

Desarrollo de una Herramienta Educativa, para la Comprensión y Reducción de Ruido en Imágenes en Escala de Grises, Usando Transformada de Wavelet

C. Córdova¹, P. Chávez²

Facultad Ingeniería en Electricidad y Computación (FIEC)

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL (ESPOL)

Campus Gustavo Galindo, Km 30.5 vía Perimetral. Apartado 09-01-5863. Guayaquil, Ecuador

cdcordova@gmail.com¹, pchavez@fiec.espol.edu.ec²

Resumen

El constante crecimiento de las facilidades tecnológicas, nos llevan a diario a manejar una gran cantidad de información, la cual debe ser enviada almacenada, el tamaño de la información es muy importante, ya que a mayor tamaño, mayor es el costo de transmisión o de almacenamiento. Es por ello que es muy importante buscar algún tipo de técnica o proceso que nos permita comprimir la información. Este trabajo trata sobre compresión de imágenes para lo que usamos las transformadas de Wavelet, que son un desarrollo relativamente reciente tanto en matemáticas puras como aplicadas. Con el objeto de comprender el uso de esta transformada, he desarrollado una herramienta educativa, usando el software MATLAB, para comparar y comprender el uso de los distintos tipos de transformada de Wavelet en la compresión de imágenes en escala de grises, y a su vez ver su utilidad en la reducción de ruido en imágenes. El programa toma una imagen la convierte en escala de grises y presenta distintas opciones para realizar la compresión de la imagen. Luego se puede agregar varios tipos de ruido a la imagen original, para luego poder eliminar el ruido usando transformada de wavelet.

Palabras Claves: MATLAB, Compresión de Imágenes, Reducción de Ruido, Transformada de Wavelet

Abstract

The regular growth of the technological facilities lead us daily to handle a large amount of information which must be stored, the size of the information is very important because when is larger, the cost of transmission or storage is higher. That is why it is very important to find some kind of technique or process that allows us to compress the information. This work is about image compression, we use Wavelet transforms, which are a relatively recent development in both pure and applied mathematics. In order to understand the use of this transform, I have developed an educational tool, using the MATLAB software to compare and understand the use of different types of wavelet transform in compression of grayscale images, and then to see its usefulness to reduce noise in images. The program takes an image, makes it grayscale and presents different options for image compression. Then you can add various types of noise to the original image, and then to remove noise using wavelet transform.

Keywords: MATLAB, Image Compression, Denoised Image, Wavelet Transform

1. Introducción

La transformada wavelet consiste en comparar la señal con ciertas funciones wavelet, las cuales se obtienen a partir de las wavelet madre. La comparación permite obtener unos coeficientes que son susceptibles de interpretación y posterior manipulación.

1.1 Transformada de Wavelet

La transformada de wavelet es un caso especial de la transformada de Fourier, y así como para la transformada de Fourier se usan unos algoritmos discretos como el FFT, así mismo en la transformada de wavelet usamos el DWT, es decir, la transformada de Wavelet discreta.

El cálculo de la transformada wavelet para todas las posibles escalas supone una gran cantidad de información. Escoger solo aquellas escalas y posiciones que resulten interesantes para ciertos estudios es una tarea difícil.

Para muchas señales la información más importante se encuentra en las frecuencias bajas, mientras que en las altas frecuencias se encuentran los detalles o matices de la señal. Por ejemplo, en el caso de la voz humana, si eliminamos los componentes con altas frecuencias, la voz suena diferente pero se sigue entendiendo su mensaje. En cambio, si lo que se elimina son las componentes de bajas frecuencias, el mensaje se vuelve irreconocible. Por eso el análisis wavelet permite descomponer la señal en aproximaciones y detalles, a éste proceso se le conoce con el nombre de análisis [1].

1.2 Descomposición de la Imagen.

Si a una imagen le aplicamos la DWT obtenemos cuatro tipos de coeficientes: aproximaciones, detalles horizontales, detalles verticales y detalles diagonales. La aproximación contiene la mayor parte de la energía de la imagen, es decir, la información más importante, mientras que los detalles tienen valores próximos a cero.

