

A Lidita, por el ejemplo, el sacrificio  
y la coherencia de toda su vida.  
A Juan, por el apoyo incondicional y  
los consejos.  
A Diego, porque es mi orgullo y siem-  
pre voy a estar con él.  
A Alberto, por estar en las buenas y  
en las malas.  
A Moni, por su paciencia en todos los  
fines de semana.

A los que luchan cada día por la  
*Educación Pública y Gratuita.*  
A los que no pierden de vista la  
utopía en tiempos tan grises  
y desmovilizados.

# Índice general

<b>1. Introducción</b>	<b>7</b>
1.1. Compresión	7
1.2. Objetivos	8
1.3. Contenido del trabajo	9
<b>2. Compresión con pérdida de información</b>	<b>10</b>
2.1. Medidas de Error	10
2.1.1. Norma 2	10
2.1.2. Error cuadrático medio - MSE	11
2.1.3. PSNR	11
2.2. Diversas técnicas de compresión con pérdida de información	12
2.2.1. Submuestreo	12
2.2.2. Umbral	13
2.2.3. Cuantización	14
<b>3. Wavelets</b>	<b>22</b>
3.1. Espacios V y W	22
3.2. Función escala y función wavelet	23
3.3. Descomposición de una función	24
3.4. El algoritmo cascada	26
3.5. Propiedades de la transformada ortogonal	27
3.5.1. Forma matricial	29
3.6. Multiwavelets	30
3.7. Procesamiento de señales	31
3.7.1. Señales unidimensionales	31
3.7.2. Señales bidimensionales	32
<b>4. Estudio de las bandas</b>	<b>34</b>
4.1. Umbral sobre transformadas	34
4.2. Cuantización sobre transformadas	37
4.2.1. Cuantización por banda vs. cuantización para toda la señal	40
4.2.2. Cuantización uniforme vs. Max-Lloyd	42
4.3. Influencia de cada banda	46
<b>5. Distribución de bits por banda</b>	<b>50</b>
5.1. Medición de la calidad final	50
5.2. Distribución de error	51
5.3. Descarte de las bandas más finas	52

<i>ÍNDICE GENERAL</i>	3
5.4. Error por coeficiente . . . . .	53
5.5. Método de la derivada . . . . .	55
5.5.1. Desarrollo teórico . . . . .	59
<b>A. Resultados comparativos entre métodos</b>	<b>63</b>

# Índice de cuadros

4.1. Comparación del umbral aplicado en una imagen y su transformada wavelet . . . . .	38
4.2. Comparación del desempeño del umbral y la cuantización uniforme en una transformada wavelet. . . . .	39
4.3. Cantidad de coeficientes por banda . . . . .	48
5.1. PSNR, Bits por pixel y valor de la derivada para dos ejemplos. .	57
5.2. Bits por pixel, Paso de cuantización y Cantidad de clases por Banda de detalle según PSNR. . . . .	58

# Índice de figuras

2.1. Aplicación de distintos umbrales a una señal. . . . .	13
2.2. Distribución de una señal original e intervalos definidos por la cuantización uniforme. . . . .	16
2.3. Distribución de una señal original e intervalos definidos por la cuantización Max-Lloyd. . . . .	18
2.4. Comparación de la entropía resultante entre cuantización uniforme y Max-Lloyd . . . . .	19
2.5. Comparación entre el desempeño de la cuantización uniforme y Max-Lloyd . . . . .	20
2.6. Comparación entre cuantización uniforme y Max-Lloyd para una imagen . . . . .	20
3.1. Algoritmo cascada . . . . .	27
3.2. Aplicación de la transformada wavelet a una señal unidimensional .	32
3.3. Aplicación de la transformada wavelet a una señal bidimensional	33
4.1. Ubicación de las bandas de detalle en señales bidimensionales . .	35
4.2. Mapas de significancia luego de la aplicación de umbrales a una transformada . . . . .	36
4.3. Comparación del umbral aplicado en una imagen y su transformada wavelet . . . . .	38
4.4. Comparación del desempeño del umbral y la cuantización uniforme en una transformada wavelet. . . . .	39
4.5. Comparación de los histogramas correspondientes a la primer banda y a la imagen transformada total. . . . .	40
4.6. Comparación del tamaño estimado de la imagen codificada según si se codifica la imagen como un todo o se codifica por banda . .	41
4.7. Histogramas correspondientes a las distintas Bandas de Detalle con idéntica escala para todos. . . . .	43
4.8. Histogramas correspondientes a las distintas Bandas de Detalle. .	44
4.9. Comparación de la entropía resultante entre cuantización uniforme y Max-Lloyd para la primer banda de una transformada .	45
4.10. Comparación de la cuantización Max-Lloyd y Uniforme para la primer banda de una transformada . . . . .	45
4.11. Relación entre compresión (Bits totales) y error (MSE) para la banda de detalle más fino. . . . .	47
4.12. Relación entre compresión (Bits totales) y error (MSE) para cada una de las bandas de detalle . . . . .	48

5.1. Relación entre compresión (Bits por pixel) y calidad (PSNR) para el descarte progresivo de las bandas más finas. . . . .	52
5.2. Curva de desempeño del método de Error por Coeficiente en comparación con el de Descarte progresivo de las bandas más finas. . .	54
5.3. Curvas asociadas a las bandas de detalle 2 y 3 con sus respectivas derivadas. . . . .	56
5.4. Imagen Lena comprimida a 0,5 BPP. . . . .	56
5.5. Comparación del desempeño entre el método de iguales derivadas y el del error por coeficiente. . . . .	57
5.6. Comparación de la relación calidad/compresión del algoritmo de Iguales derivadas y el standard JPEG 2000 NR (No Reversible). . . . .	58
A.1. Comparación del desempeño entre los tres métodos propuestos (Imagen: Lena; Wavelet: Db4). . . . .	70
A.2. Comparación del desempeño entre tres wavelets para el método de Iguales derivadas (Imagen: Lena). . . . .	70
A.3. Comparación del desempeño entre los tres métodos propuestos (Imagen: Peppers; Wavelet: Sa4). . . . .	71
A.4. Comparación del desempeño entre tres wavelets para el método de Iguales derivadas (Imagen: Peppers). . . . .	71
A.5. Algunas de las imágenes utilizadas. . . . .	72

# Capítulo 1

## Introducción

### 1.1. Compresión

Con el avance y uso creciente de las tecnologías multimedia, la compresión de imágenes digitales adquiere una importancia cada vez mayor. Una imagen digital es, básicamente, una señal bidimensional que puede estar compuesta de una o más bandas (a veces también llamadas componentes). Las imágenes en tonos de gris están compuestas sólo de una banda, mientras que las de color están formadas por tres, en donde el significado de cada una está determinado por el sistema de color que se utilice (RGB, YUV, etc.). Otro ejemplo son las imágenes satelitales, que pueden estar formadas por una gran cantidad de bandas (7 ó más), donde la información de cada una de ellas no está relacionada con el color sino con otro tipo de información.

Las imágenes del mundo real, como las fotos, suelen ser imágenes suaves. Esto quiere decir que si un *pixel* es de un determinado tono de gris, es muy probable que los *pixel* muy próximos tengan un valor que difiera muy poco. Llamaremos a esto *correlación espacial*. Características como esta suelen ser aprovechadas por las diferentes técnicas de transformación de señales de manera de eliminar esta redundancia y poder comprimir la imagen.

Dentro del proceso de compresión se pueden distinguir tres etapas bien diferenciadas:

- Transformada
- Cuantización
- Codificación

La aplicación de una transformada a una señal original permite expresar la misma en otra base que resulte más conveniente para el posterior procesamiento.

Ejemplos de transformada son:

- la transformada discreta del Coseno (DCT)
- la transformada de Fourier
- la transformada wavelet

El paso de la aplicación de la transformada es *reversible*. Esto quiere decir que se puede recuperar la señal original sin pérdida de información.

Una vez que la señal es transformada, salvo que se trabaje con transformadas wavelet muy específicas, es absolutamente necesaria una etapa de *cuantización* para reducir la cantidad de bits para su almacenamiento, debido a que los valores de los coeficientes resultantes pueden ser de precisión arbitraria.

Esta segunda etapa es optativa y, a grandes rasgos, determina si la compresión será con o sin pérdida. Si se aplica cualquiera de los algoritmos de cuantización estaremos, de una u otra manera, descartando información de la señal original (o transformada), por lo que será imposible la reconstrucción perfecta de la misma. El propósito final de la etapa de cuantización es alcanzar el máximo grado de compresión descartando información de manera de representar a los coeficientes con una precisión no mayor a la necesaria para la calidad final requerida [Wal91].

Ejemplos de métodos de cuantización son:

- Cuantización uniforme
- Cuantización vectorial
- Cuantización Max-Lloyd

Por último, la etapa de codificación es la encargada de codificar la señal resultante de las primeras dos etapas. Según las características que se esperan encontrar en la señal a codificar existe una gran variedad entre los distintos algoritmos que se pueden utilizar. Algunos ejemplos de estos son:

- Run Length Encoding: Conveniente si hay varias apariciones consecutivas del mismo símbolo.
- Arbol de Huffman: Codificador por entropía donde se representan los símbolos más frecuentes con una menor cantidad de bits.
- Codificación aritmética: Otro codificador por entropía donde con un número fraccionario trata de representarse una gran cantidad de coeficientes mediante la partición recursiva de intervalos.

## 1.2. Objetivos

Las transformadas wavelet proyectan sucesivamente la señal sobre diferentes espacios de aproximación, llamados  $V_j$ , a distintos niveles de resolución. Cada uno de estos incluye otro espacio de aproximación con la mitad de resolución. La diferencia entre ambos es capturada por los espacios  $W_j$ , llamados *subespacios de detalle*. Por lo tanto, conservando las diferencias entre cada uno de los niveles y el siguiente se puede reconstruir la señal original. De existir una fuerte correlación espacial, la diferencia entre dos proyecciones sucesivas de la señal será muy pequeña y, por lo tanto, los coeficientes de detalle serán pequeños o nulos. Estas pequeñas diferencias suelen ser imperceptibles a los ojos.

Los coeficientes resultantes de la aplicación de la transformada wavelet en general son reales. Por lo tanto, para poder comprimir es necesario realizar una cuidadosa cuantización. Para obtener una imagen resultante de una calidad

especificada con anterioridad, se puede descartar información de cualquiera de los subespacios de detalle que componen la señal transformada.

El problema aquí planteado se presenta cada vez que se lleva a cabo una transformada wavelet independientemente de cuál sea la wavelet elegida. Es por esto que, así como en los últimos años se han realizado grandes avances en la creación y diseño de nuevos filtros para su utilización en transformadas wavelets, surge la necesidad de encontrar métodos independientes de la wavelet utilizada para el aprovechamiento de las características del proceso de la transformada en las etapas de *cuantización* y *codificación*.

El objetivo de esta tesis ha sido analizar y contrastar diversas técnicas de compresión y codificación para las distintas bandas de detalle que la componen. A su vez, se propone un análisis en profundidad de la relación entre las bandas y el diseño de nuevos métodos para el problema de la asignación de bits a las mismas, así como también la inclusión de estas en un marco teórico que sustente la propuesta.

Todas las actividades realizadas en este trabajo son (en principio) independientes de la familia de wavelets que se utilice para llevar a cabo las transformadas. Con esto se supone que los resultados y conclusiones obtenidos en el mismo podrán en un futuro seguir vigentes a pesar de los avances en la creación de nuevas familias de wavelets.

Las técnicas se aplican a diversas transformadas wavelet y multiwavelet para comparar su desempeño. También se comparan con compresores standard.

### 1.3. Contenido del trabajo

En el capítulo 2 se presentan varias medidas de error que se utilizarán a lo largo del trabajo para la comparación de los resultados entre distintos métodos. También se ven diversas técnicas básicas de compresión con pérdida de información.

En el capítulo 3 se presenta la teoría relacionada con las transformadas wavelet y multiwavelet, así como también todo lo relacionado con su ortogonalidad. Se incluyen también ejemplos de su utilización para el procesamiento de señales uni y bidimensionales.

En el capítulo 4 se aplican diversas de las técnicas presentadas en el capítulo 2 a las transformadas wavelet y se analizan las características en las distribuciones de las bandas de detalle de la transformada.

En el capítulo 5 se plantea el problema de la alocaión de bits entre las bandas y se presentan los tres métodos propuestos así como los resultados obtenidos. Finalmente se plantean los fundamentos teóricos que justifican los resultados del mejor de los tres.

En el Apéndice se incluyeron *todos* los resultados para los tres métodos propuestos para distintas imágenes, wavelets y multiwavelets.

## Capítulo 2

# Compresión con pérdida de información

### 2.1. Medidas de Error

En la compresión con pérdida el resultado es una señal que es *similar* a la original. Lo que se necesita es saber *cuán* similar es con respecto a otras señales que resulten producto de otros métodos. Descartado queda el artesanal método de mirar la señal resultante (ver cuál se parece más a la original), no sólo por subjetivo, sino por impráctico. Si se tuvieran que comparar cientos de imágenes se necesitaría una forma automática y objetiva de hacerlo. Para esto es que se formaliza la siguiente definición de error para una señal unidimensional  $\tilde{f}$  aproximada.

$$e_k = f_k - \tilde{f}_k \quad (2.1)$$

En esta, el error resultante es un vector de exactamente la misma longitud que la señal original y la aproximada. La pregunta es: ¿Cómo definimos un orden entre los posibles errores?

#### 2.1.1. Norma 2

Una de las *medidas de error* que se utilizan para discernir esto es la norma 2 al cuadrado del vector, la cual se define de la siguiente manera:

$$\|e\|_2^2 = \sum_{k=0}^{N-1} (e_k)^2 \quad (2.2)$$

Esta medida permitirá, por ejemplo, comparar los resultados de la aplicación de dos (o más) métodos de compresión a la misma señal, dando por supuesto que las señales resultantes tienen el mismo tamaño que la original. Al calcular el error de cada una de ellas y, posteriormente, la norma 2 al cuadrado quedará definido un orden entre los errores a evaluar.

### 2.1.2. Error cuadrático medio - MSE

Sea una señal  $f$  de  $2N$  componentes de longitud. Definimos un método de compresión con pérdida de información que luego de aplicarlo a la señal  $f$  cumple la siguiente propiedad:

$$\forall k : 0 \leq k < 2N : e_k = 1$$

donde  $e$  se calcula según la ecuación 2.1.

A partir de esta propiedad se sabe que

$$\|e\|_2^2 = \sum_{k=0}^{2N-1} (e_k)^2 = 2N$$

Sin embargo, si se comprime sólo la primer mitad de la señal  $x(0 \dots N)$  se llega a que

$$\|e\|_2^2 = \sum_{k=0}^N (e_k)^2 = N$$

con lo cual, esta medida de error no resulta eficaz para evaluar métodos que sean aplicados sobre señales de distinta longitud. Esto se produce debido a que el error es acumulativo y aunque un determinado método muestre una mejor performance que otro, siempre se lo puede evaluar sobre una señal lo suficientemente extensa como para que tenga mayor norma 2.

Para resolver esto se necesita una medida de error que sea independiente del tamaño de la señal. Introduciendo esta mejora, se llega a la definición del MSE<sup>1</sup>:

$$\text{MSE}(e) = \frac{1}{N} \sum_{k=0}^{N-1} (e_k)^2 \quad (2.3)$$

donde  $N$  es la longitud de la señal.

En esta definición se puede ver que el error es ponderado por el tamaño de la señal, lo cual permitirá poder comparar resultados de pruebas realizadas sobre señales de distinta cantidad de componentes.

### 2.1.3. PSNR

Si se supone que se evalúan dos métodos sobre señales distintas pero de la misma longitud y que el vector *error* es exactamente igual para las dos, de manera tal que

$$\forall i : 0 \leq i < N : e_i = 1$$

Si se utiliza la norma 2 o el MSE como medida de error, será imposible decir que uno de los métodos es mejor que el otro, pues el cálculo de cualquiera de ellas es función del vector *error*. Que el vector *error* tome el valor 1 en todas sus componentes quiere decir que la señal aproximada está desplazada hacia abajo en una unidad en los resultados de ambos métodos.

<sup>1</sup>Medium Square Error: Error cuadrático medio

Sin embargo, si la señal original de la primer prueba toma valores entre 1.000.000 y 2.000.000 mientras que la señal original de la segunda prueba toma valores entre 3 y 8, el peso de sufrir un desplazamiento igual a uno (1) en una señal como la segunda es mucho mayor que en una como la primera. Para poder comparar resultados de pruebas hechas sobre señales que se muevan en rangos distintos se necesita una medida de error que incluya esta información en su definición. Es así como llegamos a la definición de PSNR<sup>2</sup>:

$$\text{PSNR}(e) = 10 \cdot \log_{10} \left( \frac{(\text{rango\_de\_la\_senal})^2}{\text{MSE}(e)} \right) \text{dB} \quad (2.4)$$

Esta definición salva todos y cada uno de los inconvenientes que hemos apuntado anteriormente. A diferencia de las otras, a medida que el valor es más alto, menor es el error. Esto se debe a que el MSE se encuentra en el denominador.

En lo que respecta a los próximos capítulos, se trabajará sólo con señales que se muevan en un rango de 0 a 255 (8 bits), con lo cual será indistinto la aplicación del MSE o del PSNR.

## 2.2. Diversas técnicas de compresión con pérdida de información

### 2.2.1. Submuestreo

Una de las formas más sencillas de comprimir, aún sin tener que analizar la señal previamente, es la técnica del *submuestreo*, también llamado *decimación*. Esta consiste en un concepto tan sencillo como enviar (o preservar) uno de cada  $n$  componentes. Esta acción se denomina “Submuestrear de a  $n$ ”. Obviamente el tamaño de la señal resultante será  $n$  veces más pequeña.

Si bien la técnica de submuestreo detalla específicamente qué información se debe perder, dejando de lado cualquier posibilidad de duda, no pasa lo mismo cuando se necesita recuperar las componentes que fueron descartadas.

Una posibilidad es repetir el valor anterior. Sin embargo, eso produciría como resultado una señal más escalonada y menos suave. Si se supiera que la señal era muy suave, se podría completar con el promedio de los valores vecinos (en el caso del submuestreo de a 2). Si se submuestreara por un  $n$  mayor, existen infinidad de funciones que podrían completar la información faltante con mayor o menor éxito.

El submuestreo de a 2, por ejemplo se define mediante

$$y = x \downarrow 2$$

$$y_k = x_{2k}$$

A continuación se muestra un ejemplo para la primer opción explicada. Si una señal es submuestreada de a  $n$ , para reconstruir una señal aproximada a la original se tendría que repetir  $n - 1$  veces cada componente que aparezca en la señal comprimida.

Si se definiera la señal  $x$  de la siguiente manera:

$$x = [4 \ 5 \ 9 \ 8 \ 3 \ 2 \ 4 \ 7 \ 9 \ 6]$$

<sup>2</sup>PSNR: Peak-Signal Noise Ratio

Señal	Umbral	Resultado
[ 34 87 23 56 83 51 38 12 ]	30	[ 34 87 0 56 83 51 38 0 ]
[ 34 87 23 56 83 51 38 12 ]	50	[ 0 87 0 56 83 51 0 0 ]
[ 34 87 23 56 83 51 38 12 ]	80	[ 0 87 0 0 83 0 0 0 ]

Figura 2.1: Aplicación de distintos umbrales a una señal.

y se la submuestreara de a 2 ( $x \downarrow 2$ ), daría por resultado:

$$x \downarrow 2 = [4\ 9\ 3\ 4\ 9]$$

Tal como se mencionó anteriormente, la reconstrucción de la señal aproximada se llevaría a cabo a través de la repetición de cada una de las componentes según de a cuanto se haya submuestreado. En este caso, cada componente del vector submuestreado equivale a 2 componentes del original. Es por esto que se repite una vez cada una para suplantar a la ausente. La señal reconstruida sería:

$$\tilde{x} = [4\ 4\ 9\ 9\ 3\ 3\ 4\ 4\ 9\ 9]$$

Es muy sencillo ver dónde se realiza la compresión mediante esta técnica. Cada vez que submuestreamos, estamos descartando componentes que nunca enviaremos al receptor. De esta manera, los niveles de compresión están directamente determinados por el  $n$  elegido.

### 2.2.2. Umbral

La técnica del *Umbral* consiste en poner en cero todas las componentes de una señal que no sobrepasen un determinado valor.

Para una señal  $[x_0 \dots x_N]$  dada,  $y$  es la señal resultante de aplicar la función  $u(x, h)$  a cada componente de la señal:

$$\forall j : 0 \leq j < N : y_j = u(x_j, h)$$

Se define la función  $u(x, h)$  como

$$u(x, h) = \begin{cases} 0 & \text{si } x < h \\ x_j & \text{si } x \geq h \end{cases}$$

Se dice que  $y$  es el resultado de aplicar un umbral igual a  $h$  a la señal  $x$  mediante la función  $U$ .

Algunos ejemplos pueden verse en la figura 2.1, en el que la aplicación de distintos umbrales da como resultado señales con mayor cantidad de ceros a medida que el umbral crece.

Obviamente, para que tenga sentido, el umbral fijado debe encontrarse dentro del rango de valores que puede tomar la señal. De lo contrario, el resultado de la operación será una señal de ceros (en el caso de situarse por encima del rango) o la señal original sin alteraciones (si se encuentra por debajo). Igualmente, aunque se pueda asegurar que el umbral se encontrará dentro del rango de valores posibles, muchas veces no tiene sentido fijarlo de antemano, ya que si se desconoce el rango efectivo<sup>3</sup> de la señal, no se tendrá forma de saber ni siquiera una cantidad estimada de valores reemplazados por cero. En estos casos, lo que más conviene es asegurarse de establecer un umbral relativo al rango de los valores en que se mueve la señal o a la distribución de los mismos. Una forma de hacerlo es ordenar de forma descendente los valores absolutos y tomar el  $n$ -ésimo valor como umbral, con lo cual el resultado de la operación dejará el  $\frac{100-n}{N}$  % de valores distintos de cero.

