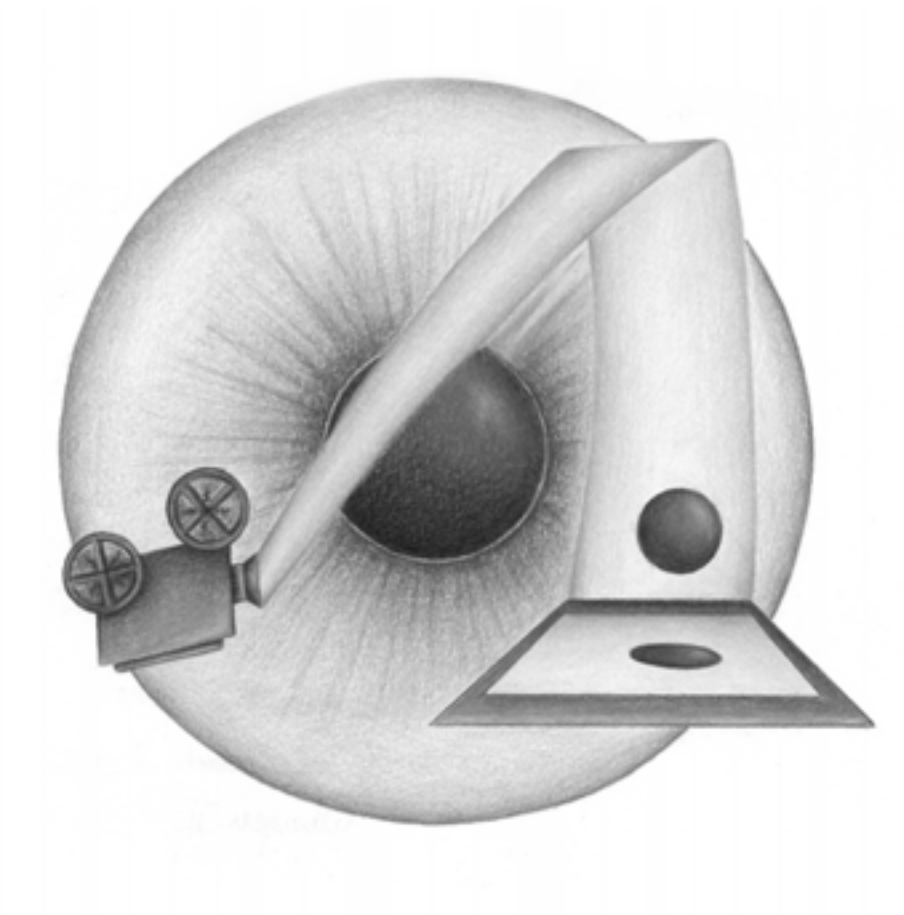
Compresión de vídeo para multimedia



Índice

Etapa 1: Códecs de vídeo para PC

Introducción

Las empresas de *software* para PC pronto consideraron las posibilidades de incorporar materiales multimedia en los programas, con lo que podía conseguirse un gran atractivo en juegos (escenas de tránsito entre apartados registradas en vídeo) o en aplicaciones como enciclopedias o bases de datos. La necesidad y las posibilidades tecnológicas de llevar a cabo reproducciones de audio o vídeo en ordenadores surgió con bastante anterioridad a los primeros estándares del MPEG, por lo que empezaron a aparecer distintos formatos de fichero y algoritmos de compresión propietarios.

Los primeros ordenadores personales en los que se introdujeron aplicaciones de reproducción y registro de señales audiovisuales fueron los Macintosh de Apple y, sobre todo, los Amiga de Atari. La idea básica de estos sistemas se trasladó pronto al PC de IBM, lo que condujo a la popularización de varias aplicaciones que permitían la reproducción multimedia. A pesar de que en 1990 aparece el MPEG-1, cuya intención inicial es desplazar todos los codificadores de vídeo propietarios anteriores, la respuesta de los desarrolladores no es inmediata y se siguen produciendo nuevas versiones de algoritmos propietarios y mejoras constantes, de manera que actualmente varios codificadores comparten el mercado de vídeo para PC. Actualmente el mercado está dividido entre algoritmos que provienen de un estándar, algoritmos totalmente propietarios o algoritmos basados en algún estándar con mejoras propias.

En este apartado veremos las características básicas de la arquitectura de ficheros AVI. Esta arquitectura es suficientemente flexible para permitir que los desarrolladores puedan introducir nuevas técnicas de compresión de vídeo y audio que mejoran tanto la calidad de reproducción como la capacidad de compresión. La arquitectura QuickTime tiene características parecidas y no será considerada en detalle. También veremos los principales códecs de vídeo, las técnicas de compresión que utilizan y los parámetros básicos para la codificación.

La arquitectura AVI

Conceptos elementales

El formato de fichero AVI (*Audio Video Interleave*: 'audio y vídeo entralazados') fue definido por Microsoft para permitir que diferentes desarrolladores pudieran integrar de forma sencilla aplicaciones de compresión y descompresión de materiales multimedia en el sistema operativo Windows.