Al Descomponer la imagen, estamos logrando tener una imagen que contendrá la mayor parte de la información, si a esta imagen le aplicamos el algoritmo de Huffman para comprimirla, dado su histograma no comprimirá mucho más que la original, sin embargo, por el proceso de diezmado, esta imagen ya es más pequeña, además sabemos que tendremos otras tres imágenes que contienen los detalles, o el matiz de la señal, y que estos detalles tienen valores próximos a cero, esto nos permite tener un histograma concentrado y nos permitirá comprimir más al aplicar Huffman. Todo esto se explicará más adelante, por ahora se nombra para que el lector entienda superficialmente que objetivo se busca al aplicar la transformada discreta de wavelet.

1.3 Código Huffman

En 1952, David Huffman propuso un método estadístico que permitía asignar un código binario a los diversos símbolos a comprimir (píxeles o caracteres, por ejemplo). La longitud de cada código no es idéntica para todos los símbolos: se asignan códigos cortos a los símbolos utilizados con más frecuencia (los que aparecen más a menudo), mientras que los símbolos menos frecuentes reciben códigos binarios más largos.

El codificador Huffman crea una estructura arbórea ordenada con todos los símbolos y la frecuencia con que aparecen. Las ramas se construyen en forma recursiva comenzando con los símbolos menos frecuentes.

La construcción del árbol se realiza ordenando en primer lugar los símbolos según la frecuencia de aparición. Los dos símbolos con menor frecuencia de aparición se eliminan sucesivamente de la lista y se conectan a un nodo cuyo peso es igual a la suma de la frecuencia de los dos símbolos. El símbolo con menor peso es asignado a la rama 1, el otro a la rama 0 y así sucesivamente, considerando cada nodo formado como un símbolo nuevo, hasta que se obtiene un nodo principal llamado raíz.

El código de cada símbolo corresponde a la sucesión de códigos en el camino, comenzando desde este carácter hasta la raíz. De esta manera, cuanto más dentro del árbol esté el símbolo, más largo será el código.

1.4 Reducción de Ruido

El procedimiento de reducción de ruido involucra 3 pasos.

- Descomponer. Escoger el tipo de transformada a utilizar, escogiendo de igual manera el nivel de descomposición.

- Aplicar el umbral (Threshold) a los coeficientes de descomposición. Para cada uno de los niveles de descomposición y aplicar el soft thresholding a los coeficientes de descomposición.

- Reconstrucción. Realizar la reconstrucción wavelet usando los coeficientes de Aproximación originales de nivel N y los coeficientes modificados de nivel 1 hasta N [2].

2. Procedimiento en Matlab

2.1 Compresión de la Imagen

La imagen es primero transformada en una imagen en escala de grises por la función "Conv_Gray_Scale".

Luego procedemos a aplicar la codificación Wavelet. Para lo cual hemos seleccionado la función `wavedec2(X, wname, level)` que viene dentro del paquete Wave toolbox de Matlab. El resultado de esta operación son los coeficientes de descomposición que estarán almacenados en las matrices C y S. La función

mencionada tiene los siguientes parámetros a ser escogidos por el usuario:

- Imagen a descomponerse
- Wname. Nombre de la transformada a utilizarse para la descomposición, en nuestro programa el usuario tiene las siguientes opciones: (Haar, Coiflets, Symlets, Dmeyer, Daubechies).
- Level. Nivel de descomposición wavelet.

Luego se usa la función `wdcbm2` del paquete Wave toolbox, para el cálculo las matrices THR y NKEEP. THR es una matriz de tamaño N (nivel de descomposición) que contiene el respectivo threshold para cada nivel de descomposición. NKEEP contiene la cantidad de coeficientes que deben ser conservados en cada uno de los niveles.

Finalmente para realizar la compresión de la imagen, usamos la función `wdecnmp` de parámetros: C y S que son las matrices que se obtuvieron al codificar la imagen a comprimir, wname que es el nombre de la wavelet usada, level que es el nivel de descomposición, thr que es la matriz que contiene el threshold para cada nivel de descomposición.

