

## **Cancelación de ruido en imágenes mediante la Transformada Wavelet Discreta usando un FPGA**

**Cuauhtémoc S. Carbajal, PhD**

Profesor Investigador, ITESM CEM, Atizapán de Zaragoza, Estado de México, México,  
carbajal@itesm.mx

**Enrique I. González, Ing.**

Estudiante, ITESM CEM, Atizapán de Zaragoza, Estado de México, México,  
egonzal@exatec.itesm.mx

### **Abstract**

El problema de la estimación de una señal que está corrompida con ruido aditivo ha sido de interés de muchos investigadores por razones tanto prácticas como teóricas. Muchos de los métodos tradicionales de cancelación de ruido (denoising) han estado usando métodos lineales, tales como el filtrado de Wiener. Recientemente, los métodos no-lineales, especialmente aquellos basados en onditas (wavelets) se han vuelto más populares, debido a que presentan varias ventajas sobre los métodos lineales. Este artículo presenta la implementación de la transformada wavelet discreta bidimensional para la cancelación de ruido en imágenes. Se implementó el algoritmo Db4 de Daubechies basado en lifting scheme en un FPGA Spartan-3 de Xilinx. Mediante la metodología de codiseño hardware/software se determinó que algunas partes fueran implementadas en hardware y otras en software en el procesador suave Microblaze con resultados muy alentadores.

### **Keywords**

Cancelación de ruido, Transformada Wavelet Discreta, Codiseño Hardware/Software, Lifting Scheme, FPGA.

### **1. Introducción**

La cancelación de ruido en imágenes tiene aplicaciones importantes en distintas áreas de la ciencia y la ingeniería. En este artículo presenta el diseño de un sistema de cancelación de ruido por medio de wavelets. Para ello, se implementó en software el algoritmo para la Db4 basada en lifting. Como se explica en la siguiente sección, una ventaja de utilizar este algoritmo es que se puede extender el caso unidimensional a dos dimensiones de manera simple, además de que las operaciones realizadas en el caso de la transformación inversa son un espejo de las llevadas a cabo en la transformación directa.

Al realizar la transformada wavelet discreta (DWT) en la imagen, se obtienen cuatro matrices, que corresponden respectivamente a la aproximación (A) y los detalles vertical (V), horizontal (H) y diagonal (D). Para llevar a cabo la cancelación de ruido, se eligió el método de Custom Thresholding, que se explica en la sección 3, el cual es aplicado a las matrices V, H y D, dejando intactos los coeficientes de la matriz A. Posteriormente, se lleva a cabo la transformación inversa con las matrices de coeficientes ya modificadas para obtener una reconstrucción de la imagen. El sistema se implementó en un FPGA Spartan-3. Básicamente consiste en un microprocesador suave (Microblaze)

con sus periféricos. El microprocesador utiliza como vía de comunicación el puerto serie de la tarjeta, a través de la cual recibe y envía las imágenes. Se programó el algoritmo de la Db4 en C y fue grabado en memoria de instrucciones para ser ejecutado por el Microblaze. Una explicación más detallada del sistema se da en el capítulo 4, y resultados del funcionamiento del sistema se muestran en la sección 5.

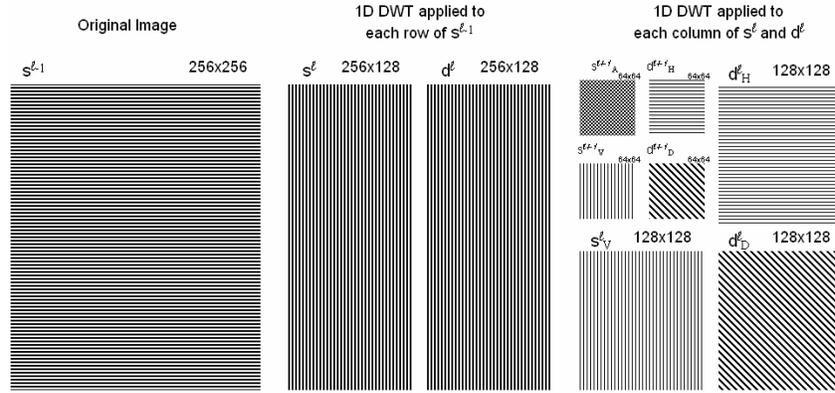
## 2. El algoritmo DWT Db4 bidimensional