¿Dónde se gana en compresión cuando se aplica un Umbral? No resulta obvio que se esté mejorando algo. La cantidad de coeficientes sigue siendo la misma con lo que parece que la señal ocupará lo mismo que antes, pero si se analiza con detenimiento la señal resultante se notará que esta tiene dos posibles ventajas, no excluyentes entre sí, para aprovechar:

1. Si la cantidad de ceros que deja el umbral lo justifica, se puede aplicar sobre la señal algún método de codificación como el RLE<sup>4</sup>. Para aprovechar al máximo la ubicación de los ceros, incluso se podría determinar si suelen aparecer en algunas zonas en particular para poder enviar los componentes en un determinado orden en el cual la mayor cantidad de ceros posibles puedan ser enviados simultáneamente. Lo que pueda comprimirse depende tanto del umbral aplicado como del orden en que se envíen los componentes.
2. Como consecuencia de haber aplicado el umbral, el histograma de la señal resultante tendrá en el valor cero (0) una frecuencia relativa igual a *la suma de todas las frecuencias relativas de los valores que se encontraron por debajo del umbral*. Esto provocará que, a medida que el umbral suba, la entropía decrezca cada vez más, obteniendo una señal que puede ser altamente comprimible con cualquier codificador por entropía (Ej.: Huffman).

Así, la técnica del umbral transforma a la señal original de manera que cualquiera de estas características puedan ser aprovechadas en la etapa de codificación.

### 2.2.3. Cuantización

La cuantización consiste en la definición de uno o más intervalos en los que todos los valores de la señal que se encuentren dentro de este serán reemplazados por un representante asociado al mismo.

<sup>3</sup>Se llamará rango efectivo a los determinados por el mínimo y el máximo valor que toma la señal.

<sup>4</sup>Run Length Encoding: Método de codificación por el cual se envía una dupla donde la primer componente es un valor y la otra es la cantidad de apariciones consecutivas que representa

Dada una cuantización en  $N$  intervalos, sea  $d$  un vector de longitud  $N+1$  con los límites de los intervalos y  $r$  un vector de longitud  $N$  con los representantes. Entonces

$$Q(x) = r_j \text{ si } x \in [d_{j-1}, d_j)$$

Si el representante pertenece también al intervalo,

$$r_j \in [d_{j-1}, d_j)$$

se dice que la cuantización es *regular*.

Existe más de una técnica de cuantización, sin embargo, el hecho de que la cantidad de valores distintos disminuya, en cualquiera de las técnicas, trae aparejadas dos consecuencias:

1. disminuye la cantidad de bits que se necesita para representar cada uno de los coeficientes cuantizados.
2. la entropía bajará por haberse achicado el rango de los valores resultantes.

Dos de los métodos más conocidos de cuantización son el de Cuantización Uniforme y el de Max-Lloyd [Lim90].

### Cuantización uniforme

En la técnica de cuantización uniforme, la totalidad de los intervalos definidos tienen exactamente la misma longitud. De esta manera, se define como

$$\tilde{x} = Q(x) = \left[ \frac{x}{q} \right] \cdot q \quad (2.5)$$

donde la longitud del intervalo es igual a  $q$ , al que se lo llama el *paso de cuantización*.

Sean  $d_j$  los límites de los intervalos definidos y  $r_i$  los representantes de estos. Los representantes tienen la forma

$$\forall i : i \in Z : r_i = (i - 1)q$$

y los límites de los intervalos

$$\forall j : j \in Z : d_j = j \cdot q$$

La cuantización uniforme define una cantidad de intervalos igual a  $\left\lceil \frac{\text{rango}}{q} \right\rceil$ , aunque según la distribución de la señal original puede que no haya apariciones en alguno de los intervalos.

En la figura 2.2 se puede ver la distribución de los valores de la señal original y los intervalos definidos. Tal como mencionábamos, se sabe que el rango de las señales es de 0 a 255, pero la señal del ejemplo no alcanza esos valores, por lo que el último intervalo quedará vacío.

Queda claro que si se aplicara una cuantización uniforme con la misma cantidad de intervalos a otra señal, la definición de los intervalos sería idéntica, ya que no se utiliza en ningún momento información sobre la distribución de los valores de la señal.

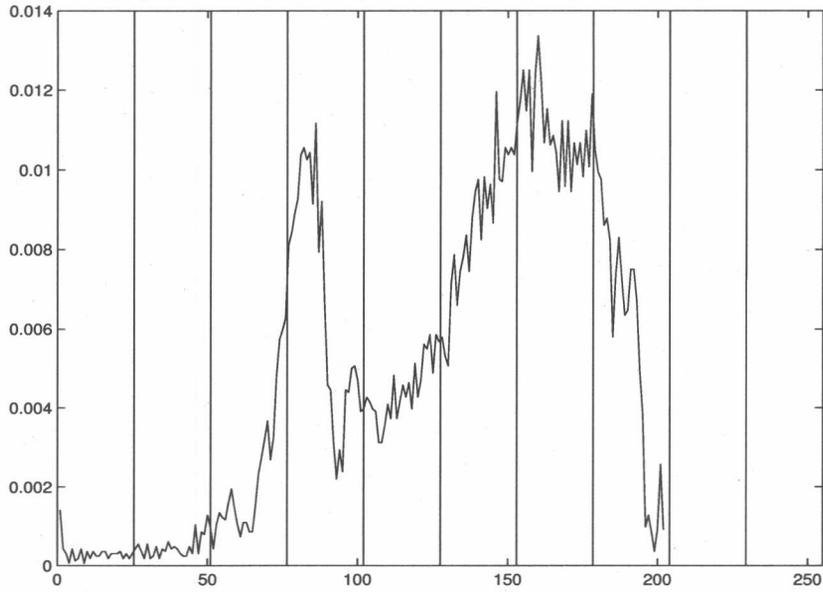


Figura 2.2: Distribución de una señal original e intervalos definidos por la cuantización uniforme.

### Cuantización Max-Lloyd

Dada una determinada señal a la cual se quiere cuantizar en una cantidad de intervalos igual a  $N$ , el algoritmo de cuantización de Max-Lloyd, encuentra la distribución óptima de intervalos y representantes de manera de minimizar el error cuadrático medio. Se define a la *distorsión promedio* ( $D$ ) como

$$D = \sum_{i=1}^N \sum_{d_{i-1} \leq x_j < d_i} (x_j - r_i)^2 \cdot p(x_j) \quad (2.6)$$

que es igual al error cuadrático medio de la cuantización.

Se probarán dos propiedades con respecto a la elección de representantes e intervalos. En primer lugar, se quieren encontrar los representantes  $r_i$  óptimos para los intervalos dados.

Como se quiere el mejor  $r_i$  para cada intervalo de manera que la distorsión sea mínima, se deriva con respecto a uno de los representantes  $r_k$  y se iguala a cero.

$$\frac{\partial D}{\partial r_k} = \sum_{d_{k-1} \leq x_j < d_k} 2(x_j - r_k) \cdot p(x_j)(-1) = 0 \quad (2.7)$$

$$\sum_{d_{k-1} \leq x_j < d_k} (x_j - r_k) \cdot p(x_j) = 0 \quad (2.8)$$

$$\sum_{d_{k-1} \leq x_j < d_k} x_j p(x_j) = r_k \sum_{d_{k-1} \leq x_j < d_k} p(x_j) \quad (2.9)$$

$$r_k = \frac{\sum_{d_{k-1} \leq x_j < d_k} x_j p(x_j)}{\sum_{d_{k-1} \leq x_j < d_k} p(x_j)} \quad (2.10)$$

Con lo cual, si se quiere minimizar la distorsión ( $D$ ),  $r_k$  debe ser *el valor esperado de la señal para el intervalo*  $[d_{k-1}, d_k]$ .

En segundo lugar, se supone que los representantes  $r_i$  están definidos y se quieren encontrar los límites de los intervalos  $d_i$  óptimos para minimizar  $D$ . Para simplificar las ecuaciones, reemplazaremos la sumatoria dentro del intervalo (discreto) por su integral (continuo).

$$D_c = \sum_{i=1}^N \int_{d_{i-1}}^{d_i} (x - r_i)^2 p(x) dx \quad (2.11)$$

Sólo en dos términos aparece  $d_k$ , por lo que

$$\frac{\partial D_c}{\partial d_k} = \frac{\partial}{\partial d_k} \left( \int_{d_{k-1}}^{d_k} (x - r_k)^2 p(x) dx + \int_{d_k}^{d_{k+1}} (x - r_{k+1})^2 p(x) dx \right) \quad (2.12)$$

$$= \frac{\partial}{\partial d_k} \left( \int_{d_{k-1}}^{d_k} (x - r_k)^2 dx - \int_{d_{k+1}}^{d_k} (x - r_{k+1})^2 dx \right) = 0 \quad (2.13)$$

$$= (d_k - r_k)^2 p(d_k) - (d_k - r_{k+1})^2 p(d_k) = 0 \quad (2.14)$$

Suponemos que  $p(d_k) \neq 0$  y

$$\Rightarrow d_k = \frac{r_k + r_{k+1}}{2} \quad (2.15)$$

Por lo tanto, dados los representantes, si se quiere minimizar el error cuadrático medio al elegir los límites de los intervalos, estos deben estar exactamente entre los dos representantes más cercanos.

El algoritmo de Max-Lloyd utiliza estos dos resultados (2.10 y 2.15) de manera iterativa para ir mejorando cada vez los intervalos y los representantes. Lo que el algoritmo necesita como datos de entrada es:

- la distribución de los valores de la señal a cuantizar
- una configuración inicial de intervalos a partir de los cuales empezar a calcular el algoritmo
- una condición de corte de ejecución

Dados los intervalos propuestos inicialmente, en un primer paso, el algoritmo encuentra los representantes óptimos para esos intervalos mediante su definición a través de la ecuación 2.15.

Una vez definidos los representantes, en un segundo paso, se reemplazan los intervalos utilizados por un nuevo cálculo de los mismos a partir de la definición de los representantes y la ecuación 2.10.

Estos dos pasos pueden repetirse tantas veces como se quiera. En algún momento, la aplicación de estos no producirá cambio alguno, indicando que el

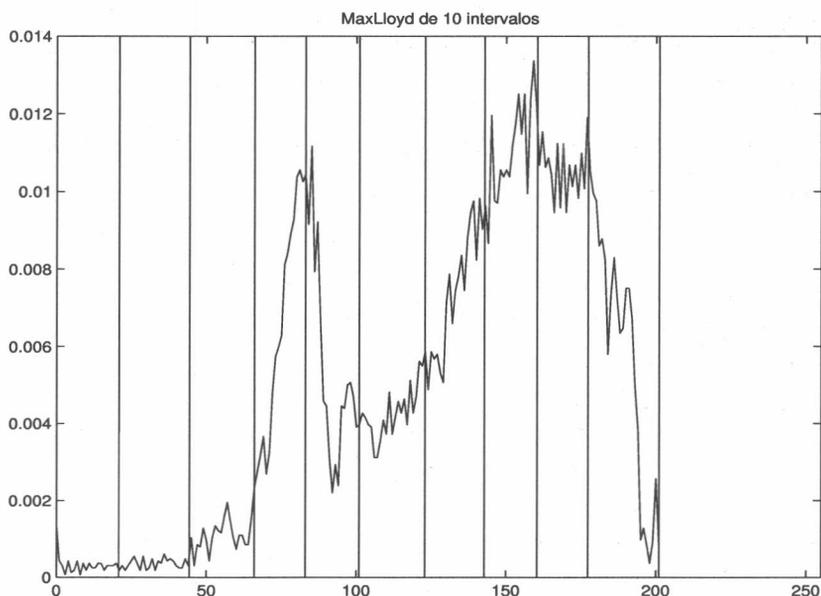


Figura 2.3: Distribución de una señal original e intervalos definidos por la cuantización Max-Lloyd.

algoritmo ha convergido a la solución buscada. Sin embargo, existe la posibilidad de que el algoritmo produzca mínimos cambios indefinidamente alrededor de la solución, por lo que suele definirse una cantidad máxima de ciclos en la que se supone que el error cometido será despreciable.

De esta manera, iterativamente, la definición de intervalos y representantes tienden a una distribución que minimiza el error cuadrático medio.

### Ventajas y desventajas de cada una de las dos

La cuantización uniforme se define de manera muy sencilla ya que sólo toma en cuenta el rango de los valores posibles de la señal, dividiéndolo en intervalos de igual longitud. Esto quiere decir que para cualquier otra señal con el mismo rango, la definición de los intervalos será la misma. Si se define una cantidad suficiente de intervalos, la distribución resultante suele ser muy similar a la original.

En la cuantización Max-Lloyd, se asegura que el error cuadrático medio será minimizado mediante el recálculo iterativo de los intervalos y sus representantes. Los intervalos serán más finos donde se concentre más la información, por ejemplo, en los picos de la distribución de valores. Esto hace que la distribución resultante de valores en la imagen cuantizada sea más uniforme que la original.

Relacionando las características de estos dos tipos de cuantizaciones podemos asegurar que, para la misma cantidad de niveles, la imagen resultante de la cuantización Max-Lloyd tendrá menos error que la de la cuantización uniforme. Por otro lado, para que Max-Lloyd minimice el MSE fijará más intervalos en

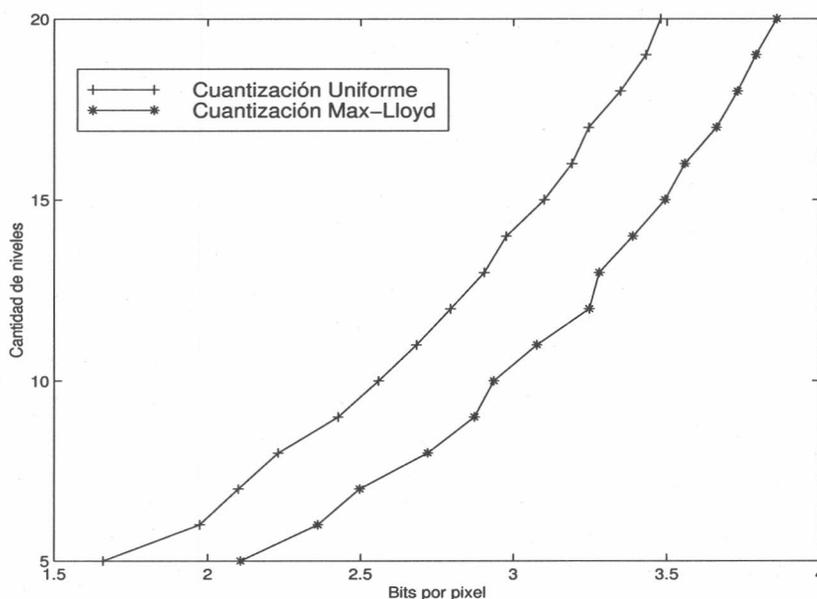


Figura 2.4: Comparación de la entropía resultante entre cuantización uniforme y Max-Lloyd (Imagen Woman)

donde más apariciones hubiera. Esto, si bien minimizará el MSE, también provocará que los *picos* del histograma pierdan altura relativa y ensanchen su base si los comparamos en el histograma antes de la cuantización y en el de los representantes. El efecto final es una tendencia a la *uniformización* del histograma. Asociándolo con la *entropía*, se recuerda que esta se maximiza cuando la probabilidad de aparición de cada uno de los símbolos es la misma, o sea, tiene una distribución uniforme. Si a esto se le agrega que la cuantización uniforme no garantiza en lo más mínimo que los representantes tengan una distribución uniforme, se concluirá que la señal resultante de la cuantización Max-Lloyd va a tener *en general* mayor entropía que la cuantizada uniformemente.

En la figura 2.4 se grafica la relación entre cantidad de intervalos y los bits por píxel resultantes de la cuantización para ambos métodos en una imagen particular.

En la figura 2.5 se grafica, para distinta cantidad de intervalos de cuantización, la calidad de la imagen (PSNR) en función de la compresión alcanzada (BPP<sup>5</sup>) para ambos algoritmos de cuantización.

Puede verse que mientras la cantidad de intervalos es alta, ambos métodos se comportan de manera similar. La explicación de esto es que, en ese caso, el algoritmo de Max-Lloyd tiene que ubicar una cantidad de intervalos que es del orden de la cantidad de valores originales y no le queda mucho margen para moverlos. Es decir, la mejor distribución de los intervalos, la que minimiza el MSE, es muy parecida a la uniforme. La gran diferencia se empieza a notar a

<sup>5</sup> BPP: Bits por píxel. Cantidad promedio de bits que se necesitará para codificar la imagen, en este caso particular, cuantizada.

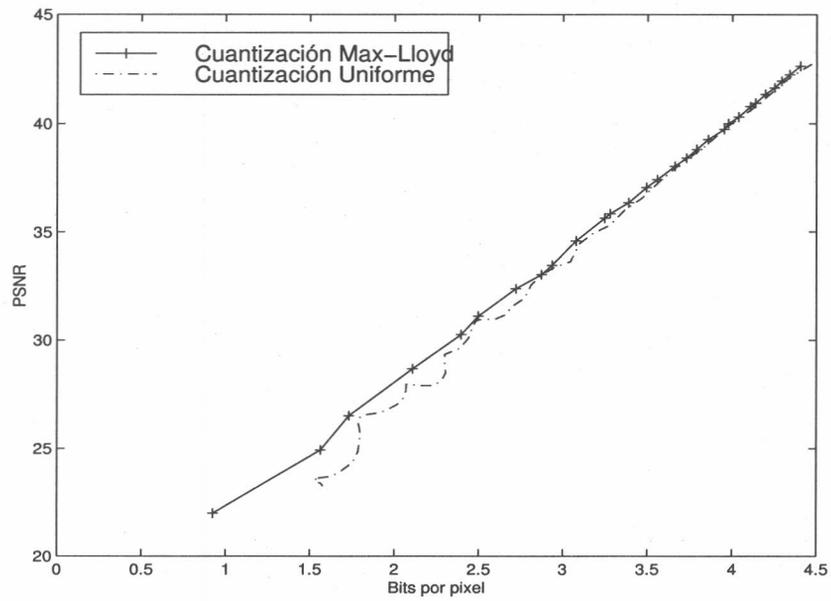


Figura 2.5: Comparación entre el desempeño de la cuantización uniforme y Max-Lloyd (Imagen Woman)

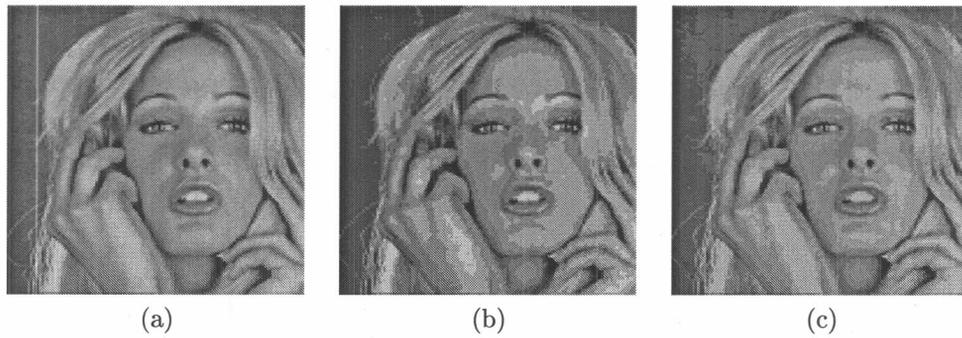


Figura 2.6: (a) Imagen original -  $H = 7.11$ . (b) Cuantización uniforme -  $H = 2.54$  -  $MSE = 52$ . (c) Cuantización Max-Lloyd -  $H = 2.94$  -  $MSE = 29$ .

medida que el número de intervalos decrece. A partir de una codificación aproximada de 3 bpp, la cuantización por Max-Lloyd empieza a mejorar cada vez más la distribución de intervalos. Al tener pocos intervalos tiene más espacio para, en caso que la distribución de la imagen esté concentrada, mover los intervalos una mayor distancia. Por debajo de los 2 bpp la cuantización uniforme empieza a tener un comportamiento bastante irregular y por debajo del de Max-Lloyd. Por el contrario, Max-Lloyd mantiene la pendiente de su curva más allá del número de intervalos con el que se cuantice, mostrando un comportamiento más previsible que la uniforme.

En la otra figura puede observarse un ejemplo de cómo son las distorsiones de las imágenes según el método de cuantización para una misma cantidad de niveles. Al ser una imagen con mayor presencia de grises claros, presentes en rostro y cabello, Max-Lloyd asigna un mayor número de intervalos a ese rango de valores. Es notable la diferencia de calidad en el rostro, el cabello e incluso en las manos con respecto a las mismas zonas de la imagen cuantizada uniformemente. Sin embargo, el haber corrido los intervalos hacia la zona de los tonos más claros perjudicó a los tonos más oscuros. Esto puede observarse en el fondo, donde la cuantización uniforme asigna tonos muy similares a los originales, mientras que Max-Lloyd asigna tono muy claros en algunos sectores como la parte superior izquierda.

## Capítulo 3

# Wavelets

Contamos con una señal a la que se llamará  $f$ . La idea principal de una transformada es poder escribir esta señal de otra manera. En el caso de las transformadas wavelet, como combinación de funciones  $\psi_{j,i}$ .

$$f(x) = \sum_j \sum_i a_{j,i} \cdot \psi_{j,i}(x) \quad (3.1)$$

Justamente la función  $\psi$  es la función que se denomina *wavelet*, también llamada ondita, ondícula u ondelette. Más adelante se definirán los dos parámetros ( $j$  e  $i$ ) que acompañan a la función  $\psi$ .

### 3.1. Espacios $V$ y $W$

Llamaremos

$$L^2(\mathfrak{R}) = \left\{ f(x) : f : \mathfrak{R} \rightarrow \mathfrak{R} \mid \int_{-\infty}^{+\infty} |f^2(x)| dx < \infty \right\} \quad (3.2)$$

Se definirán a los espacios  $V_j$  de manera tal que cumplan con las siguientes características [Uyt94]:

$$\forall j \in Z : V_j \subset V_{j+1} \quad (3.3)$$

$$\bigcap_{j=-\infty}^{+\infty} V_j = \{0\} \quad (3.4)$$

$$\lim_{j \rightarrow \infty} V_j = L^2(\mathfrak{R}) \quad (3.5)$$

Los espacios  $W_j$  quedarán definidos en base a los  $V_j$  y los  $V_{j+1}$ . Como estos dos últimos están incluidos uno dentro del otro, denominaremos  $W_j$  al complemento ortogonal de  $V_j$  en relación a  $V_{j+1}$ .