Esta función nos devuelve la imagen comprimida [3].

2.2 Reducción de Ruido.

Primero se le agrega ruido a la imagen con la función "Add_Noise". En la cual el usuario podrá escoger entre los siguientes tipos de ruido:

- Gaussiano con media = 0 y varianza 0,01.
- Speckle con media 0 y varianza 0,04.
- Salt & Pepper con densidad de ruido 0,05.
- Poisson.

Luego se usa la función `ddenncmp` del paquete Wave toolbox, para el cálculo de los valores de THR, SORH, KEEPAPP.

THR es una matriz de tamaño N (nivel de descomposición) que contiene el respectivo threshold para cada nivel de descomposición.

SORH indica si es soft o hard threshold.

KEEPAPP permite conservar los coeficientes de aproximación.

Finalmente para realizar la compresión de la imagen, usamos la función `wdecnmp` de parámetros:

- X imagen ruidosa.
- Wname que es el nombre de la wavelet usada.
- Level que es el nivel de descomposición.
- Thr que es la matriz que contiene el threshold para cada nivel de descomposición [4].

3. Análisis Cualitativo y Cuantitativo de los Datos.

Se realizó un análisis cuantitativo y cualitativo para analizar ambos procedimientos, el de compresión por medio de encuestas, y resultados de tamaño en bytes. Para el procedimiento de reducción de ruido se usó el Mínimo error cuadrático medio normalizado y una

serie de encuesta comparando imágenes obtenidas con las originales

3.1 Compresión

Para analizar el efecto de la compresión usando transformada de Wavelet, usamos el siguiente procedimiento:

- Se selecciona una imagen de colores en formato mapa de bits.
- Se convierte dicha imagen en escala de grises.

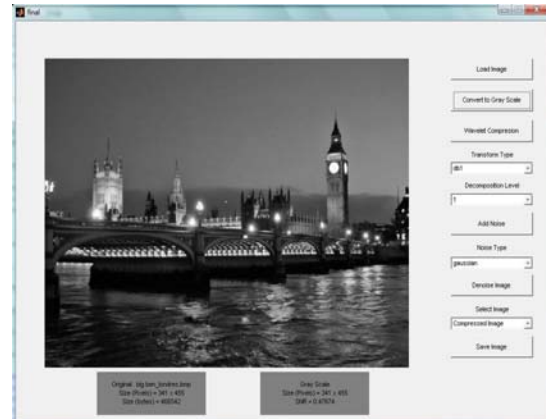


Figura 1. Imagen convertida en escala de grises.

- Se selecciona uno de los tipos de Transformada de Wavelet para poder comprimir la imagen, las transformadas disponibles en el programa son Haar, Daubechies, Symlets.

- También se debe de seleccionar el nivel de compresión. Los Niveles de compresión a elegir van desde el 1 hasta el 6. A mayor nivel, mayor será el radio de compresión, pero se perderán más detalles.

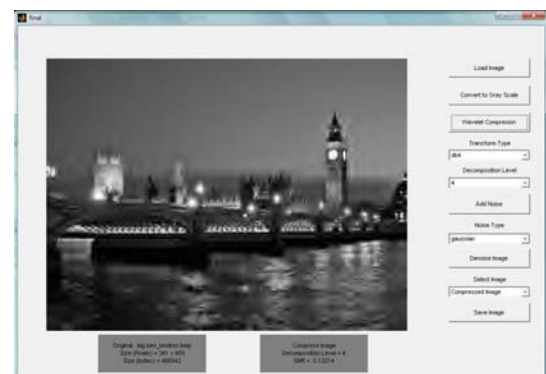


Figura 2. Imagen comprimida con db4 y nivel 4.

3.2 Reducción de Ruido

Para analizar el efecto de reducción de ruido usando transformada de Wavelet, usamos el siguiente procedimiento:

- Se procede a agregar ruido a la imagen se debe seleccionar el tipo de ruido que se desea agregar.