La principal característica de este formato es su flexibilidad para que diferentes empresas de *software* puedan proporcionar herramientas de compresión y descompresión distintas que a su vez pueden ser utilizadas por otros programas. Consideremos como ejemplo el Adobe Premier, que es una aplicación para la edición de vídeo sobre PC. Este programa puede utilizar los algoritmos de compresión desarrollados por Intel (por ejemplo, Indeo 5.2) sin necesidad de que los programadores de Adobe necesiten conocer ningún detalle técnico de cómo realiza y gestiona Indeo la codificación de vídeo. Por otra parte, los archivos codificados por Indeo pueden ser reproducidos por el reproductor multimedia de Windows sin que los programadores de Intel tengan la más remota idea de cuál es el código del reproductor multimedia. Además, este último tampoco necesita disponer de los detalles de Indeo, que utiliza técnicas de compresión basadas en la transformada Wavelet y que no guardan ninguna relación directa con los algoritmos de los codificadores de Microsoft.

En principio, un formato AVI puede contener cualquier tipo de compresor o técnica de compresión de vídeo siempre que el desarrollador del código siga las normativa de Microsoft para integrar su códec en el sistema operativo. Actualmente muchos formatos de vídeo para PC, con extensiones diferentes de AVI, siguen siendo en esencia formatos AVI. El cambio de nombre en la extensión se debe en muchas ocasiones a que el desarrollador desea ejecutar su propio programa de reproducción en vez del reproductor multimedia del sistema operativo. El formato AVI se desarrolló para las primeras versiones de Windows (Windows 3.1) y sigue teniendo una posición dominante en las aplicaciones de captura y edición de vídeo para PC.

En un principio, Microsoft proporcionó soporte para desarrollar aplicaciones basadas en archivos AVI por medio de la API Video for Windows. Actualmente, el soporte se proporciona mediante DirectShow (un conjunto de librerías y objetos que el desarrollador de programas puede utilizar para manejar archivos AVI). DirectShow también permite el manejo de archivos según el estándar MPEG y según los nuevos formatos de vídeo de Microsoft WMA y WMV (Windows Media Audio y Windows Media Video).

Uno de los principales problemas del formato AVI (y de muchos otros formatos de archivos multimedia como el WAV, AUD, etc.) es que la reproducción del material audiovisual requiere que el archivo esté localizado en el disco duro o en el CD-ROM del ordenador. No es posible reproducir estos ficheros de forma directa como un cliente conectado a un servidor multimedia. Esta característica ya está incorporada en los nuevos archivos WMA y WMV, por lo que se prevé que progresivamente vayan sustituyendo a los archivos AVI. Veremos también algunas ideas elementales sobre la arquitectura de estos nuevos formatos al final de esta etapa.

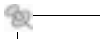
Estructura del fichero AVI

Los ficheros AVI son un caso especial de los ficheros RIFF (*Resource Interchange File Format*). La estructura de ficheros RIFF fue definida por Microsoft y es una actualización de la estructura RIF que utilizaba originalmente el ordenador Amiga de Atari.

Los ficheros de audio no comprimido WAV también son un caso especial del formato RIFF. Los formatos AVI y WAV se han convertido en un estándar de facto.

La esencia del formato AVI es que contiene una cabecera en la que pueden definirse varias pistas de vídeo, audio, texto (básicamente para subtítulos) o información suplementaria. Cada pista dispone de una cabecera propia donde se identifica el tipo de codificador que se ha utilizado para realizar la compresión. Las muestras de audio, vídeo o texto están en paquetes (*chunks*) cuyos tamaños puede configurarlos el desarrollador del códec. Los paquetes de audio y vídeo deben estar entrelazados en el fichero, pero tampoco se especifica cuál debe ser la relación de entrelazado.

Esta libertad proporciona mucha flexibilidad al desarrollador para organizar los datos según su conveniencia. Así, es posible que un desarrollador considere oportuno registrar varios *frames* consecutivos de vídeo y después el audio asociado a estos *frames*, mientras que otro considere más adecuado intercalar las muestras correspondientes al audio entre los *frames* individuales de vídeo. Es incluso posible, y algunos de los primeros códecs así lo implementaban, que toda la parte de audio esté situada al final de la parte de vídeo. Esta circunstancia es una de las causas principales de que los archivos no puedan ser reproducidos directamente desde conexiones remotas a Internet.



Especificación formato AVI

La especificación completa del formato y la estructura de los ficheros RIFF o AVI puede encontrarse fácilmente en internet (Microsoft y muchos otros web). El documento que especifica el formato tiene una extensión de varias de páginas de lectura muy tediosa. Se reproducen las primeras líneas del estándar para que el lector tenga una idea de lo que significa "tedioso" en este contexto.

'RIFF' (4 byte file length) 'AVI' // file header (a RIFF form)
'LIST' (4 byte list length) 'hdrl' // list of headers for AVI file
The 'hdrl' list contains:
    'avih' (4 byte chunk size) (data) // the AVI header (a chunk)
    'strl' lists of *stream* headers for each *stream* (audio, video, etc.) in the AVI file. An AVI file can contain zero or one video *stream* and zero, one, or many audio *streams*. For an AVI file with one video and one audio *stream*:
'LIST' (4 byte list length) 'strl' // video *stream* list (a list)
The video 'strl' list contains:
    'strh' (4 byte chunk size) (data) // video *stream* header (a chunk)
    'strf' (4 byte chunk size) (data) // video *stream* format (a chunk)

Evidentemente, no es necesario que un desarrollador de aplicaciones multimedia conozca todos estos detalles para realizar programas que permitan editar o manipular los contenidos del fichero. Microsoft proporciona varias utilidades para el manejo de los archivos en un nivel más elevado.

