

Compresión de señales digitales

Fases de la compresión

Señal original → compresión → señal comprimida

Transformación a una representación invertible más adecuada (normalmente, una transformada integral). En el caso de video, transformación del movimiento (cambios entre frames) a una representación más compacta.

→

Alteración de la representación en el espacio transformado, normalmente por cuantización no uniforme, para obtener valores repetidos próximos. Esta alteración es el único paso que puede ser **no invertible**, y por tanto provocar pérdidas.

→

Codificación de los valores obtenidos, mediante Huffman, códigos *run-length* o alguna otra codificación no uniforme.

Señal original alterada ← descompresión ← señal comprimida

Tecnologías Multimedia en Web

Compresión de señales digitales (x)

Formatos de audio comprimido

Compresión de audio sin pérdidas: [FLAC](#)

FLAC (free lossless audio compression) es un formato libre bajo el patrocinio de la [fundación Xiph](#). Su funcionamiento se basa en:

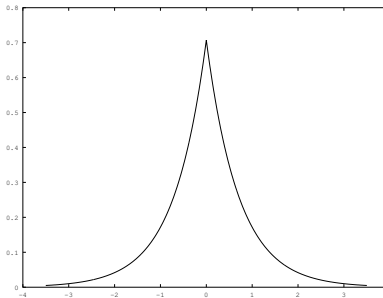
- Dividir la señal en bloques que se comprimirán independientemente. Por defecto son de 4096 muestras consecutivas.
- Decorrelar las señales de los canales izquierdo y derecho construyendo con ellas otras dos señales, $mid = \frac{left+right}{2}$ y $side = left - right$.
- Buscar cuál es el predictor autoregresivo óptimo para la señal. Un predictor autoregresivo es una forma de obtener el valor de la señal en un instante del tiempo t a partir de valores anteriores:

$$\hat{s}(t) = \sum_{i=1}^p a_i s(t-i)$$

con coeficientes $a(i)$ cuantizados a 7 bits.

Formatos de audio comprimido

- Este predictor no es perfecto, y el error es $e(t) = s(t) - \hat{s}(t)$. Experimentalmente se ve que estos errores siguen una distribución Laplaciana, $p(x) = \frac{1}{\sqrt{2}\sigma} e^{-\frac{\sqrt{2}}{\sigma}|x|}$.



- El paso final es codificar (p. ej. por Huffmann) los coeficientes $a(i)$ y los errores de predicción para cada bloque de la señal.

En la versión completa del algoritmo se estiman los coeficientes en cada instante; en una versión más sencilla se escogen de entre cuatro conjuntos predeterminados los que den menos error.

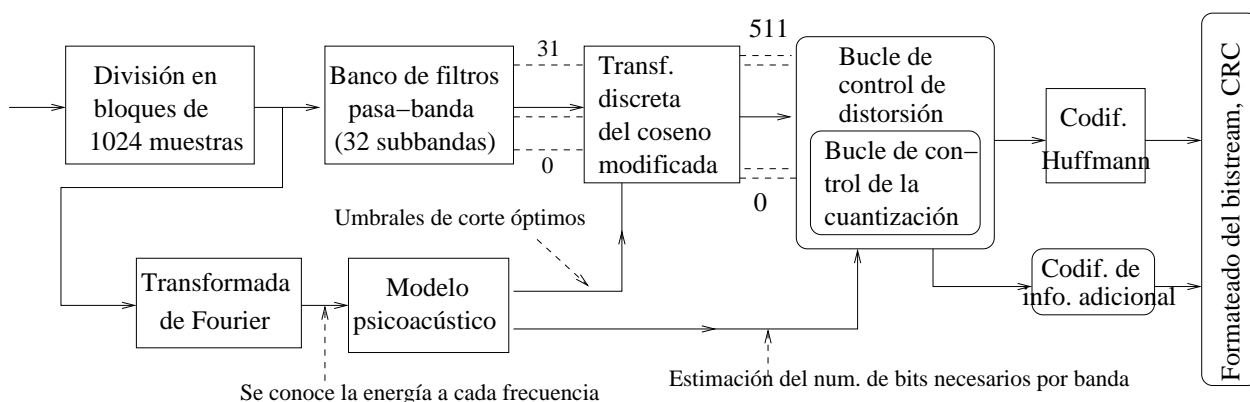
Formatos de audio comprimido

Compresión de audio con pérdidas-I: MP3

MP3 (MPEG-1/2 layer 3, no confundir con MPEG3) es el compresor de audio diseñado por el grupo de expertos MPEG (Moving Picture Experts Group) como canal de audio para los compresores de video MPEG-1 y MPEG-2. MP3 está sometido a derechos de patente, por lo que su uso no es aconsejable.

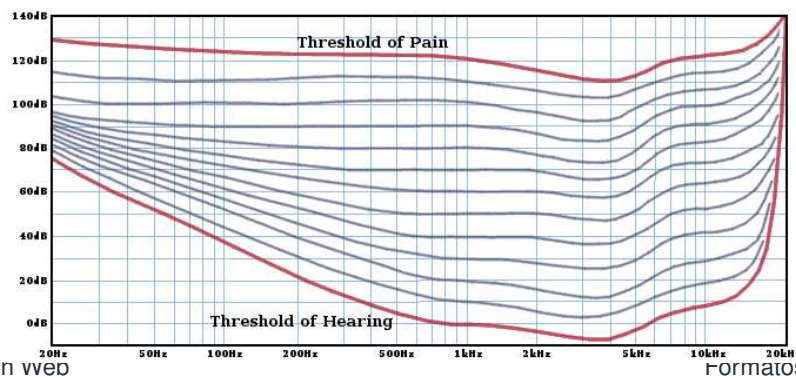
Las frecuencias de muestreo estándar que acepta son 16, 22.05, 24, 32, 44.1 y 48 KHz.

