

# **Compresión de Imágenes mediante Onditas**

## **Trabajo Práctico Final Análisis de Señales mediante Onditas**

Autor: Ariel Hernández  
Profesor: Dr. Carlos D'Attellis

**Universidad Tecnológica Nacional  
Regional Buenos Aires**

2014

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Marco del trabajo</b>	<b>3</b>
2.1. Metodología . . . . .	3
2.2. Wavelets . . . . .	3
2.3. Técnicas de compresión de imágenes . . . . .	5
<b>3. Análisis de resultados</b>	<b>6</b>
3.1. Compression Ratio . . . . .	6
3.2. Peak Signal to Noise Ratio . . . . .	8
3.3. Comparación con las técnicas DCT/JPEG . . . . .	8
<b>4. Conclusiones</b>	<b>16</b>

## 1. Introducción

Con el avance de la tecnología y el cambio hacia esta nueva era digital debemos manejar cada vez mayor cantidad de información, en donde en muchas oportunidades nos puede traer ciertas dificultades. Toda esta información debe ser almacenada de una manera eficiente, de tal manera que pueda ser manipulada de forma efectiva y práctica. Esto es un gran desafío para esta generación y las que vienen, ya que el aumento de la información es exponencial. Para llevar a cabo esto lo primero que debemos pensar es en el almacenamiento de los datos y métodos de compresión, de tal forma de eliminar cualquier tipo de información redundante.

*Wavelets* (u *Onditas*) es una herramienta matemática que codifica los datos en diferentes capas (*layers*) de acuerdo al nivel de aproximación. Estas aproximaciones pueden ser almacenadas utilizando menos espacio que la imagen original sin pérdida de información (*lossless*) o bien mucho menos espacio disminuyendo la calidad de la imagen hasta niveles aceptables (*lossy*).

Este trabajo, que se encuentra puntualmente focalizado en la minimización del tamaño de imágenes, es realizado en el marco de la materia “*Análisis de señales mediante onditas*” y busca describir distintas técnicas de compresión utilizando la *Transformada Discreta Wavelet 2D (DWT)*. Se estudian las diferentes técnicas de *thresholding* o *valor umbral* y se los compara con otras herramientas (*DCT*) y algoritmos comerciales (*JPEG*) que son utilizados para el mismo propósito. Para realizar esta comparación se utilizan los parámetros *Compression Ratio (CR)*, *Mean Squares Error (MSE)* y *Peak Signal to Noise Ratio (PSNR)*.

## 2. Marco del trabajo

Como se ha mencionado anteriormente, este trabajo es realizado como requisito de la materia “Análisis de señales mediante ondas” y se utiliza la herramienta *Transformada Discreta Wavelet 2D (DWT)* para la compresión de imágenes.

### 2.1. Metodología

Se realiza una **GUI** (*Graphical User Interface*) que nos permite interactuar con las distintas técnicas de compresión implementadas. Esta interfaz gráfica, que es denominada *icdwt* (Fig. 1) por *Image Compression Discrete Wavelet Transform*, es programada dentro del entorno de MATLAB<sup>1</sup>. Además se utilizaron bases de datos de imágenes no comprimidas de *Sam Houston State University*<sup>2</sup> y de la *University of Southern California*<sup>3</sup>.



Figura 1: La interfaz gráfica *icdwt* realizada en MATLAB

### 2.2. Wavelets

La herramienta matemática que se elije para efectuar la descomposición de una *imagen no comprimida* es la *Transformada Discreta Wavelet*, comúnmente conocida como *DWT*, y se va a utilizar la ondita más simple, la *Haar Wavelet*. Se puede decir que es la primera ondita, e incluso aparece antes de que se desarrollara el concepto de *wavelet* [2].

Wavelets son señales locales en tiempo y en escala, que generalmente tienen una forma irregular. Además exhiben una propiedad ideal para la representación compacta de una señal: *ortogonalidad*. Esta propiedad evita la redundancia de datos y la señal se puede descomponer en muchas pequeñas señales desplazadas y

<sup>1</sup><http://www.mathworks.com/discovery/matlab-gui.html>

<sup>2</sup><http://www.shsu.edu/~qxl005/New/Downloads/index.html>

<sup>3</sup><http://sipi.usc.edu/database/>

escaladas de la *wavelet* original. Estas onditas tienen la gran ventaja de ser capaz de aislar detalles muy finos en una señal de detalles más secundarios.