Cabecera del fichero AVI

La cabecera del formato AVI empieza con la palabra clave (en modo texto) "avih" (*avi header*). La información que contiene la cabecera se muestra en el siguiente ejemplo, en el que se comentan algunos de los parámetros que tienen más interés.

- ***Frames*Number**: define el número total de *frames* que contiene el fichero.
- ***Streams*Number**: define el número total de *streams* que contiene el fichero. Es habitual que el número de *streams* sea igual a dos, que corresponde a una pista de vídeo y una de audio. En el caso de un fichero estéreo dispondríamos de una pista de vídeo más dos de audio. Si se incorpora información de subtítulos, el número aumenta al considerar los *streams* de texto.
- **Initial*Frames***
- **MaxBytes**
- ***BufferSize*.**
- **MicroSecondsPer*Frame***
- ***Frames*PerSecond**: define el número de imágenes por segundo que están codificadas en el fichero.
- **Size**: indica el tamaño de la imagen en píxeles, por ejemplo 320×240.
- **Flags**

Cabecera de *stream* de audio o vídeo

Los parámetros incluidos en la cabecera de *stream* son fundamentales para comprender la filosofía general de la arquitectura AVI. Los parámetros que se incluyen en la cabecera, comentando el significado de los más importantes, son:

- ***Stream*Header.** Este parámetro siempre tiene el texto "strh". Con ello, al leer el fichero se puede identificar que empieza la cabecera que define uno de los *streams* del fichero. La palabra clave "strh" no puede encontrarse en ninguna otra parte del fichero.
- ***Stream*Type.** Este parámetro también está proporcionado en modo texto. Puede tomar los valores "avid" o "auds". El primer caso indica que el *stream* asociado es de vídeo, mientras que el segundo especifica que se trata de información de audio.
- ***Stream*Handler.** Este parámetro especifica el códec con el que se ha realizado la codificación del *stream*. Es también una cadena de cuatro caracteres ASCII. Cada códec tiene una palabra clave asociada. Las palabras clave pueden solicitarse a Microsoft para que las registre, aunque no es estrictamente necesario para que el sistema funcione correctamente. Así, por ejemplo, el Cinepack tiene como palabra clave "cvid".
- **SamplesPerSecond.** Número de muestras por segundo (*frames* de vídeo o muestras de audio según el contexto). Este parámetro ya se proporciona como un número entero.

- **Priority.** Es posible especificar la prioridad con la que el decodificador debe tratar cada pista. Este parámetro es informativo y muchos decodificadores no lo utilizan.

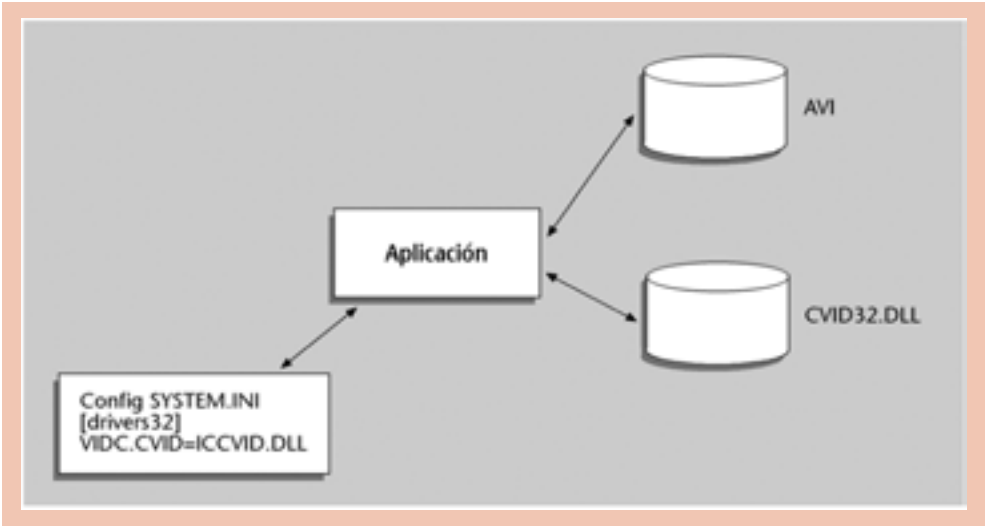
- **Quality.** Coincide con el parámetro de calidad que solicitan muchos codificadores al configurar la calidad. Se expresa como un número entero entre 1 y 100.

- **SampleSize.** Tamaño de cada muestra.

- **BufferSize.** Tamaño del *buffer*.

Estructura de librerías

En la siguiente figura se representa un diagrama explicativo de cómo gestiona el sistema el registro o reproducción de ficheros AVI que utilizan distintos códecs.