El algoritmo para la Db4 bidimensional, se construye fácilmente a partir del algoritmo para la Db4 unidimensional basada en lifting scheme, de tal manera que es simplemente una extensión de las operaciones, lo cual es una ventaja algorítmica resultado de la factorización de los filtros de la Db4 en lifting steps. Para comenzar, consideremos el caso unidimensional cuyas ecuaciones se muestran en la figura 1a. En ellas se pueden diferenciar los tres pasos de lifting del algoritmo, los cuales son: un paso inicial de actualización ( $U_1$ ), un paso intermedio de predicción (P) y un paso final de actualización ( $U_2$ ). Finalmente hay una etapa de normalización de los coeficientes (Ns y Nd). Se ha simplificado la notación de las ecuaciones para destacar tres parámetros importantes, el parámetro  $\ell$  que representa el nivel de análisis actual, de tal manera que  $\ell - 1$  representa un coeficiente calculado en un nivel de análisis anterior y los parámetros “i” y “j”, que representan el i-ésimo renglón y la j-ésima columna de una imagen  $S_{m \times n}$  respectivamente, donde m y n son potencias de 2.

Update 1	$s_{i,j}^{\ell} = s_{i,2j}^{\ell-1} + \sqrt{3} s_{i,2j+1}^{\ell-1}$	
Predict	$d_{i,j}^{\ell} = s_{i,2j+1}^{\ell-1} - \frac{\sqrt{3}}{4} s_{i,j}^{\ell} - \frac{\sqrt{3}-2}{4} s_{i,j-1}^{\ell}$	
Update 2	$s_{i,j}^{\ell} = s_{i,j}^{\ell} - d_{i,j}^{\ell}$	
Normalize s	$s_{i,j}^{\ell} = \frac{\sqrt{3}-1}{\sqrt{2}} s_{i,j}^{\ell}$	
Normalize d	$d_{i,j}^{\ell} = \frac{\sqrt{3}+1}{\sqrt{2}} d_{i,j}^{\ell}$	
	a)	b)

**Figura 1:** Muestra en a) las ecuaciones para los tres pasos de lifting del algoritmo Db4 así como los factores de normalización, y en b) el esquema de lifting scheme extendido a dos dimensiones, el cual es simétrico, es decir, se aplica una siguiente etapa de lifting scheme tanto a la aproximación como al detalle previamente calculados.

Como se muestra en la figura 1b, aplicar un nivel de análisis de la DWT bidimensional (DWT 2D) es un proceso de dos etapas. En la primera etapa, la DWT 1D se aplica a cada renglón de una imagen  $S_{m \times n}$ , con lo cual se obtienen dos matrices de coeficientes de  $m \times \frac{n}{2}$ , la aproximación y detalle de nivel de análisis  $\ell$   $s^{\ell}$  y  $d^{\ell}$  respectivamente. En la segunda etapa, se aplica la DWT 1D a cada columna de las matrices  $s^{\ell}$  y  $d^{\ell}$ , con lo cual se obtienen cuatro matrices de  $\frac{m}{2} \times \frac{n}{2}$ , a partir de  $s^{\ell}$  obtenemos la aproximación y el detalle horizontal  $s_A^{\ell}$  y  $d_H^{\ell}$ , y a partir de  $d^{\ell}$  obtenemos el detalle vertical y el detalle diagonal  $s_V^{\ell}$  y  $d_D^{\ell}$  respectivamente. La figura 2, muestra este proceso de manera gráfica. Posteriormente, para realizar un análisis de segundo nivel, se repite todo proceso con la matriz  $s_A^{\ell}$ , de manera que se obtienen cuatro matrices de  $\frac{m}{4} \times \frac{n}{4}$  que corresponden al siguiente nivel de análisis  $s_A^{\ell+1}$ ,  $d_H^{\ell+1}$ ,  $s_V^{\ell+1}$  y  $d_D^{\ell+1}$ . Cabe señalar que para todo nivel de análisis, la cantidad de coeficientes totales después del análisis es idéntica a la cantidad de datos en la imagen original.



