



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACIÓN

“Análisis de Compresión y Descompresión de imágenes en escala de grises usando la Transformada Wavelet y sus ventajas con respecto a la Transformada de Fourier”

INFORME DE MATERIA DE GRADUACIÓN

Previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Pablo Augusto Quijije Mejía

Nadia Jackeline Sanjinés Flores

GUAYAQUIL – ECUADOR

AÑO: 2009

A G R A D E C I M I E N T O

A Dios por permitirnos culminar nuestros estudios y subir un escalón más.

A la ESPOL por abrirnos sus puertas y permitirnos recibir la formación académica de excelencia durante estos años.

A nuestros profesores por el conocimiento y experiencia compartido durante los años de estudio, especialmente al Ing. Patricia Chavez, Profesora de la Materia de Graduación, por su paciencia e invaluable ayuda.

DEDICATORIA

A mi madre, ejemplo de trabajo incansable, por su incondicional apoyo pese a las adversidades de la vida. A mi padre por sus consejos, a mi hermano por su paciencia.

Pablo Quijije M.

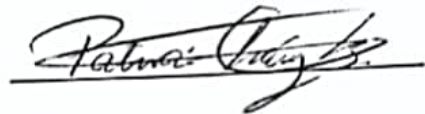
Nadia Sanjines F.

TRIBUNAL DE GRADUACIÓN

Ing. Jorge Aragundi

SUBDECANO DE LA FIEC

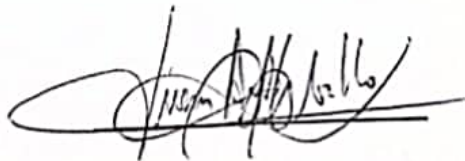
PRESIDENTE



Ing. Patricia Chávez Burbano

PROFESOR DE MATERIA DE

GRADUACION



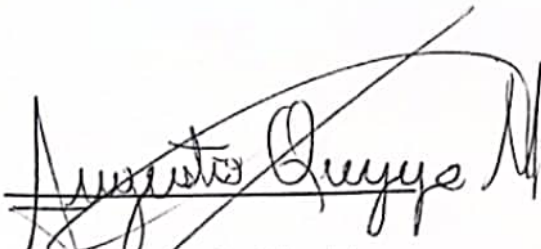
Ing. Juan Carlos Avilés

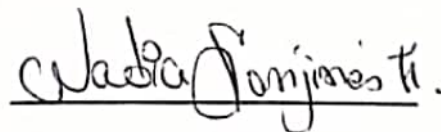
DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este proyecto de graduación nos corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)


Pablo Quijije M.


Nadia Sanjines F.

RESUMEN

El presente trabajo analiza la Comprensión y Descompresión de imágenes en escala de grises usando la Transformada Wavelet y muestra las ventajas con respecto a la Transformada de Fourier. De esta manera podremos incluir la importancia que tiene el estudio de la Transformada Wavelet dentro del curriculum universitario. El análisis se realizará a través de un programa desarrollado en la herramienta matemática, Matlab, usando algoritmos de descomposición y reconstrucción.

En el capítulo 1 se da a conocer ciertos conceptos que permitirán entender los distintos términos que se manejan en el proyecto.

El capítulo 2 describe brevemente aplicación de wavelet en la comprensión de imágenes.

En el último capítulo se menciona como se desarrolló el proyecto.

.

ÍNDICE GENERAL

INTRODUCCIÓN.....	12
Capítulo 1.....	14
1. Conceptos Generales	14
1.1. Filtro Digital.....	14
1.1.1. Filtros Pasa Alto(HPF)	15
1.1.2. Filtros Pasa Bajo(LPF).....	15
1.2. Transformada de Fourier.....	16
1.2.1. Limitaciones de la transformada de Fourier	16
1.2.2. Ventana de Fourier.....	17
1.3. Transformada Wavelet	17
1.3.1. Transformada Continua Wavelet.....	18
1.3.1.1. Desplazamiento del wavelet.....	19
1.3.1.2. Relación entre escala del wavelet y frecuencia de la señal	21
1.3.2. Descomposición de la frecuencia	21
1.3.2.1. Downsampling	22
1.3.2.2. Descomposición en múltiples niveles	22
1.3.3. Reconstrucción de la señal.....	23
1.3.3.1. Upsampling	23
1.3.3.2. Filtros de Reconstrucción.....	23
1.4. Cuantificadores	24
1.4.1. NMSE	24
1.4.2. Histograma de una Imagen.....	25
Capítulo 2.....	26
2. Aplicación de Wavelet a la compresión de Imágenes.....	26
2.1. Codificación de imágenes	26
2.1.1. Descomposición de la Imagen	26
2.1.2. Cuantización.....	27
2.1.2.1. Umbral.....	28
2.2. Decodificación de Imágenes.....	28
2.2.1. Descuantizador	28
2.2.2. Reconstrucción	28

Capítulo 3.....	30
3. Desarrollo del Proyecto	30
3.1. Elección de Imágenes.....	30
3.2. Definición de Parámetros	31
3.3. Ejecución del algoritmo.....	31
3.4. Funcionalidad del Programa	33
3.5. Resultados.....	34
3.6. Análisis Cuantitativo	35
3.7. Análisis Cualitativo	39
Conclusiones.....	43
Recomendaciones.....	44
ANEXO A	46
Interactuando con la interfaz	46
ANEXO B.....	50
Código Fuente del Programa.....	50
ANEXO C.....	61
Encuesta de Estudio de Mercado	61
Bibliografía.....	68

ÍNDICE DE FIGURAS

Figura 1.1: Representación básica de un filtro.....	14
Figura 1.2: Filtro Pasa Alto.....	15
Figura 1.3: Filtro Pasa Bajo.....	15
Figura 1.4: Representación de la Transformación de Fourier.....	16
Figura 1.5: Transformación a través de la ventana de fourier.....	17
Figura 1.6: Transformación al dominio Wavelet.....	18
Figura 1.7: Gráficas de funciones de escala.....	19
Figura 1.8: a. Función Wavelet, b. Función Wavelet desplazada.....	19
Figura 1.9: (a) Paso 2, (b) Paso 3.....	20
Figura 1.10: Proceso Básico de descomposición wavelet.....	21
Figura 1.11: Representación en bloques de submuestreo.....	22
Figura 1.12: Árbol de descomposición de wavelet.....	22
Figura 1.13: Inserción de ceros a la componente de la señal.....	23
Figura 1.14: Bloques de reconstrucción de la señal.....	24
Figura 1.15: Imagen con su respectivo Histograma.....	25
Figura 2.1: Imagen x descompuesta en sus coeficientes de aproximación y detalles.....	27
Figura 3.1:lena.jpg 256x256.....	30
Figura 3.2:estanque.jpg 1024x768.....	30
Figura 3.3:barbara.jpg 512x512.....	30
Figura 3.4:escalera.jpg 640x480.....	30
Figura 3.5: Función escalar de la familia Wavelet Daubechies 6.....	31
Figura 3.6: Estanque en su descomposición wavelet de nivel 1.....	32
Figura 3.7: Estanque en wavelet piramidal de Nivel 2 cuantizada.....	33
Figura 3.8: NMSE de Lena.....	36

Figura 3.9: NMSE de Estanque.....	36
Figura 3.10: NMSE de Bárbara.....	36
Figura 3.11: NMSE de Escalera.....	36
Figura 3.12: Energía de las imágenes de Lena.....	37
Figura 3.13: Energía de las imágenes de escalera.....	37
Figura 3.14: Tasa de compresión alcanzada en estanque.jpg.....	38
Figura 3.15: Tasa de compresión alcanzada en barbara.jpg.....	38
Figura 3.8: Barbara Original y Reconstruida desde el nivel 3 con wavelet.....	39
Figura 3.9: Barbara Original y Reconstruida desde el nivel 3 con fourier.....	40
Figura A-1.- Interfaz gráfica inicial del programa.....	46
Figura A-2.- Ingreso y carga de la Imagen.....	47
Figura A-3.- Selección de la transformada e ingreso del nivel de descomposición.....	48
Figura A-4.- Resultado final del programa al someter a la imagen de escalera a la compresión/descompresión de imágenes usando wavelet de nivel uno.....	49
Figura A-5.- Resultado final del programa al someter a la imagen de lena a la compresión/descompresión de imágenes usando fourier de nivel uno.....	49
Figura C-1. Resultado de el método de compresión/descompresión escogido.....	67
Figura C-2. Resultados de cada uno de los factores que influyen en el análisis de compresión/descompresión de imágenes.....	67

ÍNDICE DE TABLAS

Tabla 3.1: Resultados obtenidos de las imágenes con el análisis wavelet.....	34
Tabla 3.2: Resultados obtenidos de las imágenes con el análisis fourier.....	35

INTRODUCCIÓN

En el mundo de las ciencias especialmente las matemáticas, la física y la computación, se ha tenido la necesidad de analizar diversas funciones. Esta necesidad surge, principalmente, debido que en la naturaleza muchos de los procesos pueden ser modelados como funciones, ya sean discretas y continuas.

El avance de la tecnología permite que hoy en día se disponga de nuevos elementos y recursos para el análisis y procesamiento de señales e imágenes. El área del procesamiento de señales e imágenes se ha venido desarrollando a través del estudio teórico de las diferentes transformadas, La gran mayoría de estos estudios se ha empleado a través del uso de la transformada de Fourier, que se manifiesta en la creación de algoritmos que proyectan soluciones a problemas de la vida real, soportados por avanzados elementos de software y de hardware.

El procesamiento digital de imágenes ha venido desempeñando un papel de gran importancia en el área de ingeniería e incluso en la medicina. Gracias a este procesamiento hoy en día es posible manipular imágenes para obtener información objetiva que nos ayude a tomar decisiones para la resolución de problemas.

Nuestro proyecto presenta un estudio de la transformada de Wavelet como una alternativa a la implementación de algoritmos de compresión/descompresión de imágenes en escala de grises. A través del presente se realiza un análisis cuantitativo y cualitativo para establecer las ventajas y desventajas con respecto al uso de de la transformada de Fourier.

Capítulo 1

1. Conceptos Generales

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

A través de este capítulo, se describen los conceptos fundamentales que implican en el desarrollo de la comprensión de imágenes. Resumiendo así, el concepto de las transformadas de fourier y wavelet.

Finalmente se revisarán conceptos básicos de cuantificadores que serán el pilar del análisis cuantitativo – comparativo que se realizará más adelante.

1.1. Filtro Digital

Es un algoritmo matemático, cuyo objetivo es ofrecer un proceso selectivo en frecuencias de la señal de entrada ya sea suprimiendo o atenuando ciertos componentes de una señal. Es decir, permite el paso de un cierto rango de frecuencias.



Figura 1.1: Representación básica de un filtro

A continuación se revisarán los conceptos de los tipos de filtros que se usan dentro del análisis de este proyecto.

1.1.1. Filtros Pasa Alto(HPF)

Elimina todas las frecuencias desde cero hasta la frecuencia de corte, f_c y permite el paso de todas las frecuencias por encima de de dicho valor.

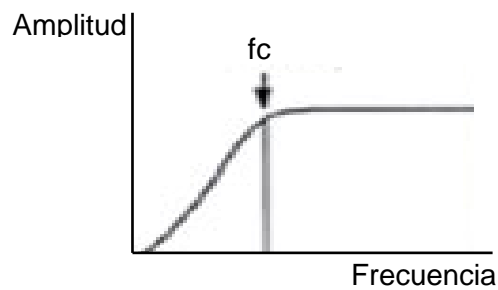


Figura 1.2: Filtro Pasa Alto

1.1.2. Filtros Pasa Bajo(LPF)

Hace exactamente lo contrario que el filtro pasa alto, es decir se caracteriza por permitir el paso de las frecuencias más bajas y atenuar las frecuencias más altas.

