

# Codificació

Alexandre Ribelles García

PID\_00198488



Los textos e imágenes publicados en esta obra están sujetos –excepto que se indique lo contrario– a una licencia de Reconocimiento-NoComercial-SinObraDerivada (BY-NC-ND) v.3.0 España de Creative Commons. Podéis copiarlos, distribuirlos y transmitirlos públicamente siempre que citéis el autor y la fuente (FUOC. Fundació para la Universitat Oberta de Catalunya), no hagáis de ellos un uso comercial y ni obra derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>

# Índice

<b>Introducción</b> .....	5
<b>Objetivos</b> .....	6
<b>1. Conceptos de codificación</b> .....	7
1.1. Codificación y compresión .....	9
1.2. Redundancia y entropía .....	11
1.3. Medida de la compresión de imagen .....	12
1.4. Medida objetiva y subjetiva de la calidad de imagen .....	13
1.5. Clasificación de las técnicas de compresión .....	14
1.5.1. Compresión sin pérdidas .....	14
1.5.2. Compresión con pérdidas .....	15
1.6. Tratamiento del color: representación y submuestreo .....	17
1.6.1. Representaciones del color: espacios RGB e YCbCr .....	18
1.6.2. Submuestreo de color .....	20
1.7. Ejercicios .....	22
<b>2. Codificación de imagen fija</b> .....	24
2.1. El formato JPEG .....	24
2.1.1. Modo secuencial .....	25
2.1.2. Otros modos de trabajo de JPEG .....	33
2.2. JPEG 2000 .....	35
2.3. JPEG XR .....	36
2.4. Tendencias futuras de la codificación de imagen .....	36
2.5. Ejercicios .....	37
<b>3. Codificación de vídeo y audio</b> .....	45
3.1. Modelos de codificación de audio y vídeo .....	45
3.2. Compresión de audio digital .....	47
3.3. Diferenciando contenedor y códec .....	49
<b>4. El estándar MPEG-2</b> .....	52
4.1. Antecedentes .....	53
4.2. Qué es MPEG-2 .....	54
4.3. Conceptos básicos: GOP Y MPEG .....	54
4.3.1. GOP .....	54
4.3.2. MPEG .....	57
4.3.3. Ejemplos de uso .....	59
4.4. Los flujos elementales de datos .....	60
4.4.1. Flujo elemental de vídeo .....	60
4.4.2. Flujo elemental de audio .....	62

4.4.3. Flujo elemental empaquetado .....	62
4.5. Almacenamiento y transmisión .....	63
4.5.1. Almacenamiento .....	63
4.5.2. Transmisión .....	63
4.6. Perfiles y niveles en MPEG-2 .....	64
4.6.1. Perfiles .....	64
4.6.2. Niveles .....	64
4.6.3. Combinaciones de perfiles y niveles .....	65
4.7. MPEG-2 en la industria .....	65
4.8. Medida de la calidad de imagen en MPEG-2 para DVB .....	66
<b>5. El estándar MPEG-4 y H.264/AVC.....</b>	<b>69</b>
5.1. El estándar y sus patentes .....	70
5.2. H.264 o MPEG-4 Part10 .....	70
5.3. Almacenamiento .....	71
5.4. Transmisión .....	71
5.5. Perfiles y niveles .....	72
5.5.1. Perfiles .....	72
5.5.2. Niveles .....	73
5.6. H.264 en la industria .....	73
5.7. Gestión de objetos multimedia con MPEG-4 .....	74

## Introducción

Se inicia en este módulo la visión en detalle de los estándares de codificación de imagen, sonido y vídeo. Para ello se discute su necesidad, se introducen los conceptos de redundancia y entropía, espacio y submuestreo de color, se describen los tipos de compresión y se brindan las herramientas empíricas y subjetivas para valorar la compresión.

El primer estándar de codificación que se analiza, JPEG para imagen, incorpora elementos que forman parte de la mayoría de sistemas de codificación, por lo que es descrito en detalle tanto desde el punto de vista práctico como teórico. Se comparan, además, sus modos de trabajo con y sin pérdidas, introduciendo finalmente las versiones más actuales del estándar: 2000 y XR.

A continuación, y aumentando en complejidad, se adentra en la codificación de audio y vídeo mediante el estándar MPEG-2, introduciendo los tipos y las estructuras de imagen, así como la descripción de su codificador (*coder*), decodificador (*decoder*) y los modos de trabajo por perfiles y niveles.

Por último, cierra el módulo el estándar H.264/AVC, denominado vulgarmente MPEG-4, comparándose con su antecesor y describiendo las novedades y aplicaciones de este estándar actual según sus perfiles y niveles.

## Objetivos

Los objetivos de este módulo son:

- 1.** Identificar los parámetros que caracterizan la imagen y el sonido.
- 2.** Diferenciar los conceptos de codificación y compresión.
- 3.** Identificar y diferenciar los estándares de codificación más habituales.
- 4.** Capacitaros para seleccionar el mejor formato de codificación en función de la calidad esperada, el tipo de material y el destino de la reproducción.

## 1. Conceptos de codificación

La digitalización de los sistemas de información iniciada en los años ochenta del siglo XX ha conllevado la generación constante, ininterrumpida y, sobre todo, creciente de datos que se almacenan en dispositivos y/o transmiten por sistemas de comunicaciones que no han evolucionado al mismo ritmo en cuanto a capacidad. La situación se acelera con el auge de los productos y servicios multimedia, verdaderos consumidores incansables de todo dispositivo de almacenamiento y culpables directos del “estrés” sobre la calidad de servicios de los proveedores de redes de datos. Nunca hay un disco suficientemente grande, nunca existe una red suficientemente rápida...

Dejando la exageración aparte, la **necesidad de comprimir la información** es lógica: las imágenes requieren considerablemente más espacio en bits que el texto. El audio y el video incluso demandan aún más en términos de almacenamiento y velocidad de transmisión. Unos pocos números ilustran este aspecto, introduciendo así algunos conceptos básicos que utilizaremos a lo largo de toda la asignatura:

a) Una página de este libro contiene unas 50 líneas de 80 caracteres; si cada carácter puede representarse, almacenarse y transmitirse digitalmente con 1 byte (8 bits), cada página ocupa 32.000 bits.

b) Un diseño de un edificio en vectores (es decir, creado con líneas, como los diseños de Autocad o una fuente escalable) contiene unos 500 trazos. Si cada trazo se representa simplemente por sus coordenadas de inicio y final (cuatro números en total) y cada número por 2 bytes, la imagen ocupa 32.000 bits.

c) La música en calidad CD supone 44.100 muestras de sonido cada segundo; si las muestras son de 16 bits y es estéreo, se generan 1.411.200 bits por segundo (casi 1,5 Mbits/segundo o lo que es lo mismo, 1,5 Mbps).

d) Una imagen fotográfica de 720 por 576 píxeles contiene 414.720 píxeles en total; si la imagen es en blanco y negro, cada píxel se representa por un byte, con lo que ocupa 3.317.760 bits (más de 3 Mbits).

e) Si la imagen anterior fuese en color, cada píxel se representaría por 3 bytes (uno para el rojo, otro para el verde y otro para el azul), ocupando el triple, 9.953.280 bits (casi 10 Mbits).

f) El sonido digital envolvente puede llegar a suponer casi 9 Mbps sin tratamiento alguno.

g) En el caso de vídeo, un clip de calidad estándar de 625 líneas a 25 cuadros/segundo necesitaría, sin comprimirse, unos 216 Mbps.

h) Si fuese HD estándar a 1.080 líneas (1.080i), aumentaría a 1,15 Gbps; o incluso el doble, 2,30 Gbps, si fuese HD puro (1.080p).

i) Actualmente, las soluciones propietarias de cine digital, Ultra HD y Hi-Vision están en torno a las 4.000 líneas de resolución, y no sería posible transmitir las sin compresión por debajo de los 10 Gbps.

j) Desde el año 2005, se han realizado emisiones experimentales en Super-Hi-Vision (8.000 líneas), por ejemplo, los Juegos Olímpicos del 2012. La señal pura de 24 Gbps sería imposible de transmitir actualmente incluso en fibra óptica si no se comprimiase previamente, y menos por Internet.



Los formatos de 4.000 líneas son denominados 4K, y los de 8.000 líneas 8K; actualmente son solo utilizados en posproducción de vídeo HD y cine.

Ilustración 1. TV LED Super Hi-Vision de 8.000 líneas presentado en el IBC 2012 en Ámsterdam



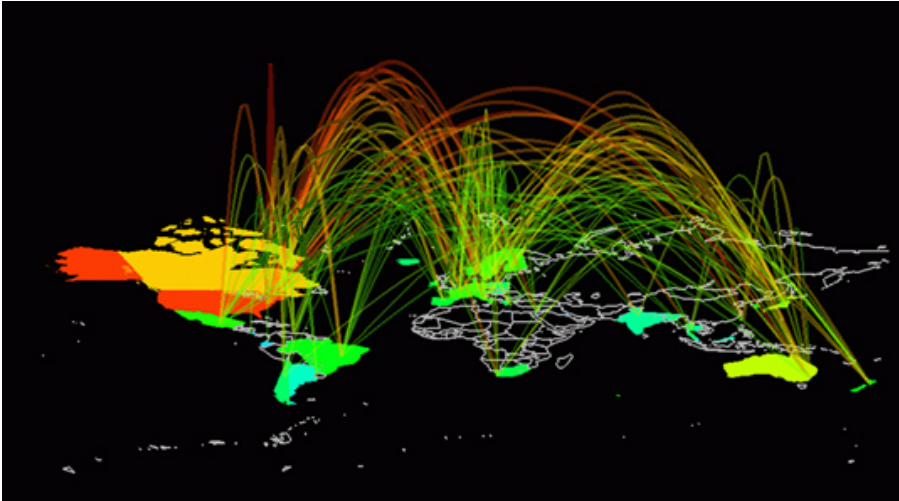
Foto del autor.

Los requerimientos de memoria y velocidad de transmisión para estas informaciones deben reducirse de manera notable para que puedan ser gestionados por los equipos actuales, con capacidades de almacenamiento del orden de Gbytes y memorias intermediarias<sup>1</sup> del orden de Mbytes, y de sistemas de transmisión en torno a algunos Mbps. Necesitamos comprimir, y en algunos casos de manera drástica, aunque suponga pérdida de parte de su contenido.

<sup>(1)</sup>En inglés, *buffers*.



Ilustración 2. Representación virtual de los grandes flujos de datos entre los principales países



### El impacto de la compresión

Más allá del gigabyte (Gbyte) y del terabyte (Tbyte), existen unidades cuya familiaridad va en aumento año tras año: el petabyte (1.024 Gbytes), el exabyte (1.024 Pbytes), el zettabyte (1.024 Ebytes) y el yottabyte (1.024 Zbytes). Por ejemplo, el tráfico global de Internet cada segundo es aproximadamente 20 Tbytes, de los cuales casi el 70% es transmisión en tiempo real (*streaming*) de datos de vídeo, el 10% es transferencia de ficheros, el 15% son correos electrónicos y navegación por webs, etc.

Así pues, la distribución de contenidos (ya sean multimedia o de otro tipo) puede suponer cerca del 80% del tráfico de la Red en cualquier momento, y sin duda comprimidos. Por ello, toda técnica que pueda suponer una mejora de la compresión tendrá un impacto económico de grandes magnitudes.

### 1.1. Codificación y compresión

Los términos *codificación* y *compresión* se utilizan habitualmente de manera equivalente, pero es incorrecto: *codificación* es sencillamente expresar una información de una manera diferente utilizando un nuevo código, mientras que *compresión* es reducir su tamaño total en bits. La compresión es un caso particular de codificación en el que el nuevo código utiliza menos elementos para representar la misma información.

Por ejemplo, la tabla ASCII codifica los caracteres expresándolos en formato binario, no existiendo ninguna compresión en el proceso. Por el contrario, una codificación que use una herramienta informática y represente un fichero de datos con otro de menor tamaño en bits (como .zip o .rar en Windows, .sit en Mac OS X o .tgen Linux) es un caso de compresión.

La compresión debe encontrarse completamente especificada tanto en el codificador en origen como en el decodificador del destinatario; así se asegura que la recepción es correcta. En el auge de Internet a finales de los años noventa del siglo pasado aparecieron innumerables formatos “propietarios” (es decir, exclusivos de una empresa) que obligaban al uso de programas propios para su visualización, incompatibles con los de la competencia: para RealVideo y RealAudio de Real Networks debía utilizarse la aplicación RealPlayer, o el Reproductor de Windows si el fichero era formato Windows Media Video

#### La tabla ASCII

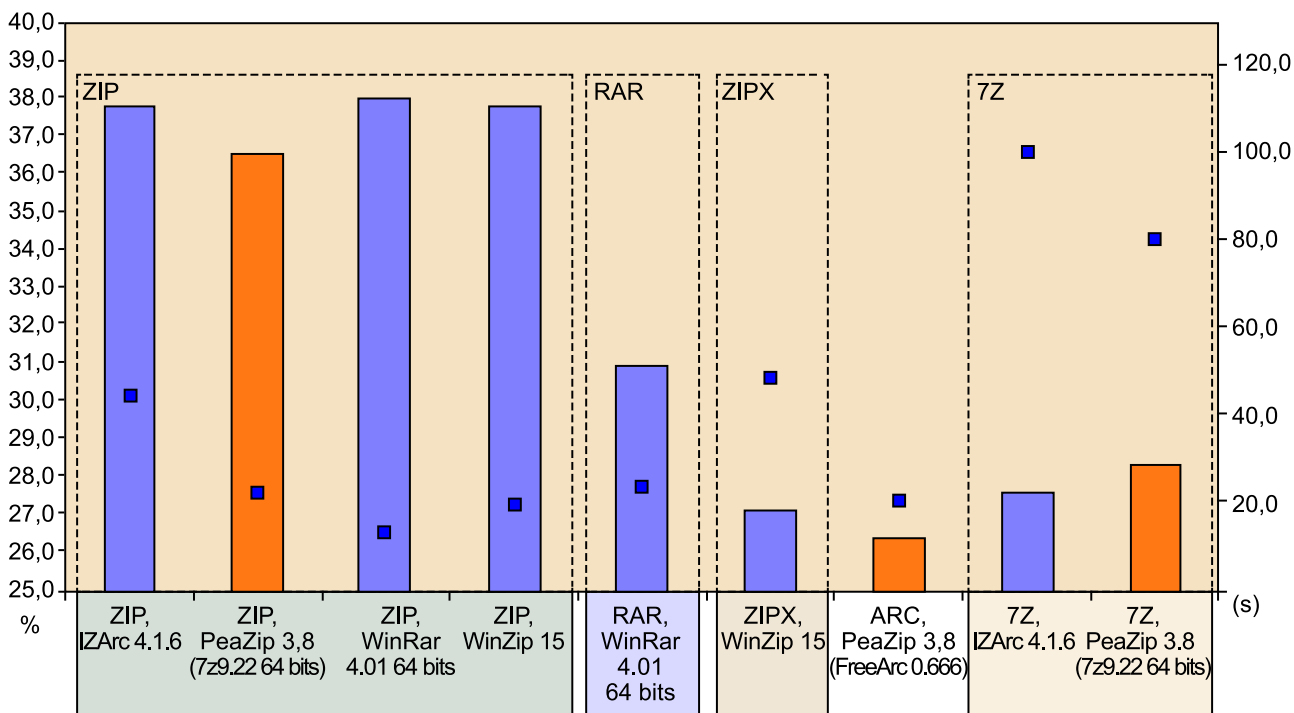
Los ordenadores y todo sistema digital en general solo interpretan números. La tabla ASCII es una convención que asigna un número a cada carácter para así poderlo representar en sistemas digitales.

de Microsoft. Hasta bien entrado el siglo XXI no aparecieron aplicaciones capaces de soportar diferentes formatos, lo que minimizó el caos y facilitó su universalización.

En la especificación de una **compresión**, o **códec**, se representa el video original por una codificación, es decir, por una gran ristra de valores fácilmente manejables. Idealmente, esta codificación debe ser eficiente (utilizar la menor cantidad de información posible, es decir, bits) y eficaz (debe representar de la forma más fiel posible el vídeo original).

Lógicamente, estos dos objetivos se encuentran en conflicto y es necesario un compromiso entre ambos. Habitualmente, un estándar de códec presenta varios **perfiles** o configuraciones con distintos valores de compromiso entre eficacia y eficiencia con el fin de adaptarse a una mayor cantidad de sistemas de vídeo digital.

Ilustración 3. Razón de compresión (barras) y tiempo de cálculo (puntos)



En ZIP, la eficiencia de la compresión varía según la aplicación utilizada, así como el tiempo que se requiere para realizarla. Sin embargo, el resultado siempre es un fichero del que pueden recuperarse todos los datos sin pérdida alguna, es decir, es eficaz.

Aún podemos perfilar más la definición de compresión: cuando indicamos que ha de representar la misma información, no significa que en todos los casos información final tenga que ser exactamente la misma que la original; es decir, la compresión puede ser sin pérdidas<sup>2</sup> o con pérdidas<sup>3</sup>. En el primer caso, podemos realizar el proceso inverso para volver a tener la misma información original exacta sin perder un ápice: por ejemplo, de todo fichero .zip o .rar puede recuperarse el fichero original. En el segundo caso, aceptamos una degradación del mensaje en aras de conseguir una compresión efectiva, siempre y cuando no haga inservible la información que lleva. Por ejemplo, al copiar

<sup>(2)</sup>En inglés, *lossless*.

<sup>(3)</sup>En inglés, *lossy*.

un fichero de audio desde el CD original hasta iTunes se codifica automáticamente en un formato de audio con pérdidas (AAC, MP3, etc.) mucho más reducido en tamaño pero de experiencia sonora muy parecida a la original, aunque no exacta.

## 1.2. Redundancia y entropía

¿A qué nos referimos con información inservible? En muchos casos, la información que se transmite tiene **datos redundantes**, información predecible o repetitiva que puede comprimirse de forma intensa, como por ejemplo, varios espacios en blanco consecutivos en un texto, o varios píxeles consecutivos de un mismo color en un fax. Su reconstrucción al descomprimir es exacta.

En la **compresión sin pérdidas** se elimina la redundancia de la información, sin tocar un ápice la información fundamental.

La entropía es una medida de la cantidad de información promedio que poseen los símbolos que se utilizan para representar esa información. Si podemos representar una imagen con un conjunto de símbolos pequeño, cada uno de ellos con gran cantidad de información, la compresión será muy alta sin tener ninguna pérdida. Así, una información poco probable posee más información que otra más probable, por lo que puede asignarse más bits a la poco probable y menos a la más probable. Algunas técnicas de codificación que parten de este principio entrópico son la base de la compresión sin pérdidas de ficheros de datos y también en alguna de las etapas de los procesos de compresión con pérdidas.

Actualmente, las compresiones sin pérdidas pueden llegar a relaciones de compresión 2:1, lo cual es un éxito, pero la mayoría de las veces no es suficiente, por lo que la compresión con pérdidas es la técnica que impera en el mundo de la imagen y el sonido, en donde se necesitan tasas de reducción mucho más altas a costa de una pérdida controlada de fidelidad.

La base de la **compresión con pérdidas** es la eliminación de los datos irrelevantes en función de las limitaciones del destinatario, como por ejemplo, las pequeñas variaciones de color entre píxeles cercanos (ya que el ojo humano tiene poca resolución para el color) o los sonidos de frecuencias superiores a los 16.000 Hz (pues pocos seres humanos adultos son capaces de escucharlos). En la **compresión con pérdidas**, la información irrelevante se suprime completamente, lo cual modifica la información de manera permanente, aunque tolerable para el destinatario al que vaya dirigida.

### Compresión sin pérdidas

La compresión sin pérdidas es la ideal para ficheros de datos (zip, rar, sit, tg, etc.), algunos formatos de imagen como GIF, PIC, un modo de trabajo específico del formato de compresión de imagen JPEG 2000, etc.

### La entropía en la teoría de la información

En el ámbito de la teoría de la información, la entropía es un concepto algo diferente a la entropía en termodinámica o mecánica.

## Compresión con pérdidas

Ejemplos de formatos estándar de compresión con pérdidas son los siguientes:

- Formatos de compresión de imagen como JPEG, EZW, SPIHT y casi todos los modos de trabajo de JPEG 2000.
- Formatos de compresión de audio como AAC o MP3.
- Formatos de compresión de vídeo como la familia MPEG (1, 2, 4).

Evidentemente, el éxito de cualquier compresión dependerá de cuanta información redundante o irrelevante haya en el mensaje, por lo que es habitual que el rendimiento de la compresión varíe casi constantemente. La **entropía** es una medida de la información esencial de un mensaje (la diferencia entre la cantidad total de datos que contiene el mensaje y su redundancia). A mayor entropía, más datos no pueden eliminarse, por lo que la codificación con pérdidas rendirá poco. A menor entropía, la codificación con pérdidas será muy efectiva.

### 1.3. Medida de la compresión de imagen

Para medir la compresión conseguida en una imagen y así poder comparar diferentes sistemas de compresión, como haremos en la asignatura, utilizamos el **factor de compresión**, un valor numérico que aumenta con la compresión conseguida.

Dada una imagen original y su equivalente comprimido, el **factor de compresión** es:

$$\text{Factor de compresión} = \frac{\text{Peso de la imagen "original"}}{\text{Peso de la imagen "comprimida"}}$$

recordando indicar ambos pesos siempre en la misma unidad (bits, bytes, Kbytes, etc.).

Otra manera de representar la compresión es indicando el número de bits que necesita cada píxel de la imagen, el **bit por píxel (bpp)**:

$$\text{bpp} = \frac{\text{Peso de la imagen (expresado en bits)}}{\text{Número total de píxeles de la imagen}}$$

De manera sencilla, podemos deducir que el bpp de una imagen original y el bpp de su imagen comprimida se relacionan con el factor de compresión entre ellas mediante la expresión siguiente:

$$\text{Factor de compresión} = \frac{\text{bpp de la imagen "original"}}{\text{bpp de la imagen "comprimida"}}$$

## Ejercicio

Si una imagen es de  $1.600 \times 1.200$  píxeles con 24 bits de color/píxel y se comprime en un formato tal que el fichero resultante es de 230 Kbytes, ¿cuál es el bpp de la imagen original?; ¿y el de la comprimida?; ¿cuál será el factor de compresión conseguido? Atención con las unidades, no mezcléis bits con bytes.