De esta manera, se cumple que

$$W_i \perp V_i \quad (3.6)$$

$$\forall j \in Z : W_j \oplus V_j = V_{j+1} \quad (3.7)$$

De lo anterior y de la ecuación 3.5 se puede deducir que

$$\bigoplus_{j=-\infty}^{+\infty} W_j = L^2 \quad (3.8)$$

### 3.2. Función escala y función wavelet

Ahora que están definidos los espacios  $V_j$  y  $W_j$  se definirán las funciones que los generan.

Se llamará *función de escala* a una función  $\phi$  tal que

$$\phi(x) \in V_0$$

y que

$$\{\phi(x - i)\} \text{ con } i \in Z$$

es una base para el espacio cuya clausura es  $V_0$ . Exigiremos que esta base sea ortonormal, es decir que

$$\langle \phi(\cdot - j), \phi(\cdot - i) \rangle = \delta_{j,i}$$

Así como se puede trasladar a la función  $\phi$ , también se puede escalar. Los escalamientos que consideraremos para  $\phi$  son diádicos, y además definimos

$$\phi_{j,i}(x) = 2^{\frac{j}{2}} \phi(2^j x - i) \quad (3.9)$$

Esto permitirá que su norma sea siempre la misma, ya que se cumple que

$$\begin{aligned} \forall j \in Z : \langle \phi_j(x), \phi_j(x) \rangle &= \int_{-\infty}^{+\infty} 2^{\frac{j}{2}} \cdot \phi(2^j x) \cdot 2^{\frac{j}{2}} \cdot \phi(2^j x) dx = \\ &= 2^j \cdot \int_{-\infty}^{+\infty} \phi(2^j x) \cdot \phi(2^j x) dx = \\ &= 2^j \cdot \int_{-\infty}^{+\infty} \phi(y) \cdot \phi(y) \frac{dy}{2^j} = \\ &= \langle \phi(y), \phi(y) \rangle \end{aligned}$$

Análogamente para los espacios  $W$ , se llamará *función wavelet* a una función  $\psi$  tal que

$$\psi(x) \in W_0$$

y que

$$\{\psi(x - i)\} \text{ con } i \in Z$$

es una base para el espacio cuya clausura es  $W_0$ . Al igual que con  $\phi$ , también exigiremos que la base sea ortonormal, es decir

$$\langle \psi(\cdot - j), \psi(\cdot - i) \rangle = \delta_{j,i}$$

Los escalamientos para  $\psi$  son diádicos y quedarán definidos por

$$\psi_{j,i}(x) = 2^{\frac{j}{2}} \psi(2^j x - i) \quad (3.10)$$

La norma de  $\psi_j$  también será siempre la misma, ya que se cumple que

$$\begin{aligned} \forall j \in Z : \langle \psi_j(x), \psi_j(x) \rangle &= \int_{-\infty}^{+\infty} 2^{\frac{j}{2}} \cdot \psi(2^j x) \cdot 2^{\frac{j}{2}} \cdot \psi(2^j x) dx = \\ &= 2^j \cdot \int_{-\infty}^{+\infty} \psi(2^j x) \cdot \psi(2^j x) dx = \\ &= 2^j \cdot \int_{-\infty}^{+\infty} \psi(y) \cdot \psi(y) \frac{dy}{2^j} = \\ &= \langle \psi(y), \psi(y) \rangle \end{aligned}$$

### 3.3. Descomposición de una función

Como los subespacios  $V$  están incluidos uno dentro del otro, se puede expresar la *función de escala*, correspondiente a un determinado nivel, como una combinación lineal de las funciones que generan el espacio que lo contiene. Es decir que como

$$V_j \subset V_{j+1}$$

se puede expresar que

$$\phi_{j,0}(x) = \sum_i h_i \phi_{j+1,i}(x)$$

y si  $j = 0$

$$\phi_{0,0}(x) = \sum_i h_i \phi_{1,i}(x) \quad (3.11)$$

$$= \sum_i h_i 2^{\frac{1}{2}} \phi(2x - i) \quad (3.12)$$

entonces

$$\phi(x) = \sqrt{2} \sum_i h_i \phi(2x - i)$$

Análogamente se puede hacer lo mismo para la *función wavelet*. Como

$$W_j \subset W_{j+1}$$

puedo expresar que

$$\psi_{j,0}(x) = \sum_i g_i \psi_{j+1,i}(x)$$

y si  $j = 0$

$$\psi_{0,0}(x) = \sum_i g_i \psi_{1,i}(x) \quad (3.13)$$

$$= \sum_i g_i 2^{\frac{1}{2}} \psi(2x - i) \quad (3.14)$$

entonces

$$\psi(x) = \sqrt{2} \sum_i g_i \psi(2x - i)$$

Sea  $f \in L^2(\mathfrak{R})$ , y se supone que para un determinado  $j$  se cumple que  $f \in V_j$ . Si suponemos que  $j = 0$ , se puede expresar  $f$  como

$$f(x) = \sum_k a_k \phi(x - k) \quad (3.15)$$

Se sabe también que el espacio  $V_0$  contiene los espacios  $V_{-1}$  y  $W_{-1}$  y aparte que la suma directa de ambos genera exactamente  $V_0$  (ver ecuación 3.6), por lo que se puede expresar  $f(x)$  como la suma entre la proyección en  $V_{-1}$  y la proyección en  $W_{-1}$ .

Para encontrar su proyección en  $V_{-1}$  se puede plantear el producto interno con cada uno de los integrantes de la base que generan el espacio. Por ejemplo, para averiguar el  $i$ -ésimo coeficiente de su proyección se resuelve

$$\langle f, \phi_{-1,i} \rangle = \left\langle \sum_k a_k \phi(\cdot - k), \frac{1}{\sqrt{2}} \phi\left(\frac{\cdot}{2} - i\right) \right\rangle \quad (3.16)$$

$$= \frac{1}{\sqrt{2}} \sum_k a_k \langle \phi(\cdot - k), \phi\left(\frac{\cdot}{2} - i\right) \rangle \quad (3.17)$$

$$= \frac{1}{\sqrt{2}} \sum_k a_k \left\langle \phi(\cdot - k), \sqrt{2} \sum_j h_j \phi(\cdot - 2i - j) \right\rangle \quad (3.18)$$

$$= \sum_k a_k \sum_j h_j \langle \phi(\cdot - k), \phi(\cdot - 2i - j) \rangle \quad (3.19)$$

donde se cumple que

$$\langle \phi(\cdot - k), \phi(\cdot - 2i - j) \rangle = \begin{cases} 1 & \Leftrightarrow k = 2i + j \\ 0 & \Leftrightarrow k \neq 2i + j \end{cases} \quad (3.20)$$

Reemplazando el producto interno en la ecuación 3.19,

$$\langle f, \phi_{-1,i} \rangle = \sum_k a_k h_{k-2i} \quad (3.21)$$

y definiendo  $h'$  como

$$h'_i = h_{-i} \quad (3.22)$$

se llega a que este resultado es

$$\sum_k a_k h_{k-2i} = \sum_k a_k h'_{2i-k} = ((a \star h') \downarrow 2)_i \quad (3.23)$$

Es decir, la *convolución* entre  $a$  y  $h'$  submuestreada de a 2.

El procedimiento para obtener la proyección de  $f$  en  $W_{-1}$  es análogo y el resultado será

$$\langle f, \psi_{-1,i} \rangle = ((a \star g') \downarrow 2)_i \quad (3.24)$$

Obviamente, si luego de la convolución se submuestra la señal de a 2, quedarán sólo con la mitad de los coeficientes. Es decir, la mitad de coeficientes para la proyección en  $V_{-1}$ , pero se necesitarán una cantidad similar para la proyección en  $W_{-1}$ .

### 3.4. El algoritmo cascada

El procedimiento anterior será la base del algoritmo que se detallará a continuación y que se denomina *algoritmo cascada*. Una vez que se proyecta la señal original, perteneciente al espacio  $V_j$ , en el subespacio  $V_{j-1}$  nada impide seguir descendiendo en los subespacios incluidos en este, en un ciclo en el que cada vez la proyección se realiza sobre un espacio más pequeño.

Si se llama  $f_0(x)$  a la función original que pertenece al espacio  $V_0$ , se demuestra que se puede expresar como

$$f_0(x) = f_{-1}(x) + r_{-1}(x) \quad (3.25)$$

donde  $f_{-1}(x)$  es su proyección en el espacio  $V_{-1}$  y  $r_{-1}(x)$  en el espacio  $W_{-1}$ .

Repetiendo la operación para las  $f_j$  con  $j < 0$

$$\begin{aligned} f_0(x) &= f_{-1}(x) + r_{-1}(x) \\ &= f_{-2}(x) + r_{-2}(x) + r_{-1}(x) \\ &= \dots \\ &= f_{-p}(x) + r_{-p}(x) + \dots + r_{-1}(x) \end{aligned}$$

se puede ver que se llega a descomponer la función original en distintas funciones  $r_j$  que pertenecen a espacios  $W_j$  y sólo una  $f_{-p}$  que pertenece a  $V_{-p}$ .

Debe recordarse que cada vez que se desciende un nivel se obtienen como resultado dos señales, donde cada una está compuesta por la mitad de los coeficientes que la del nivel anterior, con lo que la cantidad de coeficientes total sigue siendo la misma. Vale aclarar que sólo se podrá descender una cantidad de niveles igual al  $\log_2$  de la cantidad de coeficientes de  $f_0$ , ya que en el último paso  $f_{-p}$  tendrá sólo un coeficiente.

Uniendo este resultado con el procedimiento para proyectar una señal sobre los espacios  $V$  y  $W$  incluidos en el espacio original, se grafica el algoritmo cascada en la figura 3.1.

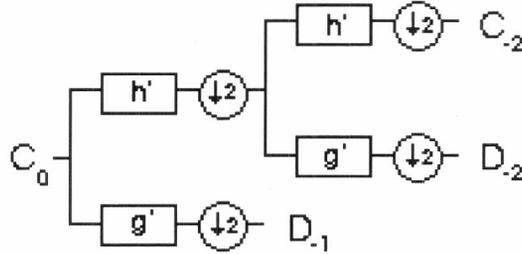


Figura 3.1: Algoritmo cascada

### 3.5. Propiedades de la transformada ortogonal

Para que una transformada sea ortogonal, deben cumplirse las siguientes propiedades, como condiciones necesarias [Val99] pero no suficientes:

$$\langle \phi(\cdot), \phi(\cdot - i) \rangle = \begin{cases} 1 & \Leftrightarrow i = 0 \\ 0 & \Leftrightarrow i \neq 0 \end{cases} \quad (3.26)$$

$$\langle \psi(\cdot), \psi(\cdot - i) \rangle = \begin{cases} 1 & \Leftrightarrow i = 0 \\ 0 & \Leftrightarrow i \neq 0 \end{cases} \quad (3.27)$$

y como

$$\forall j : V_j \perp W_j$$

entonces

$$\forall i \in Z : \langle \phi(\cdot), \psi(\cdot - i) \rangle = 0 \quad (3.28)$$

Para que se cumpla la primer propiedad, el filtro asociado  $h$  debe tener ciertas características, las cuales se derivan del siguiente desarrollo.

$$\begin{aligned} \langle \phi(\cdot), \phi(\cdot - i) \rangle &= \left\langle \sqrt{2} \sum_k h_k \phi(2 \cdot - k), \sqrt{2} \sum_j h_j \phi(2 \cdot - 2i - j) \right\rangle \quad (3.29) \\ &= \sum_k h_k \sum_j h_j \left\langle \sqrt{2} \phi(2 \cdot - k), \sqrt{2} \phi(2 \cdot - 2i - j) \right\rangle \quad (3.30) \end{aligned}$$

Pero de la ecuación 3.20 y 3.10 se puede deducir que

$$\langle \sqrt{2}\phi(2 \cdot -k), \sqrt{2}\phi(2 \cdot -2i - j) \rangle = \begin{cases} 1 & \Leftrightarrow k = 2i + j \\ 0 & \Leftrightarrow k \neq 2i + j \end{cases} \quad (3.31)$$

por lo que cuando  $i = 0$ , se anularán todos los términos en que  $k \neq j$  (segundo caso) y sólo quedarán los términos en que  $k = j$  con lo cual debe cumplirse que

$$\sum_k h_k^2 = 1 \quad (3.32)$$

En el caso de que  $i \neq 0$ ,  $j = k - 2i$  y debe cumplirse que

$$\sum_k h_k h_{k-2i} = 0 \quad (3.33)$$

Mediante un razonamiento análogo en base a la segunda propiedad se llega a que debe cumplirse que

$$\sum_k g_k^2 = 1 \quad (3.34)$$

y

$$\sum_k g_k g_{k-2i} = 0 \quad (3.35)$$

En cuanto a la propiedad referida a la ortogonalidad entre los espacios  $V_j$  y  $W_j$ , se puede deducir, análogamente, que

$$\langle \phi(\cdot), \psi(\cdot - i) \rangle = \left\langle \sqrt{2} \sum_k h_k \phi(2 \cdot -k), \sqrt{2} \sum_j g_j \phi(2 \cdot -2i - j) \right\rangle \quad (3.36)$$

$$= \sum_k h_k \sum_j g_j \langle \sqrt{2}\phi(2 \cdot -k), \sqrt{2}\phi(2 \cdot -2i - j) \rangle \quad (3.37)$$

donde

$$\langle \sqrt{2}\phi(2 \cdot -k), \sqrt{2}\phi(2 \cdot -2i - j) \rangle = \begin{cases} 1 & \Leftrightarrow k = 2i + j \\ 0 & \Leftrightarrow k \neq 2i + j \end{cases} \quad (3.38)$$

Cuando  $k \neq 2i + j$  es trivial, pues anula todos los términos, por lo que resta ver qué pasa en el caso en que  $k = 2i + j$ .

$$= \sum_k h_k \sum_j g_j \langle \sqrt{2}\phi(2 \cdot -k), \sqrt{2}\phi(2 \cdot -2i - j) \rangle \quad (3.39)$$

$$= \sum_k h_k g_{k-2i} \quad (3.40)$$

por lo que se deduce que la condición que deben cumplir  $h$  y  $g$  para que la transformada sea ortogonal es que:

$$\sum_k h_k h_{k-2i} = \begin{cases} 1 & \Leftrightarrow i = 0 \\ 0 & \Leftrightarrow i \neq 0 \end{cases} \quad (3.41)$$

$$\forall i \in Z : \sum_k h_k g_{k-2i} = 0 \quad (3.42)$$

### 3.5.1. Forma matricial

Se puede expresar la transformada wavelet como el producto de un vector (señal unidimensional de entrada) por una matriz construida en base a los filtros  $h'$  y  $g'$  (ver ecuación 3.22). Para esto, se pueden separar las dos etapas de la aplicación de un paso de la transformada. En primer lugar, la convolución (tanto con  $h'$  como con  $g'$ ) y en segundo lugar, el submuestreo.

Sea una señal de entrada ( $c$ ) de longitud  $N$  y los filtros  $h'$  y  $g'$  con  $r$  componentes. Si se considera la señal a transformar como periódica para evitar problemas en los bordes al realizar la convolución, se puede expresar como

$$c_0 \star h' = \begin{bmatrix} h'_0 & 0 & 0 & \dots & 0 & h'_{r-1} & h'_{r-2} & \dots & h'_1 \\ h'_1 & h'_0 & 0 & \dots & 0 & 0 & h'_{r-1} & \dots & h'_2 \\ \vdots & & & \ddots & & & & & \\ 0 & 0 & 0 & \dots & h'_{r-1} & h'_{r-2} & h'_{r-3} & \dots & h'_0 \end{bmatrix} \cdot \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ \vdots \\ c_{0,N-1} \end{bmatrix} \quad (3.43)$$

donde la matriz asociada a la convolución es cuadrada y tiene la misma cantidad de filas y columnas que las componentes de la señal de entrada<sup>1</sup>.

Revisando el gráfico del algoritmo cascada se puede observar que falta la segunda parte del proceso; el submuestreo. Para esto se pueden descartar los coeficientes que correspondan una vez que se haya hecho la multiplicación por la matriz o, lo que es mejor aún, descartar las filas relacionadas con los componentes impares, que son los que serían descartados. Incluyendo el submuestreo, la matriz quedaría de la siguiente manera:

$$(c_0 \star h') \downarrow 2 = \begin{bmatrix} h'_0 & 0 & 0 & \dots & h'_{r-1} & h'_{r-2} & h'_{r-3} & \dots & h'_1 \\ h'_2 & h'_1 & h'_0 & \dots & 0 & 0 & h'_{r-1} & \dots & h'_3 \\ \vdots & & & \ddots & & & & & \\ 0 & 0 & 0 & \dots & h'_{r-1} & h'_{r-2} & h'_{r-3} & \dots & h'_0 \end{bmatrix} \cdot \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ \vdots \\ c_{0,N-1} \end{bmatrix} \quad (3.44)$$

donde la matriz tiene sólo las filas pares de la matriz anterior.

Esto resuelve sólo el cálculo de los valores asociados a la aproximación del nivel inferior. Para calcular los valores de detalle se debe realizar otra multiplicación similar pero con una matriz construida en base al vector  $g$ . El resultado será otra matriz de  $\frac{N}{2}$  filas por  $N$  columnas.

<sup>1</sup>La cantidad de valores resultantes de la convolución es igual al de la entrada debido a la periodización.

$$(c_0 * g') \downarrow 2 = \begin{bmatrix} g'_0 & 0 & 0 & \cdots & g'_{r-1} & g'_{r-2} & g'_{r-3} & \cdots & g'_1 \\ g'_2 & g'_1 & g'_0 & \cdots & 0 & 0 & g'_{r-1} & \cdots & g'_3 \\ \vdots & & & \ddots & & & & & \\ 0 & 0 & 0 & \cdots & g'_{r-1} & g'_{r-2} & g'_{r-3} & \cdots & g'_0 \end{bmatrix} \cdot \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ \vdots \\ c_{0,N-1} \end{bmatrix} \quad (3.45)$$

El paso completo de la transformada se puede expresar de la siguiente manera:

$$\begin{bmatrix} c_{-1,0} \\ c_{-1,1} \\ \vdots \\ c_{-1,\frac{N}{2}-1} \\ d_{-1,0} \\ d_{-1,1} \\ \vdots \\ d_{-1,\frac{N}{2}-1} \end{bmatrix} = \begin{bmatrix} h'_0 & 0 & 0 & \cdots & h'_{r-1} & h'_{r-2} & h'_{r-3} & \cdots & h'_1 \\ h'_2 & h'_1 & h'_0 & \cdots & 0 & 0 & h'_{r-1} & \cdots & h'_3 \\ \vdots & & & \ddots & & & & & \\ 0 & 0 & 0 & \cdots & h'_{r-1} & h'_{r-2} & h'_{r-3} & \cdots & h'_0 \\ g'_0 & 0 & 0 & \cdots & g'_{r-1} & g'_{r-2} & g'_{r-3} & \cdots & g'_1 \\ g'_2 & g'_1 & g'_0 & \cdots & 0 & 0 & g'_{r-1} & \cdots & g'_3 \\ \vdots & & & \ddots & & & & & \\ 0 & 0 & 0 & \cdots & g'_{r-1} & g'_{r-2} & g'_{r-3} & \cdots & g'_0 \end{bmatrix} \cdot \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ c_{0,N-1} \end{bmatrix} \quad (3.46)$$

donde los coeficientes  $c$  son los correspondientes a la aproximación de ese nivel y los  $d$  los del detalle. Se llamará  $W$  a esta matriz cuadrada.

Retomando las propiedades que se le piden a los filtros  $h$  y  $g$  en las ecuaciones 3.41 y 3.42, puede verse que es exactamente pedir que

$$W W^T = I \quad (3.47)$$

Por lo tanto,

$$W^T = W^{-1} \quad (3.48)$$

y la norma 2 de  $W$  es 1 [Str89].

De esta manera, se puede asegurar que la norma 2 se mantendrá constante al aplicar la (anti)transformada. Por lo tanto, el error de cuantización (en norma 2) será igual al error de la imagen reconstruida [Dau92], permitiendo conocer, luego de descartar información, la calidad final de la reconstrucción sin necesidad de antitransformar, ya que en el paso inverso se cumplirá exactamente la misma propiedad.

### 3.6. Multiwavelets

Hace algunos años comenzó a desarrollarse un agregado a la teoría de wavelets llamado *Multiwavelets*. A diferencia de las wavelets, mediante las multiwavelets se podía disponer de ortogonalidad, simetría y pequeño soporte en el mismo par de filtros.

El concepto de *análisis de multiresolución* se mantiene intacto. Sin embargo,  $V_0$  ya no es generado por una función  $\phi(t - k)$  y sus corrimientos sino por un conjunto de estas llamado  $\Phi(t)$ , donde [SHS<sup>+</sup>99]

$$\Phi(t) = [\phi_1(t), \dots, \phi_N(t)] \quad (3.49)$$

De manera análoga al caso escalar (wavelets), existe una matriz de ecuaciones de dilatación para la funciones de escala.

$$\Phi(t) = \sum_k C[k] \Phi(2t - k) \quad (3.50)$$

donde  $C[k]$  ya no es un escalar sino una matriz de  $N$  por  $N$ .

Para las funciones wavelet asociadas a las funciones de escala la matriz de ecuaciones es

$$\Psi(t) = \sum_k D[k] \Phi(2t - k) \quad (3.51)$$

donde  $\Psi(t) = [\psi_1(t), \dots, \psi_N(t)]$  es el vector de wavelets asociado a  $\Phi(t)$  y  $D[k]$  es una matriz de  $N$  por  $N$ .