**Figura 2:** Muestra de manera gráfica el desarrollo de aplicar el algoritmo Db4 basado en lifting a una imagen. A la izquierda se tiene la matriz de datos original, o bien la aproximación calculada en un nivel de análisis anterior. En el centro, se tienen las matrices intermedias de aproximación y detalle al aplicar la Db4 a cada renglón de la matriz anterior. A la derecha, se tienen las cuatro matrices que resultan del análisis en el nivel  $\ell$ . En la gráfica, la aproximación de nivel  $\ell$  ha sido analizada con el algoritmo de la Db4, y descompuesta en sus cuatro matrices resultantes para el siguiente nivel de análisis  $\ell + 1$ .

Otra ventaja algorítmica de la Db4 basada en lifting scheme es que la transformación inversa se obtiene directamente invirtiendo el orden y las operaciones de los pasos de lifting. Desde el punto de vista de las ecuaciones, implica simplemente despejar de las ecuaciones anteriores el término que corresponde al dato original o al coeficiente del nivel de análisis anterior, que es en este caso el que deseamos recuperar a partir de los coeficientes de la DWT. En la figura 3a se pueden apreciar las ecuaciones para la IDWT 1D, en ellas se puede apreciar la inversión en los tres pasos de lifting del algoritmo en el siguiente orden  $U_2^{-1}$ ,  $P^{-1}$  y  $U_1^{-1}$ .

De normalize d 
$$d_{i,j}^\ell = \frac{\sqrt{2}}{\sqrt{3+1}} d_{i,j}^\ell$$

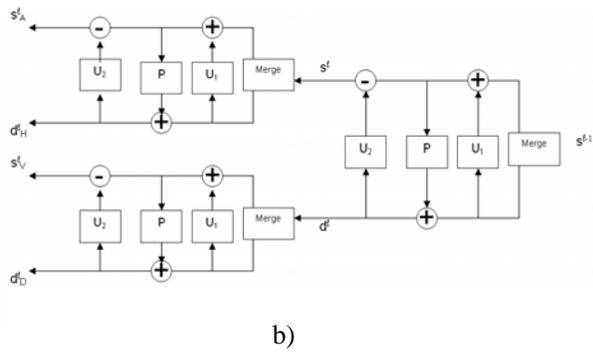
De normalize s 
$$s_{i,j}^\ell = \frac{\sqrt{2}}{\sqrt{3-1}} s_{i,j}^\ell$$

De update 2 
$$s_{i,j}^\ell = s_{i,j}^\ell + d_{i,j+1}^\ell$$

De predict 
$$s_{i,2j+1}^{\ell-1} = d_{i,j}^\ell + \frac{\sqrt{3}}{4} s_{i,j}^\ell + \frac{\sqrt{3}-2}{4} s_{i,j-1}^\ell$$

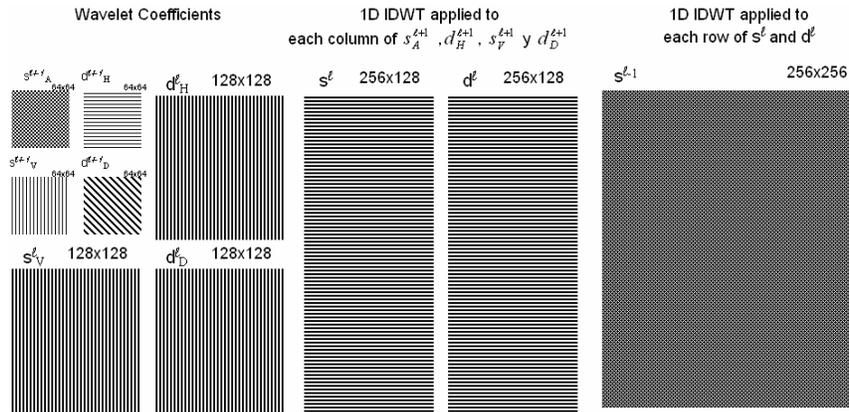
De update 1 
$$s_{i,2j}^{\ell-1} = s_{i,j}^{\ell-1} + \sqrt{3} s_{i,2j+1}^{\ell-1}$$

a)



**Figure 3:** Muestra en a) las ecuaciones para los tres pasos de lifting de la IDWT así como los factores de de-normalización, y en b) el esquema de lifting scheme extendido a dos dimensiones, el cual es simétrico, es decir, se aplica una siguiente etapa de lifting scheme tanto a la aproximación como al detalle previamente calculados.