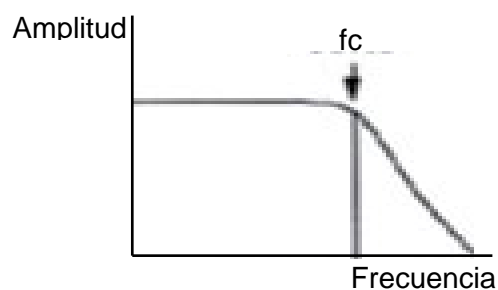


Figura 1.3: Filtro Pasa Bajo

1.2. Transformada de Fourier [1]

La transformada de Fourier se utiliza para pasar al «dominio frecuencial» una señal, y de esta forma obtener información que no es evidente en el «dominio temporal». El análisis de Fourier es muy útil, debido al contenido de frecuencias en la señal.

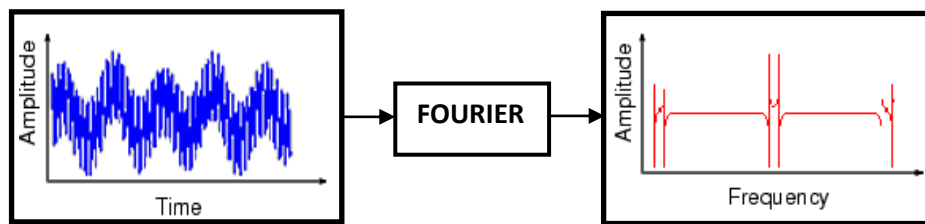


Figura 1.4: Representación de la Transformación de Fourier

1.2.1. Limitaciones de la transformada de Fourier

Fourier descompone una señal en un conjunto de ondas sinusoidales de diferentes frecuencias; sin embargo se pierde la información temporal al pasar al dominio frecuencial. Es imposible decir cuándo ocurrió un evento particular.

Si las propiedades de la señal no cambian mucho con el tiempo, esto es, si la señal es estacionaria, no importa mucho. Señales cuyo espectro varía con el tiempo (señales no estacionarias) presentan tendencias, cambios bruscos, y comienzos y terminaciones de eventos, para los cuales el análisis de Fourier no es adecuado.

1.2.2. Ventana de Fourier

La ventana de Fourier divide la señal en pequeñas ventanas, a las que se les aplica por separado la función de Fourier gracias a ello se logra una mejor localización de la aparición de una singularidad en una señal. El inconveniente es que una vez elegido el tamaño de ventana de tiempo, este es el único que se le aplica a toda la señal. Por lo que sólo se conocerá en qué intervalo de tiempo se produce la singularidad, debido a que la localización depende del ancho elegido para la función ventana. Además, los eventos no podrán ser resueltos si aparecen muy cerca unos de otros, ya que no será posible distinguir diferentes comportamientos dentro de una misma amplitud de ventana.

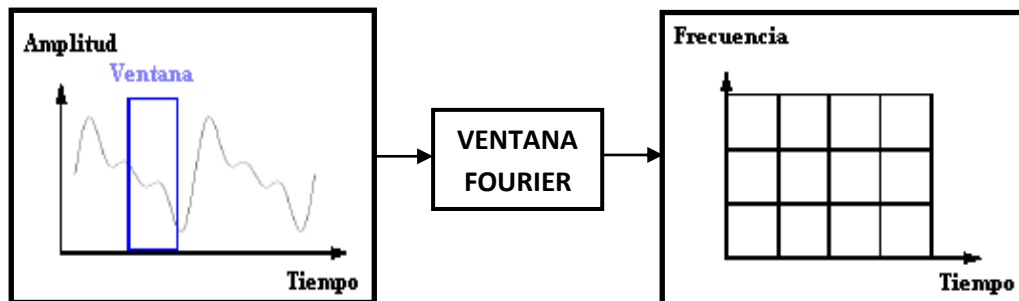


Figura 1.5: Transformación a través de la ventana de fourier

1.3. Transformada Wavelet

Es una forma de onda que representa una señal en términos de versiones trasladadas y dilatadas de una onda finita (limitada) (denominada ondula madre).

Es capaz de revelar aspectos de los datos como tendencias, puntos de quiebre, discontinuidades en las derivadas, y auto-similaridad.

El análisis de wavelets puede muchas veces comprimir o eliminar ruido sin degradación apreciable descomponiendo una señal en versiones escaladas móviles de la wavelet original (“madre”).

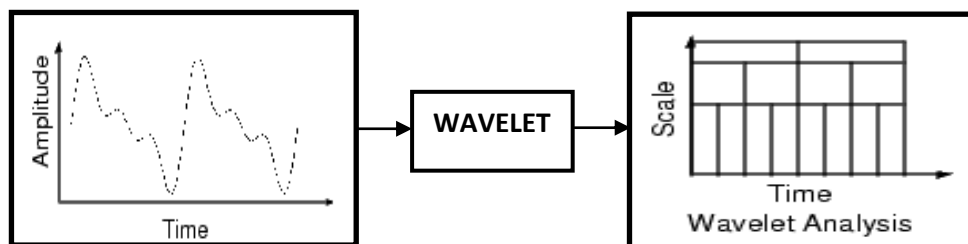


Figura 1.6: Transformación al dominio Wavelet

1.3.1. Transformada Continua Wavelet

Se define como la suma de todo el tiempo de la señal multiplicado por la escala, cambiando la función wavelet. Da como resultado coeficientes wavelets, que están en función de la escala y la posición.

$$C(\text{scale}, \text{position}) = \int_{-\infty}^{\infty} f(t)\psi(\text{scale}, \text{position}, t)dt$$

C es la suma sobre toda la señal multiplicada por versiones móviles, escaladas, de la función **wavelet** ψ . La C se llama transformada continua de **wavelet** (**CWT**). Nótese que el análisis de wavelet no utiliza una región de tiempo-frecuencia, sino una de

tiempo-*escala*. Escalar un wavelet significa estirarlo o encogerlo, se le denomina **factor de escala** y se representa por a .

En el seno el factor de escala es fácil de ver:

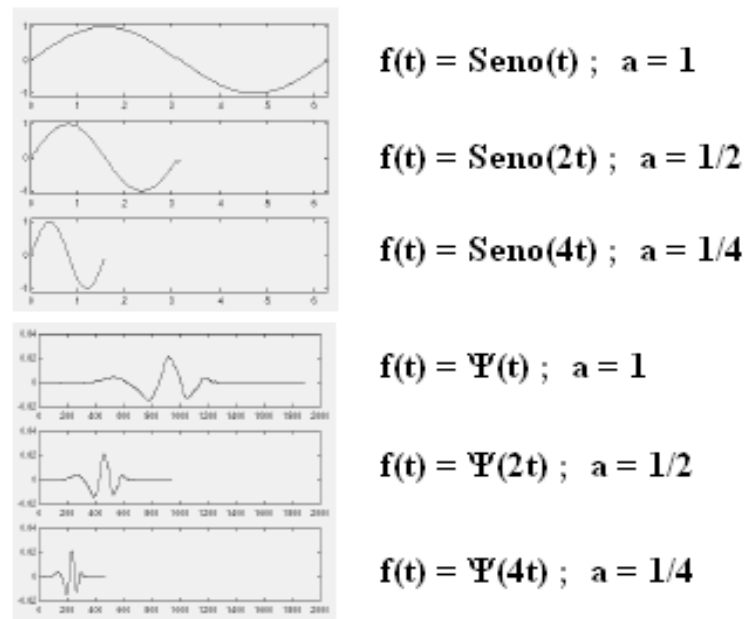


Figura 1.7: Gráficas de funciones de escala

1.3.1.1. Desplazamiento del wavelet

Desplazar un wavelet significa adelantarlo o retrasarlo al original en el tiempo.

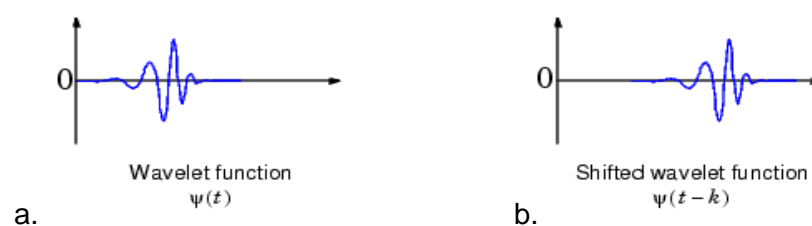


Figura 1.8: a. Función Wavelet, b. Función Wavelet

desplazada

A continuación se describe el proceso del cálculo de la transformada continua de wavelet.

1. Tome una **wavelet** y compárela con una sección al inicio de la señal original.
2. Calcule un número, C , que representa qué tanto se correlaciona la **wavelet** con la sección de la señal. Entre mayor sea C , mayor es la semejanza. C se puede interpretar como el coeficiente de correlación. Hay que hacer notar aquí que los resultados dependen de la forma de la **wavelet** que se elija.
3. Desplazar el wavelet a la derecha y repetir los pasos 1 y 2, hasta cubrir toda la señal.
4. Escale (estire) la **wavelet** y repita los pasos 1 al 3.
5. Repita los pasos 1 al 4 para todas las escalas.

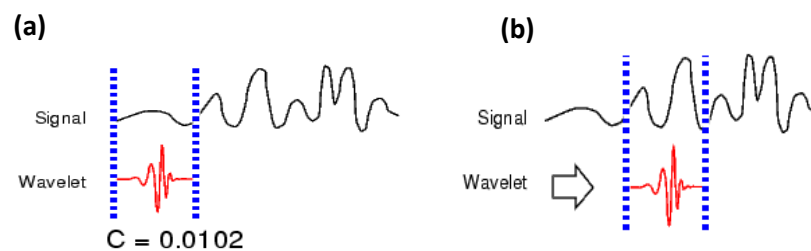


Figura 1.9: (a) Paso 2, (b) Paso 3

Al terminar, se tendrán los coeficientes producidos a diferentes escalas, por las diferentes secciones de la señal. Los coeficientes constituyen los resultados de una regresión de la señal original obtenida por las wavelets.

1.3.1.2. Relación entre escala del wavelet y frecuencia de la señal

- Pequeña escala del wavelet (wavelet comprimido) → se obtiene la alta frecuencia.
- Alta escala del wavelet (wavelet estirado) → se obtiene la baja frecuencia.

1.3.2. *Descomposición de la frecuencia*

Calcular los coeficientes wavelet en todas las posibles escalas es mucho trabajo. Por ello se elige un subconjunto de estiramientos y translaciones. Estos subconjuntos serán potencia de dos.

Una manera de implementar la DWT es utilizando filtros. Se aplican dos filtros, uno de baja frecuencia y otro de alta frecuencia:

- Con el de baja frecuencia se obtienen datos de aproximación, el contenido es importante le da a la señal su identidad.
- Con el de alta frecuencia se obtienen datos de detalle imparten complementación.

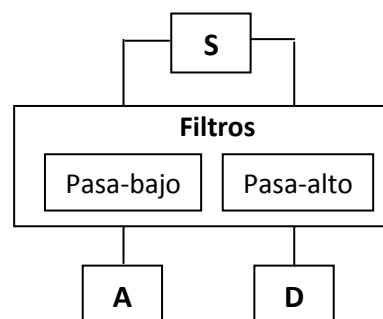


Figura 1.10: Proceso Básico de descomposición wavelet

1.3.2.1. Downsampling

Mediante un submuestreo (downsampling) elimina la duplicación de los datos:

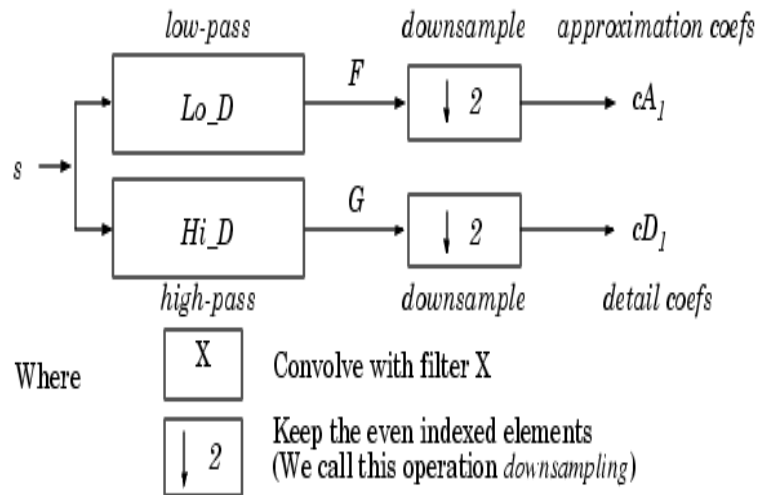


Figura 1.11: Representación en bloques de submuestreo

1.3.2.2. Descomposición en múltiples niveles

El proceso de descomposición puede ser iterado, a fin de que una señal sea descompuesta en muchas componentes de resolución inferior.

