

I

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Desarrollo de una herramienta Educativa, para la comprensión y reducción de ruido en imágenes en escala de grises, usando transformada de Wavelet”

INFORME DE MATERIA DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Carlos David Córdova Mendiburo

GUAYAQUIL – ECUADOR

AÑO 2010

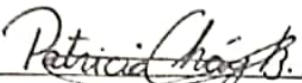
AGRADECIMIENTO


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

DEDICATORIA

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

TRIBUNAL DE SUSTENTACIÓN

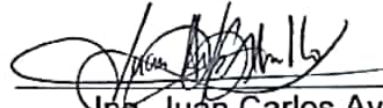

Ing. Patricia Chávez
PROFESORA DE LA
MATERIA


Ing. Juan Carlos Avilés
DELEGADO DEL DECANO

TRIBUNAL DE SUSTENTACIÓN



Ing. Patricia Chávez
PROFESORA DE LA
MATERIA



Ing. Juan Carlos Avilés
DELEGADO DEL DECANO

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.

INDICE GENERAL

RESUMEN	VII
INDICE GENERAL	VIII
INTRODUCCION	XI
CAPÍTULO 1.....	1
2. Transformada de Wavelet	1
2.1 Introducción	1
2.2 Descomposición de la Imagen.....	2
2.3 Código Huffman.....	3
2.4 Reducción de Ruido	4
CAPÍTULO 2.....	5
3. Procedimiento en Matlab	5
3.1 Compresión de la Imagen.....	5
3.2 Reducción de Ruido.....	6
CAPÍTULO 3.....	8
4. Análisis Cualitativo y Cuantitativo de los Datos.....	8
4.1 Introducción	8
4.2 El Análisis.....	13
4.2.1 Compresión.....	13
4.2.2 Compresión.....	14
4.2.3 Mínimo Error Cuadrático Medio Normalizado	16
4.2.4 Análisis Cualitativo	17
4.2.4.1 Encuesta # 1.....	17
4.2.4.2 Encuesta # 2.....	17
4.3 Resultados	18

4.3.1	Cualitativo – Reducción de Ruido	18
4.3.2	Cuantitativo – Compresión de Imágenes	20
4.3.3	Cuantitativo – Reducción de imágenes	24
4.3.4	Resultados Finales.....	26
CONCLUSIONES.....		XII
BIBLIOGRAFIA		XV
ANEXOS		XI
Anexo 1. Manual de usuario		XII
1.	Ingreso a la aplicación.....	- 3 -
2.	Partes de la aplicación	- 5 -
Anexo 2. Código en Matlab.....		- 11 -
Conversión a Escala de Grises.....		- 12 -
Error Cuadrático Medio Normalizado		- 13 -
Compression		- 13 -
Eliminación de Ruido		- 15 -

INDICE DE FIGURAS

Fig. 3.1 Big Ben – Londres.....	19
Fig. 3.2 La Catedral Notedrame	20
Fig. 3.3 El Grito – Edward Munch.....	20
Fig. 3.4 La Noche Estrellada – Van Gogh.....	21
Fig. 3.5 Nacimiento de Venus	21
Fig. 3.6 Partenón de Grecia.....	22
Fig. 3.7 La Persistencia del tiempo – Salvador Dalí.....	22
Fig. 3.8 Imagen convertida en escala de grises.	23
Fig. 3.9 Imagen comprimida con db4 y nivel 4.	24
Fig. 3.10 Imagen con ruido gaussiano.....	25
Fig. 3.11 Imagen con reducción de ruido.....	25
Fig. 3.12 Porcentaje de aceptación de las imágenes en las que se aplico la reducción de Ruido Gaussiano y Speckle.....	34
Fig. 3.13 Porcentaje de aceptación de las imágenes en las que se aplico la reducción de Ruido Poisson y Salt & Pepper.....	34
Fig. 3.14 Promedio Final de aceptación de imágenes.....	35
las que se aplico la reducción de Ruido.....	35
Fig. 3.15 Promedio compresión de imágenes Nivel 1 y 2.....	35
Fig. 3.16 Promedio compresión de imágenes Nivel 3 y 4.....	36
Fig. 3.17 Promedio Final compresión de imágenes.....	36
Fig. 3.18 Promedio Error Cuadrático Medio Normalizado en la compresión de imágenes Nivel 1 y 2.....	37
Fig. 3.19 Promedio Error Cuadrático Medio Normalizado en la compresión de imágenes Nivel 3 y 4.....	37
Fig. 3.20 Promedio Final Error Cuadrático Medio Normalizado en la compresión de imágenes.....	38

INDICE DE TABLAS

Tabla 3.1 Porcentaje de aceptación de las Imágenes con ruido Gaussiano.....	28
Tabla 3.2 Porcentaje de aceptación de las Imágenes con ruido Speckle.....	28
Tabla 3.3 Porcentaje de aceptación de las Imágenes con ruido Salt & Pepper.....	29
Tabla 3.4 Porcentaje de aceptación de las Imágenes con ruido Poisson.....	29
Tabla 3.5 Tamaño de Imagen Big Ben comprimida en Bytes.....	30
Tabla 3.6 Error Cuadrático Medio Normalizado en compresión de la Imagen Big Ben.....	30
Tabla 3.7 Tamaño de Imagen La Catedral comprimida en Bytes.....	30
Tabla 3.8 Error Cuadrático Medio Normalizado en compresión de la Imagen La Catedral.....	31
Tabla 3.9 Tamaño de Imagen El Grito comprimida en Bytes.....	31
Tabla 3.10 Error Cuadrático Medio Normalizado en compresión de la Imagen El Grito.....	31
Tabla 3.11 Error Cuadrático Medio Normalizado en reducción de ruido de la Imagen La Noche Estrellada.....	32
Tabla 3.12 Error Cuadrático Medio Normalizado en reducción de ruido de la Imagen Nacimiento de Venus.....	32
Tabla 3.13 Error Cuadrático Medio Normalizado en reducción de ruido de la Imagen Partenón de Grecia.....	33
Tabla 3.14 Error Cuadrático Medio Normalizado en reducción de ruido de la Imagen Persistencia del Tiempo.....	33

INTRODUCCION

Se define la compresión de imágenes como el proceso por el cual se disminuye el tamaño de almacenamiento de una imagen.

El objetivo de los distintos métodos existentes para la compresión de imágenes es lograr la mayor fidelidad de la imagen comprimida, con respecto a la imagen original.

El presente trabajo se concentra en la compresión usando la transformada de Wavelet. De la misma manera he dedicado parte de la aplicación, para experimentar la eficiencia de la transformada wavelet en la reducción de ruido.

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.

CAPÍTULO 1

1. Transformada de Wavelet

1.1 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.

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.

1.2 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.

CAPÍTULO 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 `wdencmp` 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.

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 `ddencmp` 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 `wdencmp` 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.