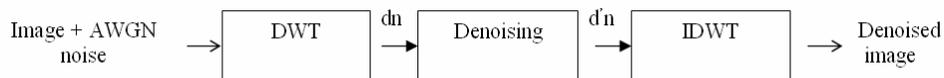
Al igual que en el caso de la DWT, la transformación inversa consiste en dos pasos. El primer paso inicia con la de-normalización de las matrices de coeficientes  $s_A^{\ell+1}$ ,  $d_H^{\ell+1}$ ,  $s_V^{\ell+1}$  y  $d_D^{\ell+1}$  indicada por  $Ns^{-1}$  y  $Nd^{-1}$ , tras lo cual se aplica la IDWT 1D a cada columna de las cuatro matrices. A partir de  $s_A^{\ell+1}$ ,  $d_H^{\ell+1}$  se obtiene  $s^\ell$  y a partir de  $s_V^{\ell+1}$  y  $d_D^{\ell+1}$  se obtiene  $d^\ell$ . En la segunda etapa, se aplica la IDWT a cada renglón de  $s^\ell$  y  $d^\ell$ , a partir de lo cual se obtiene la reconstruye la secuencia de datos de la imagen original, o bien los coeficientes de la etapa de análisis anterior.



**Figura 4:** Muestra de manera gráfica el desarrollo de aplicar la IDWT con el algoritmo Db4 basado en lifting a una imagen. A la izquierda se tienen las matrices de coeficientes del nivel de análisis actual. En el centro, se tienen las matrices intermedias de aproximación y detalle al aplicar la IDWT a cada columna de las matrices anteriores. A la derecha, se tienen la matriz resultante que corresponde a la imagen original, o bien a los coeficientes de análisis en el nivel anterior,  $\ell - 1$ . En la gráfica, la aproximación de nivel  $\ell$  está descompuesta en sus cuatro matrices resultantes para el siguiente nivel de análisis  $\ell + 1$

### 3. Cancelación de ruido con wavelets

Una vez obtenidos los coeficientes de la DWT para un nivel de análisis  $\ell$  determinado, es posible procesarlos, para obtener un efecto deseado en la imagen. Como se muestra en la figura 5, se ha insertado una etapa intermedia para cancelación de ruido (Denoising). Dos de los métodos más comunes para cancelación de ruido son el Hard thresholding y el Soft thresholding.



**Figura 5:** Muestra la técnica de wavelet denoising, en la cual los coeficientes de la DWT son procesados por la técnica de thresholding antes de ser reconstruïdos.

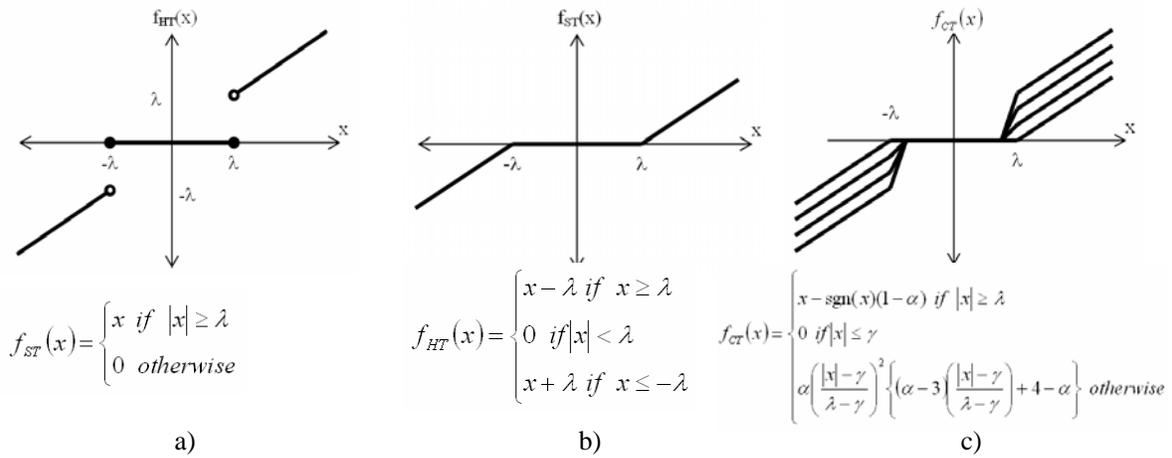
En la técnica Hard thresholding, todos los coeficientes que son mayores que un valor de umbral  $|\lambda|$  determinado se conservan y son multiplicados por la función de threshold, mientras que todos los demás son convertidos a ceros, como se muestra en la figura 6a. El valor del umbral  $\lambda$  se calcula en base a la potencia de la imagen y la varianza del ruido, por lo cual, el criterio de selección consiste en que todos los coeficientes mayores que  $\lambda$  poseen un valor significativo y se consideran parte de la imagen.