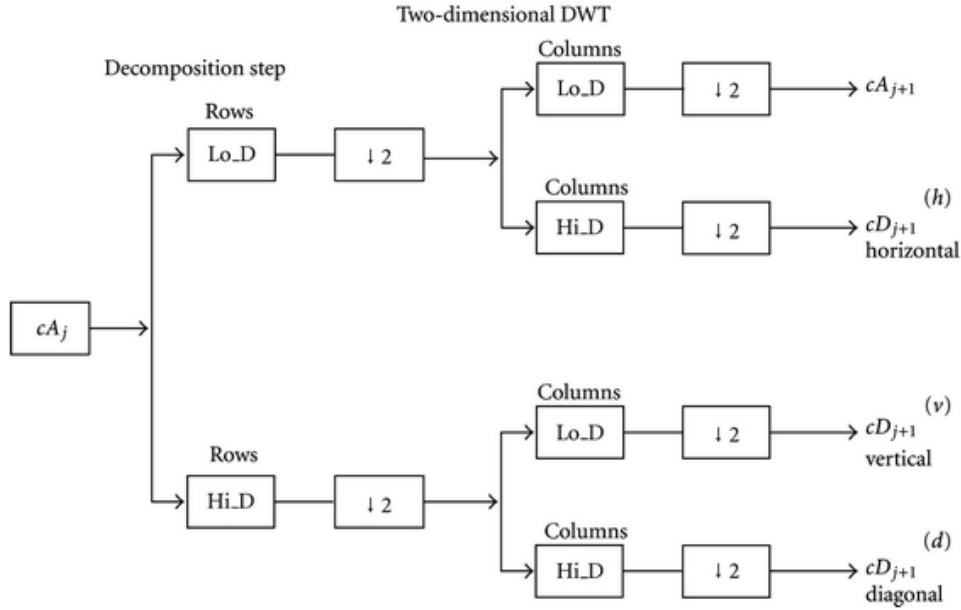


Figura 2: *Descomposición de una señal.*

Dado que una imagen es de dos dimensiones (2-D) se requiere la 2-D DWT, por lo tanto un uso típico es aplicar de forma separada la transformada en *filas* y *columnas* de la matriz. Esto sería como utilizar la 1-D DWT en cada dimensión de la imagen. En palabras sencillas, la DWT aplica dos tipos de filtros a la imagen, uno pasabajos (**L**) y otro pasaltos (**H**) sobre las componentes *horizontales* y *verticales* de manera independiente. Estos filtros están relacionados con las funciones de *escala* y las de *wavelets*[1], y son los que permiten obtener los tres tipos de *imágenes de detalle*: *vertical* (**LH**), *horizontal* (**HL**) y *diagonal* (**HH**)[2]. Luego las operaciones se pueden repetir de forma recursiva sobre la imagen de *aproximación* o *promedio* (**LL**), incrementando el nivel de descomposición de la imagen (Fig. 3). Es claro que esta herramienta matemática en cada nivel de descomposición va separando mayor cantidad de detalle (o componente de alta frecuencia).

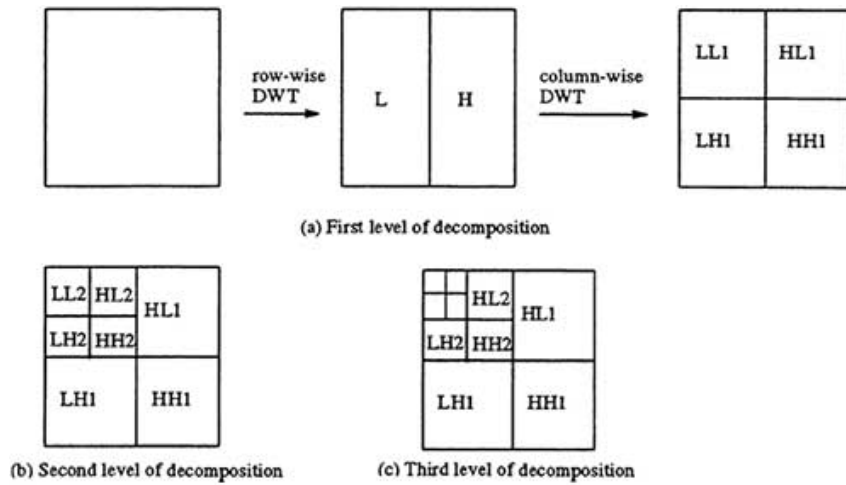


Figura 3: *Descomposición de una imagen.*

Las Figs. 2 y 3 describen el procedimiento de descomposición a una imagen mediante el uso de la

2 –  $D$  DWT. En primera instancia, se aplican los filtros a las *filas* y luego a las *columnas*, quedando perfectamente determinadas las *cuatro* sub-imágenes de la imagen original. Si se requiere continuar con los siguientes niveles de descomposición se debe volver a aplicar ambos filtros en la sub-imagen de *promedio o aproximación LL1* (Fig. 3) que contiene los coeficientes  $cA_{j+1}$  (Fig. 2) y así sucesivamente.

### 2.3. Técnicas de compresión de imágenes

Existen principalmente dos tipos de compresión en general: *lossy* y *lossless*. Cuando se refiere a un tipo de compresión *lossless*, es efectivamente cuando mediante una transformación a un conjunto de datos, en este caso en particular sería una imagen, se logra disminuir su tamaño sin pérdida de información, es decir, la imagen comprimida y la original serían numéricamente idénticas. Por el contrario, cuando se utiliza un tipo de compresión *lossy* siempre existe una degradación relativa a la imagen original. Una ventaja de éste último método es que se logran niveles de mayores de compresión que el primer caso[2].

Este trabajo solo se focaliza en el tipo de compresión *lossy*, en el cual se muestran diversas técnicas basadas en el método *thresholding*, que se aplica sobre los coeficientes de la *Transformada Discreta Wavelet*. Este método se caracteriza por fijar a *cero* o a un valor estipulado a todos los coeficientes de la transformada que superen un cierto *umbral*. Dado el valor umbral  $\epsilon$  se utilizan las siguientes técnicas de *thresholding*:

- **Hard Thresholding**

$$T(\epsilon, x) = \begin{cases} 0 & \text{si } |x| < \epsilon \\ x & \forall \quad x \end{cases} \quad (1)$$

- **Soft Thresholding**

$$T(\epsilon, x) = \begin{cases} 0 & \text{si } |x| < \epsilon \\ \text{Sign}(x)(|x| - \epsilon) & \forall \quad x \end{cases} \quad (2)$$

- **Universal Thresholding** Para este caso en particular se utiliza la función de MATLAB `wdencomp`<sup>4</sup>, que permite realizar la compresión mediante el uso de distintos valores de umbral calculados a partir de la función `wthrmngr`<sup>5</sup> en base a los siguientes métodos:

- **rem\_n0**: Este método calcula el valor del umbral a partir de la `median(abs(coeficientes de la DWT))`.
- **bal\_sn**: La cantidad de información retenida por una imagen luego de la compresión es denominada *retained energy* y es proporcional a la suma de los cuadrados de los valores de los píxeles. Si este valor es del 100 % entonces la compresión es *lossless*[6]. Este método calcula el valor del umbral a partir de igualar los porcentajes de *retained energy* y el número de ceros, esto es:

$$\text{Retained Energy} = 100 * \frac{(\text{norm}(CCD, 2))^2}{(\text{norm}(\text{original signal}, 2))^2} \quad (3)$$

$$\text{Number of Zeros} = 100 * \frac{ZCD}{\text{No. of coeficientes}} \quad (4)$$

En donde  $CCD$  y  $ZCD$  son los *coeficientes* y el *número de ceros* en la actual descomposición respectivamente<sup>6</sup>.

- **sqrthal\_sn**: Es la raíz cuadrada de **bal\_sn**.

Cada una de estas técnicas fueron utilizadas en nuestra **GUI icdwt** y los resultados serán expuestos en la Sección 3.