CAPÍTULO 3

3. Análisis Cualitativo y Cuantitativo de los Datos.

3.1 Introducción

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



Fig. 3.1 Big Ben – Londres



Fig. 3.2 La Catedral Notedrame

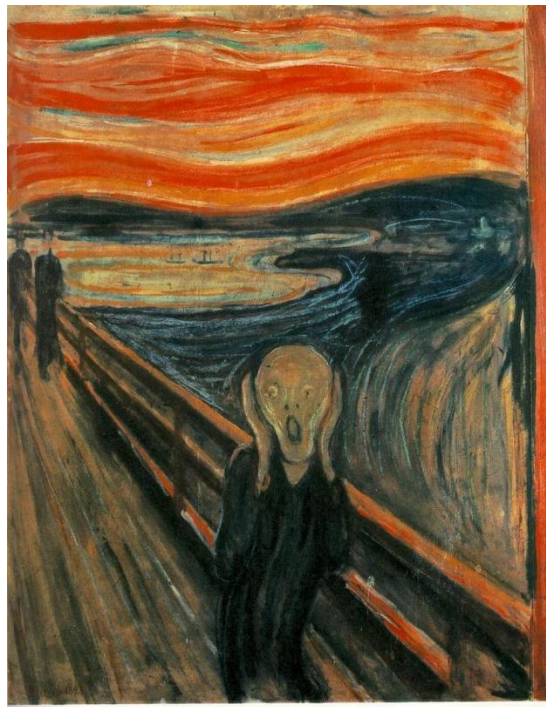


Fig. 3.3 El Grito – Edward Munch



Fig. 3.4 La Noche Estrellada – Van Gogh



Fig. 3.5 Nacimiento de Venus



Fig. 3.6 Partenón de Grecia

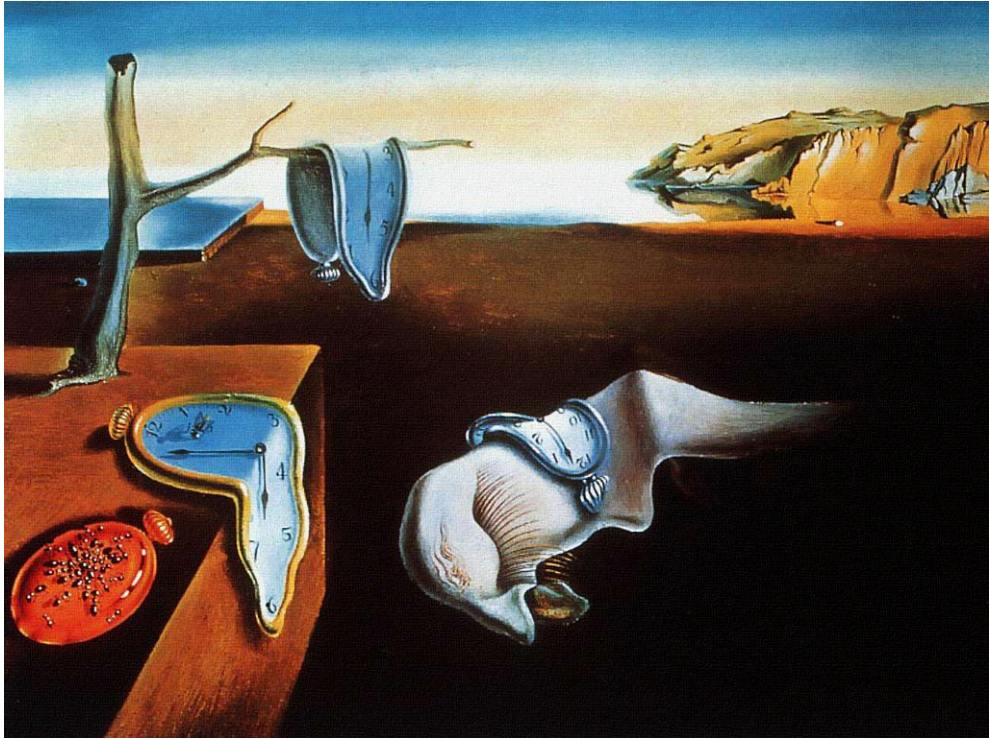


Fig. 3.7 La Persistencia del tiempo – Salvador Dalí

3.2 El Análisis

3.2.1 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.

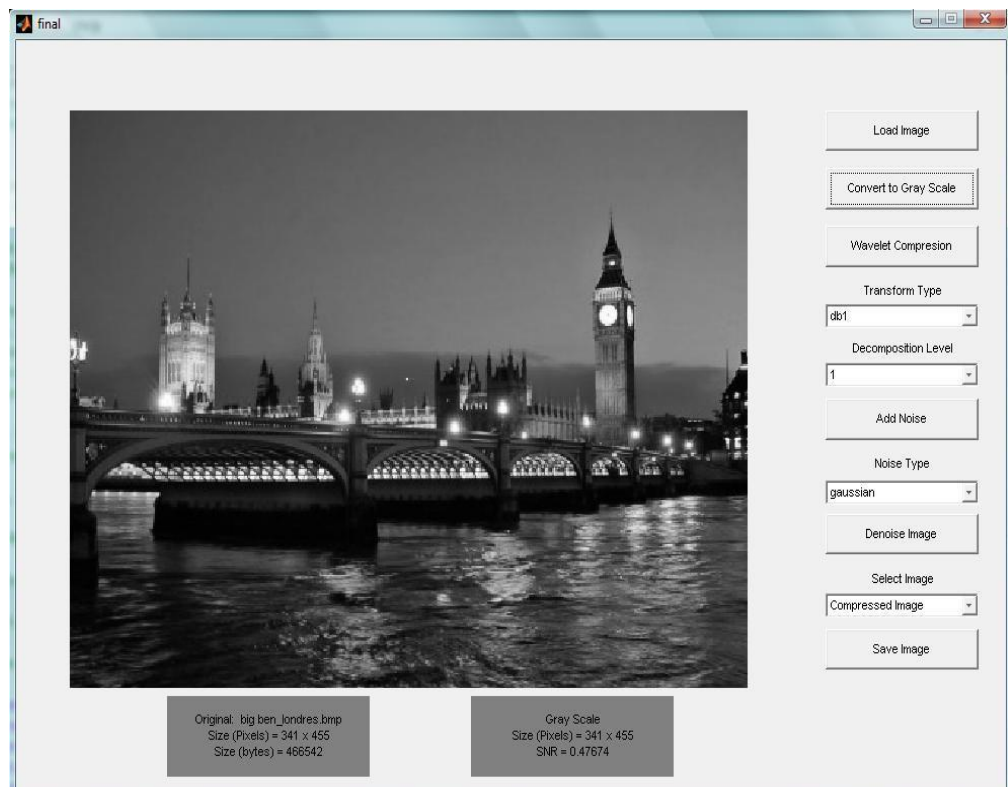


Fig. 3.8 Imagen convertida en escala de grises.

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

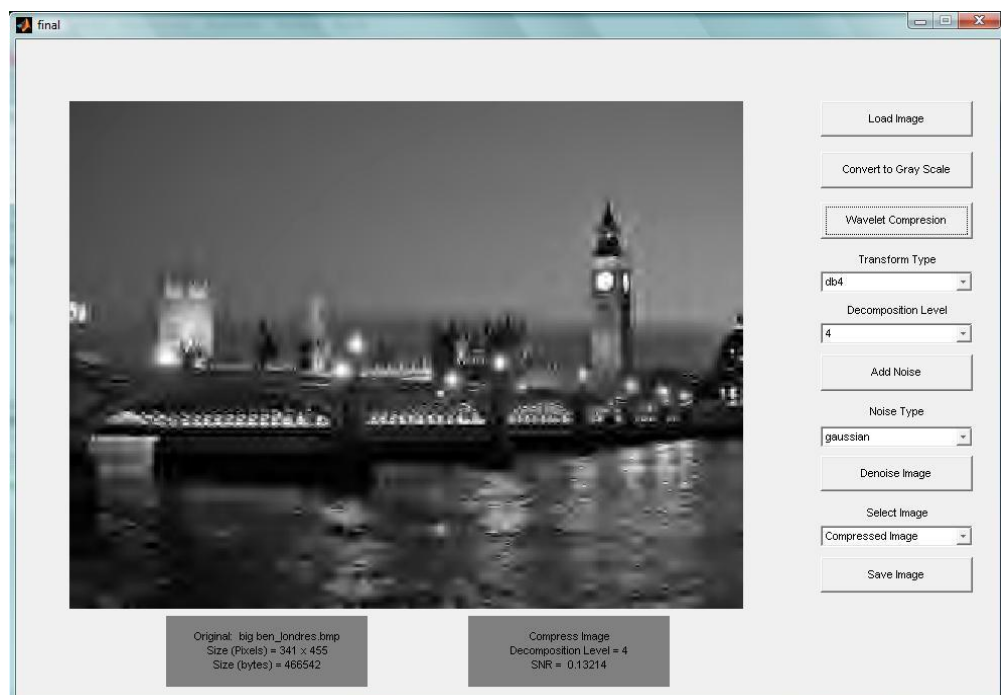


Fig. 3.9 Imagen comprimida con db4 y nivel 4.

3.2.2 Compresión

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.

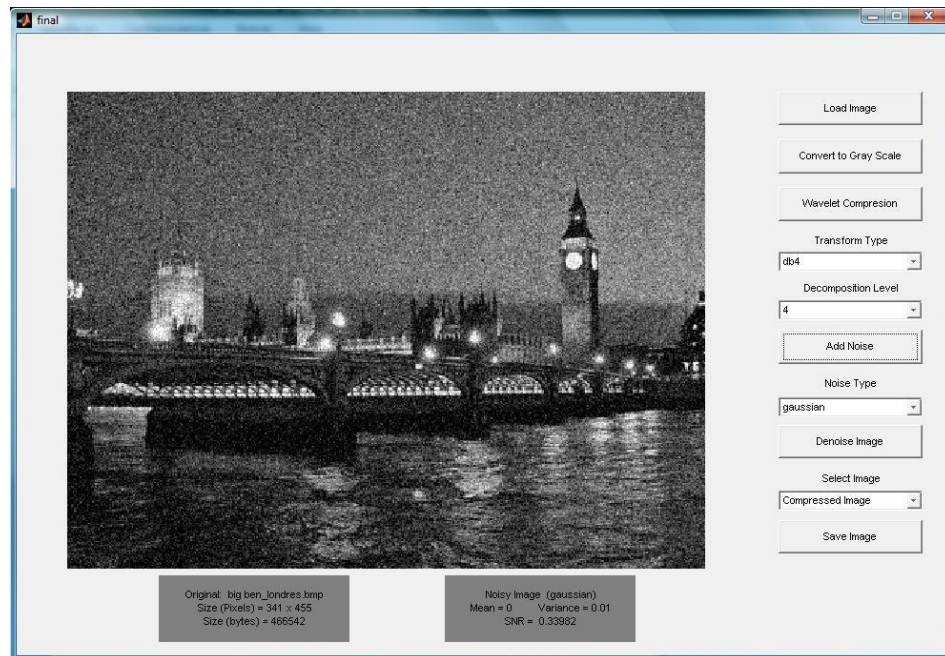


Fig. 3.10 Imagen con ruido gaussiano.

