $l$ , pero  $k$  puede tomar cualquier valor a conveniencia del área de aplicación.
- $p_{min}$ : Puede tomar cualquier valor menor que  $p_{max}$ .

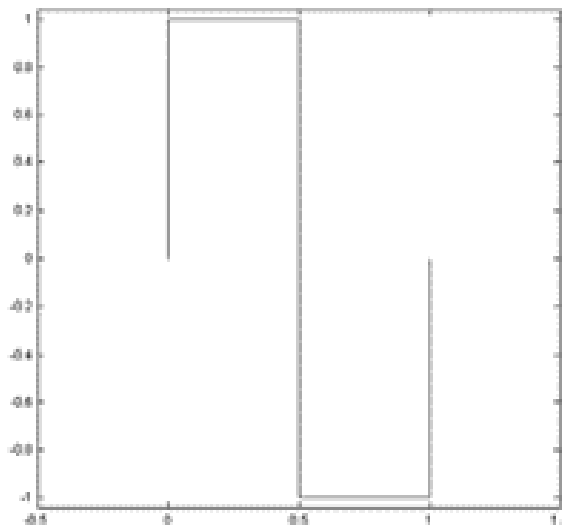


Fig. 5. Wavelet Haar

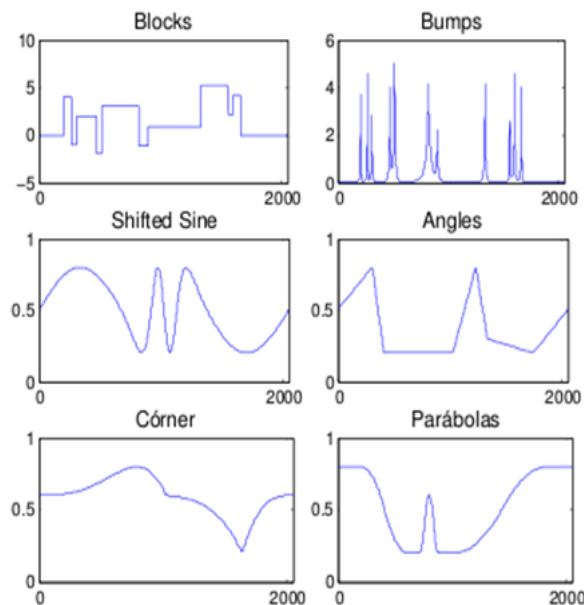


Fig. 6. Señales de prueba

### 3. Pruebas experimentales

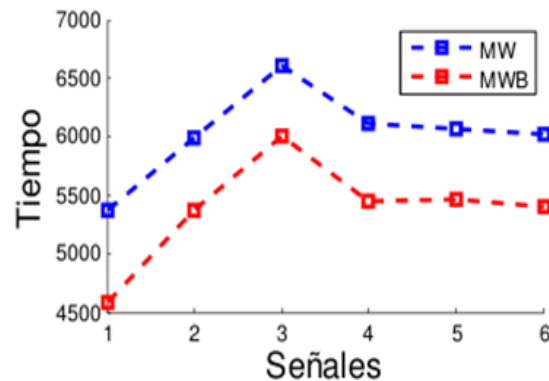
#### 3.1. Diseño del experimento

En el experimento se mide el tiempo de ejecución de los algoritmos MW y MWB para la descomposición *wavelet online* de un conjunto de señales sintéticas propuestas en [10, 11] (ver Figura 6). Cada señal está formada por 2048 muestras, la descomposición *online* se realiza 100 veces sobre cada señal y se mide el tiempo total de ejecución para los métodos en cuestión.

Los parámetros utilizados en el experimento fueron los siguientes:

**Tabla 1.** Tiempo de ejecución de MW y MWB en milisegundos para 100 corridas

	MW	MWB
Blocks	5,373	4,588
Bumps	5,986	5,366
Shifted Sine	6,604	6,009
Angles	6,120	5,454
Córner	6,069	5,469
Parábolas	6,019	5,396



**Fig. 7.** Tiempos de ejecución para 100 corridas (1: Blocks, 2: Bumps, 3: Shifted Sine, 4: Angles, 5: Córner, 6: Parábolas)

MW:  $l=192, p=64$

MWB:  $l=192, p_{max}=64, p_{min}=8$

Las pruebas se realizaron en un computadora personal con procesador Intel Core 2 Duo T5500, a una velocidad de 1.66 GHz, 1 GB de memoria RAM y sobre la distribución de Linux Ubuntu en su versión 11.10. Estos datos resultan de interés a la hora de comparar los tiempos de corrida de los algoritmos en otros ambientes de ejecución.

