

Compresión de Imágenes

Redundancia en Imágenes

Tipos

- Redundancia de Código
- Redundancia Interpíxel
- Redundancia Psicovisual

Comprimir en imágenes es el proceso por el que se elimina alguna de estas redundancias.

Las formas de compresión en imágenes se pueden dividir en:

- aquellas que no produce errores en la imagen;
- aquellas que sí lo producen.

Los procesos de compresión que eliminan la redundancia de código y/o entre píxeles no producen errores. (Lossless)

Los procesos de compresión que eliminan la redundancia psicovisual, siempre producen errores. (Lossy)

Casos de Uso de Compresión

En muchas aplicaciones sólo es tolerable la ***compresión sin error***:

- Imágenes médicas, satelitales, documentos legales
- Transacciones bancarias, archivos binarios, ejecutables

En otras ***se pueden tolerar pérdidas de información y obtener grandes compresiones***:

- Internet, multimedia, etc.

Tipos de Compresión

- Compresión sin pérdida (**lossless**) o reversible: la imagen (o secuencia) reconstruida es idéntica a la original.
Ejemplos winzip, winrar.
En imágenes: GIF y PNG.
- Compresión con pérdida (**lossy**) o irreversible: la imagen (o secuencia) reconstruida es más o menos diferente de la original.
Ejemplos en imágenes: jpeg, jpeg2000.
En video: MOV (Apple Quicktime) y AVI (Video para Windows).

Cuantificar la compresión: Razón de compresión

número de bits en la imagen original/ número de bits en la comprimida.

Ejemplo: 3:1

Compresión de Datos

Proceso mediante el que se reduce la cantidad de datos requerido para almacenar una determinada cantidad de información.

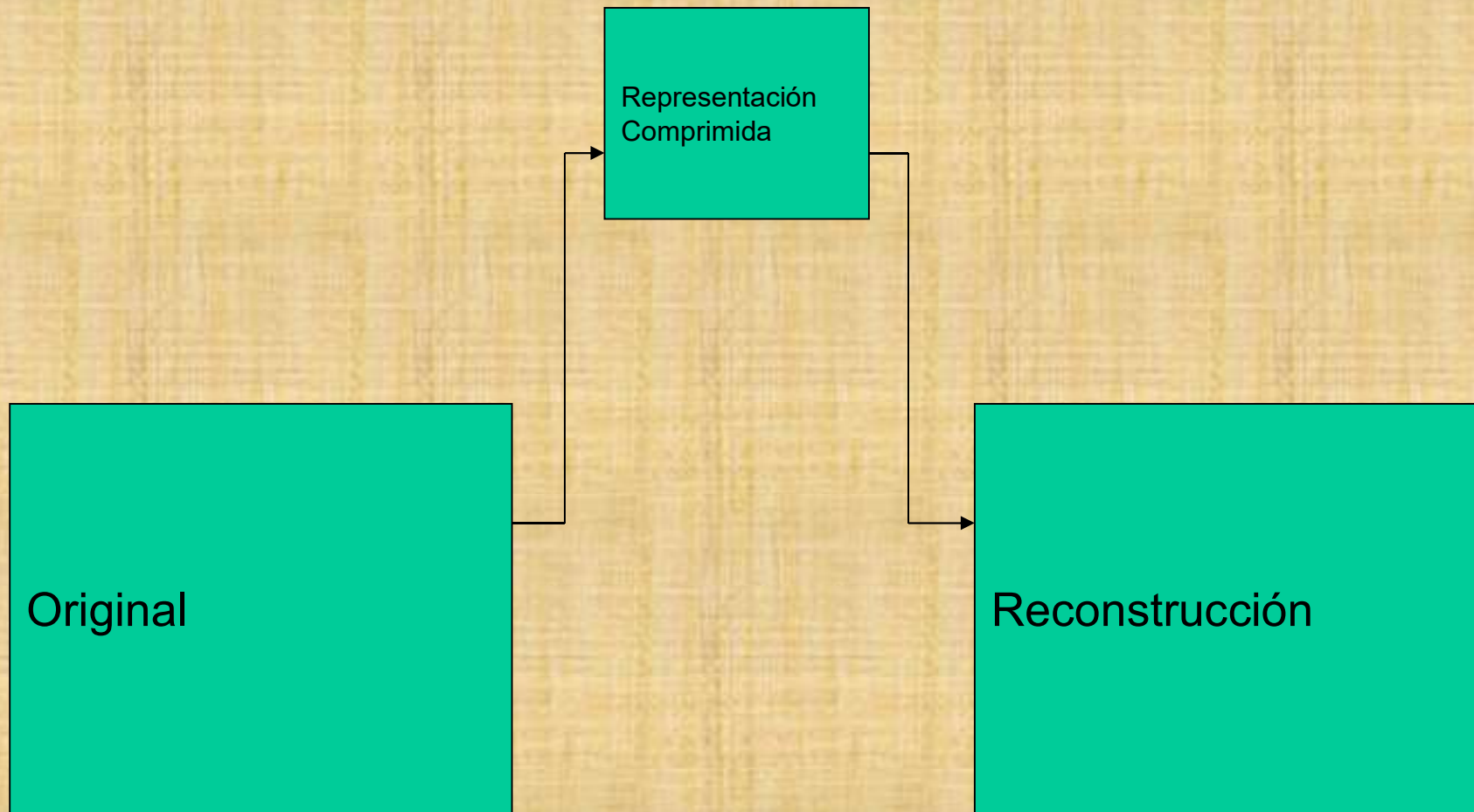
Distintas cantidades de datos pueden transmitir la misma información. (Ejemplo: charla entre personas).

Si se incluyen datos no esenciales hay

Redundancia de Datos

Estos datos se pueden eliminar.