- Se selecciona y se realiza la reducción de ruido.

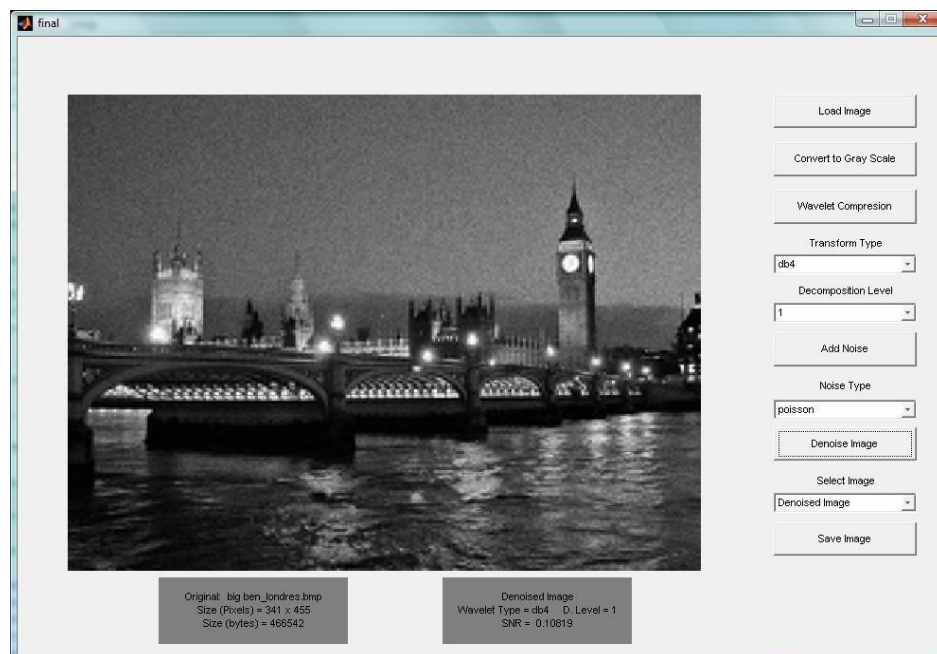


Fig. 3.11 Imagen con reducción de ruido.

3.2.3 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.

3.2.4 Análisis Cualitativo

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

3.2.4.1 Encuesta # 1.

Consta de las 7 imágenes, una por hoja, en donde se pide al encuestado que seleccione la imagen más parecida al original. Cada imagen tiene 5 alternativas además de la original, en cada una se ha usado un método de compresión de nivel 2, usando un tipo de transformada distinto. Las transformadas usadas fueron: Haar, Daubechies, Coiflets, Symlets y DMeyer.

3.2.4.2 Encuesta # 2.

Consta de las 7 imágenes, en donde se pide al encuestado que seleccione la imagen más parecida al original. A estas imágenes se les agrego 4 tipos de ruido distinto, una para cada hoja. Se uso ruido Gaussiano, Poisson, Salt & Pepper y Speckle. Cada imagen tiene 5 alternativas además de la original, en cada una se ha usado un método de reducción de ruido, usando un tipo de transformada distinto. Las transformadas usadas fueron: Haar, Daubechies, Coiflets, Symlets y DMeyer.

3.3 Resultados

3.3.1 Cualitativo – Reducción de Ruido

Tabla 3.1 Porcentaje de aceptación de las Imágenes con ruido Gaussiano

Imagen	Dauchebies	Haar	Coifets	Symlets	DMeyer
Big Ben	16%	24%	16%	30%	14%
El Grito	24%	30%	18%	24%	4%
La Noche Estrellada	16%	26%	22%	30%	6%
Partenón de Grecia	18%	24%	14%	32%	12%
Nacimiento de Venus	20%	30%	18%	22%	10%
La Catedral Notedrame	18%	22%	20%	26%	14%
La Persistencia del tiempo	20%	28%	16%	20%	16%

Tabla 3.2 Porcentaje de aceptación de las Imágenes con ruido Speckle

Imagen	Dauchebies	Haar	Coifets	Symlets	DMeyer
Big Ben	20%	16%	24%	16%	24%
El Grito	24%	12%	28%	14%	22%
La Noche Estrellada	22%	18%	24%	18%	18%
Partenon de Grecia	26%	10%	22%	14%	28%
Nacimiento de Venus	14%	14%	26%	18%	28%
La Catedral Notedrame	26%	18%	22%	16%	18%
La Persistencia del tiempo	24%	12%	24%	12%	28%

Tabla 3.3 Porcentaje de aceptación de las Imágenes con ruido Salt & Pepper

Imagen	Dauchebies	Haar	Coifets	Symlets	DMeyer
Big Ben	24%	20%	8%	24%	24%
El Grito	28%	18%	12%	22%	20%
La Noche Estrellada	26%	22%	6%	18%	28%
Partenon de Grecia	30%	18%	14%	24%	14%
Nacimiento de Venus	28%	16%	10%	20%	26%
La Catedral Notedrame	24%	20%	4%	18%	34%
La Persistencia del tiempo	30%	18%	8%	22%	22%

Tabla 3.4 Porcentaje de aceptación de las Imágenes con ruido Poisson

Imagen	Dauchebies	Haar	Coifets	Symlets	DMeyer
Big Ben	20%	26%	14%	18%	22%
El Grito	16%	20%	16%	24%	24%
La Noche Estrellada	16%	30%	14%	20%	20%
Partenon de Grecia	18%	24%	4%	18%	36%
Nacimiento de Venus	22%	30%	6%	22%	20%
La Catedral Notedrame	14%	32%	12%	24%	18%
La Persistencia del tiempo	18%	22%	10%	22%	28%

3.3.2 Cuantitativo – Compresión de Imágenes

Tabla 3.5 Tamaño de Imagen Big Ben comprimida en Bytes

Wavelet	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Haar	121132	91942	54542	32044
Coiflets	126714	116620	108714	105914
Symlets	121132	91942	54542	32044
DMeyer	128426	120800	116798	117698
Dauchebies	121132	91942	54542	32044

Tabla 3.6 Error Cuadrático Medio Normalizado en compresión de la Imagen Big Ben

Wavelet	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Haar	0.00075931	0.051491	0.10879	0.14296
Coiflets	0.0008481	0.042261	0.10181	0.13825
Symlets	0.00075931	0.051491	0.10879	0.14296
DMeyer	0.00061867	0.01687	0.053641	0.088429
Dauchebies	0.00075931	0.051491	0.10879	0.14296

Tabla 3.7 Tamaño de Imagen La Catedral comprimida en Bytes

Wavelet	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Haar	110912	85090	51456	27352
Coiflets	112414	109760	108594	108790
Symlets	110912	85090	51456	27352
DMeyer	113828	113364	112998	113370

Dauchebies	110912	85090	51456	27352
-------------------	--------	-------	-------	-------

Tabla 3.8 Error Cuadrático Medio Normalizado en compresión de la Imagen La Catedral

Wavelet	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Haar	0.010257	0.14085	0.21809	0.27644
Coiflets	0.0095519	0.12808	0.21868	0.28095
Symlets	0.010257	0.14085	0.21809	0.27644
DMeyer	0.00651	0.0784456	0.16793	0.23091
Daubechies	0.010257	0.14085	0.21809	0.27644

Tabla 3.9 Tamaño de Imagen El Grito comprimida en Bytes

Wavelet	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Haar	831450	613284	331408	172252
Coiflets	815786	807136	783422	750752
Symlets	831450	613284	331408	172252
DMeyer	817572	815708	800166	773998
Daubechies	831450	613284	331408	172252

Tabla 3.10 Error Cuadrático Medio Normalizado en compresión de la Imagen El Grito

Wavelet	Nivel 1	Nivel 2	Nivel 3	Nivel 4
Haar	0.0072018	0.065917	0.13156	0.18214
Coiflets	0.0062107	0.054782	0.11691	0.16816
Symlets	0.0072018	0.065917	0.13156	0.18214
DMeyer	0.0044264	0.042312	0.099121	0.15085

Dauchebies	0.0072018	0.065917	0.13156	0.18214
-------------------	-----------	----------	---------	---------

3.3.3 Cuantitativo – Reducción de imágenes

Tabla 3.11 Error Cuadrático Medio Normalizado en reducción de ruido de la Imagen La Noche Estrellada