#### 3.2. Resultados

En la Tabla 1 se muestran los resultados del experimento, los mejores tiempos se resaltan en letra cursiva. Como se puede observar el esquema propuesto (MWB) tiene mejor tiempo de ejecución en todos los casos, o lo que es lo mismo, realiza el proceso de transformación *wavelet* (directa e inversa) *online* más rápido.

En la gráfica de la Figura 7 se puede apreciar la importante diferencia de tiempos de ejecución entre los algoritmos para cada una de las señales.

### 4. Conclusiones

Los resultados obtenidos a partir de las pruebas realizadas se muestran en la Tabla 1 y en la Figura 6, indican que el esquema propuesto (MWB) muestra mejor rendimiento en cuanto a tiempo de ejecución que el esquema MV.

## Agradecimientos

Este trabajo ha sido patrocinado de una forma u otra por la Universidad de las Ciencias Informáticas (Habana, Cuba) y el Instituto Superior Minero Metalúrgico de Moa (Holguín, Cuba).

## Referencias

1. **Dolobdjian, Ch., Fadili, J., & Huertas Leyva, E. (2002).** Classical low-pass filter and real-time wavelet-based denoising technique implemented on DSP a comparison study. *The European Physical Journal Applied Physics*, Vol. 20, pp. 135–140.
2. **Ma, Yinping & Huang, Yongxing (2012).** Adaptive Threshold Based on Wavelet Transform Fingerprint Image Denoising. *Computer Science and Electronics Engineering, International Conference*, Vol. 3, pp. 494–497.
3. **Li, Z., Ni, J., & Gu, X. (2012).** A Denoising Framework for ECG Signal Preprocessing. *International Conference on Internet Computing in Science and Engineering*, pp. 176–179.
4. **Nounou, M.N., Nounou, H.N., Meskin, N., Datta, A., & Dougherty, E.R. (2002).** Multiscale Denoising of Biological Data: A Comparative Analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 9, pp. 1539–1545.
5. **Xia, R., Meng, K., Qian, F. & Wang, Zh.-L. (2007).** Online Wavelet Denoising via a Moving Window. *Acta Automatica Sinica*, Vol. 33, No. 9, pp. 897–901.
6. **Khadem Olama, E. & Jazayeri-Rad, H. (2011).** Online Averaging Wavelet Denoising Method. *Computer Modeling and Simulation, UKSIM European Symposium*, IEEE Computer Society, pp. 202–204.
7. **Huang, Hsiao-Ping & Luo, Kuo-Yuan (2007).** On-Line Wavelets Filtering with Application to Linear Dynamic Data Reconciliation. *Industrial and Engineering Chemistry Research*, Vol. 46, No. 25, pp. 8746–8755.
8. **Chui, C.K. (1992).** *An introduction to wavelets*. Academic Press, San Diego, CA.
9. **Daubechies, I. (1998).** Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, Vol. 41, pp. 909–996.
10. **Donoho, D., Johnstone, I., & Johnstone, I.M. (1993).** Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, Vol. 81, 425–455.
11. **Antoniadis, A., Bigot, J., & Sapatinas, T. (2001).** Wavelet estimators in nonparametric regression: a comparative simulation study. *Journal of Statistical Software*, Vol. 6, pp. 1–83.

**Antonio Cedeño Pozo** es Ingeniero en Ciencias Informáticas, trabaja en el Centro de Informática Industria de la Universidad de las Ciencias Informáticas, Cuba.

**Rafael Trujillo Codorniú** es Licenciado en Matemática y Doctor en Ciencias Físico Matemáticas. Trabaja en el Instituto Superior Minero Metalúrgico de Moa, Cuba.

*Article received on 19/10/2012, accepted on 27/01/2014.*