<sup>4</sup>Ver documentación en <http://www.mathworks.com/help/wavelet/ref/wdencomp.html>

<sup>5</sup>Ver documentación en <http://www.mathworks.com/help/wavelet/ref/wthrmngr.html>

<sup>6</sup>Ver documentación en <http://www.mathworks.com/help/wavelet/ug/data-compression.html>

### 3. Análisis de resultados

En esta sección se van a mostrar diversos resultados a partir de aplicar las técnicas de compresión de imágenes explicadas anteriormente. En la Figs. 4a y 4b se muestran las dos imágenes utilizadas en este informe. Si bien la imagen de *Lena* se encuentra en color, el programa la trabaja en escala de grises.



(a) *Lena*



(b) *Houses*

Figura 4: Imágenes en formato .bmp para el análisis de resultados

#### 3.1. Compression Ratio

Uno de los parámetros más importantes en un proceso de compresión es el *compression ratio* (CR) o razón de compresión, que brinda un valor cuantitativo respecto a la compresión. Este valor se define como la razón entre el *tamaño de la imagen original* sobre el *tamaño de la imagen resultante* como se muestra en la Ecuación (5).

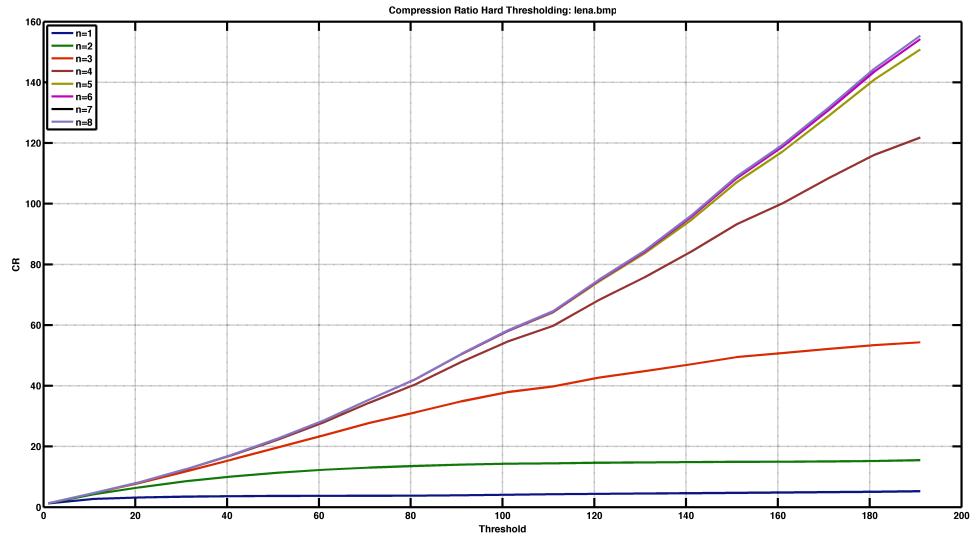
$$CR = \frac{\text{size}(\text{original image})}{\text{size}(\text{compressed image})} \quad (5)$$

En las Figs. 5 y 6 se muestra la variación del valor del CR respecto al nivel de umbral seleccionado. Se puede ver claramente que en la medida que aumentamos el valor de *threshold*, aumenta la razón de compresión. Este comportamiento se manifiesta de esta manera porque al aumentar el valor de umbral estamos eliminando mayor cantidad de información, dando como resultado un menor tamaño de la imagen.

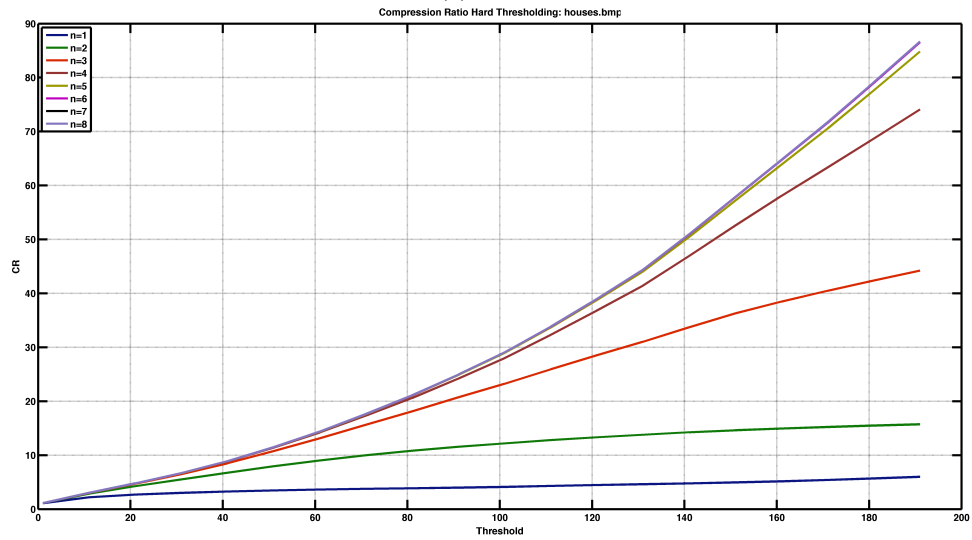
En ambas imágenes de prueba, *lena.bmp* y *houses.bmp*, se observaron comportamientos muy similares respecto a las tendencias de los niveles de CR respecto al valor del umbral, tanto en *hard thresholding* (Fig. 5) como en *soft thresholding* (Fig. 6).

Otro factor que afecta el valor del CR es el *nivel de descomposición* ( $n$ ) de la *DWT* en la imagen. Como se puede observar en las Figs. 5, 6 y 7 cuando se aumenta el valor de  $n$ , el CR normalmente también se incrementa.

La *calidad* de la imagen comprimida depende del *número de descomposiciones*  $n$ , a medida que este número se eleva se tendrá mayor éxito en descartar los coeficientes de la *DWT* menos importantes, que son los que pertenecen a pequeños detalles de la imagen (aquellos que el sistema visual humano es menos sensitivo) y por esta razón el valor de CR se incrementa. Sin embargo, si utilizamos un valor grande de  $n$  podríamos causar gran pérdida de eficiencia en el algoritmo de codificación y una caída significativa en la calidad de la imagen. Por lo tanto, el *nivel de descomposición* debe ser una relación de compromiso entre la calidad de la imagen resultante y la complejidad de cálculo.



