

5. Codificación y compresión de imagen

1. Introducción.
2. Características de la imagen.
 - 2.1 Captura y digitalización de imagen.
 - 2.2 Tipos de imagen
3. Compresión de imagen. Redundancia espacial.
 - 3.1 Estándar JPEG
 - 3.2 Wavelets
 - EZW
 - JPEG 2000
4. Conclusiones.

Bibliografía

- [FLU95] Understanding networked multimedia
- [GIB98] Digital Compression for Multimedia
- [TSU99] Introduction to video coding standards for multimedia communication
- [JPEG2000] JPEG 2000 performance evaluation and assessment
- [Geo99] Wavelet-based Image Coding: An Overview
- [EZW] Embedded image coding using zerotrees of wavelet coefficients

Las imágenes que percibimos están compuestas de ondas electromagnéticas (λ : 250nm - 780nm).

A diferentes longitudes de onda, diferentes sensaciones de color.

El ojo es más sensible a unos colores que a otros.

Dadas tres fuentes de luz de la misma intensidad y distinto color (una roja, otra verde y otra azul), el ojo percibe la verde con el doble de intensidad que la roja, y seis veces más intensa que la azul.

Los mecanismos de percepción visual humanos son menos sensibles y estrictos que los auditivos.

Ej.: Variaciones de frecuencia, supresión de imágenes, etc.

Mezclando 3 colores (RGB) podemos obtener otro.

Integra la información que recibe.

2.1 Captura y digitalización de imagen I

Las imágenes digitales están compuestas de píxeles (*picture element*).

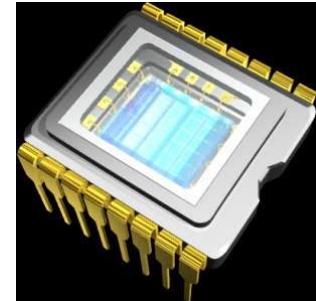
Una cámara fotográfica digital utiliza un CCD (*charge coupled device*) para realizar el proceso de adquisición analógica.

El CCD tiene una serie de pequeños diodos sensibles a la luz que convierten luz en cargas eléctricas (o sea, fotones en electrones).

Cada diodo del CCD captura un píxel de la imagen a adquirir.

Para poder situar cada píxel de la imagen (luz entrante) en su diodo correspondiente del CCD se utiliza una lente.

Mediante la lente se puede conseguir también realizar zoom óptico (no confundir con zoom digital)



2.1 Captura y digitalización de imagen II

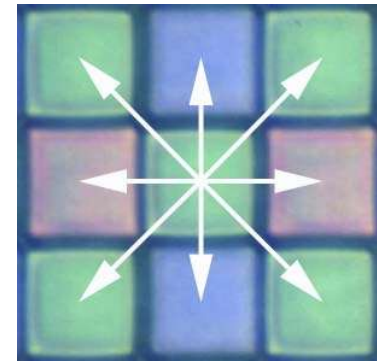
Problemática del color:

Si el CCD captura la luz directa que recibe de la lente, sólo tenemos la intensidad de luz, pero no su color.

Añadimos un filtro (R, G ó B) a cada píxel, de manera que algunos píxels reciben sólo la luz roja, otros la verde y otros la azul.

El número de píxels que reciben luz verde es el mismo que la suma de los que reciben luz roja y azul.

La información de color que no se ha obtenido en cada píxel se interpola directamente de sus vecinos, usando un DSP.



2.1 Captura y digitalización de imagen III

El CCD es un dispositivo analógico.

Es necesario un conversor analógico digital (ADC) que obtenga la representación digital de cada píxel a partir de la señal eléctrica generada por cada diodo.

Una cámara digital necesita un DSP para gestionar el funcionamiento de la cámara.

Realiza el acceso y almacenamiento de fotos en memoria, el proceso de compresión, la interpolación de los colores, gestión de menús, etc.

Uno de los más usados, el TMS320DSC24 de Texas Instruments, funciona a 80 Mhz y es utilizado por Kodak en sus productos.



2.1 Captura y digitalización de imagen IV

Codificación y recodificación.

Cada muestra RGB se codifica con una cantidad de bits por componente de color (p.ej., 8 bits/componente → 24 bits/muestra).

A veces resulta interesante codificar el nivel de brillo de una muestra (luminancia, o componente Y) y las diferencias de color (crominancias azul, roja y verde, o componentes Cb, Cr, Cg).

La conversión de RGB a YCbCr (YUV) se realiza mediante una matriz de conversión (aproximada):

$$Y = 0.3R + 0.6G + 0.1B \text{ (Nivel de brillo o luminancia)}$$

$$U = B - Y \text{ (Diferencia de color azul) (equiv. } Cb=U/2+128)$$

$$V = R - Y \text{ (Diferencia de color rojo) (equiv. } Cr=V/1.6+128)$$

Cada uno de los componentes se codifica con 8 bits.

Y (8 bits): rango 16-235

Cb (8 bits) y Cr (8 bits): rango 16-240

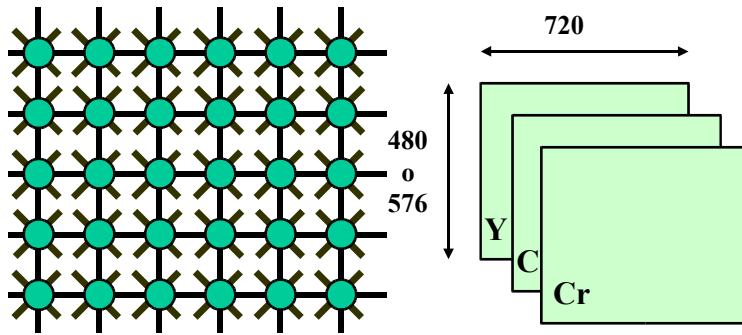
La diferencia de color verde (Cg) es redundante y no se almacena, ya que se puede obtener a partir de la Y, la Cb y la Cr.

2.1 Captura y digitalización de imagen V

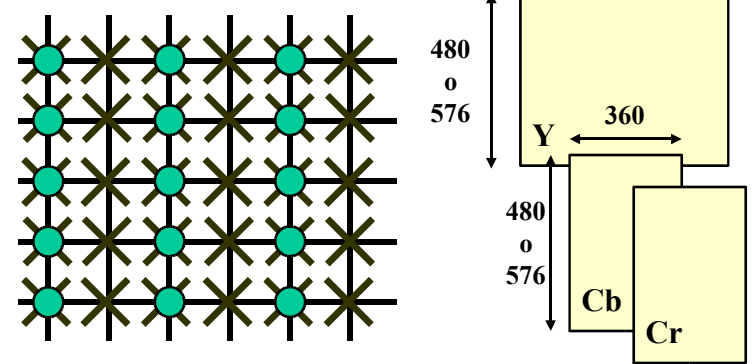
Subsampling: El ojo es más sensible a la información de luminancia que de crominancia.

● Muestra Cr + Muestra Cb

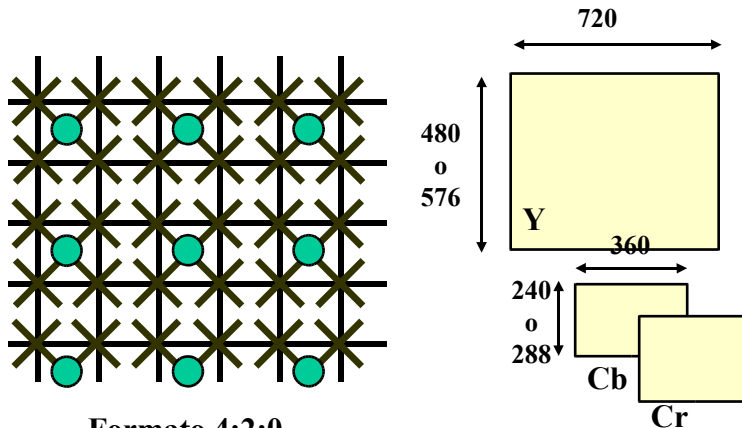
✕ Muestra Y



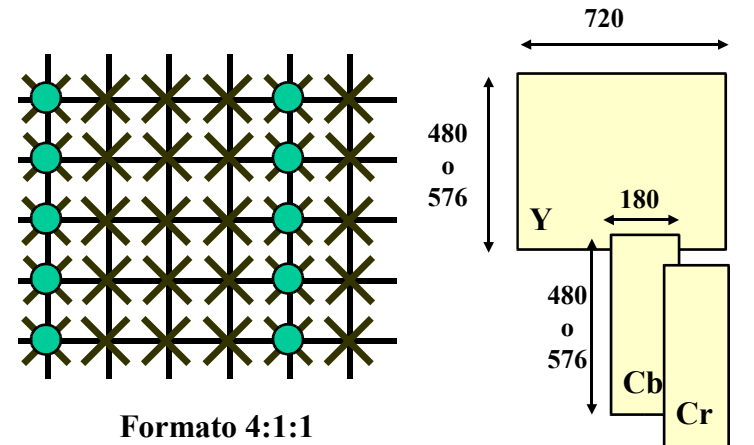
Formato 4:4:4



Formato 4:2:2



Formato 4:2:0



Formato 4:1:1

2.2 Tipos de imagen (según su resolución)

La resolución de una imagen se mide según el número de píxels por lado (ancho x alto).

En cámaras digitales se suele medir en Megapixels (millones de píxels por imagen)

- 1) **Common Intermediate Format (CIF)** (352x288): Utilizado habitualmente en videoconferencia (junto con Quarter CIF)
- 2) **VGA** (640x480): Usado por cámaras de baja calidad.
- 3) **n-Megapixels**: Ofrecido por cámaras de mayor calidad.