### 1.4. Medida objetiva y subjetiva de la calidad de imagen

Actualmente, se siguen investigando técnicas cuantitativas que puedan asignar un valor numérico a la calidad visual de cualquier imagen, proporcionando una idea de lo degradada que ha resultado tras aplicarle cierta compresión con pérdidas y ayudándonos así a elegir el estándar de compresión más adecuado.

En todo caso, presentamos una medida, el error cuadrático medio, una manera de calcular las diferencias que hay entre una imagen y su comprimida. Básicamente, una aproximación a diferenciar en cuánto son diferentes, algo que el ojo humano hace de manera natural en instantes pero que, matemáticamente, es muy difícil de concretar, pues no se conocen con detalle los mecanismos del cerebro humano que posibilitan tal hazaña. La idea del error cuadrático medio consiste en realizar una versión monocroma de la imagen (los colores se desprecian) y restar píxel a píxel una de otra, sumando finalmente las diferencias al cuadrado (así todas son positivas y no es posible que algunas se cancelen con otras). El resultado es un valor que cuanto más se acerca a cero, más parecidas son las imágenes.

Si la imagen original  $O$  es de tamaño  $N \times M$  píxeles, y la imagen  $C$  es la misma pero comprimida, el **error cuadrático medio**  $\epsilon^2$  resulta:

$$\epsilon^2 = \frac{1}{N \cdot M} \sum_{i=1}^N \sum_{j=1}^M (o[i, j] - c[i, j])^2$$

en donde  $o[i, j]$  es el valor del píxel de la fila  $i$  columna  $j$  de la imagen  $O$  monocroma, y  $c[i, j]$  la de la imagen  $C$  monocroma.

Tras ver el formato JPEG, veremos una manera mucho más visual y entretenida de detectar los errores mediante la aplicación Photoshop, que es capaz de calcular la imagen diferencia, es decir, una resta visual píxel a píxel entre, por ejemplo, una imagen sin comprimir y su equivalente comprimido, que pone en evidencia la información que se pierde en la compresión.

Otra medida derivada de esta es la **relación señal a ruido de pico (PSNR<sup>4</sup>)**:

<sup>(4)</sup>PSNR es la sigla de la expresión inglesa *peak signal-to-noise ratio*.

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX^2}{\epsilon^2} \right)$$

donde MAX es el valor máximo que puede tomar un píxel, usualmente 255. El resultado es en decibelios (dB), y los valores habituales suelen situarse entre 35 y 50 dB. Cuanto mayor es el valor, más parecida es la imagen comprimida a la original.

Sin embargo, la relación señal a ruido de pico es una medida no del todo fiable, ya que los errores visuales son percibidos por el ojo humano en mayor medida en función del entorno en que aparecen, cosa que no se valora con estas herramientas, más globales.

La segunda familia de mediciones se basa en encuestas que recogen la impresión subjetiva de los usuarios ante ambas imágenes, y valoran su comparativa mediante unos pesos, por ejemplo:

Ranking	Descripción de la comparativa
9-10	Muy buena
7-8	Buena
4-6	Aceptable
2-3	Mala
0-1	Muy mala

## 1.5. Clasificación de las técnicas de compresión

La principal clasificación de las técnicas de compresión divide los códecs en dos grandes familias, tal como se ha comentado brevemente al principio del apartado: sin pérdidas o con pérdidas.

### 1.5.1. Compresión sin pérdidas

Las **compresiones sin pérdidas** (*lossless*) proporcionan la máxima calidad visual, y se utilizan en entornos de documentación o posproducción de imagen y vídeo en los que la imagen debe mantenerse a la máxima calidad de origen.

Los códecs Matrox, Aja o Blackmagic se popularizaron en este entorno para vídeo de definición estándar, y con la llegada de la alta definición continúan utilizándose en su versión HD. El formato DPX de Kodak, basado en su clásico sistema de digitalización Cineon (1993), permite trabajar en HD con vídeo en puro RGB y muestreo 4:4:4 a 10 bits/muestra. Las mejoras del formato de fichero de imagen OpenEXR de Industrial Light & Magic (2007), como un formato abierto sin licencia con una variada colección de algoritmos de compresión sin pérdida capaces de comprimir hasta en ratios 2:1, han modificado el panorama de la posproducción.



Sin embargo, para reducir las necesidades de espacio y de ancho de banda de trabajo en HD (entre 1 y 1,5 Gbps), se han comenzado a aplicar discretamente códecs de compresión con pérdidas, reduciendo así los requerimientos de almacenamiento en *arrays* de discos.



En resumen, las técnicas sin pérdida quedan acotadas a entornos específicos y dedicados, teniendo siempre tasas de compresión muy bajas (por no decir casi inexistentes si se aplican sobre material de vídeo visualmente rico y muy cambiante).

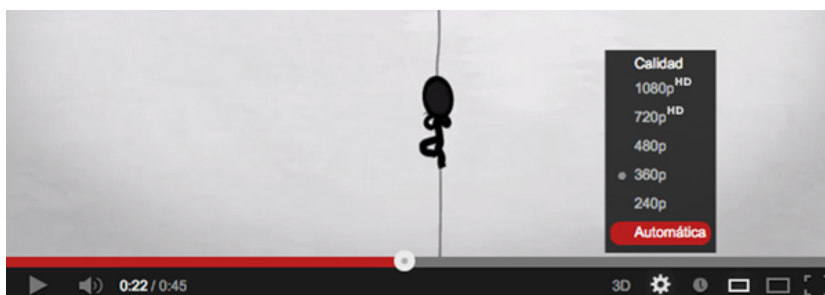
### 1.5.2. Compresión con pérdidas

Comprimir con pérdidas (*lossy*) no es renunciar a una calidad final de trabajo, sino que se define la calidad adecuada para el tipo de destinatario al que va dirigido.

#### Calidad según el destinatario

Un clip informativo colgado en YouTube no necesita ser de máxima calidad si su objetivo es informar, y un cortometraje puede necesitar ser HD para presentar la riqueza cromática de sus escenarios a través de la misma plataforma. En ambos casos, contamos en el servidor de YouTube con el fichero de vídeo en cierta calidad inicial, y YouTube lo reduce visualmente según la definición requerida por los posibles destinatarios, siempre en sentido descendente desde la calidad original a inferiores:

Ilustración 4. Opciones de visualización de un vídeo 1.080p en YouTube



En el caso de vídeo en tiempo real (*live streaming*), la compresión con pérdidas es la apuesta necesaria e imprescindible para facilitar la llegada de la imagen y el sonido con la cadencia necesaria para ser entendible. No hay tiempo material para hacer una codificación óptima como se haría en un vídeo almacenado: “obligaciones del directo”. Y aun así no

asegura el éxito en este objetivo al haber otros elementos que pueden impedirlo, como veremos más adelante.

Ilustración 5. Canal de noticias 324

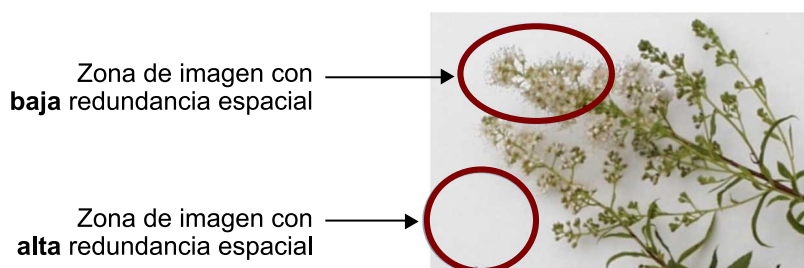


Muchos canales de TV emiten en tiempo real por Internet.

Las técnicas más comunes de compresión se basan, por una parte, en eliminar la redundancia espacial y temporal presentes en el vídeo “crudo”, características habituales del vídeo; y, por otra, en aprovechar las limitaciones de los sentidos humanos:

**a) Redundancia espacial.** Se basa en la suposición de que los píxeles de una zona de una imagen son muy parecidos a los de las zonas que lo rodean, salvo donde haya un perfil (la transición de un objeto a otro representado en la imagen). El estándar JPEG de imagen se basa en estas similitudes para comprimir la imagen fija, almacenando solo las leves diferencias entre grupos de píxeles y sus vecinos. Veremos que el estándar MPEG también lo aplicará, en este caso a vídeo, sobre algunos de sus cuadros de imagen.

Ilustración 6. Una imagen con zonas muy diferenciadas de detalle



En otras palabras, las zonas homogéneas de la imagen son candidatas a una buena compresión espacial con poco esfuerzo. Sin embargo, el detalle, es decir, la riqueza de la imagen, es la que se pone en juego. El **compromiso entre compresión y calidad de imagen** diferenciará un buen planteamiento de compresión de otros menos acertados.



b) **Redundancia temporal.** Se basa en la suposición de que hay pocas diferencias entre un fotograma y sus inmediatamente anteriores o posteriores.

Ilustración 7



Un *pan* o 'barrido suave' de la cámara de izquierda a derecha sobre un escaparate presenta muy pocas diferencias entre dos fotogramas consecutivos del vídeo generado. Sin necesidad de calcular la imagen diferencia, se puede asegurar que hay una alta redundancia temporal.

c) **Redundancia psicovisual.** El ojo humano tiene una respuesta limitada a los detalles espaciales finos (poca sensibilidad a la alta frecuencia espacial), y es menos sensible al detalle (léase "a los errores") cuando este se sitúa cerca de los bordes de objetos o en un cambio de plano. En consecuencia, estas limitaciones pueden camuflar efectos en la imagen decodificada cuando se define el algoritmo de compresión con pérdidas.

En los estándares de compresión de imagen fija, audio y vídeo que se presenta en los siguientes módulos, se verá el uso por separado o combinado de estas tres técnicas.

### 1.6. Tratamiento del color: representación y submuestreo

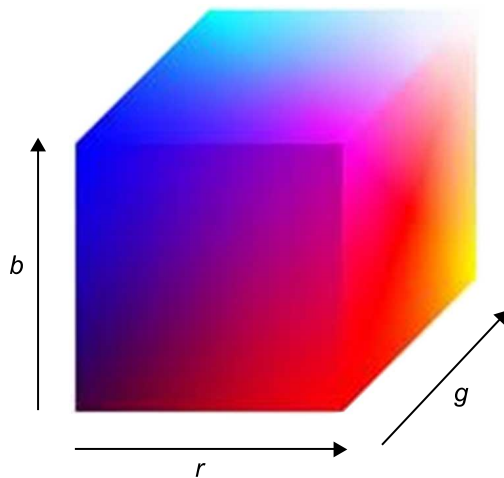
Ya que toda imagen y vídeo se genera actualmente en color, este tiene un tratamiento específico muy habitual en todos los estándares de compresión que se presentarán a continuación. En primer lugar, el color puede representarse de diferentes maneras o espacios, unos más adecuados que otros a la hora de codificar. Y en segundo lugar, es habitual reducir el número de muestras de color (submuestrear) en todos ellos.

### 1.6.1. Representaciones del color: espacios RGB e YCbCr

Todo píxel de una imagen está localizado en ella mediante sus dos coordenadas  $X$  e  $Y$ , pero además posee tres valores enteros que representan su color en un espacio de colores. Los espacios de color más conocidos son el RGB, YCbCr (para vídeo digital) e YUV (para vídeo analógico).

RGB se basa en los colores básicos rojo ( $R$ ), verde ( $G$ ) y azul ( $B$ ). Es un espacio de color intuitivo en el que la mayoría de colores puede representarse como una combinación de estos tres colores.

Ilustración 8. Colores RGB



Dado que la gama es infinita en cualquiera de ellos, usualmente se reduce a 256 valores enteros por color, del 0 al 255. Así,  $R = 0$  es el rojo negro y  $R = 255$  el rojo más vívido y puro. Hay que especificar, no obstante, que esta codificación es usada por pura conveniencia.

Los otros dos espacios de color en los que se puede definir también cualquier color con tres valores son equivalentes a RGB, solo que plantean cada color como un combinado de blanco y negro y dos componentes de color.

Así, en YCbCr el valor  $Y$  es el valor de luminancia (brillo),  $Cb$  el valor de crominancia azul y  $Cr$  el de crominancia rojo. El valor de crominancia verde puede deducirse de los tres anteriores. El espacio YUV también sigue la misma filosofía.

#### Crominancia azul

Se denominan crominancias azul y roja pero no son los componentes de color rojo y azul, ni siquiera corresponden a colores, son solo valores numéricos que incluso pueden ser negativos. Estas señales también se denominan diferencias de color.

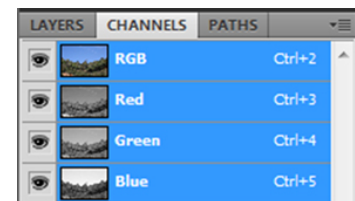
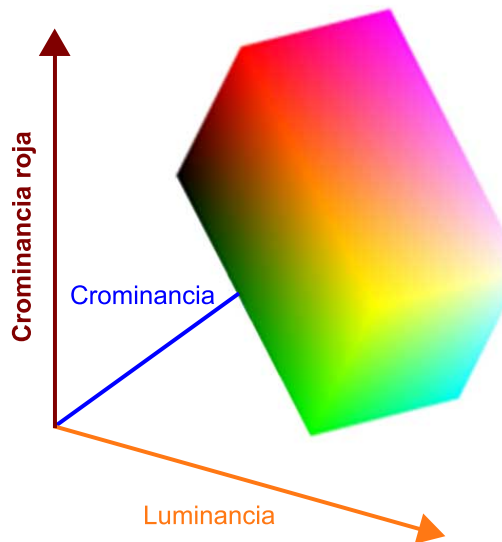


Ilustración 9. Canales de color.



Por aritmética pueden deducirse los valores  $Y$ ,  $Cb$  y  $Cr$  de una señal digital (para 8 bits por muestra de color) a partir de los valores RGB de un píxel. Suponiendo que el rango de valores  $R$ ,  $G$  o  $B$  es  $[0 \dots 255]$ , tenemos:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,169 & -0,331 & 0,500 \\ 0,500 & -0,419 & -0,081 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

donde el rango de valores posibles de  $Y$ ,  $Cb$  y  $Cr$  es  $[0 \dots 255]$ . La operación contraria es:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1,4 \\ 1 & -0,343 & -0,711 \\ 1 & 1,765 & 0 \end{bmatrix} \cdot \begin{bmatrix} Y \\ Cb - 128 \\ Cr - 128 \end{bmatrix}$$

La razón por la que es más habitual utilizar el espacio YCbCr que el RGB es doble: por una parte, descomponemos un color en un componente que representa la luminancia y en dos de color o crominancia, y así se da pie a poder procesarlos por separado; por otra, la estructura de bastones y conos del ojo sigue el mismo principio y adolece de menor sensibilidad al color que al brillo o luminancia. La comprobación es sencilla: ponemos un hilo de color rojo a 5 metros de un observador. Usualmente detectará la existencia del hilo, pero no su color. El color, parece ser, era poco relevante para la supervivencia de nuestros ancestros, como mínimo menos relevante que la intensidad de luz.

Conociendo, entonces, que se puede representar cada cuadro de imagen por sus tres cuadros equivalentes  $Y$ ,  $Cb$  y  $Cr$ , y sabiendo que la información visual está sobre todo en su cuadro  $Y$ , se podrá comprimir el cuadro  $Y$  con pocas pérdidas (es decir, una compresión más bien suave) y comprimir con más pérdidas (mayor compresión) los cuadros  $Cb$  y  $Cr$ .

#### Los valores $Y$ , $Cb$ y $Cr$

Estas fórmulas solo son válidas para aquellos formatos que permitan almacenar todo el rango de valores de cada componente, por ejemplo: JPEG.

Así ocurre en el estándar **JPEG de compresión de imagen fija**. La primera etapa de su proceso es convertir la imagen del espacio de color RGB (tal como se genera en el sensor de cámara) al espacio YCbCr.

### 1.6.2. Submuestreo de color

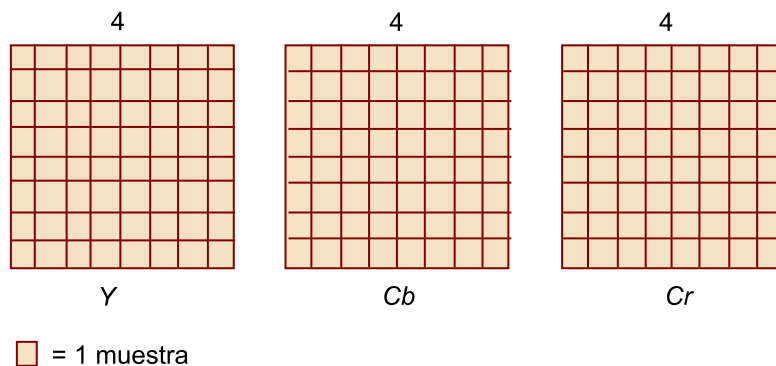
Ya que el ojo humano es menos sensible al color, no es imprescindible mantener todas las muestras de los componentes *Cb* y *Cr* de todos los píxeles, sino solo de algunos. Esta idea se aplica en el mero proceso de muestreo del color en cada imagen y se denomina **submuestreo**.

Así, se realiza un muestreo más denso en su equivalente *Y* y menos denso en sus equivalentes de color sin afectar seriamente a la calidad final.

La sistemática de muestreo está estandarizada en la familia MPEG, basada en YCbCr:

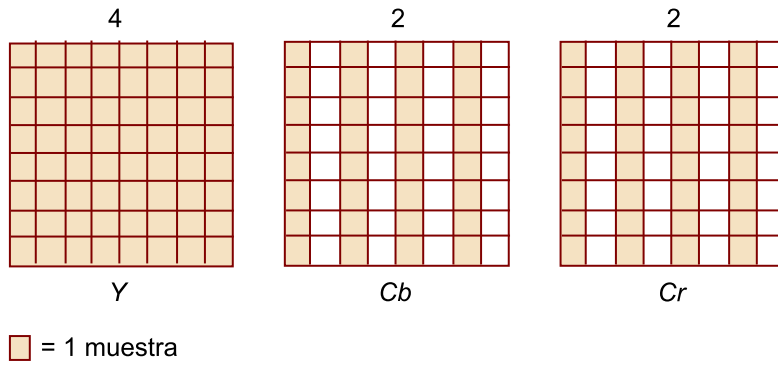
a) **4:4:4**: los tres componentes (*Y*, *Cb* y *Cr*) presentan la misma resolución y existe una muestra de cada componente en cada píxel. Solo es para entornos profesionales en los que no se desea compresión (HDCAM, MPEG-2, H.264):

Ilustración 10. Muestras de cada componente en el caso 4:4:4



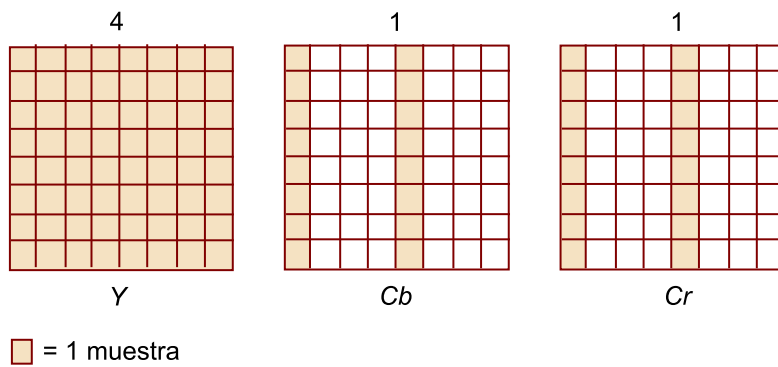
b) **4:2:2** (conocido también como YUY2): presenta dos componentes de crominancia por cada cuatro de luminancia (Digital Betacam, DVCPRO 50, MPEG-2, H.264):

Ilustración 11. Muestras de cada componente en el caso 4:2:2



c) **4:1:1**: presenta un componente de crominancia por cada cuatro de luminancia:

Ilustración 12



d) **4:2:0** (conocido también como YV12): presenta un componente de crominancia por cada cuatro de luminancia (DVD, MPEG-1, MPEG-2, MPEG-4), aunque según el estándar se selecciona un lugar diferente del cuadro para realizar la muestra de los componentes de color:

Ilustración 13. Muestreo de cada componente en el caso MPEG-1 (4:2:0)

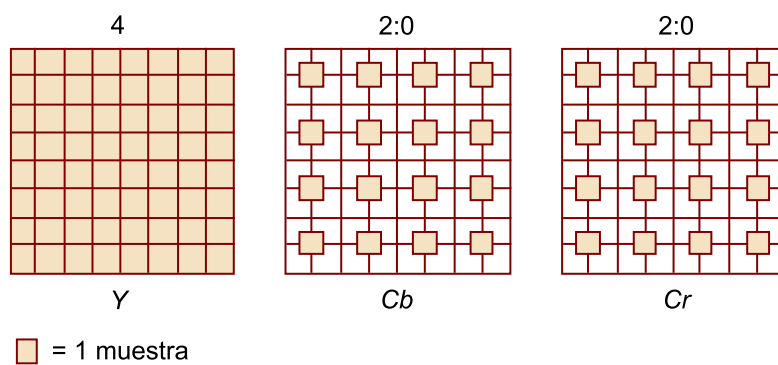
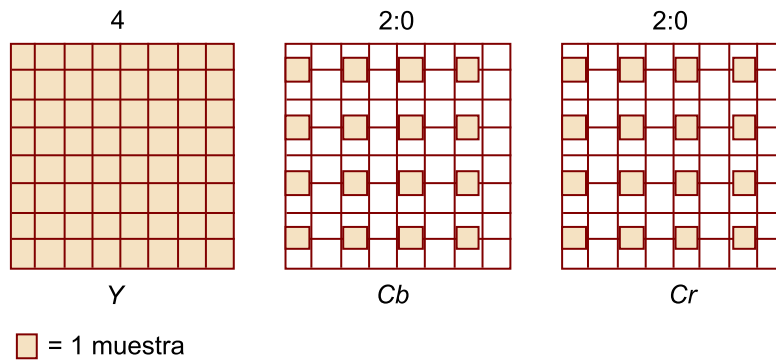


Ilustración 14. Y en el caso MPEG-2 (4:2:0)



En resumen, los tres patrones tienen una distribución especificada de los píxeles que se muestrean en cada caso:

- En el caso 4:4:4, se muestrean todos y cada uno de los píxeles de los tres componentes de cada cuadro.
- En el caso 4:2:2, se muestrean todos los píxeles del componente de luminancia  $Y$  y la mitad de cada componente de color.
- Finalmente, en 4:2:0, se muestrean todos los píxeles del componente de luminancia  $Y$ , y solo uno de cada 4 en los componentes de color, variando según la norma.