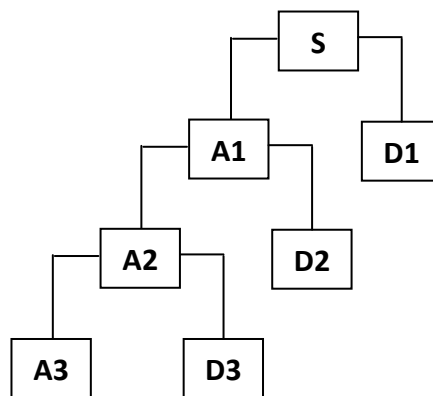


Figura 1.12: Árbol de descomposición de wavelet

1.3.3. Reconstrucción de la señal

Consiste en ensamblar los componentes obtenidos en el proceso de descomposición, obteniendo la señal original sin pérdida de información. A este proceso se le denomina *Transformada Inversa de Wavelet (IDWT)*.

Mientras que en la descomposición se realizaba filtrado y downsampling, y en la reconstrucción se realiza upsampling y filtrado.

1.3.3.1. Upsampling

El sobremuestreo(upsampling) es el proceso de alargar la señal componente insertando ceros entre muestreos.

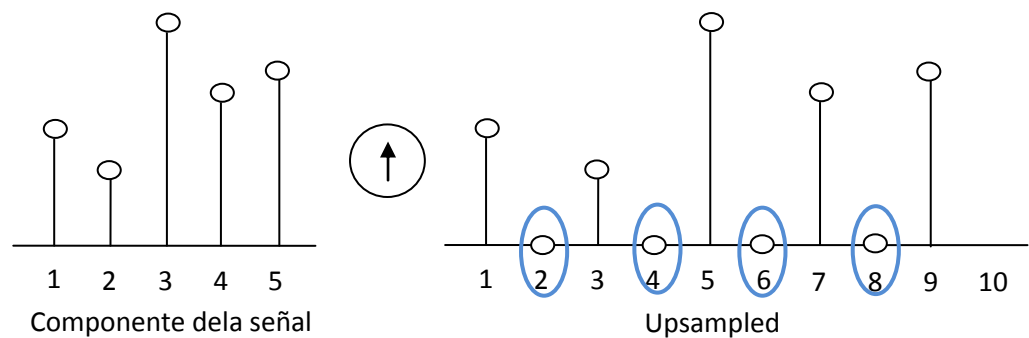


Figura 1.13: Inserción de ceros a la componente de la señal

1.3.3.2. Filtros de Reconstrucción

Al aplicar el downsampling sobre la señal, se obtiene una distorsión, denominada aliasing, en las componentes. Por esto hay que elegir bien los filtros de descomposición y

reconstrucción, ya que deben estar estrechamente relacionados, y así eliminar la distorsión en el proceso de la reconstrucción. La reconstrucción de la señal original se realiza a partir del vector de coeficientes, obtenido en la descomposición.

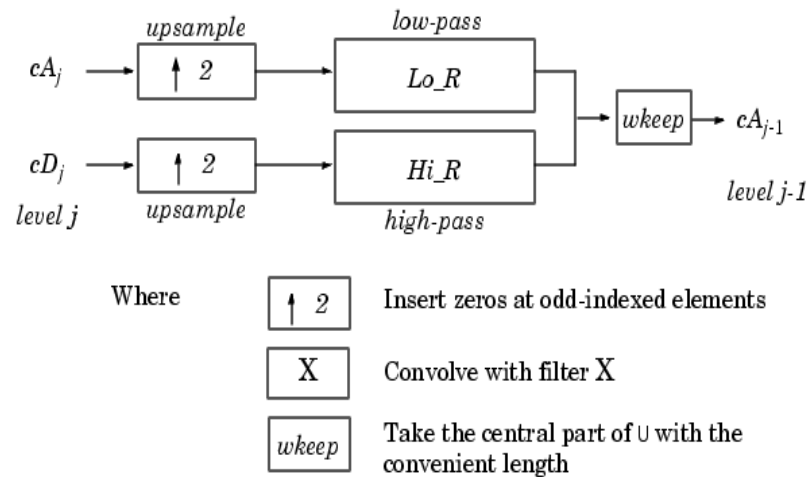


Figura 1.14: Bloques de reconstrucción de la señal

1.4. Cuantificadores [2]

1.4.1. NMSE

Esta es una técnica de análisis numérico que es usada para estimaciones y cuando hay residuos (errores estimados). Este valor se emplea para cuantificar las diferencias entre la imagen original y las imágenes escaladas, es decir este número 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.

1.4.2. *Histograma de una Imagen*

Es un instrumento de medida que ayuda a la visión humana, realizar un diagnóstico de la imagen. El histograma de una imagen contiene la información de la probabilidad de aparición de las distintas tonalidades de color que se pueden dar en cada caso, ya que podemos trabajar en distintos tipos de colores o en escala de grises.

El histograma muestra una serie de 256 líneas verticales, representativas a la cantidad de píxeles que hay en una imagen por cada valor de luminosidad. En el extremo izquierdo está el valor 0 (negro) y en el derecho el 255 (blanco), como indica la barra de degradado que suele acompañarlo.

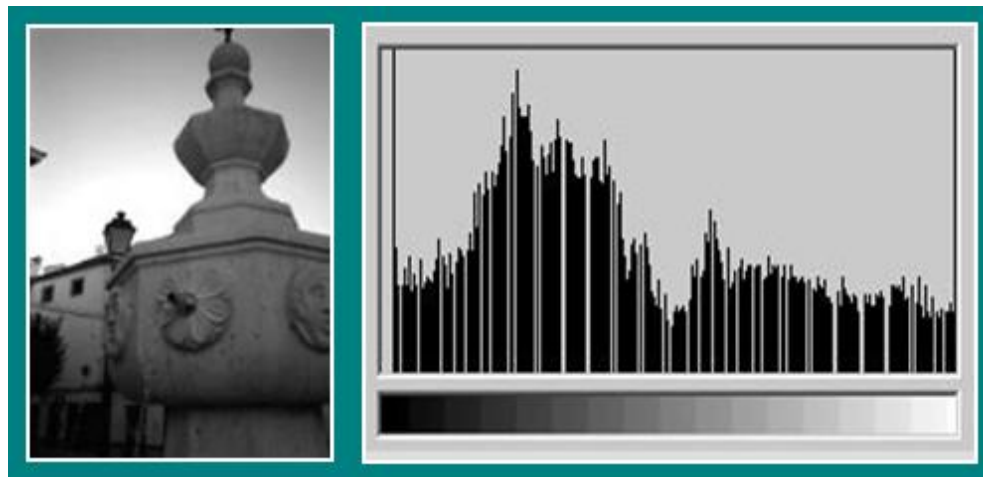


Figura 1.15: Imagen con su respectivo Histograma

Capítulo 2

2. Aplicación de Wavelet a la compresión de Imágenes

La Transformada Discreta de Wavelet (DWT) juega un papel importante en el proceso de compresión de imágenes. A partir de un mismo archivo pueden obtenerse imágenes con diferentes resoluciones.

2.1. Codificación de imágenes [3]

La compresión de imágenes basadas en wavelets puede ser como una forma de codificación subbanda en la cual la imagen es descompuesta en bandas de frecuencia mediante un banco de filtros. Un codificador subbanda es una técnica de codificación donde la señal de entrada es filtrada y separada en bandas de frecuencia.

Si la wavelet utilizada como base, puede concentrar la mayoría de la información visualmente importante en unos pocos coeficientes, los coeficientes restantes pueden truncarse a cero, con una pequeña distorsión de la imagen.

2.1.1. *Descomposición de la Imagen*

Este análisis consiste en descomponer una señal o imagen en una serie de aproximaciones y detalles organizados jerárquicamente en niveles. Al aplicar la DWT se obtiene una

matriz de coeficientes los que se conocen como **coeficientes wavelets**. Si a una imagen x , le aplicamos dicha transformada obtendremos 4 tipos de coeficientes:

Coeficientes de Aproximación, a_1 ; Coeficientes de Detalles Horizontales, h_1 ; Coeficientes de Detalles Verticales, v_1 ; Coeficientes de Detalles Diagonales, d_1 .

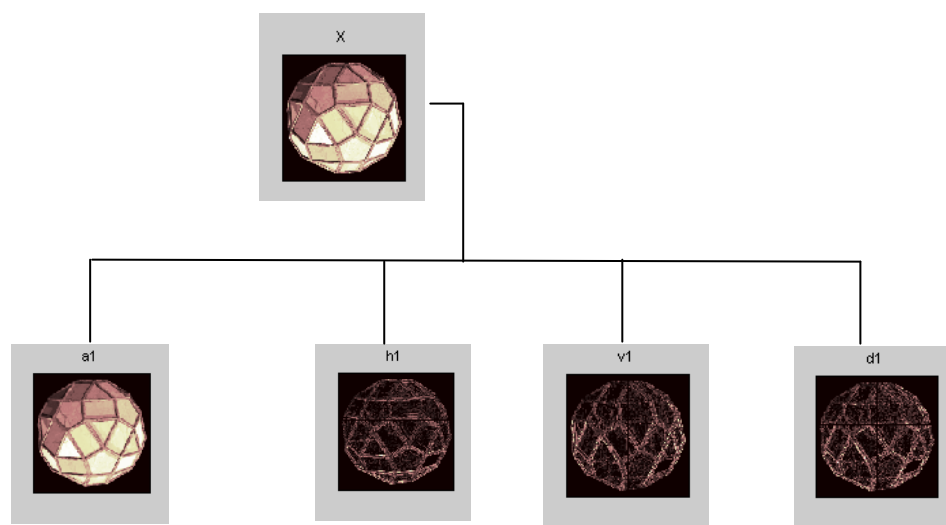


Figura 2.1: Imagen x descompuesta en sus coeficientes de aproximación y detalles.

La aproximación contiene la mayor parte de energía de la imagen, es decir, la información más importante. Los detalles tienen valores próximos a cero.

2.1.2. **Cuantización**

Este proceso sirve para incrementar el número de ceros y reducir la magnitud de los coeficientes de transformación. Por lo que se obtiene una matriz de coeficientes donde existe un mayor número

de ceros. Los coeficientes de la transformada son cuantificados en base a un nivel de umbral para obtener el mayor de número de ceros posibles.

2.1.2.1. Umbral

Es un valor que se puede establecer como límite con algún fin. En compresión de imágenes todos los píxeles cuyo nivel de gris se encuentren por debajo de dicho valor se llevan a cero. El umbral hace posible destacar formas u objetos en una imagen.

2.2. Decodificación de Imágenes [4]

2.2.1. Descuantizador

Se aplica la función de cuantización y a partir de la matriz de coeficientes se obtiene la matriz de coeficientes wavelets, que no tiene que coincidir con la matriz de coeficientes que la obtenida en el proceso de codificación ya que si hemos aplicado un algoritmo con pérdidas hay información que ha sido eliminada.

2.2.2. Reconstrucción

El proceso de reconstrucción de la imagen, se realiza a través de la obtención de la señal a partir de los detalles y aproximaciones. Este proceso se lleva a cabo con la transformada wavelet discreta inversa (idwt).

Se aplica el “upsampling” a los coeficientes de transformación. El mismo que consiste en insertar ceros entre cada muestra.

Capítulo 3

3. Desarrollo del Proyecto

Para el análisis de la transformada wavelet en la compresión de imágenes se ha desarrollado un programa en el software matemático Matlab que realizará la descomposición y reconstrucción de imágenes en escala de grises.