(a) *Lena*



(b) *Houses*

Figura 5: *Compression Ratio con Hard Thresholding*



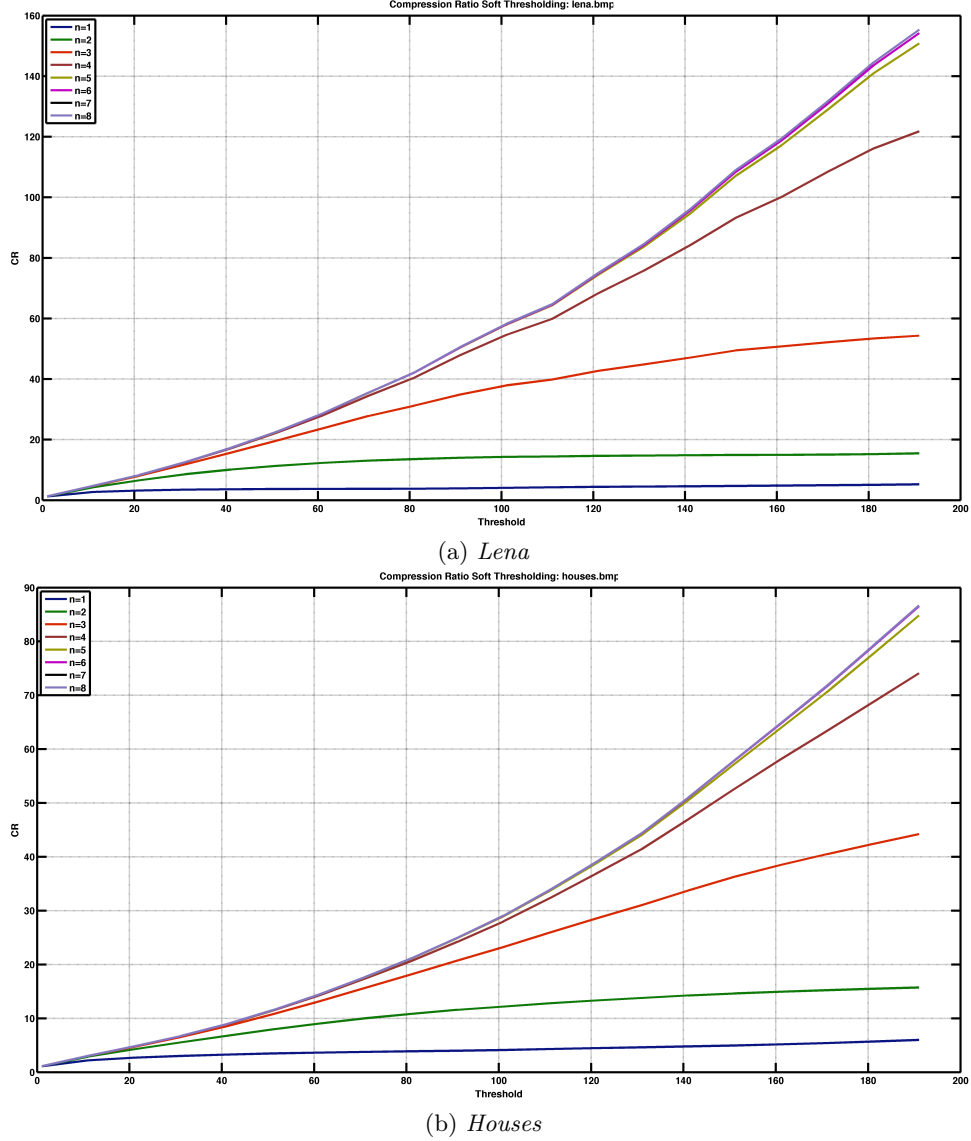


Figura 6: *Compression Ratio con Soft Thresholding*

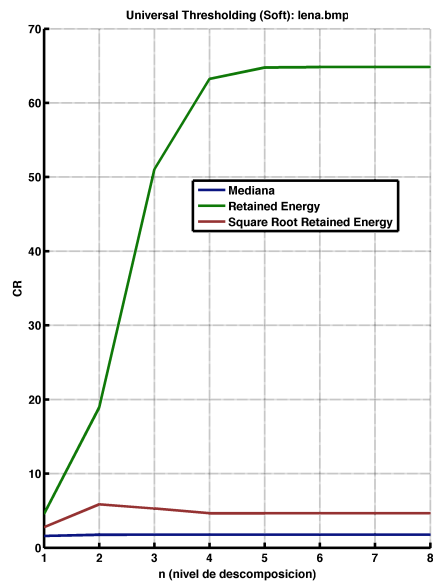
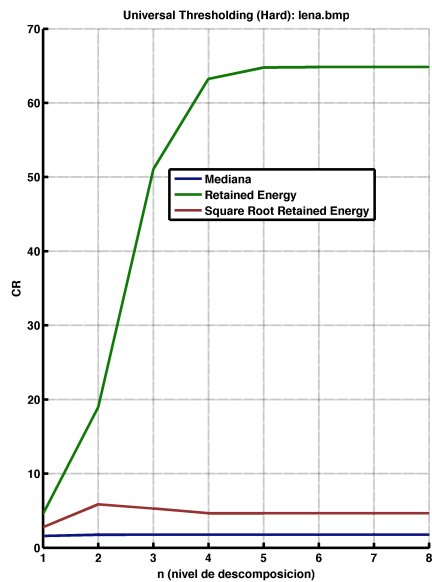
En todas las gráficas se puede concluir que superando un cierto valor en el nivel de descomposición no se obtienen resultados significativos en el CR dentro de ciertos *umbrales*. A raíz de esto, se podría obtener un valor empírico de la cantidad de descomposiciones que se puede utilizar en una imagen cuya resolución es de  $2^p \times 2^p$ , esto es,