En todos ellos, cada muestra conseguida se representa posteriormente con 8 o 10 bits, según la precisión de la muestra que se desee.

## 1.7. Ejercicios

### Ejercicio 1

Buscad en Internet estimaciones de tamaño de los siguientes conceptos:

- ¿Cuántos Pbytes ocupa Facebook?
- ¿Cuántos Pbytes diarios gestiona Google?
- ¿En cuántos Ebytes se estima el tamaño total de Internet?

### Ejercicio 2

Calculad los valores  $Y$   $Cb$   $Cr$  de un color cuyos componentes RGB valen  $R = 60$ ,  $G = 190$  y  $B = 210$ . Efectuad la operación inversa para confirmar que los valores hallados son correctos realizando la operación inversa.

### Ejercicio 3

Dada una imagen en color de  $N \times M$  píxeles en la que cada componente  $R$ ,  $G$  y  $B$  es un valor almacenado en 8 bits (es decir, 24 bits por píxel):

- a) ¿Qué peso total en bytes tiene la imagen?
- b) Se transforma a YCbCr mediante un muestreo 4:4:4. ¿Qué peso poseerá ahora? ¿Qué factor de compresión se ha logrado?
- c) Si la transformación hubiese sido 4:2:2, ¿cuál sería su peso? ¿Qué factor de compresión se consigue?
- d) Realizad la misma operación si fuese 4:2:0.

## 2. Codificación de imagen fija

Las imágenes requieren considerablemente más espacio que el texto, y el audio y el vídeo aún demandan más en términos de almacenamiento y velocidad de transmisión.

Comencemos viendo el primer sistema de codificación de imagen fija e introduciendo una serie de conceptos que utilizaremos a lo largo de todo el curso.

### 2.1. El formato JPEG

JPEG es un comité (Joint Photographic Experts Group), no un estándar de codificación, que a finales de los años ochenta ideó un proceso por el cual una imagen digital puede comprimirse manteniendo un compromiso con la calidad visual.

En aquellos años, ya comenzaba a ser necesario algún sistema para que las imágenes no ocupasen los reducidos espacios de almacenamiento digital existentes en la época, y para poder ser transmitidas por líneas de telefonía digital de baja velocidad. En general, el proceso es suficientemente robusto y de resultados aceptables, especialmente con fotografías naturales e imágenes realísticas, con la que se llega a un promedio de compresión de 10:1 con pérdidas imperceptibles visualmente, por lo que sigue siendo un formato utilizado intensamente.

Por primera vez en esta asignatura, hemos de distinguir entre el sistema de codificación (**códec**) de la imagen, que es JPEG, y el **contenedor** o formato de fichero, donde se guardan los datos resultantes de la codificación, del cual hay dos tipos, JFIF y EXIF. Así, JPEG/JFIF es el formato clásico, mientras que JPEG/EXIF posibilita añadir metadatos (datos adicionales asociados a la imagen) como, por ejemplo, la fecha y hora de la captura, la velocidad del obturador, el nombre de la cámara, los datos de configuración, etc. Actualmente, algunos teléfonos móviles con cámara y GPS integrada añaden la longitud y latitud en que fue tomada. Nuestro interés se centra en el códec.

JPEG puede comprimir con pérdidas o sin pérdidas en función del modo seleccionable de trabajo, pero donde destaca es en los modos con pérdidas, consiguiendo compresiones de hasta 20-25 sin efectos visuales aparentes.

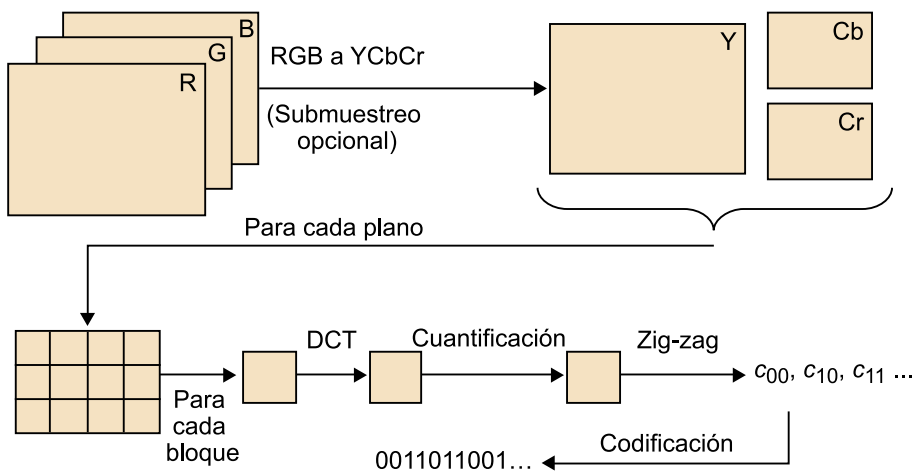


Los tres modos **con pérdidas** son: el más utilizado, el **modo secuencial**, con el que introduciremos varios conceptos de codificación relevantes en cada paso del proceso de codificación, el **modo progresivo** y el **modo jerárquico**.

### 2.1.1. Modo secuencial

Un esquema general de los diferentes pasos que se realizan en el modo secuencial es el siguiente:

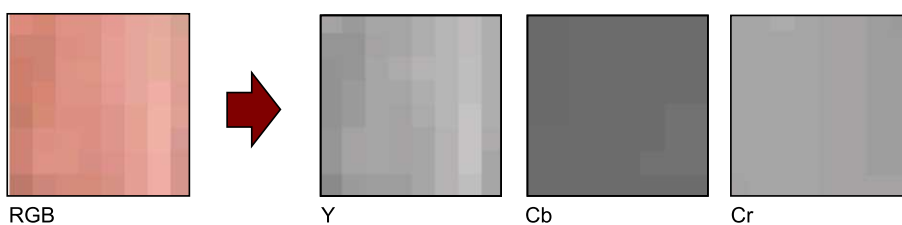
Ilustración 15



#### Paso 1. Cambio de espacio de color y submuestreo del color

Si la imagen es RGB, en primer lugar se transforma a YCbCr. Como ya vimos a principios de este módulo, ya que el ojo humano tiene menor resolución de color, las imágenes *Cb* y *Cr* se submuestran reduciendo su número de píxeles (en proporción 4:4:4, 4:2:2, 4:1:1 o 4:2:0 en función de la calidad elegida). La imagen *Y* (blanco y negro) se mantiene inalterada.

Ilustración 16



#### Paso 2. División en bloques y secuenciado

Cada uno de los tres componentes se comprime por separado, dividiéndose en bloques de 8 × 8 píxeles. Así, cada componente se convierte en una secuencia de bloques, comenzando por el de su esquina superior izquierda y acabando con el de su esquina inferior derecha. A partir de este punto, el proceso se realiza individualmente por cada bloque.

### Paso 3. Codificación de cada bloque

Cada bloque de imagen de 8 × 8 píxeles se transforma matemáticamente en otro bloque de 8 × 8 valores (también enteros) que ya no representan una pequeña imagen sino las frecuencias espaciales de esta. El bloque, pues, contiene la misma información que tenía pero con otros valores: un puro ejemplo de codificación. Y sin pérdidas hasta este punto, pues podríamos realizar el proceso inverso y recuperar exactamente la pequeña imagen original.

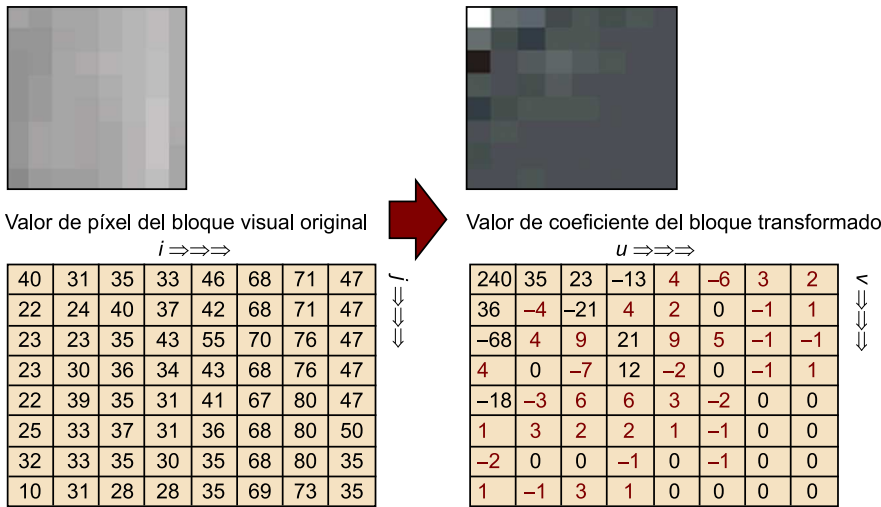
**8 × 8 píxeles**

Los píxeles son de valor entero de 0 a 255, pero en este punto se le resta a cada uno 128 para que estén en el margen de -128 a 127. Así, la transformación DCT siguiente da un rango de valores menos extenso y más fácil de codificar.

La transformación utilizada se denomina DCT<sup>5</sup> transformación coseno discreto, y es la base de la compresión de imagen y vídeo hoy en día.

<sup>(5)</sup>DCT es la sigla de la expresión inglesa correspondiente a *discrete cosine transform*.

Ilustración 17. Ejemplo de transformación DCT



Dado el bloque de 8 × 8 píxeles de la imagen original –en el que el valor del píxel de la columna  $i$  fila  $j$  es  $g_{i,j}$ – se puede generar cada uno de los coeficientes de la tabla transformada en donde  $g_{u,v}$  es el coeficiente de la frecuencia espacial horizontal  $u$  y de la vertical  $v$  mediante la fórmula de la DCT siguiente:

$$g_{u,v} = \alpha(u) \cdot \alpha(v) \cdot \sum_{i=0}^7 \sum_{j=0}^7 g_{i,j} \cdot \cos\left[\frac{\pi}{8} \cdot \left(i + \frac{1}{2}\right) \cdot u\right] \cdot \cos\left[\frac{\pi}{8} \cdot \left(j + \frac{1}{2}\right) \cdot v\right]$$

en donde  $\alpha(v)$  vale  $1/\sqrt{8}$  si  $u = 0$  o  $1/\sqrt{4}$  en el resto de casos, y los valores de  $u$  y  $v$  van de 0 a 7.

### Paso 4. Cuantificación de cada bloque

Si ambos bloques contienen la misma información, ¿por qué buena parte de los valores o coeficientes del segundo bloque son de valor muy reducido, incluso nulo, respecto a los restantes? Este es el quid del proceso: la transformación DCT compacta la información asociada a la imagen en los primeros coeficientes (situados cerca de la esquina superior izquierda), con lo que nos posibilitaría eliminar buena parte del resto con valores nulos o de bajo valor (indicados en rojo en la ilustración 17), reduciendo el número total de coeficientes que representan el bloque.

Pero ¿qué criterio elegir para eliminar los coeficientes pequeños? ¿Qué significa “pequeño”? Para automatizar la selección de los coeficientes que se van a eliminar en cada bloque de una manera razonable, se utiliza una tabla de referencia denominada **tabla de cuantificación**, que contiene  $8 \times 8$  valores de peso o ponderación, definidas tras muchas pruebas de calidad subjetiva realizadas con grupos de observadores. Esta tabla contiene pesos pequeños en las posiciones cercanas a la esquina superior izquierda y pesos crecientes cuanto más lejanos están de esta. Cada coeficiente de la imagen transformada se divide por el peso que le corresponde por posición, de manera que los alejados de la esquina superior izquierda quedan divididos por pesos elevados, dando resultado cero (se desprecian los decimales).

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

Ilustración 18. Una tabla de cuantificación utilizada en JPEG  
Fuente: Losheller.

Si realizásemos la transformación inversa, denominada IDCT, para recuperar el bloque de imagen original a partir del bloque transformado y cuantificado, el efecto visual de la pérdida de coeficientes sería mínimo. En el ejemplo, el hecho de descartar los indicados en rojo (ilustración 17) no afecta sensiblemente al resultado “destransformado” (ilustración 19).

Ilustración 19. Ejemplo de transformación DCT inversa



Valores de coeficiente del bloque transformado y debidamente cuantificado

15	3	2	-13	0	0	0	0
3	0	-1	0	0	0	0	0
-4	0	0	1	0	0	0	0
0	0	0	1	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	0
0	0	0	0	0	0	0	0



Valor de píxel del bloque visual destransformado resultante

40	30	33	32	42	59	65	44
22	23	41	33	45	61	69	44
23	22	36	46	51	73	73	43
22	31	36	33	41	70	75	44
23	38	33	31	40	63	81	43
24	32	34	29	32	63	75	53
31	29	33	32	32	60	76	35
8	29	25	25	34	67	75	35

Sin duda, almacenar en un fichero los coeficientes transformados que nos interesan en vez de todos los píxeles originales daría, sin duda, una buena tasa de compresión. Si además descartamos aquellos con valor despreciable, la tasa mejora sensiblemente minimizando el efecto visual de las pérdidas introducidas cuando se invierte la transformación y se recupera la imagen. Además, la DCT no necesita de un procesador potente de cálculo, pudiéndose ejecutar en hardware sencillo. Sin embargo, el proceso de compresión JPEG aún no ha acabado.

Hagamos una pausa en el proceso de explicación y razonemos:

a) ¿Cómo es posible que la DCT haga insensible visualmente la eliminación de buena parte de los 64 valores?

Tras años de pruebas y comparativas con todas y cada una de las transformaciones conocidas, la DCT es la que mejor condensa lo importante en menos datos sin necesidad de mucha potencia de cálculo (factor que en los años ochenta era crítico: el 80386 de Intel era el procesador del momento hasta 1989). Hay transformaciones mejores, pero todas necesitan cálculo intenso superior, como por ejemplo JPEG2000, que veremos más tarde en este mismo módulo.

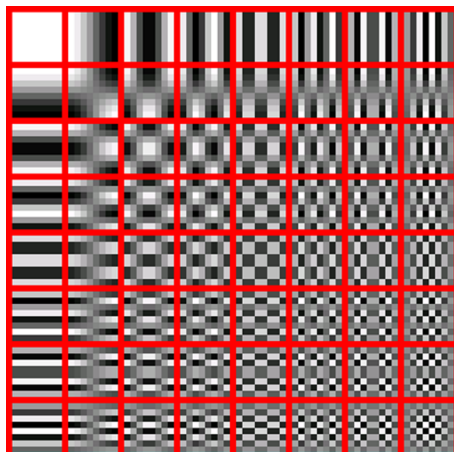
b) ¿Qué significa visualmente cada uno de los 64 coeficientes?

Tal como se ha comentado, estos datos generados por la DCT representan frecuencias espaciales, es decir, cada coeficiente, por sí solo, generaría un bloque de imagen simple correspondiente a una oscilación visual.

### **Ejemplo completo de la imagen generada por cada coeficiente por separado**

La imagen adjunta es un resumen del tipo de bloque de imagen generado por cada coeficiente en función de su posición en el bloque transformado. Por ejemplo, el coeficiente que se ubica en la esquina superior izquierda genera, por sí solo, un bloque de imagen de  $8 \times 8$  píxeles como el indicado en esa misma esquina, un bloque visualmente homogéneo. El segundo coeficiente genera un bloque de imagen de  $8 \times 8$  píxeles similar a una onda horizontal de medio periodo, el tercero una onda horizontal de un periodo, etc.

Ilustración 20. Ejemplo completo de la imagen generada por cada coeficiente por separado



Pues bien, cualquier fragmento de fotografía en blanco y negro de  $8 \times 8$  píxeles que se elija puede representarse como una suma de estos 64 bloques, debidamente ponderados (multiplicados) por un valor. Si se tuviese tiempo (años) y paciencia, podría irse probando sumas de estos bloques debidamente amplificadas cada uno por un valor a nuestra elección. El día que consiguiésemos la suma perfecta que reprodujese exactamente el fragmento de fotografía, descubriríamos que los 64 valores seleccionados coinciden con los generados por la sencilla DCT.

El primer coeficiente del bloque es el peso o valor de la frecuencia cero, que denominamos **componente de continua o baja frecuencia (visual)**, el cual, como puede verse (ilustración 21), recrea el valor medio de brillo de todo el bloque y genera por sí solo un bloque visual de  $8 \times 8$  homogéneo. Así, si borrásemos todos los componentes del bloque salvo este y aplicáramos la transformada inversa para volver a tener un bloque de imagen, tendríamos lo que llamamos un efecto mosaico.

El ojo humano es muy sensible a este valor medio y se percata de cualquier error en este valor, de tal manera que cualquier variación que haya entre dos cuadros consecutivos genera un efecto mosaico. Por ello, JPEG mima el valor de este coeficiente como ningún otro, como veremos en el paso siguiente.

Los 63 coeficientes restantes se llaman **coeficientes de alterna**, o **coeficientes de alta frecuencia**, y son responsables de las variaciones de brillo a partir de este nivel de base. Los bloques de imagen con mucho detalle visual, al aplicárseles la DCT, tienen muchos coeficientes de este tipo con gran valor, al contrario que los bloques de poco detalle.

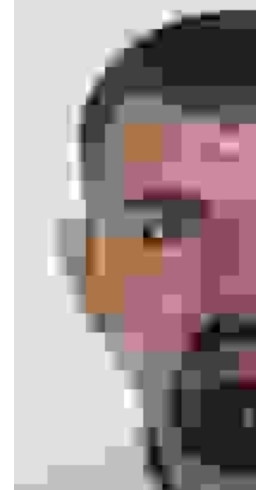
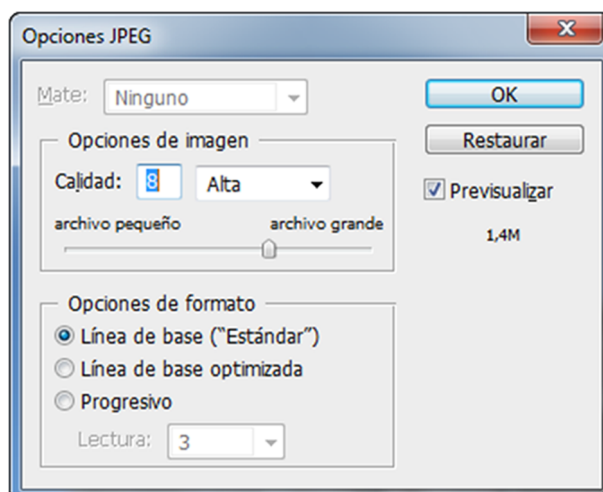


Ilustración 21. Imagen codificada solo con el componente de continua de cada bloque

### Calidad de imagen y las tablas de cuantificación

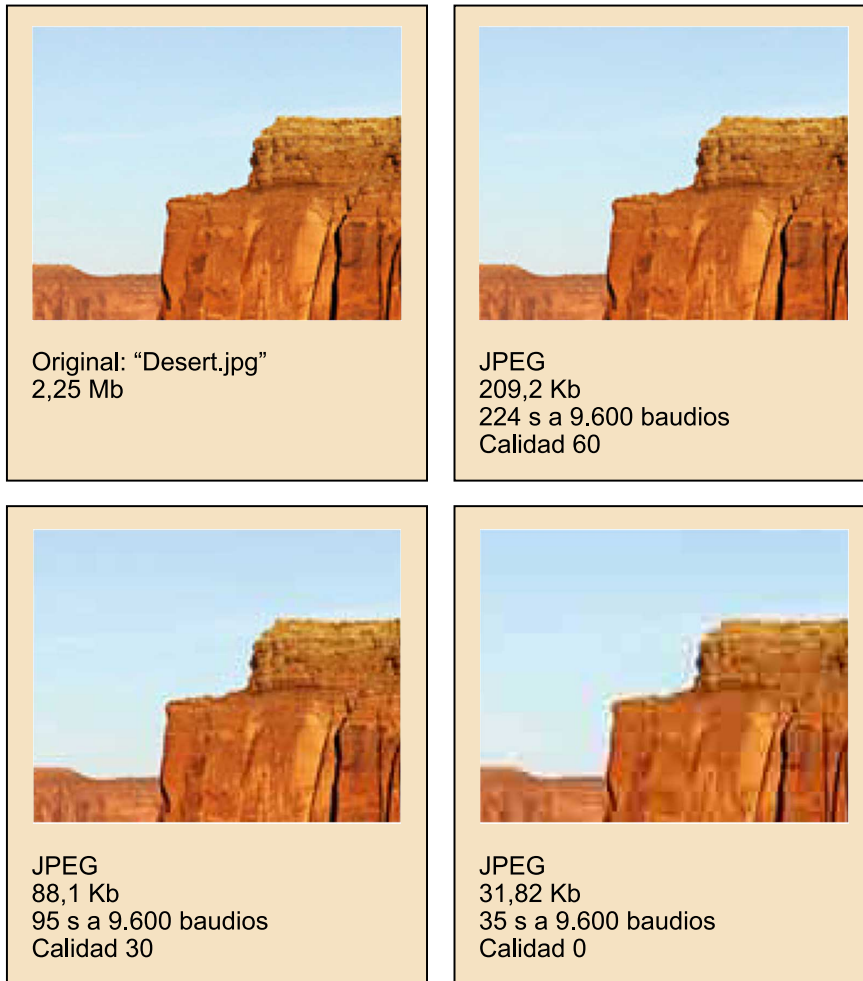
Cuando se desea codificar una imagen en formato JPEG, el software (por ejemplo, Photoshop) pregunta el nivel de calidad que deseamos mantener de la imagen original en la nueva imagen JPEG.

Ilustración 22. Opciones de compresión JPEG de Photoshop



Usualmente es un parámetro cuantitativo (como un valor de 0 a 10, o de 0 a 12, o un porcentaje de 0 a 100%, etc.) siendo el primero la peor calidad y máxima compresión, y el último máxima calidad y mínima compresión. En otras ocasiones, posibilita seleccionar la calidad o compresión mediante valores cualitativos (de “menor detalle” a “mayor”), como por ejemplo en las configuraciones de las cámaras digitales.