Procesamiento de los Datos

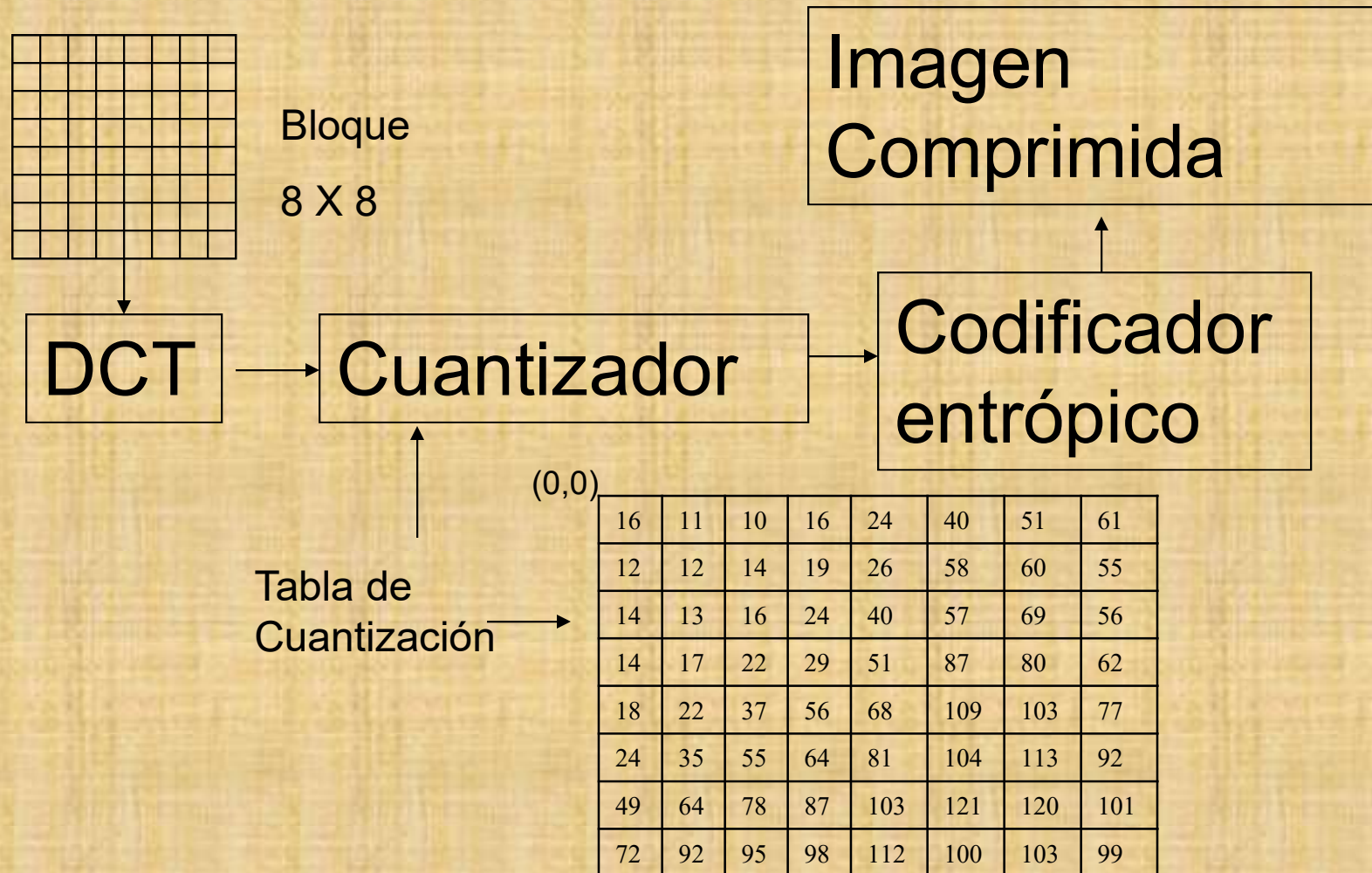


Uso de Transformadas

Transformada 2-DCT

JPEG

Compresión JPEG



Estándar de Compresión JPEG

JPEG= Joint Photograph Experts Group

- **La transformada discreta del coseno (DCT)** es la herramienta clave en el estándar de compresión JPEG.
- La **DCT** es una variación de la transformada discreta de Fourier donde la imagen se descompone en sumas de cosenos (y no de senos y cosenos como en la de Fourier).

La **DCT** unidimensional para $f(x)$ se define como:

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos \left(\frac{(2x+1)u\pi}{2N} \right)$$

La inversa se define como

$$f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos \left(\frac{(2x+1)u\pi}{2N} \right) ; \quad \alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, \dots, N-1 \end{cases}$$

Estándar de Compresión JPEG

DCT

Transformada discreta del coseno bidimensional

Para cada valor (u,v) con $u=0,1,2,\dots,N-1$ y $v=0,1,2,\dots,N-1$, se tiene que

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left(\frac{(2x+1)u\pi}{2N} \right) \cos \left(\frac{(2y+1)v\pi}{2N} \right)$$

La inversa de la transformada tiene por fórmula:

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) C(u, v) \cos \left(\frac{(2x+1)u\pi}{2N} \right) \cos \left(\frac{(2y+1)v\pi}{2N} \right)$$

para $x=0,1,2,\dots,N-1$ e $y=0,1,2,\dots,N-1$ y

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, \dots, N-1 \end{cases}$$

Estándar de Compresión JPEG

DCT

La DCT es de núcleo separable y simétrico,

$$C(u, v) = \alpha(u) \sum_{x=0}^{N-1} C(x, v) \cos \frac{(2x+1)u\pi}{2N}$$

donde

$$C(x, v) = \alpha(v) \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2y+1)v\pi}{2N}$$