Cuando el usuario solicita reproducir un fichero AVI, el reproductor lee la cabecera de los *streams* del fichero y busca las palabras clave que especifican el códec. Para determinar el códec de vídeo buscará la palabra clave "avid" y posteriormente leerá el código de cuatro caracteres que especifica el compresor. Si los cuatro caracteres son "cvid", el sistema operativo abre el fichero de configuración del sistema SYSTEM.INI y busca en la sección [drivers32] la palabra clave que identifica el codificador utilizado. En nuestro ejemplo, la palabra clave será VIDC.CVID (VIDC significa *video* códec, y todos los codificadores de vídeo deben comenzar con estas cuatro letras seguidas por un punto). La palabra clave indica la librería dinámica que el reproductor deberá cargar para poder decodificar correctamente el vídeo.

Las librerías dinámicas asociadas a un códec proporcionan las herramientas necesarias para que los diferentes programas puedan utilizar el codificador y el descodificador de vídeo. Las rutinas proporcionadas interactúan con los entornos de desarrollo de alto nivel como Video for Windows (VFW) o DirectShow de forma que el programador dispone de diferentes funciones para manipular los archivos. Una de las funciones (simplificaremos mucho el proceso real, nombres de funciones y parámetros

para no extendernos en detalles) podría ser GetVideo*Frame*, que retorna un puntero a la información del siguiente *frame* en formato descomprimido. Este *frame* descomprimido puede ser utilizado por una aplicación de reproducción de ficheros AVI para mostrar el *frame* por pantalla. La función recíproca PutVideo*Frame* tendría como parámetro de entrada un puntero a *frame* sin comprimir y se encargaría de realizar la compresión y gestionar su almacenamiento en el fichero. Esta función puede utilizarse por un programa de captura para comprimir y almacenar un fichero de vídeo.

Cuando instalamos un nuevo códec en el sistema operativo, el programa de instalación del códec copia las DLL (*Dinamic Link Library*, 'librerías dinámicas') en el directorio de sistema operativo. Después modifica el archivo SYSTEM.INI introduciendo la palabra código de cuatro caracteres que identificará a este compresor e indica las librerías dinámicas que deben cargarse para poder utilizarlo.

Como ejemplo, supongamos que deseamos instalar un nuevo códec definido por la palabra clave VDTT. Nuestro programa de instalación copiaría las librerías dinámicas VDTT32.DLL en el sistema operativo y modificaríamos la sección [drivers32] del archivo SYSTEM.INI introduciendo la sentencia VIDC.VDTT = VDTT32.DLL. Nuestra librería DLL deberá proporcionar las funciones necesarias para que cualquier programa pueda utilizarla con Video for Windows o DirectShow.

Los códecs instalados en un determinado sistema pueden visualizarse en el icono de Multimedia de la configuración del sistema. En la figura se muestra un listado típico de códecs instalados en un sistema operativo.

Listado típico de códecs de vídeo instalados en un sistema operativo

La filosofía utilizada por QuickTime para poder gestionar distintos códecs en los archivos con formato MOV es parecida a la que hemos descrito para Windows. Los detalles se han expuesto para el sistema operativo Windows 3.1, que es la primera versión en la que se incluyeron los archivos multimedia AVI. Para otras versiones de Windows, el proceso de instalación y uso de las librerías es parecido, con la salvedad de que el archivo SYSTEM.INI se sustituye por el registro de Windows.

Codificación de vídeo para PC

La selección de un códec depende de muchos factores y no existe un único criterio para establecer el que tiene mejores prestaciones. Generalmente, cada códec está pensado para un entorno de aplicación diferente. Los procedimientos para codificar vídeo que debe reproducirse desde disco duro pueden ser muy distintos de los que se usan para reproducir el vídeo desde un CD-ROM. La razón fundamental de estas diferencias es la velocidad con la que el reproductor puede acceder a los datos.

Criterios generales de selección de códecs

Uno de los primeros aspectos que hay que tener en cuenta al elegir un códec es la velocidad a la que se podrá acceder a los datos del fichero durante la reproducción del material audiovisual. Generalmente se suponen los siguientes casos típicos:

- **Disco duro.** Se utiliza en algunos videojuegos en los que algunas secuencias cortas se descargan al disco duro durante la instalación para que puedan ser reproducidas con gran calidad. La principal ventaja que ofrece la reproducción desde el disco duro es que la velocidad de escritura y lectura es muy alta (8 MBytes/s en discos IDE y hasta 40 MBytes/s en SCSI II. Tened en cuenta que estas cifras se dan en *megabytes*, mientras que las tasas de compresión se suelen proporcionar en megabits). Otro ejemplo típico de vídeo destinado a disco duro son los servidores de vídeo para intranets. El vídeo en disco duro también se utiliza para realizar el proceso de captura y edición. Los códecs más utilizados en este tipo de aplicaciones son los M-JPEG (compresión independiente en cada *frame*. No utilizan compresión temporal, de forma que la edición puede realizarse con gran calidad). Algunos sistemas de captura y edición de vídeo profesionales utilizan, cada vez más, el MPEG-2 con un perfil de color 4:2:2 y muy poca compresión. Este perfil del MPEG-2 permite la edición con calidad y el flujo de vídeo puede llegar hasta los 50 Mbps. Otro ejemplo típico en el que se realiza la grabación en disco duro es el que forman los sistemas de vídeo-vigilancia digitales. En este caso las imágenes suelen comprimirse bastante (calidad equivalente a VHS) debido a que el disco duro debe contener una gran cantidad de horas de información. Los códecs típicos que se utilizan son el Cinepack, Indeo, etc., con factores de calidad altos y compresión en tiempo real.