El rango de *bitrates* (tasas binarias a las que el sonido será comprimido) oscila entre los 32 y los 320 Kbits/seg, pudiendo variar a lo largo de la duración de la señal.



Formatos de audio comprimido

El primer banco de filtros está constituido por 32 filtros pasa-banda que barren todo el rango de frecuencias audibles, con un cierto solapamiento entre cada par de filtros adyacentes. A continuación la transformada discreta del coseno modificada (MDCT) subdivide más aún cada rango de frecuencias en otros 18, y umbraliza los coeficientes resultantes con umbrales que dependen del umbral de percepción del oído humano para cada frecuencia (ver figura) y también de cuál sea el valor de otras componentes frecuenciales en instantes próximos (separados no más de 5 ms. Nótese que, a $f_s = 44100\text{Hz}$, 1024 muestras son 23ms.). Esto es para tener en cuenta el hecho de que de dos sonidos simultáneos de distintas frecuencias, uno mucho más fuerte que el otro, sólo oiremos el más fuerte.



Tecnologías Multimedia en web

Formatos de audio comprimido (iv)

Formatos de audio comprimido

Compresión de audio con pérdidas-II: AAC

AAC (Advanced Audio Codec) es otro compresor de audio diseñado por el grupo de expertos MPEG (Moving Picture Experts Group) como canal de audio para el compresor de video MPEG-4. Fue adoptado también por Apple para sus productos iPod y el software iTunes. Se usan especialmente dos perfiles: AAC-LC (*low complexity*) y AAC-Plus.

Las diferencias fundamentales con el MP3 son:

- Mayor resolución en la frecuencia; en lugar de los 576 (32×18) canales del MP3 pueden usarse hasta 1024.
- En el profile Plus, incorporación de un predictor similar al ya mencionado en el FLAC.
- Mejora de la representación de las señales estereofónicas.
- Codificación de Huffmann mejorada, con códigos específicos mejor adaptados a la distribución de los errores de codificación.

Compresión de audio con pérdidas-III: [Vorbis](#)

Vorbis (free loopy audio compression) es un formato libre bajo el patrocinio de la [fundación Xiph](#). Su funcionamiento se basa en el uso de la transformada discreta del coseno modificada (MDCT) y en un modelo psicoacústico complejo para el compresor, pero que permite una descompresión bastante simple. En este sentido es una alternativa viable a MP3, con menor coste computacional para la descompresión (pero empleo de más memoria).

Como en MP3, la entrada es dividida en bloques y pasada por un banco de filtros, y luego descompuesta con una MDCT. La diferencia fundamental con MP3 es que las tablas de códigos completas usadas para la decodificación se transmiten íntegras en la cabecera de la stream de audio, y son específicas para varios bloques (hasta que las estadísticas de la señal de entrada cambien significativamente).

El documento completo de especificación está [disponible](#) y la implementación de referencia es la [libvorbis](#). Vorbis es el formato de audio escogido para el contenedor de audio/video WebM, propuesto por Google.

Formatos de imagen comprimida

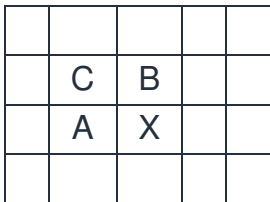
Compresión de imagen sin pérdidas: **compresión predictiva**

De acuerdo al mecanismo de asignación de símbolos a valores visto antes (codificación de Huffman), cuanto mayor sea el desequilibrio en el número de apariciones de símbolos entre unos valores y otros (o sea, cuantos más valores en el histograma sean 0) mayor será la compresión. Ello hace que sean más compresibles imágenes con sus histogramas fuertemente picados, y que no sea fácilmente compresible una imagen con su histograma plano (valores uniformemente repartidos por todo el rango). De hecho, la imagen más típica con histograma plano (el ruido blanco) es totalmente incompresible por ningún método.

Formatos de imagen comprimida

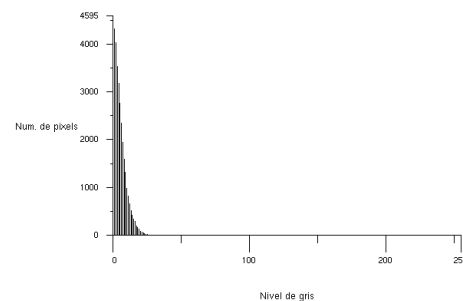
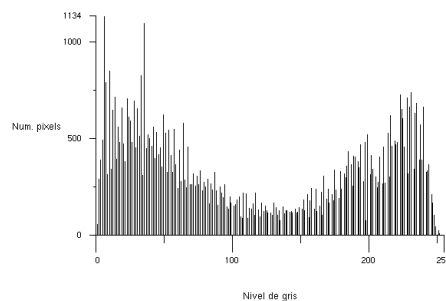
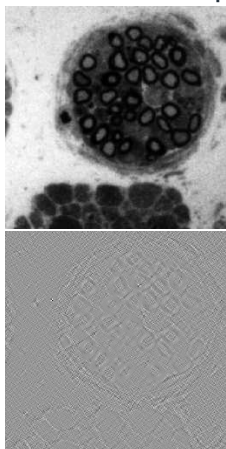
Para la compresión predictiva se supondrá que se da un orden prefijado para el recorrido de la imagen, de tal modo que se podrá hablar de pixels anteriores y posteriores en la ordenación. Generalmente se supone la ordenación usual de lectura occidental.