En la técnica Soft thresholding, todos los coeficientes que son mayores o iguales que un valor de umbral  $\lambda$  determinado se conservan y son multiplicados por  $x - \lambda$ , todos los coeficientes que son menores o iguales que un valor de umbral  $-\lambda$  determinado se conservan y son multiplicados por  $x + \lambda$ , mientras que todos los demás son convertidos a ceros, como se muestra en la figura 6b. La función Soft thresholding es continua, a diferencia de la función Hard thresholding, razón por la cual se utiliza con más frecuencia en cancelación de ruido.

Para el desarrollo de este artículo, se utilizó una técnica propuesta por Yoon and Vaidyanathan llamada Custom thresholding, la cual combina lo mejor de las ventajas de Hard y Soft thresholding (Yoon and Vaidyanathan, 2004). Por un lado, la función es continua alrededor del umbral a la vez que se adapta a las características de la señal de entrada. La función Custom thresholding aparece en la figura 6c. El parámetro  $\alpha$  puede tomar los valores  $0 \leq \alpha \leq 1$ . Por otro lado, el parámetro  $\gamma$  depende de  $\lambda$  de la siguiente forma  $0 < \gamma < \lambda$ , y finalmente,  $\lambda$  se calcula a partir de la longitud de la imagen y la varianza del ruido de la siguiente manera:

$$\lambda = \sigma \sqrt{2 \text{Log}(MN)}$$

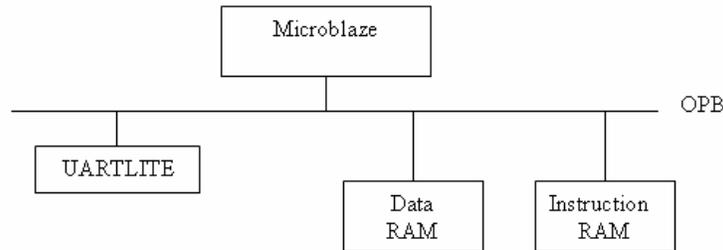
Donde  $\sigma$  es la varianza del ruido, y  $M \times N$  es el numero de elementos en la imagen  $S_{M \times N}$ . El parámetro  $\alpha$  determina la forma de la función, de tal manera que para  $\alpha = 0$  tenemos el caso de Soft thresholding, mientras que  $\alpha = 1$  tenemos el caso de Hard thresholding.



**Figura 6:** Muestra las funciones de Hard thresholding (a), Soft thresholding (b) y Custom thresholding (c).

#### 4. El sistema Diseñado

Se implemento en un FPGA Spartan-3 de Xilinx el siguiente sistema, el cual consiste en el microprocesador suave Microblaze, con IP adicionales como periféricos. El algoritmo de la Db4 basada en lifting scheme fue escrito en lenguaje C, programado y ejecutado en el microprocesador. La imagen es manipulada en Matlab, y enviada por el puerto serie a la tarjeta Spartan-3.