- **DVD.** El DVD-Video establece el MPEG-2 *Main Level Main Profile* como estándar de codificación. Las tasas de bits típicas con las que registra el vídeo están situadas entre los 8 Mbps y los 12 Mbps. La calidad de imagen es excelente. En un DVD-ROM puede utilizarse otro tipo de códecs. El uso de un códec como DIVX o MPEG-1 permitiría aumentar considerablemente el número de horas de registro manteniendo calidades aceptables. De todas formas, dado que tanto la velocidad de acceso como la capacidad de almacenamiento son muy altas, es recomendable utilizar factores de compresión reducidos a favor de la calidad.

- **CD-ROM.** Los códecs más típicos son el MPEG-1, el DIVX o alguna otra versión del MPEG-4. Al ajustar los parámetros de compresión es conveniente tener en cuenta que no todos los usuarios disponen de CD-ROM con la misma velocidad de lectura. Es habitual suponer que todavía existen usuarios con CD-ROM con velocidades típicas de x2 (300 Kbytes/s) o x4 (600 Kbytes/s). Estas velocidades permiten codificar el vídeo con tasas entre 1,2 Mbps (velocidad x1) a 4,8 Mbps (velocidad x4). Esta última velocidad permite incluso utilizar codificadores MPEG-2 de gran calidad, aunque reduce el tiempo total de reproducción a una cuarta parte. Pueden utilizarse también codificadores del tipo Intel, Cinepack o Sorenson, que producen resultados parecidos. No es aconsejable aumentar excesivamente la tasa de codificación aunque se disponga de CD-ROM de muy alta velocidad de lectura. Tened en cuenta que la velocidad nominal que proporcionan los fabricantes es el valor máximo y que los valores sostenidos suelen estar muy por debajo de los valores máximos.

- **Internet.** Es donde se presentan mayores problemas en la selección del codificador debido a las diferencias entre velocidades de descarga que pueden tener los usuarios. Son populares los codificadores orientados a tasas de codificación muy bajas para videoconferencia a través de RDSI o línea telefónica (H.263, MPEG-4). También proporcionan excelentes resultados los códecs escalables, que pueden orientarse a distintos tipos de usuarios y que deben ser gestionados desde un servidor de *streaming*. Las diferentes versiones de los WMV de Microsoft presentan excelentes propiedades para su distribución a través de Internet. Además, todos ellos permiten ser reproducidos durante la descarga de los ficheros, tanto si se utiliza un servidor web convencional como si se utiliza un servidor de *streaming*.

Selección de parámetros

En este apartado comentaremos el significado de algunos de los parámetros que el usuario puede seleccionar al elegir un determinado codificador. La mayoría de parámetros que comentamos pueden ser elegidos en todos los programas de edición o captura de vídeo (por ejemplo, Adobe Premier).

- **Tamaño de imagen.** El tamaño de la imagen depende del destino final del material multimedia. Durante la fase de digitalización y edición es aconsejable trabajar con el tamaño máximo que permita el sistema y utilizar tasas de compresión muy bajas. Si el destino final del vídeo es disco duro o DVD, es conveniente no reducir el tamaño de codificación. En CD-ROM es aceptable reducir la resolución a un formato del tipo CIF (calidad comparable a VHS). En Internet depende de la velocidad a la que se prevea la conexión. Suelen utilizarse tamaños Q-CIF. Con códecs escalables puede elegirse una opción Q-CIF para la capa base y CIF para las capas auxiliares. Cuando se recodifica un vídeo de calidad alta a un formato de baja calidad es importante que el factor de reducción del tamaño sea un número entero. De este modo se obtiene mayor calidad, ya que la reducción es más simple. Siempre es aconsejable utilizar filtros espaciales para reducir el tamaño. Los que proporcionan mejores resultados son los filtros cúbicos, aunque su compleji-

dad computacional es más elevada. Los filtros lineales proporcionan resultados bastante aceptables. Es fundamental no aumentar nunca el tamaño de la imagen al recodificar, ya que la calidad se degrada notablemente.

- ***Frames* por segundo.** Dependen también del destino final del vídeo. En disco duro y DVD es altamente recomendable utilizar las veinticinco imágenes por segundo (25 fps –*frames per second*–). La distribución de vídeo a través de redes locales de alta velocidad también puede realizarse a 25 fps (instalaciones de vídeo *on demand*). En CD-ROM también pueden usarse 25 fps. En algunos casos se puede reducir la velocidad a 15 fps para aumentar la capacidad del disco. Con 15 fps se obtiene una calidad aceptable en la reproducción del movimiento. En aplicaciones de Internet o videoconferencia es necesario reducir a 8 fps o incluso menos.