Es decir, $C(u, v)$ puede expresarse

$$C(u, v) = \sum_{x=0}^{N-1} \left(\sum_{y=0}^{N-1} f(x, y) g(y, v) \right) g(x, u)$$

donde

$$g(a, b) = \alpha(b) \cos \frac{(2a+1)b\pi}{2N}$$

Por tanto, la DCT se puede obtener matricialmente

$$C = MFM^T$$

Estándar de Compresión JPEG

DCT

Usa la Transformada Discreta del Coseno

EJEMPLO: Consideremos la siguiente imagen **I** y su matriz asociada **F**



I

31	23	60	93	49	57	57	35
31	21	74	73	55	52	45	44
25	16	82	65	61	38	50	51
26	38	91	61	62	28	49	50
17	85	70	76	50	34	37	43
29	90	68	53	43	25	30	49
116	184	108	25	52	26	38	36
151	144	182	129	44	23	41	39

F

Estándar de Compresión JPEG

DCT

Ejemplo (continuación):

La transformada discreta del coseno de F, C, es una nueva matriz de las mismas dimensiones que la anterior:

$C = MFM^T =$

466.2500	100.9063	-40.3393	-91.1315	-25.2500	2.6791	-3.3152	16.6482
-88.4065	-146.0661	-67.5059	26.1243	33.4120	44.3008	24.6658	-0.4861
69.5604	73.4218	17.6909	8.1087	-1.2285	31.3229	12.3824	-7.9303
-44.0839	-28.5187	4.3866	18.4638	-11.9507	-32.1620	-31.1853	-32.1898
27.5000	12.2552	-25.4491	-19.8833	9.5000	35.2306	3.2352	-10.1438
-2.2299	6.3770	26.7637	18.3775	-13.4471	-20.9802	-10.1116	19.2261
-10.4122	-22.5187	-20.8676	-15.9436	17.2859	25.7420	9.5591	-4.7470
8.0193	11.5534	7.5823	9.4544	-9.8801	-12.0629	-13.2570	-7.4175

Los valores mayores se encuentran en la parte triangular superior-izquierda de la matriz.

Estándar de Compresión JPEG

DCT

Ejemplo (continuación):

Para almacenar la matriz C se realiza un proceso de normalización, es decir, se busca una función $Q(u,v)$ (*matriz de normalización*) tal que:

$$C^*(u,v) = \text{Redondeo}(C(u,v)/Q(u,v))$$

sea una matriz con "muchos" ceros. Esta nueva matriz, C^* es la que se almacena en el siguiente paso.

JPEG recomienda una matriz de normalización estandarizada para las imágenes con 256 niveles de **intensidad**, que es la indicada

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Estándar de Compresión JPEG

DCT

Ejemplo (continuación)

Aplicando esta normalización a nuestro ejemplo obtenemos la nueva matriz:

29	9	-4	-6	-1	0	0	0
-7	-12	-5	1	1	1	0	0
5	6	1	0	0	1	0	0
-3	-2	0	1	0	0	0	-1
2	1	-1	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

que tiene muchos ceros en la parte triangular inferior-derecha.

Estándar de Compresión JPEG

OBSERVACIONES:

- Es en el paso de normalización donde radica la pérdida de información. Dependiendo de cómo normalicemos C^* , conseguiremos comprimir más pero, a la vez, perder más información.
- Según el estándar JPEG
 - Se divide la imagen original en matrices cuadradas de 8 x 8 píxeles.
 - Se aplica DCT sobre cada uno de estos bloques.
 - Se obtiene una matriz C^* , mediante un proceso de normalización aplicado a cada bloque transformado.

Estándar de Compresión JPEG

OBSERVACIONES:

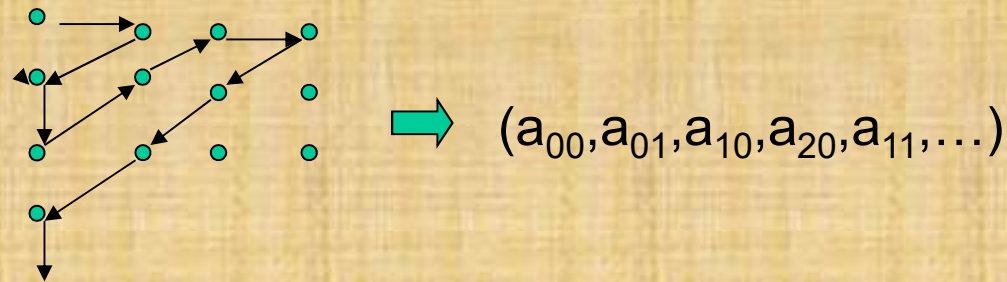
• En la mayoría de los casos, los programas que han implementado el método JPEG, antes de realizar la DCT, realizan una traslación en los colores de la imagen para pasar de $[0,255]$ a $[-128,127]$.

- Es decir, resta todas las intensidades de la imagen por 128.
- De esta forma se consigue que los valores de intensidad estén distribuidos alrededor del cero.
- El único cambio que se produce en realidad, es en el primer elemento de la matriz transformada.