3.1. Elección de Imágenes

Las imágenes que se manipularon para el estudio y análisis del presente proyecto fueron escogidas bajo los siguientes parámetros:

- Dimensiones de la imagen
- Imagen con alto nivel de detalles



Figura 3.1:lena.jpg 256x256



Figura 3.2:estanque.jpg 1024x768



Figura 3.3:barbara.jpg 512x512



Figura 3.4:escalera.jpg 640x480

La figura 3.1 – 3.3 tienen la característica de ser imágenes de dimensiones simétricas, la figura 3.2 – 3.4 son asimétricas.

3.2. Definición de Parámetros

En los capítulos anteriores estudiamos de los parámetros que inciden en la descomposición y reconstrucción de imágenes. A continuación se detallan los parámetros definidos:

Función wavelet: Daubechies, 'db6'. Al usar Fourier este parámetro no está definido.

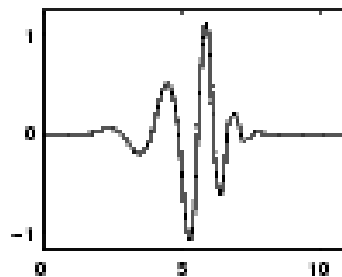


Figura 3.5: Función escalar de la familia Wavelet Daubechies 6

Nivel de descomposición: Se realizaron las pruebas en los cinco primeros niveles.

Umbral: este valor fue calculado en cada nivel, se tomó la mediana del valor absoluto de los coeficientes del detalle horizontal.

3.3. Ejecución del algoritmo

Para realizar la compresión/descompresión de imágenes en Matlab usando la DWT, empezamos creando los filtros pasa-bajos y pasa-altos para la wavelet seleccionada a través del comando *wfilters*.

Los coeficientes wavelet de los detalles obtenidos a través de la transformación discreta de la imagen (*dwt2*) para cada nivel se fueron guardando en un vector y finalmente se adjuntó la aproximación del último nivel.

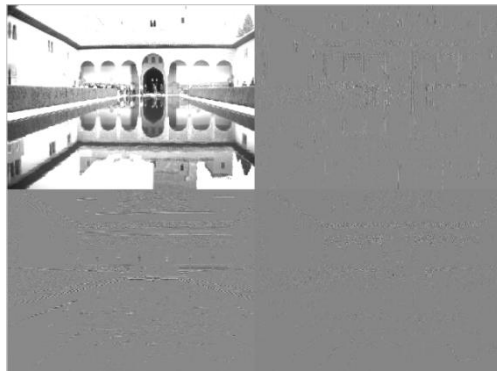


Figura 3.6: Estanque en su descomposición wavelet de nivel 1

Además, se generó una matriz donde se guardaron los tamaños (filas, columnas) correspondientes a los distintos detalles. Gracias a dicha matriz es que se pueden recuperar los coeficientes para efectuar el proceso de reconstrucción de la imagen.

A la par se definió una función de umbralado que fija un valor de referencia para cuantizar los valores de las matrices de detalles generando una gran cantidad de ceros para lograr tasas de

compresión de margen considerable, sin disminuir la calidad de la imagen reconstruida.



Figura 3.7: Estanque en wavelet piramidal de Nivel 2 cuantizada

Para la reconstrucción de la imagen se realizó el proceso inverso, aplicando finalmente el algoritmo de la Transformada discreta Inversa. La implementación del algoritmo (código en Matlab) se muestra en el ANEXO B.

3.4. Funcionalidad del Programa

El programa está diseñado para cargar una imagen en grises, la cual se desea someter al análisis de compresión/descompresión de imágenes. Se podrá descomponer y reconstruir una imagen a través del uso de las wavelets y Fourier. Se puede ingresar el nivel de compresión deseado. La interfaz de interacción se presenta en el ANEXO A.

El programa presentará la descomposición en niveles de la imagen. Mostrará la imagen original y la reconstruida de tal forma que se

realice un examen visual. Además se presentarán los cálculos del NMSE, la energía de las imágenes original y reconstruida y el número de ceros como medición del porcentaje de compresión realizado sobre la imagen. Finalmente generará los histogramas correspondientes a la Imagen Original y Reconstruida para evaluar el cambio que ha ocurrido.

3.5. Resultados

A continuación se presentan los resultados obtenidos al analizar cuatro imágenes previamente escogidas.

TRANSFORMADA	NIVEL	CUANTIFICADOR	IMAGEN			
			lena.jpg	estanque.jpg	barbara.jpg	escalera.jpg
WAVELET	1	NMSE	0,000281523	0,000639574	0,000207746	1,56E-02
		EIO	11588,2	255127	54343,7	59176,9
		EIR	11588,4	255121	54344	59172,9
		COMPRESION	39,4101	42,5584	37,7952	50,1096
	2	NMSE	0,00107884	0,00151567	0,00101827	0,0409147
		EIO	11588,2	255127	54343,7	59176,9
		EIR	11588,2	255115	54343,3	59156,4
		COMPRESION	47,5658	52,9135	46,5315	60,689
	3	NMSE	0,0024311	0,00330147	0,00241058	0,061488
		EIO	11588,2	255127	54343,7	59176,9
		EIR	11588,4	255112	54343,7	59144,3
		COMPRESION	49,5905	55,6536	48,9417	63,1524
	4	NMSE	0,00602354	0,00868416	0,00579663	0,0726403
		EIO	11588,2	255127	54343,7	59176,9
		EIR	11587,5	255092	54344,2	59131
		COMPRESION	50,1048	56,4371	49,6221	63,7394
	5	NMSE	0,0136286	0,0253748	0,012549	0,0780399
		EIO	11588,2	255127	54343,7	59176,9
		EIR	11585,6	254903	54344,9	59129,4
		COMPRESION	50,3611	56,6828	49,8541	63,9089

Tabla 3.1: Resultados obtenidos de las imágenes con el análisis wavelet.

TRANSFORMADA	NIVEL	CUANTIFICADOR	IMAGEN			
			lena.jpg	estanque.jpg	barbara.jpg	escalera.jpg
FOURIER	1	NMSE	0,0605467	0,0539571	0,0890082	0,143324
		EIO	11588,2	255,127	54343,7	59176,9
		EIR	11588,3	254989	54337,4	59081,7
		COMPRESION	45,7642	40,5376	47,5075	39,1969
	2	NMSE	0,127273	0,102267	0,137161	0,250657
		EIO	11588,2	255127	54343,7	59176,9
		EIR	11580,2	254777	54332,4	58998,1
		COMPRESION	56,9748	50,8719	58,754	49,8809
	3	NMSE	0,212488	0,152635	0,178312	0,298434
		EIO	11588,2	255127	54343,7	59176,9
		EIR	11559,8	254370	54315,7	58882,7
		COMPRESION	59,8251	53,3728	61,6806	52,4583
	4	NMSE	0,309855	0,225412	0,239617	0,336313
		EIO	11588,2	255127	54343,7	59176,9
		EIR	11464,9	253596	54258,5	58903,5
		COMPRESION	60,5499	53,9448	62,4187	53,0882
	5	NMSE	0,373707	0,31569	0,315089	0,382425
		EIO	11588,2	255127	54343,7	59176,9
		EIR	10991,3	251210	54045,9	58383,9
		COMPRESION	60,7361	54,0888	62,5999	53,2744

Tabla 3.2: Resultados obtenidos de las imágenes con el análisis fourier.

3.6. Análisis Cuantitativo

Se tomaron los resultados para realizar las gráficas que se presentan a continuación y predecir el comportamiento del método de compresión aplicado en las imágenes.

A continuación se presentará el análisis comparativo del NMSE en las imágenes dentro de los cinco primeros niveles de descomposición entre las transformadas de estudio.

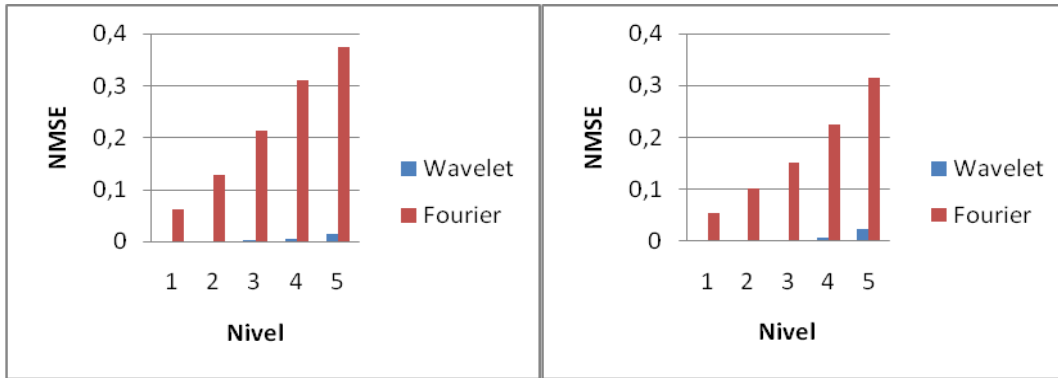


Figura 3.8: NMSE de Lena

Figura 3.9: NMSE de Estanque

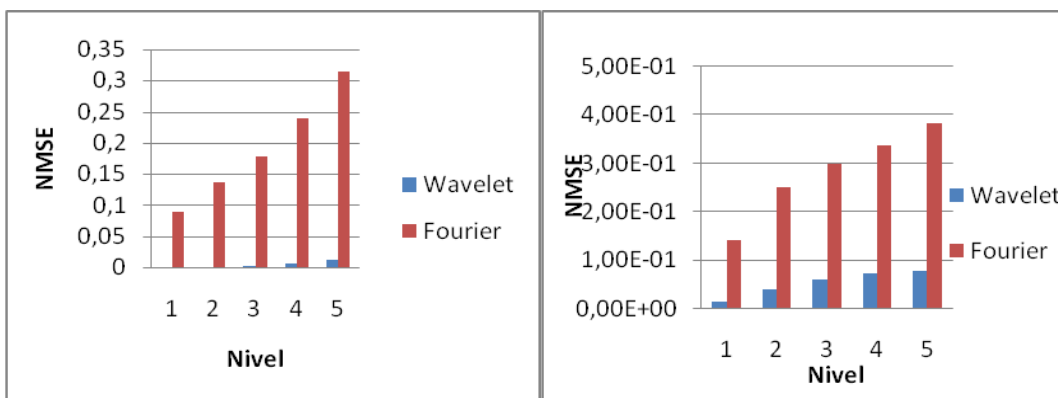


Figura 3.10: NMSE de Bárbara **Figura 3.11:** NMSE de Escalera

Al observar las barras que describen el resultado del NMSE, podemos fijarnos que este valor en wavelet es muy pequeño relacionado con el valor obtenido al aplicar la transformada de Fourier en las imágenes. Recordemos que mientras mayor sea el NMSE mayor será el error.