A veces, la resolución real de una cámara digital no coincide con la del CCD de esa misma cámara.

P.ej, una cámara de 3,3 MP ofrece una resolución de 2048x1536.

Parte de la circuitería del CCD que transporta los datos al ADC está situada en determinados diodos que no pueden ser usados.

Resumen

1. Introducción.
2. Características de la imagen.
 - 2.1 Captura y digitalización de imagen.
 - 2.2 Tipos de imagen (según su resolución)
3. **Compresión de imagen. Redundancia espacial.**
 - 3.1 **Estándar JPEG**
 - 3.2 Wavelets
 - EZW
 - JPEG 2000
4. Conclusiones.

3. Compresión de imagen.

Una imagen suele presentar redundancia espacial:

Redundancia espacial:

Las imágenes tienen información redundante susceptible de ser eliminada o reducida (por ejemplo, el color del cielo en una foto suele ser uniforme y azul :-).

El proceso de compresión de imagen consistirá en:

- 1) Eliminar en la medida de lo posible la redundancia espacial utilizando técnicas de *source encoding* (normalmente mediante transformada matemática).
- 2) Codificar los datos obtenidos en el paso anterior usando *entropy encoding* (elimina aun más la redundancia espacial).

Para conseguir mayores índices de compresión, este proceso será con pérdidas (cuantización de los datos).

3.1 Redundancia espacial: JPEG

Es un estándar ISO ('91) cuyo origen proviene del grupo JPEG (Joint Photographic Expert Group).

Codifica imágenes de tono-continuo

Dispone de cuatro modos de operación (incluyendo codificación sin pérdidas).

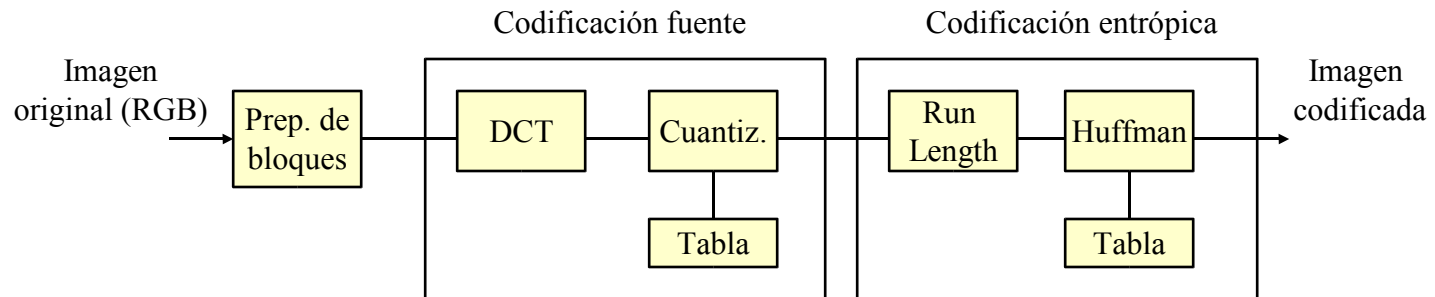
Se definen una serie de parámetros que permiten codificar las imágenes para obtener una gran variedad de calidades de compresión.

Factor de compresión ronda 20:1*

Es un sistema de codificación simétrico.

Forma parte de otros estándares de compresión de secuencias de vídeo (MPEG y H.26*).

Codificación JPEG en modo secuencial con pérdidas



Paso 1: Preparación de la imagen.

No define el formato de imagen original.

Podría ser RGB, YUV, YIQ, YCrCb, etc.

Convierte la imagen a formato YCbCr utilizando una reducción de color 4:1:1 (sub-sampling)

Ej.: RGB 640x480 (VGA): Y (640x480), Cb y Cr (320x240)

Se divide la imagen en bloques de 8x8 elementos

Ej. anterior: 4800 bloques Y, 1200 Cb y 1200 Cr.

Codificación JPEG: Transformada DCT

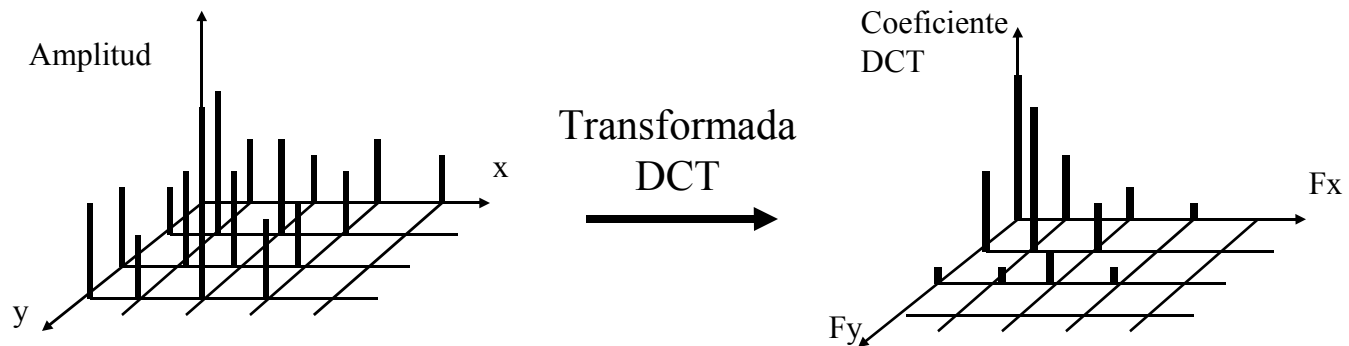
Paso 2: Transformada discreta del coseno (DCT).

Transforma un dominio de amplitudes al dominio de la frecuencia.

Las componentes frecuenciales más altas son susceptibles de ser eliminadas (percepción visual)

Se aplica esta transformada a cada bloque de 8x8 obteniendo la matriz de coeficientes DCT asociada

Componente (0,0): el nivel de continua DC del bloque (Media)



Codificación JPEG: Transformada DCT(II)

DCT-1D: (vector 8 elementos)

$$S(u) = \frac{C(u)}{2} \sum_{x=0}^7 s(x) \cos\left[\frac{(2x+1)u\pi}{16}\right]$$

$$s(x) = \sum_{u=0}^7 \frac{C(u)}{2} S(u) \cos\left[\frac{(2x+1)u\pi}{16}\right]$$

DCT-2D: (matriz 8x8 elementos)

$$DCT(i, j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} pixel(x, y) \cos\left[\frac{(2x+1)i\Pi}{2N}\right] \cos\left[\frac{(2y+1)j\Pi}{2N}\right]$$

$$pixel(x, y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j) DCT(i, j) \cos\left[\frac{(2x+1)i\Pi}{2N}\right] \cos\left[\frac{(2y+1)j\Pi}{2N}\right]$$

**Transformada
directa**

$$\text{Si } u = 0, C(u) = \frac{1}{\sqrt{2}}$$

$$\text{Si } u = 1, C(u) = 1$$

**Transformada
inversa**

Codificación JPEG: Cuantificación

Paso 3: Cuantificación (*quantization*).

Se eliminan los coeficientes menos representativos de la DCT (transformación con pérdidas).

Cada coeficiente de la matriz 8x8 es dividido por un valor almacenado en una tabla (*quantization table*).

El estándar sugiere dos tablas una para la componente Y y otra para las componentes Cb y Cr.

Estas tablas se pueden escalar con otro parámetro Q que nos permitirá ajustar el índice de compresión requerido.

150	70	38	16	4	0	1	0
88	56	22	9	2	0	0	0
21	34	12	4	0	0	0	0
4	6	3	7	0	1	0	0
1	0	5	0	2	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Coeficientes DCT

Tabla de cuantificación

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

150	70	19	4	0	0	0	0
88	56	11	8	0	0	0	0
10	17	6	1	0	0	0	0
1	1	1	2	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Coeficientes DCT cuantificados

Codificación JPEG: Codificación entropía

Paso 4: Codificación DPCM de los componentes DC de cada bloque.

Bloques sucesivos tienen un valor medio muy similar.

Paso 5: Codificación *run-length* de todos los componente de un bloque.

Se hace un barrido “zig-zag” con el fin de agrupar todos los componentes nulos.

150	70	19	4	0	0	0	0
88	56	11	8	0	0	0	0
10	17	6	1	0	0	0	0
1	1	1	2	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

150-70-88-10-56-19-4-11-17-1-0-1-6-8-0-0-0-1-1-0-0-0-0-0-2-...(39 0's)



150-70-88-10-56-19-4-11-17-1-0-1-6-8-A0/3-A1/2-A0/5-2-A0/39

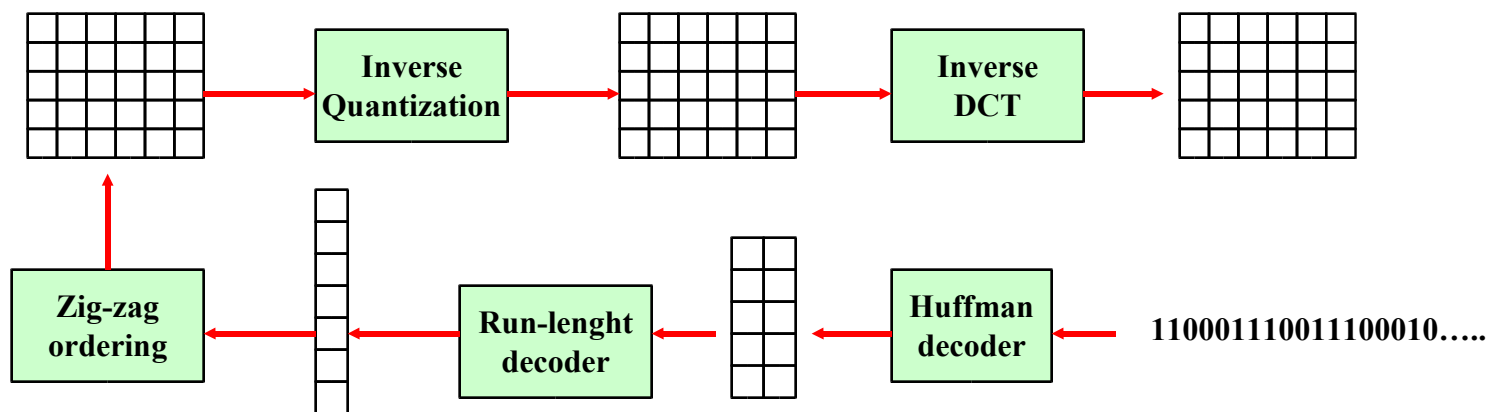
Codificación JPEG: Codificación entropía

Paso 6: Codificación estadística VLC: *Huffman*

A lo obtenido en el paso anterior se aplica el algoritmo de Huffman para comprimir aún más la información.