Una de las primeras multiwavelets fue la llamada GHM. Contiene dos funciones de escala y dos de detalle. Los valores resultantes de las ecuaciones de dilatación y refinamiento son

$$\begin{aligned} C[0] &= \begin{bmatrix} \frac{3}{5} & \frac{4\sqrt{2}}{5} \\ -\frac{1}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix} & C[1] &= \begin{bmatrix} \frac{3}{5} & 0 \\ \frac{9}{10\sqrt{2}} & 1 \end{bmatrix} \\ C[2] &= \begin{bmatrix} 0 & 0 \\ \frac{9}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix} & C[3] &= \begin{bmatrix} 0 & 0 \\ -\frac{1}{10\sqrt{2}} & 0 \end{bmatrix} \end{aligned} \quad (3.52)$$

y

$$\begin{aligned} D[0] &= \frac{1}{10} \begin{bmatrix} -\frac{1}{\sqrt{2}} & -3 \\ 1 & 3\sqrt{2} \end{bmatrix} & D[1] &= \frac{1}{10} \begin{bmatrix} \frac{9}{\sqrt{2}} & -10 \\ -9 & 0 \end{bmatrix} \\ D[2] &= \frac{1}{10} \begin{bmatrix} \frac{9}{\sqrt{2}} & -3 \\ 9 & -3\sqrt{2} \end{bmatrix} & D[3] &= \frac{1}{10} \begin{bmatrix} -\frac{1}{\sqrt{2}} & 0 \\ -1 & 0 \end{bmatrix} \end{aligned} \quad (3.53)$$

Este sistema de funciones de escala es un ejemplo del potencial de la teoría de multiwavelets ya que combina simetría, ortogonalidad y una aproximación de segundo orden; algo imposible de realizar en un sistema escalar de una función de escala.

## 3.7. Procesamiento de señales

### 3.7.1. Señales unidimensionales

Sea  $f_0(x)$  una señal unidimensional original a procesar. Siguiendo la figura del algoritmo cascada de la página 27, se convoluciona la señal con el filtro  $h'$  y luego se la submuestra de a 2 para obtener la aproximación de nivel -1. Los filtros  $h'$  y  $g'$  están determinados por la wavelet que se utilice. En este caso,

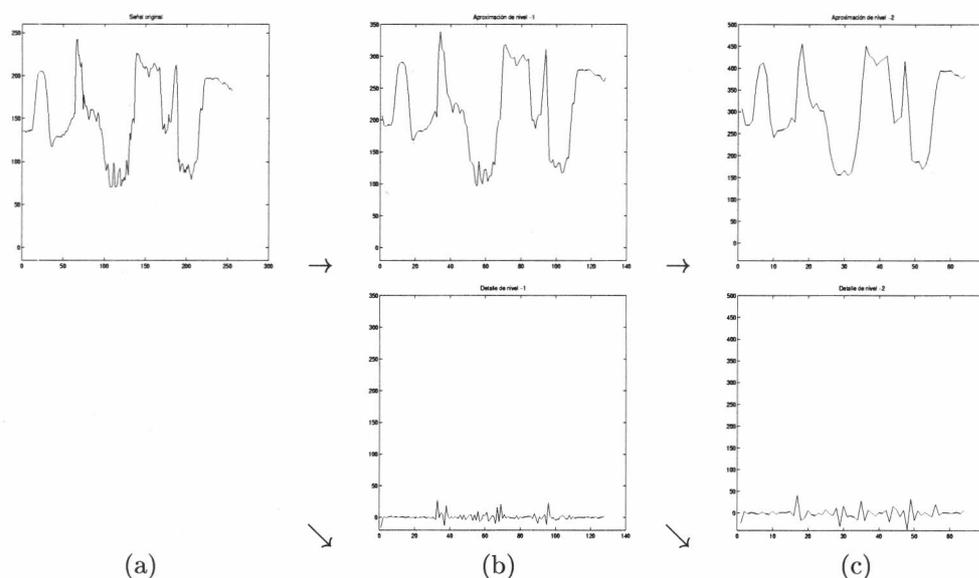


Figura 3.2: Aplicación de la transformada wavelet a una señal unidimensional. (a) Señal original. (b) Resultado del primer paso. (c) Resultado del segundo paso.

se procesará la señal con una *Symmlet8*. El detalle del mismo nivel se obtiene siguiendo los mismos pasos pero convolucionando con el vector  $g'$ .

La aproximación de nivel -1 puede observarse en el primer gráfico de la segunda columna de la figura 3.2. Bajo este puede observarse el gráfico correspondiente al detalle del mismo nivel.

Cada una de estas dos señales (aproximación y detalle) tienen exactamente la mitad de componentes que la señal original. Con estas se puede reconstruir sin pérdida la señal original.

Si se quisiera realizar otro paso de la transformada, basta con repetir las operaciones anteriores tomando como entrada a la aproximación de nivel -1. Luego de aplicarlo, se necesita conservar el resultado del último paso, así también como los detalles de todos los pasos anteriores. En este caso, los dos niveles de detalle y la aproximación de nivel -2.

En la tercer columna se puede observar el gráfico de la aproximación de nivel -2 y el del detalle.

### 3.7.2. Señales bidimensionales

Sea una imagen  $I$  a la que se le quiere aplicar una transformada wavelet. Como los filtros  $h$  y  $g$  son unidimensionales no servirán para filtrar una señal bidimensional. Es por esto que se utilizan para filtrarla por filas. Cada una de las filas de la señal es filtrada dando como resultado una aproximación y un detalle. Utilizando estas como filas de dos nuevas señales bidimensionales tendremos como resultado una señal de aproximación y una de detalles *verticales*. Este es considerado un estado intermedio del paso completo del algoritmo cascada en

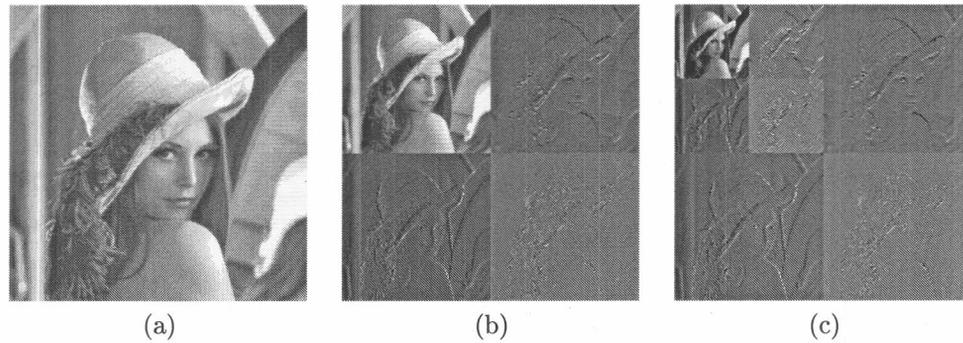


Figura 3.3: Aplicación de la transformada wavelet a una señal bidimensional. (a) Imagen original. (b) Con un paso de la transformada. (c) Con dos pasos de la transformada.

dos dimensiones.

Luego, se filtran y submuestran estas dos nuevas señales pero por columnas, produciendo, de cada una de ellas, dos nuevas señales bidimensionales. Del filtrado de la aproximación obtenida en primera instancia, se obtienen la aproximación definitiva y los detalles *horizontales*. Del filtrado de la señal con los detalles verticales, se obtienen la señal con los detalles verticales definitivos y la señal con los detalles diagonales.

En el punto (b) de la figura 3.3, pueden observarse las cuatro señales resultantes de la aplicación de este primer paso del algoritmo cascada en señales bidimensionales.

Para la reconstrucción de la señal original son necesarias estas cuatro señales, las cuales tienen exactamente la mitad de las componentes por lado que la señal original.

En caso de querer aplicar otro paso de la transformada, basta con repetir los pasos para la aproximación obtenida. Luego, para reconstruir la señal original, serán necesarios *todos* los detalles (verticales, horizontales y diagonales) y la aproximación obtenida en el último paso.

En el punto (c) puede observarse el resultado de la aplicación de dos pasos sucesivos de la transformada wavelet.

## Capítulo 4

# Estudio de las bandas

Se denomina *banda de detalle*, o simplemente *banda*, a la señal de detalle resultante de la aplicación de un paso de la transformada. De esta manera, la primer banda será la que resulte como detalle del primer paso en sus 3 orientaciones: vertical, horizontal y diagonal. La segunda, será el detalle del segundo paso aplicado sobre la aproximación devuelta por el primero y así sucesivamente. Las bandas de detalle resultantes de los primeros pasos se denominan, también, de detalle fino y las de los últimos pasos de detalle grueso. Esto está relacionado con la amplitud de la zona de influencia de los componentes de las bandas. Mientras que un componente de la primer banda tiene una zona de influencia asociada de  $r$  por  $r$  (siendo  $r$  el tamaño del filtro), a medida que se calcula esta área en las bandas interiores (más cercanas a la aproximación) esta zona se vuelve cada vez más amplia. En la figura 4.1 se puede ver cómo se distribuyen las bandas después de dos pasos de la transformada wavelet. La primer banda consta de  $D_{-1_h}$ ,  $D_{-1_v}$  y  $D_{-1_d}$ , mientras que la segunda está formada por  $D_{-2_h}$ ,  $D_{-2_v}$  y  $D_{-2_d}$ .

### 4.1. Umbral sobre transformadas

Hasta ahora las pruebas, por una cuestión de simplicidad, se habían limitado a la aplicación de un umbral dado a la señal transformada. Es, ni más ni menos, que una forma de compresión muy sencilla y poco elaborada pero no por eso ineficiente. Esto no implicaba ninguna diferenciación (por lo menos explícita) entre las bandas que componían la transformada. Sin embargo, existe una diferenciación implícita entre los valores de las distintas bandas.

Al aplicar un paso de la transformada a una señal, el rango de valores resultantes puede ampliarse tanto en la aproximación como en el detalle. Los distintos niveles de detalle, así como la aproximación, que se obtienen mediante la aplicación del algoritmo cascada, son el resultado de filtrar la señal original con los vectores  $h$  y  $g$ . Por ejemplo, el detalle de nivel  $-t$  se obtiene filtrando  $t - 1$  veces (y submuestreando de a 2) la señal con el vector  $h$  y la última vez con el vector  $g$ . El vector  $h$  debe cumplir una propiedad por la cual

$$\sum h_k = \sqrt{2} \quad (4.1)$$

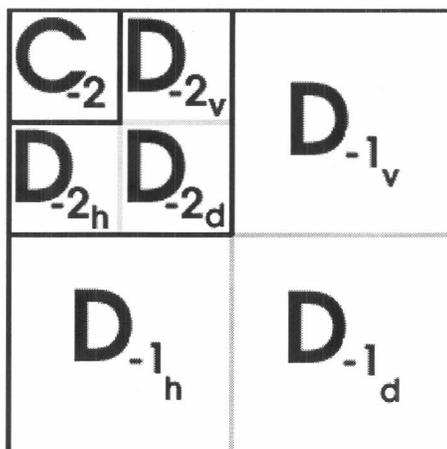


Figura 4.1: Ubicación de las bandas de detalle en señales bidimensionales luego de dos pasos de la transformada wavelet

Para ejemplificarlo, si la señal  $A$  cumple que  $\forall i : 0 \leq i < N : A_i = 1$ , el resultado de la convolución con  $h$  ( $A \star h$ ) será

$$\forall k : 0 \leq k < N : (A \star h)_k = \sqrt{2}$$

Es por esto que los valores que se encuentren en las bandas de detalle más grueso tendrán generalmente valores más altos por haber sido objeto de varios pasos de la transformada. Consecuentemente, la aplicación de un umbral dejará mayor cantidad de ceros en las bandas de detalle más fino e intactos los valores de aproximación y detalle más grueso.

Esto puede observarse con claridad en la figura 4.2 en donde se grafican los mapas de significancia ([Mal99] - pag. 551) para tres umbrales distintos aplicados a la misma transformada. El mapa de significancia consta de una señal binaria ( $b[i, j]$ ) del mismo tamaño que la original, de manera que

$$b[i, j] = \begin{cases} 0 & \text{si } A[i, j] = 0 \\ 1 & \text{si } A[i, j] \neq 0 \end{cases}$$

Si algún valor de la primer banda sobrevive a un umbral considerable, quiere decir que en su area de influencia asociada la señal tiene un salto de magnitud que, muy probablemente, no se quiera perder. Podría ser, por ejemplo, el borde de alguna figura. Se sabe esto pues el desnivel debe estar dentro de esta area, la cual queda determinada por su pertenencia a la primer banda y su posición dentro de ella. Esta area es la más compacta de todas y abarca una zona muy reducida. Los valores más pequeños indicarán, en general, *degradeés* o pequeños cambios de tonalidad.

Si se encontrara el mismo valor en la última banda de detalle (obviamente es otro sobreviviente) probablemente quiera decir que la señal tenía un cierto desnivel, pero no entre puntos sucesivos, sino entre puntos cuya lejanía es directamente proporcional a la cantidad de bandas de detalle (por ser la última). Esto está relacionado, nuevamente, con el area de influencia asociada a

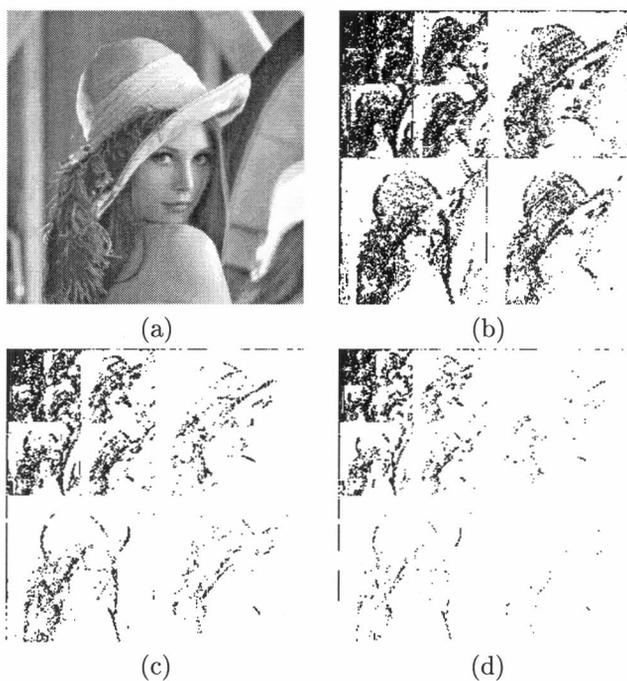


Figura 4.2: Resultado de la aplicación de umbrales a la transformada de la imagen original. (a) imagen original. Mapas de significancia luego de aplicar un umbral de: (b) 25% de coeficientes intactos; (c) 10% de coeficientes intactos; (d) 5% de coeficientes intactos.

---

este valor. Por ser un valor ubicado en la última banda de detalle, cuya función generadora del espacio fue escalada varias veces, el área será mucho mayor a la anteriormente descrita, pudiendo llegar incluso a abarcar la imagen entera. Estos valores son los que representan las formas de la imagen. Por lo tanto, el mismo valor cobra mayor o menor importancia según el lugar (banda) en que se lo encuentre. Obviamente, se quiere poder reconstruir las formas o figuras de la imagen original y, de ser necesario perder algo, se elegirán los pequeños detalles que muchas veces son imperceptibles a primera vista y sin una revisión más profunda.

Esto da la pauta de que a pesar de lo rudimentario de poner a cero todo lo que está por debajo de un cierto valor, si lo combinamos con la tendencia a una cierta distribución de valores, resultante de la aplicación de la transformada wavelet, la aplicación de un umbral es una herramienta de compresión bastante poderosa y no por eso costosa en cuanto a implementación.

Observando la figura 4.3 se puede apreciar la mala performance del umbral cuando se lo aplica a una imagen sin previo procesamiento. Por el contrario, cuando a la imagen se le ha aplicado una transformada wavelet, el rendimiento del umbral es infinitamente mayor, permitiendo bajar la entropía sin perder calidad en la imagen de resultado. Ya en la última medición del umbral sobre la imagen original, en la que sólo se pierde el 1 % de los coeficientes, el PSNR llega a bajar aproximadamente a 31 mientras que en la transformada se mantiene sobre los 107. Es interesante notar que cualquier imagen recuperada que tenga un PSNR superior a los 40 dB puede considerarse de una excelente calidad. En las mediciones del umbral sobre la transformada wavelet no se baja de este valor hasta luego de haber descartado el 85 % de los coeficientes aproximadamente.

En cuanto a la entropía, fue calculada tomando sólo dos decimales de precisión en las componentes, las cuales son en realidad números reales de precisión arbitraria.

## 4.2. Cuantización sobre transformadas

Pero, ¿qué pasa si se trata de dar un paso más después de aplicar el umbral? Como resultado se tiene un conjunto de valores en cero y el resto idénticos a como los había dejado la transformada. Se ha resuelto sólo una parte del problema de decidir qué hacer con cada valor. Sólo lo que concierne a valores pequeños. Esto tiene una limitación bastante notoria. Por más que se haya dicho que la aplicación de un umbral puede ser muy beneficiosa en la transformada wavelet, basándose en las propiedades antes mencionadas, no se podrá obtener una muy buena compresión sin deteriorar la calidad visual sólo descartando los valores más pequeños que generalmente se encuentran en las bandas de detalle más fino.

Por otro lado, tal como se mencionó en la sección anterior, las componentes son números reales con tantos decimales como sean necesarios para asegurar una reconstrucción perfecta. Esto dificulta su codificación mediante un número finito de bits ([Mal99] pag. 537; [Wal91] pag. 4).

Para salvar este punto es que se introduce la cuantización. Esta permitirá que se sigan aprovechando todos los resultados obtenidos hasta ahora, ya que el rango de valores que va entre 0 (cero) y  $q$  (ver ecuación 2.5) serán transformados en cero. Por otro lado, los valores mayores a  $q$  también serán transformados en

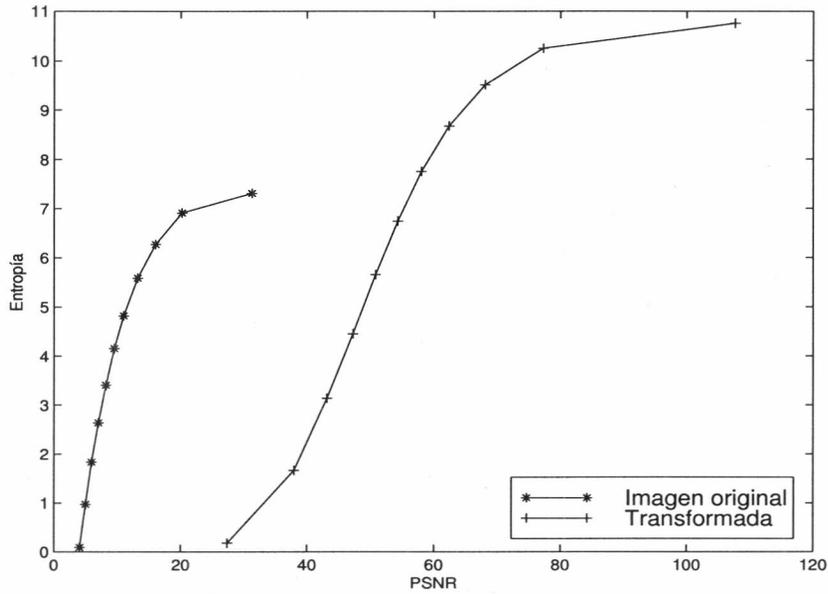


Figura 4.3: Comparación del umbral aplicado en una imagen y su transformada wavelet. Las mediciones corresponden al porcentaje de coeficientes que permanecen intactos (1 %, 10 %, 20 %, 30 %, 40 %, 50 %, 60 %, 70 %, 80 %, 90 %, 99 %).

% de coef. intactos	Original		Transformada	
	PSNR	Entropía <sup>1</sup>	PSNR	Entropía
1	4.05	0.08	27.36	0.17
10	4.98	0.97	37.92	1.66
20	5.94	1.83	43.16	3.13
30	7.01	2.63	47.25	4.45
40	8.18	3.40	50.84	5.65
50	9.52	4.15	54.32	6.74
60	11.00	4.81	58.05	7.75
70	13.21	5.58	62.39	8.67
80	16.04	6.27	68.06	9.51
90	20.23	6.91	77.29	10.24
99	31.27	7.31	107.65	10.75

Cuadro 4.1: Detalle de los valores correspondientes a la figura 4.3.

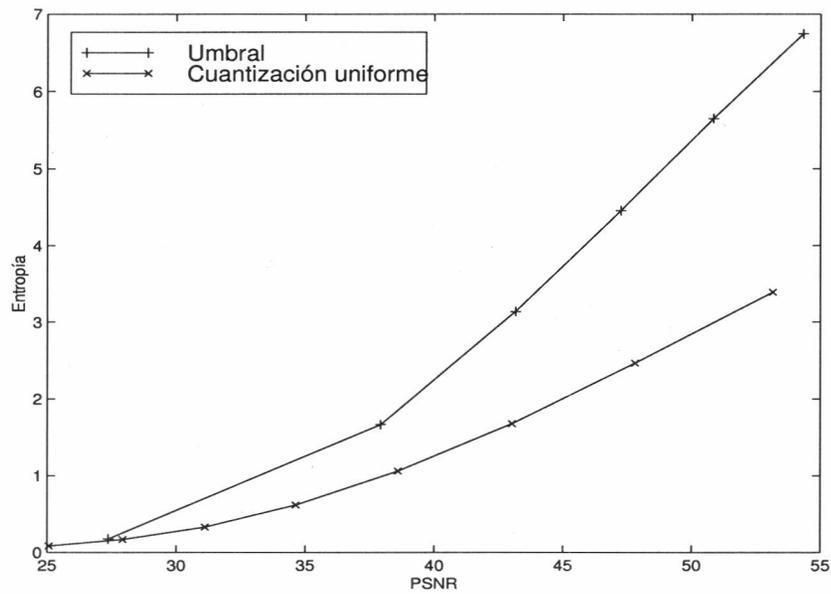


Figura 4.4: Comparación del desempeño del umbral y la cuantización uniforme en una transformada wavelet.