Ahora presentaremos un análisis comparativo entre las Energías de la imagen original y su reconstruida con ambas transformadas dentro de los cinco primeros niveles de descomposición. Para ello hemos tomado datos de la imagen de lena.jpg y escalera.jpg

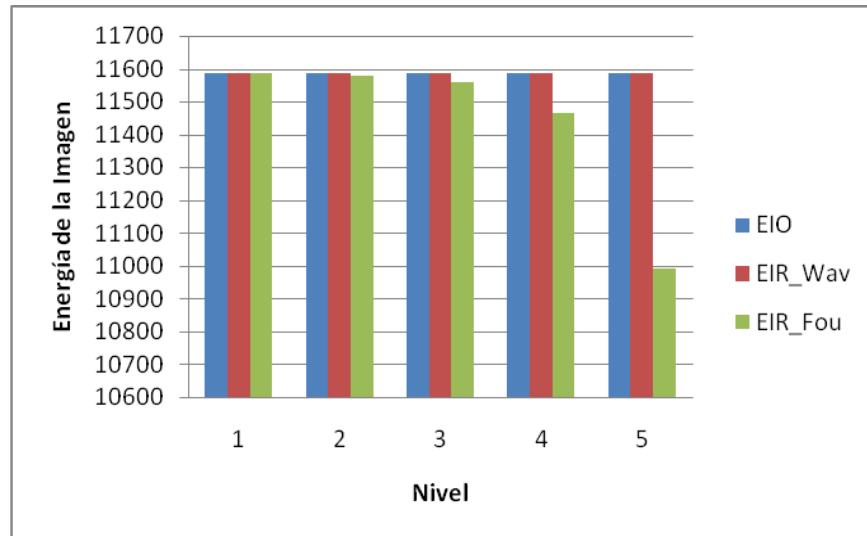


Figura 3.12: Energía de las imágenes de Lena

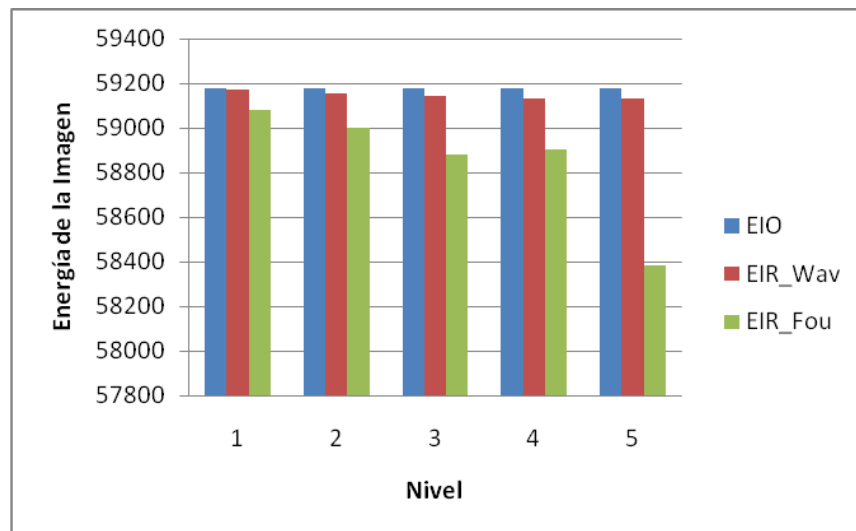


Figura 3.13: Energía de las imágenes de escalera

Con estas gráficas vemos que las imágenes reconstruidas con wavelets tienen una energía que difiere en un mínimo valor. En Fourier podemos argumentar que mientras más aumentamos el nivel de descomposición es mucho mayor la diferencia con respecto a la original. Podemos predecir entonces que la imagen reconstruida a ese nivel ha de haber tomado un gran cambio

perceptible al ojo humano y lo comprobaremos en el análisis visual que describiremos más adelante.

Finalmente presentaremos el análisis comparativo de la Tasas de Compresión alcanzada en la imágenes dentro de los cinco primeros niveles de descomposición entre las transformadas.

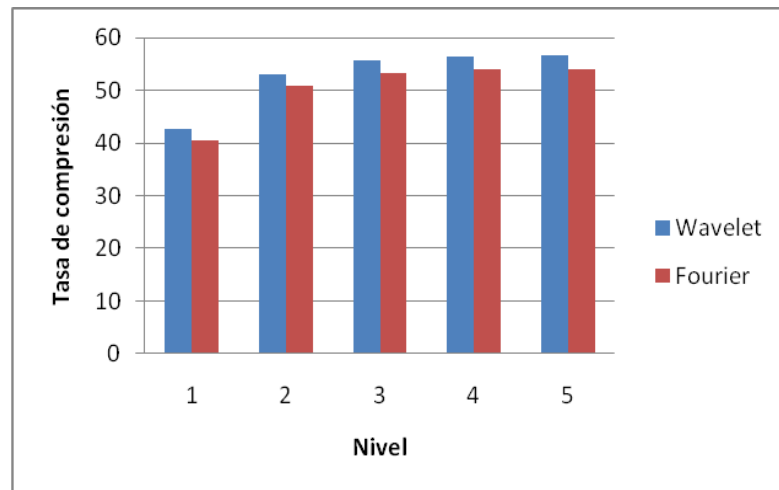


Figura 3.14: Tasa de compresión alcanzada en estanque.jpg

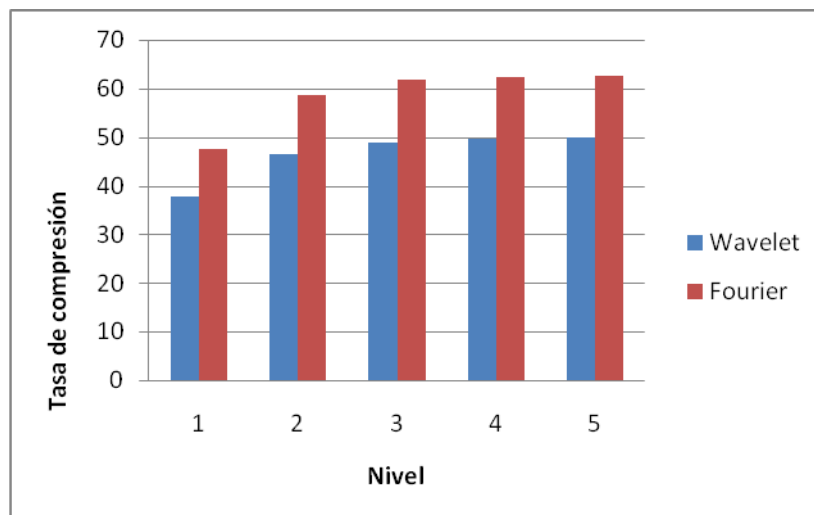


Figura 3.15: Tasa de compresión alcanzada en barbara.jpg

Encontramos que wavelet consigue una mayor tasa de compresión en imágenes asimétricas con respecto a Fourier y en imágenes sucede lo opuesto.

3.7. Análisis Cualitativo

Después de haber analizado cuantitativamente los cambios que han sufrido las imágenes al ser sometidas a la compresión/descompresión; habíamos predicho que mientras aumentamos el nivel de compresión la imagen va teniendo cambios que se van acentuando, los mismos que son más notorios en Fourier. Ahora vamos a tomar como muestra la imagen de Bárbara manipulada en nivel tres de compresión y procederemos a realizar un examen visual para corroborar nuestra predicción a nivel cuantitativo.

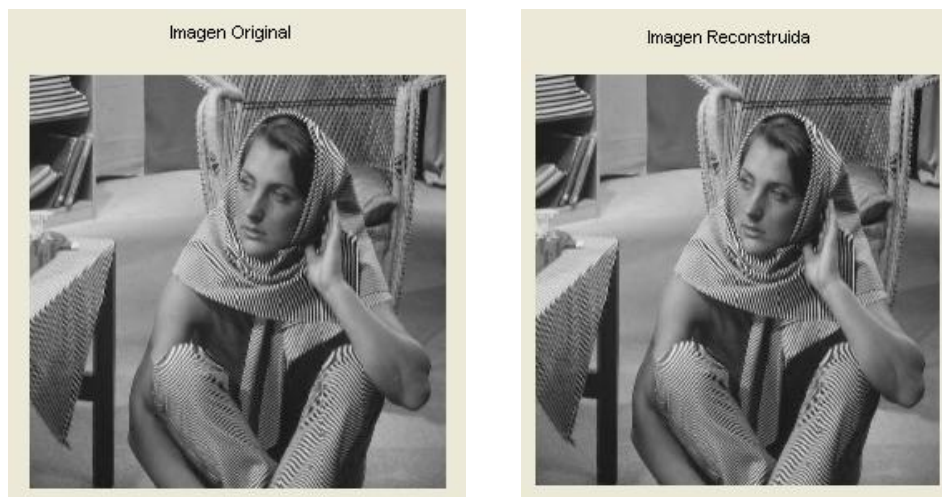


Figura 3.8: Barbara Original y Reconstruida desde el nivel 3 con wavelet

Al observar las imágenes arriba, no hemos notado diferencia entre las mismas. Bárbara tiene la característica de ser una imagen con muchos detalles, los mismos que se han conservado después de someterse a la compresión con wavelet según nuestra apreciación visual, si comparamos esto con el resultado obtenido en la tabla 3.1 donde el NMSE alcanzado fue de un 0.00241058 que nos indicaba que la variación de la imagen a ese nivel es mínima con respecto a la original o incluso si revisamos los valores de energía de la imagen en ese nivel podemos encontrar que obtuvimos un valor igual tanto en su imagen original como en la reconstruida de esta forma comprobamos que nuestros resultados cualitativos van a la par del análisis realizado previamente.

Ahora comprobaremos los resultados en Fourier.



Figura 3.9: Barbara Original y Reconstruida desde el nivel 3 con fourier

A simple vista detectamos una distorsión de la imagen. Hemos notado la pérdida de detalles claramente en el mantel, pantalón y el manto sobre la cabeza de Bárbara, así mismo retomando los datos de la tabla 3.2 tenemos que el NMSE calculado fue de 0.178312 cuyo valor es mayor al obtenido mediante la wavelet y que respalda el cambio que hemos observado.

Para tener una percepción general hemos realizado una encuesta a 30 personas, centrándonos básicamente en conocer cuál es la reacción de esta población al observar el resultado de cuatro imágenes sometidas al análisis de compresión/descompresión de nivel uno. Hemos tomado como estudio este nivel debido que hasta la evaluación que hemos realizado es en el que ambas transformadas se han desempeñado con mejor eficacia en mantener una la información de las imágenes. La encuesta y los resultados obtenidos los encontrará en el ANEXO C.

Particularmente, el 100% de los encuestados encontró que en las tres primeras imágenes era clara que la opción A era la más cercana a ser la original. En la cuarta imagen encontramos que al menos el 30% se inclinó por la opción B.

De manera notoria el 100% de los encuestados se inclinó por el uso de la opción A como mejor forma de

compresión/descompresión después de haber analizado las cuatro imágenes.

Finalmente los factores que más afectan a este análisis según el criterio de los evaluadores fueron: Nivel de compresión, tamaño de la imagen y detalle de la imagen siendo este último el que más coincidencia alcanzó dentro de la población con un 77%.

Conclusiones

- 1) Comprobamos que la reconstrucción de la imagen a través de wavelet se logra con un margen de error muy pequeño prácticamente imperceptible para el sistema de la visión humana.
- 2) Destacamos que, si bien dentro del proceso está involucrada la cuantización, produciéndose como consecuencia la pérdida de información, la imagen que logra reconstruirse con wavelet posee poca pérdida de detalle a las grandes razones de pérdida de Fourier.
- 3) Aún cuando Fourier, en imágenes simétricas alcanza un mayor porcentaje de compresión no es la mejor vía debido a sus grandes pérdidas de información.
- 4) Después de evaluar todos los resultados obtenidos hemos concluido que utilizar la transformada de wavelet es un método muy efectivo para la compresión/descompresión de imágenes debido a la calidad de la imagen que se puede obtener aún cuando se incremente su nivel de compresión.
- 5) El grado de compresión que alcanzó wavelet en los niveles analizados fue de un promedio del 50 %, de esta forma se puede transmitir imágenes ahorrando ancho de banda sin mayor pérdida de información.
- 6) La fidelidad de la imagen dependerá de los niveles de compresión que se alcancen y del tamaño de la imagen y los detalles de la misma.

Recomendaciones

- 1) Se deben considerar imágenes de gran detalle para la evaluación ya que esto permitirá realizar un análisis visual minucioso de los resultados.
- 2) En la compresión mediante wavelet es importante considerar con que familia wavelet se va a implementar el algoritmo debido a que puede ser un factor que influya en obtener una mayor compresión en las imágenes, esto sería un análisis que se podría realizar posteriormente para entablar las beneficios que pueden brindar cada una en este estudio.

ANEXOS

ANEXO A

Interactuando con la interfaz [5]

La interfaz está diseñada para realizar el proceso de compresión/descompresión de imágenes usando la transformada de Wavelet y/o Fourier.