$$\text{Límite teórico de } n = p \quad (6)$$

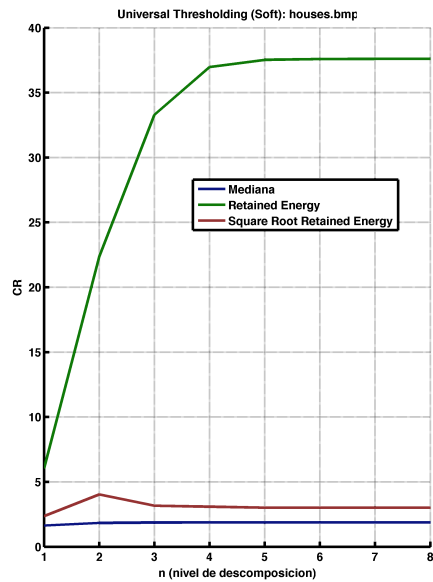
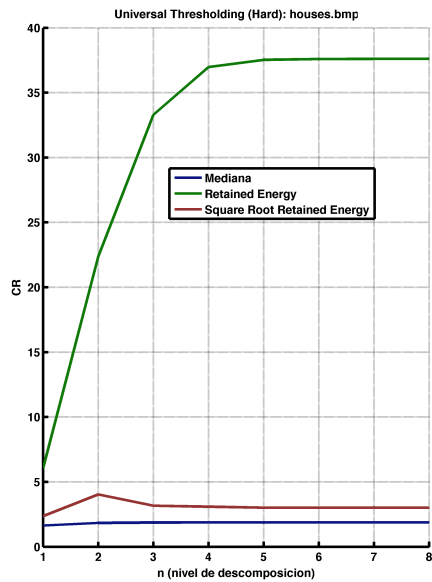
$$\text{Valor a utilizar de } n = \frac{p-1}{2} \quad (7)$$

Las imágenes *lena.bmp* y *houses.bmp* tienen una resolución de  $256 \times 256$  y  $512 \times 512$  respectivamente, lo que nos daría un límite teórico de 8 y 9 respectivamente. En este análisis se puede observar que si se utiliza un valor *mayor* de  $n = 4$  en ambos casos la razón CR no varía significativamente.

En la Fig. 7 se observa claramente que si superamos la cantidad de descomposiciones de  $n = 4$ , el valor de CR se mantiene prácticamente constante. De hecho, en los métodos *Mediana* y *Square Root Retained Energy* la CR es constante para los niveles de descomposición de  $n = 1$  a  $n = 8$ .

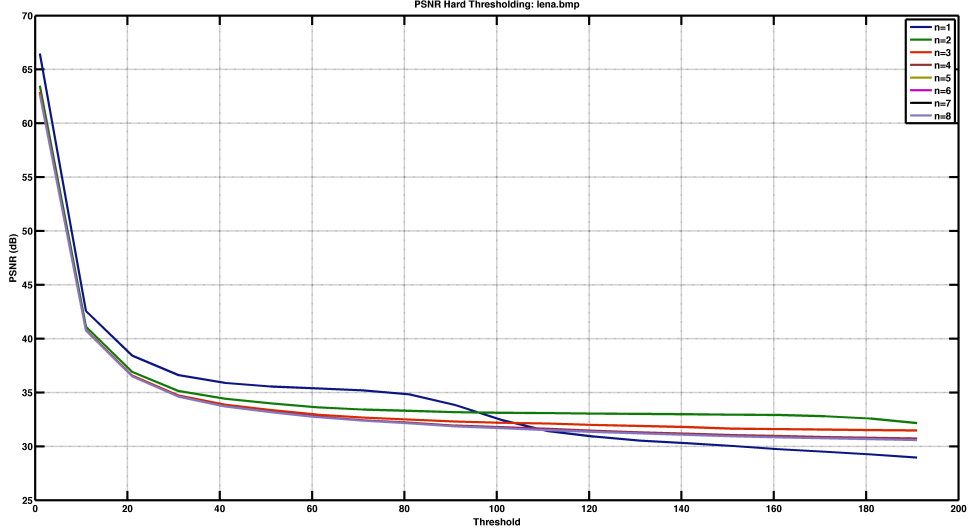


(a) *Lena*

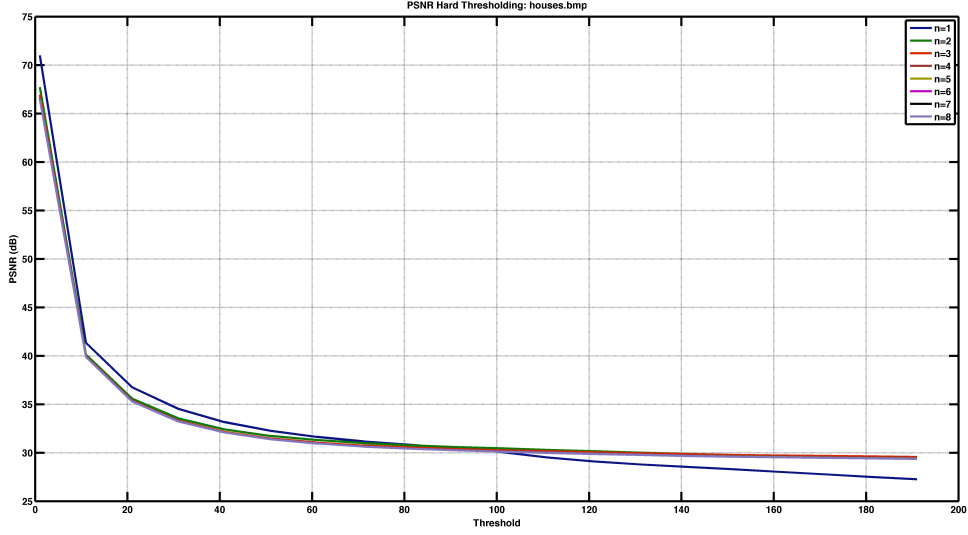


(b) *Houses*

Figura 7: *Compression Ratio con Universal Thresholding*



(a) *Lena*



(b) *Houses*

Figura 8: *Peak Signal to Noise Ratio con Hard Thresholding*

### 3.2. Peak Signal to Noise Ratio

En esta sección se va a analizar el comportamiento del *Peak Signal to Noise Ratio* (PSNR) o relación señal a ruido pico en función del valor de *threshold* y el *nivel de descomposición*  $n$ . En la Ecuación (8) se define el PSNR para una imagen escala de grises (8 *bits/pixel*) [2].

$$PSNR(dB) = 20 * \log_{10}\left(\frac{255}{\sqrt{MSE}}\right) \quad (8)$$

En donde MSE es el *Mean Square Error* definido por,

$$MSE = \frac{1}{r * c} \sum_{y=1}^r \sum_{x=1}^c (I(x, y) - I'(x, y))^2 \quad (9)$$

En la Ecuación (9)  $I(x, y)$  e  $I'(x, y)$  son la imagen original y la comprimida respectivamente de dimensiones  $rx c$ .