Hecho esto, se intentará predecir el valor de un pixel a partir de los anteriores usando alguna fórmula simple dada, y lo que se guardará no es el valor del pixel, sino la diferencia entre éste y su predicción. Si la predicción es aceptable esta diferencia será pequeña, y el pequeño número de valores que toma hará que se repita muy frecuentemente. Como ejemplo, se puede predecir el valor de X a partir del de sus vecinos A,B y C según alguna de las fórmulas 0 a 7 de la tabla dada:

	Núm. de fórm.	Fórmula	Núm. de fórm.	Fórmula
	0	Sin predicción	4	$A+B-C$
	1	A	5	$A+((B-C)/2)$
	2	B	6	$B+((A-C)/2)$
	3	C	7	$(A+B)/2$

Formatos de imagen comprimida

Como ejemplo de codificación predictiva se muestra una imagen, su histograma, la imagen que se almacena usando la fórmula número 4 y su histograma. Esta última ha sido normalizada por propósitos de visualización.



La entropía de la imagen original calculada como $H = - \sum_{i=1}^{256} p_i \log(p_i)$ resulta ser de 6.884 bits, y la de la diferencia con la predicción de 5.163. Como se ve, el histograma se estrecha. Es por tanto especialmente apropiado para aplicar ahora una codificación de Huffmann. Con ambos pasos se consiguen tasas de compresión en torno al 50 %.

Formatos de imagen comprimida

Compresión de imagen sin pérdidas: el [estándar PNG](#)

PNG es un formato de imagen extensible para la compresión sin pérdidas de imágenes raster. PNG proporciona una alternativa sin derechos de patente al formato GIF y puede reemplazar muchos usos del TIFF. Admite imágenes en niveles de gris, color indexado y color real más un canal alfa opcional para transparencia. El número de bits por pixel puede ir de 1 a 16 por componente (hasta 64 bits para RGBA).

A diferencia de otros formatos, la cabecera guarda las coordenadas de color del blanco de referencia y la γ del monitor en el cual la imagen se ve como el autor espera. Ello permite la corrección automática para mostrarla en otros monitores.

Formatos de imagen comprimida

La compresión se realiza en dos pasos:

1) Una codificación predictiva usando uno de los métodos 0 a 4:

				Método	Nombre	Valor predicho
				0	None	0 (sin predicción)
	C	B		1	Sub	A
	A	X		2	Up	B
				3	Average	$\text{floor}((A+B)/2)$
				4	Paeth	A, B ó C, el más cercano a $(A+B-C)$

2) Compresión de los errores de predicción usando el método DEFLATE, que consiste en el uso del algoritmo LZ77 o algoritmo Lempel-Ziv LZ1 usado por compresores como ZIP ó gzip, seguido de una codificación de Huffman.

Una variante del PNG permite un recorrido en forma de [entrelazado](#) de siete pasadas que reduce la tasa de compresión, pero permite la transferencia de una versión de baja resolución de la imagen primero, que se va refinando a medida que ésta llega.

Existe una versión para gráficos animados (MNG) que pocos navegadores soportan de modo nativo.

Formatos de imagen comprimida

Compresión de imagen con pérdidas-I: **el estándar JPEG** (Joint **P**hotographic **E**xperts **G**roup)

- Adoptado en 1993 a propuesta del **comité JPEG** del ISO.
- Pretende ser genérico (no orientado a una clase particular de imágenes).
- Existen dos subestándares, uno basado en codificación predictiva sin pérdidas, y otro mucho más usado basado en la DCT, con pérdidas.

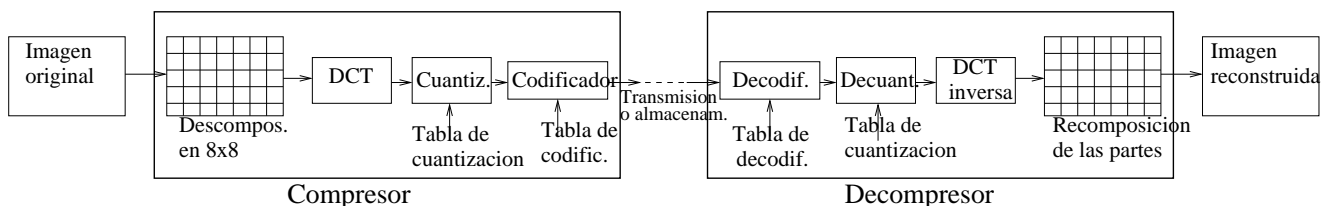
El compresor consta de tres pasos:

- La división de la imagen en bloques de 8×8 pixels, y la ejecución de la DCT sobre cada uno.
- La cuantización de los valores resultantes de la DCT a un número menor de niveles.
- La codificación de los valores cuantizados para convertirlos en cadenas de bits de la menor longitud posible.

El decompresor opera invirtiendo cada paso, en orden también inverso.

Formatos de imagen comprimida

El esquema del compresor/decompresor (llamado a veces codificador/decodificador o CODEC) es:



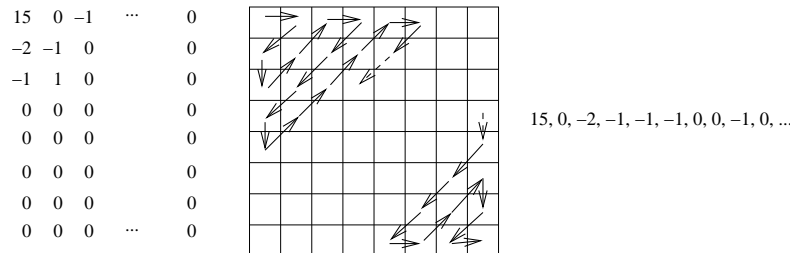
Después del primer paso, cada bloque de 8×8 ha sido sustituido por su DCT. Obsérvese que las funciones trigonométricas que aparecen en la DCT sólo se calculan en un número reducido de valores ($\frac{2(n+1)\pi u}{16}$ con $n = 0 \dots 8$ y $u = 0 \dots 8$) que se pueden tabular.