El resultado de este paso es lo que debemos enviar o almacenar.

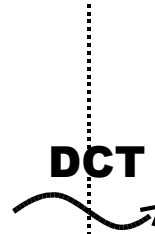
La decodificación JPEG consiste en realizar el proceso inverso:



Codificación JPEG: Ejemplo real (Quant)

40	44	47	40	40	55	79	75
44	52	40	47	40	48	67	79
52	55	36	67	63	62	52	72
68	45	56	60	52	55	36	60
62	48	56	48	40	36	47	62
47	67	40	55	55	40	36	62
36	56	23	67	62	44	49	47
48	55	36	55	52	47	47	36

Bloque de muestras (pixels)



411	-18	14	-8	24	-10	-14	-18
20	-34	27	-9	-11	11	14	7
-11	-23	-1	5	-19	4	-20	-2
-8	-5	14	-14	-8	-3	-3	9
-3	9	7	2	-10	17	18	16
3	-2	-17	8	7	-3	1	-8
8	1	-2	3	-2	-7	-1	-2
1	-8	-4	2	2	3	-7	2

Bloque de muestras transformadas



39	49	38	47	45	55	70	76
50	49	38	46	45	57	65	72
54	46	43	55	54	60	53	60
58	47	50	59	55	57	43	55
64	52	51	54	45	48	41	60
56	52	51	55	44	45	42	55
43	50	47	58	51	49	43	42
42	50	40	53	50	51	48	39

Bloque recuperado de muestras



102	-2	1	0	1	0	0	-1
2	-4	2	0	0	0	0	0
-1	-2	0	0	-1	0	-1	0
0	0	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Bloque de muestras cuantizadas

Codificación JPEG: Ejemplo real I (RLE+VLC)

Codificación RLE+VLC de los coeficientes cuantizados

102	-2	1	0	1	0	0	-1
2	-4	2	0	0	0	0	0
-1	-2	0	0	-1	0	-1	0
0	0	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Bloque de muestras cuantizadas

Número de bits	Código
0	100
1	00
2	01
3	1
4	1
5	1110
6	1111 0
7	1111 10
8	1111 110
9	1111 1110
10	1111 1111 0
11	1111 1111 1

Tabla para la DC

Paso 1. Se codifica la DC usando codificación diferencial DPCM

- Si DC Bloque anterior es 98 → codificar 102-98
- Se codifica como:
 Num. bits necesarios (tabla VLC) + codif + signo
- DC se codifica como: 101 100 0

Codificación JPEG: Ejemplo real II (RLE+VLC)

102	-2	1	0	1	0	0	-1
2	-4	2	0	0	0	0	0
-1	-2	0	0	-1	0	-1	0
0	0	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Bloque de muestras cuantizadas

Paso 2: Se codifica en zig-zag pares <Run (cuenta de ceros), coeficientes>

Run	Nivel	Código
EOB		10
0	1	11s
0	2	0100 s
0	3	0010 1s
0	4	0000 110s
0	5	0010 0110 s
	...	
1	1	011s
1	2	0001 10s
1	3	0010 0101 s
1	4	0000 0011 00s
	...	
2	1	0101 s
2	2	0000 100s
	...	
Escape		0000 01

Tabla para pares <Run, Niveles>

Run (Num. De Ceros)	Valores	Código VLC
0	-2	0100 1
0	2	0100 0
0	-1	111
0	-4	0000 1101
0	1	110
1	2	0001 100
	...	

Parte del bloque codificado con VLC

Existe código de escape:
0000 01 **RRRRR** **RR** **NNNN** **NNNN**

Codificación JPEG: Ejemplo real III (RLE+VLC)

Resultado final de la codificación RLE+VLC

10	2	1	0	1	0	0	-1
2	4	2	0	0	0	0	0
-1	-2	0	0	-1	0	-1	0
0	0	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Bloque de muestras cuantizadas



Run (Num. De Ceros)	Valores	Código VLC
N/A	4	101 1000
0	-2	0100 1
0	2	0100 0
0	-1	111
0	-4	0000 1101
0	1	110
1	2	0001 100
0	-2	0100 1
5	1	0001 110
3	1	0011 10
5	-1	0001 111
0	-1	111
2	-1	0101 1
4	-1	0011 01
7	-1	0001 001
EOB		10

Bloque codificado con VLC

Tasa de compresión:

Stream final: **1011000010010100011100001101** ... 0001001**10** (85 bits)

Bits por píxel: (Núm bits/ Núm píxels) $85/64 = 1'33$ bpp

Factor de compresión:

Tam_comprimida:Tam_original = $85:(8*8*8) = 85:512$

$1:Tam_original/Tam_comprimida = (85/85):(512/85) = 1:6$

Codificación JPEG: Ejemplo real IV (Calidad)

40	44	47	40	40	55	79	75
44	52	40	47	40	48	67	79
52	55	36	67	63	62	52	72
68	45	56	60	52	55	36	60
62	48	56	48	40	36	47	62
47	67	40	55	55	40	36	62
36	56	23	67	62	44	49	47
48	55	36	55	52	47	47	36

Bloque de muestras (pixels)

39	49	38	47	45	55	70	76
50	49	38	46	45	57	65	72
54	46	43	55	54	60	53	60
58	47	50	59	55	57	43	55
64	52	51	54	45	48	41	60
56	52	51	55	44	45	42	55
43	50	47	58	51	49	43	42
42	50	40	53	50	51	48	39

Bloque recuperado de muestras

Medida objetiva del error:
MSE (*Mean Square Error*)

$$MSE = \frac{1}{N} \sum_{i=1}^N \left(s(i) - s'(i) \right)^2$$

Medida objetiva de la calidad:
PSNR (*Peak SNR*)

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}$$

Valores del ejemplo:

$$MSE = 49'53$$

$$PSNR = 31'18 \text{ dB}$$

Resumen.

1. Introducción.
2. Características de la imagen.
 - 2.1 Captura y digitalización de imagen.
 - 2.2 Tipos de imagen (según su resolución)
3. Compresión de imagen. Redundancia espacial.
 - 3.1 Estándar JPEG
 - 3.2 **Wavelets**
 - EZW
 - JPEG 2000
4. Conclusiones.

3.1 Redundancia espacial: Wavelets.

Problemas con la DCT

Efectos de borde en los bloques

Base ortonormal fija
(cosenos a distinta frec)

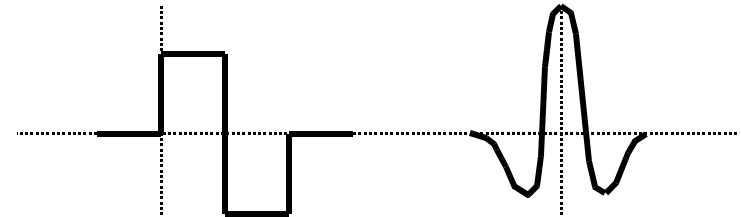


Se basa en un conjunto de funciones básicas que se derivan de una función prototipo (“madre”).

Estas funciones son escalados y desplazamientos de la función prototipo.

Permiten analizar regiones de la señal con diferente detalle (resolución).

Analiza de forma independiente las bajas y altas frecuencias de la señal (realmente actúa como un filtro).

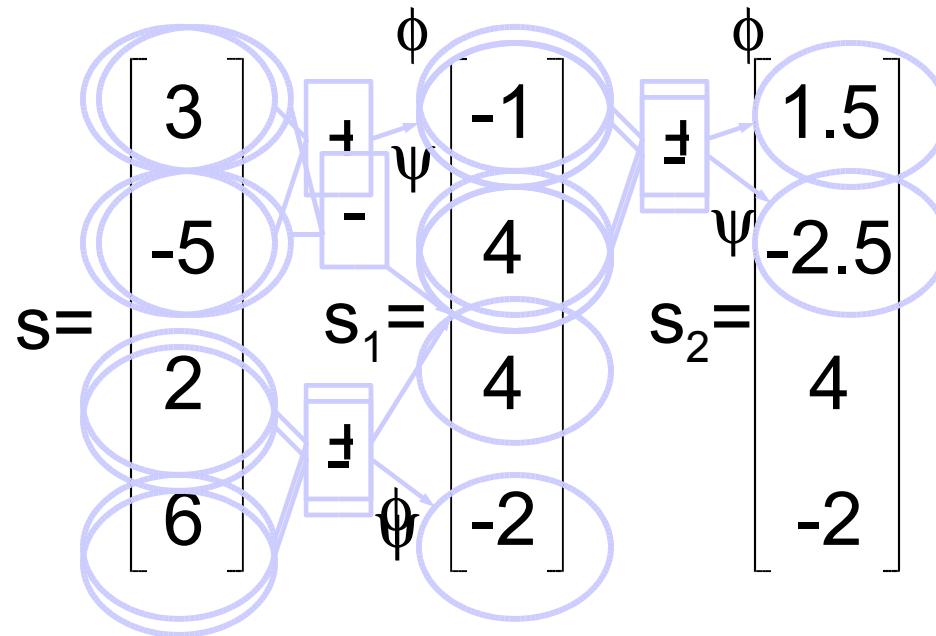


Descomposición Wavelet: ejemplo.