Estándar de Compresión JPEG

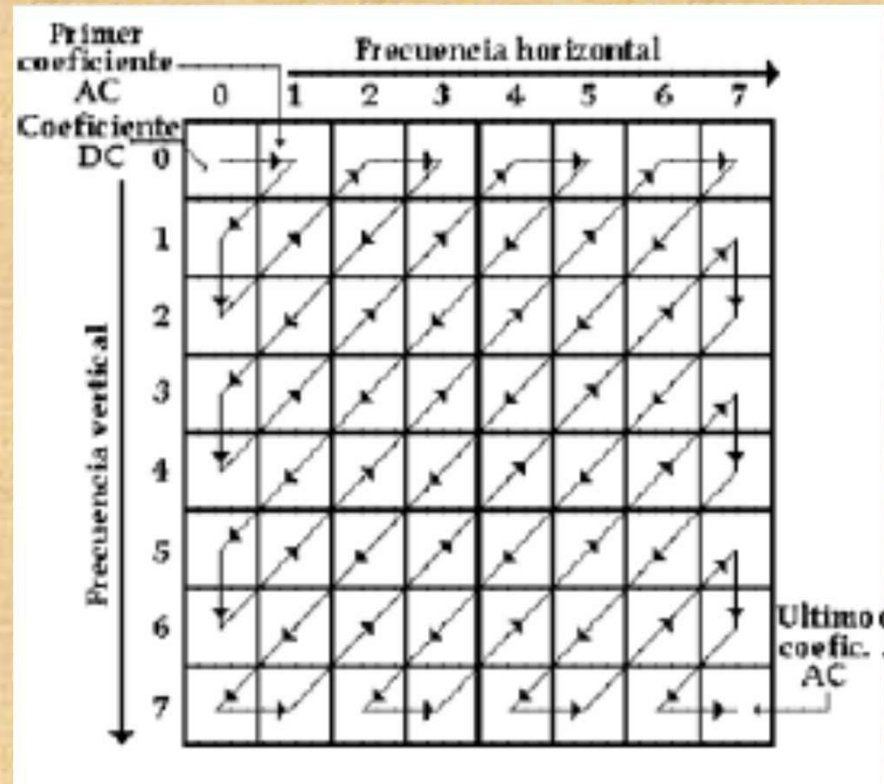
Codificación

Para almacenar la imagen normalizada, se sigue un recorrido en **zig-zag** en la matriz. De esta forma en la lista obtenida los ceros quedan al final.



- 1) Se usa **Run Length Encoding** o codificación secuencial para codificar la lista resultante, ya que suele tener bastantes secuencias de ceros.
- 2) Se usa luego el algoritmo de **Huffman** que se basa en utilizar el menor espacio posible en bits para aquellos caracteres más repetidos. Se puede usar una compresión de Huffman propia, o bien unas tablas estandarizadas que permiten obtener el código de Huffman para cualquier valor.

Estándar de Compresión JPEG



El primer coeficiente se conoce como DC (de continua) y los otros como AC.

Estándar de Compresión JPEG

Ejemplo (continuación)

,

29	9	-4	-6	-1	0	0	0
-7	-12	-5	1	1	1	0	0
5	6	1	0	0	1	0	0
-3	-2	0	1	0	0	0	-1
2	1	-1	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

la lista sería:

29, 9,-7,5,-12,-4,-6,-5,6,-3,2,-2,1,1,-1,0,1,0,0,1,0,0,0,
1,1,0,1,0,0,0,1,0,-1,F

Con la letra F (de fin) se indica que desde ese elemento hasta el final de la lista son todos ceros (hasta completar los 64 elementos de la lista).

El siguiente paso es almacenar la lista usando algún algoritmo que elimine la redundancia de código.

Estándar de Compresión JPEG

Descompresión: transformada inversa del coseno IDCT

1. Para descomprimir la imagen, hemos de decodificarla para obtener la matriz normalizada $C^*(u,v)$.
2. Deshacemos la normalización: $C'(u,v)=C^*(u,v)N(u,v)$, donde N es la matriz de normalización, y se aplica la transformada inversa de C' para obtener F' .

En el ejemplo, la matriz obtenida tras este proceso es:

34.2475	20.3801	61.1903	118.0446	41.3988	62.9105	62.2683	25.1212
26.8929	19.3063	67.7891	78.8857	42.4697	35.6132	63.3540	44.3871
17.6383	32.3336	83.8671	56.3275	60.1460	21.6380	55.1341	55.2890
11.3492	56.6311	87.7546	63.6566	75.0159	36.9026	36.3071	50.7470
16.5757	76.3151	65.5285	62.7385	54.8188	52.5775	20.5240	49.6389
47.8647	97.1627	59.2336	51.1976	23.0037	45.4744	19.2728	54.6494
104.6033	134.9000	112.7492	65.8845	30.0050	27.2591	28.9129	48.7557
152.9988	172.2212	182.4752	98.2682	62.6639	16.3330	37.6809	34.7164

Estándar de Compresión JPEG

Descompresión: transformada inversa del coseno IDCT

Redondeamos, para obtener una matriz de enteros, y la matriz descomprimida sería:

34	20	61	118	41	63	62	25
27	19	68	79	42	36	63	44
18	32	84	56	60	22	55	55
11	57	88	64	75	37	36	51
17	76	66	63	55	53	21	50
48	97	59	51	23	45	19	55
105	135	113	66	30	27	29	49
153	172	182	98	63	16	38	35

31	23	60	93	49	57	57
31	21	74	73	55	52	45
25	16	82	65	61	38	50
26	38	91	61	62	28	49
17	85	70	76	50	34	37
29	90	68	53	43	25	30
116	184	108	25	52	26	38
151	144	182	129	44	23	41

Matriz original

3	-3	1	25	-8	6	5	-10
-4	-2	-6	6	-13	-16	18	0
-7	16	2	-9	-1	-16	5	4
-15	19	-3	3	13	9	-13	1
0	-9	-4	-13	5	19	-16	7
19	7	-9	-2	-20	20	-11	6
-11	-49	5	41	-22	1	-9	13
2	28	0	-31	19	-7	-3	-4

Matriz diferencia

Estándar de Compresión JPEG

- El estándar recomienda:
 - Una matriz de cuantificación para luminancia
 - Las altas frecuencias tienen un paso mayor (menor exactitud) por la menor sensibilidad del ojo
 - Una matriz de cuantificación para cada color
 - La del color azul tiene un paso mayor (menor exactitud) por la menor sensibilidad del ojo.
- La matriz de cuantificación se puede escalar.
- Se obtienen así distintos grados de compresión con distintas calidades.
 - Observación: Incluso con calidad $Q=100$ hay compresión.
 - A mayor calidad menor compresión. Depende de la imagen.