Al final de este paso cada subimagen del plano DCT suele tener valores altos cerca de la esquina superior derecha y bajos en el resto. Ello es porque las imágenes usuales suelen contener pocas frecuencias espaciales altas. Un ejemplo típico podría ser:

100.0	80.3	-16.4	3	0	0	0	0
78.0	60.0	10.0	0	0	0	0	0
27.0	12.0	5.0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	1.7	1	2.2
0	0	0	0	0	0	0	0

Formatos de imagen comprimida

En el siguiente paso (cuantización) cada elemento de la tabla se divide por un coeficiente específico de su posición en la subimagen 8×8 (el mismo para todos los bloques de la imagen). Estos coeficientes (de 1 a 255) se han determinado experimentalmente y tienen que ver con la mínima discriminación en frecuencia espacial que el ojo es capaz de percibir a esa cierta frecuencia espacial. Esto genera una tabla con casi todos sus elementos nulos, que se recorre en zig-zag, como se muestra:



La lista de 64 elementos resultante se codifica como un alfabeto de símbolos de la forma:

- El primero (el término de continua, DC) como (tamaño)(diferencia con el DC anterior), excepto para el primer DC, que es (tamaño)(valor)
- Los demás como (long. de la cadena de ceros, tamaño)(valor)

Formatos de imagen comprimida

Como ejemplo, supongamos que el término DC anterior hubiera sido 12. Entonces la cadena anterior queda: (*) (3) (1,*) (-2) (0,*) (-1) (0,*) (-1) (0,*) (-1) (2,*) (-1) (0,0) donde la * será el tamaño en bits asignado al símbolo por un código de Huffman, p. ej.:

(2)	011	(0,0)	1010
(3)	11	(0,1)	00
(-2)	01	(1,2)	11011
(-1)	0	(2,1)	11100

con lo que la cadena queda: (2)(3) (1,2)(-2) (0,1)(-1) (0,1)(-1) (0,1)(-1) (2,1)(-1) (0,0) y la codificación final: 011,11;11011,01;00,0;00,0;00,0;11100,0;1010

(signos de puntuación añadidos por legibilidad, en realidad es 011111101101000000001110001010).

El total en este caso es de 31 bits sobre los 64 bytes iniciales, es decir una razón de compresión de 16:1 (0.5 bits/pixel).

Las pérdidas se producen sólo en el paso de cuantización (aunque la representación decimal de la DCT puede perder decimales, pero es despreciable).

El estándar MPEG (Moving Picture Experts Group)

- Adoptado a partir de 1998 a propuesta del [comité MPEG](#) del ISO.
- Pretende ser genérico (no orientado a una clase particular de imágenes) aunque funciona mejor con imágenes que contengan pocos objetos en movimiento, o en el que éste sea lento.
- Existen 4 subestándares principales:
 - MPEG1: pensado para el VCD con tasas de 1.5 Mb/s, hoy casi en desuso
 - MPEG2 (H.262): un estándar generalizado, incluyendo el DVD mediante contenedores tipo PS (program stream) y la TV de alta definición (HDTV) mediante contenedores tipo TS (transport stream), así como otros en los que puede configurarse a voluntad la tasa de transmisión binaria.
 - MPEG4: estándar con el objetivo de conseguir tasas binarias muy bajas para transmitir la señal por RDSI a 32 ó 64 Kb/s.
 - MPEG7: estándar de almacenamiento de imágenes en bases de datos, como un SQL para imágenes. No trata la compresión.

Formatos de video comprimido

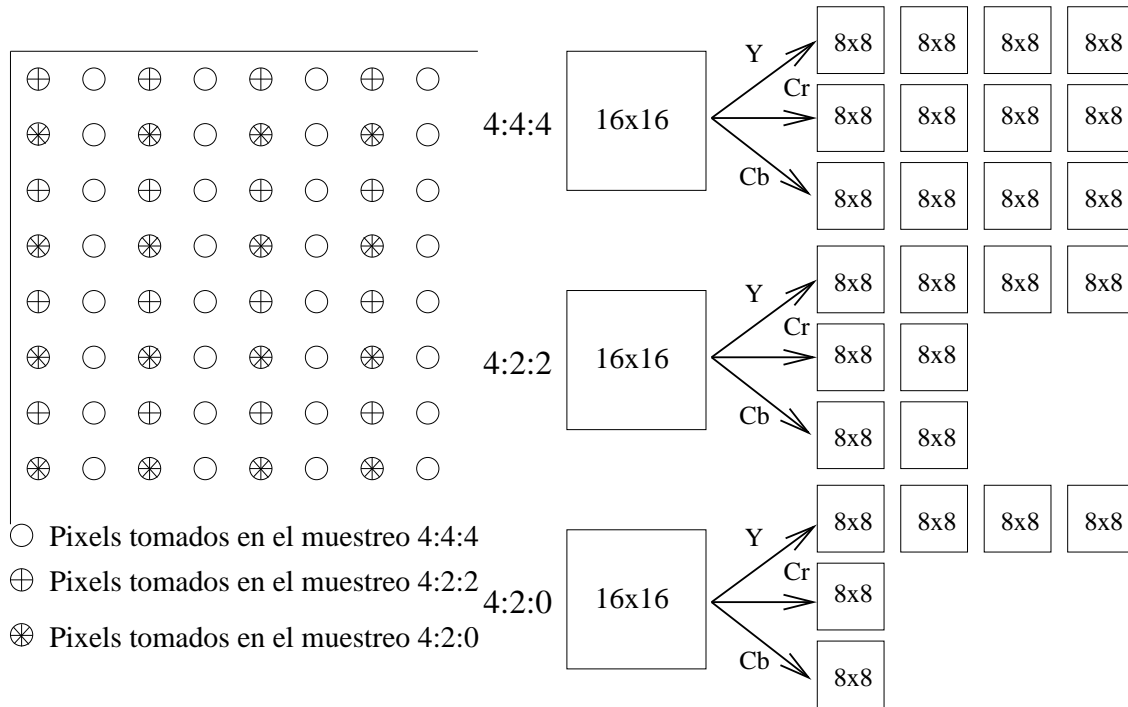
MPEG2

Se supone que se parte de una secuencia de imágenes en color, cada una de las cuales contiene la banda de intensidad (Y) y las dos bandas de crominancia (C_r y C_b) que son imágenes $M \times N$.