Figura 3. Imagen con ruido gaussiano.

- Se selecciona y se realiza la reducción de ruido.



Figura 4. Imagen con reducción de ruido.

3.3 Mínimo Error Cuadrático Medio Normalizado

Se comparan las imágenes resultantes de los métodos de escalamiento con la imagen original utilizando el método de MSE (Minimum Square Error) Normalizado.

Esta es una técnica de análisis numérico que es usada para estimaciones y cuando hay residuos (errores estimados).

Nosotros lo emplearemos para cuantificar las diferencias entre la imagen original y las imágenes comprimidas y recuperadas, por lo que haremos énfasis en el sentido residual del MSE ya que este asigna un número que indica cuán lejos o errados están los datos observados de los datos originales.

Un MSE de cero indica que no hubo error y cualquier número mayor indica que hay diferencia entre lo que debería haber con lo que hay, mientras mayor sea el número, mayor será en error.

3.4 Análisis Cualitativo

En esta parte del análisis nos concentramos en realizar una serie de encuestas en las que comparamos la composición.

3.5 Resultados Finales.

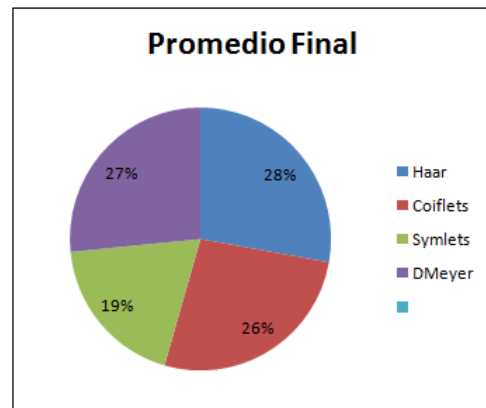


Figura 5. Aceptación en la reducción de Ruido.

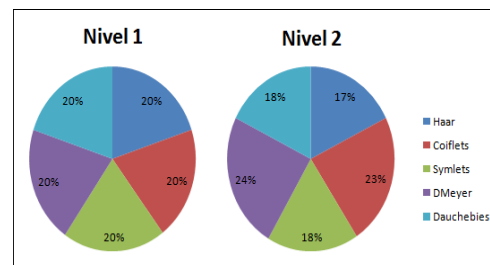


Figura 6. Compresión de imágenes Nivel 1 y 2

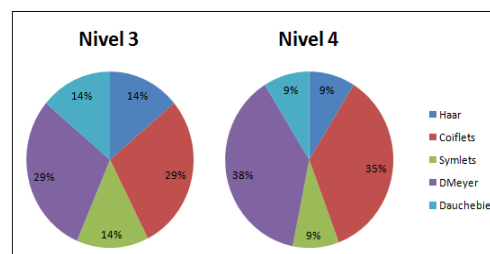


Figura 7. Compresión de imágenes Nivel 3 y 4.

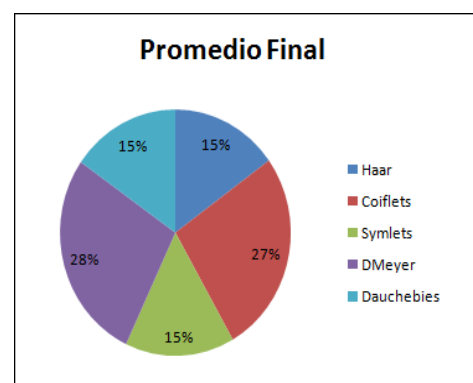


Figura 8. Promedio Final compresión de imágenes.