- **Factor de calidad.** Generalmente, los compresores admiten que el usuario especifique un factor de calidad en porcentaje del 1 al 100.

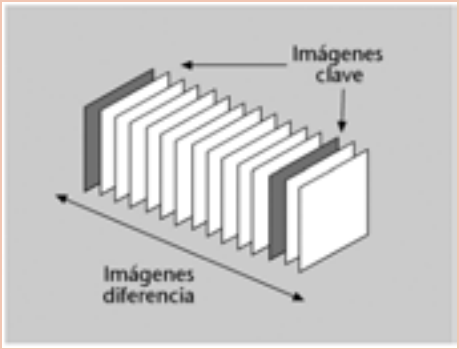
- **Tasa de compresión.** Algunos compresores permiten que el usuario especifique una tasa de compresión objetivo. Puede seleccionarse que se dé prioridad a la tasa de compresión o al factor de calidad. Hay que tener en cuenta que muchos códecs están optimizados para trabajar en un margen prefijado de compresión. Así, el Cinepack y el Indeo 3.2 producen excelentes resultados cuando la tasa de compresión se sitúa en torno a 300 Kbytes/s (CDx2). Si se aumenta el número de bits no se obtienen mejoras aparentes. El H.263 está optimizado para tasas de 64 kbps. El Intel Indeo 5.2 es uno de los códecs que tienen un mayor espectro de funcionamiento. Admite un gran margen de tasas de compresión con buenas prestaciones en todas ellas.

- **Ratio de audio.** Generalmente es conveniente dar preferencia a la calidad de audio frente a la de vídeo. La interrupción de la imagen durante la reproducción es aceptable mientras que la interrupción del audio es intolerable. Debe tenerse en cuenta que todo el ancho de banda asignado al audio no se puede utilizar para el vídeo. Exceptuando las aplicaciones orientadas a DVD o a disco duro, es aconsejable comprimir el audio. El audio digital sin comprimir (calidad CD) requiere un ancho de banda de 1,2Mbps. Pueden conseguirse bandas sonoras de excelente calidad utilizando compresores tipo MP3 utilizando tasas de 128 kbps (estéreo) o 64 kbps (mono). Los codificadores ADPCM también consiguen buena calidad con estas tasas. Para videoconferencia o voz es recomendable utilizar compresores de videoconferencia del tipo G.728 o similares (consultad módulo 4).

- **Relación de *Interleaving Audio Video*.** No todos los códecs permiten que el usuario configure este parámetro. En caso de admitirlo, es recomendable insertar audio con una frecuencia relativamente alta. Relaciones de audio entre cada dos *frames* o audio entre cada cinco o seis son adecuadas. Esperar más *frames* a insertar audio suele originar problemas.

- ***Key frames*.** Muchos compresores permiten que el usuario especifique el número de *frames* delta que se producen entre cada dos *key frames*. Un *key frame* está codificado sin utilizar como referencia otras imágenes, mientras que los *delta frame* utilizan la redundancia temporal entre las imágenes para poder comprimir con

mayor eficiencia. En la figura siguiente se representa una secuencia de imágenes que indican la intercalación entre los dos tipos de *frames*. Prestad atención al hecho de que en el contexto de compresores propietarios se habla de *key frames* y *delta frames* en lugar de *frames* de tipo I, tipo P o tipo B (como en los estándares). La razón es que cada desarrollador puede elegir estrategias de predicción diferentes que no tienen por qué basarse en la compensación de movimiento *forward* y *backward*. En general, los *key frames* se utilizan para el acceso aleatorio a la secuencia de vídeo. Si el usuario debe reproducir el vídeo y puede pararlo, rebobinarlo, etc., es conveniente que el número de *key frames* sea lo más elevado posible. Valores de un *key frame* cada cuatro o seis *frames* son adecuados para estas aplicaciones. Cuando se trata de comprimir mucho la información (Internet) es adecuado reducir el número de *key frames* para mejorar el factor de compresión. Un *key frame* cada quince o veinticinco *frames* puede ser adecuado, o incluso menos. También hay que tener en cuenta el movimiento de la secuencia de vídeo. Si existe mucho movimiento debe aumentarse el número de *key frames* para obtener buenas referencias. Aunque parezca contradictorio, con ello se puede aumentar el factor de compresión, ya que los *delta frame* que vienen detrás del *frame* de referencia pueden comprimirse mucho más que si están alejados temporalmente de la referencia. En aplicaciones de videoconferencia (bustos parlantes), el número de *key frames* puede reducirse mucho, ya que se produce muy poco movimiento entre imágenes sucesivas. Al editar un vídeo por corte es conveniente imponer que el *frame* en el que se realiza el cambio de plano sea un *frame* de referencia. Aplicaciones como Adobe Premier permiten que el usuario imponga que algunos *frames* se codifiquen como *key frames*. Es aconsejable imponer que todos los primeros *frames* de un cambio de plano sean de referencia. En algunos casos es conveniente tener en cuenta que algunos códecs resultan adecuados para trabajar con un determinado número de *key frames*. El Cinepack produce excelentes resultados si se inserta un *key frame* entre cada siete o quince *frames*. En cambio, Indeo 3.2 produce los mejores resultados cuando se inserta un *key frame* cada cuatro *frames*.