Se comienza dividiendo cada banda en bloques de 16×16 pixels, que llamaremos **macrobloques**, y submuestreando las bandas C_r y C_b de acuerdo con uno de los esquemas mostrados en la siguiente página: 4:4:4, 4:2:2 ó 4:2:0.

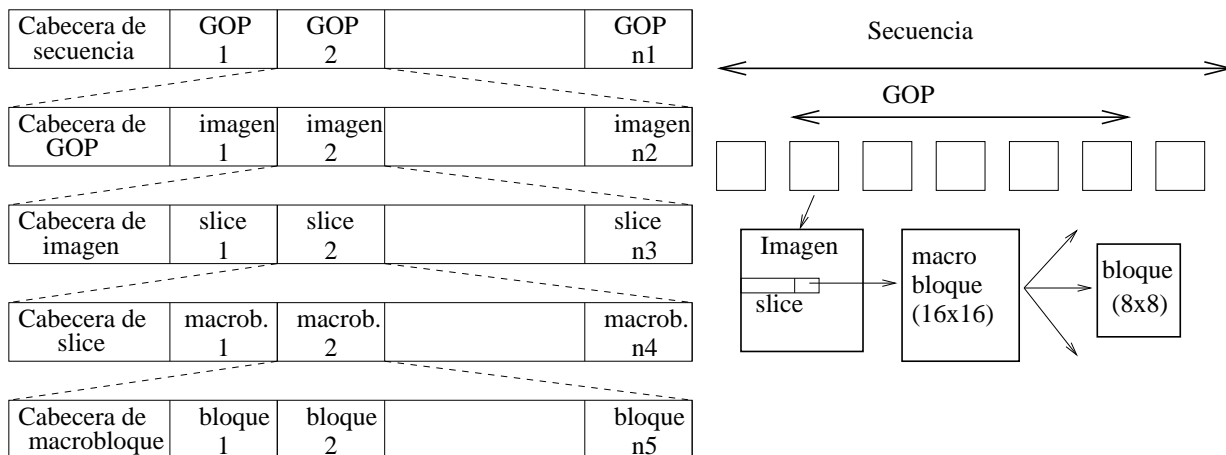
Los procesos de compresión posteriores se aplicarán a cada una de las matrices 8×8 resultantes, que llamaremos **bloque**. Obviamente, el esquema 4:4:4 es el que da mejor calidad, especialmente cromática, pero también mayor tasa binaria.

Formatos de video comprimido



Formatos de video comprimido

La secuencia de imágenes se dividirá jerárquicamente en varias estructuras, como muestra el siguiente diagrama:



Cada grupo de imágenes (GOP) se comprimirá de forma independiente de los otros. Dentro de él MPEG2 establece que puede haber tres tipos de cuadros (imágenes): el I, el P y el B. La cabecera del GOP indica cuántas imágenes siguen, y de qué tipo es cada una.

Formatos de video comprimido

Es posible elegir la composición de los GOPs, pero en general se suelen usar GOPs de 15 imágenes, de la forma IBBPBBPBBPBBPBB (codificación llamada 15/3) o GOPs de 12, de la forma IBBPBBPBBPBB (llamada 12/3).

Los tipos de imágenes I, P y B se diferencian por su codificación como sigue:

- Las imágenes de tipo I no requieren información de ninguna otra imagen para decodificarse. En concreto, se pueden considerar como imágenes estáticas compridas con el estándar JPEG, aunque con una matriz de cuantización distinta.
- Las imágenes de tipo P contienen las "diferencias" entre la imagen de ese instante y la imagen de referencia más cercana. Dicha imagen de referencia sólo puede ser o bien una imagen tipo I o bien otra de tipo P, pero siempre anteriores a la dada.
- Las imágenes de tipo B contienen las "diferencias" entre la imagen de ese instante y la predicha a partir de las imágenes de tipo I y de tipo P más cercanas, incluso en el futuro. Esto significa que las imágenes no se pueden transmitir en el orden en que se toman, p. ej. un GOP tipo 12/3 debería transmitirse como:

Original : I1 B2 B3 P4 B5 B6 P7 B8 B9 P10 B11 B12 I13 ...

Transmitido: I1 P4 B2 B3 P7 B5 B6 P10 B8 B9 I13 B11 B12 ...

Por tanto, las imágenes se mostrarán con un retraso de un GOP.

Formatos de video comprimido

Explicemos qué se entiende en la página anterior por "diferencias".

Respecto a las imágenes de tipo P, sea R su imagen de referencia (la I ó P anterior más cercana). Se buscan macrobloques en P que:

- Sean idénticos al correspondiente macrobloque de R. Esto se indica con un código especial y el macrobloque se omite. Son las áreas de la imagen sin cambios.
- Sean idénticos a otro macrobloque de R. En ese caso sólo se envía el vector de movimiento (número de macrobloques del desplazamiento en x e y). Esto corresponde a objetos que se han movido de un lugar a otro.
- Sean similares al correspondiente, o a otro, macrobloque de R. En ese caso, se usa un codificador predictivo que genera a partir de R el bloque predicho, sea P_c . Entonces se envía el vector de movimiento y las diferencias entre P y P_c comprimidas en JPEG. Esto corresponde a objetos que se han movido una distancia no múltiplo del tamaño del macrobloque, o que han cambiado ligeramente.
- No sea similar a ningún macrobloque de R. Debe enviarse el contenido del macrobloque de P, comprimido en JPEG. Esto ocurre cuando aparecen objetos nuevos en una imagen.