Transformada	Gaussian	Speckle	Poisson	Salt&Pepper
Haar	0.32675	0.32407	0.29893	0.30775
Coiflets	0.32201	0.32043	0.29074	0.30670
Symlets	0.36675	0.32407	0.29893	0.30775
DMeyer	0.32181	0.32012	0.29440	0.31456
Dauchebies	0.32675	0.32407	0.29893	0.30775

Tabla 3.12 Error Cuadrático Medio Normalizado en reducción de ruido de la Imagen Nacimiento de Venus

Transformada	Gaussian	Speckle	Poisson	Salt&Pepper
Haar	0.2861	0.27507	0.23696	0.24922
Coiflets	0.27987	0.27453	0.23723	0.26800
Symlets	0.2841	0.27527	0.23696	0.24922
DMeyer	0.2861	0.28032	0.24217	0.27806
Dauchebies	0.2861	0.27507	0.23696	0.24922

**Tabla 3.13 Error Cuadrático Medio Normalizado en
reducción de ruido de la Imagen Partenón de Grecia**

Transformada	Gaussian	Speckle	Poisson	Salt&Pepper
Haar	0.24786	0.28752	0.14276	0.12142
Coiflets	0.22627	0.27144	0.12065	0.13699
Symlets	0.24786	0.28752	0.14276	0.12142
DMeyer	0.23112	0.27224	0.10283	0.14618
Daubechies	0.24786	0.28752	0.14276	0.12142

**Tabla 3.14 Error Cuadrático Medio Normalizado en
reducción de ruido de la Imagen Persistencia del Tiempo**

Transformada	Gaussian	Speckle	Poisson	Salt&Pepper
Haar	0.2126	0.2122	0.11566	0.11724
Coiflets	0.20026	0.20717	0.11204	0.14574
Symlets	0.2126	0.2122	0.11566	0.11724
DMeyer	0.20952	0.21085	0.11022	0.16409
Daubechies	0.2126	0.2122	0.11566	0.11724

3.3.4 Resultados Finales.

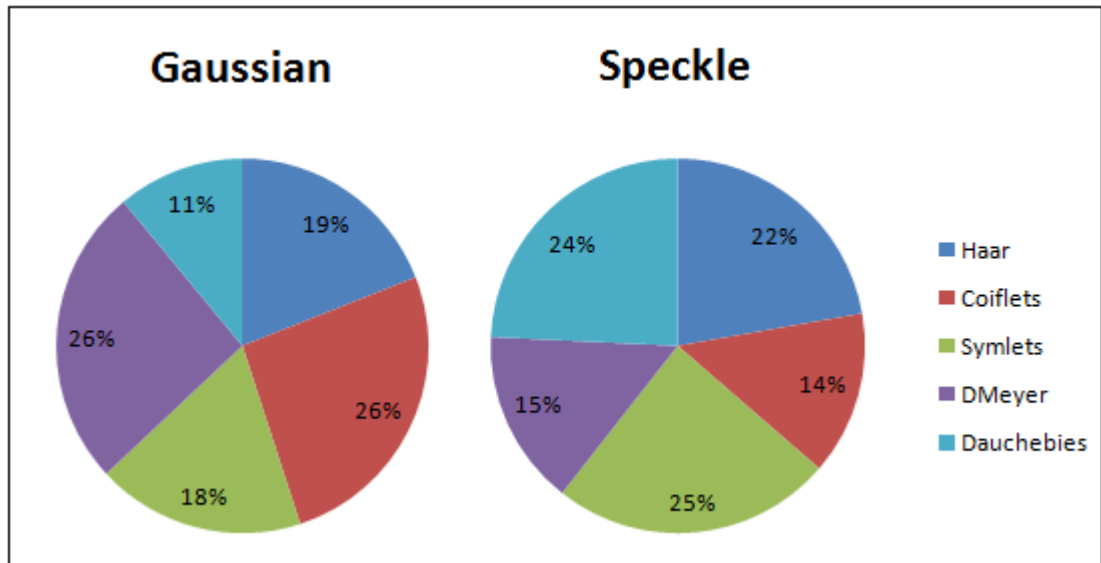


Fig. 3.12 Porcentaje de aceptación de las imágenes en las que se aplico la reducción de Ruido Gaussiano y Speckle

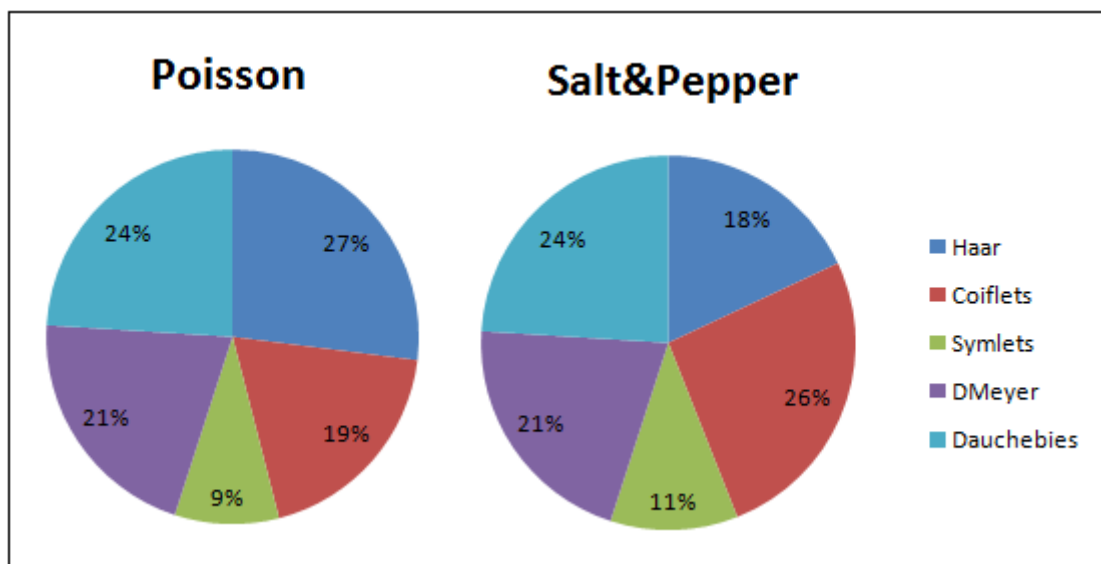


Fig. 3.13 Porcentaje de aceptación de las imágenes en las que

se aplico la reducción de Ruido Poisson y Salt & Pepper

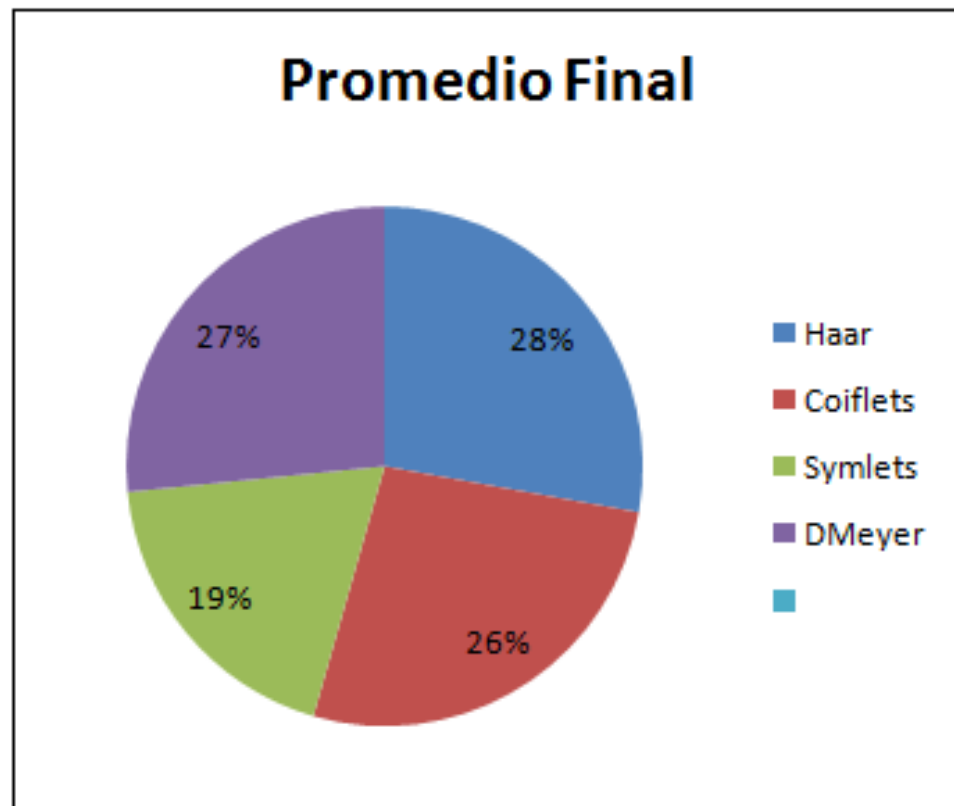


Fig. 3.14 Promedio Final de aceptación de imágenes

las que se aplico la reducción de Ruido.