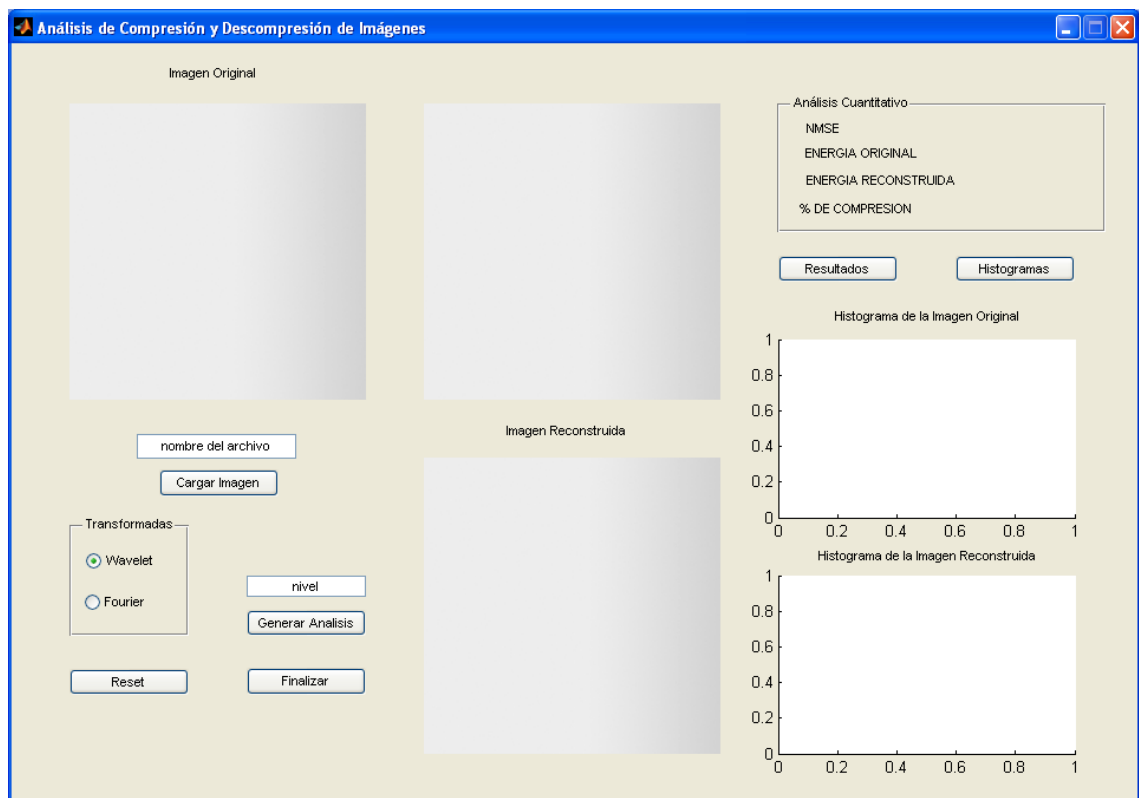


Figura A-1.- Interfaz gráfica inicial del programa

Ingreso de la Imagen

Nombre del archivo: Se debe cargar el archivo que se desea someter a la compresión/descompresión. Se ingresa el nombre del archivo y luego se da click en Cargar Imagen.



Figura A-2.- Ingreso y carga de la Imagen

Parámetros configurables por el usuario:

Selección de la Transformada: El usuario puede seleccionar la transformada con la que desea realizar la compresión/descompresión.

Ingreso del nivel: El usuario debe ingresar el nivel de descomposición que desea alcanzar. Una vez ingresado este valor da click en el botón de Generar Análisis para proceder a la primera parte de la ejecución del programa donde obtendremos las imágenes de descomposición en niveles y la reconstrucción de la imagen.

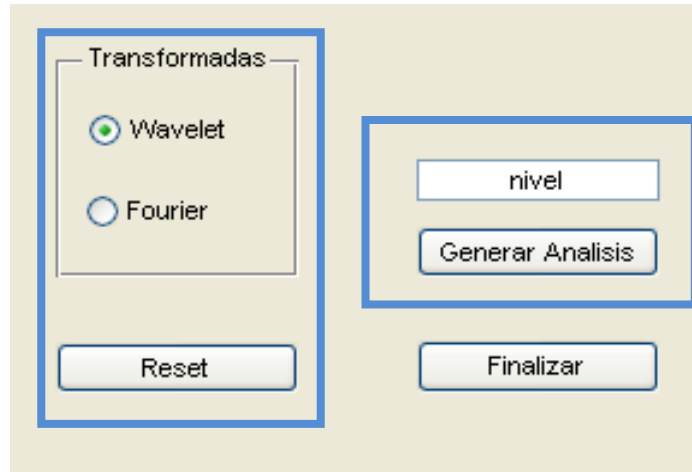


Figura A-3.- Selección de la transformada e ingreso del nivel de descomposición.

Análisis de comparación entre la imagen original y reconstruida:

Resultados: Al dar clic en el botón de resultados, el usuario obtendrá los valores del PSNR, la energía de las imágenes, el número de ceros como porcentaje de compresión.

Histogramas: Al hacer clic en histogramas, el usuario visualizará los histogramas correspondientes a las imágenes para apreciar el cambio que ha sufrido la imagen después de la compresión/reconstrucción.

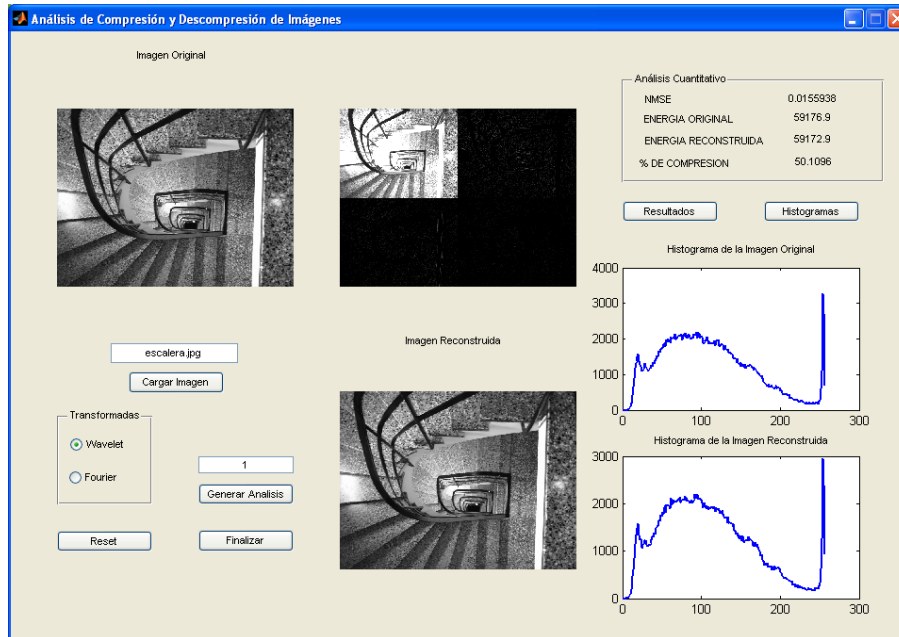


Figura A-4.- Resultado final del programa al someter a la imagen de escalera a la compresión/descompresión de imágenes usando wavelet de nivel uno.

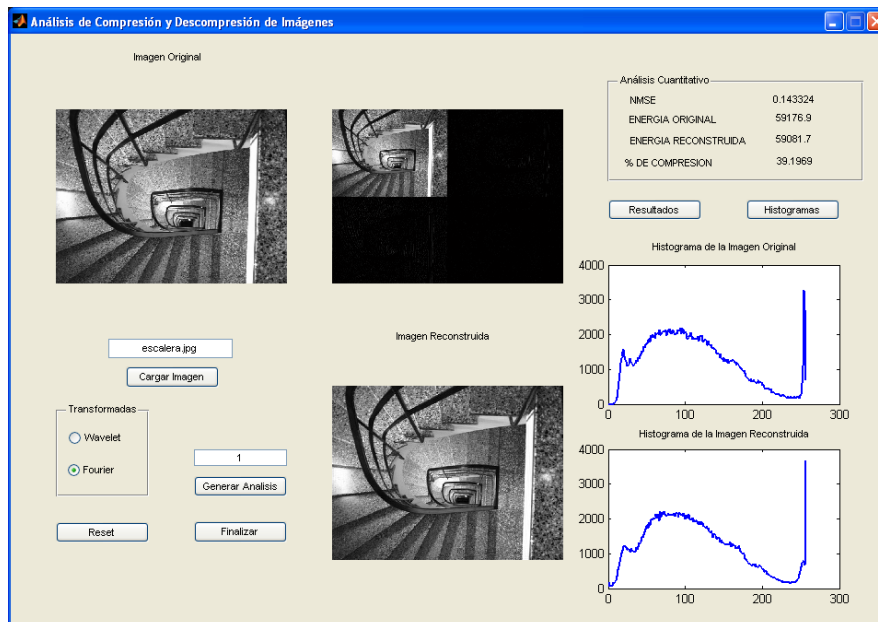


Figura A-5.- Resultado final del programa al someter a la imagen de lena a la compresión/descompresión de imágenes usando fourier de nivel uno.

ANEXO B

Código Fuente del Programa

```

function varargout = proyecto(varargin)
% PROYECTO M-file for proyecto.fig
%     PROYECTO, by itself, creates a new PROYECTO or raises the
existing
%     singleton*.
%
%     H = PROYECTO returns the handle to a new PROYECTO or the handle to
%     the existing singleton*.
%
%     PROYECTO('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in PROYECTO.M with the given input
arguments.
%
%     PROYECTO('Property','Value',...) creates a new PROYECTO or raises
the
%     existing singleton*. Starting from the left, property value pairs
are
%     applied to the GUI before proyecto_OpeningFunction gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to proyecto_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help proyecto

% Last Modified by GUIDE v2.5 31-Oct-2009 12:58:41

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @proyecto_OpeningFcn, ...
                  'gui_OutputFcn',  @proyecto_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before proyecto is made visible.
function proyecto_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to proyecto (see VARARGIN)

% Choose default command line output for proyecto
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

Background_gui(hObject, handles);
% inicializa parametros por defecto
initialize_gui(hObject, handles, false);

% UIWAIT makes proyecto wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = proyecto_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function Background_gui(fig_handle, handles)
background = imread('fondo_gris.jpg');
set(handles.Imagen, 'HandleVisibility', 'ON');
axes(handles.Imagen);
imshow(background)
axis equal; axis tight;
set(handles.Imagen, 'HandleVisibility', 'OFF');

set(handles.Niveles_Imagen, 'HandleVisibility', 'ON');
axes(handles.Niveles_Imagen);
imshow(background)
axis equal; axis tight;
set(handles.Niveles_Imagen, 'HandleVisibility', 'OFF');

set(handles.Reconstruida_Imagen, 'HandleVisibility', 'ON');
axes(handles.Reconstruida_Imagen);
imshow(background)
axis equal; axis tight;
set(handles.Reconstruida_Imagen, 'HandleVisibility', 'OFF');

function initialize_gui(fig_handle, handles, isreset)
% If the metricdata field is present and the reset flag is false, it
means
% we are we are just re-initializing a GUI by calling it from the cmd
line
% while it is up. So, bail out as we dont want to reset the data.
if isfield(handles, 'metricdata') && ~isreset
return;
end
%Configuracion de parametros por defecto
handles.metricdata.Transformada = 'Wavelet';
set(handles.uipanel2, 'SelectedObject', handles.Wavelet_Transf);

```

```

handles.metricdata.Imagen_Original = [];
handles.metricdata.Imagen_Reconstruida = [];
handles.metricdata.WavCoe = [];
handles.metricdata.FouCoe = [];
% Update handles structure
guidata(handles.figure1, handles);

% --- Executes on button press in Carga_Imagen.
function Carga_Imagen_Callback(hObject, eventdata, handles)
% hObject    handle to Carga_Imagen (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    Imagen_N = get(handles.Nombre_Imagen, 'String');
    if ~isempty(Imagen_N)
        handles.metricdata.Imagen_Original =
imread(char(Imagen_N));
        set(handles.Imagen, 'HandleVisibility', 'ON');
        axes(handles.Imagen);
        imshow(handles.metricdata.Imagen_Original)
        axis equal; axis tight; axis off;
        set(handles.Imagen, 'HandleVisibility', 'OFF');
    end;
    guidata(hObject, handles);

function Nombre_Imagen_Callback(hObject, eventdata, handles)
% hObject    handle to Nombre_Imagen (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of Nombre_Imagen as
text
%         str2double(get(hObject, 'String')) returns contents of
Nombre_Imagen as a double

% --- Executes during object creation, after setting all properties.
function Nombre_Imagen_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Nombre_Imagen (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in Cerrar_Programa.
function Cerrar_Programa_Callback(hObject, eventdata, handles)
% hObject    handle to Cerrar_Programa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    opc = questdlg('¿Está seguro de salir del
programa?', 'Salir', 'Si', 'No', 'No')
    if strcmp(opc, 'No')