Dos funciones:

Función generadora del espacio wavelet (ψ)

Función de escalado (ϕ) que define la señal en el espacio original a distintas escalas



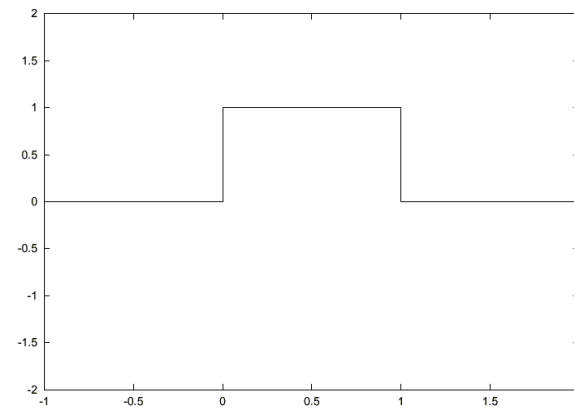
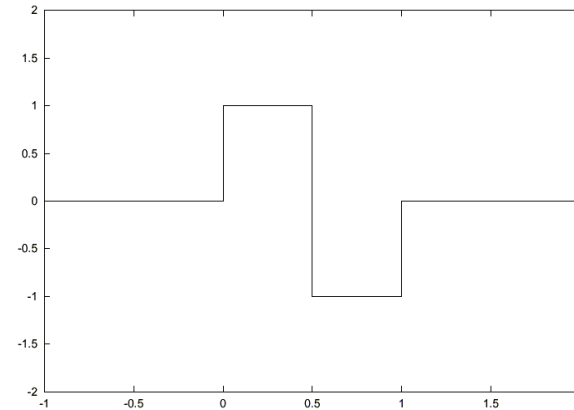
Versión ortogonal de la Haar Wavelet

Una familia de wavelets: Haar wavelet I

$$\psi(t) = \begin{cases} 1, & \text{si } 0 \leq t < 0.5 \\ -1, & \text{si } 0.5 \leq t < 1 \\ 0, & \text{en otro caso} \end{cases}$$

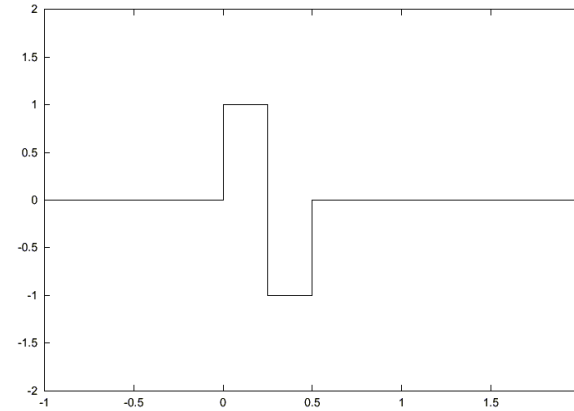
$$\psi(t) = \phi(2t) - \phi(2t-1)$$

$$\phi(t) = \begin{cases} 1, & \text{si } 0 \leq t \leq 1 \\ 0, & \text{en otro caso} \end{cases}$$

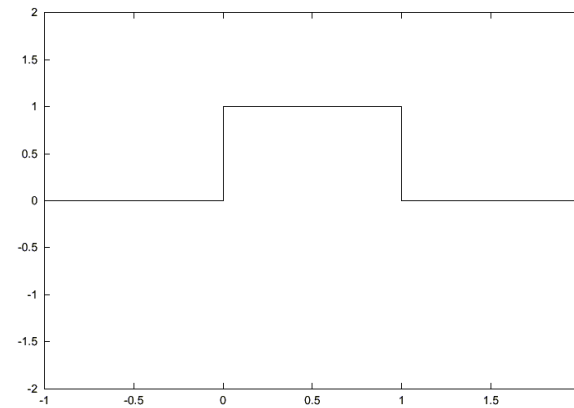


Una familia de wavelets: Haar wavelet II

$$\psi_{1,0}(t) = \psi(2t)$$



$$\phi(t) = \begin{cases} 1, & \text{si } 0 \leq t \leq 1 \\ 0, & \text{en otro caso} \end{cases}$$



Una familia de wavelets: Haar wavelet III

$$\psi_{1,0}(t) = \psi(2t)$$

$$\psi_{1,1}(t) = \psi(2t - 1)$$

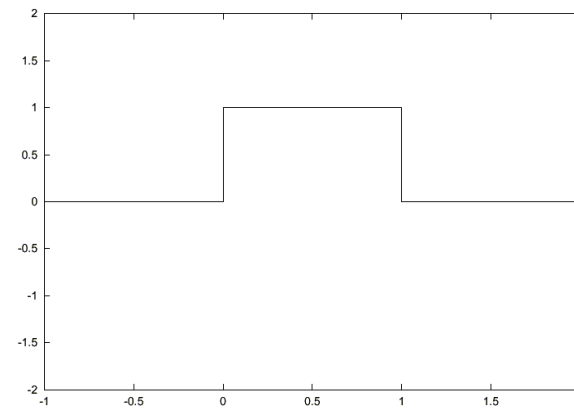
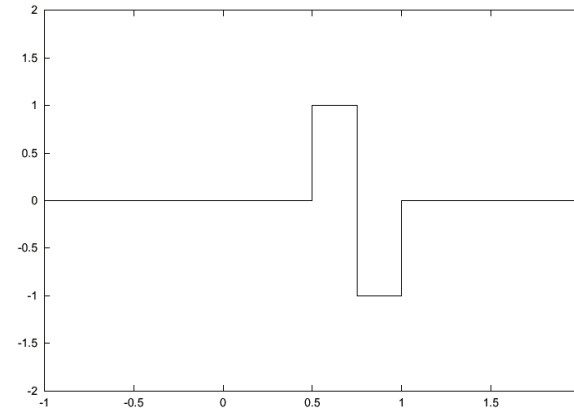
$$\psi_{n,k}(t) = \sqrt{2^n} \psi(2^n t - k),$$

$$0 \leq k \leq 2^n - 1$$

$$\phi(t) = \begin{cases} 1, & \text{si } 0 \leq t \leq 1 \\ 0, & \text{en otro caso} \end{cases}$$

n: Escala

k: Desplazamiento



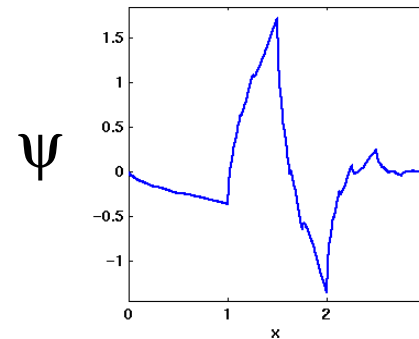
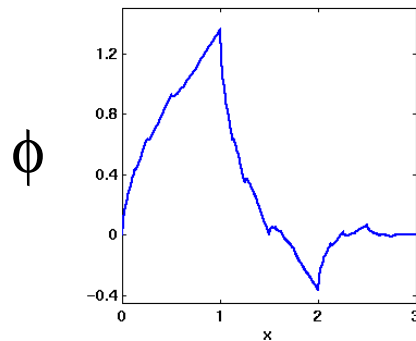
Relaciona las funciones wavelets de distintos niveles

$$\phi(t) = \sum_k h_k \sqrt{2} \phi(2t - k) \quad \psi(t) = \sum_k (-1)^k h_{1-k} \sqrt{2} \phi(2t - k)$$

Permite definir familias de funciones wavelets a partir de los coeficientes $\{h_k\}$ (usados como filtros)

Ej.: Daubechies4

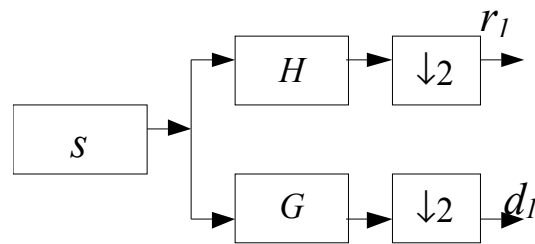
$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} \quad h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \quad h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$



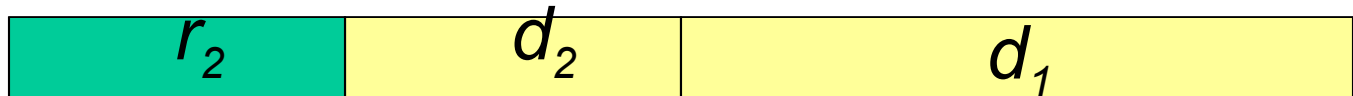
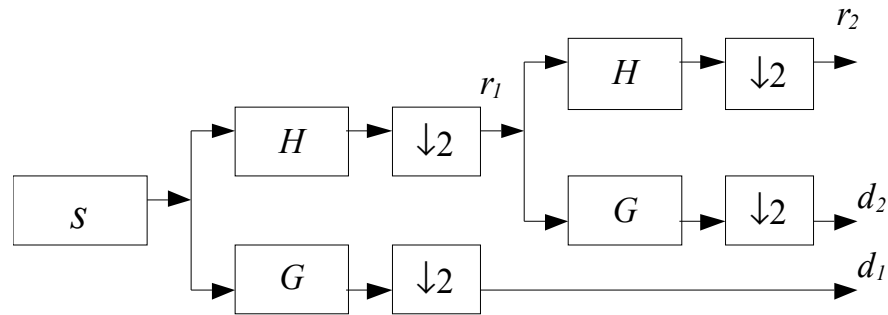
Aplicación de wavelets sobre señales 1D (I)