La tasa de compresión varía con el macrobloque, pero puede estar en el orden de 7:1.

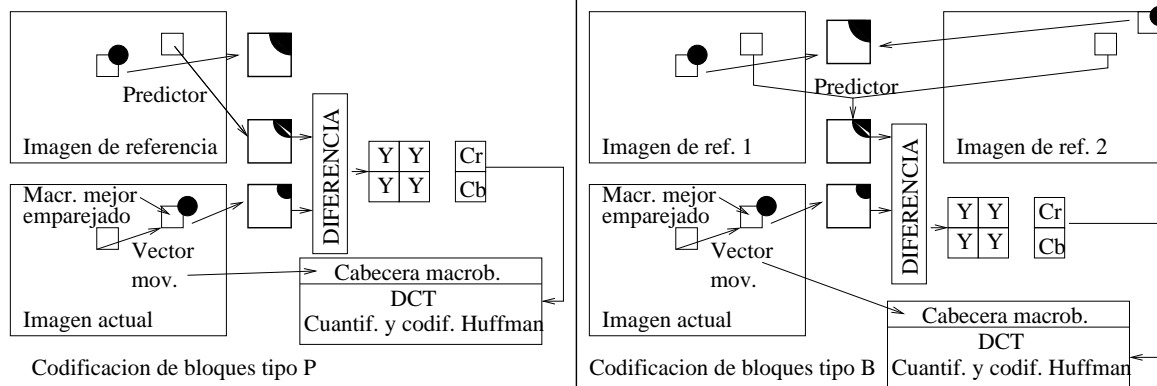
Formatos de video comprimido

Respecto a las imágenes de tipo B, se procede igual que en el caso de las P para macrobloques idénticos en valor y posición, o idénticos desplazados. La diferencia es que aquí es más probable encontrar tal identidad, dado que buscamos no en una, sino en dos imágenes de referencia.

Respecto a los macrobloques similares, se usa igualmente un algoritmo predictivo con fórmula simple, pero esta vez usando dos imágenes de referencia. La predicción, sea B_c , suele ser por tanto bastante correcta y por tanto se comprime mucho más, dado que los valores diferencia $B - B_c$ suelen ser muy pequeños. La tasa de compresión para macrobloques B puede llegar hasta 100:1.

Finalmente, hacer notar que todo el proceso puede hacerse sobre imágenes que han sido generadas de modo entrelazado, o no entrelazado. En las primeras, se pueden crear los macrobloques ordenando (entrelazando posteriormente) las líneas, y codificar esto, o crear en cada macrobloque los dos microbloques superiores con el campo impar y los dos inferiores con el campo par. Esto último es lo más apropiado para comprimir bien el movimiento.

Formatos de video comprimido



niveles/perfiles	SIMPLE 4:2:0 Sin cuadros B	PRINCIPAL 4:2:0 Cuadros I,P,B	ESCALABLE SNR 4:2:0 Cuadros I,P,B	ESCALABLE ESPACIAL 4:2:0 Cuadros I,P,B	ALTO 4:2:2 Cuadros I,P,B
ALTO 1920x1152		80 Mb/s			100 Mb/s 25 en la capa base
ALTO-1440 1440x1152		60 Mb/s		60 Mb/s 15 en la capa base	80 Mb/s 20 en la capa base
PRINCIPAL 720x576	15 Mb/s	15 Mb/s	15 Mb/s 10 en la capa base		20 Mb/s 4 en la capa base
BAJO 352x288		4 Mb/s	4 Mb/s 3 en la capa base		

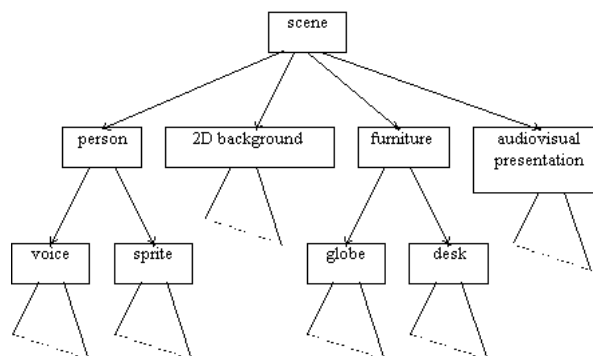
Posibles formas de codificación en MPEG-2

Formatos de video comprimido

MPEG4

A diferencia de MPEG2, cuyas unidades de compresión son las secuencias, las imágenes y los bloques, MPEG4 pone el foco en el contenido. Representa unidades de contenido aural, visual o audiovisual llamadas *media objects*, describiendo la composición de estos objetos para crear escenas audiovisuales. Para ello debe multiplexar apropiadamente los datos asociados con cada objeto, y comprimirlos de acuerdo al medio de transmisión. El uso de objetos abre la posibilidad de interactividad.