Compresión JPEG – Q=5



Efecto de cuadriculado o Blocking

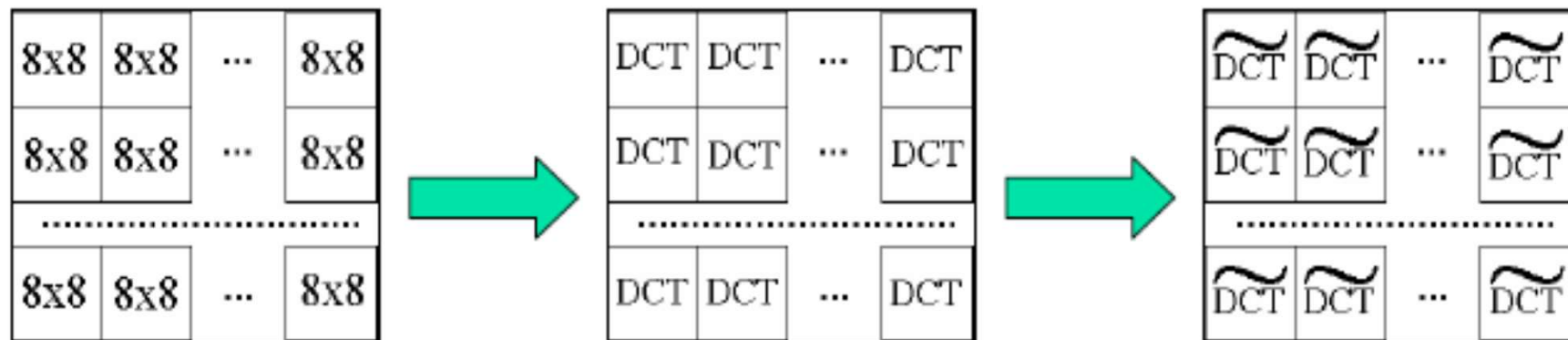
Compresión JPEG



Calidades Q 20, 15, 10 y 5 (Tasa compresión 5%,4%,3% y 2%)

Resumen Compresión JPEG

- La imagen se divide en bloques 8x8.
- Se hace la DCT de cada bloque (restando 128 previamente).
- Se cuantifica cada DCT (aquí están las pérdidas y la primera compresión).

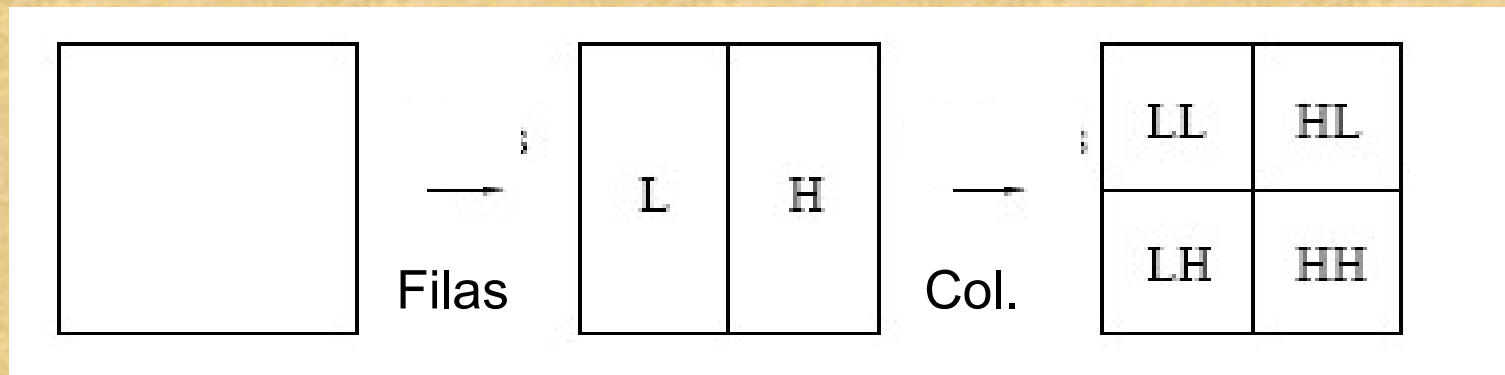


A cada bloque se le aplica compresión “sin pérdidas”:

RLE (Run Length Encoding) + VLC (Variable Length Code, tipo Huffman).