Umbral			Cuantización		
Zona de Cero	PSNR	Entropía	Zona de Cero	PSNR	Entropía
100.02	27.36	0.17	128	25.03	0.08
12.47	37.92	1.66	64	27.89	0.16
5.65	43.16	3.13	32	31.08	0.33
3.26	47.25	4.45	16	34.60	0.61
2.06	50.84	5.65	8	38.56	1.06
1.40	54.32	6.74	4	42.99	1.68
			2	47.77	2.46
			1	53.12	3.38

Cuadro 4.2: Detalle de los valores correspondientes a la figura 4.4.

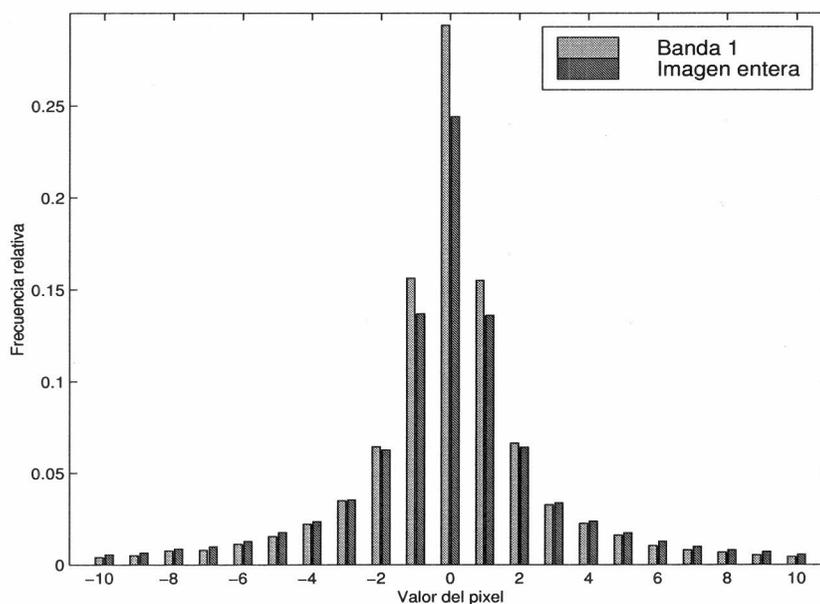


Figura 4.5: Comparación de los histogramas correspondientes a la primer banda y a la imagen transformada total.

el representante que les corresponda. Aunque  $q$  sea más pequeño que el umbral y los otros valores no sean enviados a cero sino a otro representante, en general no se tendría por qué esperar un crecimiento en la entropía. Es fácil ver que no se gana más por enviar un valor a cero que si se lo envía a otra constante. A los fines del cálculo de la entropía el resultado es el mismo. La entropía está dada por la frecuencia relativa de los valores y no por su valor en sí. *Claro que a mayor número de intervalos, mayor será el valor de la entropía.* Sin embargo, para un número relativamente acotado de intervalos, aunque el primer intervalo resulte pequeño y, por lo tanto, disminuya su frecuencia relativa en comparación a la técnica del umbral, se recuperará (y con creces) al cuantizar también *todos* los valores que el umbral dejaba intactos y eran los responsables de no poder achicar más la entropía. En la figura 4.4 puede verse cómo el umbral es superado por la cuantización uniforme tanto en calidad como en compresión.

#### 4.2.1. Cuantización por banda vs. cuantización para toda la señal

Cuando se aplicaba el umbral se lo hacía sin analizar para cada coeficiente si pertenecía a una determinada banda o a otra. Esto podía realizarse con éxito, sin embargo, debido a los rangos de valores en que cada banda se movía.

En el caso de la cuantización, se ve claramente que a mayor número de intervalos, menor será el error con el que reconstruya la señal. Existe un hecho que favorecería el aplicar la cuantización a *toda* la imagen (sin considerar cada banda de detalle por separado) aprovechando esta característica y es, otra vez, los

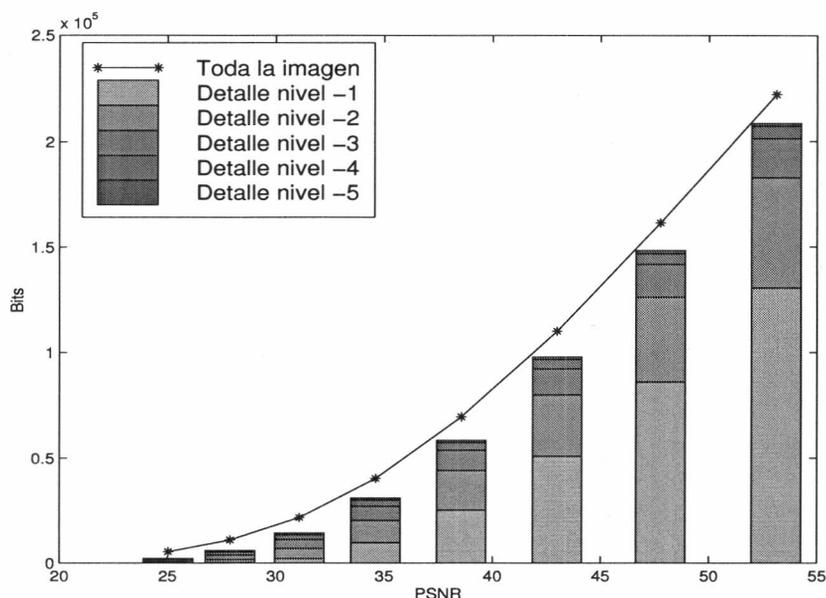


Figura 4.6: Comparación del tamaño estimado de la imagen codificada según si se codifica la imagen como un todo o se codifica por banda. Las barras corresponden a los siguientes  $q$ : 256, 128, 64, 32, 16, 8, 4, 2.

rangos asociados a cada banda. De la definición misma de las bandas de detalle surge intuitivamente la idea de tener una cantidad distinta de representantes por banda. Particularmente, se tiende a considerar a las bandas interiores (detalle más grueso) como las más importantes. Estas poseen rangos más amplios y, por lo tanto, la técnica de cuantización estaría usando más intervalos en ellas. Este efecto es ampliamente deseado ya que la idea es justamente preservar lo más que se pueda estas bandas. Sin embargo, implica una rigidez en el proceso de cuantización de la imagen el cual podría ser mejorado.

Para ejemplificarlo, se propone retomar la cuantización de una imagen transformada, independientemente de la cantidad de bandas, para distintos  $q$  que dio por resultado la primer curva que puede observarse en la figura 4.4. Luego de cada una de las cuantizaciones, se midió la Entropía alcanzada y el PSNR. ¿Pero qué pasa si se observa el histograma de la primer banda y se lo compara con el de la imagen entera (ver figura 4.5)? Se puede apreciar que la primer banda tiene una frecuencia relativa muy alta en las inmediaciones del cero. También el histograma de la imagen total. Sin embargo, en el histograma de toda la imagen la frecuencia relativa tan alta que existía cerca del cero se ve disminuida por la inclusión de los valores de las otras bandas, los cuales tienden a crecer a medida que fueron parte de más pasos de la transformada. Si se observa detenidamente, se verá que las frecuencias de los valores más grandes (en valor absoluto) se ven incrementadas en perjuicio de las cercanas al cero. Obviamente, este proceso de lenta *uniformización* del histograma, a medida que se incluyen más bandas, da como resultado un aumento de la entropía.

Esto puede llevar a recalcular la entropía, pero no ya sobre toda la imagen, sino sobre sus bandas por separado. Luego, se pondera entropía - cantidad de bits por símbolo estimada - por la cantidad de componentes de la banda, para lograr así, una estimación de cuánto podría llegar a ocupar la codificación de cada banda. En la figura 4.6 pueden verse, como barras apiladas, la cantidad de bits estimados para la codificación de cada una de las bandas. La curva que se ve por sobre las barras es la resultante de considerar a la transformada como un todo. Debe recordarse que la forma de cuantizar la transformada no cambió (por lo menos para este experimento), con lo cual el  $q$  usado es el mismo y el PSNR resultante también. Sin embargo, la diferencia entre la curva y las barras apiladas indica que hay una ganancia en cuanto a la cantidad de bits que podría llegar a ocupar la imagen codificada si se lleva a cabo esta codificación pero por bandas. Es cierto que no es una ganancia sustancial, pero abre un nuevo abanico de posibilidades.

En definitiva, se utiliza el mismo paso de cuantización ( $q$ ), se cuantiza con la misma técnica (uniforme) y se obtiene el mismo resultado en calidad (PSNR). Sin embargo, el hecho de codificar a cada banda por separado da como resultado que el tamaño total de la imagen codificada sea menor. Lo que se está aprovechando es el hecho de saber que en las bandas de detalle más fino se encontrarán la mayor cantidad de ceros por lo que la entropía en las mismas se reducirá con respecto a la imagen total.

La verdadera ganancia surge de considerar a cada banda como una entidad con características propias las cuales necesitan ser reconocidas para poder llevar a cabo una codificación más eficiente.

A partir de la cuantización por bandas de la transformada, el paso de cuantización no tiene por qué ser el mismo en todas las bandas. Ya se demostró que “incluso” cuantizando con el mismo paso en todas las bandas, el tamaño de la señal codificada podía reducirse. A partir de este cambio podemos utilizarlo para darle mayor o menor importancia a cada banda en base a algún tipo de análisis. Por esta misma razón es que es preferible procesar las bandas por separado, tratando de reducir el MSE y los BPP. Cada banda tiene particularidades en su distribución (aunque a veces sean similares) que merecen ser aprovechadas. La concentración de valores muy cercana al cero en la primer banda de detalle (como suele ocurrir generalmente), por ejemplo, no tendría por qué verse afectada por los valores exageradamente altos que puedan encontrarse en la última banda, ya que esto elevará la entropía.

En la página 43 se pueden observar los histogramas resultantes del análisis por bandas graficados en una misma escala y en la página 44 se los grafica con distinta escala de manera de poder observar con mayor nivel de detalle la distribución de cada uno.

### 4.2.2. Cuantización uniforme vs. Max-Lloyd

Intuitivamente (y no tanto), se da por sentado que en casi todas las aplicaciones prácticas la cuantización Max-Lloyd será mejor que la uniforme. Esta afirmación está inducida por el mejor desempeño que muestra Max-Lloyd en cuanto a calidad en imágenes de tonos de gris. La diferencia es que las señales que se quieren cuantizar no son de tonos de gris sino un conjunto de bandas de transformadas wavelet con características bien diferenciadas. Por ejemplo, las bandas de detalle más fino van a tener coeficientes cercanos al cero y algunos

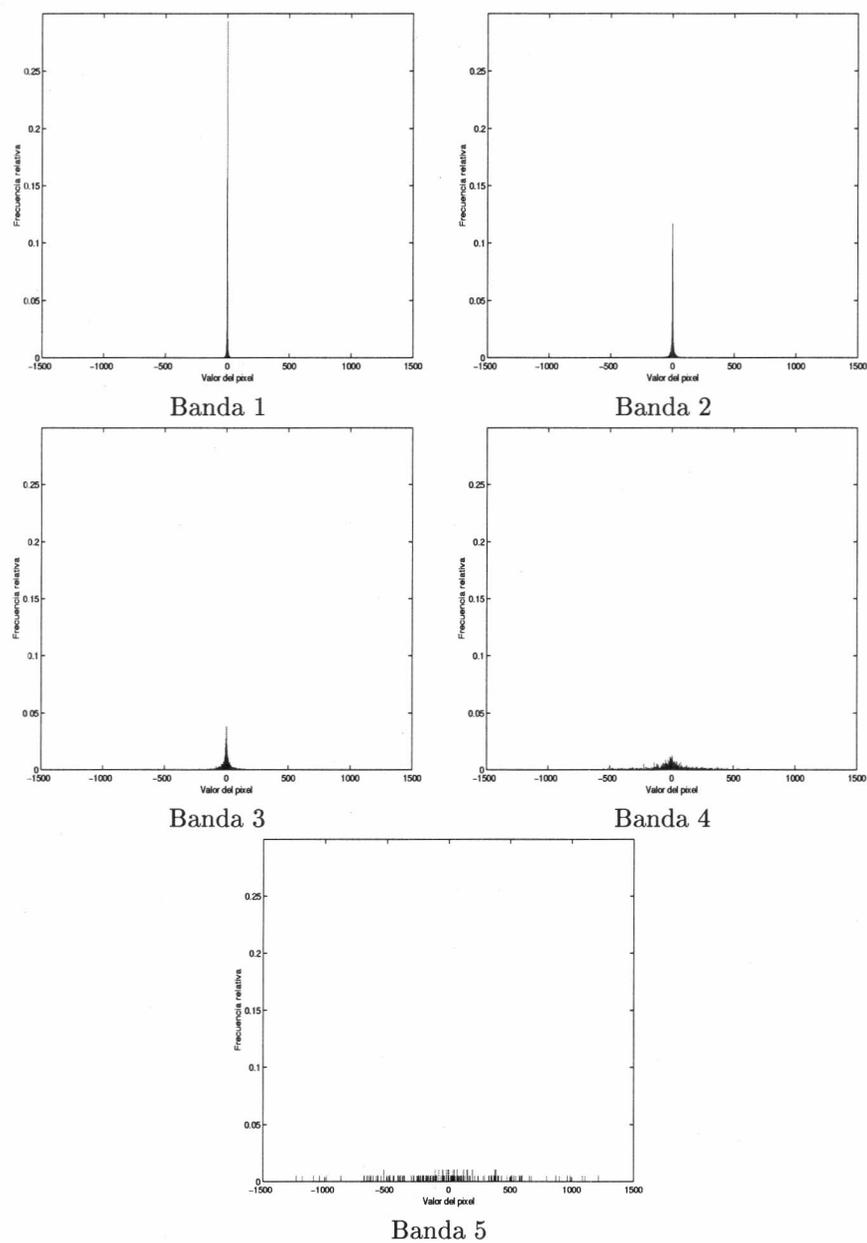


Figura 4.7: Histogramas correspondientes a las distintas Bandas de Detalle con idéntica escala para todos.

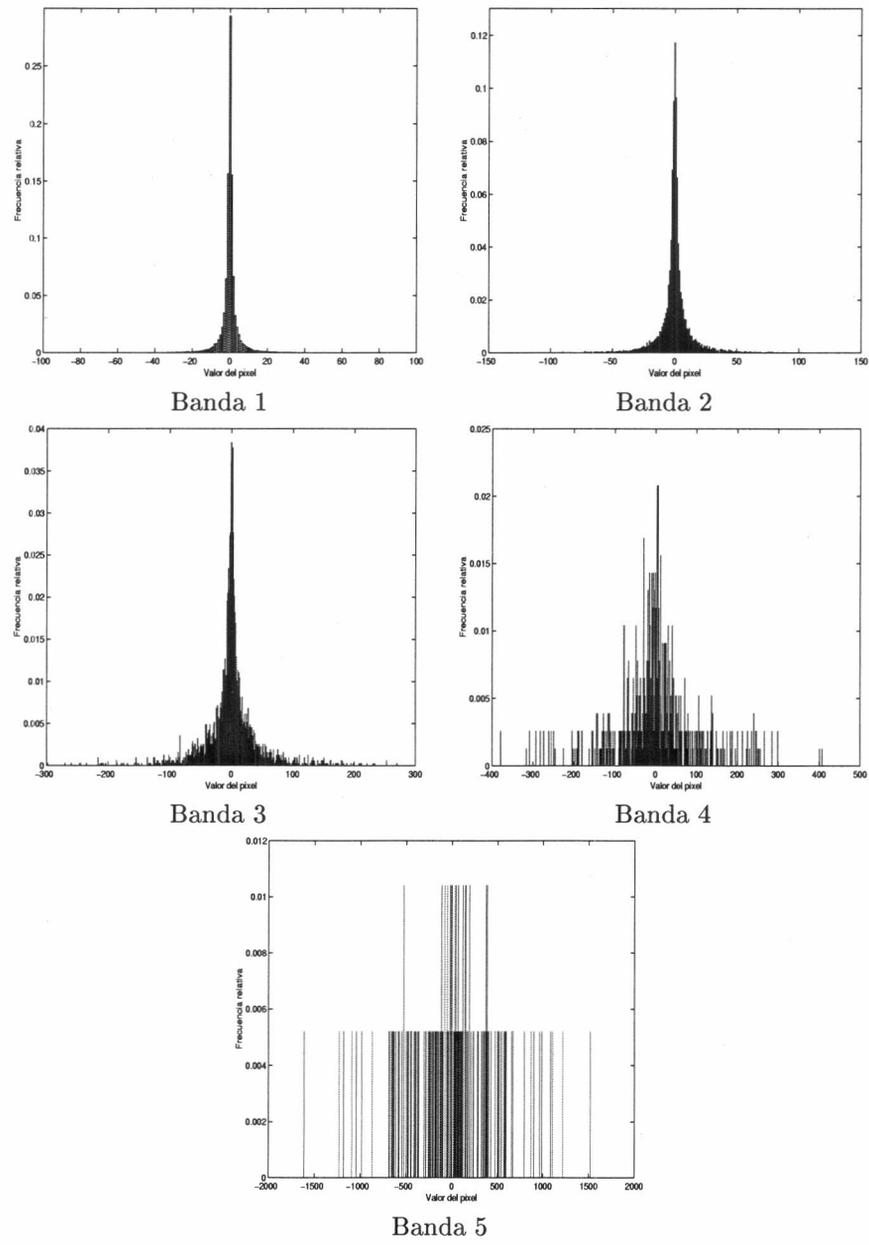


Figura 4.8: Histogramas correspondientes a las distintas Bandas de Detalle.

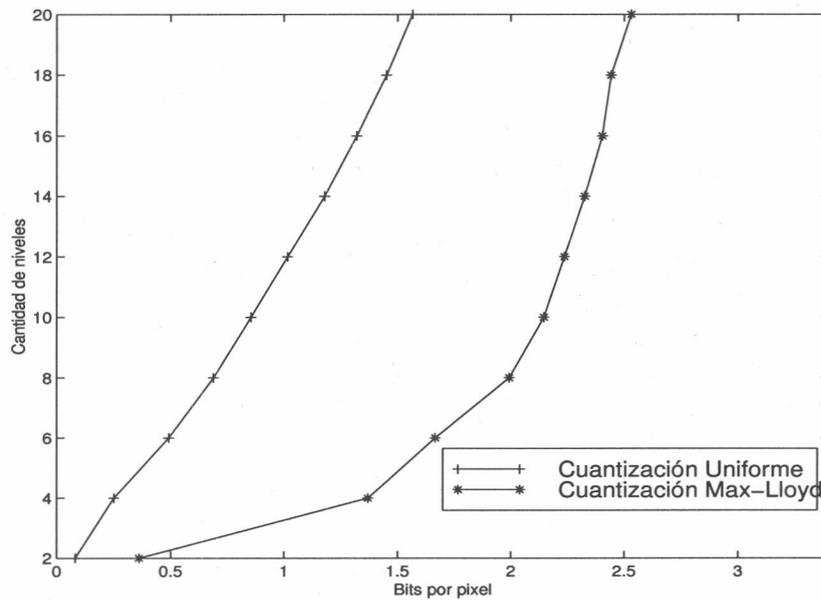


Figura 4.9: Comparación de la entropía resultante entre cuantización uniforme y Max-Lloyd (Imagen Woman, Wavelet Sym8)

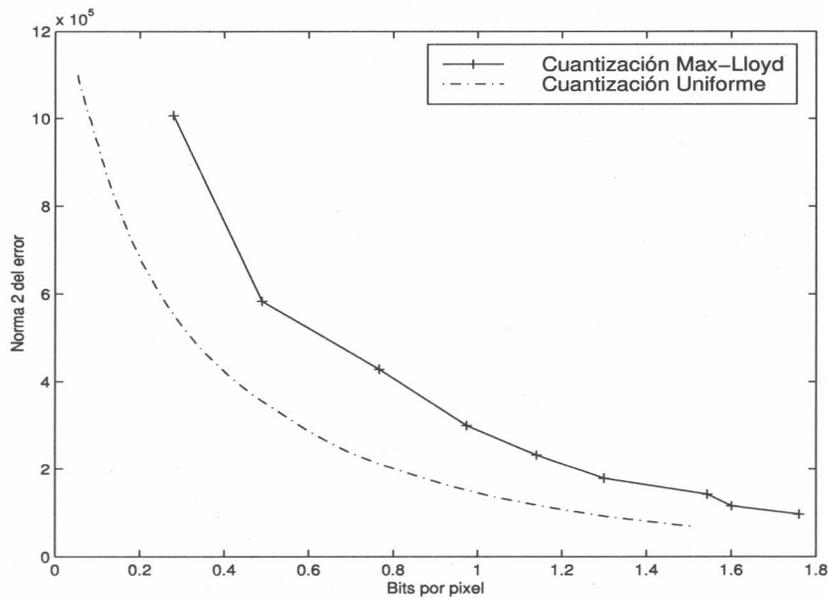


Figura 4.10: Comparación de la cuantización Max-Lloyd y Uniforme para la primera banda de una transformada (Imagen Lena, Wavelet Sym8)

pocos coeficientes más grandes en valor absoluto, que son los que indican la ubicación de saltos, discontinuidades, etc. y le darán a la imagen una apariencia más nítida (ver histograma en página 44). ¿Son tan importantes los valores cercanos al cero como para que sean diferenciados con mucha precisión entre ellos? Si los valores son tan pequeños, también será pequeño su aporte a la calidad de la imagen, sobre todo si pertenecen a las bandas de detalle más fino.

Debe recordarse que Max-Lloyd asigna un mayor número de intervalos a los rangos de valores donde se supone que la información está más concentrada (hay mayor cantidad de apariciones). Para dar sólo un ejemplo, en las bandas de detalle fino, los valores tienden a moverse muy cerca del cero y es justamente el rango de valores que son menos importantes. Los valores más importantes serán los que tomen valores más grandes que lo habitual. Ya se había señalado que en general, a igual cantidad de intervalos, Max-Lloyd suele necesitar una mayor cantidad de BPP que la cuantización uniforme. Este efecto se potencia cuando la distribución de lo que se va a cuantizar está tan concentrada en un pequeño rango de valores. En la figura 4.9 se puede observar la gran diferencia entre la entropía resultante de Max-Lloyd y la cuantización uniforme.