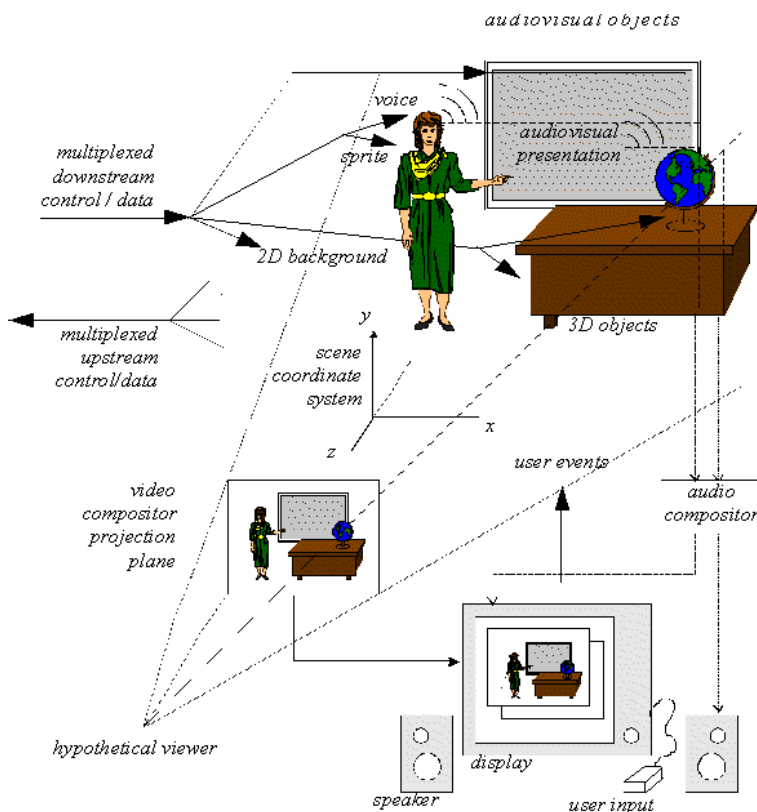
MPEG4 toma para la descripción de escenas conceptos de VRML (Virtual Reality Modelling Language), tanto en la estructura (nodos de objeto y árbol de escena) como en la funcionalidad (cómo se representa la escena).



Tecnologías Multimedia en Web

Formatos de video comprimido (IX)

Formatos de video comprimido



En este ejemplo de escena hay varios tipos de *media objects*: objetos 3D (mesa, globo terráqueo) sintetizados a partir de primitivas geométricas, fondo de la imagen (imagen estática), sprite (la cara, pequeño cuadro de imagen en movimiento), sucesión de imágenes (la presentación), streams de audio comprimido (la voz, el sonido de la presentación).

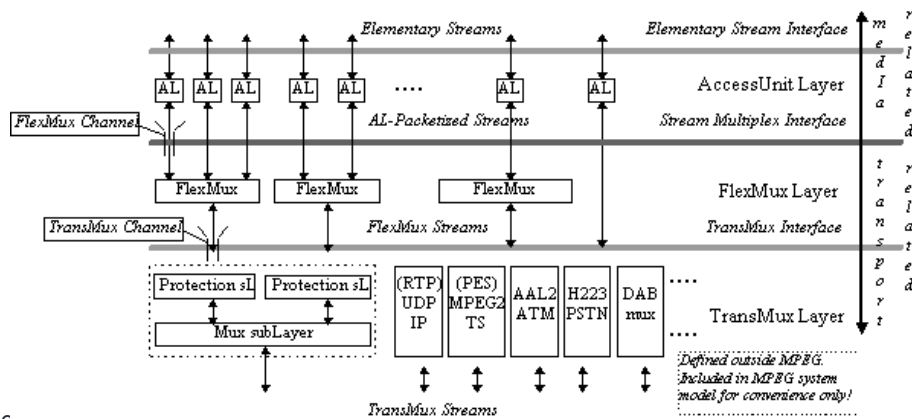
La representación de objetos 3D requiere un marco de referencia (coordenadas de la escena) y la especificación de la posición del hipotético espectador. La presentación de todo requiere sincronización perfecta de los tiempos asociados a cada objeto. El usuario podría decidir representar sólo algunos de los objetos, o hacerlo de otro modo.

Tecnologías Multimedia en Web

Formatos de video comprimido (X)

Formatos de video comprimido

Los streams elementales (ES's) pueden ser datos comprimidos en diversos formatos, definidos por otros estándares. La capa de acceso contiene varias unidades de acceso (AL's) que identifican la ES's presentes y les añaden información del reloj de referencia y marcas de tiempo. Luego, la capa de multiplexado flexible (*FlexMux*) junta todos estos ES's tomando las partes de éstos que deban llegar al destino de modo simultáneo (o anterior a su presentación), p. ej. la siguiente porción de video y su sonido asociado. También, y dependiendo de los recursos reales de la red de transmisión, pueden omitir parte de las ES's, si éstas están preparadas para transmisión de calidad creciente.

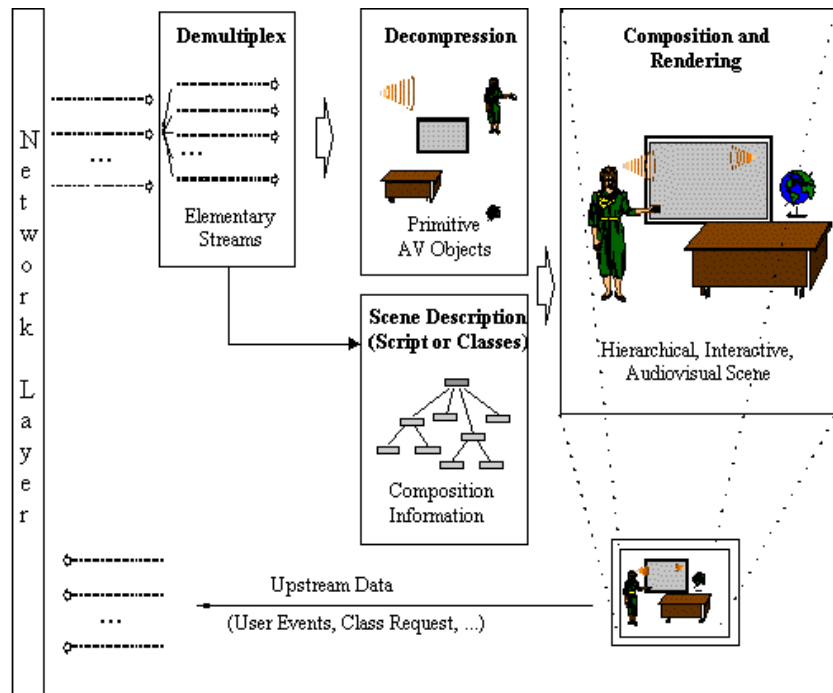


Tecnologías Multimedia en Web

Formatos de video comprimido (XI)