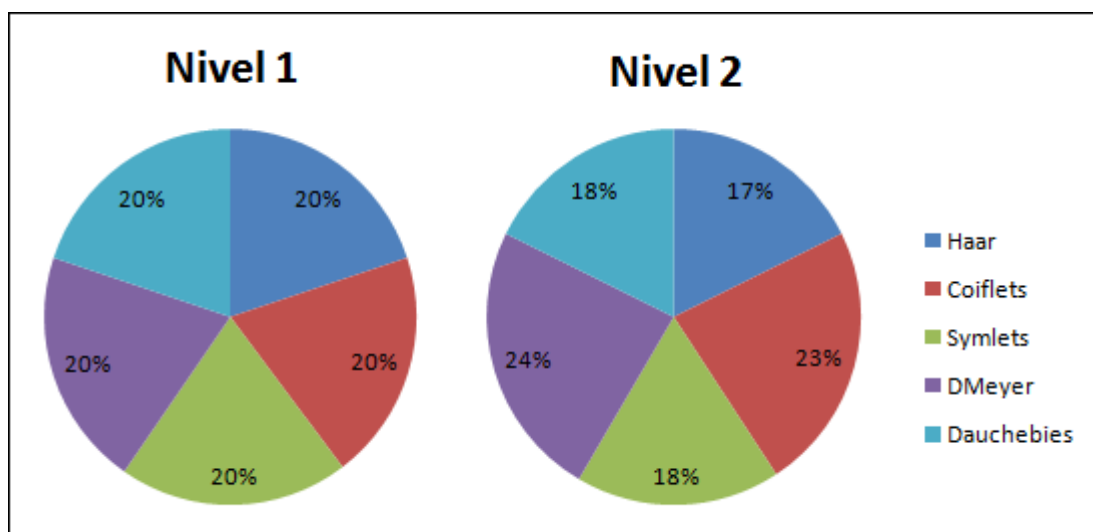


Fig. 3.15 Promedio compresión de imágenes Nivel 1 y 2.

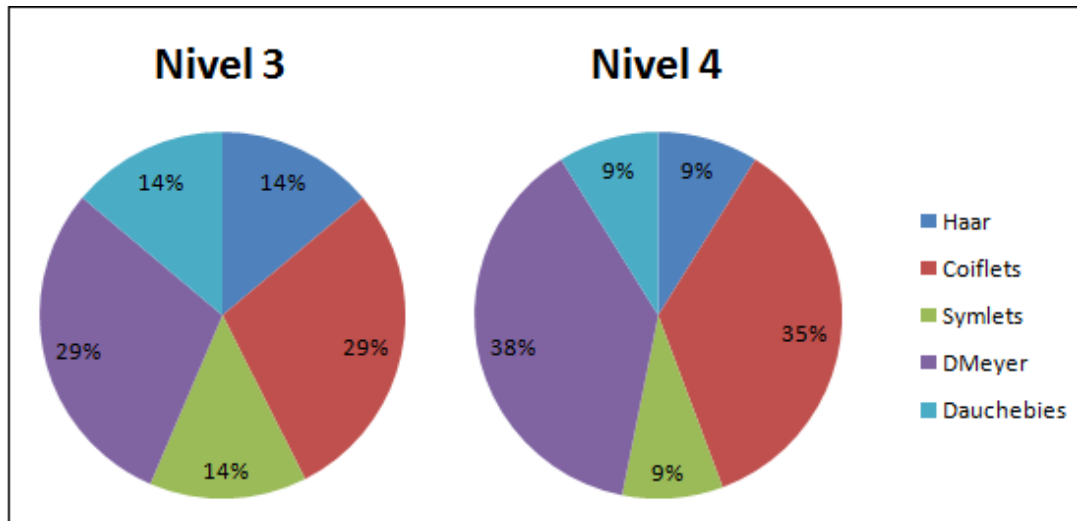


Fig. 3.16 Promedio compresión de imágenes Nivel 3 y 4.

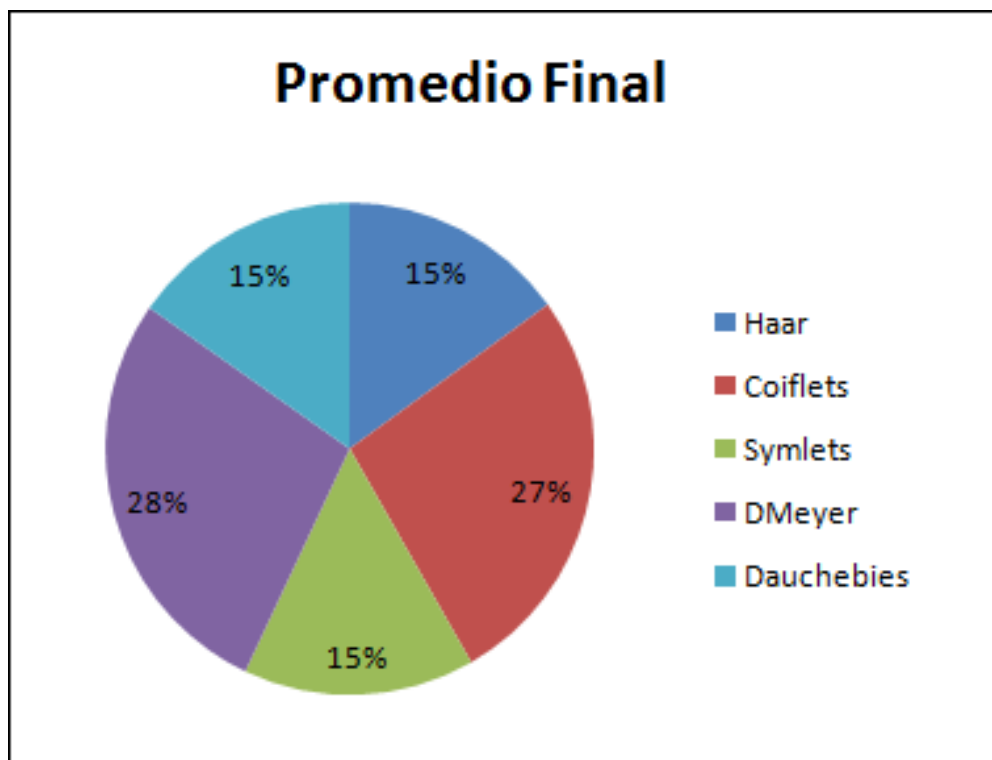


Fig. 3.17 Promedio Final compresión de imágenes.

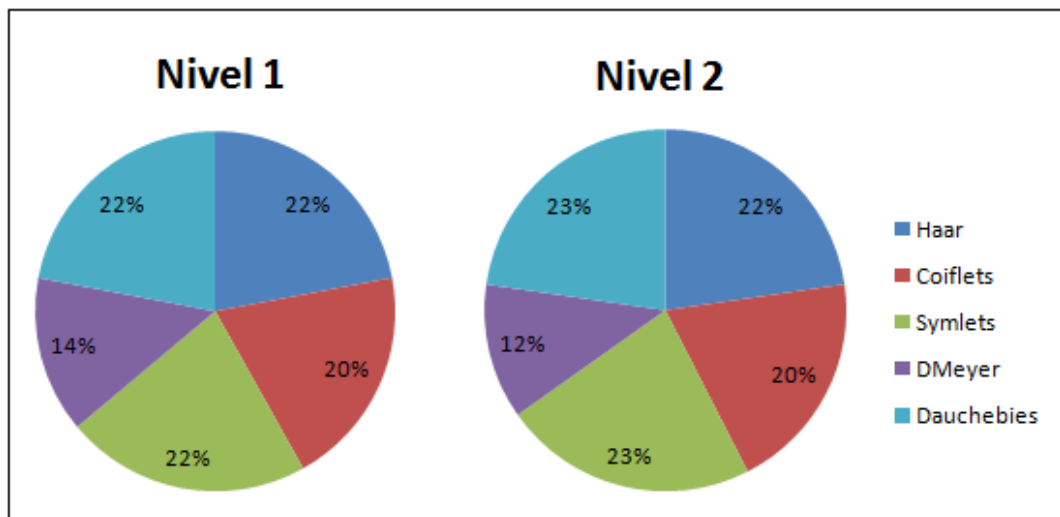


Fig. 3.18 Promedio Error Cuadrático Medio Normalizado en la compresión de imágenes Nivel 1 y 2.