Ilustración 23. Cuadro comparativo de compresiones JPEG sobre una misma imagen



Este parámetro de calidad seleccionado por el usuario está íntimamente ligado a la tabla de cuantificación que se vaya a aplicar para eliminar los coeficientes de alta frecuencia de todos los bloques transformados de la imagen. Así, a mayor valor de calidad, los valores de la tabla de cuantificación se reducen, por lo que la división da un resultado mayor y "sobreviven" más coeficientes de alta frecuencia. Cuando la calidad es máxima, los valores de la tabla de cuantificación son todos 1, y no afectan en absoluto.

El valor de calidad también puede permitir seleccionar al codificador entre las técnicas de submuestreo de color 4:2:2 o 4:2:0, para reducir, sensiblemente, el número de bloques de color y mejorar la compresión sin necesidad de ninguna acción por parte del usuario.

### Paso 5. Codificación entrópica de los coeficientes

Si bien los 64 coeficientes son enteros, en este paso se convierten a binario y se almacenan en fichero mediante una codificación que reduce la necesidad de bits para su almacenamiento. Ya que el coeficiente de continua es tan importante, se trata con una codificación específica.

En primer lugar, se ordenan siguiendo un orden en zigzag establecido (ilustración 24).

Siguiendo nuestro ejemplo, el zigzag da como resultado:

15, 3, 3, -4, 0, 2, -13, -1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ..., 0

A continuación, se codifican de la siguiente manera:

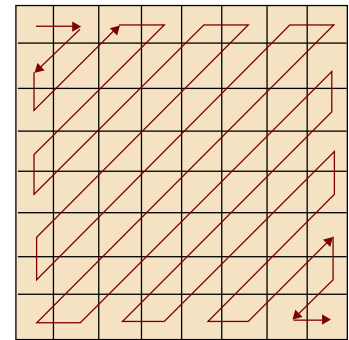


Ilustración 24. El camino en zigzag de ordenación de los coeficientes

**a) Coeficiente de baja frecuencia (de continua):** guardar la diferencia entre dos valores ocupa casi siempre menos espacio que guardar los dos valores, siempre y cuando estos valores sean parecidos. Es lo que se denomina **codificación diferencial (DPCM)**, y se aplica al coeficiente de continua de cada bloque: ya que en la mayoría de imágenes el valor medio de un bloque es usualmente similar al del bloque inmediatamente anterior, se almacena el valor del primero y, posteriormente, las diferencias de cada uno con el siguiente coeficiente de continua. Si además se almacena en el fichero JPEG antes que el resto de coeficientes, tendremos la imagen mosaico de manera inmediata como vista preliminar de esta. Por ejemplo, el coeficiente de continua de este bloque es 15. Si el del anterior bloque fuese 23, lo que almacenaríamos sería 7, la diferencia entre ellos.

**b) Coeficientes de alta frecuencia (de alterna):** en la ristra secuencial de valores generada por la ordenación en zigzag aparecen usualmente muchos valores repetidos, la mayoría ceros, por lo que se aplican técnicas de compresión entrópica que aprovechen esta característica (**codificación Huffman** o, más modernamente, **codificación aritmética**).

Los coeficientes de alta frecuencia se codifican de manera que a su lado les preceda el número de ceros que tienen a su izquierda. Así, nuestra ristra de 63 coeficientes queda simplemente como:

(0,3) (0,3) (0,-4) (1,2) (0,-13) (0,-1) (2,-1), (6,1) (6,1) EOB

siendo EOB un marcador que indica “fin de bloque”.

Ahora, si es una codificación Huffman, se utiliza una tabla en la que se asigna el conjunto mínimo de bits necesarios para almacenar cada número:

Categoría	Valor del coeficiente	Representación en bits
1	-1, 1	0, 1
2	-3, -2, 2, 3	00, 01, 10, 11
3	-7, -6, -5, -4, 4, 5, 6, 7	000, 001, 010, 011, 100, 101, 110, 111

Categoría	Valor del coeficiente	Representación en bits
4	-15, ..., -8, 8, ..., 15	0000, ... 0111, 1000, ..., 1111
5	-31, ..., -16, 16, ..., 31	00000, ... 01111, 10000, ... 11111
6	-63, ..., -32, 32, ..., 63	000000, ..., 011111, 100000, ... 111111
7	-127, ..., -64, 64, ..., 127	0000000, ..., 0111111, 1000000, ..., 1111111
8	-255, ..., -128, 128, ..., 255	00000000 ... 11111111
9	-511, ..., -256, 256, ..., 511	000000000 ... 111111111
10	-1023, ..., -512, 512, ..., 1023	0000000000 ... 1111111111
11	-2047, ..., -1024, 1024, ...2047	00000000000 ... 11111111111

El número derecho de cada pareja se codifica mediante esta tabla, indicando además la categoría, y el EOB se convierte siempre a (0,0):

(0,2,11) (0,2,11) (0,3,011) (1,2,10) (0,4,0010) (0,1,0) (2,1,0) (6,1,1) (6,1,1) (0,0)

Los dos primeros valores siempre pueden representarse con 1 byte: sus 4 bits más altos se encargan de guardar el número de ceros previos, y los 4 bits bajos la categoría del valor. Se hace una tabla de Huffman a partir de cierta política de frecuencias de aparición y este valor de 8 bits de cada pareja se reduce, manteniéndose el tercer valor. Por ejemplo:

(0,2,11)  $\Rightarrow$  00 11  
 (0,2,11)  $\Rightarrow$  00 11  
 (0,3,011)  $\Rightarrow$  010 011  
 (1,2,10)  $\Rightarrow$  0011 10

Mientras tanto, el coeficiente de continua que ha quedado reducido a una diferencia de valor 7 se codifica también con esta tabla; por ejemplo, resulta: 11110.

La ristra final que se almacena en el fichero JPEG es 11110, 0011, 0011, 010011, 001110, etc.

Según la manera en que se almacenen en el fichero los componentes de color, se habla de **componentes entrelazados** (en el que se van almacenando los bloques de luminancia y de color alternativamente) o **componentes no entrelazados** (primero se almacenan los de luminancia y posteriormente los de color).



## Decodificación de una imagen JPEG

El fichero JPEG posee toda la información necesaria para recuperar la imagen comprimida según el valor de calidad con el que se generó, ya que no solo incluye los coeficientes transformados de todos los bloques, sino la información necesaria para recuperarlos a partir de sus versiones codificadas, el valor de calidad seleccionado y la tabla de cuantificación seleccionada para eliminar elementos.

### 2.1.2. Otros modos de trabajo de JPEG

El modo secuencial base explicado es válido para la mayoría de aplicaciones de tratamiento de imagen en que estas poseen 8 bits por cada uno de los componentes (RGB) e incluso una cuarta capa de transparencia.

Sin embargo, existen otros modos (hasta 44) para aplicaciones específicas. Por ejemplo, para imágenes de muy alta calidad con 12 bits por componente, o para aplicar otros sistemas de codificación de coeficientes más efectivos (codificación aritmética en vez de Huffman).

#### a) Modo secuencial base optimizado

Una versión mejorada del modo secuencial base, el **modo secuencial base optimizado**, permite una mejora del 2% al 8% en el tamaño del fichero final. Sin embargo, algunas aplicaciones pueden no ser compatibles con este modo.

#### b) Modo progresivo

El segundo modo interesante con pérdidas, el **modo progresivo**, muy utilizado para imágenes de sitios web cuyos servidores tienen conexiones de baja velocidad o tienden a estar muy saturados, posibilita ver una imagen mosaico (es decir, la imagen con solo el coeficiente de continua de cada bloque) y sus sucesivas mejoras de calidad en función del orden de recepción de los datos, con lo que el espectador puede tener vistas parciales cada vez más detalladas de la imagen que va recibiendo.

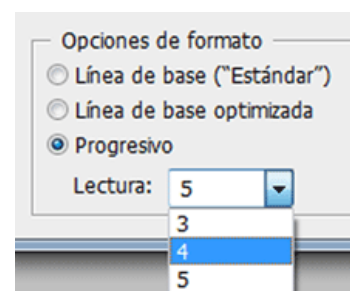
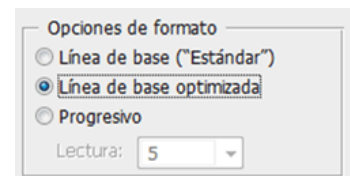
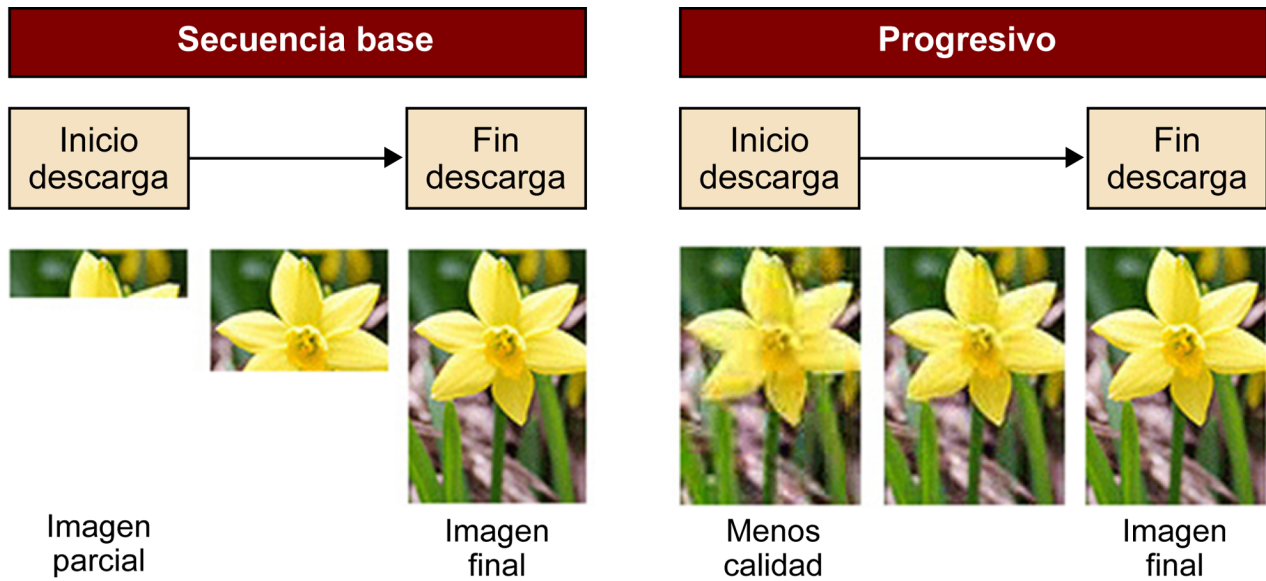


Ilustración 25. Secuencia de descarga de una imagen JPEG secuencial base (izquierda) y la misma imagen codificada en modo progresivo (derecha)



El tiempo de descarga es aproximadamente el mismo, pero la experiencia de espera del usuario es diferente.

Hay varias maneras de implementar esta solución, aunque la más generalizada es almacenar en el fichero JPEG, en primer lugar, los coeficientes de continua y los primeros coeficientes de alterna de todos los bloques, posteriormente coeficientes de alterna de menor relevancia visual (mayor detalle), después de relevancia aún menor (y más detalle), etc. El proceso usualmente realiza tres, cuatro o cinco fases de separación de los coeficientes. Un fichero JPEG en modo progresivo tiene, normalmente, un tamaño en bytes superior que su equivalente en modo secuencial o secuencial optimizado.

### c) Modo jerárquico

El tercer modo con pérdidas es el jerárquico. El más complejo, pues provee de representaciones progresivas en resolución de la misma imagen, y requiere más espacio. En esencia, comprime la imagen en múltiples resoluciones de manera que la imagen de más baja definición puede ser accesible sin tener que descomprimir la imagen al completo.

A partir de la imagen original, la filtra y submuestra en los múltiplos de 2 que se deseen, codifica la imagen reducida así en tamaño, decodifica esta imagen reducida e interpola una segunda imagen en un factor de 2 horizontal y verticalmente, utilizando el mismo algoritmo de decodificación que usaría el receptor. Aprovecha esta imagen como la predicción de la original y codifica la imagen diferencia (es decir, el error). Luego vuelve al principio y repite el proceso.

### d) Modo sin pérdidas

Para finalizar, el **modo JPEG-LS** (JPEG sin pérdidas<sup>6)</sup>) es un modo creado unos años más tarde (1993) que codifica la imagen sin pérdidas. Utiliza una DCT para codificar, pero no utiliza tablas de cuantificación, sino que hace previsiones de los píxeles a partir de sus píxeles vecinos. Destinado a aplicaciones médicas de imagen, consigue unos factores de compresión de 2:1, no teniendo relevancia fuera de este campo.

<sup>6)</sup>En inglés, *lossless JPEG*.

## 2.2. JPEG 2000

Una década más tarde de la aparición de JPEG, el escenario había cambiado sensiblemente: se había investigado con éxitos nuevas transformaciones que superaban a la clásica DCT aun obligando a un cálculo intensivo, pero la capacidad de cálculo de los equipos domésticos y profesionales quintuplicaba el de diez años antes y posibilitaba un nuevo formato de codificación que daba respuesta a las nuevas exigencias del mercado, entre otras:

- Codificación con pérdidas o sin pérdidas
- Alto factor de compresión con menores artefactos visibles
- Robustez contra errores para poder transmitir a baja velocidad, etc.
- Capacidad de añadir datos adicionales o metadatos
- Mejor codificación de imágenes sintéticas o bitono

Lamentablemente, la situación legal del formato JPEG2000 fue siempre controvertida, ya que existen patentes sobre partes del proceso, lo que ha frenado desde el principio su popularización.

JPEG da calidades subjetivas de imagen muy bajas en compresiones menores de 0,25 bpp, donde JPEG2000 aún puede dar calidad incluso hasta en 0,1 bpp. Y en los rangos de compresión habituales, este nuevo sistema de codificación supera a JPEG.

El secreto es la **transformación *wavelet***, que trata la imagen como un todo (no la descompone en bloques), aplicándole varios filtros espaciales y aprovechando de manera muy eficiente la codificación de los resultados. Según los filtros que se utilicen, esta transformación puede ser sin pérdidas o con pérdidas, mejorando en este último caso la compresión a costa de algún efecto visual en los contornos (la alta frecuencia espacial).

Así, si JPEG2000 es capaz de comprimir con calidad a mayor tasa, es mejor para transmitir imagen a baja velocidad. Como, además, el tipo de decodificación es progresivo, la imagen recibida se decodifica a partir de una primera

### Patentes sobre la compresión JPEG2000

Buena parte del software de tratamiento de imágenes no incluye las partes del proceso patentadas, o lo hacen opcionalmente, como Adobe Photoshop. En el caso de formatos de compresión sin pérdida, actualmente PNG es el más extendido.

versión básica (baja frecuencia espacial) que mejora en detalle, de ahí que resulte perfecto para transmisión: si se interrumpe en cierto momento, al menos tendremos una imagen completa parcialmente detallada.

El estándar posibilita incluir datos adicionales (metadatos) en el fichero de imagen. JPEG ya lo posibilitaba en su formato EXIF, pero JPEG2000 lo hace de manera más estandarizada siguiendo el formato XML.

### 2.3. JPEG XR

En el 2006, Microsoft introdujo un codificador propietario de imagen, inicialmente denominado HD Photo y dirigido, en exclusiva, al sector de la fotografía digital que, desde el 2009, año en que se convirtió en estándar ISO, es conocido como JPEG XR<sup>7</sup>. La extensión de los ficheros de imagen es .wpd.

<sup>(7)</sup>Del inglés, *extended range*.

Sus características principales son:

- Codificación con pérdidas o sin pérdidas
- Decodificación progresiva
- Mejor compresión que JPEG
- Un abanico de formatos amplio: 8, 16 o 32 bits por canal de color, los píxeles pueden tener valor entero o incluso flotante para adaptarse a cámaras fotográficas de rango dinámico muy alto, y un formato en blanco y negro de 16 bits.
- El formato contenedor puede incluir metadatos y un canal alfa.
- No está destinado a ser utilizado en imágenes sintéticas o con bitonos (texto o gráficos).

Si bien solo Windows soportó este formato desde su versión Vista y existían algunos *plug-ins*<sup>8</sup> comerciales (por ejemplo, para Adobe Photoshop ofrecidos por Microsoft), su avance es lento frente al estándar abierto de imagen DNG (Adobe, 2004). La liberación de JPEG XR en el 2010 por parte de Microsoft quiso darle una posibilidad de éxito al facilitar su uso en aplicaciones de código abierto, pero aún es un tema abierto.

<sup>(8)</sup>Del inglés, *complementos*.

### 2.4. Tendencias futuras de la codificación de imagen

A mediados de los años ochenta, el matemático Michael Barnsley presentó la base de un sistema de compresión de imágenes con pérdidas basado en fractales que generó gran expectación.

Un fractal es un elemento sencillo, el cual, iterado sobre sí mismo repetidas veces, genera un nuevo elemento complejo. La idea de Barnsley era generar una imagen original a partir de un elemento visual básico mediante iteraciones. Así, solo se debería almacenar o transmitir el fractal o grupo de fractales, no la imagen. Incluso, según esa idea, era posible reconstruir la imagen original a resoluciones superiores a las de partida con muy poca capacidad de cálculo, es decir, era un método independiente de la resolución, denominado **escalado fractal**.

En compresiones normales, la compresión fractal consigue resultados similares a los de los algoritmos basados en la DCT, como por ejemplo JPEG, con ratios de 50:1. A mayores compresiones, supera en calidad a la DCT. Curiosamente, a mayor riqueza visual de la imagen (mayor alta frecuencia espacial), la codificación fractal mejora.

El problema principal residía en cómo encontrar de manera sencilla el fractal o elemento visual básico (o el conjunto de fractales básicos) que pudiese lograrlo. La necesidad de una gran capacidad de cálculo para conseguir deducirlos y la falta de un algoritmo eficiente para todo tipo de imágenes ha frenado hasta la fecha este sistema de compresión, aunque la investigación sigue abierta incluso para su aplicación en vídeo.

## 2.5. Ejercicios

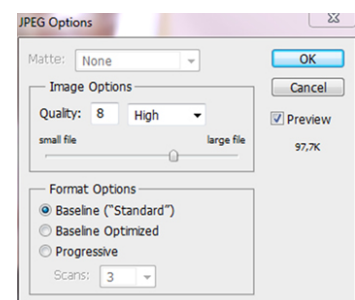
### Ejercicio 4

Se comprimirá una imagen en formato JPEG mediante Photoshop para ver su efecto en la calidad de la imagen y en el tamaño del fichero resultante. Para ello, abrid la aplicación y leed una imagen. A continuación, mediante la opción del menú ARCHIVO > GUARDAR COMO seleccionad un nombre nuevo y elegid como formato de grabación JPEG (\*.JPG, \*.JPEG, \*.JPE, las tres extensiones habituales que se añaden al nombre del fichero). Tras aceptar, aparece la siguiente ventana, en la que se pregunta la calidad visual en la que se desea comprimirlo (un valor de 0 a 12). Por ahora, mantened como opción el modo *baseline* (estándar), del cual se hablará en el apartado siguiente.

Tras aceptar, Photoshop almacenará el fichero comprimido. Visualizad el resultado y, mediante el sistema operativo que se posea (Windows, OSX, etc.), tomad nota del tamaño en Kbytes del fichero resultante. Puede calcularse sencillamente el factor de compresión.

### Orientación

Este ejercicio también se puede hacer con la aplicación de software libre Gimp.



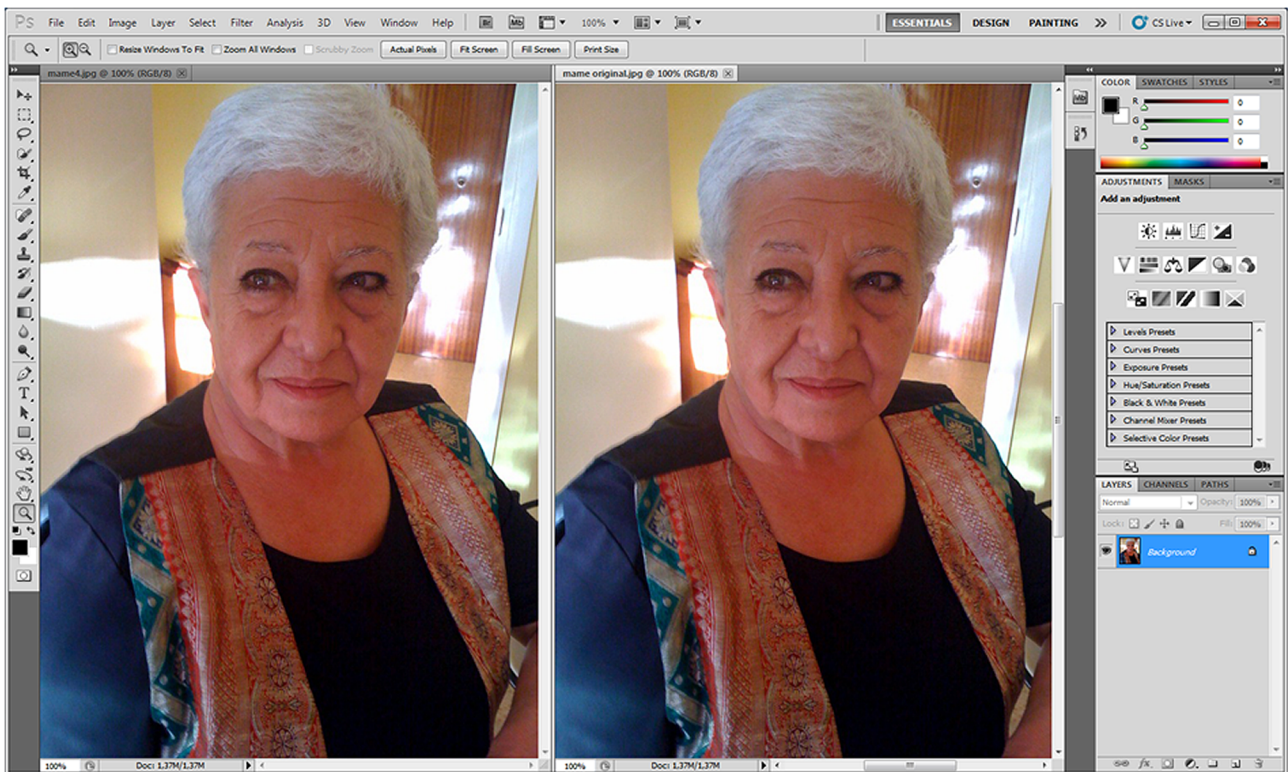
## Ejercicio 5. Cómo crear una imagen diferencia con Photoshop

El factor de compresión, como podéis observar, puede dar una idea aproximada del resultado, pero no es útil si se desea descubrir realmente dónde se genera la pérdida de información visual. Para ello, se puede crear una imagen diferencia que sea la resta píxel a píxel entre la imagen sin comprimir y la comprimida; en este caso, en JPEG.