Transformada Wavelet JPEG 2000

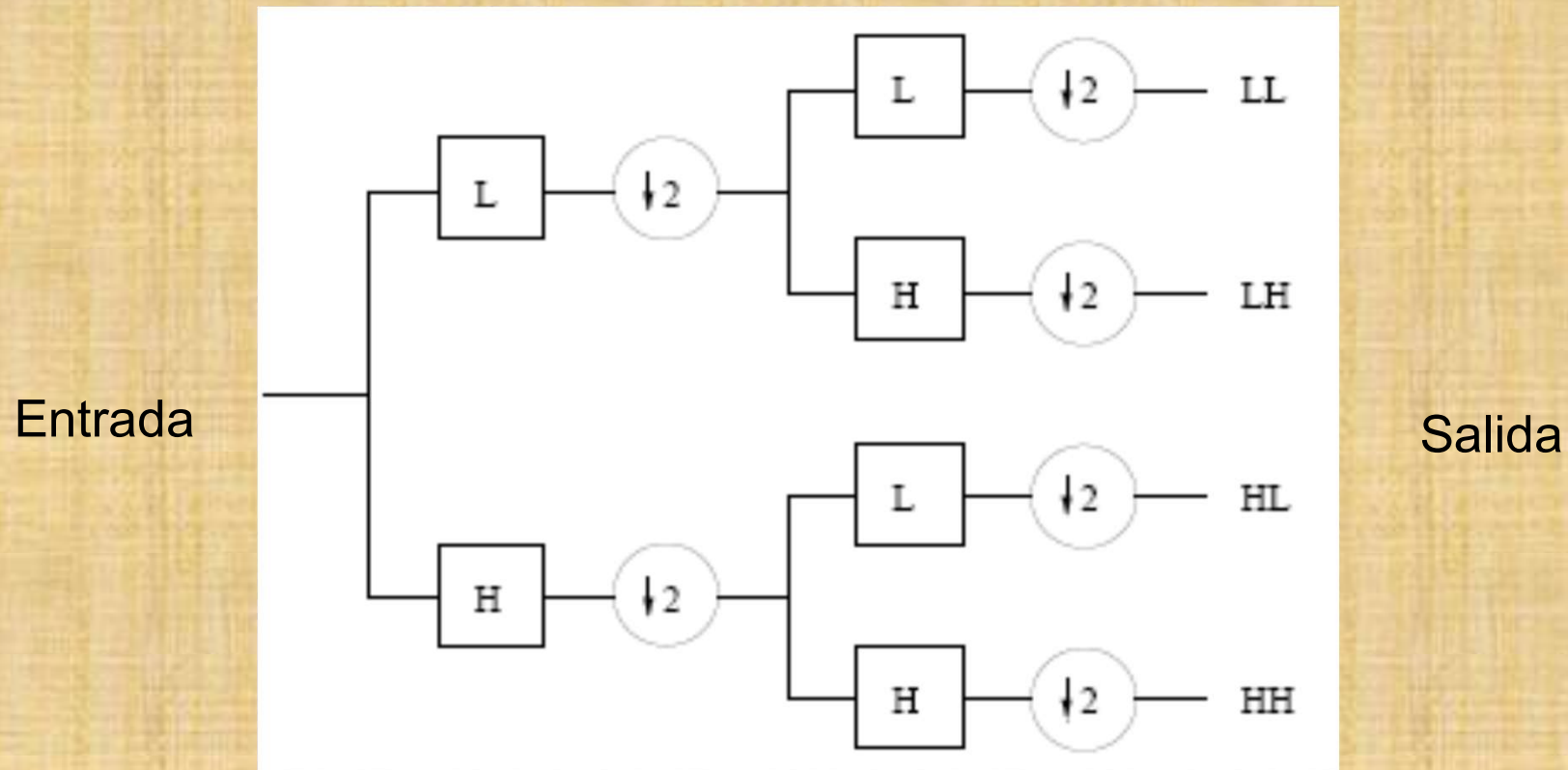
Transformada wavelet



Descomposición en una etapa

Transformada wavelet

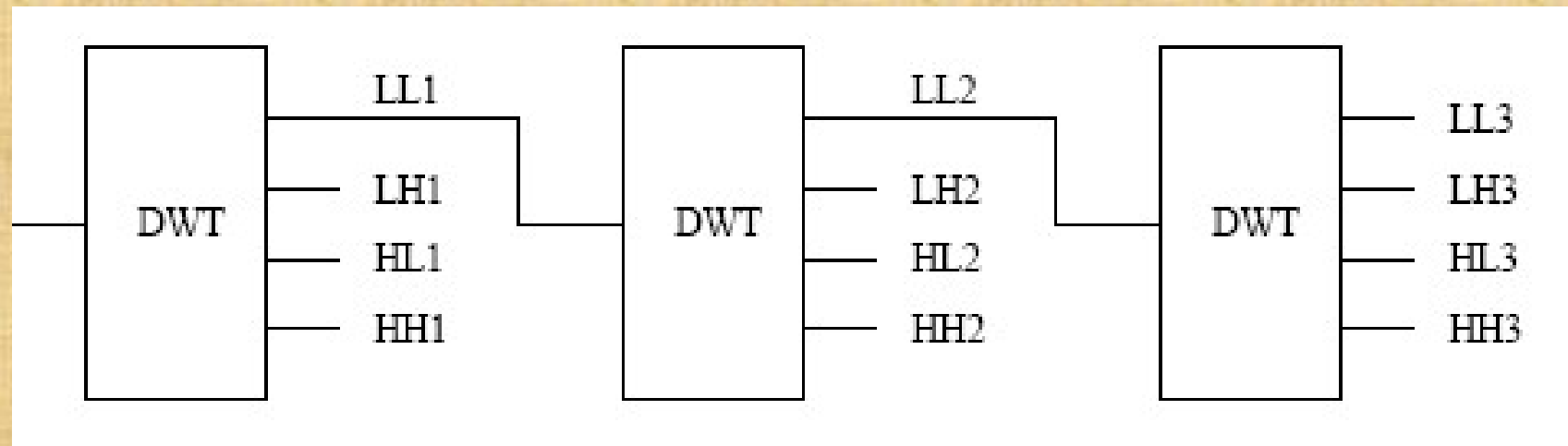
Análisis Multirresolución



Descomposición Wavelet de una Imagen en cuatro Sub-imágenes
Se aplican filtros PB y PA
Se hace submuestreo por 2 (se quitan elementos)