```

```

        return;
    end
    clear, clc, close all

function Nivel_Descomposicion_Callback(hObject, eventdata, handles)
% hObject    handle to Nivel_Descomposicion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
Nivel_Descomposicion as text
%         str2double(get(hObject,'String')) returns contents of
Nivel_Descomposicion as a double

% --- Executes during object creation, after setting all properties.
function Nivel_Descomposicion_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Nivel_Descomposicion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Main.
function Main_Callback(hObject, eventdata, handles)
% hObject    handle to Main (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    Nivel = str2double(get(handles.Nivel_Descomposicion,'String'));
    if isnan(Nivel)
        error('Error, este es un parámetro numérico.')
        set(handles.Nivel_Descomposicion,'String','nivel')
        %while (n_compresion<1 | n_compresion>lev_min)
        %    procedimiento=input(' Por favor ingrese un nivel de
compresión adecuado: ');
        %end
        % Obtención de los coef wavelets
    elseif (Nivel==0)
        error('Error, este valor no es un nivel de
descomposición.')
        set(handles.Nivel_Descomposicion,'String','nivel')
    else
        if (handles.metricdata.Transformada == 'Wavelet')
            [handles.metricdata.WavCoe,Pos] =
Descomposicion_Wav(handles.metricdata.Imagen_Original,Nivel);
            % Imagen en el dominio Wavelet
            [Imagen_Wav,Posic] =
Imagen_DWavelet(handles.metricdata.WavCoe,Pos);
            set(handles.Niveles_Imagen,'HandleVisibility','ON');
            axes(handles.Niveles_Imagen);
            imshow((uint8 (Imagen_Wav)))
            axis equal; axis tight; axis off;

```

```

        set(handles.Niveles_Imagen,'HandleVisibility','OFF');
        [handles.metricdata.WavCoe,Pos] =
Coef_DWavelet(Imagen_Wav, Posic,Nivel);
        % Reconstrucción de la imagen
        [handles.metricdata.Imagen_Reconstruida] =
reconstruccion(handles.metricdata.WavCoe,Pos);
        else
            [handles.metricdata.FouCoe,Pos] =
Descomposicion_Fou(handles.metricdata.Imagen_Original,Nivel);
            [Imagen_F,Posic] =
Imagen_DWavelet(handles.metricdata.FouCoe,Pos);
            set(handles.Niveles_Imagen,'HandleVisibility','ON');
            axes(handles.Niveles_Imagen);
            imshow(uint8(Imagen_F))
            axis equal; axis tight; axis off;
            set(handles.Niveles_Imagen,'HandleVisibility','OFF');
            [handles.metricdata.FouCoe,Pos] = Coef_DWavelet(Imagen_F,
Posic,Nivel);
            [handles.metricdata.Imagen_Reconstruida] =
Reconstruccion_Fou(handles.metricdata.FouCoe,Pos);
            end
            set(handles.Reconstruida_Imagen,'HandleVisibility','ON');
            axes(handles.Reconstruida_Imagen);
            imshow(uint8(handles.metricdata.Imagen_Reconstruida))

imwrite(handles.metricdata.Imagen_Reconstruida,'Imagen_reconstruida.jp
g');
        axis equal;
        axis tight;
        axis off;
        set(handles.Reconstruida_Imagen,'HandleVisibility','OFF');
        guidata(hObject,handles)
    end

% --- Executes on button press in Analisis_Cuantitativo.
function Analisis_Cuantitativo_Callback(hObject, eventdata, handles)
% hObject    handle to Analisis_Cuantitativo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%*****
% Resultados

%*****
%calculo del error medio cuadrático normalizado y el pico de señal a
ruido
    [NMSE PSNR]=
mse_psnr_image(handles.metricdata.Imagen_Original,uint8(handles.metric
data.Imagen_Reconstruida));
    [EIO EIR]=
energy_image(handles.metricdata.Imagen_Original,uint8(handles.metricda
ta.Imagen_Reconstruida));

%*****
% Porcentaje de compresión logrado
%(calculado en base a la cantidad de 0s del vector de descomposición)
    if(handles.metricdata.Transformada == 'Wavelet')
        n = length(handles.metricdata.WavCoe);
        ceros = length(find(handles.metricdata.WavCoe==0));
    else

```

```

n = length(handles.metricdata.FouCoe);
ceros = length(find(handles.metricdata.FouCoe==0));
end
porc_ceros = 100 * (ceros/n);

set(handles.Valor_NMSE, 'String', NMSE);
set(handles.Valor_EIO, 'String', EIO);
set(handles.Valor_EIR, 'String', EIR);
set(handles.Porcentaje_COMPRESION, 'String', porc_ceros);
% -----
function uipanel2_SelectionChangeFcn(hObject, eventdata, handles)
% hObject    handle to uipanel2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles = guidata(hObject);
if (hObject == handles.Wavelet_Transf)
    handles.metricdata.Transformada='Wavelet';
else
    handles.metricdata.Transformada='Fourier';
end
guidata(hObject,handles)

% --- Executes on button press in Inicializa.
function Inicializa_Callback(hObject, eventdata, handles)
% hObject    handle to Inicializa (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.Nombre_Imagen, 'String', 'nombre del archivo');
set(handles.Nivel_Descomposicion, 'String', 'nivel');
set(handles.Valor_NMSE, 'String', ' ');
set(handles.Valor_EIO, 'String', ' ');
set(handles.Valor_EIR, 'String', ' ');
set(handles.Porcentaje_COMPRESION, 'String', ' ');
handles.metricdata.Imagen_Original = [];
handles.metricdata.Imagen_Reconstruida = [];
Background_gui(hObject, handles)

set(handles.Hist_ImagR, 'HandleVisibility', 'ON');
axes(handles.Hist_ImagR);
y = 0;
plot(handles.Hist_ImagO, y, 'LineWidth', 2);
set(handles.Hist_ImagR, 'HandleVisibility', 'OFF');

set(handles.Hist_ImagR, 'HandleVisibility', 'ON');
axes(handles.Hist_ImagR);
y = 0;
plot(handles.Hist_ImagR, y, 'LineWidth', 2);
set(handles.Hist_ImagR, 'HandleVisibility', 'OFF');

guidata(hObject, handles);

% --- Executes on button press in Histogramas.
function Histogramas_Callback(hObject, eventdata, handles)
% hObject    handle to Histogramas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

plot(handles.Hist_ImagO,imhist(handles.metricdata.Imagen_Original),'LineWidth',2);

plot(handles.Hist_ImagR,imhist(handles.metricdata.Imagen_Reconstruida),
,'LineWidth',2);
    guidata(hObject,handles)

%*****
function [WavCoe, Pos] = Descomposicion_Wav(Img,Lev);
%*****
%Entrada
%     Img : Imagen que se someterá a la descomposición
%     Lev : Nivel de descomposición
%Salida:
%     WavCoe : Vector de coeficientes Wavelets
%     Pos   : Matriz de tamaños
%*****
%Carga la wavelet seleccionada
load db6;
%Construye los filtros correspondiente a la Wavelet seleccionada
%Lo_D Decomposition low-pass filter
%Hi_D Decomposition high-pass filter
[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('db6');
%borra los filtros de reconstruccion que no se van a usar
clear Lo_R Hi_R
%Se inicializan Variables
cA = Img; cH = []; cV = []; cD = [];
Pos = []; WavCoe = [];
Pos = [size(cA)];
%Descomposición en niveles
for j = 1:Lev,
    [cA,cH,cV,cD] = dwt2(cA,Lo_D,Hi_D);
    %figure, imshow(uint8(cA))
    [cH,cV,cD] = Umbralado(cH,cV,cD);
    %figure, imshow(xSWT)
    %Acumulador de dimensiones
    Pos = [size(cH);Pos];
    %Acumulador de coeficientes
    WavCoe = [cH(:);cV(:);cD(:);WavCoe ];
end;
%f = idwt2(cA,cH,cV,cD,Lo_R,Hi_R);
%figure, imshow(uint8(f))
Pos = [size(cA);Pos];
WavCoe = [cA(:);WavCoe]';
return;

```



```

%*****
function [WavCoe, Pos] = Descomposicion_Wav(Img,Lev);
%*****
%Entrada
%       Img : Imagen que se someterá a la descomposición
%       Lev : Nivel de descomposición
%Salida:
%       WavCoe : Vector de coeficientes Wavelets
%       Pos    : Matriz de tamaños
%*****
%Carga la wavelet seleccionada
load db6;
%Construye los filtros correspondiente a la Wavelet seleccionada
%Lo_D Decomposition low-pass filter
%Hi_D Decomposition high-pass filter
[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('db6');
%borra los filtros de reconstruccion que no se van a usar
clear Lo_R Hi_R
%Se inicializan Variables
cA = Img; cH = []; cV = []; cD = [];
Pos = []; WavCoe = [];
Pos = [size(cA)];
%Descomposición en niveles
for j = 1:Lev,
    [cA,cH,cV,cD] = dwt2(cA,Lo_D,Hi_D);
    %figure, imshow(uint8(cA))
    [cH,cV,cD] = Umbralado(cH,cV,cD);
    %figure, imshow(xSWT)
    %Acumulador de dimensiones
    Pos = [size(cH);Pos];
    %Acumulador de coeficientes
    WavCoe = [cH(:);cV(:);cD(:);WavCoe ];
end;
%f = idwt2(cA,cH,cV,cD,Lo_R,Hi_R);
%figure, imshow(uint8(f))
Pos = [size(cA);Pos];
WavCoe = [cA(:);WavCoe]';
return;

```

```

%*****
function [Coef,S] = Coef_DWavelet(Img_DWav, S,level);
%*****
%Entrada
%      Img_DWav :   Imagen en el dominio wavelet
%      S         :
%      Lev       :   Nivel de descomposición
%Salida:
%      Coef      :   Vector de coeficientes Wavelets
%      S         :   Matriz de tamaños
%*****
*****
L = length(S);
Coef = zeros(1,S(1,3)+3*sum(S(2:L-1,3)));
% Coeficientes de Aproximación
Coef(1:S(1,3)) = reshape( Img_DWav( 1:S(1,1) , 1:S(1,2) ), 1 , S(1,3)
);
for k = 2:L-1
    fils = [sum(S(1:k-1,1))+1:sum(S(1:k,1))];
    cols = [sum(S(1:k-1,2))+1:sum(S(1:k,2))];
    % Coeficientes de Detalles Horizontales
    ini = S(1,3) + 3*sum(S(2:k-1,3)) + 1;
    fin = S(1,3) + 3*sum(S(2:k-1,3)) + S(k,3);
    Coef(ini:fin) = reshape(Img_DWav( 1:S(k,1) , cols ) , 1, fin-
ini+1);
    % Coeficientes de Detalles Verticales
    ini = S(1,3) + 3*sum(S(2:k-1,3)) + S(k,3) + 1;
    fin = S(1,3) + 3*sum(S(2:k-1,3)) + 2*S(k,3);
    Coef(ini:fin) = reshape(Img_DWav( fils , 1:S(k,2) ) , 1 , fin-
ini+1 );
    % Coeficientes de Detalles Diagonales
    ini = S(1,3) + 3*sum(S(2:k-1,3)) + 2*S(k,3) + 1;
    fin = S(1,3) + 3*sum(S(2:k,3));
    Coef(ini:fin) = reshape(Img_DWav( fils , cols ) , 1 , fin-ini+1);
end
if (( L - 2) > level) %set those coef. in higher scale to 0
    temp = zeros(1, length(Coef) - (S(1,3)+3*sum(S(2:(level+1),3))));
    Coef(S((level+2),3)+1 : length(Coef)) = temp;
end
save Coef Coef
S(:,3) = [];
return;