Es por esto que se prefiere usar la cuantización uniforme antes que la de Max-Lloyd. La cuantización uniforme va a llevar a cero todos los coeficientes cuyo valor absoluto sea pequeño<sup>2</sup> y va a cuantizar a todos los otros valores por igual. Por lo tanto, un alto porcentaje de valores (los menos importantes) quedarán en un mismo intervalo y el resto podrá ser cuantizado con una precisión equivalente al paso ( $q$ ). En cambio, Max-Lloyd cuantizará a los coeficientes con un pequeño valor absoluto en varios intervalos por lo que luego no podrá asignar la suficiente precisión a los coeficientes realmente importantes.

En la figura 4.10 se grafican las curvas de desempeño de ambos métodos para la cuantización de la primera banda de detalle. Puede observarse que la cuantización uniforme consigue, para una determinada calidad, siempre una codificación en menos BPP que Max-Lloyd; y para una cantidad fija de BPP consigue una imagen resultante de mejor calidad.

### 4.3. Influencia de cada banda

Una vez decidida la técnica utilizada para cuantizar, es fundamental conocer cuánto puede brindar cada banda, tanto a la calidad de imagen como a su compresión. Para esto se realizaron una serie de mediciones en la banda de detalle más fino, en las que se puede observar en base al aumento o disminución de la cantidad de representantes (intervalos) en la cuantización, la relación entre los bits estimados que ocuparía la banda codificada y el error resultante en la imagen comprimida (ver figura 4.11).

La entropía  $H(k)$  de la banda de detalle  $k$  es una estimación de los bits por coeficiente (o bits por pixel) que la banda ocupará en la imagen comprimida. Los valores  $y_p^{(k)}$  son los coeficientes cuantizados de la  $k$ -ésima banda de detalle  $Q^{(k)}(x_{ij}^{(k)})$ , sin repeticiones ( $y_p^{(k)} \neq y_{p'}^{(k)} : \forall p : p \neq p'$ ) y  $p(y_p^{(k)})$  es la frecuencia de aparición [Sha48].

$$H(k) = - \sum_p p(y_p^{(k)}) \cdot \log_2(p(y_p^{(k)})) \quad (4.2)$$

<sup>2</sup>Los coeficientes que se encuentren entre  $q$  y  $-q$ .

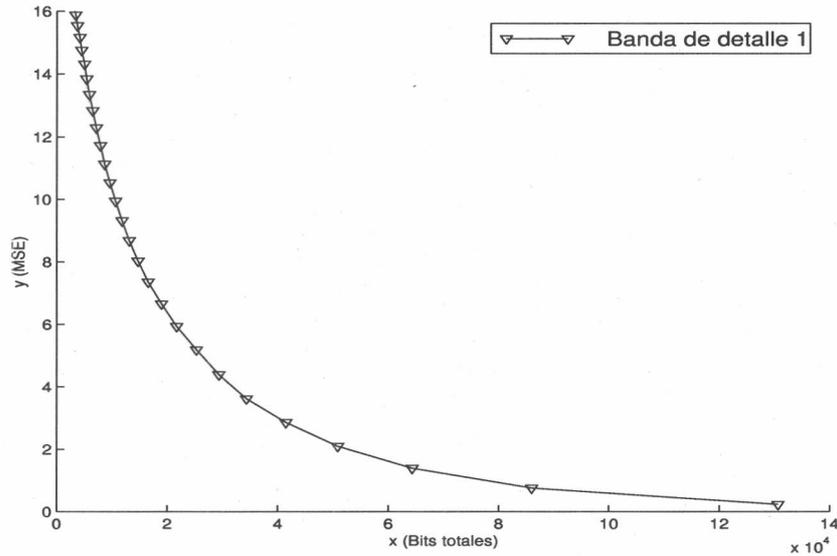


Figura 4.11: Relación entre compresión (Bits totales) y error (MSE) para la banda de detalle más fino.

El número total de bits que cada banda ocupará finalmente en la imagen comprimida ( $B(k)$ ) puede ser estimado multiplicando la entropía de la banda por el número de coeficientes de la misma.

$$B(k) = H(k) \frac{3}{4^k} N^2 \quad (4.3)$$

Si se grafican juntas las curvas asociadas a cada una de las bandas de detalle (ver figura 4.12), se puede observar que las bandas de detalles más finos tienen una pendiente mucho más suave a medida que aumenta la cantidad de bits totales. Esto indica (sobre todo en la primera banda) que se podrá disminuir drásticamente el número de representantes sin que eso resulte en una gran pérdida de calidad en la imagen final. Por otro lado, también se observa que en la banda de detalle más gruesa la reducción del número de representantes se paga caro en cuanto a calidad y no aporta nada a la compresión. Esto resulta fácilmente explicable por la cantidad de coeficientes por la que está compuesta cada banda (un dato obvio pero importante).

En el cuadro 4.3 se pueden ver los valores. Debe notarse que cada vez que se reducen los representantes del proceso de cuantización se está reduciendo la entropía. Sin embargo, este hecho será mejor aprovechado en las bandas más densamente pobladas. Para la transformada de una imagen cualquiera con  $N^2$  pixels, la ganancia en bits (RB) en una banda  $k$  provendrá de la variación de la entropía en la banda ( $\Delta H(k)$ ) multiplicada por la cantidad de coeficientes pertenecientes a la banda en cuestión ( $B(k)$ ).

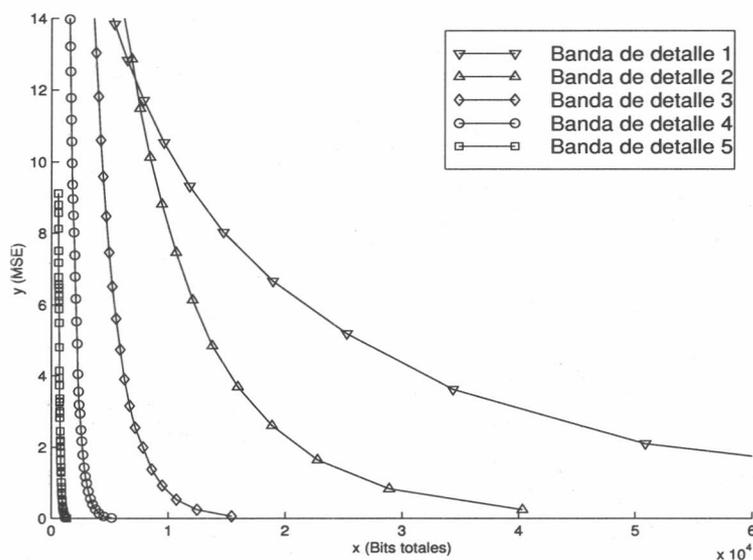


Figura 4.12: Relación entre compresión (Bits totales) y error (MSE) para cada una de las bandas de detalle de la transformada (Imagen Lena, wavelet Sym8)

Banda	Nivel	Coefficientes	Coefficientes (%)
Aproximación		64	0,10
Detalle	5	192	0,29
	4	768	1,17
	3	3.072	4,69
	2	12.288	18,75
	1	49.152	75,00

Cuadro 4.3: Cantidad de coeficientes por banda. Cálculo realizado en una imagen de 256 por 256 luego de bajar cinco niveles en el algoritmo cascada de la transformada wavelet.

$$RB_k = \Delta H(k) \cdot N^2 \cdot \frac{3}{4^k} \quad (4.4)$$

Como al intentar lo mismo en las bandas menos pobladas (donde  $k$  es mayor) la cantidad de coeficientes ( $N^2 \cdot \frac{3}{4^k}$ ) es considerablemente menor, para alcanzar el mismo rendimiento se tendría que reducir demasiado la cantidad de representantes como para que la variación de la entropía ( $\Delta H(k)$ ) pueda crecer lo suficiente.

Ejemplo: a dos bandas de distancia, la cantidad de coeficientes es 16 veces menor con lo cual la ganancia de entropía necesitaría ser 16 veces mayor. Algo realmente impensable, no sólo porque la entropía no es algo tan sencillo de reducir, sino porque al ver los histogramas por bandas resultantes de la aplicación de una transformada wavelet cualquiera, podemos ver cómo las bandas interiores cada vez se alejan más del cero y tienden a tener una mayor varianza (Figura 4.8).

Más allá de esto, la razón verdaderamente importante por la cual “no” se debería intentar ganar en compresión cuantizando los coeficientes de la banda de detalle más grueso, es que cada vez que se cuantiza uno de ellos, se está modificando una gran cantidad de pixels de la imagen original. Debe recordarse que el area de influencia asociada a los coeficientes de las bandas de detalle es cada vez más amplia a medida que aumenta el número de la banda, es decir, a medida que se avanza hacia las bandas interiores. Si se cuantiza demasiado en estas bandas, obviamente se vería reflejado en un gran aumento del MSE entre las imágenes.

## Capítulo 5

# Distribución de bits por banda

### 5.1. Medición de la calidad final

Una de las características más interesantes de un compresor es su capacidad de comprimir una imagen de manera que la imagen reconstruida tenga una cierta calidad visual determinada. Obviamente, la idea es que para la calidad definida la señal codificada resulte lo más pequeña posible. Debe recordarse que la medida que se usará como aproximación a la calidad visual es el PSNR. Este pedido que parece tan sencillo, conlleva una dificultad que es que, para saber la calidad final de la imagen, tendría que antitransformarla y verificar si el resultado es o no de la calidad deseada. Afortunadamente, las propiedades de las transformadas ortogonales (ver Sección 3.5) vuelven a acudir en nuestra ayuda. Estas aseguran que el error en norma 2 producido al cuantizar<sup>1</sup> en la transformada será igual que el error luego de que sea antitransformada. De esta manera, sabiendo el PSNR (ver ecuación 2.4) que debe tener la imagen final, se puede despejar el MSE<sup>2</sup> que se permite tener entre la imagen original y la reconstruida. Así, cumplir con el pedido de una cierta calidad se interpreta como que el MSE de la diferencia de la transformada antes y luego de cuantizarla sea como máximo el MSE despejado de la fórmula del PSNR.

$$MSE = \frac{(\text{rango})^2}{10^{\frac{PSNR}{10}}} \quad (5.1)$$

Ahora, sólo falta saber qué información conviene descartar de manera tal de comprimir lo más que se pueda sin afectar la calidad final.

Luego de aplicar la transformada, se tiene por resultado una señal de  $L$  bandas de detalle y una de aproximación. Cada una de estas será cuantizada por separado. El problema es saber cuántos representantes va a convenir tomar para cada una de ellas. El hecho es que se tiene una cierta cantidad de error que se puede cometer en la aproximación de la imagen original y es necesario

<sup>1</sup>Se menciona la técnica de cuantización pero lo mismo se aplica a cualquier otra operación que resulte en pérdida de información, como por ejemplo el umbral.

<sup>2</sup>Se suponen imágenes de rangos de 0 a 255.

saber cómo repartirlo entre las bandas. A partir de la definición del MSE (ver ecuación 2.3) se puede expresar de la siguiente manera:

$$\text{MSE} = \frac{1}{N^2} \left( \sum_i \sum_j (x_{ij}^{(1)} - Q^{(1)}(x_{ij}^{(1)}))^2 + \dots + \sum_i \sum_j (x_{ij}^{(L)} - Q^{(L)}(x_{ij}^{(L)}))^2 \right) \quad (5.2)$$

donde los  $x_{ij}^{(k)}$  para  $k < L$  son los coeficientes correspondientes a la  $k$ -ésima banda de detalle; los  $x_{ij}^{(L)}$  son los coeficientes de aproximación y  $Q^{(k)}$  es la función de cuantización de la  $k$ -ésima banda de detalle (o de aproximación si  $k = L$ ).

Si se define el MSE correspondiente a la banda  $k$  como

$$M_k = \frac{1}{N^2} \sum_i \sum_j (x_{ij}^{(k)} - Q^{(k)}(x_{ij}^{(k)}))^2 \quad (5.3)$$

entonces

$$\text{MSE} = \sum_k M_k \quad (5.4)$$

## 5.2. Distribución de error

Para repartir el error entre las bandas de detalle se plantearán, en principio, dos heurísticas como alternativas para tomar una decisión en cuanto al reparto sin involucrar cálculos complejos. Posteriormente, se presenta un tercer método cuya fundamentación teórica y cuyos resultados en varias pruebas lo posicionan como al mejor de los tres.

Hasta este punto, se ha considerado que deben tratarse a las bandas de detalle como si fueran señales independientes. El próximo paso es tratar de encontrar una relación entre ellas que permita asegurar el mejor equilibrio entre la calidad y la compresión de la imagen entera. Para esto, se analiza el comportamiento de las bandas en relación al aporte que puede brindar a la imagen en cualquiera de estos dos aspectos mediante las figuras 4.11 y 4.12. En estas se grafica, para cada una de las cinco primeras bandas de detalle, la contribución de la banda  $k$  al MSE total ( $M_k$ ) en función de la cantidad total de bits utilizados ( $B(k)$ ).

Si se observa la curva de la primera banda ( $\nabla$ ) en las figuras 4.11 y 4.12, puede verse que es la más apta para ser comprimida sin que eso resulte en una considerable pérdida de calidad. En ciertas partes de su curva queda determinado que se puede llegar a *ahorrar* hasta 44.000 bits de su tamaño estimado total aumentando el MSE en sólo medio punto (0,52). Por el contrario, en la figura 4.12, la última banda ( $\square$ ) cae casi verticalmente dejando en claro que no hay mucho para comprimir y sí mucho para perder en calidad, ya que cualquier aumento del error que se permita no disminuirá en nada la cantidad de bits necesarios para su codificación. En gran parte, esto se da tanto por la distribución de valores dentro de la banda (ver Figura 4.8), por la ínfima cantidad de coeficientes que conforman la misma y por lo extremadamente alto de sus valores absolutos (ver

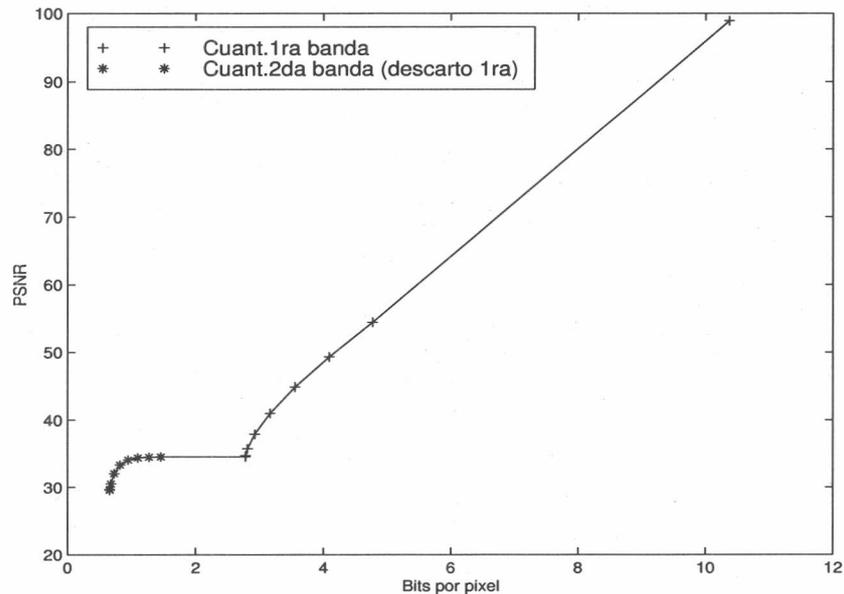


Figura 5.1: Relación entre compresión (Bits por píxel) y calidad (PSNR) para el descarte progresivo de las bandas más finas. Imagen Lena con wavelet Sym8.

Tabla 4.3). Si se piensa que las últimas bandas tienen tan pocos coeficientes, con valores tan altos y una dispersión tan amplia entre sus valores es un indicio de que, en éstas, no quedará mucho margen para comprimir.

¿Cabe la posibilidad de que sea mejor dejar intactas a las bandas de detalle más grueso? ¿Deberían las bandas de detalle fino esforzarse hasta el límite de sus posibilidades antes de perder información en las de detalle más grueso? ¿Será correcto pensar que todas las bandas deberían hacer el mismo aporte para que, al codificar la imagen con la calidad pedida, llegue a obtenerse una mejor compresión?

### 5.3. Descarte de las bandas más finas

Mediante este método se tratará de aprovechar las características mencionadas en la sección anterior, por las que las bandas de detalle más fino sería en las que conviene comprimir de manera de no alterar tanto la calidad de la imagen reconstruida.

Dado el error deseado para la imagen resultante, se comenzará reduciendo la cantidad de representantes de la primera banda de detalle hasta que el error cometido sea el mismo que el deseado. En caso de haber reducido la cantidad de representantes hasta el mínimo posible (sólo un representante) y no haber alcanzado el error pedido para la imagen resultante, se descartará por completo la primera banda de detalle. Luego, se comenzará a reducir la cantidad de representantes de la segunda banda de detalle. Así sucesivamente hasta alcanzar el error solicitado.

Por lo tanto, si se descartan  $r$  bandas de detalle

$$M_k = \frac{1}{N^2} \sum_i \sum_j (x_{ij}^{(k)})^2 \quad \text{si } 1 \leq k < r \quad (5.5)$$

$$M_k = \frac{1}{N^2} \sum_i \sum_j (x_{ij}^{(r)} - Q^{(r)}(x_{ij}^{(r)}))^2 \quad \text{si } k = r \quad (5.6)$$

$$M_k = 0 \quad \text{si } k > r \quad (5.7)$$

Debe recordarse que los coeficientes de todas las bandas de la transformada son números reales. A causa de esto, es fundamental realizar una primera cuantización de manera de no tener los inconvenientes propios de querer representar un número de precisión arbitraria con una cantidad finita de bits. Se decide conservar una precisión de hasta dos decimales para lo cual se cuantiza inicialmente con paso 0,01 y luego se va aumentando el paso para reducir la cantidad de representantes.

En la figura 5.1 puede observarse el desempeño del algoritmo propuesto. La curva indica la calidad estimada de la imagen resultante en base a la cantidad de bits por pixel con la que se codifique. Se graficó con dos símbolos distintos la reducción de representantes en la primer banda (+) y la de la segunda (\*). Es notoria la diferencia que se produce entre la última medición antes y después de descartar la primer banda. La gran diferencia se produce porque en una no se cuantiza la segunda banda mientras que en la otra se la cuantiza con paso 2. La cantidad de bits por pixel baja abruptamente debido a la pérdida de todos los decimales de la segunda banda, los cuales incrementaban la entropía pero no aportan significativamente a la calidad final.

Lo que queda claro, es que podríamos haber aumentado el paso de cuantización de la segunda banda, reduciendo la cantidad de BPP, antes de seguir aumentando el paso de cuantización de la primer banda, lo que en sus últimos incrementos no provocó sino una pérdida de calidad en la reconstrucción. Esto indicaría la conveniencia de cuantizar las distintas bandas de forma simultánea.

#### 5.4. Error por coeficiente

Tal como se planteó en la sección 4.3, la cantidad de coeficientes que conforman una banda de detalle es uno de los factores fundamentales en la variación del tamaño en bits de la banda codificada.

Lo que se propone mediante este método es que el error que se le permita acumular a cada banda sea proporcional a la cantidad de coeficientes que la conforman. Entonces, a partir de la ecuación 5.3 se establece que, para una determinada banda de detalle  $k$

$$M_k = \frac{1}{N^2} \sum_i \sum_j (x_{ij}^{(k)} - Q^{(k)}(x_{ij}^{(k)}))^2 = \frac{3}{4^k} \cdot \text{MSE} \quad \text{para } k < L \quad (5.8)$$

y también que para la banda de coeficientes de aproximación  $L$

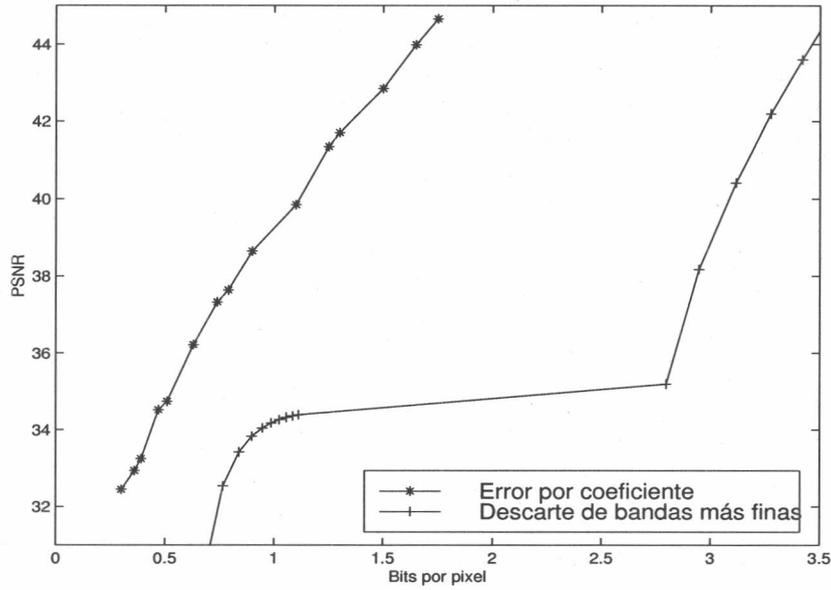


Figura 5.2: Curva de desempeño del método de Error por Coeficiente en comparación con el de Descarte progresivo de las bandas más finas.

$$M_L = \frac{1}{4^L} \cdot \text{MSE} \quad (5.9)$$

Sabemos que  $\frac{3}{4^k}$  es la proporción de coeficientes de la banda de detalle  $k < L$  en relación a los de la señal entera y  $\frac{1}{4^k}$  es la proporción de coeficientes de aproximación (banda  $L$ ). Se puede ver que

$$\text{MSE} = \sum_k M_k = \left[ \sum_{k=1}^{L-1} \frac{3}{4^k} + \frac{1}{4^L} \right] \cdot \text{MSE} \quad (5.10)$$

A diferencia del método explicado en la sección anterior, en este se cuantizan todas las bandas simultáneamente. De esta manera, se puede aprovechar en todas las bandas a la vez la reducción de los bits por pixel, resultado inicialmente del descarte de los bits menos significativos en cada coeficiente, los cuales no tenían un peso significativo en la calidad de la imagen final.