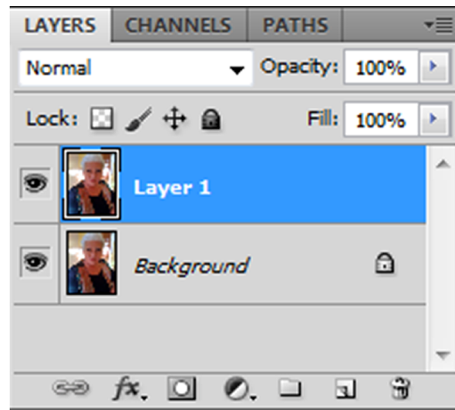
### Orientación

Este ejercicio también se puede hacer con la aplicación de software libre Gimp.

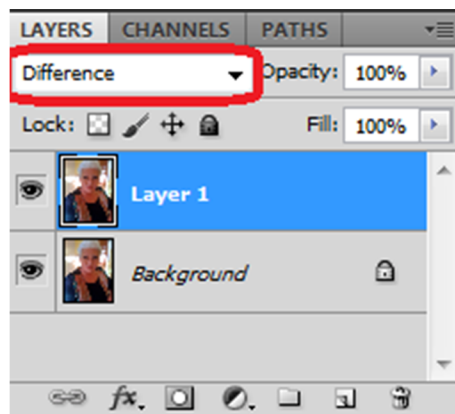
Abrid Photoshop, leed la imagen original y también la imagen comprimida (se aconseja elegir una imagen comprimida en un factor de 4 a 6, así los errores serán más evidentes).



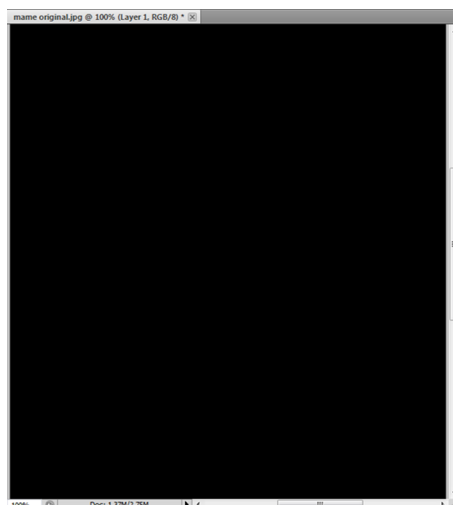
A continuación, seleccionad con un clic la imagen original sin comprimir y, mediante CTRL-J (Windows) o COMMAND-J (Mac), duplicad su capa fondo (o *background*, según el idioma del programa) en la paleta de capas. El resultado son dos capas: la capa fondo original abajo y una copia que Photoshop denomina *Capa 1* (*Layer 1*).



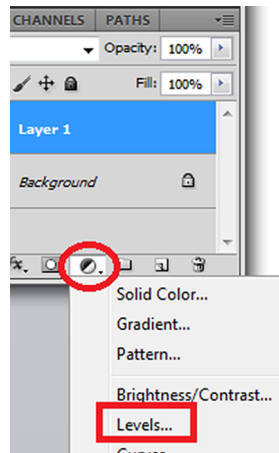
Ambas capas son idénticas. Como demostración, presentaremos la imagen diferencia: se puede avanzar que, si ambas son iguales, la diferencia entre píxeles correspondientes será 0, con lo que todos los píxeles de la imagen diferencia serán de valor 0, es decir, totalmente negros. En el desplegable de la ventana de capas, elegid *Diferencia (Difference)*:



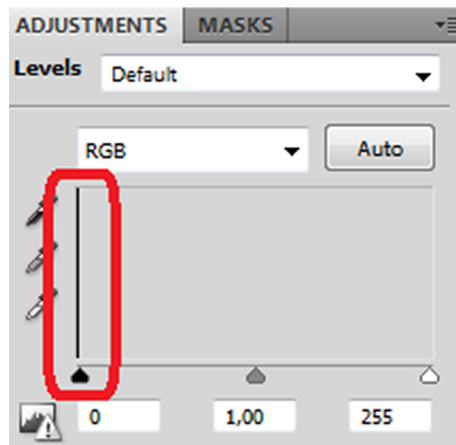
Tal como era previsible, la **imagen diferencia** aparece totalmente negra.



Seamos escépticos y confirmemos que es así de una manera más explícita. Para ello, seleccionamos la herramienta de ajuste y, de entre las opciones de que se dispone, elegimos *Niveles (Levels)*.

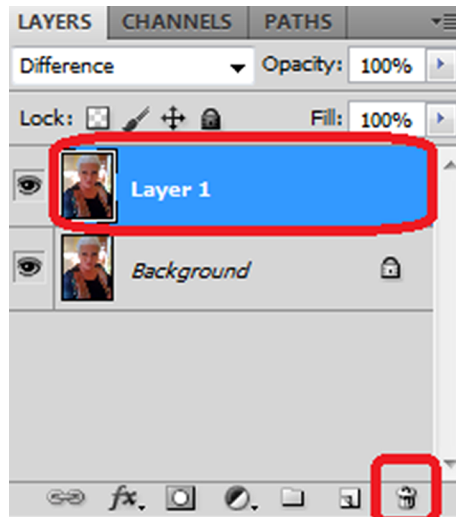


Se activará una ventana de diálogo en la que se refleja el histograma de la imagen diferencia. Es muy estrecho porque todos los píxeles son del mismo valor, no se distribuyen en otros valores más que el cero.



Una vez demostrada que la imagen diferencia entre dos imágenes iguales es negra, borrad la *Capa 1 (Layer 1)* seleccionándola con un clic en la ventana de capas y pulsando la papelera situada en esa misma ventana.

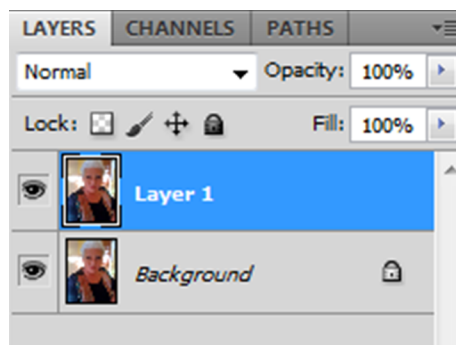




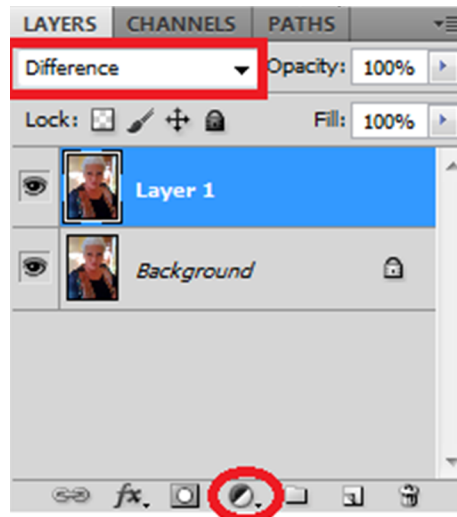
Ahora que se sabe cómo realizar una imagen diferencia y hemos vuelto al punto de partida, volvamos a efectuar la operación, pero con las dos imágenes que aún están abiertas en Photoshop. Seleccionamos la herramienta *Mover* en la barra de herramientas, pulsamos el ratón sobre la **imagen comprimida** sin soltarlo, arrastramos el ratón hasta la imagen sin comprimir, pulsamos y mantenemos la tecla *Mayus*, y finalmente soltamos el ratón. Si bien visualmente no parece haber habido ningún efecto, en la ventana de capas se verá que se ha creado una segunda capa, *Capa 1* o *Layer 1*, que es la imagen comprimida, mientras que *Fondo*, o *Background*, es la imagen sin comprimir.



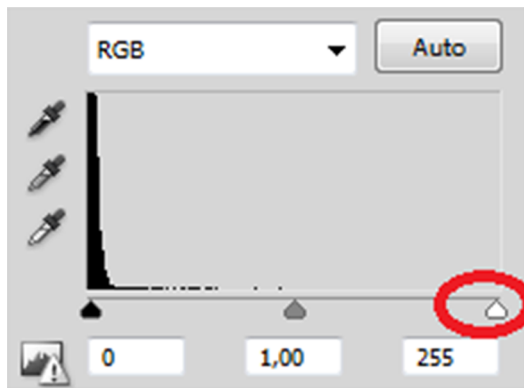
Herramienta  
*Mover*



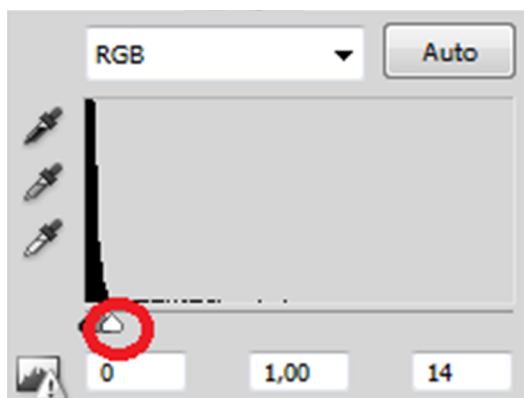
Ya se tienen ambas imágenes en capas, por lo que se puede volver a realizar la operación *Diferencia* (aparecerá esta, aproximadamente negra) y a continuación, la herramienta de capas, eligiendo *Niveles*:



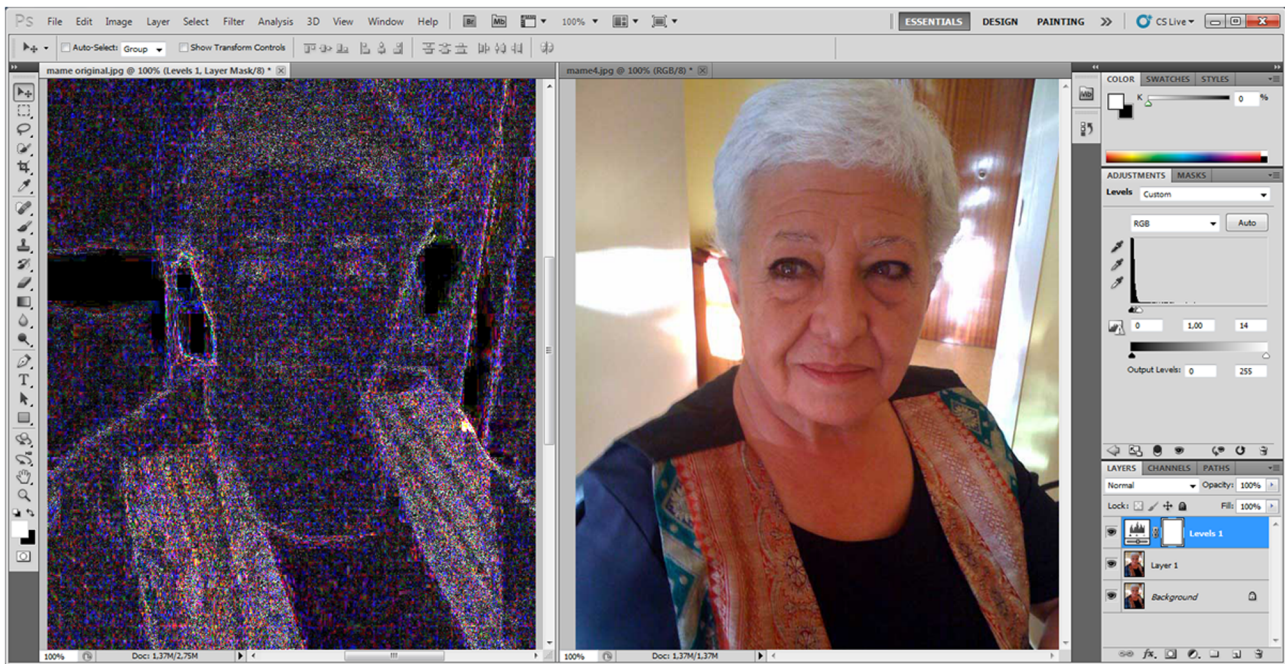
Casi con seguridad, el histograma ya no será tan uniforme:



Para poder discernir con claridad los píxeles diferencia, cabe desplazar el *slider* blanco hacia la izquierda y así realzar visualmente sus valores.



Claramente, todos los errores generados por la codificación con pérdidas de JPEG se reflejan situados en zonas específicas de la imagen, como los bordes de los objetos y, en general, las zonas con gran detalle visual.



Realizando el proceso con niveles de compresión más elevados, aparecen igualmente errores en zonas de color homogéneo y un claro efecto de bloques en general, intrínsecos a la manera en que JPEG trabaja la imagen.

### Ejercicio 6

Visualizad la imagen diferencia en el caso de compresión JPEG en modo “Línea de base optimizado” (“BaselineOptimized”) y valorar la diferencia con JPEG “Standard” (“Baseline”) utilizando la misma imagen original del ejercicio anterior y comprimiéndola a la misma calidad. Valorad la calidad visual y factor de compresión de la resultante.

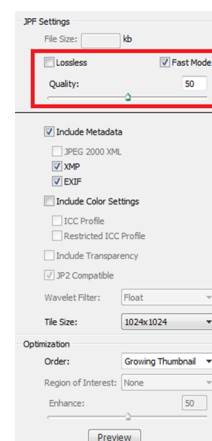
### Ejercicio 7

Realizad el ejercicio anterior eligiendo en este caso el modo “Progresivo” (“Progressive”) con cuatro lecturas. Mediante un navegador o programa de visualización, abrid el fichero de imagen generado y comprobar que se visualiza con una mejora gradual.

### Ejercicio 8

Photoshop lee y graba en formato JPEG2000 (extensiones \*.JPF, \*.JPX, \*.JP2, \*.J2C y \*.JPC), aceptando una configuración personalizada en cada grabación. Entre las opciones de personalización de que se dispone, destacamos si se desea una compresión “Sin pérdidas (“Lossless”) en una calidad indicada por un valor de 0 a 100.

Realizad algunas compresiones con este formato en modo con pérdidas y probar el formato sin pérdidas, valorando el nivel de compresión conseguido por lo que se refiere al tamaño del fichero.



**Ilustración 26.** Opciones de codificación con o sin pérdidas de JPEG2000

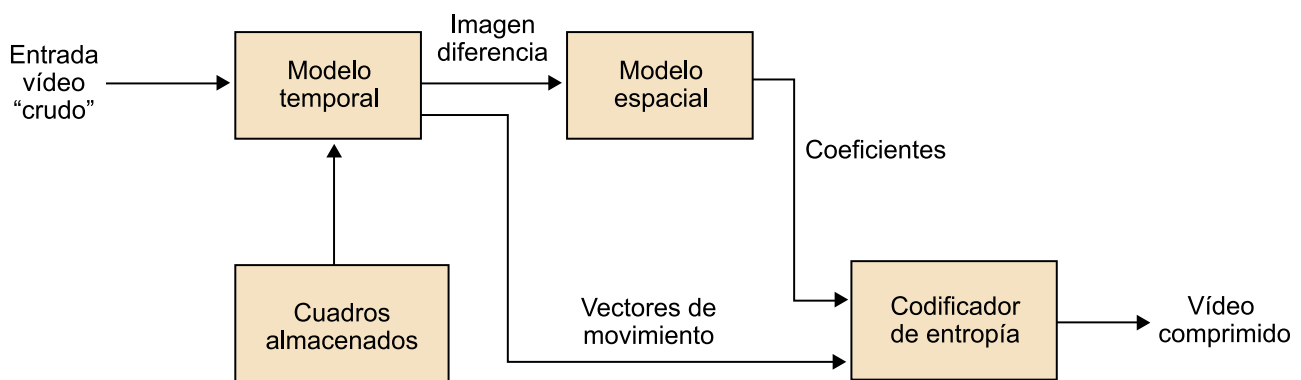
### 3. Codificación de vídeo y audio

Vista la codificación de imagen fija, los mismos conceptos de redundancia espacial, temporal y psicovisual son aplicables incluso con mayor éxito en audio y vídeo. Sin embargo, las metodologías para aprovechar estas redundancias son más complejas habida cuenta de la complejidad de este tipo de información y el volumen de datos por segundo que genera.

#### 3.1. Modelos de codificación de audio y vídeo

La codificación con pérdidas impera en casi todos los estándares de compresión existentes para ellos, pues son los únicos que actualmente consiguen ratios de compresión interesantes. Buena parte de ellos pueden representarse por el siguiente esquema, que representa un **codificador genérico de vídeo con pérdidas (códec)**:

Ilustración 27. Modelo básico de un codificador de vídeo con pérdidas



La familia MPEG, H.264, Theora, VC-1, etc., son ejemplos de códecs que utilizan este modelo, realizando con mayor o menor gracia cada una de estas etapas:

a) La primera fase es el **modelo temporal**, el cual, a partir de los cuadros almacenados del vídeo leídos hasta el momento, “prevé” el siguiente cuadro que llegará por la entrada de vídeo “crudo”. Esta previsión, usualmente, no está especificada y depende de cada fabricante, lo que condiciona buena parte de la calidad final de todo el proceso. Por ejemplo, un codificador MPEG-2 de la empresa Tandberg utiliza un planteamiento diferente de previsión del siguiente cuadro que el codificador MPEG-2 de la empresa Thomson. Sin duda, el funcionamiento del modelo temporal de cada una de estas empresas es un secreto bien guardado.

b) La siguiente fase se centra en el **modelo espacial**. La imagen diferencia es comprimida utilizando la similitud de sus píxeles: “comprimir una diferencia” sería básicamente un resumen en tres palabras de todo el proceso. Cada siste-

ma de compresión de vídeo utiliza aquí sus propias técnicas de compresión de imagen, aunque las utilizadas por H.264, por ejemplo, no distan mucho grosso modo de las del estimado JPEG. Sin duda, la mayoría con pérdidas. Y manteniendo esa idea en mente, al igual que JPEG, el resultado es una ristra de valores binarios que también son denominados coeficientes espaciales, realizando el mismo papel: representando la imagen resultante con otro vocabulario matemático.

c) Finalmente, en una última vuelta de tuerca, tanto los coeficientes de la imagen como los vectores del movimiento detectado en la misma son empaquetados de forma eficiente en el **codificador de entropía**, que analiza la frecuencia de aparición de estos coeficientes y vectores y les asigna menos bits cuanto más frecuentes son, reduciendo así el tamaño final del fichero o del flujo de datos de transmisión<sup>9</sup> al mínimo posible (o al valor más cercano a ese mínimo).

<sup>(9)</sup>En inglés, *stream*.

El resultado es una ristra binaria compacta, que contiene toda la información necesaria para poder ser descomprimida. Los sistemas de descompresión están claramente definidos, de manera que en este caso no hay diferencia alguna entre decodificadores de una empresa u otra.

### Ejemplos de estándares

Actualmente, los dos estándares de códecs más habituales en los sistemas de IPTV son H.264 (también llamado MPEG-4 Part 10 o AVC), desarrollado en conjunto entre ITU-T y MPEG, y VC-1 (núcleo de Windows Media Audio y Windows Media Video 10), desarrollado por Microsoft. Desde el 2010, Google TV, con el códec VP8, intenta hacerse un sitio apoyando sin fisuras WebM.

Para los sistemas que utilizan redes basadas en IP, como Internet y las redes locales, o servicios IP sobre 3G como medio de transporte, las compresiones más utilizadas son:

a) **MPEG-2, H.264/MPEG-4 AVC y 3GPP**: son formatos estándares principalmente utilizados por QuickTime en su línea de servidores Darwin/QuickTime Stream Server y en su reproductor QuickTime. Igualmente, Adobe en su Flash Video Server utiliza H.264 como códec desde su versión 9. Cabe destacar que la utilización de los estándares MPEG requiere el pago de licenciamiento tanto en servidores como en reproductor. Por su calidad, H.264/MPEG-4 es también utilizado en almacenamiento de vídeo en Blu-Ray y en las emisiones HD por antena, cable o satélite, mientras que MPEG-2 ha quedado relegado como estándar de almacenamiento para DVD y emisiones digitales en calidad estándar.

b) **Windows Media**: codificación propietaria desarrollada por Microsoft; tiene su fortaleza en la transmisión en tiempo real de bajo ancho de banda (como las de Internet). Puede ser servido por la línea de servidores Microsoft utilizando el servicio Windows Media Services y reproducido por el Windows Media Player. Esta codificación requiere un permiso especial y pago de licencia a Microsoft para poder ser utilizado en otra línea de servidor/reproductores, por lo que su uso está en declive frente a H.264 o VC-1.

c) **VP8**: es el códec abierto de vídeo para transmisión en tiempo real de WebM, la propuesta de formato de Google para ser utilizado en HTML 5. De calidad semejante a H.264, se basa en una biblioteca de software libre denominada x264 bajo licencia GNU. Su contenedor se basa en el contenedor abierto Matroska.

d) **Theora**: formato abierto cuyo principal atractivo es que no requiere el pago de licencia para su utilización. Presenta una gama de estándares en continuo desarrollo que pretende competir con la línea de estándares MPEG-4. La codificación de vídeo Theora está basada en la codificación On2's VP3, y su contenedor más conocido para almacenamiento es el formato Ogg.

e) **RealVideo**: codificación propietaria, desarrollada por RealNetwork, empresa pionera en transmisión de datos de audio y vídeo en tiempo real en Internet actualmente en desuso. Presentaba su fortaleza en transmisión en tiempo real de bajo ancho de banda.



Podía ser servido por la línea de servidores Helix y reproducido por el RealPlayer (o sus antecesores HelixPlayer y PlayerOne), todos de la misma empresa RealNetwork. Esta codificación requería un permiso especial y pago de licencia a RealNetwork para poder ser utilizado en otra línea de servidor/reproductores, por lo que el coste de este códec y la falta de inversión en innovación ha supuesto su casi desaparición del mercado actual.