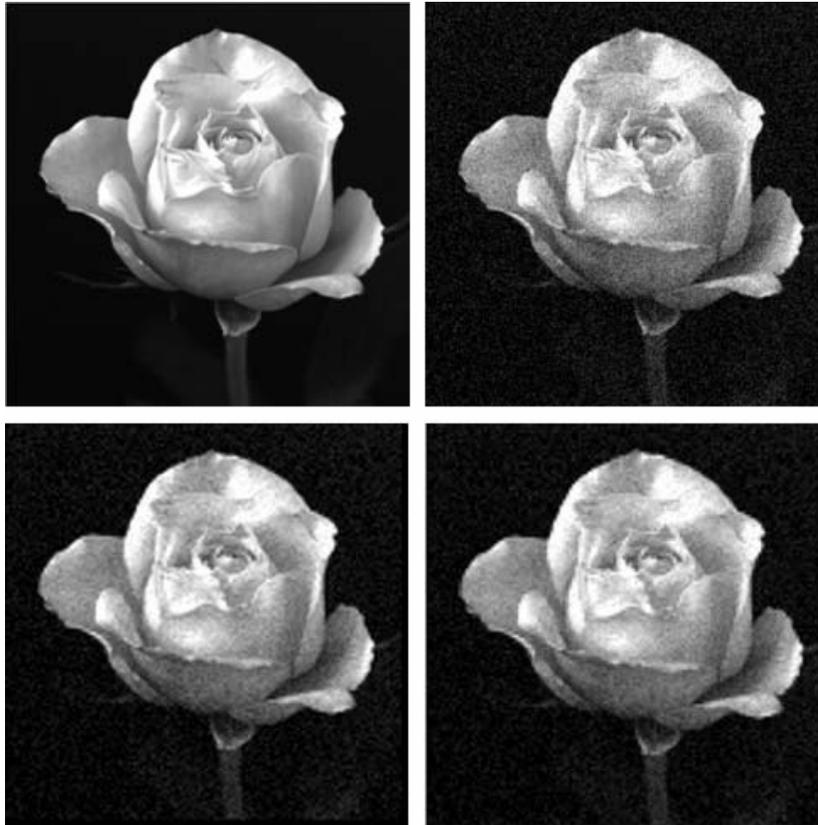
**Figura 7:** Muestra el diagrama a bloques del sistema diseñado.

Una vez recibidos los pixeles de la imagen, se procesan utilizando la DWT, Custom thresholding y la IDWT, tras lo cual son reenviados a la PC, vía el puerto serial.

## 5. Resultados y Conclusiones

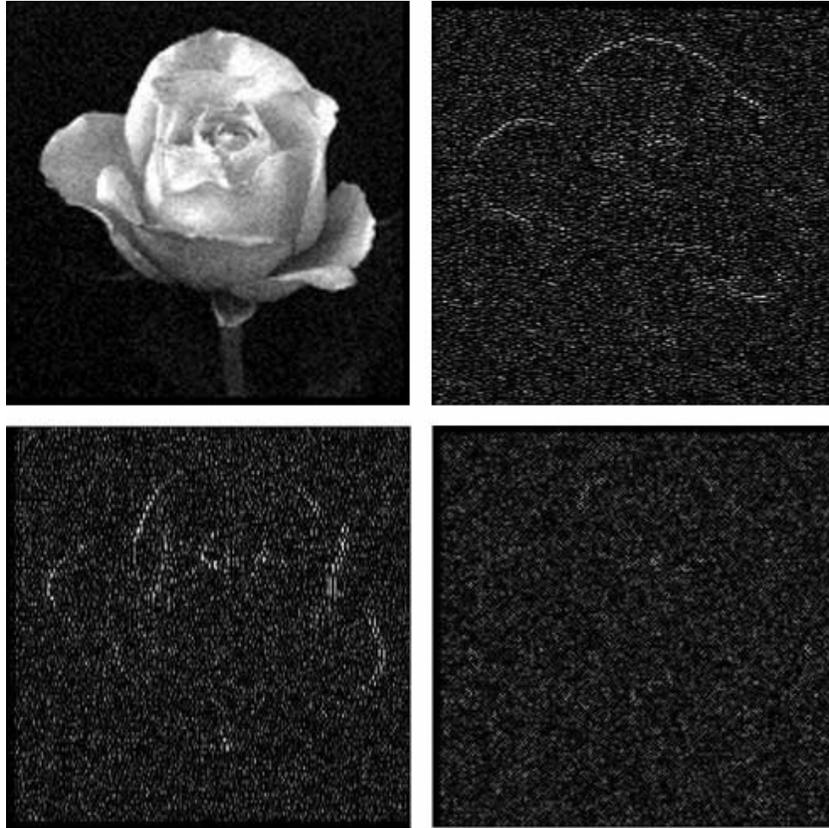
Se calculó la potencia de la imagen en watts a partir de la ecuación [1], la cual resultó ser de 0.0924 W = -10.3427 dB. Con estos datos, se generó ruido blanco aditivo gaussiano utilizando la función **wgn** (M,N,PWRnoiseDb) de Matlab, la cual permite generar ruido AWGN con un valor de potencia especificado en decibeles por PWRnoiseDb. Para este caso, se utilizó ruido con una potencia igual a 1/8 de la potencia calculada en la imagen. Entonces, la matriz de ruido gaussiano de MxN generada se añade elemento por elemento a la matriz de la imagen. En la figura (7) a y b, se muestran la imagen original y la imagen contaminada con el ruido AWGN generado, respectivamente.