Transformada wavelet

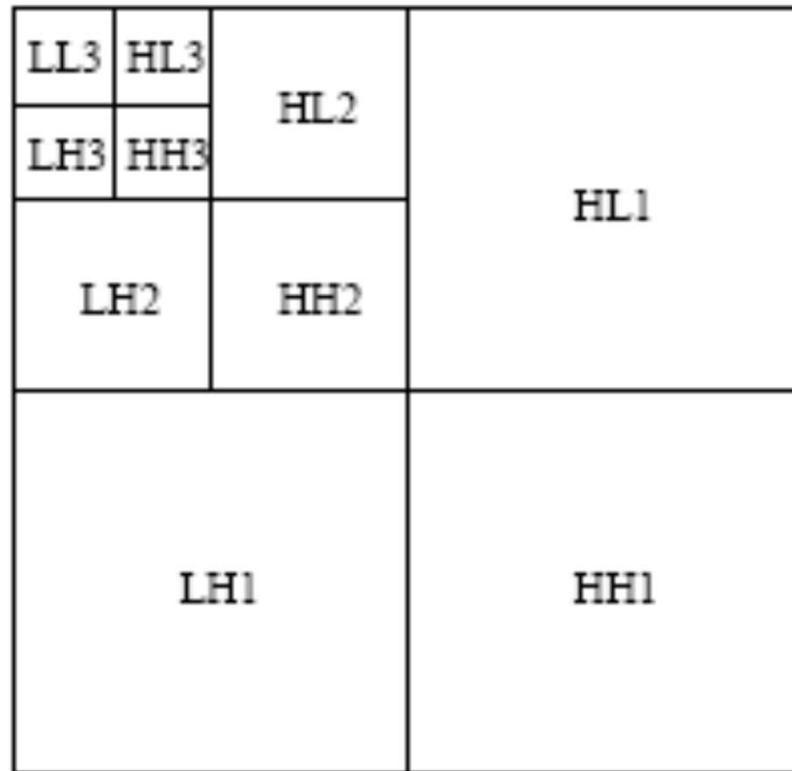
Descomposición Piramidal en tres Etapas



Descomposición de una Imagen en tres Etapas

Transformada wavelet

Descomposición Piramidal en tres Etapas



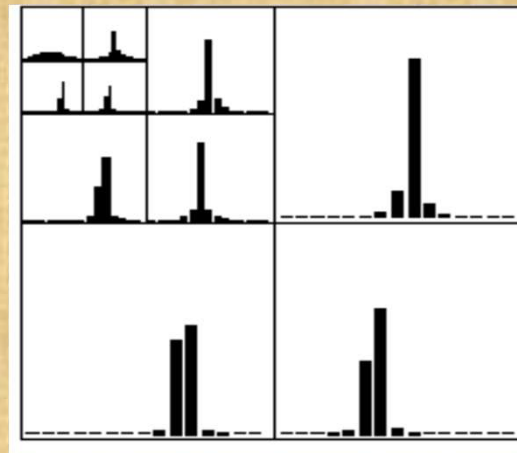
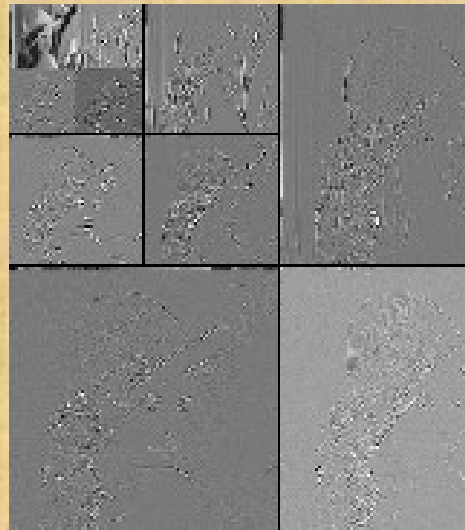
Transformada wavelet

Descomposición Piramidal en 3 etapas

Imagen Original



Imagen transformada



Transformada wavelet

Descomposición Piramidal en 3 etapas

Imagen Original

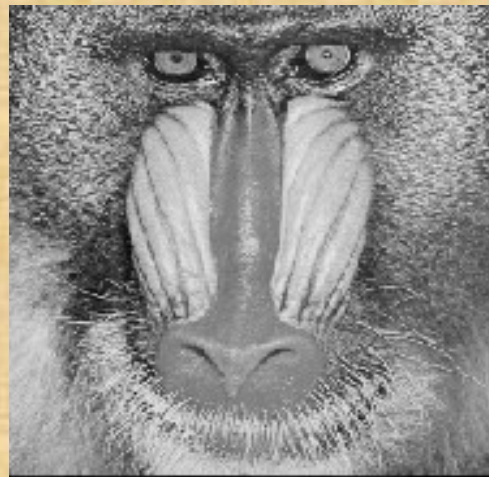
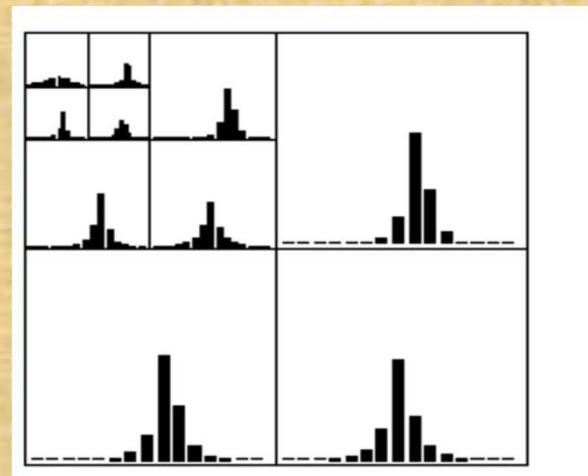
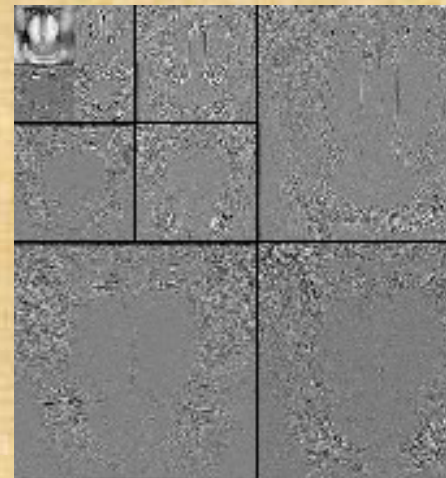


Imagen transformada



JPEG2000

Para solucionar las limitaciones de JPEG surge JPEG2000.