Formatos de video comprimido

En la descompresión y presentación se realiza el proceso inverso: la stream es descompuesta en streams elementales, éstas se descomprimen y se sitúan en la parte de la pantalla que indica el árbol de descripción de escenas (o, en el caso de objetos 3D, se representan como si se estuvieran viendo desde el punto que indica el árbol). El usuario puede cambiar algunas cosas, o incluso pedir que se le envíen datos diferentes (p. ej.: si cambia el punto de vista, el fondo visto desde la nueva posición).



Formatos de video comprimido

Compresión de video con pérdidas: los estándares [H.261](#), [H.263](#) y [H.264](#).

Son estándares aprobados por la unión internacional de telecomunicaciones (ITU) destinados sobre todo a la transmisión de video a través de canales con ancho de banda limitado. Las especificaciones completas están disponibles públicamente.

En concreto, el H.261 está pensado para videoconferencia, con tasas binarias entre los 64Kbps y los aprox. 2Mbps. El H.263 está pensado para la transmisión de video y audio digital desde el formato PAL hasta el HDTV. Los algoritmos de H.261 y H.263 son muy similares respectivamente al MPEG1 y al MPEG2.

Respecto a H.264, es muy similar a MPEG4 (véanse las diferencias [aquí](#)), aunque con algunas mejoras importantes, y en general está sometido a casi las mismas patentes. H.264 fue escogido p. ej. por Apple (es el formato contenido en el contenedor QuickTime) y por fabricantes de hardware que realizan implementaciones hardware del descompresor, frecuentemente asociadas a [arquitecturas ARM](#).

Formatos de video comprimido

Compresión de video con pérdidas: el [estándar VP8](#).

En respuesta a los problemas ocasionados por los formatos propietarios y sus patentes, Google decidió impulsar el desarrollo de un estándar que, si bien está patentado, es público y con la garantía de poder usarse por cualquiera sin pagar regalías.

Funciona siempre partiendo de imágenes en el espacio de color YUV, a 8 bpp, 4:2:0.

Cada frame se descompone en macrobloques de 16×16 pixels en Y (8×8 en U y V) y cada uno de éstos en microbloques de 4×4 en Y (2×2 en UV). La unidad mínima es el microbloque, no el pixel.

Las transformadas DCT y WHT (coseno y Walsh-Hadamard) operan sobre microbloques, y están implementadas en su versión de enteros, no de coma flotante.

Hay sólo dos tipos de frames: intra (el tipo I del MPEG) que se decodifican sin referencia a ningún otro e inter (tipo P del MPEG) que se decodifican usando sólo frames I ó P anteriores (no hay predicción bidireccional, ni por tanto frames tipo B).

Los frames tipo I se comprimen usando DCT ó DWT, y los residuos se comprimen a su vez sin pérdidas usando compresión predictiva.

Todo se codifica luego usando un código de Huffman adaptativo (la tabla de probabilidades para construir el árbol va actualizándose).

Compresión de video con pérdidas: el [estándar Theora](#).

Está basado en un procedimiento anterior llamado VP3, desarrollado por una empresa ([On2](#)) que cedió los derechos del compresor a la [fundación Xiph.org](#). Posteriormente On2 fue comprada por Google, que usó también VP3 como base de VP8. Así pues, Theora y VP8 son bastante similares, aunque Theora se distribuye bajo una licencia libre tipo BSD, lo cual es una garantía frente a posibles litigios incluso mayor que la de Google.

El documento completo del estándar [está accesible](#), así como la implementación de referencia, a través de la [biblioteca libtheora](#).

Formatos de video comprimido

Contenedores de audio y video

Son formatos de los archivos que contienen streams comprimidas de audio y/o su video asociado. Básicamente establecen cómo se intercalan (*interleave*) las streams que intervienen, en qué formato está cada una, cuál es su tamaño, metadatos relacionados con el contenido, etc. indicando cómo y dónde (bien en la cabecera, bien a lo largo del archivo, o en ambos sitios) se guardará esta información.

Entre los formatos contenedores más populares encontramos:

- [AVI](#), propiedad de Microsoft. Existe [una descripción](#) más detallada que la de la propia empresa.
- [Flash](#), propiedad de Adobe Inc., aunque recientemente Adobe ha anunciado que lo abandona. Nunca estuvo debidamente documentado, precisamente porque su objetivo era permitir la visualización de contenido pero no su descarga, y ello exigía *plug-ins* (programas o bibliotecas adicionales) de código cerrado específicas para cada sistema operativo.

- [Matroska](#) es un formato contenedor basado en XML y completamente abierto y libre de patentes. Es probablemente el más flexible, dado que permite almacenar cualquier formato de audio o video presente o futuro, que debería autodescribirse usando los metadatos en XML.
- [QuickTime](#) es el formato propietario de Apple. Está pensado especialmente para almacenar video en H.264 y audio en ACC ó MP3.
- [Ogg](#) es el contenedor libre de la fundación Xiph.org. Su objetivo es sobre todo el almacenamiento de audio con pérdidas en Vorbis o sin perdidas en FLAC y video con pérdidas en Theora o en [Dirac](#).
- [WebM](#) es el formato desarrollado por Google para almacenar el video en VP8. Se usa p. ej. por la popular plataforma [YouTube](#).

Presentación WEB de información multimedia: HTML5

Véase la bibliografía indicada. No hay transparencias de este tema.