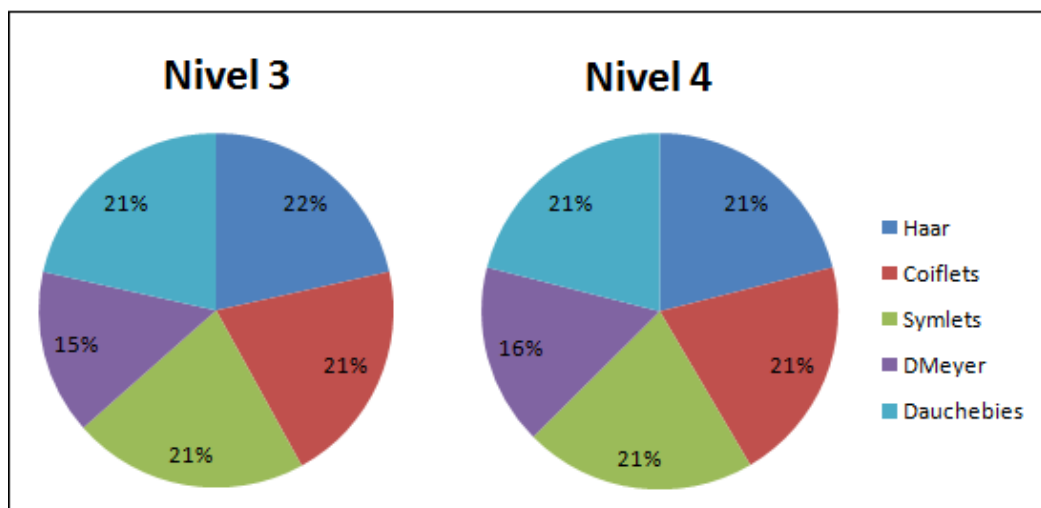


Fig. 3.19 Promedio Error Cuadrático Medio Normalizado en la compresión de imágenes Nivel 3 y 4.

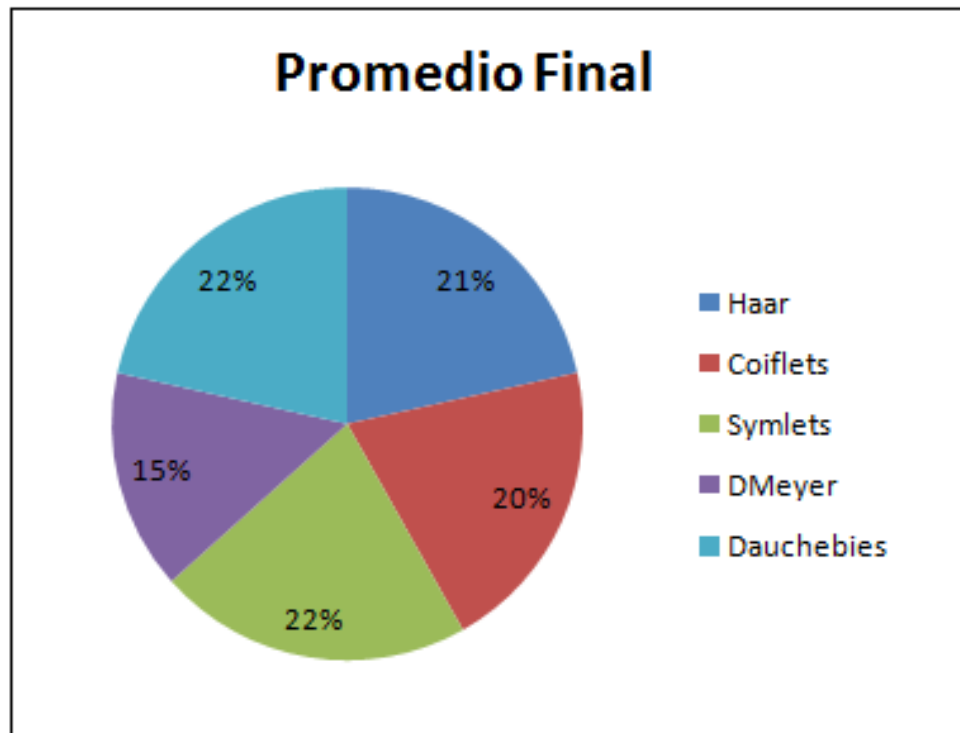


Fig. 3.20 Promedio Final Error Cuadrático Medio Normalizado en la compresión de imágenes.

CONCLUSIONES Y RECOMENDACIONES

1. 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 transformada fue la de Dauchebies mostrando un 24% en tres de los cuatro ruidos, aunque no respondió bien al ruido Gaussiano, solo un 11%. Los porcentajes completos se muestran en la Figura 3.14.
2. 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.

3. 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.

4. 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 presento mejores resultados en promedio es la transformada de Dauchebies.

5. Puedo concluir de igual manera que debido a estos resultados variados, que es muy importante reconocer y analizar la imagen a procesar, realizar las respectivas pruebas con los varios métodos hasta encontrar el que cumpla de mejor manera nuestras expectativas.

6. Por ello considero que la herramienta diseñada en el presente trabajo, 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.

BIBLIOGRAFIA

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 Sprijan; Image & Video Compression Depot; <http://www.mathworks.com/>; 2005

ANEXOS

Anexo 1. Manual de usuario

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

APLICACIÓN GUIDE EN MATLAB:
Comprensión y reducción de ruido en
imágenes en escala de grises, usando
transformada de Wavelet

MANUAL DE USUARIO

Carlos Córdova M.

Índice

1.		I
	Ingreso a la Aplicación	
2.		P
	Herramientas de la Aplicación	
	2.1 Selección de Imagen	
	2.2 Herramientas de compresión	
	2.3 Herramientas para agregar ruido	
	2.4 Herramientas para reducción de ruido	
	2.5 Menú comparación de resultados.	

1. Ingreso a la aplicación

1. Ingrese a MATLAB.
2. Ubique el directorio a aquel directorio donde haya guardado el *.m* y *.fig* del proyecto.
3. Encuentre el archivo *final.m* y ejecútelo.