Puede verse en la figura 5.2 la curva de desempeño para el algoritmo propuesto, en comparación con el presentado en la sección anterior.

A pesar de las mejoras obtenidas en el desempeño por haber incorporado información sobre la cantidad de coeficientes al criterio de reparto del error entre las bandas, existe información que este método no toma en cuenta para la distribución del error permitido entre las bandas: “la distribución de los valores”. Por consiguiente, estaremos asignando *siempre* la misma cantidad de error para las diferentes bandas más allá de la imagen con la que se esté trabajando.

## 5.5. Método de la derivada

Esta aproximación trabajará bajo la suposición de que es fundamental lograr un aporte equivalente en cuanto a calidad y compresión entre todas las bandas de detalle. Lo primero que se necesita para lograrlo, es tener una medida que de una idea del mismo. Para esto volveremos sobre la figura 4.12. Si se denomina  $C^{(k)}$  a la curva asociada con la  $k$ -ésima banda, lo que se busca es una forma de relacionar cada punto de una curva con un punto en cada una de las otras cuatro. Se hace referencia a un punto de cada curva, el cual está determinado por el par  $(x, y) = (B_k, M_k)$ . Luego, la medida que se menciona debería determinar que en cada uno de esos puntos (uno por banda), las cinco bandas están colaborando en las mismas proporciones entre calidad y compresión.

La medida propuesta será *la derivada de la curva*.

$$\frac{\partial C^{(k)}}{\partial x} \approx \frac{\Delta y}{\Delta x} = \frac{\Delta M_k}{\Delta B_k} \quad (5.11)$$

¿Qué significa la derivada en cualquiera de las curvas asociadas a las bandas? Si se observa que la derivada en un determinado punto es, por ejemplo, de -0,003 quiere decir que por cada bit menos (o más) que ocupe la codificación de esta banda el MSE aumentará (o disminuirá) en 0,003. También puede verse de manera inversa, por cada unidad de MSE en el que el error por la codificación de la banda se incremente (o decremente), la cantidad de bits necesarios para codificar toda la banda bajará (o subirá) 3000 unidades.

Para una banda determinada de detalle  $k$ , la cantidad de representantes o de intervalos que resulten de la aplicación de la función de cuantización  $Q^{(k)}$ , determina exactamente un punto  $(x_k, y_k)$  sobre la curva  $y = C^{(k)}$ . El punto  $y_k$  es el que antes llamamos  $M_k$ . En este modelo, se considerará que las bandas están logrando el mismo aporte, siempre y cuando sus derivadas sean iguales (dentro de lo posible) en el punto determinado por la función  $Q^{(k)}$  correspondiente a cada banda.

$$\forall k : C'^{(k)}(x_k) = K \quad (5.12)$$

De esta manera, si la cantidad total del error de la imagen codificada ( $MSE$ ) es menor al permitido (ver 5.2), todas las bandas podrán disminuir la cantidad de sus representantes (modificando  $Q$ ), para aumentar el error por banda ( $M_b$ ), de forma tal de situarse en una nueva posición sobre sus respectivas curvas ( $c_b$ ) en la que sus derivadas vuelvan a ser iguales. Si el error fuera mayor, el desplazamiento por las curvas sería en el sentido contrario.

Se define el paso de los distintos  $Q^{(k)}$  de manera tal que determinen los  $M_k$  para cumplir con la ecuación 5.12 así como también que

$$MSE = \sum_k M_k \quad (5.13)$$

En la figura 5.3 se grafican, para mayor claridad, sólo dos de las curvas con sus correspondientes derivadas. Debajo, en la figura 5.4 se ve la imagen original y la reconstruida que se logró por este método con la derivada antes mencionada. En la tabla 5.1 vemos la comparación de los resultados obtenidos para el experimento anterior (Ejemplo 1) con los obtenidos para otra derivada

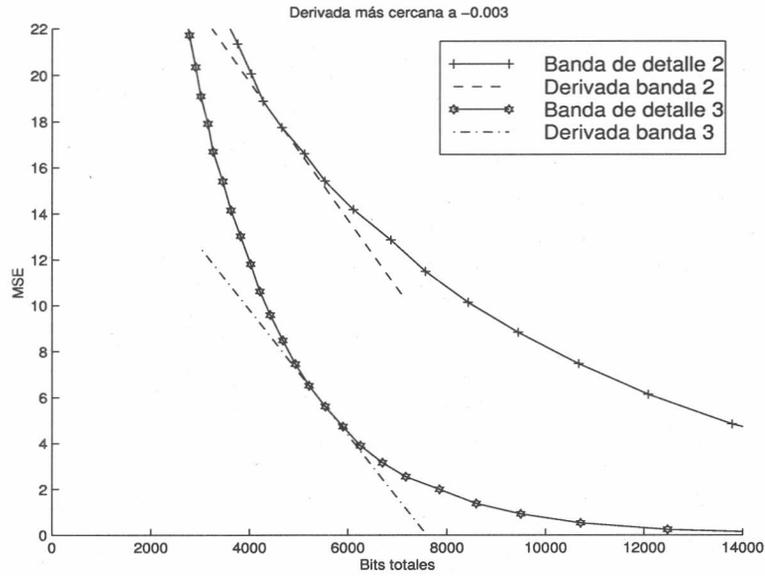


Figura 5.3: Curvas asociadas a las bandas de detalle 2 y 3 con sus respectivas derivadas. Las derivadas en el punto graficadas en ambas curvas tienen el mismo valor.

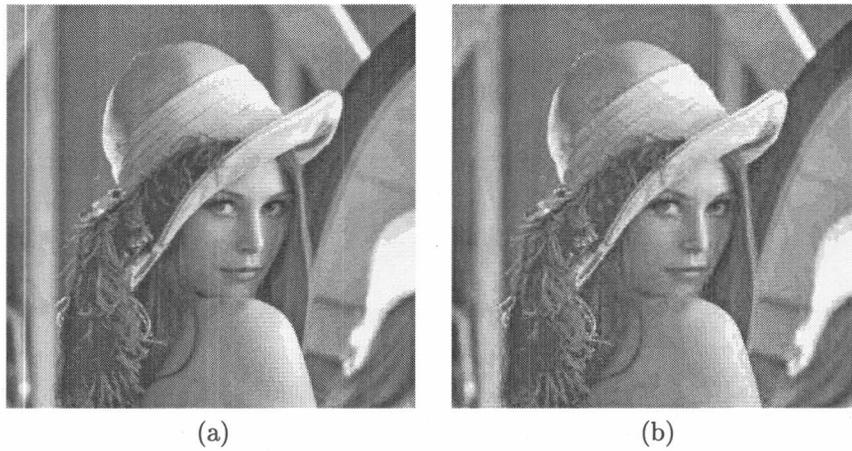


Figura 5.4: (a) Imagen original. (b) Imagen reconstruida a 0,5 bpp. El PSNR es de 35,08 y el tamaño estimado total de codificación es de 33125 bits.

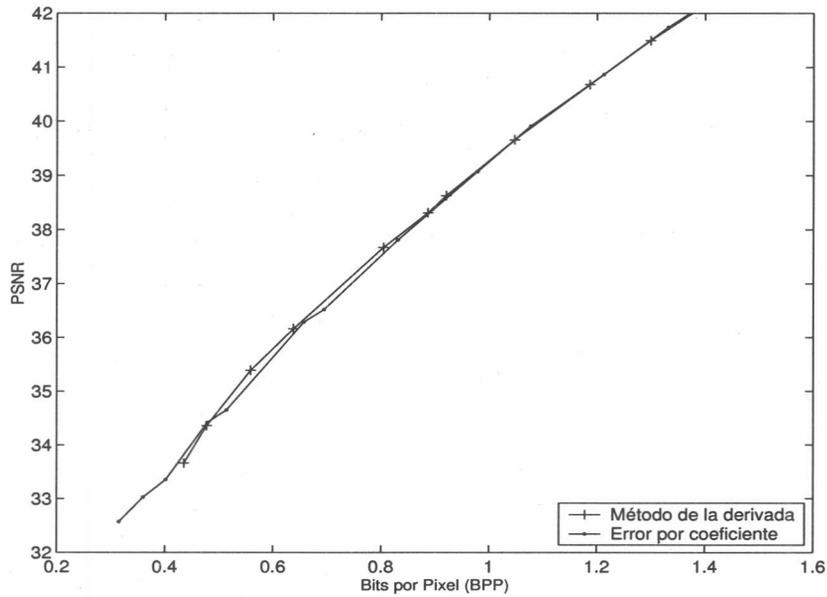


Figura 5.5: Comparación del desempeño entre el método de iguales derivadas y el del error por coeficiente.

(Ejemplo 2). En la tabla 5.2 se comparan otros datos de los mismos ejemplos pero desagregados por cada una de las bandas de detalle.

En la figura 5.5 se puede observar la comparación de desempeño entre el método de iguales derivadas y el del error por coeficiente. Si bien el método de iguales derivadas es superior al del error por coeficiente, la diferencia no es tan notoria como en el otro caso y se mantiene a lo largo de toda la curva.

Resulta necesario tener algunas mediciones standard para comparar el rendimiento de los algoritmos. En la figura 5.6 podremos observar el desempeño del algoritmo propuesto comparándolo con el modo *no reversible* (NR) de JPEG 2000 [SCE01]. Puede observarse que para cualquier cantidad de BPP se supera el resultado alcanzado por el JPEG 2000 NR. Sobre todo para el caso de menos BPP en que la diferencia llega a ser de 2.5 puntos de PSNR.

	Ejemplo 1	Ejemplo 2
PSNR	35	40
BPP	0.5	1.05
Derivada	-0.003	-0.00024

Cuadro 5.1: PSNR, Bits por pixel y valor de la derivada para dos ejemplos.

Banda	Ejemplo 1 (PSNR = 35 dB)			Ejemplo 2 (PSNR = 40 dB)		
	BPP	Q	Clases	BPP	Q	Clases
1	0.16	36	6	0.56	15	13
2	1.02	27	12	1.96	11	28
3	2.56	24	25	3.75	10	59
4	4.44	23	51	5.71	9	131
5	5.58	26	121	6.52	9	349

Cuadro 5.2: Bits por pixel, Paso de cuantización y Cantidad de clases por Banda de detalle según PSNR.

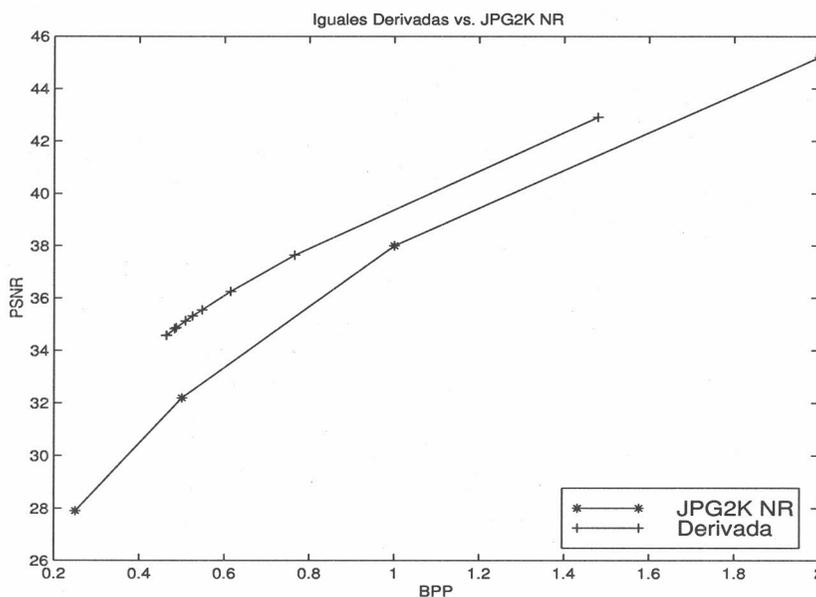


Figura 5.6: Comparación de la relación calidad/compresión del algoritmo de Iguales derivadas y el standard JPEG 2000 NR (No Reversible).

### 5.5.1. Desarrollo teórico

Para definir una función que dé cuenta del “costo” de comprimir una señal debe tenerse en cuenta el error cometido y la cantidad de bits que resultará de la codificación [GG92].

Si se supone una señal a cuantizar, donde  $MSE(q)$  es el error cuadrático medio y  $B(q)$  es la cantidad de bits totales que se obtienen de la aplicación de un cuantizador uniforme con paso igual a  $q$ , podemos definir a la función “costo” como

$$C(q) = MSE(q) + K \cdot B(q) \quad (5.14)$$

donde  $K$  es el valor que determina la importancia del aumento de la cantidad total de bits en relación a la disminución del error.

Como lo que se quiere es minimizar el costo, se busca que la derivada de esta función se anule, por lo que

$$\begin{aligned} \frac{\partial C}{\partial q} &= 0 \\ \frac{\partial MSE}{\partial q} + K \cdot \frac{\partial B}{\partial q} &= 0 \\ \frac{\partial MSE}{\partial q} &= -K \cdot \frac{\partial B}{\partial q} \end{aligned}$$

Entonces, siempre que quiera minimizar el costo, debe valer que

$$K = -\frac{dMSE}{dB} \quad (5.15)$$

Este desarrollo supone un mismo cuantizador  $q$  para todos los coeficientes de la imagen.

Si se tomaran pasos de cuantización  $(q_1, \dots, q_L)$  diferentes para cada banda, podríamos redefinir las funciones como

$$MSE(q_1, \dots, q_L) = \sum_{j=1}^L M_j(q_j) \quad (5.16)$$

como la función que devuelve el error cometido al cuantizar las  $L$  bandas de la señal con los pasos de cuantización  $q_j$ .

Análogamente, se define

$$B(q_1, \dots, q_L) = \sum_{j=1}^L B_j(q_j) \quad (5.17)$$

como la función que devuelve la cantidad de bits necesarios para codificar la señal luego de cuantizar las  $L$  bandas.

Por último, la función “costo” para toda la señal se define como

$$C(q_1, \dots, q_L) = M(q_1, \dots, q_L) + K \cdot B(q_1, \dots, q_L) \quad (5.18)$$

donde  $K$  es el peso relativo que le asigno a cada bit que ocupe la codificación en relación al error cometido.

Tal como en el caso anterior, lo que quiero es minimizar este costo. Por lo tanto, para que la solución encontrada sea un mínimo, debo pedir que todas sus derivadas parciales en ese punto se anulen.

$$\forall i : 1 \leq i \leq L : \frac{\partial C}{\partial q_i} = 0 \quad (5.19)$$

$$\begin{aligned} \frac{\partial C}{\partial q_i} &= \frac{\partial \text{MSE}}{\partial q_i} + K \cdot \frac{\partial B}{\partial q_i} \\ &= \sum_{j=1}^L \frac{M_j(q_j)}{\partial q_i} + K \cdot \sum_{j=1}^L \frac{B_j(q_j)}{\partial q_i} \\ &= \frac{\partial M_i(q_i)}{\partial q_i} + K \cdot \frac{\partial B_i(q_i)}{\partial q_i} = 0 \end{aligned}$$

$$\forall i : 1 \leq i \leq L : K = -\frac{\partial M_i}{\partial B_i} \quad (5.20)$$

con lo que la única forma de minimizar el costo es encontrar puntos sobre cada curva donde sus derivadas valgan lo mismo.

El resultado obtenido es equivalente al que se logra a partir de la resolución del problema

$$\begin{aligned} &\text{Minimizar } M(q_1, \dots, q_L) \\ &\text{sujeto a que } B(q_1, \dots, q_L) = b \end{aligned}$$

por el método de minimización de Lagrange.

Puede verse que si se quiere minimizar el costo sólo existe una variable libre. Si se fija el valor de alguno de los  $q_i$ , este determinará el valor de  $K$  y, como todas las otras derivadas deben valer igual, determinará también el valor de los otros pasos de cuantización de manera que cumplan la ecuación 5.20.

Así, podemos considerar a  $K$  como la única variable libre en nuestro sistema de ecuaciones, ya que por sí sola determina el valor de todos los pasos de cuantización ( $q_i$ ).

¿Y qué significan los valores que toma  $K$ ? Tal como se expresara anteriormente, puede verse como la cantidad máxima de error dispuestos a incrementar en la señal reconstruida por cada bit que se ahorre en el proceso de cuantización y codificación. También quiere decir que los puntos que dan una alocaión óptima de bits están sobre la cápsula convexa interna de cada curva.

Esta idea fue utilizada por Goldschneider [GR98] como criterio para la conformación del árbol óptimo de wavelet packets de una señal unidimensional y por Ramchandran y Vetterli [ROV94] para su utilización con wavelet packets para una señal auditiva.

**Estimación de las derivadas**

Originalmente, sobre la curva  $C_1$ , por ejemplo, se tomaban dos puntos que correspondían a

$$(B_1(q_1(n)), M_1(q_1(n))) \quad (5.21)$$

y a la izquierda sobre la curva

$$(B_1(q_1(n-1)), M_1(q_1(n-1))) \quad (5.22)$$

donde  $q_1(n)$  es el paso necesario en la banda de detalle 1 para obtener una cantidad de intervalos de cuantización igual a  $n$ . La derivada era estimada como la pendiente de la recta secante que pasaba por ambos puntos.

$$\frac{\Delta M_1}{\Delta B_1} = \frac{M_1(q_1(n)) - M_1(q_1(n-1))}{B_1(q_1(n)) - B_1(q_1(n-1))} \quad (5.23)$$

Para las curvas correspondientes a las bandas de detalle, la variación entre dos puntos determinados por la cuantización en  $n$  intervalos y en  $n-1$ , sufre oscilaciones locales debido a errores de redondeo. Este efecto se incrementa a medida que se trata de bandas de detalle más grueso.

Debido a esta dificultad, se toman incrementos más grandes a medida que se estima sobre bandas de detalle más grueso. En general, la derivada para la curva de desempeño de la banda de detalle  $j$  en el punto determinado por el paso  $q_j$  que da exactamente  $n$  intervalos de cuantización es

$$\frac{\Delta M_j}{\Delta B_j} = \frac{M_j(q_j(n)) - M_j(q_j(n-2^{j+1}))}{B_j(q_j(n)) - B_j(q_j(n-2^{j+1}))} \quad (5.24)$$

## Conclusiones

A lo largo de este trabajo se ha demostrado la conveniencia de cuantizar y codificar a cada banda de detalle por separado en vez de la transformada wavelet entera.

Posteriormente, se analizó la ventaja que brinda la aplicación de la cuantización uniforme por sobre el algoritmo de Max-Lloyd, a pesar de que este último converge a una solución que minimiza el MSE para una misma cantidad de intervalos que el primero.

En el capítulo 5 se propusieron tres métodos de asignación de bits, donde cada uno aprovecha de una manera diferente la distribución de las distintas bandas de detalle de la transformada. Los tres métodos fueron comparados entre sí.

Al ver los resultados del primer método se concluyó que no era eficiente asignar todo el error disponible a una sola banda, ya que incrementando levemente el paso de cuantización en las otras se bajaba de manera considerable la cantidad de BPP manteniendo la misma calidad final.

El segundo método solucionó este punto pero no era muy flexible en el reparto del error por banda ya que, para dos imágenes del mismo tamaño, el error permitido por banda es exactamente el mismo. Esto se produce, básicamente, porque no se utiliza la información relacionada a la distribución de valores. Sin embargo, este método cuya implementación es muy sencilla, siempre (en las imágenes de prueba consideradas) tuvo un desempeño apenas inferior al óptimo.

El tercer método, o método de iguales derivadas, es el óptimo según los resultados obtenidos en varias pruebas. Además, posee una sólida fundamentación teórica. Se planteó la función "costo", que expresa una correcta proporción entre la calidad y la compresión; y la medida de esta proporción es la derivada. Se propuso una variante en el cálculo aproximado de las derivadas para descartar las oscilaciones de las bandas de detalle más grueso.

A partir de los resultados alcanzados en este trabajo se pueden plantear líneas de investigación a seguir, como la utilización del método de la derivada para la transmisión progresiva de imágenes o la compresión sin pérdida, de manera de intentar mejorar la codificación que suele realizarse en base a bit-planes y/o la relación espacial de ciertos coeficientes en función de los de otras bandas.

También podría desarrollarse alguna variante para la cuantización de las bandas de detalle que mejore el desempeño de la uniforme, de manera de obtener una mejor calidad sin incrementar la cantidad de BPP.