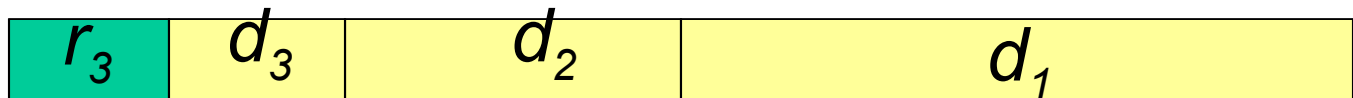
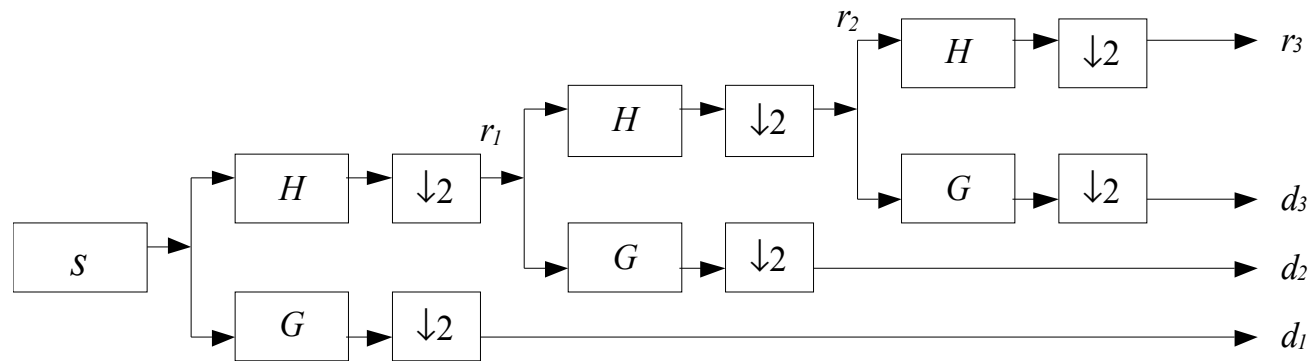
Aplicación de wavelets sobre señales 1D (II)



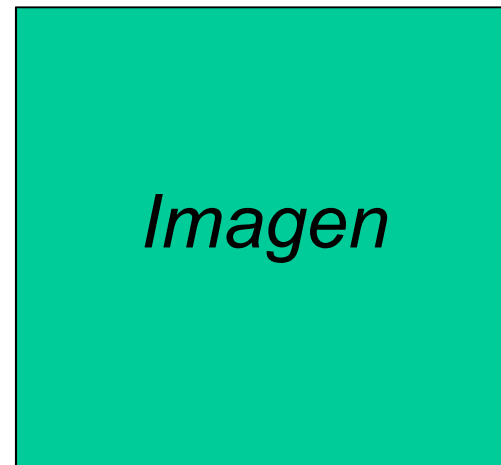
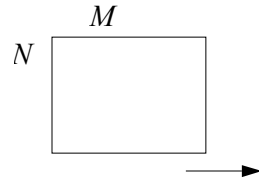
Aplicación de wavelets sobre señales 1D (III)



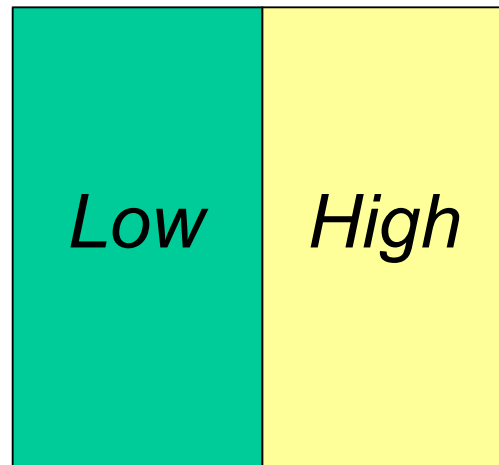
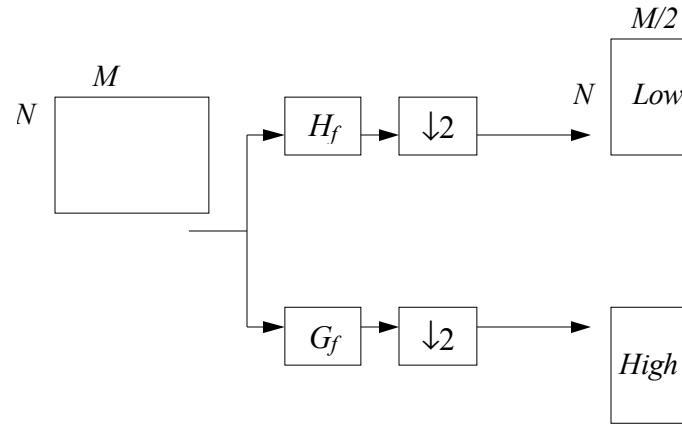
Aplicación de wavelets sobre señales 1D (IV)



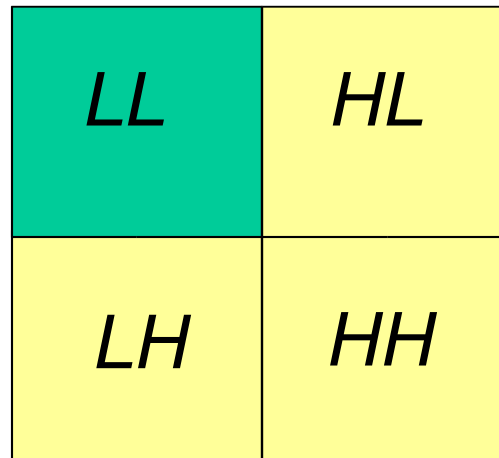
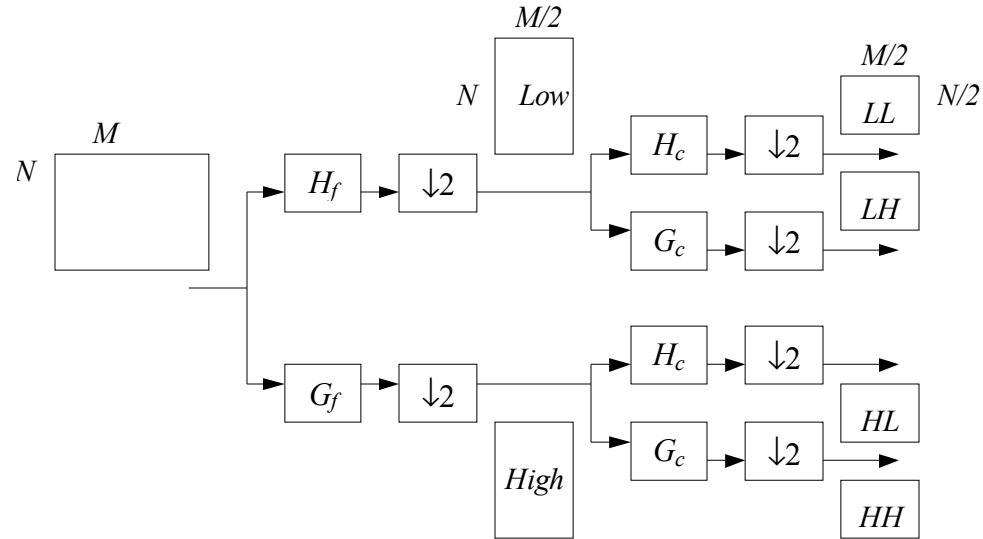
Aplicación de wavelets sobre imágenes I



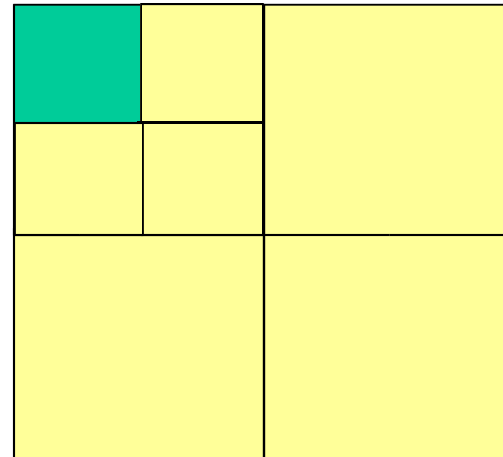
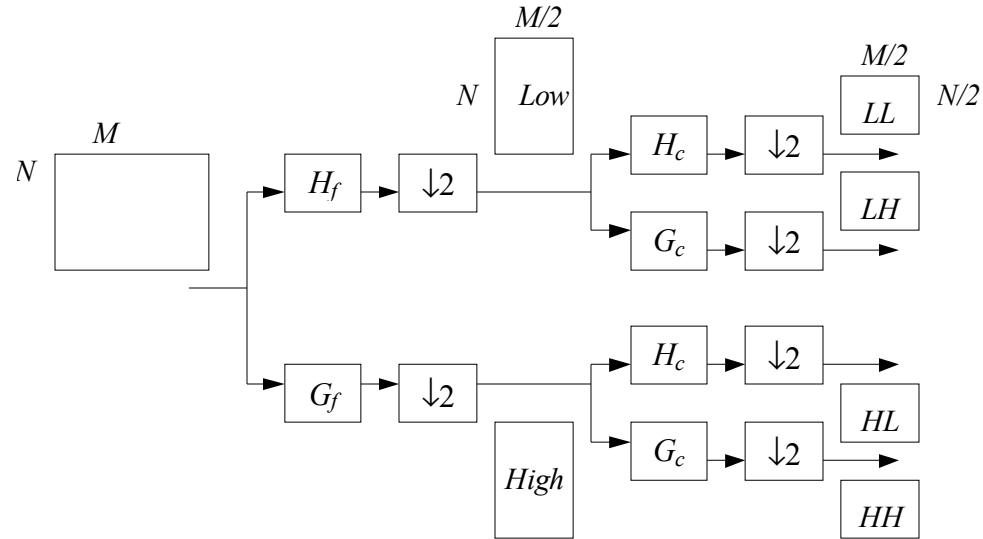
Aplicación de wavelets sobre imágenes II