### 3.2. Compresión de audio digital

Las técnicas de compresión de audio son cronológicamente anteriores a las de vídeo. Como en este, hay técnicas con pérdidas y sin pérdidas, algunas específicas para transmitir en tiempo real y otras para almacenamiento.

Todas estas técnicas se basan en la reducción de la redundancia que hay en la señal para minimizar el flujo de datos generado por segundo, sin que resulte afectada la calidad del audio según sus necesidades.

Del gran abanico de estándares existentes, unos se centran en la codificación de la voz humana para su transmisión en redes digitales. Servicios de transmisión de voz en tiempo real, como la telefonía fija, la telefonía móvil o Internet, consiguen grandes tasas de compresión al centrarse en las características específicas de esta señal. Es habitual que se basen en técnicas de predicción (en breve, se transmiten las diferencias entre las muestras de audio previstas y las reales).

#### Estándares de compresión de audio digital

Algunos ejemplos ilustrativos:

- Las redes GSM y UMTS codifican el audio en formato AMR, tanto en su emisión como en el almacenamiento de mensajes en el buzón de voz (extensión .amr), y es el formato base del grupo 3GPP para futuras redes móviles de cuarta generación.
- Skype se basa en su propio estándar Silk (SuperWideband Audio Codec).

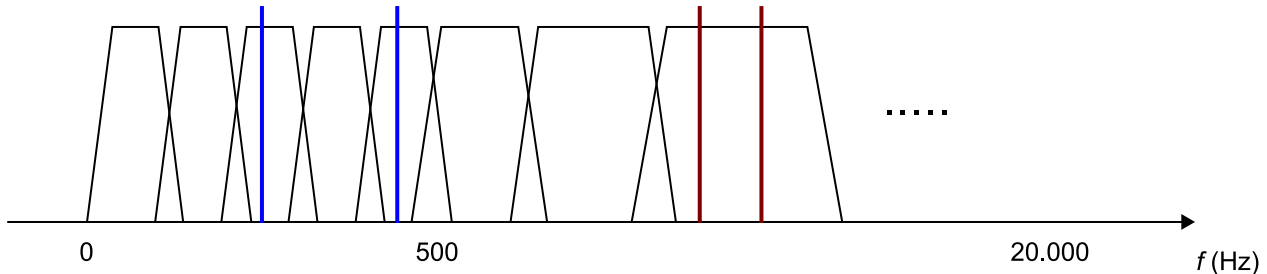
Otro gran grupo, más interesante en nuestro caso, tiene como objetivo conseguir la máxima calidad de sonido con el mínimo número de bits por segundo, incluso posibilitando sonido multicanal. Se basa en la codificación perceptual, un avance en el mundo de la codificación de audio implementado por primera vez por Philips en su ya desaparecido sistema de audio compacto digital (DCC), el cual pretendía sustituir la casete de cinta, y base del MP3.

La **codificación perceptual** nace de los estudios sobre cómo el ser humano interpreta el sonido.

Aparecen tres patrones de codificación perceptual que abren la puerta a comprimir el audio sin una pérdida de calidad audible:

1) El oído no detecta todos los sonidos que recibe, se comporta como un “banco de filtros” que más o menos se solapan en frecuencia. Dos sonidos situados en bancos diferentes son diferenciados claramente por el cerebro, pero no lo son si ambos están en el mismo banco de frecuencias.

Ilustración 28



Dos tonos (en azul) situados en frecuencias de diferentes bandas son diferenciados por el oído, mientras que dos situados en la misma banda (en rojo) son interpretados como uno solo.

2) La sensibilidad del oído ante un sonido cambia si hay más sonidos. De alguna manera, unos sonidos enmascaran otros, haciéndolos indetectables al oído. Este fenómeno se denomina **enmascaramiento frecuencial**.

3) Si escuchando un sonido débil, este se detiene, se tardará cierto tiempo hasta poder escuchar un sonido débil otra vez. Este segundo fenómeno se denomina **enmascaramiento temporal**.

El sistema está fundamentado en los datos experimentales recogidos por miles de pruebas en voluntarios, de manera que el modelo psicoacústico puede interpretarse como un comportamiento medio del oído humano. Existen, por tanto, sujetos acústicamente mejor capacitados que otros, que pueden detectar la pérdida de calidad inherente a esta codificación.

De los códecs de audio habituales actualmente, la mayoría sigue este modelo o variaciones de este, y otros han tendido hacia sistemas de compresión más conservadores.

### Códecs de audio habituales

- a) El formato de sonido **MP2** (conocido como Musicam) es el habitual en ficheros de audio profesional.
- b) El estándar MPEG-1 Audio Layer 3, nacido a partir de MP2, es un clásico de la familia de códecs con pérdidas. Se definió para almacenamiento de audio digital doméstico. Se conoce como **MP3**.
- c) **Vorbis** es también un estándar de audio con pérdidas resultado de un proyecto de software libre ofrecido como alternativa al estándar propietario MP3. Ofrece una calidad de audio similar, e incluso mejor a bajas velocidades de transmisión o lectura (< 64 Kbps).
- d) **AAC** (Advances Audio Coding), posterior y de mayor calidad que MP3; es el existente en equipos domésticos portátiles de audio de Apple (iPod, iPhone, iPad), la familia de móviles Android, Sony (PlayStation 3), Nintendo (Dsi, Wii), etc. Soporta audio multicanal (hasta 48 canales) y es una codificación con pérdidas.
- e) **Dolby AC-3** (conocido como Dolby Digital) es un estándar propietario con pérdidas nacido para el cine, pero su alta eficiencia (320 Kbps para 5 canales de audio de alta fidelidad más un sexto canal de *surround*) y una buena relación con los fabricantes de



equipos lo alzó como estándar de audio para DVD y Blu-Ray, e incluso para la TV digital en Estados Unidos (y, por empatía, también accesible en algunas TV europeas).

f) **Dolby Digital Life**, sin embargo, es un estándar de codificación en tiempo real dirigido al mercado multimedia, en especial el sector de videojuegos.

g) **Windows Media Audio** (WMA) es la apuesta de Microsoft por un estándar de codificación. Existe tanto con pérdidas como sin pérdidas (WMA Lossless).

h) Y, solo como comentario, **Real Audio** fue un estándar de compresión con pérdidas específicamente diseñado para transmisión en tiempo real, muy utilizado en los primeros años de Internet, pero su naturaleza propietaria y la falta de mejoras por parte de la empresa Real Networks lo llevó a la desaparición hace unos años.

### 3.3. Diferenciando contenedor y códec

Al almacenar la información audiovisual en un fichero para su almacenamiento y/o transmisión, se utiliza un formato de contenedor.

Un **formato de contenedor** es un formato de archivo digital que almacena determinado tipo de información codificada con uno o varios códecs estándares.



Ilustración 29

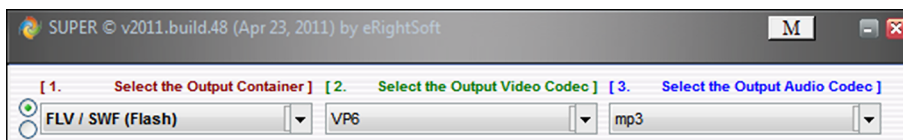
Un ejemplo de asociación pura entre códec de vídeo y contenedor

Es habitual confundir ambos conceptos y mezclar el nombre del códec con el del contenedor. Así, Flash Video es un formato contenedor cuya extensión de fichero es .flv o .f4v y en el que el vídeo puede estar codificado en H.264 o en otro códec, como SorensonSpark o VP-6, mientras que el audio puede estar codificado en MPEG-1 Audio Layer 3.

En algunos casos, el nombre del contenedor acaba sustituyendo el del códec por costumbre o simplificación. Por ejemplo, el contenedor para el códec MPEG-1 Audio Layer 3 es comúnmente conocido por MP3, y la extensión de los archivos es .mp3, pero es habitual hablar de “codificación MP3”.

Los contenedores más complejos admiten distintos códecs de audio y video y son capaces de manejar subtítulos, capítulos y metadatos (o *tags*). Uno de los roles más importantes del contenedor es propiciar información temporal para poder sincronizar la reproducción de más de un flujo de datos simultáneamente.

Ilustración 30



Buena parte de las aplicaciones de codificación de audio y vídeo indican claramente los códecs de vídeo y audio posibles de un contenedor seleccionado (en la imagen, Super ©).

Algunas características que diferencian a los contenedores entre sí son:

- Su popularidad (y soporte por parte de las aplicaciones).

- La sobrecarga<sup>10</sup> (distintos contenedores presentan distintos tamaños de archivo para el mismo contenido).
- El soporte de códecs (por ejemplo, algunos contenedores como AVI no admiten codificaciones con cuadros B).
- El soporte de subtítulos u otras características avanzadas.
- El hecho de que sea apto para los servidores de *streaming* como formato de entrada.

<sup>(10)</sup>En inglés, *overhead*.

La mayoría de los contenedores presenta la posibilidad de flujos de bits variables tanto para audio como para vídeo. Alguna de las excepciones más notorias es, por ejemplo, el contenedor AVI de Microsoft, que no lo permite. Actualmente, cuando un contenedor no cuenta con alguna característica deseada, los distintos programas realizan extensiones para poder soportarlas, aunque muchas veces estas extensiones son incompatibles entre sí, complicando el escenario. La siguiente tabla resume las características de los contenedores más populares:

Contenedor	Características			
	Códec de vídeo	Códec de audio	Transmisión de datos en tiempo real	3D
.3gp	MPEG-4 Parte II H.264/MPEG-4 AVC	AAC AAC v2	-	-
.avi	La mayoría salvo H.264/AVC	La mayoría	Sí	-
.divx	MPEG-4 Parte II	MPEG-1 Audio AC-3 PCM	Sí	-
(Flash Video) .f4v	H.264/MPEG-4 AVC	AAC MPEG-1 Audio	Sí	.
(Flash Video) .flv	H.264/MPEG-4 AVC SorensonSpark VP6	AAC MPEG-1 Audio PCM	Sí	-
.(Matroska) mkv .mka .mks .mk3d	La mayoría	La mayoría	Sí	Sí
(MPEG) .mp4	MPEG-2 Parte II H.264/MPEG-4 AVC H.263 VC-1	AC-3 MPEG-2 MPEG-4 Vorbis	Sí	Sí
(MPEG) .mpg .mpeg	MPEG-1 MPEG-2	MPEG-1 Layer I,II,III	Sí	-

Contenedor	Características			
	Códec de vídeo	Códec de audio	Transmisión de datos en tiempo real	3D
(MPEG PROGRAM STREAM) .ps	MPEG-1 MPEG-2 MPEG-4 Parte II H.264 VC-1	MPEG-1 Layer I,II,III	Sí	-
(MPEG TRANSPORT STREAM) .ts	MPEG-1 MPEG-2 MPEG-4 Parte II H.264 VC-1	MPEG-1 Layer I,II,III	Sí	-
(BLU RAY TRANSPORT STREAM) .m2ts	MPEG-2 Parte II H.264 VC-1	Dolby AC-3 Dolby Digital Plus DTS DTS HD	No aplicable	Sí
(AVID) .mxf	La mayoría	La mayoría	No aplicable	-
(XIPH.ORG) .ogg	Theora y muchos otros	Vorbis y muchos otros	Sí	-
(Apple) .qt .mov	MPEG-1 MPEG-2 parte II MPEG-4 Parte II H.264/MPEG-4 AVC Cinepak	AAC MPEG-1 Audio PCM	Sí	-
(REAL) .rmvb	Real Video	Real Audio AAC Vorbis	Sí	-
(Microsoft) .wma .wmv	La mayoría salvo H.264/AVC	La mayoría	Sí	-
(Google) WebM	VP8	Vorbis	Sí	Sí

## 4. El estándar MPEG-2

El grupo MPEG<sup>11</sup> de la ISO<sup>12</sup> es el grupo más destacado y aceptado por la industria en lo que respecta a estandarización de vídeo digital.

<sup>(11)</sup>MPEG es la sigla de Moving Pictures Experts Group.

Tras el éxito obtenido con el estándar MPEG-1 en 1993, el grupo decidió realizar un estándar más completo y mejorado, dirigido a un mercado potencial atractivo y de gran escala como el de *broadcast*<sup>13</sup> de televisión digital. Finalizado en 1995, el estándar presenta con respecto a su antecesor mejoras en la compresión, codificación entrelazada (además de progresiva) y gran flexibilidad debido a la utilización de perfiles y niveles. MPEG-2 estandariza tanto el vídeo como el audio que lo acompaña.

<sup>(12)</sup>ISO es la sigla de International Standards Organisation.



<sup>(13)</sup>Del inglés, difusión amplia.

Los perfiles y niveles son variaciones de calidad de compresión que permiten adaptar el estándar a las diferentes exigencias de los sistemas particulares, además de ofrecer a los fabricantes reglas más claras respecto a la conformidad de sus productos con el estándar. Los perfiles y niveles de MPEG-2 los veremos en unas páginas.

El estándar MPEG-2 es muy utilizado en la actualidad, aunque lentamente remplazado por MPEG-4 Part 10/H.264/AVC, que veremos más adelante, y que comparte muchos detalles con MPEG-2. Es el utilizado para los sistemas de *broadcast* de calidad estándar de televisión digital (DVB), tanto por cable (DVB-C) como por satélite (DVB-S) o terrestre (DVB-T), y está presente en el mercado doméstico a través del formato de almacenamiento DVD.

### Ved también

Los estándares MPEG-4 Part 10/H.264/AVC se estudian en el apartado 5 de este módulo didáctico.

Sin ánimo de entrar en detalles profundamente técnicos, haremos una visión general del estándar a partir de las partes que lo conforman: primero veremos el apartado más importante, la definición del formato multimedia, mientras que veremos muy por encima los flujos de datos y la forma en que se empaquetan.

El almacenamiento y la transmisión de flujos MPEG-2 se estudian con más detalle en otros módulos. Aquí daremos solo una primera aproximación a este estudio. Una breve reseña del códec estandarizado en MPEG-2 y una descripción de los perfiles y niveles de calidad permitidos cerrarán este apartado.

## 4.1. Antecedentes

Algunos estándares anteriores a MPEG-2 fueron fuente de inspiración para MPEG-2 y, en algún caso, están aún vigentes: JPEG/JPEG2000, MPEG-1 para CD-Video y H.261 y H.263 para videoconferencia.

### a) JPEG/JPEG2000

Similar al grupo de trabajo MPEG, el JPEG<sup>14</sup> es un grupo de trabajo de ISO, en este caso especializado en la compresión de imágenes. En 1992 terminó el estándar JPEG<sup>15</sup>. La compresión espacial de la codificación de vídeo digital se encuentra fuertemente ligada a las técnicas presentadas por el grupo JPEG.

<sup>(14)</sup>JPEG es la sigla de Joint Photographic Experts Group.

<sup>(15)</sup>ISO/IEC 10918-1 / ITU-T Recommendation T.81.

En el 2000, se presentó una versión mejorada del estándar conocido como JPEG2000<sup>16</sup> con un nuevo sistema de codificación más eficiente que, además, posibilitaba codificar de manera escalada e incluso con compresión sin pérdidas. En su momento, no tuvo tanta popularidad al tener una licencia, pero desde el 2011 su uso parece resurgir.

<sup>(16)</sup>ISO/IEC 15444.

### b) MPEG-1 para CD-Video

El primer estándar del grupo MPEG, es el MPEG-1<sup>17</sup>. Culminado en 1993, actualmente sigue siendo utilizado. El caso más notorio de éxito es el códec de audio MP3<sup>18</sup>.

<sup>(17)</sup>ISO/IEC 11172.

<sup>(18)</sup>Especificado en la norma MPEG-1 Part 3 Audio Layer 3.

Para vídeo, el códec MPEG-1 fue utilizado en el formato Video CD (o VCD) actualmente reproducible en la mayoría de los reproductores DVD y de calidad similar a la de un vídeo VHS doméstico.

### c) H.261 y H.263 para videoconferencia

En 1993, la ITU-T estandariza el H.261, un códec para servicios de videoconferencia de baja velocidad (se transmitía por RDSI a 64 kbps o múltiplos de este). Pensado para redes conmutadas como la de telefonía analógica pero con servicios digitales como RDSI, actualmente es el único contexto en que sigue siendo utilizado, en general por razones de compatibilidad hacia atrás.

En 1998, la ITU-T estandariza el H.263, un códec de vídeo de mayor calidad y menor flujo de bits resultante (30 kbps). En este códec aparece el concepto de distintos perfiles de codificación (variaciones de calidad en la compresión que generaron inicialmente algunos problemas de compatibilidad entre los fabricantes, resueltos cuando en H.264 se clarifican las especificaciones).

## 4.2. Qué es MPEG-2

Entremos en MPEG-2. El estándar MPEG-2 se encuentra completamente definido en la ISO/IEC 13818. Esta norma se encuentra dividida en partes según el tema. Cada parte se considera un estándar en sí mismo; por tanto, de forma más correcta: MPEG-2 es un conjunto de estándares.

Las partes que conforman el estándar MPEG-2 son, entre otras:

- El sistema: cómo sincronizar y conjuntar los flujos de vídeo y audio en un solo flujo de datos.
- El códec de vídeo para señales entrelazadas y progresivas.
- El códec de audio, que es una extensión del MPEG-1 audio (también conocido como MP3) capacitado para sonido multicanal.
- El control del flujo por parte del usuario final (para reproducir, pausar, situarse en un punto en particular, estructurar en escenas, etc.).

La propiedad intelectual del estándar MPEG-2 es compartida por múltiples corporaciones. Más de 500 patentes forman el estándar. Los vendedores de productos y servicios basados en el estándar MPEG-2 deben pagar por la explotación de la licencia. La institución beneficiaria es MPEG-LA, la cual administra el conjunto de patentes MPEG.

## 4.3. Conceptos básicos: GOP Y MPEG

Cuando nos sentamos cómodamente ante nuestro equipo para disfrutar de un DVD en nuestro reproductor, o de un fichero en formato MPEG-2 en nuestro equipo multimedia u ordenador, y deseamos ir a un punto en particular, el avance rápido o el retroceso de nuestro mando a distancia no es fluido. Antes (mucho antes), cuando avanzamos o retrocedíamos en una cinta de vídeo como VHS o Betamax, podíamos ver con más o menos claridad la película acelerada, pero ahora la imagen va dando saltos en el tiempo de segundos en segundos, presentando instantáneas de la película.

¿Dónde está el resto de fotogramas? ¿Por qué no veo a los actores moviéndose de forma acelerada? La culpa es del GOP<sup>19</sup> (grupo de imágenes).

<sup>(19)</sup>Del inglés, *group of pictures*.

### 4.3.1. GOP

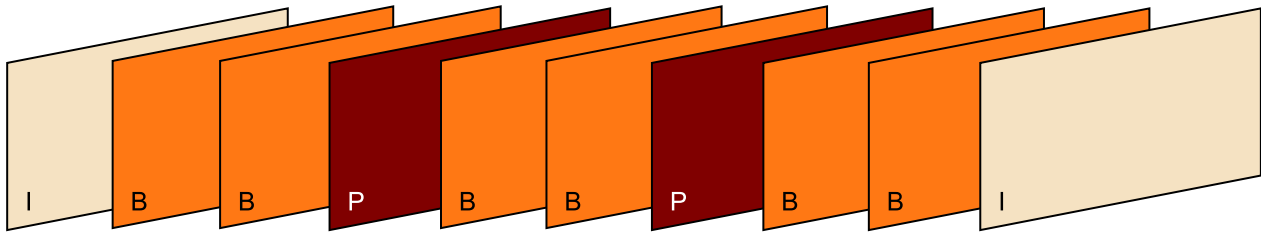
La imagen en movimiento puede comprimirse con más pérdidas que la fija, pues entre una imagen y la siguiente suele haber pocas diferencias (salvo un cambio de plano o una transición rápida). La intención es, de todas ellas, mantener intactas algunas (es decir, con una compresión similar al JPEG, denomi-

#### Ved también

El término *intraframe* se explica en el subapartado siguiente, con las imágenes I.

nada *intraframe*) que servirán de referencia al resto, que se calculará como diferencia o desplazamiento de estas. Las imágenes de referencia son denotadas con la letra I (de *intracode*).

Ilustración 31. Ejemplo de GOP de  $M = 9$  y  $N = 3$



Orden de visualización.

## Imágenes I

La imagen I no tiene ninguna referencia con las otras, se comprime espacialmente “al estilo JPEG” (es decir, se divide en bloques de  $8 \times 8$  píxeles y se transforma, se cuantifica, se codifica, etc.). Se codifican ellas solas, por ello se denominan *intracode*. La codificación utilizada es la DCT bidimensional, similar a la vista en JPEG, y al igual que JPEG, en cada imagen I se cuantifica sus coeficientes, se serializan en zigzag y se codifican.

Las imágenes I son las que nos aparecen en pantalla cuando realizamos una búsqueda rápida hacia adelante o atrás en nuestro reproductor DVD o software: un reproductor de baja calidad no necesita otras imágenes, solo unos pocos cálculos para poder descomprimir la imagen I. Es una manera económica de presentarnos el punto aproximado de la película en donde estamos.

Si codificásemos todas las imágenes de una secuencia como imágenes I, no tendríamos una gran tasa de compresión y la película no cabría en un DVD, pues el fichero sería de decenas de gigabytes. Hay que buscar un método más agresivo de compresión para el resto de imágenes, y tenemos la oportunidad de ello, pues el ojo humano no discierne con claridad ante objetos en movimiento. Este método se denomina estimación de movimiento.

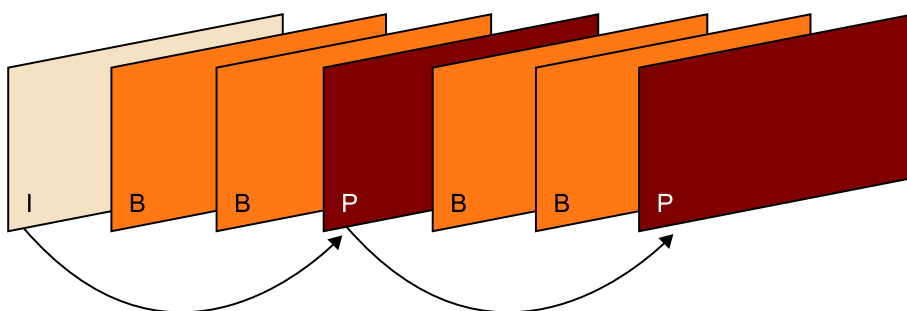
## Imágenes P y la estimación de movimiento

Teniendo los “pilares” I de una secuencia de imagen, las imágenes que existen entre dos imágenes I pueden codificarse mediante estimación de movimiento de dos maneras: imágenes P y B.