## Apéndice A

# Resultados comparativos entre métodos

Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
34.0469	0.9467	32.45	0.2984	34.25	0.4590
34.1778	0.9874	32.94	0.3506	34.67	0.4838
34.2621	1.0241	33.25	0.3885	35.57	0.5594
34.3194	1.0560	34.52	0.4669	36.39	0.6441
34.3603	1.0858	34.74	0.5002	37.74	0.7925
34.3905	1.1124	36.22	0.6283	38.37	0.8664
35.1974	2.7977	37.33	0.7313	38.91	0.9324
38.1724	2.9473	37.64	0.7812	39.54	1.0160
40.4179	3.1165	38.65	0.8902	40.95	1.2020
42.1960	3.2761	39.85	1.0507	41.54	1.2852
43.5980	3.4197	41.35	1.2460	42.82	1.4772
44.8027	3.5487	41.71	1.2962		
45.8275	3.6635	42.85	1.4640		
46.7169	3.7693	43.99	1.6491		
47.5239	3.8673	44.66	1.7469		
48.2629	3.9567				

Imagen		Tamaño		Wavelet	
Peppers		256 x 256		Sym8	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
28.4911	0.7344	30.46	0.4416	31.97	0.5051
29.4420	0.7765	32.69	0.5483	33.08	0.5775
29.9939	0.8132	33.09	0.6051	34.96	0.7565
30.3503	0.8440	34.61	0.7248	35.31	0.7929
30.5748	0.8743	34.87	0.7681	36.67	0.9571
30.7515	0.9020	36.20	0.9074	37.26	1.0281
30.8789	0.9272	37.33	1.0452	37.71	1.0901
35.1593	2.9823	38.31	1.1760	38.45	1.1950
36.7444	3.0712	38.89	1.2599	39.65	1.3833
37.9690	3.1603	40.09	1.4571	40.45	1.5241
38.9788	3.2429	41.04	1.6294	41.68	1.7563
39.7893	3.3208	41.47	1.7126		
40.4735	3.3963	43.05	2.0124		
41.0873	3.4721	43.38	2.0748		
		45.48	2.4631		

Imagen		Tamaño		Wavelet	
Woman		128 x 128		Sym8	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
28.8748	0.9278	30.64	0.7013	30.73	0.7267
28.9983	0.9752	32.00	0.8786	31.54	0.8228
29.0780	1.0155	32.37	0.9367	32.51	0.9503
29.1280	1.0532	34.02	1.1404	34.08	1.1679
29.1641	1.0839	35.36	1.3196	34.71	1.2639
29.1877	1.1121	36.50	1.4950	35.76	1.4182
30.3563	2.6973	37.41	1.6546	37.19	1.6385
34.4073	3.0056	37.79	1.7170	38.48	1.8532
37.2618	3.2775	38.56	1.8374	40.44	2.1699
39.2288	3.5212	40.24	2.1058	41.08	2.2748
40.8791	3.7255	40.78	2.2018	41.89	2.4141
42.2307	3.8929	42.63	2.5199	42.89	2.5885
43.3886	4.0446	43.46	2.6718		
44.4124	4.1798	44.99	2.9496		
45.3045	4.3050	45.16	2.9749		
46.1311	4.4151				

Imagen		Tamaño		Wavelet	
Amber		128 x 128		Sym8	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
31.2059	0.8124	30.57	0.4059	32.90	0.5597
31.3664	0.8465	32.60	0.5365	33.18	0.5764
31.4573	0.8743	32.90	0.5694	33.80	0.6286
31.5181	0.9045	33.25	0.6166	35.48	0.8019
31.5591	0.9316	35.09	0.7440	36.28	0.9026
31.5921	0.9565	36.36	0.8805	36.83	0.9668
32.9409	2.5892	36.70	0.9399	38.11	1.1263
37.1562	2.8033	37.97	1.1001	39.25	1.2825
39.5825	3.0146	38.92	1.2224	41.40	1.5799
41.5868	3.2018	40.00	1.3702	41.80	1.6342
43.1772	3.3544	40.93	1.4934	43.08	1.8263
44.4214	3.4900	41.89	1.6223	43.67	1.9247
45.4794	3.6067	42.65	1.7378		
46.3664	3.7136	44.33	1.9949		
47.1677	3.8113	44.34	1.9969		
47.8586	3.9065				

Imagen		Tamaño		Wavelet	
Lena		256 x 256		Db4	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
33.1777	0.9831	31.96	0.3259	33.66	0.4355
33.2679	1.0275	32.05	0.3376	33.76	0.4405
33.3256	1.0644	32.38	0.3826	33.85	0.4463
33.3622	1.0975	33.92	0.4759	34.36	0.4773
33.3898	1.1263	35.45	0.5972	34.70	0.5006
33.4081	1.1518	35.72	0.6373	34.88	0.5141
34.4885	2.8072	36.93	0.7519	35.39	0.5591
37.5326	2.9593	37.92	0.8594	36.16	0.6378
39.9034	3.1274	39.09	1.0034	37.67	0.8042
41.6308	3.2779	40.01	1.1162	38.31	0.8865
43.0359	3.4168	40.95	1.2539	38.63	0.9202
44.2257	3.5444	42.26	1.4336	39.66	1.0469
45.2390	3.6544	43.38	1.5982		
46.1166	3.7567	43.69	1.6514		
46.9198	3.8530	44.40	1.7616		
47.6252	3.9437				

Imagen		Tamaño		Wavelet	
Peppers		256 x 256		Db4	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
28.5130	0.7420	30.36	0.4421	31.99	0.5426
29.4636	0.7865	32.33	0.5676	33.06	0.6122
29.9812	0.8244	32.54	0.6022	34.91	0.7943
30.2999	0.8571	34.34	0.7457	35.65	0.8687
30.5262	0.8870	34.65	0.7974	37.05	1.0374
30.6650	0.9131	36.23	0.9482	37.52	1.1054
30.7788	0.9383	37.32	1.0779	37.80	1.1418
34.7452	3.0009	37.98	1.1604	38.60	1.2629
36.6516	3.1087	38.94	1.3052	39.64	1.4218
38.0069	3.1948	39.64	1.4156	40.46	1.5658
38.9826	3.2792	40.84	1.6305	41.71	1.7949
39.8124	3.3632	41.47	1.7474		
40.5077	3.4413	43.14	2.0569		
41.1578	3.5176	43.89	2.2045		
		45.65	2.5224		

Imagen		Tamaño		Wavelet	
Woman		128 x 128		Db4	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
27.9887	0.9806	30.17	0.7230	31.51	0.8571
28.0642	1.0315	32.18	0.9411	31.75	0.8861
28.1142	1.0740	32.44	0.9852	32.69	1.0056
28.1461	1.1108	33.87	1.1757	34.54	1.2764
28.1654	1.1405	35.17	1.3610	35.16	1.3626
28.1794	1.1704	36.48	1.5610	35.75	1.4560
29.8480	2.6993	36.78	1.6090	37.51	1.7185
33.5815	2.9873	37.69	1.7397	38.54	1.8901
36.3244	3.2722	38.74	1.9182	41.00	2.2826
38.4270	3.5018	40.14	2.1288	41.06	2.2915
40.0466	3.6933	41.01	2.2789	42.11	2.4853
41.3539	3.8653	42.22	2.5168	42.83	2.6134
42.5444	4.0161	43.46	2.7224		
43.5549	4.1454	43.80	2.7731		
44.4931	4.2662	44.07	2.8225		
45.2538	4.3733				

APÉNDICE A. RESULTADOS COMPARATIVOS ENTRE MÉTODOS 67

Imagen		Tamaño		Wavelet	
Amber		128 x 128		Db4	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
30.7673	0.8321	30.50	0.3982	32.90	0.5616
30.9127	0.8714	32.28	0.5163	33.15	0.5819
30.9936	0.9035	32.63	0.5707	33.66	0.6213
31.0518	0.9331	34.41	0.6999	35.82	0.8717
31.0864	0.9585	34.70	0.7431	36.04	0.8930
31.1148	0.9824	35.98	0.8838	36.79	0.9915
32.6052	2.6026	37.19	1.0397	38.14	1.1480
36.9406	2.8267	38.33	1.1682	39.28	1.3000
39.5928	3.0347	38.76	1.2378	41.62	1.6194
41.4812	3.2205	40.18	1.4123	43.02	1.8324
43.0717	3.3830	41.34	1.5728	43.70	1.9437
44.3607	3.5187	42.49	1.7433		
45.4241	3.6412	43.43	1.8820		
46.3618	3.7536	43.45	1.8842		
47.1699	3.8532	44.12	1.9863		
47.8971	3.9453				

Imagen		Tamaño		Wavelet	
Lena		256 x 256		Sa4	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
33.7343	0.8755	32.62	0.3112	33.23	0.3826
33.9536	0.9133	32.96	0.3429	34.16	0.4337
34.1040	0.9445	33.36	0.3961	35.20	0.5086
34.2011	0.9748	33.56	0.4287	35.95	0.5871
34.2809	1.0005	34.75	0.4956	37.41	0.7409
34.3379	1.0252	36.22	0.6150	38.05	0.8148
35.2687	2.7966	36.54	0.6698	38.54	0.8758
38.2652	2.9481	37.77	0.7867	39.32	0.9672
40.5408	3.1204	38.77	0.9071	40.46	1.1203
42.2721	3.2794	39.88	1.0466	41.51	1.2705
43.6694	3.4225	41.35	1.2445	42.89	1.4747
44.8669	3.5585	42.31	1.3835		
45.9075	3.6729	43.57	1.5792		
46.8261	3.7810	43.96	1.6511		
47.6424	3.8783	44.69	1.7574		
48.3568	3.9711				

Imagen		Tamaño		Wavelet	
Peppers		256 x 256		Sa4	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
28.2400	0.7293	31.63	0.4873	31.36	0.4582
29.0933	0.7732	31.86	0.5189	32.45	0.5331
29.6830	0.8090	33.48	0.6429	34.48	0.7356
30.0382	0.8425	33.69	0.6761	35.26	0.8195
30.2809	0.8700	34.99	0.8019	36.47	0.9648
30.4360	0.8958	36.06	0.9202	37.05	1.0388
30.5600	0.9200	37.06	1.0493	37.65	1.1193
34.1433	2.9540	37.82	1.1516	38.39	1.2223
35.5784	3.0304	38.89	1.2971	39.44	1.3898
36.7618	3.1063	40.04	1.4875	40.41	1.5621
37.7841	3.1804	41.36	1.7384	41.65	1.7909
38.6271	3.2500	42.47	1.9373		
39.3504	3.3140	43.18	2.0690		
39.9379	3.3744	45.58	2.5128		

Imagen		Tamaño		Wavelet	
Woman		128 x 128		Sa4	
Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
28.7178	0.9638	30.24	0.6895	31.22	0.8080
28.7972	1.0123	31.95	0.8911	31.63	0.8526
28.8588	1.0565	33.54	1.0910	32.69	0.9808
28.8975	1.0926	33.85	1.1458	34.45	1.2283
28.9245	1.1237	34.96	1.3053	35.50	1.3961
28.9431	1.1534	36.29	1.5121	35.62	1.4174
30.1964	2.6911	37.08	1.6317	37.14	1.6424
34.2696	2.9978	38.05	1.8058	38.01	1.8020
36.9851	3.2884	38.83	1.9340	40.97	2.2804
38.9909	3.5240	39.86	2.0975	41.92	2.4437
40.7036	3.7369	41.13	2.3098	42.91	2.6189
42.0707	3.9010	41.80	2.4210		
43.2321	4.0534	42.88	2.6116		
44.2850	4.1955	44.83	2.9530		
45.2205	4.3195	44.89	2.9567		
46.0646	4.4264				

Descarte de bandas finas		Error por coeficiente		Iguales derivadas	
PSNR	BPP	PSNR	BPP	PSNR	BPP
30.5253	0.7355	31.78	0.4723	31.93	0.4828
30.7495	0.7723	31.96	0.4979	32.27	0.5059
30.9194	0.8011	33.52	0.6231	33.27	0.6014
31.0270	0.8276	33.73	0.6544	35.50	0.8244
31.1085	0.8517	35.09	0.7812	36.08	0.8864
31.1715	0.8736	36.37	0.9127	36.94	0.9744
32.1991	2.5665	37.32	1.0193	37.92	1.1035
35.6830	2.7370	38.31	1.1481	38.85	1.2201
38.2230	2.9121	38.97	1.2341	41.17	1.5391
40.0357	3.0655	39.76	1.3368	42.10	1.6711
41.5017	3.2078	41.25	1.5439	42.65	1.7636
42.7772	3.3300	42.55	1.7265		
43.8324	3.4358	42.69	1.7435		
44.7167	3.5360	43.32	1.8445		
45.5349	3.6261	44.92	2.0942		
46.2105	3.7108				

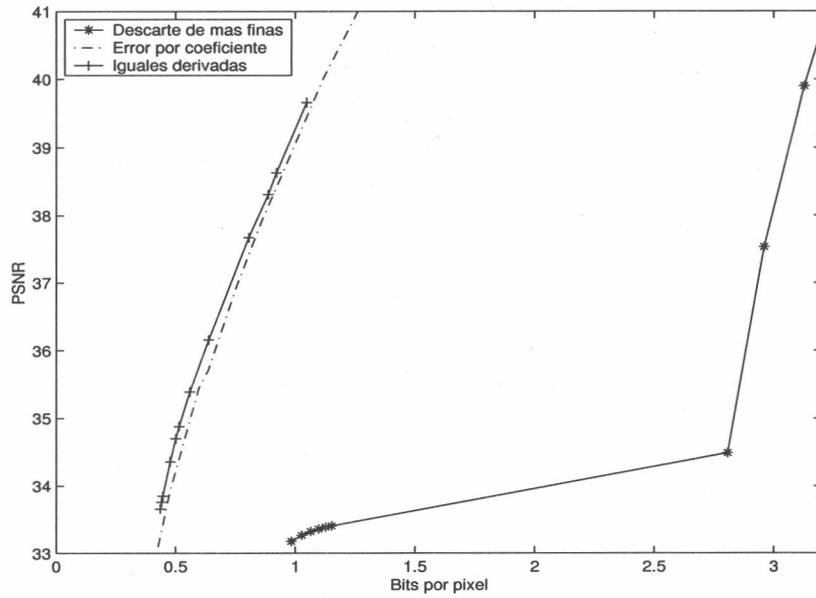


Figura A.1: Comparación del desempeño entre los tres métodos propuestos (Imagen: Lena; Wavelet: Db4).

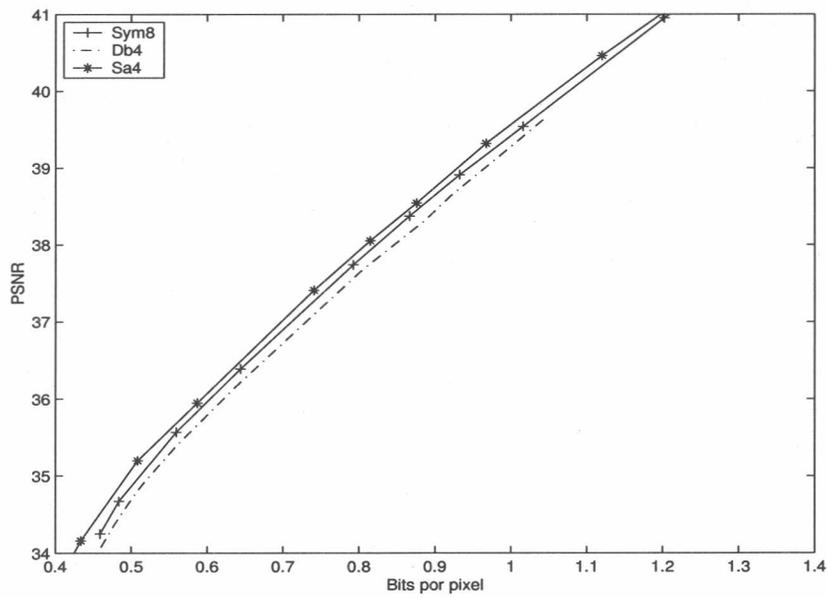


Figura A.2: Comparación del desempeño entre tres wavelets para el método de Iguales derivadas (Imagen: Lena).

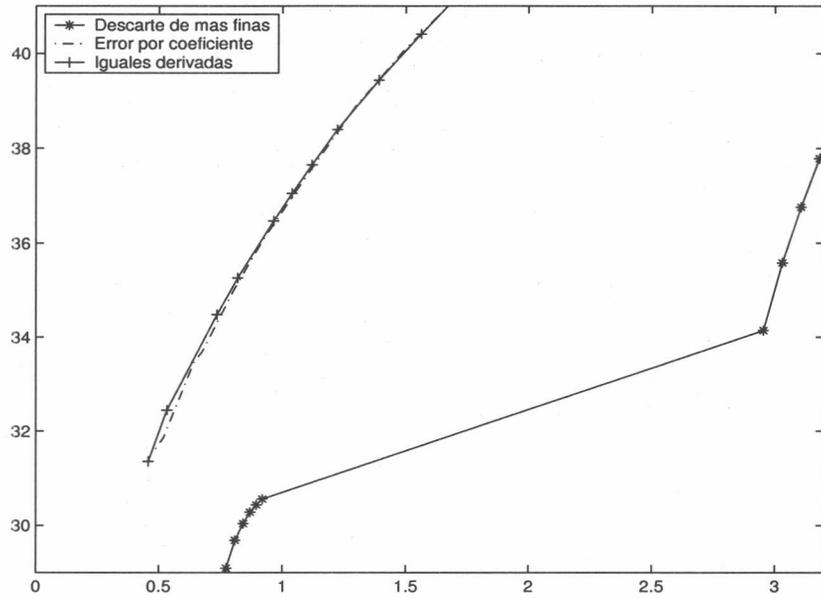


Figura A.3: Comparación del desempeño entre los tres métodos propuestos (Imagen: Peppers; Wavelet: Sa4).

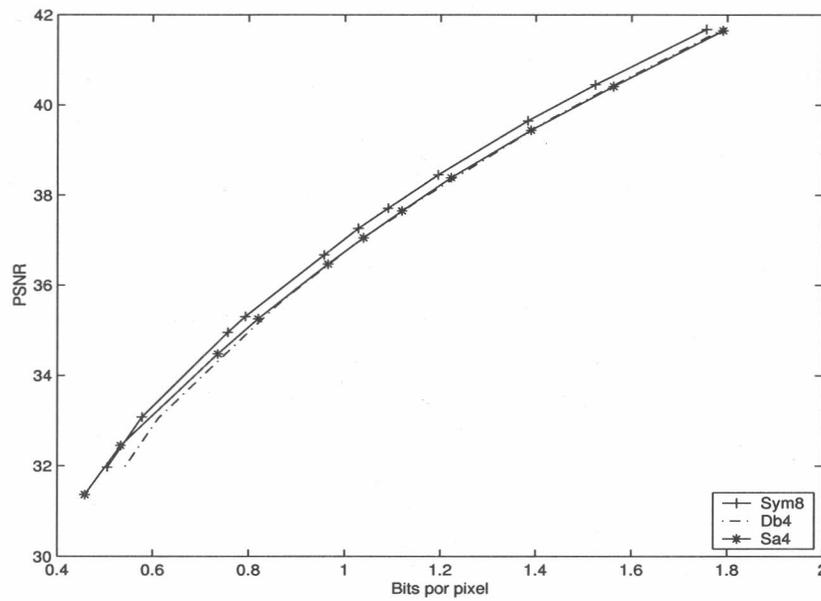
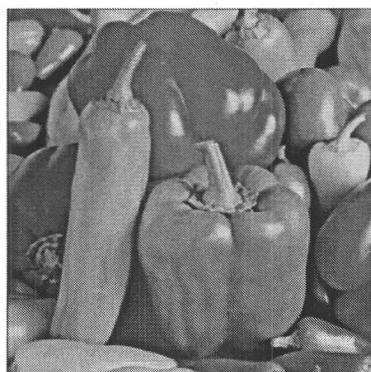


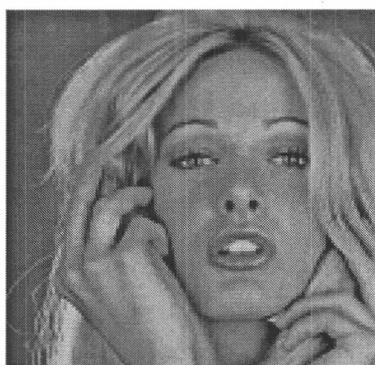
Figura A.4: Comparación del desempeño entre tres wavelets para el método de Iguales derivadas (Imagen: Peppers).



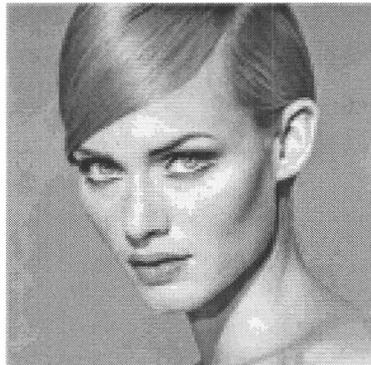
Lena



Peppers



Woman



Amber

Figura A.5: Algunas de las imágenes utilizadas.

---

# Bibliografía

- [Dau92] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, 1992.
- [GG92] Allen Gersho and Robert M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, 1992.
- [GR98] Jill R. Goldschneider and Eve A. Riskin. Optimal bit allocation and best-basis selection for wavelet packets and tsvq. In *Proceedings of the 1998 International Conference on Acoustics, Speech and Signal Processing*, 1998.
- [Lim90] J. Lim. *Two-dimensional Signal and Image Processing*. Prentice-Hall, 1990.
- [Mal99] Stéphane Mallat. *A wavelet tour of signal processing*. Academic Press, 2nd edition, 1999.
- [ROV94] Kannan Ramchandran, Antonio Ortega, and Martin Vetterli. Bit allocation for dependent quantization with applications to multiresolution and mpeg video coders. *IEEE Transactions on Image Processing*, 3(5), 1994.
- [SCE01] A. Skodras, C. Christopoulos, and T. Ebrahimi. Jpeg2000: The upcoming still image compression standard. *Elsevier, Pattern Recognition Letters*, 22:1337–1345, 2001.
- [Sha48] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.
- [SHS+99] V. Strela, P.Ñ. Heller, G. Strang, P. Topiwala, and C. Heil. The application of multiwavelet filter banks to image processing. *IEEE Trans. on Image Processing*, 1999.
- [Str89] Gilbert Strang. Wavelets and dilation equations: A brief introduction. In *Siam Review* 31, 1989.
- [Uyt94] G. Uytterhoeven. *Multiwavelets for image compression*. PhD thesis, Katholieke Universiteit Leuven, 1994.
- [Val99] C. Valens. *A really friendly guide to wavelets*. Mindless.com, 1999.
- [Wal91] Gregory K. Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 1991.

Universidad de Buenos Aires  
Facultad de Ciencias Exactas y Naturales  
Licenciatura en Ciencias de la Computación



Tesis de Licenciatura

**“Asignación óptima de bits en la cuantización de  
la transformada wavelet y multiwavelet”**

Directora: Prof. Ana María C. Ruedin  
anita@dc.uba.ar

Tesista: Javier Quinteros  
jquinte@dc.uba.ar

*Abril de 2003*