Aplicación de wavelets sobre imágenes II



Aplicación de wavelets sobre imágenes IV



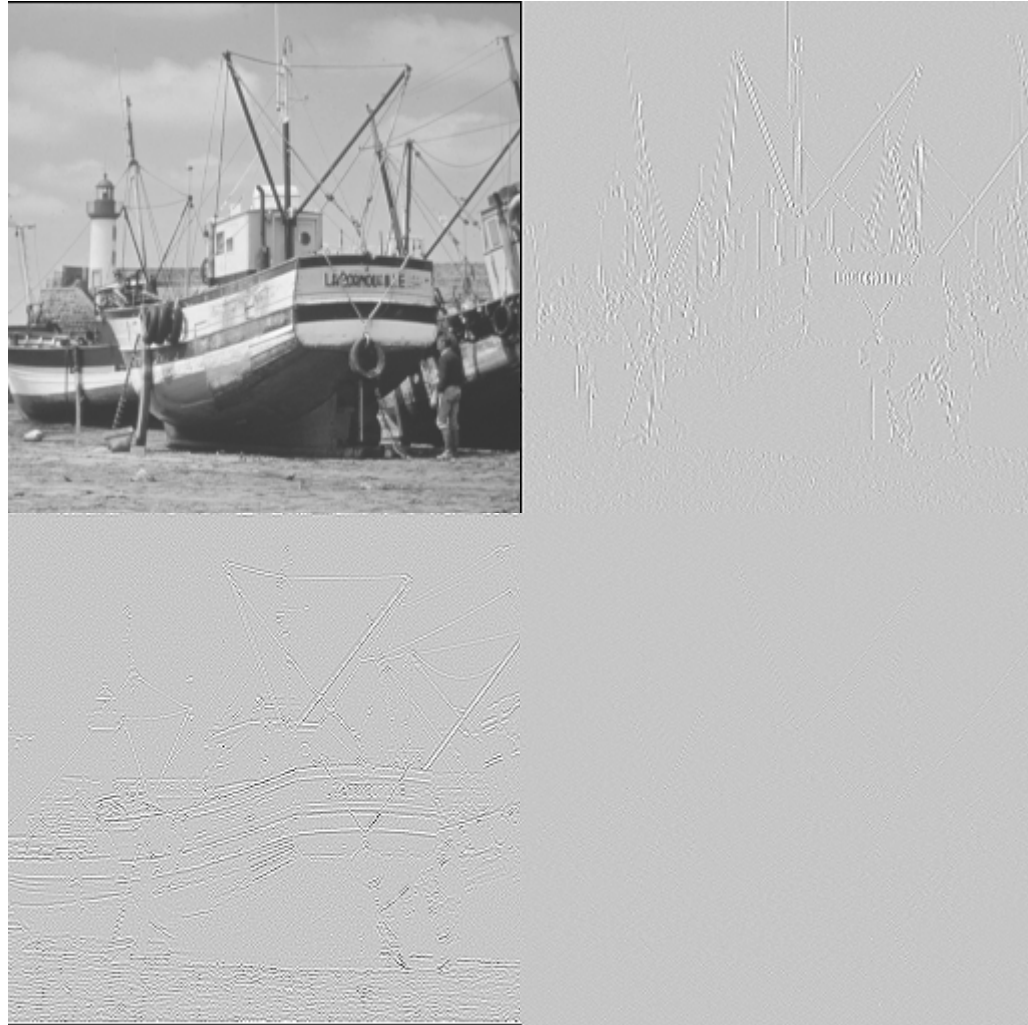
Aplicación sobre imágenes, ejemplo I



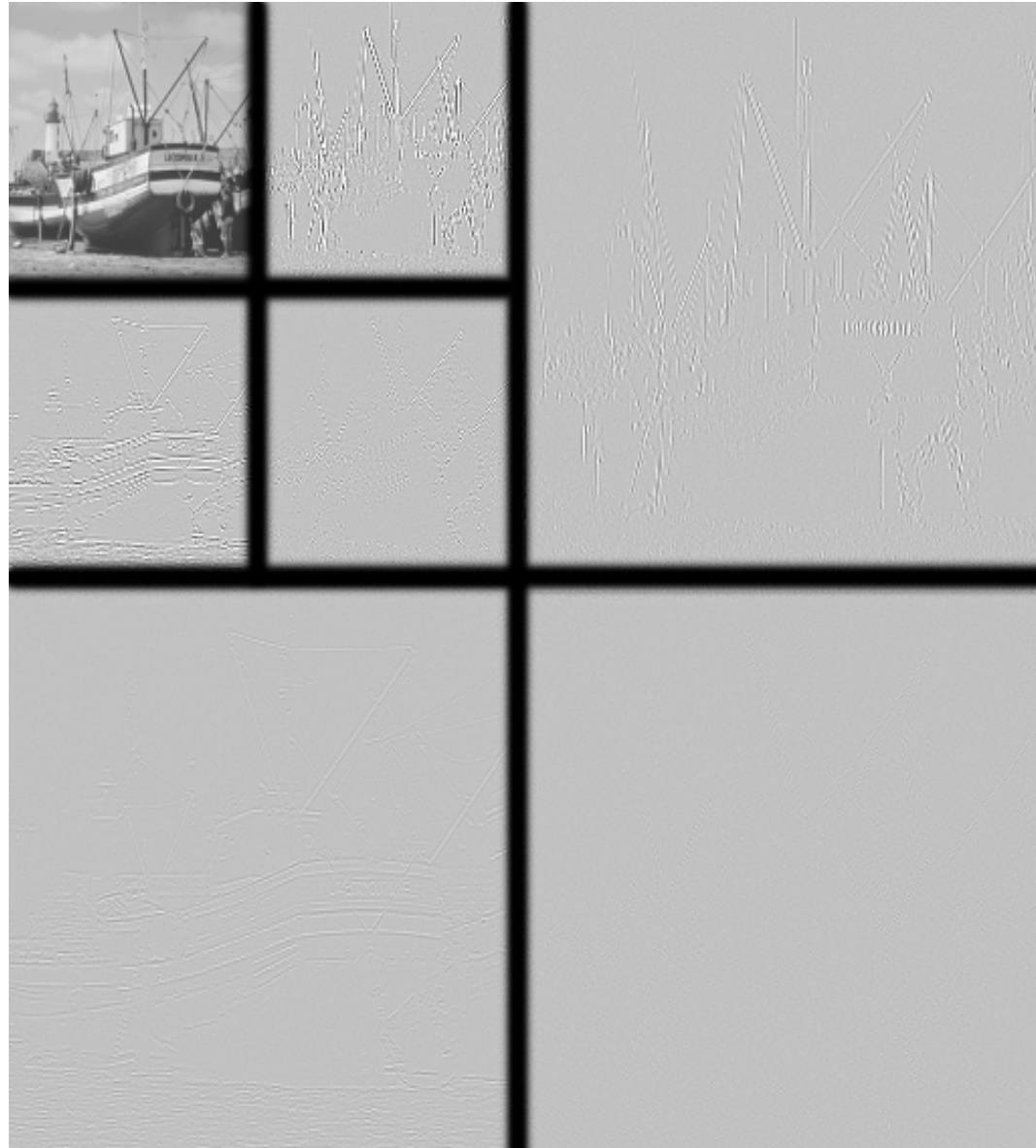
Aplicación sobre imágenes, ejemplo II



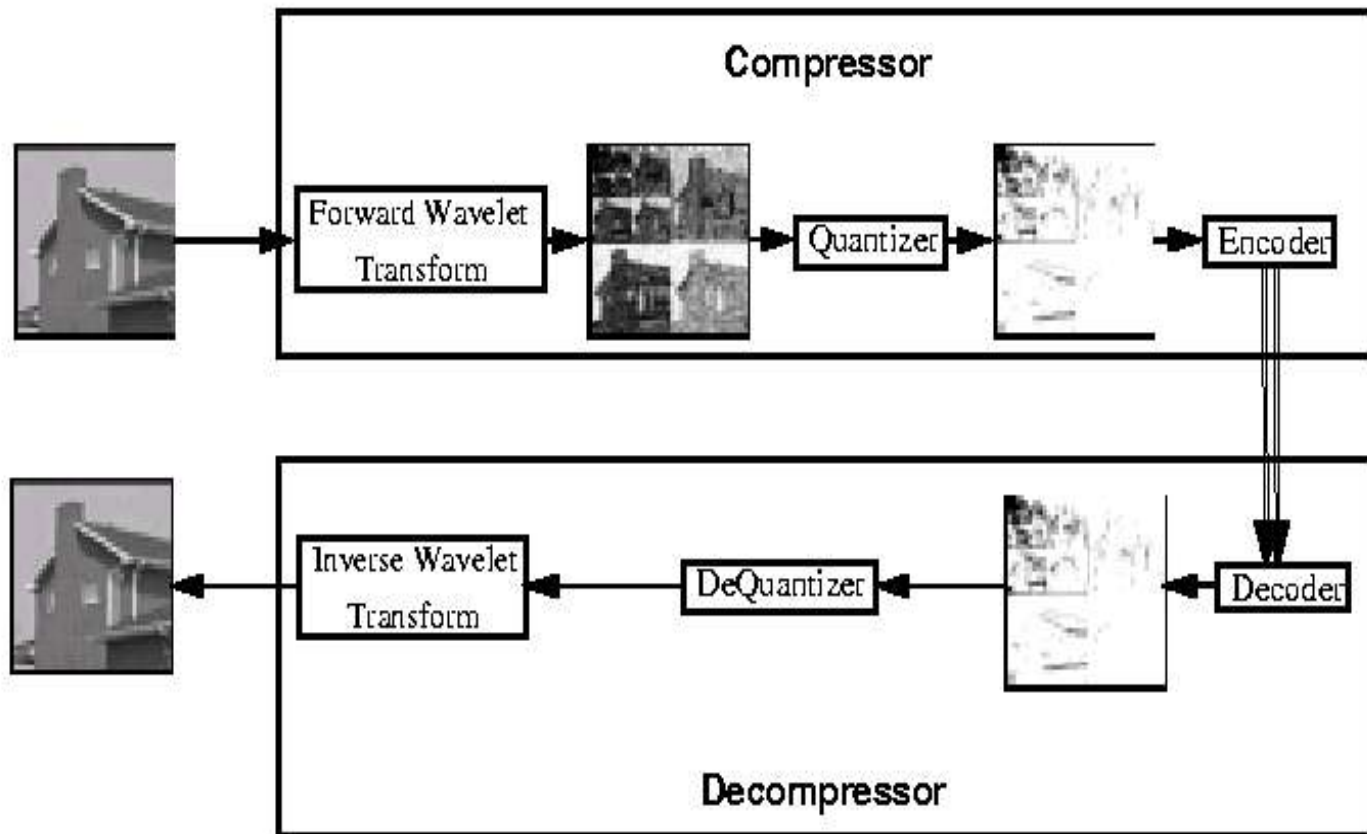
Aplicación sobre imágenes, ejemplo III



Aplicación sobre imágenes, ejemplo IV



Wavelets: Estructura de un codec.



Resumen.

1. Introducción.
2. Características de la imagen.
 - 2.1 Captura y digitalización de imagen.
 - 2.2 Tipos de imagen (según su resolución)
3. Compresión de imagen. Redundancia espacial.
 - 3.1 Estándar JPEG
 - 3.2 Wavelets

EZW

JPEG 2000
4. Conclusiones.

Una vez descompuesta la imagen en sub-bandas, se procede a cuantificar los coeficientes wavelets.

La mayor parte de la energía está concentrada en las bandas de menor frecuencia.

Existe una relación clara entre los coeficientes de la misma posición espacial en las diferentes bandas.

Algoritmos propuestos:

Embedded Zero-tree Wavelet (EZW)

Set Partitioning In Hierarchical Trees (SPIHT)

Stack-Run

Cuantificación conjunta: espacial y en frecuencia.

Lower-Tree Wavelet (LTW)

Wavelets: Algoritmo EZW.

Basado en la definición de árboles de coeficientes

Hay coeficientes en diferentes subbandas que representan la misma posición espacial en la imagen.

En imágenes naturales, la mayor parte de energía se centra en las bandas de menor frecuencia.

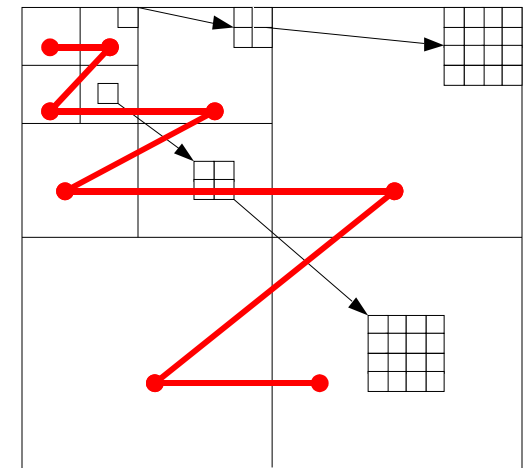
Cuanto más cerca del nodo raíz está coeficiente, mayor magnitud tiene

Si un nodo es menor que un umbral, posiblemente sus descendientes también lo serán.

El algoritmo realiza aproximaciones sucesivas (planos de bits) con dos plano

Utiliza un codificador aritmético

4 símbolos: **sp**, **sn**, **zr**, **iz**



pasos por

INICIALIZACION:

$$n = \lceil \log_2 c_{\max} \rceil$$

n: número de bits necesarios para codificar el mayor coeficiente.