$$\sigma_{i,j}^2 = \frac{MN \sum_{i=1}^M \sum_{j=1}^N x_i^2 - \left( \sum_{i=1}^M \sum_{i=1}^N x_i \right)^2}{(MN)^2} \quad [1]$$



**Figura 7.** De izquierda a derecha, de arriba abajo: la imagen original, la imagen contaminada por ruido AWGN, la aproximación de la imagen, la imagen reconstruida después de aplicar denoising.

La imagen contaminada con ruido se envió por el puerto serie al Microprocesador Microblaze, en el cual se programó el algoritmo DWT 2D para la DB4 basada en lifting scheme. El programa lleva a cabo la transformación directa, y después aplica Custom Tresholding a las matrices de coeficientes, para finalmente realizar la transformación inversa y enviar la imagen reconstruida a la PC, a través del puerto serie. La reconstrucción de la imagen se muestra en la figura 7 c. Las imágenes obtenidas en Matlab a partir de los coeficientes del primer nivel de análisis de las cuatro matrices de coeficientes resultantes, se muestran a continuación.



**Figura 8.** De izquierda a derecha de arriba abajo: Aproximación, Detalle Horizontal, Detalle Vertical, Detalle Diagonal.

La ecuación [2] se utilizó para aplicar Custom tresholding a las matrices de coeficientes Horizontal, Vertical y Diagonal.

$$f_c(x) = \begin{cases} x - \text{sgn}(x)(1 - \alpha)\lambda & \text{if } |x| \geq \lambda \\ 0 & \text{if } |x| \leq \gamma \\ \alpha \left(\frac{|x| - \gamma}{\lambda - \gamma}\right)^2 \{(\alpha - 3)\left(\frac{|x| - \gamma}{\lambda - \gamma}\right) + 4 - \alpha\} & \text{otherwise} \end{cases} \quad [2]$$

Finalmente, se obtuvo la potencia del ruido en la imagen reconstruida restando la potencia de la imagen original menos la imagen reconstruida. Los resultados se muestran en la Tabla 1.

**Tabla 1.** Muestra los resultados de aplicar cancelación de ruido utilizando wavelets. En la imagen contaminada con ruido AWGN se tiene una SNR dB de 9.03, mientras que la imagen reconstruida es de 10.849, lo cual implica un factor de 1.5201.

Imagen/Potencia	Potencia (watts)	Potencia (dB)	SNR (watts)	SNR (dB)
Original	0.0924	-10.3427	8	9.0309
Ruido	0.0116	-19.3736		
Reconstruida	0.0997	-10.0122	12.1608	10.8496
Ruido	0.0082	-20.8619		

## **6. Referencias**

Yoon Byung-Jun and Vaidyanathan P. P., (2004). "Wavelet-based denoising by customized thresholding". *Proc. 29th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal.

## **5. Authorization and Disclaimer**

Authors authorize LACCEI to publish the papers in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.