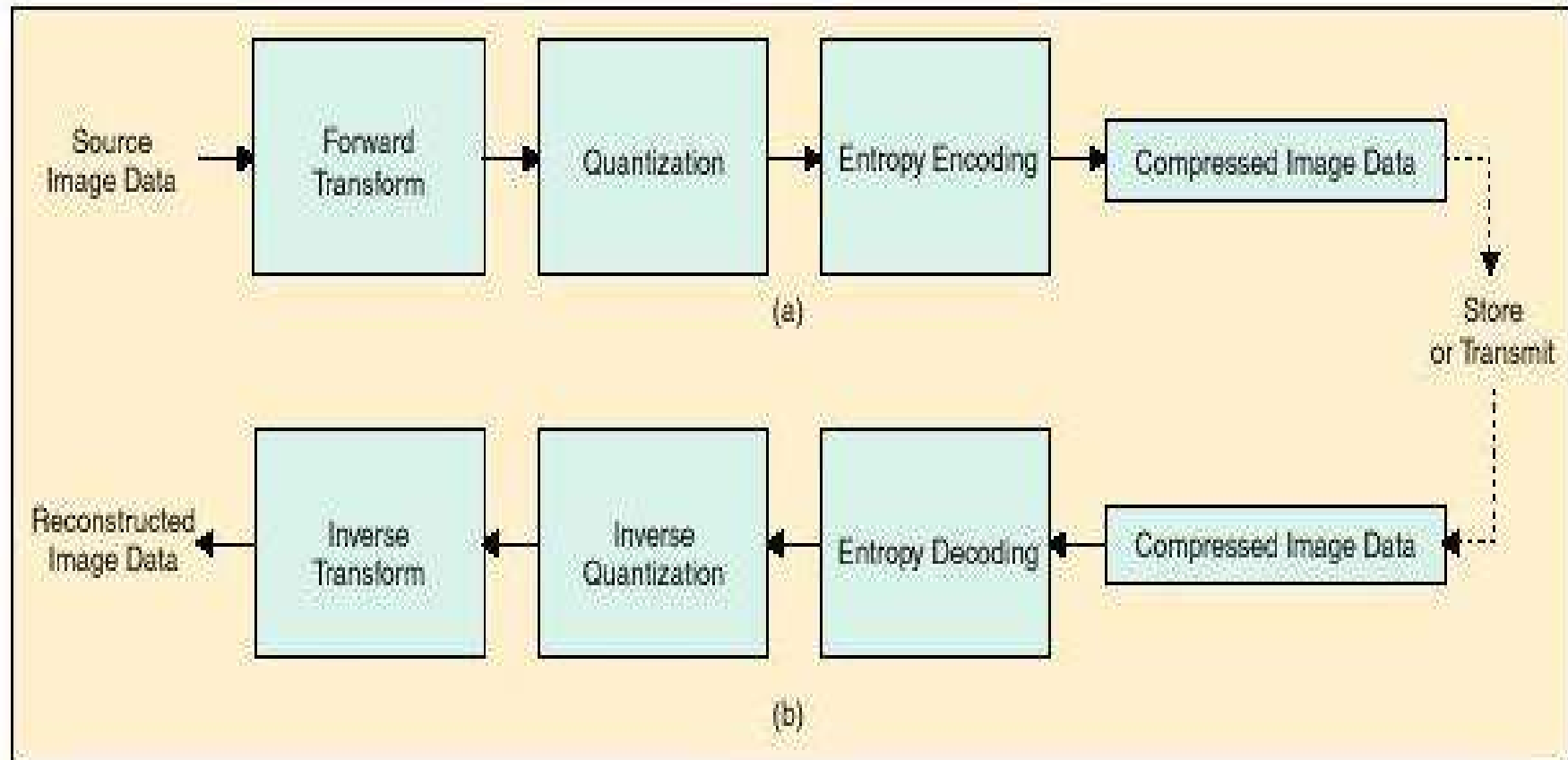
Se buscaba una compresión

- Que pudiera efectuar transmisión progresiva de imágenes
 - para imágenes satelitales
- Robusta ante errores
- Que permitiera elegir zonas de interés (ROI= Region of Interest)

El diagrama en bloques en grandes trazos no se diferencia.

Problema de JPEG 2000: Uso de patentes

JPEG2000- Diagrama Convencional



▲ 2. General block diagram of the JPEG 2000 (a) encoder and (b) decoder.

JPEG2000

La imagen se puede dividir en partes o baldosas (tiling).

Se aplica la transformada en las distintas partes con diferentes niveles de resolución.

Las subbandas de coeficientes obtenidas son cuantificadas y almacenadas en arrays rectangulares.

Los mapas de bits son codificados entrópicamente.

Se puede dar más calidad a las ROI que al fondo.

Se hacen agregados para resistencia a errores (robustez).

En la cabecera se coloca información sobre la imagen original y la codificación y decodificación seguidas.

Puede hacerse compresión con o sin pérdidas.

- Lossy (con pérdidas)

- Lossless (sin pérdidas)

JPEG2000



Comprimida JPEG 2000- Tasas 0,01- 0,03- 0,04 y 0,15

Ejemplo uso del Jasper

```
jasper -f lena.pgm -F lenaq5.jpg -O quality=5
```

Comprime según JPEG, con Q=5

```
jasper -f lena.pgm -F lena01.jp2 -O rate=.01
```

Comprime según JPEG2000, con tasa=.01

Para usarlo en MatLab:

```
!jasper -f lena.pgm -F lena.jp2 -T jp2 -O rate=0.01
```


Formatos de Imagen

Comparando tamaño de archivos:

Tipo de archivo	Tamaño-Ejemplo
TIFF con compresión LZW	901K
TIFF sin ningún tipo de compresión	928K
JPEG (para imágenes reales)	105K
PNG, compresión sin pérdida (para gráficos)	741K
GIF, compresión sin pérdida (256 colores)	131K

- Los métodos de compresión TIFF, PNG, GIF y JPEG son públicos y por tanto se pueden implementar en cualquier programa gráfico.
- GIF y JPEG son los más usados en las páginas web.
- JPEG consigue una mayor compresión en imágenes fotográficas que PNG.

Comandos MatLab

- Para comprimir hay que valerse de la función `imwrite`:

JPEG:

```
I=imread('lena.pgm');
```

```
imwrite(I,'lena.jpg','jpeg','quality',5);
```

JPEG2000

```
imwrite(I,'lena.jp2');
```

```
imwrite(I,'lena4.jp2','CompressionRatio',4)
```