Las **imágenes P** (de **predictivas**) se calculan a partir de la imagen I o la imagen P inmediatamente anterior (no significa que sea la anterior inmediata, sino la última I o P que se haya codificado).

Son también de gran calidad, aunque menor que la I, ya que se calculan mediante una estimación del movimiento de la imagen (se divide la imagen en macrobloques de  $16 \times 16$  píxeles y se intenta ver en cuánto se han movido con respecto a la imagen que toman de referencia, sea I o P).

Ilustración 32



La intención es que no guardemos de esta imagen más que unos cuantos píxeles y muchos vectores de movimiento que hagan referencia a cuánto se han movido los bloques de píxeles de la imagen I o P de referencia. Este proceso proporciona un ahorro de bits impresionante. Aquí pues, el problema es calcular los vectores de movimiento, es decir, analizar las diferencias entre la imagen P y la de referencia (I o P), compararlas, ver si existen bloques de píxeles similares y detectar en cuánto se han desplazado en horizontal y/o vertical.

En secuencias estáticas (la cámara quieta y los objetos que componen la imagen en reposo), las imágenes P son sencillamente nulas. En secuencias en movimiento, las imágenes P aumentan en espacio de memoria ocupado al tener que almacenar el movimiento de los píxeles de la imagen de referencia, y si además aparecen objetos nuevos, han de añadirlos.

## Imágenes B

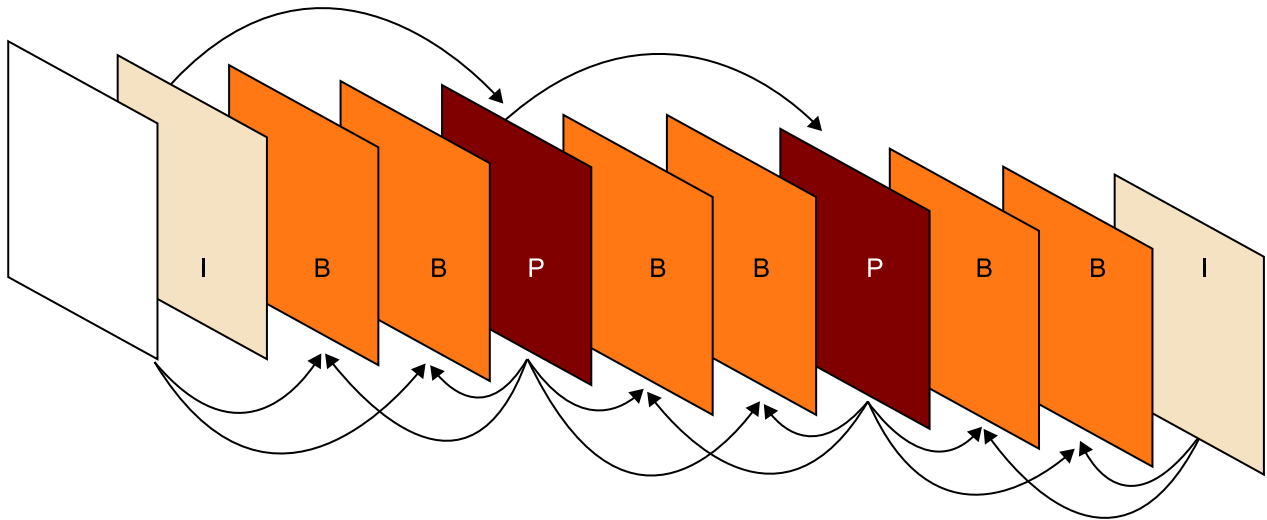
Las **imágenes B** son el caso más extremo de compresión.

La **imagen B** (de **bidireccional predictiva**) se deduce como intermedia a partir de dos imágenes de tipo I o P, con lo que contiene más información de estimación de movimiento a la vez que aumenta la compresión, aunque a una calidad relativamente baja.

Véase que si bien las P usaban estimación de movimiento en una sola dirección (desde la I o P anterior hasta ellas), las B usan ambas direcciones.



Ilustración 33



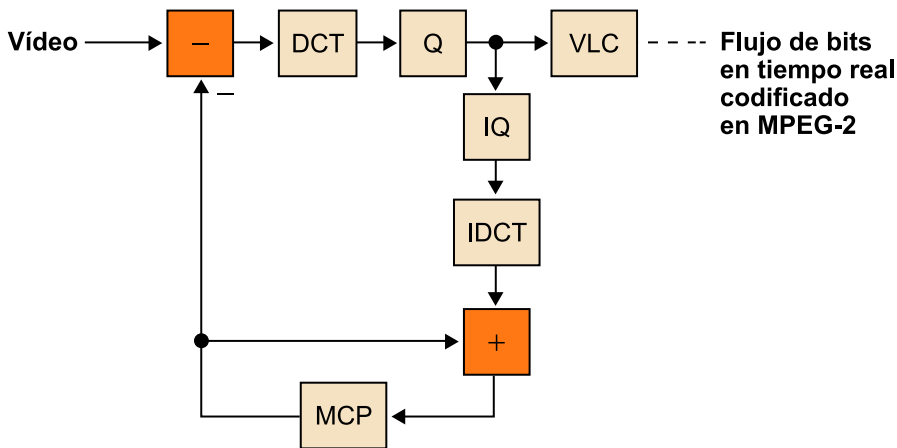
### 4.3.2. MPEG

A continuación, pasamos a explicar en qué consiste la codificación MPEG, que se basa en los conceptos que acabamos de explicar sobre el GOP.

#### Estructura del codificador MPEG-2

Un diagrama que englobe los mecanismos descritos puede ser el siguiente:

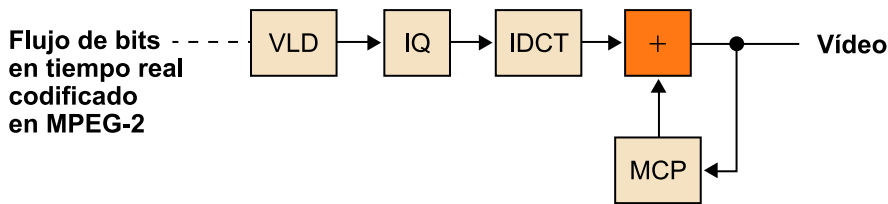
Ilustración 34. Esquema de codificador MPEG-2



Al vídeo en tiempo real se le resta, primeramente, la predicción que el sistema propone y que genera por el lazo inferior del esquema, por lo que la DCT codifica la imagen "error de predicción". Luego se cuantifica (Q) y se aplica codificación variable (VLC), además de añadir información adicional (vectores del movimiento, sincronización, etc.) para ayudar al decodificador en su tarea. Para efectuar la predicción, el flujo de datos "error" cuantificado pasa por un descuantificador (IQ) y una DCT inversa (IDCT), y se suma a la predicción de compensación de movimiento (MCP), que se autoalimenta para reducir tal error. Cuanto más óptima es el predictor de compensación de movimiento, la

resta inicial es mejor y el flujo de datos error generado que finalmente se transmite o almacena es más reducido, por lo que el canal por el que se transmite o el dispositivo de almacenamiento pueden ser de menores requerimientos.

Ilustración 35. Esquema de decodificador MPEG-2



El receptor decodifica la codificación variable (VLD), invierte la cuantificación (IQ) y, mediante una DCT inversa, recupera la imagen error, que se suma al resultado del predictor de compensación en movimiento (MCP) para dar la imagen final.

### El control de la memoria intermedia

Sin duda, la salida del codificador es un flujo de datos variable, dependiendo del contenido visual del vídeo y del éxito del predictor de compensación de movimiento, elemento clave de este sistema.

Sin embargo, en muchas ocasiones, este flujo de datos ha de ser transmitido o almacenado a una velocidad de bits fija, por lo que se añade una memoria intermedia entre el codificador (*coder*) y el canal (y otro entre el canal de recepción y el decodificador (*decoder*). Esta memoria intermedia es rellenada de forma variable por el codificador y alimenta el canal de transmisión a velocidad constante, por lo que hay que prevenir tanto que la memoria intermedia quede vacía como que se desborde. Para ello, la memoria intermedia notifica al codificador su estado y este ajusta la velocidad de transmisión de los bits de salida; por ejemplo, si la memoria intermedia se está llenando, el codificador aumenta la cuantificación,  $Q$ , para reducir el volumen de bits generados. Sin duda, ello afecta a la calidad del vídeo en recepción.

La inclusión de una memoria intermedia en emisión y otra en recepción genera además un retardo, calculable como la capacidad utilizada de la memoria intermedia dividida entre la velocidad del canal. Este retardo suele ser de algunas décimas de segundo.

MPEG-2 define el tamaño máximo que puede poseer esta memoria intermedia, aunque el codificador es libre de utilizar toda su capacidad, o solo una parte de esta.

## Propagación de errores y compresión: eterno dilema

Las imágenes I pueden contener errores, ya sean propios de la codificación intracuadro al “estilo JPEG” (una codificación con pérdidas), ya sean externos debidos a su transmisión (ruido, interferencias, etc.) o a procesos de grabación no adecuados (un soporte DVD en malas condiciones, un láser de lectura agotado, etc.). Ahora bien, si toda la ristra de imágenes P y B depende de ellas, es imaginable pensar que esos errores de base van a propagarse y afectarlas.

También las imágenes P pueden contener errores, ya sea heredados de la I o P que dependan, ya sea por un error en su transmisión o grabación.

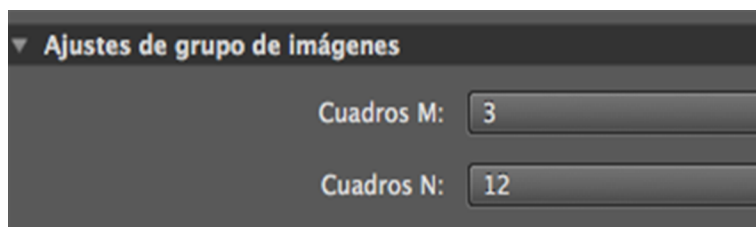
Las imágenes B, sencillamente, son las víctimas últimas de todos los errores que se hayan producido de toda clase.

El valor  $N$  es la distancia entre dos imágenes I. Repasando la ilustración 31, se deduce que el GOP debe poseer un tamaño de  $N$  imágenes, ya que comienza con una imagen I y acaba justo antes de la siguiente imagen I. Un valor de 12 es un equilibrio entre una gran distancia entre dos imágenes de referencia I y una compresión interesante con una propagación de errores aceptable.

Aumentar esta distancia supondría dejar en manos de imágenes P y B la calidad de la secuencia y arriesgarse a que pequeños errores de las imágenes I se amplificasen en todo el GOP. Reducirla sería mejorar la calidad, pero aumentando el peso de la secuencia y reduciendo la compresión.

El valor  $M$  es la distancia entre una imagen I y la siguiente imagen de tipo I o P que haya. En el ejemplo de antes (ilustración 31), es 3. Cuanto más grande es el valor de  $M$ , más errores contendrá la estimación de movimiento pero más alta será la compresión (eterno dilema). En una exportación de proyecto a MPEG-2, por ejemplo con un software como Premiere, AfterEffects, Avid, etc., deja al usuario su elección:

Ilustración 36. Ejemplo de selección de los valores  $N$  y  $M$  en una autoría DVD con MPEG-2



### 4.3.3. Ejemplos de uso

Algunos ejemplos de uso habituales de la codificación MPEG-2 son:

- DVD: al realizar una búsqueda de imagen con el mando a distancia en un DVD, el vídeo no fluye dinámicamente en pantalla, sino a saltos. Esto es debido a que va saltando de imagen I a imagen I.
- Al codificar una película en MPEG-2, cada cambio de plano fuerza el inicio de un GOP. Así, la primera I contiene los nuevos objetos.
- Vídeo *streaming* por Internet: un GOP de  $N = 12$  y  $M = 3$  es habitual.
- Edición de vídeo no lineal: cuantas más imágenes I haya, mejor. Incluso existe el estándar M-JPEG que está formado exclusivamente por imágenes I.

#### 4.4. Los flujos elementales de datos

Una vez generadas las imágenes I, P y B, a partir de ellas hay que construir un flujo final de bits para almacenar o transmitir, y este proceso tiene varias etapas. Comencemos por definir el flujo elemental ( $ES^{20}$ ), que es sencillamente la salida del codificador, que contiene toda la información necesaria para que un decodificador pueda crear una aproximación lo más certera posible al vídeo original.

<sup>(20)</sup>Del inglés, *elementary stream*.

Existen dos tipos diferentes de ES, los de audio ( $AES^{21}$ ) y los de vídeo ( $VES^{22}$ ); a continuación se describen cada uno de ellos.

<sup>(21)</sup>Del inglés, *audio elementary streams*.

##### 4.4.1. Flujo elemental de vídeo

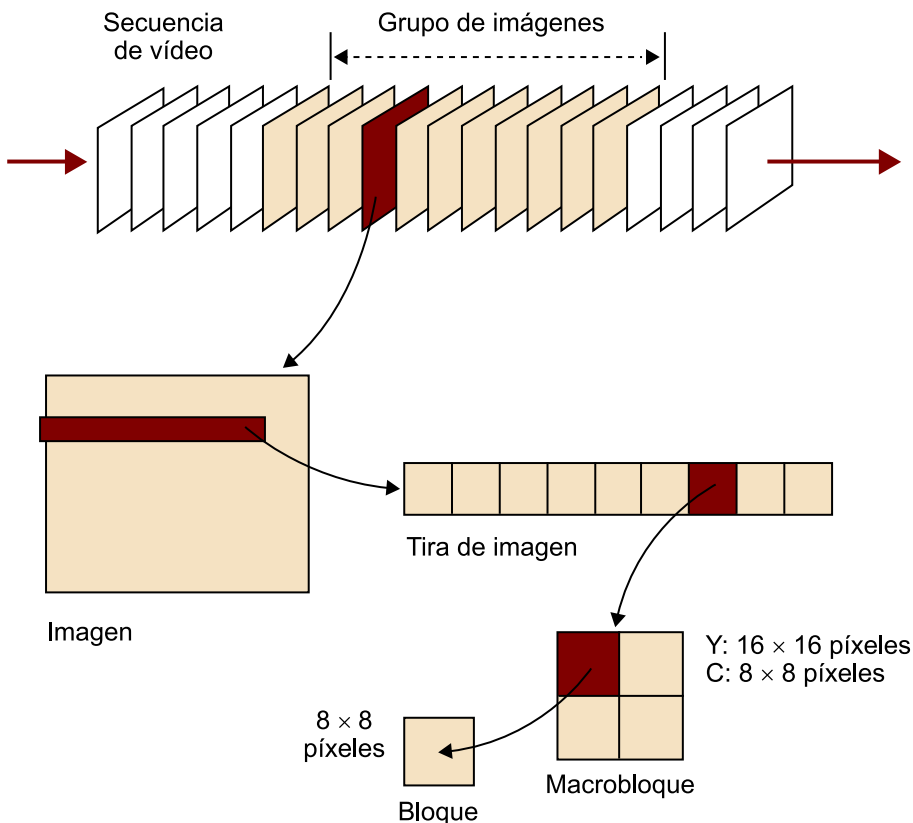
<sup>(22)</sup>Del inglés, *video elementary streams*.

Un **flujo elemental de vídeo (VES)** es una secuencia de vídeo. Habitualmente, la salida del codificador ya se encuentra en formato VES.

Como se ve en la ilustración 37, el flujo no contiene cada imagen por separado, sino que se estructura en grupos de imágenes (GOP), la unidad de trabajo de cualquier reproductor que quiera presentar este flujo (un reproductor<sup>23</sup> DVD, un programa de reproducción, etc.).

<sup>(23)</sup>En inglés, *player*.

Ilustración 37



## Construcción del VES

Cada imagen se convierte en tres matrices rectangulares de valores, cada una con los valores de  $Y$ ,  $Cb$  y  $Cr$ , respectivamente. En general, la matriz  $Y$  es cuatro veces más grande que las otras dos, pues, como ya hemos visto, es más relevante la luminancia ( $Y$ ) que los componentes de color ( $Cb$ ,  $Cr$ ).

En función del contenido y de la comparativa que realiza entre imágenes consecutivas, el codificador toma la decisión de definir una de las imágenes como de tipo I, es decir, codificable de manera independiente y pilar básico para la codificación de las inmediatas. Es habitual que, por ejemplo, la primera imagen tras un cambio de plano se defina como I, ya que no posee relación con las anteriores, pero el criterio es mucho más complejo y cada fabricante aplica el que cree más conveniente (el algoritmo no está definido en el estándar).

A partir de la imagen I, se calculan las imágenes P posteriores y las B tal como se ha explicado. Cabe destacar que, dentro de un GOP, siempre se mantiene la misma secuencia de imágenes, una vez definidos los parámetros  $M$  y  $N$  de la codificación.

### Ved también

Las imágenes I, P y B se explican en el subapartado 4.3.1 de este módulo didáctico.

Es importante recalcar que MPEG no especifica en sus estándares cómo debe ser un codificador, sino que especifica la salida del codificador para así asegurarse de la compatibilidad con el decodificador. De esta forma, codificadores

y decodificadores de distintos fabricantes pueden interoperar, además de permitir a la industria investigar y mejorar sus técnicas de codificación permaneciendo dentro del estándar.

El estándar, en cambio, sí especifica un mecanismo de decodificación (un decodificador), pero ese debe ser tomado como referencia, dejando también libre a la industria la posibilidad de cambiarlo siempre que logre el mismo resultado que el estándar (en este aspecto se piensa en mejoras de ejecución y adaptación del método a un hardware específico).

En MPEG-2 existen múltiples implementaciones de codificadores y decodificadores, algunas por hardware y muchas por software. La calidad de codificación MPEG-2 ha mejorado sensiblemente desde sus inicios y su calidad es alta incluso con flujos de menos de 2 Mbps. Es un estándar maduro, pero amenazado por el más actualizado MPEG-4, que lo supera en todos los aspectos, salvo el económico.

Habitualmente, las soluciones de IPTV utilizan equipamiento dedicado y costoso para realizar la codificación. La decodificación se realiza en el equipo del cliente, mayoritariamente en software (navegadores de Internet), pero existen algunas excepciones en hardware (TV por cable).

#### 4.4.2. Flujo elemental de audio

Un **flujo elemental de audio** (AES) es la salida del codificador de audio.

Este caso es muy diferente al de compresión vídeo; aquí no existen diferentes tipos de cuadros; los cuadros de audio se codifican todos de la misma manera, por lo que todos tienen el mismo tamaño.

#### 4.4.3. Flujo elemental empaquetado

A la hora de agrupar el *stream* de vídeo (VES) y el de audio (AES) en un solo flujo de datos, hay que tener cuidado de no generar retardos entre ambos (enviar demasiado vídeo y poco audio supondría no tener suficiente audio o tenerlo retrasado con respecto al vídeo). Hay que dividir los dos flujos de datos en paquetes de un tamaño adecuado antes de ser puestos secuencialmente uno tras otro.

Este proceso se denomina **empaquetado** y supone indicar en cada paquete su tipo y unos identificadores para que el receptor pueda reconstruirlos correctamente, así como datos adicionales para asegurar la sincronización (qué fragmento de audio va con qué fragmento de vídeo).

El flujo total es el flujo elemental empaquetado (PES).

## 4.5. Almacenamiento y transmisión

Generalmente, es necesario combinar varios PES (al menos un audio y un vídeo) para crear un contenido multimedia que, posteriormente, será reproducido. Dos posibilidades de tratamiento surgen entonces: almacenar el contenido para su posterior reproducción o transmitirlo (en nuestro caso, por una red IP).

### 4.5.1. Almacenamiento

El estándar de MPEG-2 no especifica ningún formato de contenedor de archivo. Esto también sucede con MPEG-1, H.263 y otros, por lo que pueden ser utilizados muchos contenedores: MP3 para audio, MOV para vídeo, etc.

### 4.5.2. Transmisión

El estándar del sistema de MPEG-2<sup>24</sup> define dos métodos de combinación de los datos de audio, vídeo y contenido asociado para la transmisión en un único flujo de datos:

(24) ISO/IEC 13818-1

- **Program Streams** (conocido como MPEG-2 PS, o simplemente MPEG-PS).
- **Transport Streams** (conocido como MPEG-2 TS, o simplemente MPEG-TS).

Un canal (como un canal de TDT) se encuentra formado al menos por un PES de vídeo y un PES de audio. El proceso de juntar varios PES que serán reproducidos en forma conjunta se llama multiplexado. Para lograr una reproducción conjunta sincronizada (sin desfase entre audio y vídeo), hay que enviar información de reloj en el multiplexado.

Mientras que MPEG-PS permite transportar un único canal, el MPEG-TS permite enviar varios simultáneamente (cada uno con una sincronización de reloj independiente). MPEG-TS incorpora, además, mecanismos de detección y corrección de fallos en la transmisión, tan habituales en redes públicas como Internet o 3G. Lo veremos al final del módulo por su relevancia.

#### Ved también

Los mecanismos de detección y corrección de fallos en la transmisión se estudian en el apartado 5 de este módulo didáctico.

Respecto a transmisión por red, se define un conjunto de protocolos para la transmisión de contenido multimedia por una red IP. Cabe destacar, entre ellos, para sistemas de vídeo digital de tiempo real, el protocolo *real-time protocol* (RTP).

#### Ved también

Los protocolos para la transmisión de contenido multimedia por una red IP se estudian en el subapartado 5.4 de este módulo didáctico.

## 4.6. Perfiles y niveles en MPEG-2

MPEG-2 fue diseñado para abarcar la mayor cantidad de sistemas de vídeo digital presentes en el momento de realización del estándar. Esto implica disponer de distintos grados de calidad para cada aplicación: generalmente determinado por la tasa de bits y la resolución de la codificación.

Los grados de calidad están definidos en el estándar en **perfiles** (*profiles*) y **niveles** (*levels*).