En las Figs. 8, 9 y 10 se muestran los valores de PSNR en *dB* de cada una de las técnicas de compresión. Se puede ver que el comportamiento es similar al que se observó con el de *compression ratio*. Cuando se

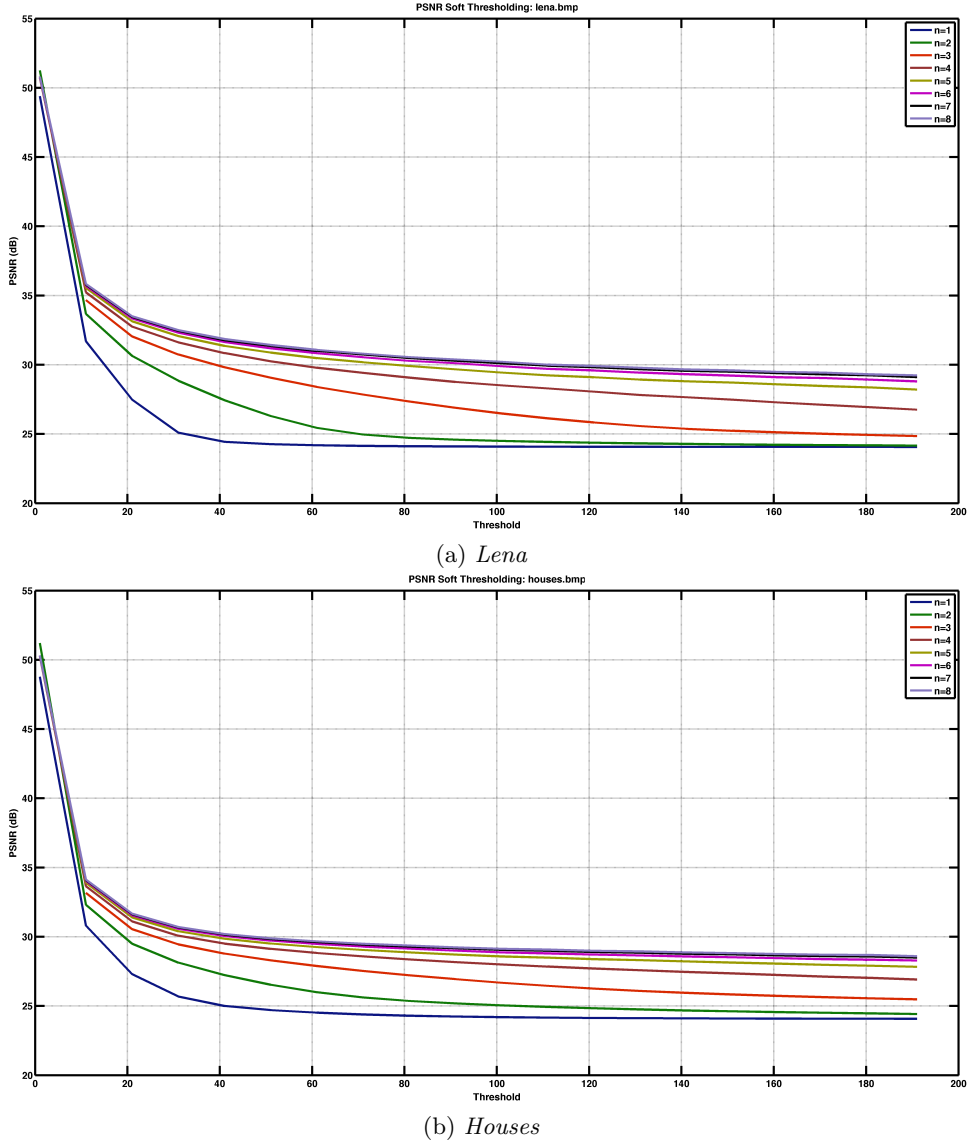


Figura 9: *Peak Signal to Noise Ratio con Soft Thresholding*

aumentan los niveles de *threshold* el valor de PSNR disminuye. También se observa, como en el caso anterior del CR, que luego de un nivel de descomposición  $n$  el PSNR se mantiene prácticamente constante.

### 3.3. Comparación con las técnicas DCT/JPEG

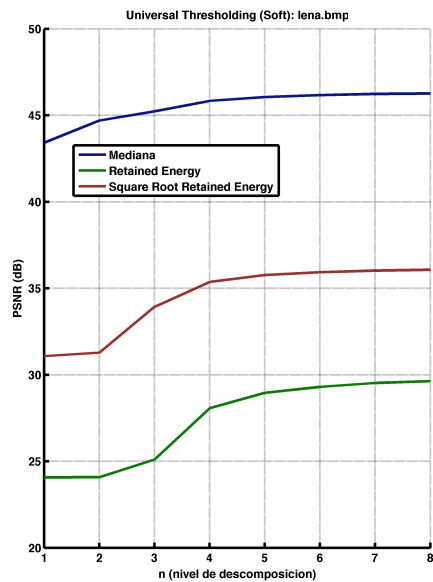
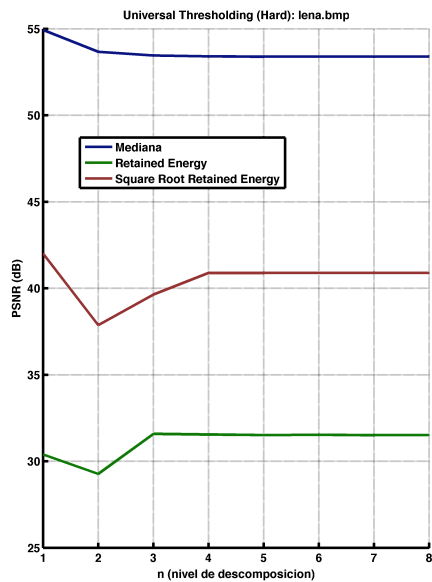
En esta sección se va a realizar una comparación entre las diferentes técnicas de compresión de imágenes expuestas en este trabajo con otras conocidas y ampliamente utilizadas, con es el caso del *JPEG*.