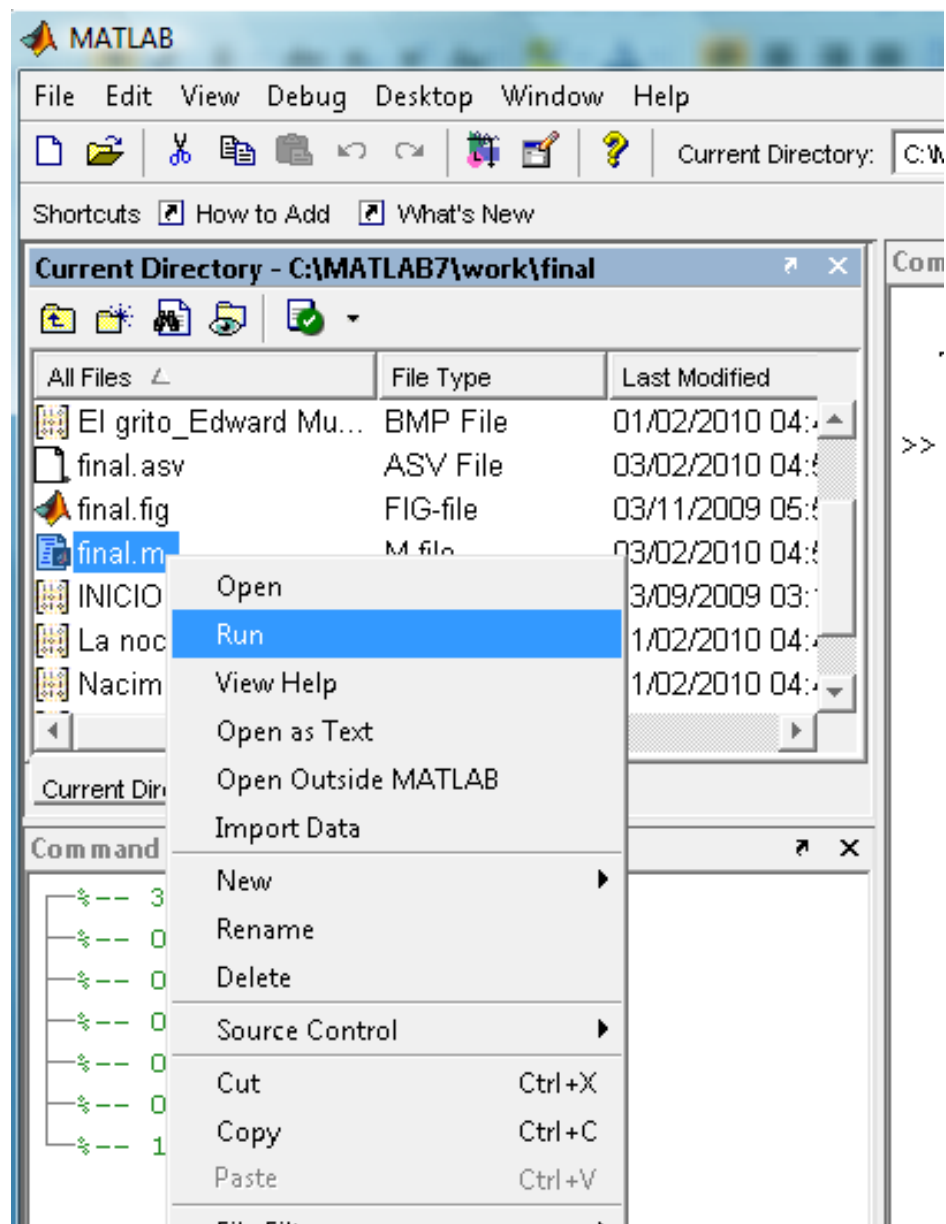
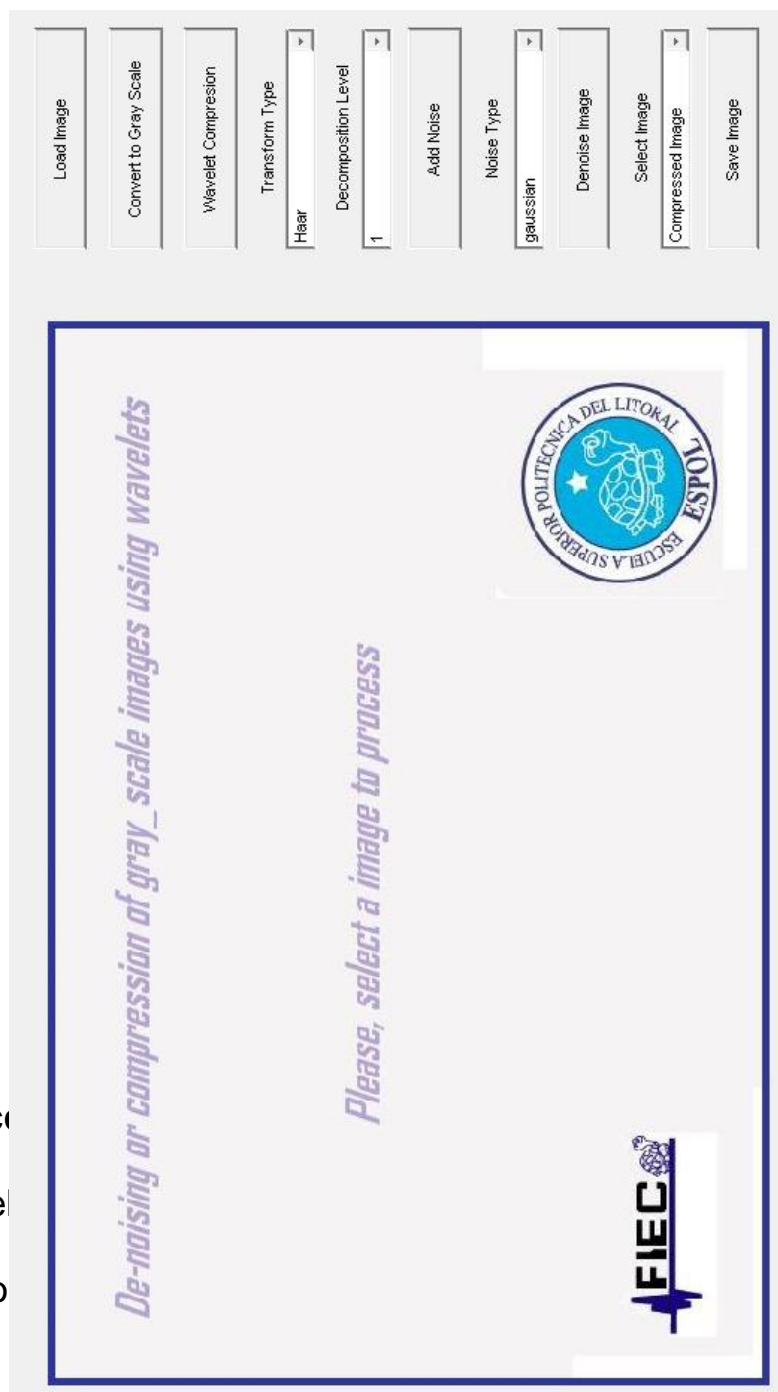


Ilustración 1

2. Partes de la aplicación

Imagen principal de la aplicación.



2.1 Selección

1. Selección

2. Configuración

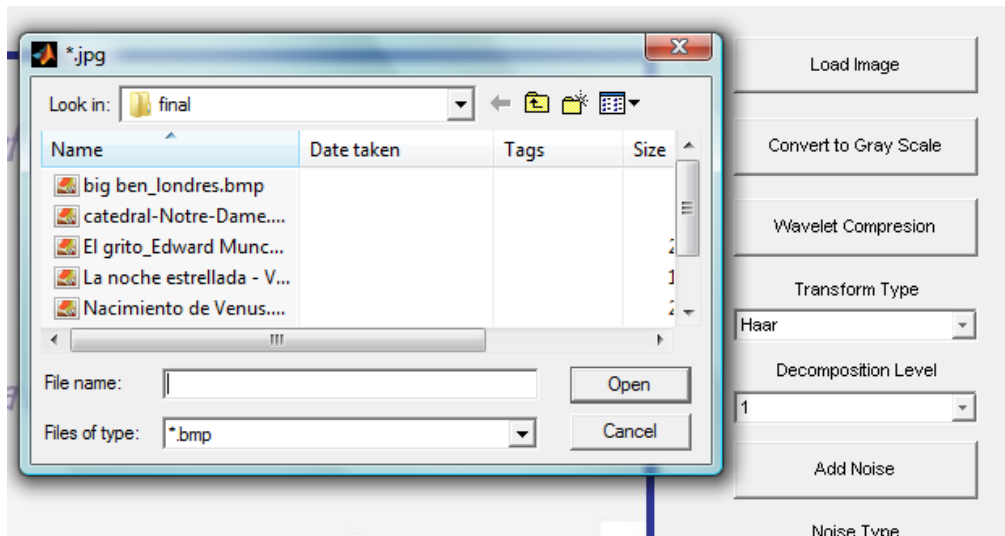


Ilustración 3

2.2 Herramientas de Compresión

1. Seleccione la transformada a utilizar.

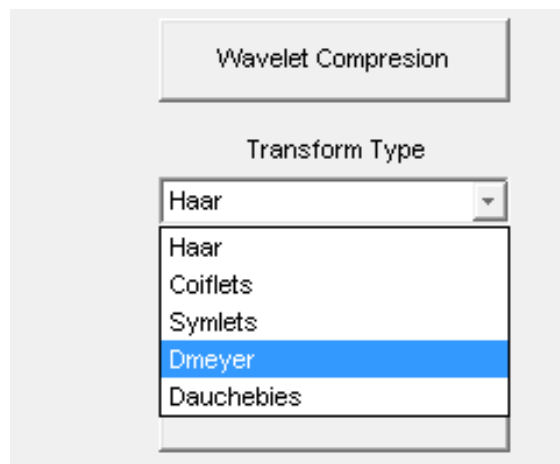
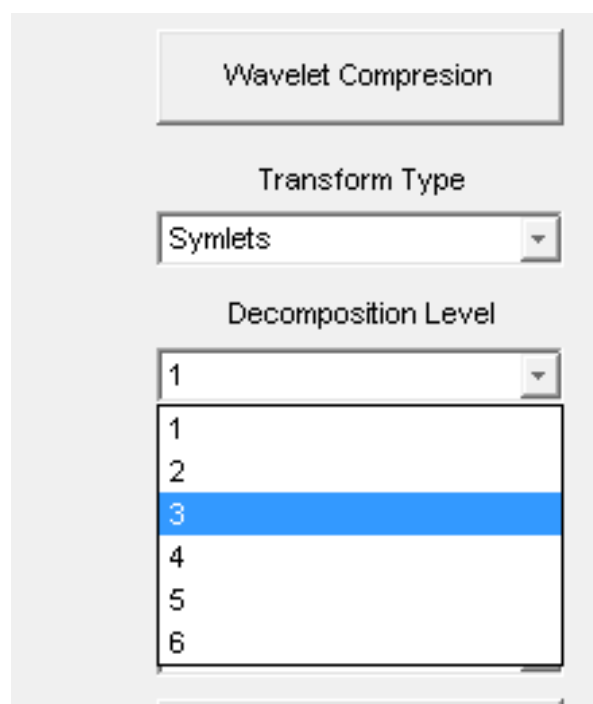


Ilustración 4

2. Seleccione el nivel de

Seleccione el



descomposición Wavelet.