### 4.6.1. Perfiles

El perfil define la resolución del espacio de colores y la escalabilidad del flujo de bits. Las opciones más habituales son:

- **Perfil alto** (*high profile, HP*): destinado a codificar señales de alta definición. En realidad, nunca se utiliza ya que MPEG-4 es mucho más efectivo en este caso.
- **Perfil principal** (*main profile, MP*): el más habitual, dirigido a conseguir una calidad estándar de vídeo. Activa el uso de imágenes I, P y B en la codificación, y posibilita codificar señales de vídeo entrelazadas.
- **Perfil simple** (*simple profile, SP*): necesario cuando el flujo de datos generado va dirigido a equipos de bajas prestaciones técnicas –telefonía móvil, PDA, tabletas, etc.–. No utiliza imágenes de tipo B, lo que simplifica el cálculo de la decodificación y la necesidad de memorias intermedias.

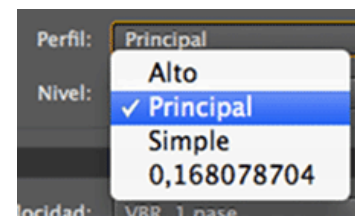


Ilustración 38. Perfiles de exportación (ejemplo: Adobe Premiere)

### 4.6.2. Niveles

El nivel define la resolución de imagen, la cantidad de muestras de luminancia (Y) por segundo, la cantidad de capas de audio y vídeo y el máximo *bit rate* por perfil:

- **Nivel alto** (*high level, HL*):  $1.920 \times 1.152$ , 80 Mbps. Para alta definición, nunca utilizado.
- **Nivel alto 1440** (*high-1440, H-14*):  $1.440 \times 1.152$ , 60 Mbps. También para alta definición, nunca utilizado.
- **Nivel principal** (*main level, ML*):  $720 \times 576$ , 15 Mbps. El habitual para imágenes de calidad estándar.



Ilustración 39. Niveles de exportación (ejemplo: Adobe Premiere)



- Nivel bajo (*low level, LL*): 288 × 352, 4 Mbps.

#### 4.6.3. Combinaciones de perfiles y niveles

No se utilizan todas las combinaciones de perfiles y niveles. La notación utilizada es perfil@nivel. Las más habituales son:

Perfil@nivel	Resolución	Hz	Muestreo	Bps	Ejemplo de aplicación
SP@LL	176 × 144	15	4:2:0	0,096	Tabletas, móviles
SP@ML	352 × 288	15	4:2:0	0,384	PDA
		320 × 240	24		
MP@LL	352 × 288	30	4:2:0	4	Decodificadores
MP@ML	720 × 480	30	4:2:0	15 (DVD: 9.8)	DVD, TDT, TV cab. y sat.
		720 × 576	25		
422P@ML	720 × 480	30	4:2:2	50	Sony IMX
		720 × 576	25		
422P@H-14	1.440 × 1.080	30	4:2:2	80	Reservado
		1.280 × 720	60		
422P@HL	1.920 × 1.080	30	4:2:2	300	Reservado
		1.280 × 720	60		

#### 4.7. MPEG-2 en la industria

MPEG-2 es muy utilizado en la actualidad, tanto en almacenamiento como en transmisión de audio y vídeo. Algunos de los sistemas que utilizan MPEG-2 son DVD y los estándares de TDT, satélite y cable en Europa (DVB) y América (ATSC). En cada caso, se implementa solo la porción necesaria del estándar.

Unos ejemplos reales meramente ilustrativos de MPEG-2 para almacenamiento:

##### a) DVD<sup>25</sup>

<sup>(25)</sup>De digital video disc.

- Contenedor de vídeo: MPEG-2 Program Stream
- Codificación de vídeo: MPEG-2

- Codificación de audio: PCM, MP2 (Musicam), Dolby Digital o DTS en Europa, y Dolby Digital o DTS en Estados Unidos
- Resolución (px):
  - NTSC: 720 × 480, 704 × 480, 352 × 480, 352 × 240
  - PAL: 720 × 576, 704 × 576, 352 × 576, 352 × 288
- Tasa de bits de audio + vídeo: Pico 15 Mbps, Mínimo 300 Kbit/s
- Relación de aspecto: 4:3, 16:9, 2.21:1
- Tasa de cuadros: 29.97 cuadros/s (NTSC), 25 cuadros/s (PAL)
- YUV 4:2:0

#### b) DVB<sup>26</sup>

<sup>(26)</sup>De digital video broadcast.

Incluye la televisión digital terrestre (DVB-T), satélite (DVB-S) y por cable (DVB-C). Resoluciones posibles:

- SDTV:
  - 720, 640, 544, 480 o 352 × 480 píxel, 24/1.001, 24, 30/1.001 o 30 cuadros/s
  - 720, 704, 544, 480 o 352 × 576 píxel, 25 cuadros/s
  - 352 × 240 píxel, 24/1.001, 24, 30/1.001 o 30 cuadros/s
  - 352 × 288 píxel, 25 cuadros/s

#### 4.8. Medida de la calidad de imagen en MPEG-2 para DVB

Al igual que se vio en imagen fija, existen métodos para analizar la calidad de vídeo comprimido en MPEG-2. En DVB hay dos grandes fuentes de problemas:

1) Las generadas por el ruido en el canal de transmisión. Mientras que en la TV analógica el ruido se plasmaba como una conocida nieve, en la TDT genera un aumento del número de errores de bit que hasta cierto punto son recuperados por los sistemas de protección de errores que posee y no son visibles ni audibles. Más allá de ese punto, sin embargo, pueden aparecer bloques de imagen incorrectos, efecto mosaico en la imagen y, en el caso extremo, el servicio cae completamente y la pantalla va a negro. Estos fenómenos son familiares en TDT (DVB-T) y satélite (DVB-S), pero poco comunes en cable (DVB-C) al ser un canal más controlado y fiable.

2) Los artefactos producidos por la propia codificación MPEG-2. Centrándonos en esta segunda fuente de problemas, la codificación MPEG-2 tiene un efecto directo en la calidad de imagen debido a su mayor o menor severidad en la compresión utilizada. Al dividirse la imagen en macrobloques de 16×16 píxeles de manera independiente a sus bloques circundantes, cuando se aplica una alta compresión aparecen discontinuidades de luminancia y crominancia entre bloques adyacentes, perceptibles por el ojo y denominado *blocking*. Además, esta unidad básica de macrobloque es la base de la codificación predictiva entre cuadros I, siendo la base de generación de los cuadros P, por lo que el problema es intrínseco al estándar.

Sin duda, el fenómeno no solo depende del nivel de compresión, sino también del contenido visual de la imagen. Una situación clásicamente compleja son los deportes: la Fórmula I es mucho más difícil de comprimir sin artefactos que una partida de ajedrez debido a su movimiento rápido y detalle.

Existe un sistema estandarizado de medición subjetiva de la calidad, llamado ITU-R BT.500, que describe los métodos de análisis de la calidad del vídeo a partir de los resultados dados por un grupo de personas seleccionadas como observadores.

Hay dos métodos subjetivos principales:






- La escala de calidad continua de doble estímulo (DSCQS): el grupo de personas compara un clip de vídeo original con su versión codificada, y la valoran en un rango de 0 (malo) a 100 (excelente).
- La escala de calidad continua de estímulo único (SSCQE): el grupo de personas valora el clip codificado, de 0 a 100, sin ver el clip original.

No existen métodos estándares para el análisis objetivo de la calidad MPEG-2. Cada fabricante de equipos de instrumentación y medida han implementado los suyos propios, pero en general se centran en la comparación de los píxeles que bordean cada macrobloque, responsables finales del efecto de *blocking*. No realizan una comparativa directa (como podría realizarse con una imagen estática), sino que toman en cuenta lo que denominan actividad espacial (riqueza de altas frecuencias espaciales) y actividad temporal (movimiento), pues ambos podrían falsear los resultados de las medidas.



Ilustración 40. Equipo Rohde&Schwarz analizador de calidad de vídeo digital

Ilustración 41. Resultados del análisis de calidad de cinco clips de vídeo con el analizador DekTec DTC-352

Video Quality Analysis								
Thumbnail	Filename	File Type	File Summary	Size / Duration	Average Quality			Status
		VIDEO			BAD	40	60	
	Demo.mpeg	VIDEO	MPEG-2 video 720 x 576 9000000 B/s 25 F/s	Filesize : 32 MB Duration: 30 sec	84			analysed
	cities.h264	VIDEO	H.264 / AVC 704 x 576 25 F/s	Filesize : 12 MB	32			analysed
	KMBC_TV.mpeg	VIDEO	MPEG-2 video 720 x 480 3250000 B/s 29.97 F/s	Filesize : 55 MB Duration: 1 min	12			analysed
	ARD.ts	VIDEO	MPEG-2 video 544 x 576 7000000 B/s 25 F/s	Filesize : 83 MB Duration: 52 sec	83			analysed
	KTVI_TV.mpeg	VIDEO	MPEG-2 video 720 x 480 3250000 B/s 29.97 F/s	Filesize : 55 MB Duration: 1 min	9			analysed

Ya que el estándar MPEG-2 no define el procedimiento de codificación, en los últimos años ha habido sustanciales mejoras de diversos fabricantes que han puesto la calidad de MPEG-2 muy por encima de los primeros codificadores existentes, también en un afán de no perder terreno ante el auge de su sucesor, MPEG-4, como se ve en detalle en el siguiente apartado.

## 5. El estándar MPEG-4 y H.264/AVC

Tras MPEG-2, finalizado en 1995, el grupo MPEG continúa su trabajo de estandarización. En 1993 (antes de terminado MPEG-2), comienza el trabajo en el estándar MPEG-4, que abarca muchos más aspectos de los sistemas de vídeo digital que sus predecesores MPEG-1 y MPEG-2.

La parte 2 del estándar MPEG-4 (ISO/IEC 14496-2), conocida como MPEG-4 Visual, especifica la codificación y decodificación de vídeo digital. MPEG-4 Visual se terminó de estandarizar en 1999. El estándar H.264 se inició por parte del grupo de trabajo Video Coding Experts Group (VCEG<sup>27</sup>) de la International Telecommunication Union (ITU-T). Las últimas etapas del trabajo se realizaron por la Joint Video Team (JVT), un grupo conformado por los grupos VCEG y el MPEG.

<sup>(27)</sup>VCEG es la sigla de *Video Coding Experts Group*.

El estándar final fue publicado en conjunto en el 2003. Debe entenderse que MPEG-4 y H.264 no son sinónimos: H.264 es una parte del estándar MPEG-4. H.264 también es llamado H.264 Advanced Video Coding (AVC). Por tanto, las formas correctas de llamar a este estándar son: **MPEG-4 Part 10** o **H.264/AVC**.



MPEG-4 Visual y H.264, aunque contemporáneos, son muy diferentes en su concepción: MPEG-4 Visual apunta a la flexibilidad abarcando la mayor cantidad de sistemas de vídeo digitales posibles, mientras que H.264 apunta a la eficacia en la compresión y la confiabilidad de la transmisión.

### La publicación del estándar en el 2003

El estándar final fue publicado por MPEG como la parte 10 del estándar MPEG-4 (ISO/IEC 14496-10), y por ITU-T, como H.264.

La codificación utilizada en la mayoría de los sistemas de IPTV es H.264, por eso dedicaremos este apartado exclusivamente a esta parte del estándar MPEG-4. Para su mejor comprensión, la estructura de este apartado se presenta de modo muy similar a la de MPEG-2, comenzando por el detalle de los estándares que la componen, siguiendo por algunos detalles del formato, los mecanismos de transmisión y almacenamiento, los perfiles y niveles definidos, y culminando con la aplicación actual de esta codificación en la industria.

El contenedor de este códec está bien especificado, basado en un formato estándar que curiosamente se inspiró en el de QuickTime de Apple: el ISO Media File Format. La extensión de los archivos MPEG-4 es .mp4.

## 5.1. El estándar y sus patentes

MPEG-4 se encuentra estandarizado en la norma ISO/IEC 14496. Al igual que para MPEG-2, se compone de varias partes, entre ellas:

- El sistema: describe la sincronización y la transmisión simultánea de audio y vídeo.
- El vídeo: el códec de compresión para elementos visuales (vídeo, texturas, imágenes sintéticas, etc.). Uno de los muchos perfiles definidos en la parte 2 es el Advanced Simple Profile (ASP).
- El audio: el conjunto de códecs de compresión para la codificación de flujos de audio; incluye variantes de Advanced Audio Coding (AAC), así como herramientas de codificación de audio y habla.
- Transporte sobre redes IP: especifica un método para transportar contenido MPEG-4 sobre redes IP.
- H.264 - Advanced Video Coding (AVC): un códec estandarizado de señales de vídeo.
- Ingeniería de aplicación y descripción de escenas (BIFS). Para contenido interactivo 2D y 3D.
- Formato para medios audiovisuales basado en ISO: un formato de archivos para almacenar contenido multimedia.
- Extensiones para el manejo y protección de propiedad intelectual (IPMP).
- El formato de archivo contenedor designado para contenidos MPEG-4.
- El formato de archivo AVC para el almacenamiento de vídeo.

Es importante resaltar la parte 10 del estándar que especifica el H.264 y la parte 8 que especifica el transporte de MPEG-4 sobre una red IP.

## 5.2. H.264 o MPEG-4 Part10

En su filosofía, la codificación H.264 no difiere de MPEG-2 en gran forma. Se especifica el formato de codificación y cómo decodificarlo (y se deja nuevamente libre a la industria el diseño de codificadores eficientes).

También utiliza la nomenclatura de perfiles y niveles para definir variantes de resolución y calidad de vídeo. Hoy en día, relativamente pocos perfiles se encuentran definidos por completo en el estándar. En sus comienzos, solo

tres perfiles fueron definidos: *baseline*, *main* y *extended*. Sin embargo, en la actualidad parece perfilarse con mayor aceptación en el mercado el perfil *high*, como ya veremos.

Una mejora con respecto al estándar MPEG-2 es que se separa la codificación de la transferencia, lo que facilita su implementación.

El formato de salida del codificador se llama Video Coding Layer (VCL), que básicamente es una secuencia de bits que representa vídeo codificado. La jerarquía de codificación de vídeo MPEG-2 (GOP, cuadros, bloques) permanece prácticamente inalterada.

Un problema habitual en MPEG-2 es que, si hay diferentes subflujos o servicios contenidos en el mismo flujo de bits, al conmutar de uno a otro, el decodificador perdía calidad unos instantes. Contra eso, en los perfiles más altos de H.264 aparece la posibilidad de utilizar dos tipos extra de cuadros (además de los I, P, B):

- **SP (*switching P*)** para facilitar cambiar de flujo de codificación (contiene macrobloques I o P).
- **SI (*switching I*)** para facilitar cambiar de flujo de codificación (contiene macrobloques SI).

### 5.3. Almacenamiento

El estándar MPEG-4 realiza una especificación de almacenamiento en disco, como se dijo al principio de este apartado. Se basa en el estándar ISO Media File Format, el cual se encuentra inspirado en el formato QuickTime de Apple. La extensión de los archivos MPEG-4 es .mp4.

Muchos otros formatos contenedores poseen esta codificación, como por ejemplo 3GPP, MOV o Flash Video.

### 5.4. Transmisión

Tanto MPEG-4 como H.264 no definen un sistema específico de transporte para la transmisión.

Con la enmienda 3 del sistema MPEG-2 (ISO/IEC 13818-1), es posible transportar MPEG-4 y, en particular, H.264, de la siguiente manera:

- **Program Streams** (conocido como MPEG-2 PS, o simplemente, MPEG-PS).

- **Transport Streams** (conocido como MPEG-2 TS, o simplemente, MPEG-TS), que permite hacer convivir canales TDT en H.264/AVC en HD con los MPEG-2 en SD.

Respecto a la transmisión por red IP, se ha actualizado el protocolo de tiempo real RTP (Real-Time Protocol) para la transmisión de MPEG-4 y H.264. Es relativamente sencillo transmitir contenido multimedia en una red de paquetes IP; incluso limitándonos al contexto de IPTV, no solo RTP es utilizado. En las redes actuales, para transmisión de vídeo digital por Internet, existe una clara preferencia por protocolos basados en TCP, y, en particular, aquellos basados en HTTP, debido a su compatibilidad con las configuraciones frecuentes de cortafuegos<sup>28</sup> y servidores intermediarios<sup>29</sup>.

<sup>(28)</sup>En inglés, *firewalls*.

<sup>(29)</sup>En inglés, *proxies*.

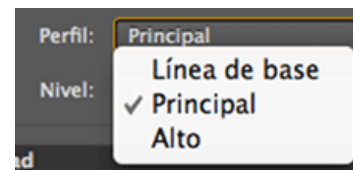
## 5.5. Perfiles y niveles

Al igual que MPEG-2, el estándar H.264/AVC presenta diferentes grados de calidad. Los grados de calidad están definidos en el estándar en términos de perfiles y niveles.

### 5.5.1. Perfiles

Existen siete perfiles en el estándar, de los que nos centraremos en los tres habituales:

- **Línea de base** (*baseline profile, BP*): usado para dispositivos sencillos como reproductores portátiles y telefonía móvil, así como para *streaming* de vídeo por red IP.
- **Principal** (*main profile, MP*): dirigido a *broadcast* (TDT HD) y almacenamiento (Blu-Ray), provee alta definición.
- **Alto** (*high profile, HP*): para aplicaciones profesionales que requieren alta definición de crominancia y luminancia (4:4:4).





### 5.5.2. Niveles

Representados con un número del 1,0 al 5,1, son configuraciones para velocidades de transmisión crecientes (desde 64 Kbps en 1,0 hasta 240 Mbps en 5,1).

### 5.6. H.264 en la industria

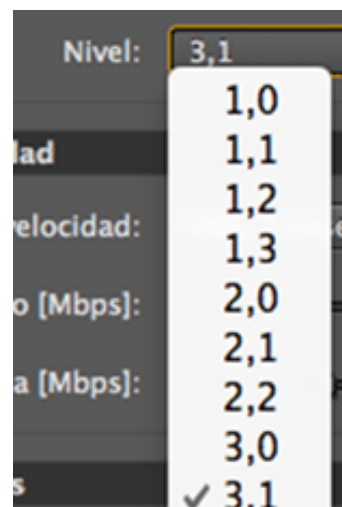
H.264 es siempre el elegido en sistemas de vídeo digital de alta resolución, como las redes de IPTV y la TV en HD por terrestre, cable o satélite. Es de esperar que, con el paso del tiempo, los distintos sistemas de transmisión de vídeo migren a este formato reemplazando los tradicionales H.262/MPEG-2, tal como ha ocurrido en Francia, que ha apostado desde el inicio de sus emisiones TDT por este formato.

Algunos de los sistemas que utilizan o plantean utilizar H.264 son:

#### a) Para almacenamiento:

Blu-Ray Disc del Blu-Ray Disc Association (BDA). Utiliza un perfil alto (HP). Como características más conocidas de Blu-Ray, tenemos:

- Contenedor de vídeo: BDAV MPEG-2 Transport Stream (ficheros de extensión .bdav), con lo que la grabación de una emisión de TDT en HD puede almacenarse directamente sin alterar el formato en absoluto.
- Codificación de vídeo: H.264/MPEG-4 AVC (o VC-1 de Microsoft)
- Codificación de audio: Dolby Digital, DTS, Dolby Digital Plus, DTS-HD, PCM
- Combinaciones de resolución y tasa de cuadros en modos progresivos:
  - **1080p**: 1.920 × 1.080 píxeles a 24/23,97 cuadros/s (Estados Unidos/Europa)
  - **720p**: 1.280 × 720 píxeles a 50/59,94 cuadros/s (Estados Unidos/Europa)
  - Y otras combinaciones progresivas (1.440 × 1.080, 1.280 × 720) a 24 c/s (modo cine)
- Combinaciones de resolución y tasa de cuadros en modos entrelazados:
  - **1080i**: 1.920 × 1.080 píxeles a 25/29,97 cuadros/s (Estados Unidos/Europa), o 1.440 1.080 píxeles a 25/29,97 cuadros/s (Estados Unidos/Europa)



#### El HD-DVD de Toshiba

El ya desaparecido formato de almacenamiento HD-DVD de Toshiba poseía características similares.

- **720i:** 1.280 × 720 píxeles a 25/29,97 cuadros/s (Estados Unidos/Europa)
- Relación de aspecto 16:9 (aunque hay dos modos 4:3 poco utilizados)
- Tasa de bits de audio + vídeo: Pico 36 Mbps
- YUV 4:2:0

#### b) Para *broadcast*:

- El consorcio europeo DVB, desde el año 2004, apostó por aportar (si no reemplazar) a la codificación H.262/MPEG-2 actual este nuevo estándar, tanto en terrestre (DVB-T) como satélite (DVB-S) y cable (DVB-C). El nuevo estándar DVB-T2, que se implantará a partir del 2015 en Europa, optimizará el aprovechamiento de las frecuencias de canal para dar cabida a más canales, cosa que podría facilitar la migración a H.264.
- El comité estadounidense ATSC especifica H.264 y VC-1 como estándares para la transmisión terrestre de televisión. Igualmente, la asociación ARIB japonesa incorpora H.264/AVC en su sistema de TDT (ISDB-T).
- En satélite, está presente en la mayoría de plataformas (Digital+, BBC HD, Euro1080, etc.).

#### c) Para red de telefonía celular móvil:

- The 3rd Generation Partnership Project (3GPP) incluye H.264/AVC como opcional. Actualmente (2011), es infrecuente poder recibir este servicio debido al ancho de banda necesario.

### 5.7. Gestión de objetos multimedia con MPEG-4

Hasta lo visto, MPEG-4 puede considerarse una versión actualizada de MPEG-2, pero es más que un sistema de codificación audiovisual: provee de herramientas para representar “objetos *media*”, es decir, unidades visuales o aurales que pueden ser sintéticas o capturadas de la realidad.

MPEG-4 posibilita describir tales **objetos multimedia** y crear “composiciones” con ellos, es decir, escenas, e incluso provee de estándares para la “interacción” del usuario con estas escenas audiovisuales.

MPEG-4 posibilita producir contenidos a los autores, potenciando la reutilización en mayor medida que la que se aplica actualmente en TV digital, web, etc. Para los proveedores de servicios de red, ofrece información transparente que puede ayudar a la mejora del transporte extremo a extremo por redes heterogéneas. Finalmente, para los usuarios finales, ofrece interacción y multimedia incluso a velocidades de transmisión de bits bajas, como las comunicaciones móviles.

No es la faceta más conocida del estándar, pero sin duda la que aún posee mayor camino de desarrollo y evolución.