Intercalación de imágenes clave e imágenes delta

- ***Padding***. Esta opción aparece en muchas aplicaciones de compresión de vídeo y es útil utilizarla cuando el vídeo se destina a soporte CD-ROM. El CD-ROM se divide en sectores de 2 Kbytes. La búsqueda aleatoria de la información se realiza buscando en las cabeceras de estos sectores, por lo que si los *frames* de vídeo no empiezan en el inicio de un sector, la búsqueda aleatoria se puede ralentizar. Al seleccionar la opción de *padding*, el compresor añade ceros hasta que la informa-

ción ocupa un sector completo. Evidentemente, el *padding* reduce la eficiencia de la compresión.

- **Reducción de ruido (filtrado).** Cuando la secuencia de vídeo original que se pretende comprimir contiene ruido se reduce mucho la eficiencia de la compresión. En efecto, la naturaleza aleatoria del ruido hace que resulte difícil de predecir y, por lo tanto, es difícil de comprimir. Cuando la fuente original contiene ruido visible es aconsejable utilizar filtros que eliminen el ruido. Esto puede ocasionar una ligera pérdida de calidad en la definición de la imagen, pero es aceptable y mejora considerablemente el funcionamiento del compresor.

- **Desentralazar.** Cuando se trabaja a resolución completa, las imágenes proceden de dos campos que han sido capturados en instantes de tiempo diferentes y cuyas imágenes se entrelazan línea a línea (las cámaras PAL trabajan así). Cuando existe mucho movimiento en las imágenes es posible que aparezcan componentes de muy alta frecuencia (debidas al movimiento) y que resultan difíciles de comprimir. El problema se ilustra en la siguiente figura, donde la imagen original ha sido capturada con una cámara de vídeo que tenía un movimiento horizontal muy rápido. Observad que las líneas entrelazadas de los dos campos producen efectos de muy alta frecuencia en los pétalos de la flor. Este efecto no se observa cuando se visualiza el vídeo analógico a velocidad normal debido a que el movimiento es muy rápido y el ojo no puede apreciar los detalles de un objeto que se desplaza a gran velocidad. En cambio, cuando intentamos comprimir la imagen completa, estas componentes de alta frecuencia pueden reducir la eficacia del compresor. La opción de desentrelazado permite que el compresor considere cada uno de los campos como imágenes separadas y, por lo tanto, podrá comprimirlas con mayor eficiencia. Es muy útil activar esta opción cuando el vídeo tiene mucho movimiento. No obstante, si la acción es muy estática, pueden obtenerse mejores resultados de compresión si se trabaja con la imagen completa (existen más líneas y, por tanto, hay más redundancia, con lo que el compresor mejora su comportamiento). Se trata de una decisión cuyos resultados debe valorar el usuario en función de las características del vídeo.

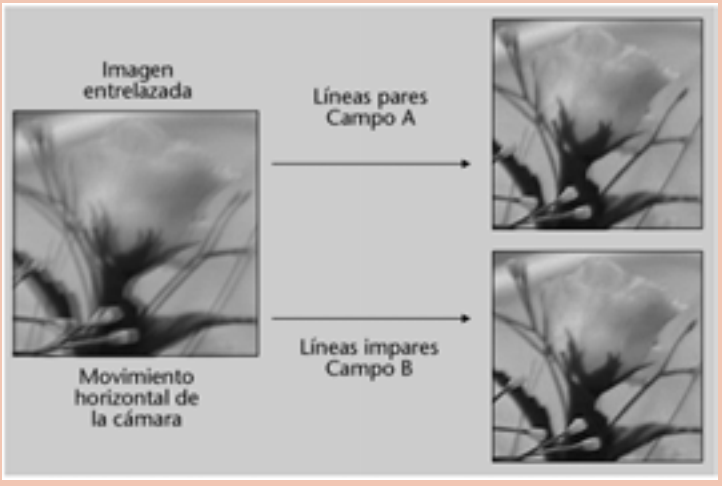


Imagen entralazada con movimiento horizontal rápido de la cámara

Características de los principales códecs

En este apartado revisaremos los principales codificadores de vídeo para PC, sus características tecnológicas y los métodos de compresión utilizados.

Microsoft RLE

[drivers32]
VIDC.MRLE = MRLE32.DLL

Oficialmente es el primer formato de vídeo para PC introducido por Microsoft. La tecnología de compresión está basada en el método RLE (*Run Lenght Encoding*). Las imágenes se descomponen en planos de bits y se codifican los planos de forma independiente mediante longitudes de ceros y unos. El compresor es bastante eficiente para animaciones gráficas (colores planos), aunque sus resultados son muy pobres en imágenes naturales. Actualmente puede considerarse obsoleto, aunque se sigue incluyendo en los sistemas operativos por cuestiones de compatibilidad.

Microsoft Video1

[drivers32]
VIDC.MSVC = MSVIDC32.DLL

También es original de Microsoft y se proporcionaba con la primera versión de Vídeo for Windows. Sólo soporta 16 bits de color y tiene muy baja calidad. También puede considerarse obsoleto.