PASO DOMINANTE (se almacena el mapa para localizar coeficientes)

1) Los coeficientes que necesitan exactamente **n** bits para ser codificados se etiquetan como significantes (positivo **sp** o negativo **sn**), se añaden a una lista de significantes y ya no se procesan en otros pasos dominantes.

2) El resto de coeficientes (que necesitan menos de **n** bits):

Si todos sus descendientes también necesitan menos de **n** bits, se etiqueta todo el árbol como raíz de un árbol de ceros (**zr**)

Sino, se etiqueta ese coeficiente como cero aislado (**iz**)

PASO SUBORDINADO (se almacenan los bits de los coeficientes)

Se codifica el **bit n** de los coeficientes de la lista de significantes

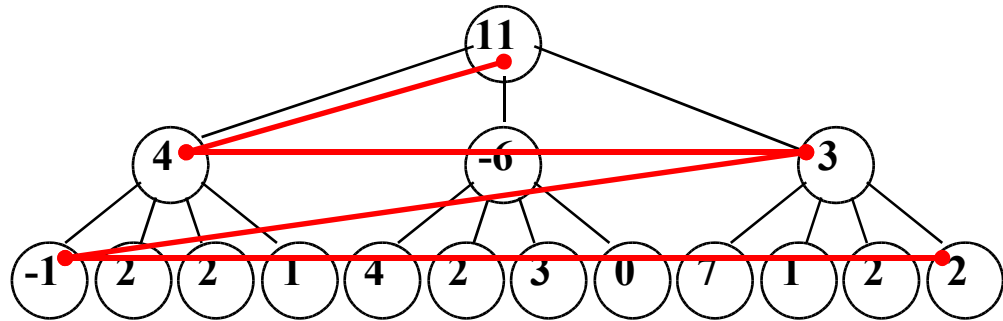
Si no se ha alcanzado la tasa de bits deseada (embedded):

Se decrementa **n** en uno (vamos al siguiente bit de menor peso)

Procedemos a realizar una nueva iteración (pasos dominante y subordinado)

Wavelets: Algoritmo EZW. Ejemplo I

11	4	-1	2
-6	3	2	1
4	2	7	1
3	0	2	2



n = 3

1

1011	0100	0001	0010
0110	0011	0010	0001
0100	0010	0111	0001
0011	0000	0010	0010

Símbolos = {sp,zr,zr,zr}
 Bits={}
 Lista signif = {}

Nº bits: 4*2+0 = 8

n = 2

2

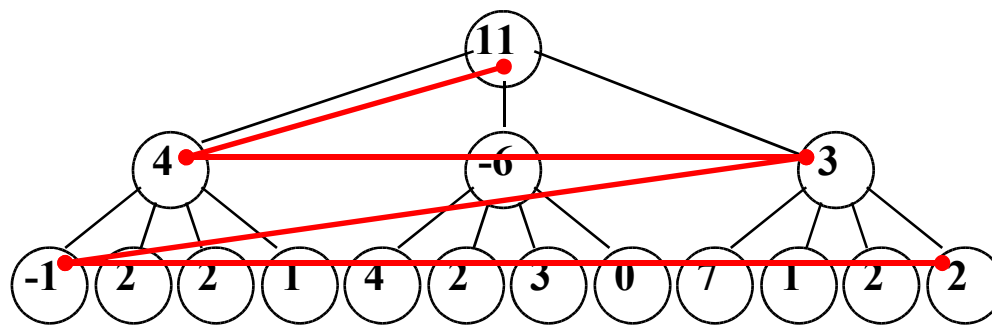
*	0100	0001	0010
0110	0011	0010	0001
0100	0010	0111	0001
0011	0000	0010	0010

Símbolos = {sp,zr,zr,zr, sp,sn,iz,
 zr,zr,zr,zr, sp,zr,zr,zr, sp,zr,zr,zr}
 Bits={0}
 Lista signif = {0, 1}

Nº bits: 19*2+1 = 39

Wavelets: Algoritmo EZW. Ejemplo II

11	4	-1	2
-6	3	2	1
4	2	7	1
3	0	2	2



n = 1

*	*	0001	0010
*	0011	0010	0001
*	0010	*	0001
0011	0000	0010	0010

3

Símbolos = {sp,zr,zr,zr, sp,sn,iz,
zr,zr,zr,zr, sp,zr,zr,zr, sp,zr,zr,zr,
sp, zr,sp,sp,zr, sp,sp,zr, zr,sp,sp}

Bits={0, 1, 0, 1, 0, 1}

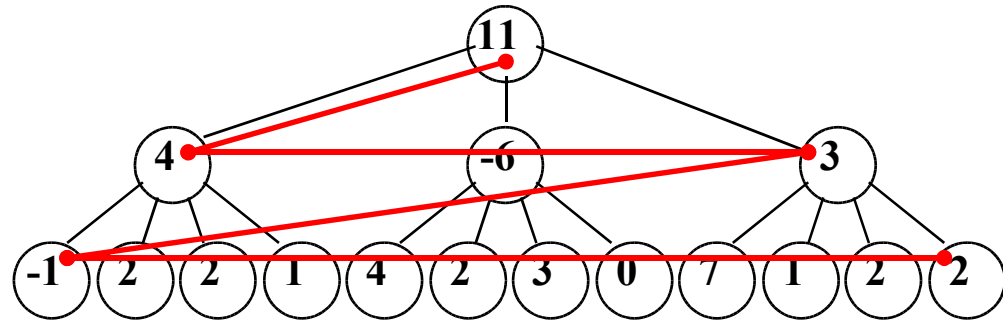
Lista signif = {

101,
010,
010,
000,
011
}

Nº bits: $30 \cdot 2 + 6 = 66$

Wavelets: Algoritmo EZW.