En la Fig. 11 se muestra el valor de PSNR en función de la razón de compresión (CR) para diferentes métodos de compresión basados en las siguientes herramientas:

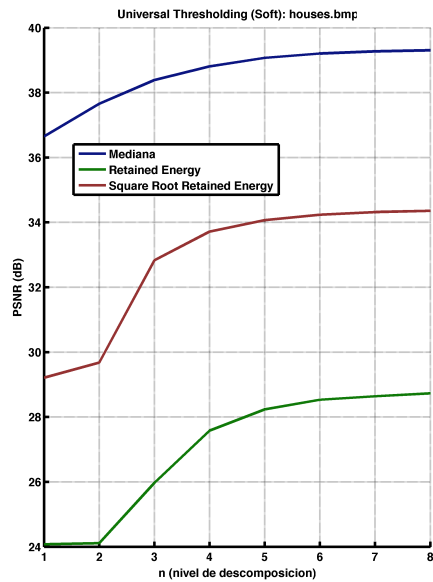
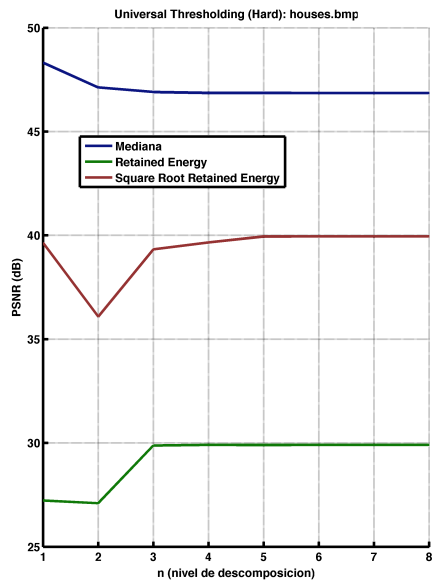
- *Joint Photographic Experts Group JPEG*<sup>7</sup>
- *Discrete Cosine Transform DCT*<sup>8</sup>

<sup>7</sup>Se utilizó directamente el algoritmo comercial que tiene MATLAB para generar una imagen *.jpg*. Ver documentación en <http://www.mathworks.com/help/matlab/ref/imwrite.html>.

<sup>8</sup>Programa para comprimir una imagen basado en la herramienta *Discrete Cosine Transform* obtenido a través de la documentación de MATLAB. Para más información ver en <http://www.mathworks.com/help/images/discrete-cosine-transform.html>.



(a) *Lena*



(b) *Houses*

Figura 10: *Peak Signal to Noise Ratio con Universal Thresholding*

■ *Discrete Wavelet Transform DWT*

Se puede observar que cuando se obtienen niveles de CR más altos, los valores de PSNR disminuyen. Si tenemos en cuenta los gráficos de la sección anterior este comportamiento es esperado, ya que al elevarse el valor del *compression ratio* existe una pérdida de información y esto impacta directamente en el PSNR.

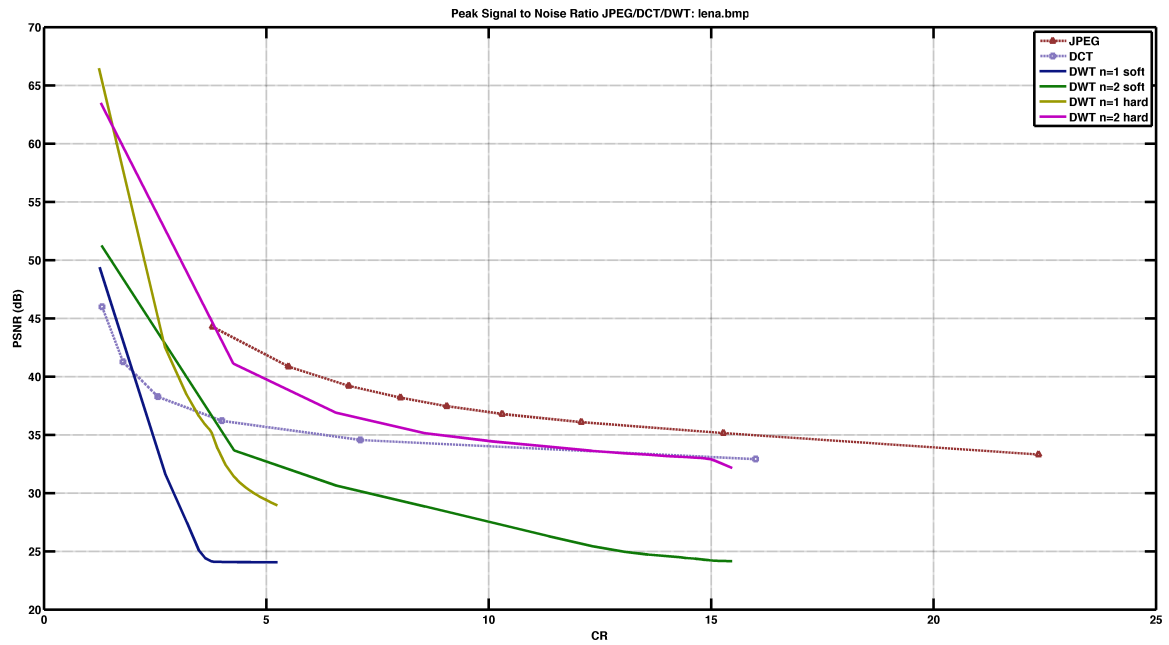
Si bien la *DCT* es la base del *JPEG*, éste último es un algoritmo comercial y por lo tanto tiene ciertas optimizaciones que mejoran su *performance* respecto al *thresholding* con *WDT* o eliminando coeficientes en la *DCT* de la imagen. De todas maneras se puede ver que el resultado obtenido con las diferentes técnicas de *umbral* con la *WDT* es aceptable si tomamos como referencia a *JPEG*. En los Cuadros 1 y 2 se muestran los parámetros CR, MSE y PSNR para los diferentes métodos de compresión utilizados en este análisis de comparació. Estos valores pertenecen a la imagen *lena.bmp*.