```

```

%*****
function [cA] = reconstruccion(WavCoe,Pos);
%*****
% Entrada:
%     WavCoe : Vector de coeficientes waveletes
%     Pos     : Matriz de dimensiones
% salida:
%     [cA]    : Imagen Reconstruida
%*****
load db6;
%Construye los filtros asociados con la wavelet seleccionada
%Lo_D Decomposition low-pass filter
%Hi_D Decomposition high-pass filter
%Lo_R Reconstruction low-pass filter
%Hi_R Reconstruction high-pass filter
[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('db6');
% Borra los filtros de descomposición que no se van a usar
clear Lo_D Hi_D
% Reconstrucción
[fil, col] = size(Pos);
% Cálculo del nivel de que debe alcanzar para la reconstrucción
lev = fil - 2;
ini = 1 ; fin = prod(Pos(1,:));
cA = reshape(WavCoe(ini:fin), Pos(1,1), Pos(1,2)) ;
Pos = Pos(2:fil,:);
for j = 1:lev,
    tamaño = prod(Pos(j,:));
    ini = fin + 1; fin = ini + tamaño - 1;
    cH = reshape(WavCoe(ini:fin), Pos(j,1), Pos(j,2)) ;
    ini = fin + 1; fin = ini + tamaño - 1;
    cV = reshape(WavCoe(ini:fin), Pos(j,1), Pos(j,2)) ;
    ini = fin + 1; fin = ini + tamaño - 1;
    cD = reshape(WavCoe(ini:fin), Pos(j,1), Pos(j,2)) ;
    % Transformada Inversa Wavelet
    cA = idwt2(cA,cH,cV,cD,Lo_R,Hi_R);
    cA = wkeep(cA, Pos(j+1,:));
end;
cA=uint8(cA);
return;

%*****
function [cH,cV,cD] = Umbralado(cH,cV,cD);
%*****
%Entrada
%     cH,cV,cD : Matrices de los detalles de la imagen
%Salida:
%     cH,cV,cD : Matrices de los detalles de la imagen
%*****
*****
%Umbralado
Umb = median(sort(abs(cH(:)))));
cH(find(abs(cH)<Umb)) = 0;
cV(find(abs(cV)<Umb)) = 0;
cD(find(abs(cD)<Umb)) = 0;
return;

```

```

%*****
function [FouCoe, Pos] = Descomposicion_Fou(Img,Lev);
%*****
%Entrada
%       Img : Imagen que se someterá a la descomposición
%       Lev : Nivel de descomposición
%Salida:
%       FCoe : Vector de coeficientes Fourier
%       Pos   : Matriz de tamaños
%*****
x11 = Img; x1h = []; xh1 = []; xhh = [];
Pos = []; FouCoe = [];
Pos = size(x11);
%Descomposición en niveles
for j = 1:Lev,
    [x11,x1h,xh1,xhh] = ideal2x2SWT(x11);
    [x1h,xh1,xhh] = Umbralado(x1h,xh1,xhh);
    Pos = [size(x1h);Pos];
    %Acumulador de coeficientes
    FouCoe = [x1h(:);xh1(:);xhh(:);FouCoe ];
end;
Pos = [size(x11);Pos];
FouCoe = [x11(:);FouCoe]';
return;

%*****
function [x11] = Reconstruccion_Fou(FouCoe,Pos);
%*****
% Entrada:
%       FouCoe : Vector de coeficientes fourier
%       Pos     : Matriz de dimensiones
% salida:
%       [x11]   : Imagen Reconstruida
%*****
[fil, col] = size(Pos);
% Cálculo del nivel de que debe alcanzar para la reconstrucción
lev = fil - 2;
ini = 1 ; fin = prod(Pos(1,:));
x11 = reshape(FouCoe(ini:fin), Pos(1,1), Pos(1,2)) ;
Pos = Pos(2:fil,:);
for j = 1:lev,
    tamaño = prod(Pos(j,:));
    ini = fin + 1; fin = ini + tamaño - 1;
    x1h = reshape(FouCoe(ini:fin), Pos(j,1), Pos(j,2)) ;
    ini = fin + 1; fin = ini + tamaño - 1;
    xh1 = reshape(FouCoe(ini:fin), Pos(j,1), Pos(j,2)) ;
    ini = fin + 1; fin = ini + tamaño - 1;
    xhh = reshape(FouCoe(ini:fin), Pos(j,1), Pos(j,2)) ;
    % Transformada Inversa Wavelet
    [x11] = ideal2x2ISWT(x11,x1h,xh1,xhh);
    x11 = wkeep(x11, Pos(j+1,:));
end;
x11=uint8(x11);
return;

```

ANEXO C

Encuesta de Estudio de Mercado

A continuación detallamos la encuesta realizada donde la opción A representa a wavelet y la opción B a Fourier, cabe recalcar que los encuestados no tenían el conocimiento de que método estaba relacionado con que opción para no influir en sus respuestas. A continuación de la encuesta encontrará los resultados obtenidos.

ENCUESTA DE ESTUDIO DE MERCADO – PROCESAMIENTO DE IMÁGENES DIGITALES “ANÁLISIS, ENTRE LOS ALGORITMOS PARA COMPRESION Y DESCOMPRESION DE IMÁGENES EN ESCALA DE GRISES USANDO LA TRANSFORMADA DE WAVELET CON RESPECTO A LA TRANSFORMADA DE FOURIER“

A continuación usted visualizará el resultado obtenido de algunas imágenes luego de haber sido sometidas a la compresión/descompresión mediante dos métodos.

1. De las siguientes imágenes cual es en su apreciación la que tiene mayor parecido a su imagen original. Escoja una opción.

Imagen original



Opción A**Opción B****Imagen Original****Opción A**

Opción B**Imagen Original****Opción A**

Opción B



Imagen Original



Opción A



Opción B



Marque con una "X la opción que Ud. Considere más apropiada

2. Luego de haber analizado las imágenes anteriormente descritas, según su percepción ¿cuál fue la mejor opción para la compresión/descompresión de imágenes?

- a) A
b) B

3. ¿Cuál cree usted, qué puede ser el factor que afecta más al realizar compresión y descompresión de imágenes digitales?

	Mucho	Poco	Nada
a) Nivel de Ruido	[]	[]	[]
b) Nivel de compresión	[]	[]	[]
c) Tamaño de la imagen	[]	[]	[]
d) Nivel de intensidad	[]	[]	[]
e) Detalle de la imagen	[]	[]	[]

RESULTADO DE LA ENCUESTA DE ESTUDIO DE MERCADO – PROCESAMIENTO DE IMÁGENES DIGITALES “ANÁLISIS, ENTRE LOS ALGORITMOS PARA COMPRESION Y DESCOMPRESION DE IMÁGENES EN ESCALA DE GRISES USANDO LA TRANSFORMADA DE WAVELET CON RESPECTO A LA TRANSFORMADA DE FOURIER“

Numero de encuestados: 30 personas

Fecha: 31 Oct-09 al 5 Nov-09

1.- De las siguientes imágenes cual es en su apreciación la que tiene mayor parecido a su imagen original. Escoja una opción

Imagen	Opción A	Opción B
LENA	30	
ESTANQUE	30	
BARBARA	30	
ESCALERA	21	9

2.- Luego de haber analizado las imágenes anteriormente descritas, según su percepción ¿cuál fue la mejor opción para la compresión/ descompresión de imágenes?

Opción A	Opción B
30	

3.- ¿Cuál cree usted, qué puede ser el factor que afecta más al realizar compresión y descompresión de imágenes digitales?

	Mucho	Poco	Nada
Nivel de Ruido	9	13	8
Nivel de Compresión	20	7	3
Tamaño de la Imagen	15	8	7
Nivel de Intensidad	7	18	5
Detalle de la Imagen	23	4	3

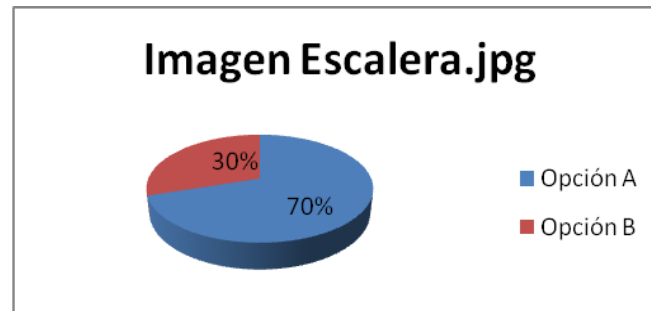


Figura C-1. Resultado del método de compresión/descompresión escogido.

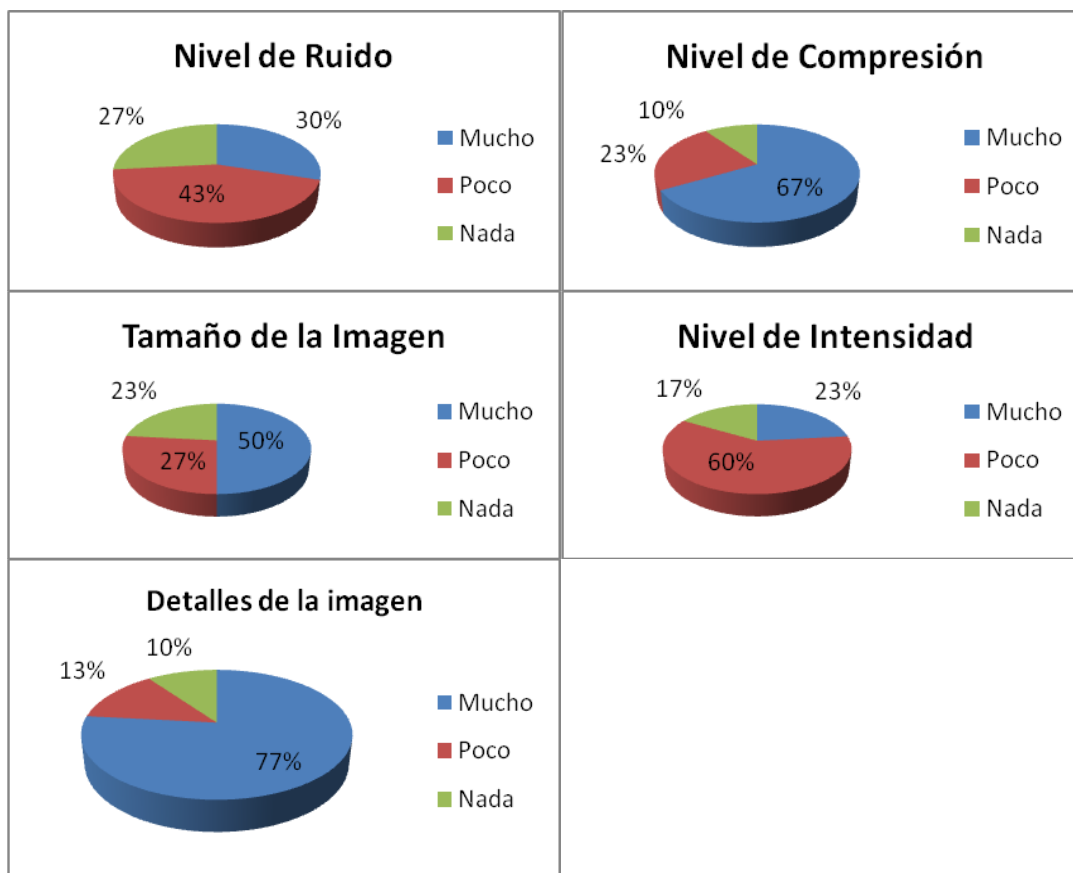


Figura C-2. Resultados de cada uno de los factores que influyen en el análisis de compresión/descompresión de imágenes

Bibliografía

- [1] MADISETTI Vijay K., WILLIAMS Douglas B., “Digital Signal Processing Handbook”, Chapman & Hall/CRCnetBase, 1999.
- [2] WOODS González., “Digital Processing Using MATLAB”, Prentice Hall,2004
- [3] BLANCHET Gerard, CHARBIT Maurice, “Digital Signal and Image Processing using Matlab”, Iste, 2006.
- [4] JAIN Anil K., “Fundamentals of Digital Image Processing”, Prentice Hall, 1989.
- [5] WILEY John & Sons Ltda. MATLAB, “Image Processing Toolbox User’s Guide”, Mathworks Inc., 2008.