Ilustración 5

2.3 Herramientas para agregar ruido

1. Seleccione el ruido a utilizar.

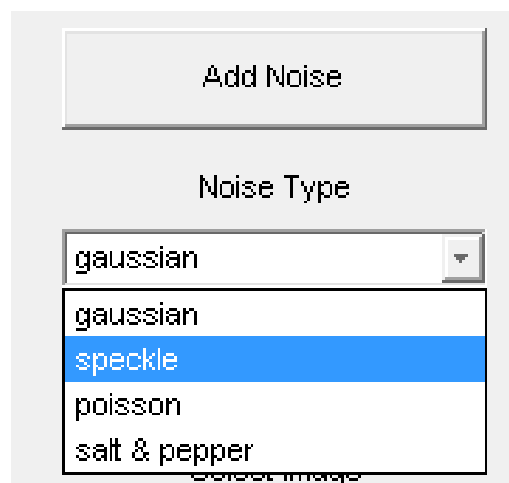


Ilustración 6

2.4 Herramientas para reducción de ruido

1. Seleccione la transformada a utilizar.
2. Realice la reducción de ruido.

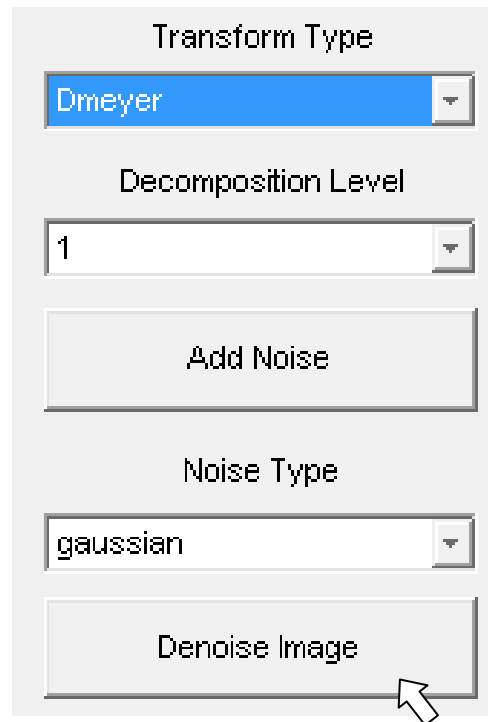


Ilustración 7

2.5 Menú comparación de resultados

1. Seleccione la imagen a mostrar.



Ilustración 8

2. Observe los resultados cuantitativos NSME y el tamaño en bytes, tanto de la imagen original como de la imagen comprimida, ruidosa, o sin ruido.

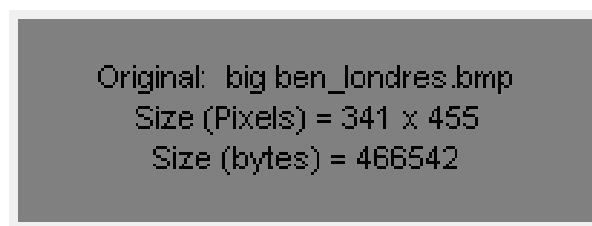


Ilustración 9



Ilustración 10

Anexo 2. Código en Matlab

Conversión a Escala de Grises

```
function Conv_Gray_Scale_Callback(hObject, eventdata, handles)

clc
X = handles.imagen;
x = double(X);
xrgb = 0.2990*x(:,:,1) + 0.5870*x(:,:,2) + 0.1140*x(:,:,3);
x = wcodemat(xrgb,255);
Y = uint8(x);
imshow(Y);
handles.image = Y;
guidata(hObject,handles)

ERROR=nsme(X,Y)
[m,n] = size(Y);
image1 = handles.dir;
info = imfinfo(image1);
Size = info.FileSize;

str1(1) = { ' };
str1(2) = {'Original: ' image1};
str1(3) = {'Size (Pixels) = ' int2str(m) ' x ' int2str(n)};
str1(4) = {'Size (bytes) = ' int2str(Size)};
uicontrol('Style','text','Position',[150 15 200 70],...
    'BackgroundColor', [0.5 0.5 0.5], 'String',str1);

str2(1) = { ' };
str2(2) = {'Gray Scale'};
str2(3) = {'Size (Pixels) = ' int2str(m) ' x ' int2str(n)};
str2(4) = {'SNR = ' num2str(ERROR)};
uicontrol('Style','text','Position',[450 15 200 70],...
    'BackgroundColor', [0.5 0.5 0.5], 'String',str2);
```

Error Cuadrático Medio Normalizado

```
function NormalizedMSE=nsme(original,input)
[ref1 ref2] = size(original);
[in1 in2] = size(input);
m1 = min(ref1,in1);
m2 = min(ref2,in2);
MSE=mean(mean(mean(((original(1:m1,1:m2)input(1:m1,1:m2)).^2))));
NormalizedMSE = MSE / mean(mean(ean(original(1:m1,1:m2).^2)));
```

Compression

```
function Wavelet_Compression_Callback(hObject, eventdata, handles)

wname = get(handles.Transform_Type, 'Value');
level = get(handles.decomposition_Level, 'Value');
switch wname
    case 1, wname='db1';
    case 2, wname='db2';
    case 3, wname='db3';
    case 4, wname='db4';
    case 5, wname='sym4';
end

Y = handles.image;
[C,S] = wavedec2(Y,level,wname);
A1 = wrcoef2('a',C,S,wname,level);
H1 = wrcoef2('h',C,S,wname,level);
V1 = wrcoef2('v',C,S,wname,level);
D1 = wrcoef2('d',C,S,wname,level);

colors = 255;
map = gray(colors);
colors = size(unique(Y));
grayLevels = colors(1);
a1_cod = wcodemat(A1,grayLevels);
d1_hcod = wcodemat(H1,grayLevels);
```

```

d1_vcod = wcodemat(V1,grayLevels);
d1_dcod = wcodemat(D1,grayLevels);
L1 = [a1_cod,d1_hcod;d1_vcod,d1_dcod];
X=uint8(L1);

alpha = 1.5; m = 2.7*prod(S(1,:));
[thr,nkeep] = wdcbm2(C,S,alpha,m)
[f,xcd,sxd,perf0,perf12] = wdencmp('lvd',C,S,wname,level,thr,'h');

colormap(map);
Y=uint8(f);
imshow(Y);
colormap(map);
handles.compressed_image = f;
handles.map = map;
guidata(hObject,handles)

Y = handles.image;
Y = uint8(Y);
X = handles.compressed_image;
X = uint8(X);
COM_ERROR=nsme(Y,X)

[m,n] = size(X);
image1 = handles.dir;
info = imfinfo(image1);
Size = info.FileSize;

str1(1) = {' '};
str1(2) = {'Original: ' image1};
str1(3) = {'Size (Pixels) = ' int2str(m) ' x ' int2str(n)};
str1(4) = {'Size (bytes) = ' int2str(Size)};
uicontrol('Style','text','Position',[150 15 200 70],...
    'BackgroundColor', [0.5 0.5 0.5], 'String',str1);

str2(1) = {' '};
str2(2) = {'Compress Image'};
str2(3) = {'Decomposition Level = ' int2str(level)};
str2(4) = {'SNR = ' num2str(COM_ERROR)};
uicontrol('Style','text','Position',[450 15 200 70],...

```

```
'BackgroundColor', [0.5 0.5 0.5], 'String',str2);
```

Eliminación de Ruido

```
Function Denoise_Image_Callback(hObject, eventdata, handles)
```

```
Y = handles.noise_image;  
wname = get(handles.Transform_Type, 'Value');  
level = get(handles.decomposition_Level, 'Value');  
switch wname  
    case 1, wname='db1';  
    case 2, wname='db2';  
    case 3, wname='db3';  
    case 4, wname='db4';  
    case 5, wname='sym4';  
end  
[thr,sorh,keepapp] = ddencmp('den','wv',Y);  
Y = wdencomp('gbl',Y,wname,level,thr,sorh,keepapp);  
  
Y=uint8(Y);  
imshow(Y)  
handles.denoise_image = Y;  
guidata(hObject,handles)  
  
Y = handles.image;  
Y=uint8(Y);  
X = handles.denoise_image;  
X=uint8(X);  
DE_ERROR=nsme(Y,X)  
  
[m,n] = size(X);  
image1 = handles.dir;  
info = imfinfo(image1);  
Size = info.FileSize;  
  
str1(1) = {' '};  
str1(2) = {'Original: ' image1};  
str1(3) = {'Size (Pixels) = ' int2str(m) ' x ' int2str(n)};
```

```
str1(4) = {'Size (bytes) = ' int2str(Size)};
uicontrol('Style','text','Position',[150 15 200 70],...
    'BackgroundColor', [0.5 0.5 0.5], 'String',str1);
```

```
str2(1) = { ' '};
str2(2) = {'Denoised Image' };
str2(3) = {'Wavelet Type = ' wname ' D. Level = ' int2str(level)};
str2(4) = {'SNR = ' num2str(DE_ERROR)};
uicontrol('Style','text','Position',[450 15 200 70],...
    'BackgroundColor', [0.5 0.5 0.5], 'String',str2);
```