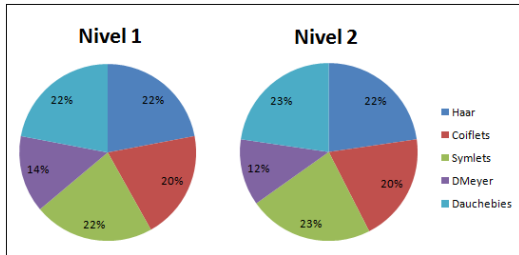


Figura 9. NSME compresión de Nivel 1 y 2

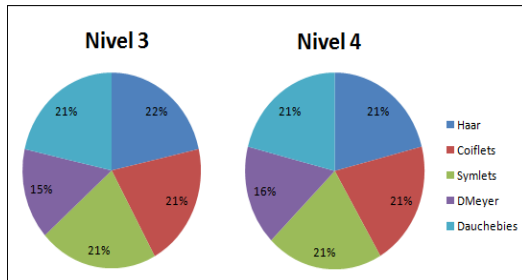


Figura 10. NSME compresión de Nivel 3 y 4

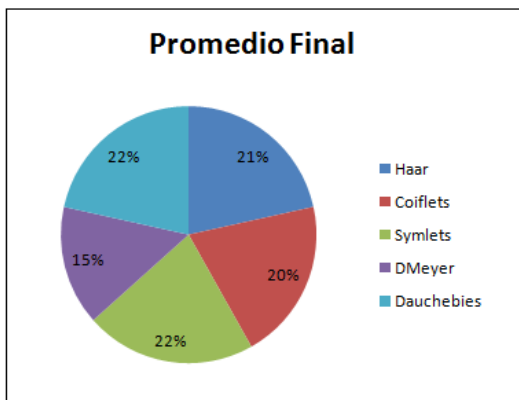


Figura 11. NSME Promedio compresión

4. Conclusiones

Después de procesar y analizar los datos obtenidos en las encuestas puedo afirmar que la transformada de Haar es en promedio la que mayor aceptación tuvo con un 22%, mostrando niveles muy altos en la eliminación de ruido Speckle y Poisson, pero una mal desempeño en Gaussian y Salt & Pepper, sin embargo la más versátil de las transformadas fue la de Daubechies mostrando un 24% en tres de los cuatro ruidos, aunque no respondió bien al ruido Gaussiano, solo un 11%.

Si analizamos las figuras 3.15 y 3.16 sobre los resultados cuantitativos de la compresión, puedo concluir que en lo que al nivel de compresión en bytes se refiere, las transformadas de Haar y Symlets mostraron los mejores resultados llegando a un nivel

de compresión del 7% de la imagen original. Por el contrario podemos decir que las transformadas de Coiflets y DMeyer, mostraron no ser para nada útiles en este propósito. Si se analiza el Error Cuadrático Medio Normalizado promedio de las imágenes, se observa que la transformada de mejor desempeño fue la de DMeyer, las otras cuatro transformadas mostraron un resultado muy similar. Todos los promedios se muestran en las figuras 3.18, 3.19 y 3.20.

Después de analizar las distintas tablas de resultados, puedo concluir que a pesar de que los resultados varían de acuerdo al procedimiento que se desee realizar, la transformada que presentó mejores resultados en promedio es la transformada de Daubechies.

5. Recomendaciones

Se recomienda que debido a estos resultados variados, reconocer lo importante que es reconocer y analizar la imagen a procesar, realizar las respectivas pruebas con los varios métodos hasta encontrar el método que cumpla de mejor manera nuestras expectativas.

Se recomienda la continuación de la investigación realizada en el presente trabajo, ya que la herramienta diseñada brinda una alternativa al momento de evaluar los resultados, compararlos entre ellos y puede ser útil al momento de comparar los distintos tipos de transformada.

6. Agradecimientos

Agradezco a mi madre, por su apoyo constante y su confianza invariable.

A mi familia que con mucho orgullo me ha apoyado en mis logros y ayudado con mis problemas.

7. Referencia

- [1] Majid Rabbani; Wavelet Compression and the JPEG2000 Standard; Eastman Kodak Research Laboratories; 2003.
- [2] Smith, S.W.; The Scientist and Engineer's Guide to Digital Signal Processing; Second Edition; California Technical Publishing; 1999.
- [3] Ingle, V.K. ; Proakis, J. G ; Digital System Processing Using MATLAB V.4; PWS Publishing Company, 1997.
- [4] Nikola Sprljan; Image & Video Compression Depot; www.mathworks.com; 2005.