Método	Porcentaje	CR	MSE	PSNR
DCT	10	15,9992675781250	33,1658172607422	32,9238965736536
JPG	10	22,3596042306380	30,2598724365234	33,3221326798810
DCT	20	7,11078559027778	22,7182922363281	34,5670467908486
JPG	20	15,2728967606619	19,7931976318359	35,1656439982570
DCT	30	3,99981689453125	15,5617675781250	36,2102143622017
JPG	30	12,0803686635945	15,9608764648438	36,1002362464306
DCT	40	2,56018283392585	9,66026306152344	38,2809140788798
JPG	40	10,3011631562402	13,6004943847656	36,7952566538937
DCT	50	2,56018283392585	9,66026306152344	38,2809140788798
JPG	50	9,05443492677535	11,6411285400391	37,4708527617657
DCT	60	1,77784107864681	4,85775756835938	41,2664452379339
JPG	60	8,01663608562691	9,84425354003906	38,1989757038185
DCT	70	1,77784107864681	4,85775756835938	41,2664452379339
JPG	70	6,85236302802175	7,79959106445313	39,2100852780655
DCT	80	1,30614075299464	1,63732910156250	45,9894437991197
JPG	80	5,49752537538797	5,32379150390625	40,8685932204350
DCT	90	1,30614075299464	1,63732910156250	45,9894437991197
JPG	90	3,79149551634365	2,42730712890625	44,2795562946789

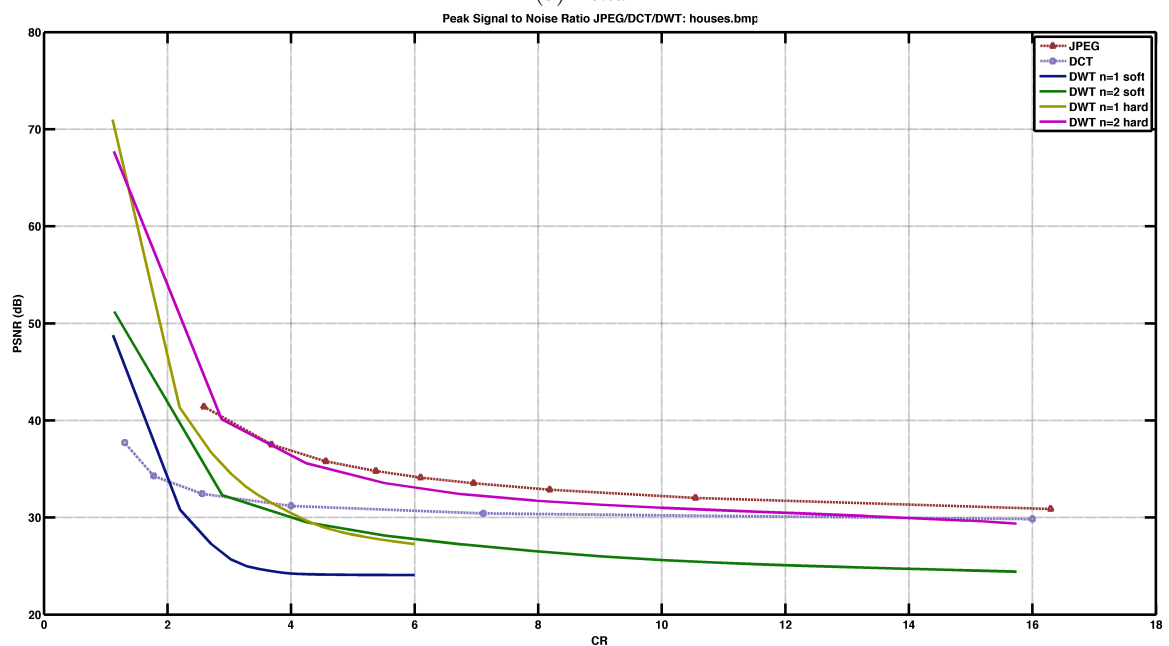
Cuadro 1: *Parámetros de compresión en los métodos de JPEG y DCT para lena.bmp*

Método	<i>n</i>	Threshold	Tipo TRH	CR	MSE	PSNR
DWT	1	50	SOFT	3,69816601167084	243,424942016602	24,2671528563775
DWT	2	50	SOFT	11,3468155197657	152,373992919922	26,3016951259925
DWT	1	50	HARD	3,69750550693576	18,0854187011719	35,5575179316681
DWT	2	50	HARD	11,3405889884763	25,9331359863281	33,9922532349666
DWT	1	-	UNI-ENERGY	4,56568404028523	59,4000549316406	30,3929351426218
DWT	2	-	UNI-ENERGY	18,9541424640783	76,9305877685547	29,2698131030115

Cuadro 2: *Parámetros de compresión en los métodos de WDT para lena.bmp*



(a) *Lena*



(b) *Houses*

Figura 11: Comparación entre *JPEG*, *DCT* y *WDT*

## 4. Conclusiones

Es importante aclarar que las técnicas de compresión expuestas en este trabajo utilizan los coeficientes de la *Transformada Discreta Wavelet* para disminuir el tamaño una imagen. La *DWT* no realiza la compresión, solo nos permite expresar nuestra imagen de tal manera que usando ciertos métodos sobre sus coeficientes se pueda lograr comprimir su tamaño.

Un punto de estudio que luego de leer varios *papers* no se suele trabajar es el *nivel de descomposición* de una imagen y su efecto tanto en la PSNR y en la CR. Creo que es necesario tenerlo muy en cuenta en el costo computacional y en la calidad de la imagen, ya que juega un rol importante en estas cuestiones.

Luego de realizar este estudio acerca de compresión de imágenes, es claro que la *DWT* es una herramienta muy poderosa para este tipo de aplicaciones y estimo que por esta razón el grupo *JPEG* realizó el estándar de compresión *JPEG* 2000 basado en *wavelets*.



## Referencias

- [1] C. D' Attellis M. Anaya M. Cavallaro F. Villaverde, *Introducción a las Onditas*, Editorial Nueva Librería, Segunda Edición, 2012.
- [2] P. Porwik A. Lisowska , *The Haar-Wavelet Transform in Digital Image Processing: Its Status and Architecture*, Machine Graphics & Vision, vol. 13, nro 1/2, pp. 79-98, 2004.
- [3] K. Hasan K. Harada, *Haar Wavelet Based Approach for Image Compression and Quality Assesment of Compressed Image*, IAENG International Journal of Applied Mathematics, 36:1, IJAM\_36\_1\_9, 2007.
- [4] S. Mallat, *A wavelet tour of signal processing*, Academic Press, Second Edition, 1999.
- [5] E. Stollnitz T. DeRose D. Salesin, *Wavelets for Computer Graphics: A Primer*, IEEE Computer Graphics and Applications, 15(3) and 15(4), 1995.
- [6] K. Lees, *Image Compression Using Wavelets*, Portland State University, 2002.