11	4	-1	2
-6	3	2	1
4	2	7	1
3	0	2	2



¿Cuándo dejamos de codificar?

Cuando terminamos con la cuota de bits

Varias opciones a la hora de que el decodificador complete los bits no transmitidos:

Relleno de ceros (p.ej 101xxx 101000)

Relleno de unos (p.ej 101xxx 101111)

Reducir el intervalo de error (p.ej 101xxx 101100)

Resumen.

1. Introducción.
2. Características de la imagen.
 - 2.1 Captura y digitalización de imagen.
 - 2.2 Tipos de imagen (según su resolución)
3. Compresión de imagen. Redundancia espacial.
 - 3.1 Estándar JPEG
 - 3.2 Wavelets
 - EZW
 - JPEG 2000**
4. Conclusiones.

3.1 Redundancia espacial: JPEG 2000.

Estándar ISO/ITU ('01) que amplía y mejora el anterior JPEG.

Mejora la compresión usa transformada Wavelet

Hay versión con pérdidas y sin pérdidas.

Permite multiresolución (gracias al análisis Wavelet).

Es *embedded* (enclavado)

- Decodificación progresiva

- Escalable SNR

Permite dividir la imagen en bloques de cualquier tamaño (no solo 8x8) (incluso un único bloque)

Permite codificar Regiones de Interés (ROI)

Es más robusto frente a errores.

Inconvenientes del EZW:

Debido a las estructuras de árboles, los errores se propagan de unas subbandas a otras → Menor robustez frente a errores.

Debido a la codificación progresiva por planos de bits, no hay escalabilidad espacial (sólo de calidad SNR).

JPEG 2000 corrige estos inconvenientes:

Codifica las distintas subbandas por bloques y de forma independiente, sin utilizar árboles (algoritmo *Embedded Block Coding with Optimized Truncation*, EBCOT).

Se utiliza un nuevo algoritmo para suplir la pérdida de eficiencia al no aprovechar la redundancia existente entre subbandas.

Se introduce una etapa de reordenación de coeficientes que permite todo tipo de escalabilidad (al no haber árboles los coeficientes se reordenan con mayor facilidad).

Estructura de un codificador JPEG 2000:

Una imagen se puede dividir en bloques denominados *tiles* (normalmente hay sólo uno).

A cada *tile* se le aplica el algoritmo JPEG2000:

Se calcula su transformada Wavelet.

Se cuantizan los coeficientes.

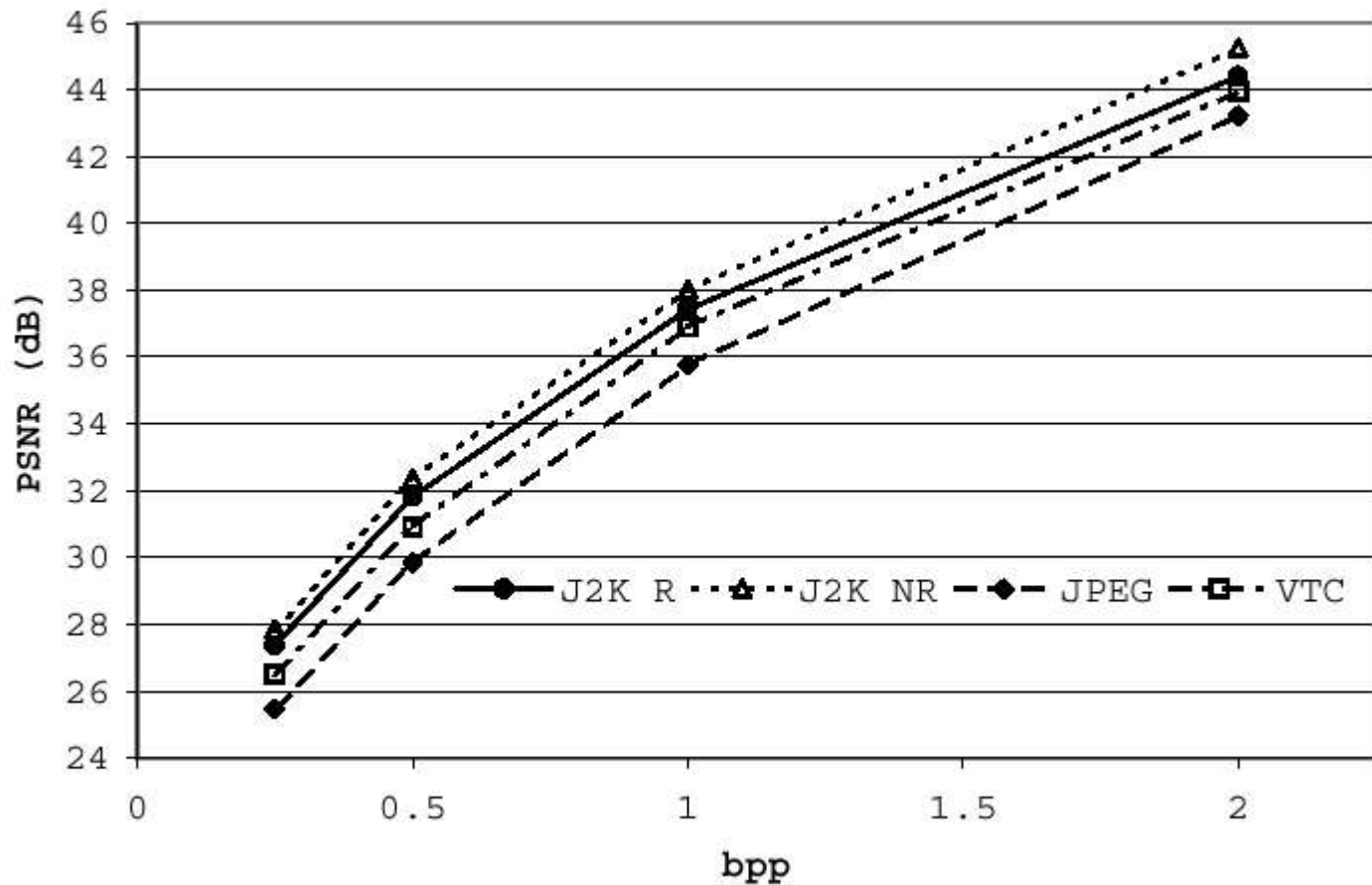
Se aplica el algoritmo EBCOT por bloques (de 32x32 ó 64x64 coeficientes) (*tier 1 coding*)

Se organiza el bit stream (*tier 2 coding*) según la escalabilidad deseada.

- Escalabilidad espacial: Se sitúan primero las subbandas de menor frecuencia.
- Escalabilidad SNR o en calidad: Se codifica primero los planos de bits de los coeficientes de mayor magnitud.

JPEG 2000: Comparativa JPEG-JPEG2000

Comparación objetiva

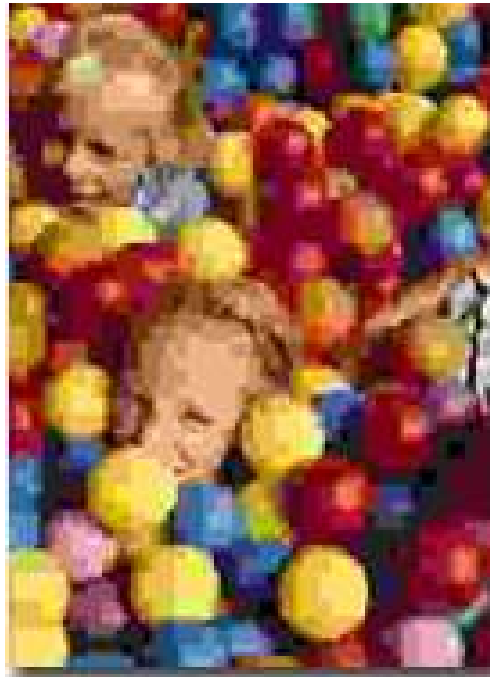


JPEG 2000: Comparativa JPEG-JPEG2000

Comparación subjetiva



Original: 700 kbytes



**Comprimida con JPEG
6,1 kbytes - 1:115**



**Comprimida con JPEG 2000
6,1 kbytes - 1:115**

Resumen.

1. Introducción.
2. Características de la imagen.
 - 2.1 Captura y digitalización de imagen.
 - 2.2 Tipos de imagen (según su resolución)
3. Compresión de imagen. Redundancia espacial.
 - 3.1 Estándar JPEG
 - 3.2 Wavelets
 - EZW
 - JPEG 2000
4. Conclusiones.