Intel Indeo 3.2

[drivers32]
VIDC.IV32 = IR32_32.DLL

Este codificador está basado en la codificación vectorial. La codificación vectorial competía directamente con la codificación basada en la transformada coseno antes de que los estándares de la ISO optaran por esta última alternativa. Esencialmente consiste en construir una base de datos de vectores (bloques de imagen) y codificar cada bloque real de la imagen por uno de los vectores de la base de datos.

Indeo 3.2, junto con el Cinepack, son los dos codificadores que dominaron los finales de la década de los ochenta y principios de los noventa. Ambos descodificadores se proporcionaban gratuitamente con el sistema operativo. El primero tiene una profundidad de color de 24 bits, aunque presenta algunos problemas de tono en la codificación. La compresión es más rápida que con Cinepack, pero requiere más carga de CPU durante la reproducción. No se adapta bien para bajas velocidades y no puede

trabajar cuando el vídeo es más alto que ancho. Independientemente de la frecuencia de imagen con la que se trabaje, es aconsejable utilizar un *key frame* cada cuatro *frames*. Originalmente se conocía como Real Time Video 2.1 (RT21).

Cinepack

[drivers32]
VIDC.CVID = ICCVID.DLL

Tiene características muy parecidas a Indeo 3.2 y fue su directo competidor. Originalmente se diseño para Apple, pero rápidamente se incorporó a Windows. El método de compresión está basado en la codificación vectorial y en las diferencias entre *frames*. No se adapta bien a tasas de muy baja velocidad (Internet) ni para imágenes de calidad. Sus mejores prestaciones se consiguen cuando se utilizan factores de compresión en la relación 10:1.

CinepackPro

[drivers32]
VIDC.CVID = ICCVID.DLL

Versión actualizada y compatible con el Cinepack. Permite un mejor control del ancho de banda, es decir, permite que el usuario pueda elegir entre factores de compresión mayores o menores que con la versión anterior. Pueden descargarse versiones demo desde Internet, pero el codificador no es gratuito.

Motion JPEG

[drivers32]
VIDC.???? = ????.DLL

Varios desarrolladores venden productos con este nombre que no son compatibles. Están basados en comprimir cada una de las imágenes con el algoritmo JPEG. Generalmente se proporcionan de manera gratuita con las tarjetas de digitalización de vídeo.

Editable MPEG

[drivers32]
VIDC.???? = ????.DLL

Esencialmente, se trata de codificadores MPEG que sólo contienen imágenes de tipo I. Desde el punto de vista conceptual, son totalmente equivalentes a los Motion JPEG y el cambio de nombre se debe fundamentalmente a cuestiones de marketing. También se proporcionan con las tarjetas de digitalización de vídeo.

VDOWave (VDOLive)

[drivers32]
VIDC.VDOM = VDOWAVE.DLL

Está basado en la transformada Wavelet más algún tipo de compensación de movimiento o diferenciación de imágenes que el desarrollador no especifica. Se instala con el reproductor multimedia de Microsoft. La versión 2.0 tiene una tasa de bits prefijada. La versión 3 es escalable y fue uno de los primeros códecs con esta característica que aparecieron en el mercado.

Indeo 4.1, 4.2, 4.5

[drivers32]
VIDC.IV41 = IV41.DLL

La estrategia de compresión está basada en la transformada Wavelet a partir de una descomposición de la imagen en mosaico. Tiene unas prestaciones excelentes y permite trabajar desde tasas de compresión muy altas hasta factores de calidad muy buenos. Probablemente, hasta la aparición de Indeo 5.1 fue el códec que podía usarse en un espectro más amplio de aplicaciones.

Indeo 5.1, 5.2

[drivers32]
VIDC.IV50 = IV50.DLL

Versión mejorada de Indeo 4.5 con una transformada Wavelet mejorada. Es escalable, de forma que una misma trama de datos puede soportar distintas calidades. Es uno de los que proporciona mejor calidad del mercado, y puede trabajar desde velocidades muy bajas hasta muy altas. Las versiones actualizadas de todos los códecs de Indeo pueden descargarse de Ligos (propietario actual de esta tecnología, que previamente fue desarrollada por Intel).

SFM (Crystal Net Corporation)

[drivers32]
VIDC.SFMC = SFMdemo.DLL

Está pensado para aplicaciones de videoconferencia RDSI y redes de telefonía analógica convencional. Obtiene muy buenos resultados en compresión. La tecnología es propia y está basada en métodos de detección de contornos y ajuste de superficies.

ClearVideo (Iterated Systems)

[drivers32]
VIDC.UCOD = CLRVIDCD.DLL

Es un codificador basado en fractales que tiene excelentes factores de compresión. La empresa desarrolladora fue absorbida por RealNetworks, que actualmente es la que desarrolla esta tecnología.

H.261

[drivers32]
VIDC.M261 = MSH261.DLL

Es un codificador basado en un estándar de la ITU. Existen diferentes empresas que proporcionan versiones de este estándar. Analizaremos sus detalles en secciones posteriores.

H.263

[drivers32]
VIDC.M263 = MSH263.DLL

También está basado en un estándar de la ITU que mejora la H.261. Será analizado con detalle en otros capítulos.

MPEG-4 (Versión 1, 2, 3 y DIVX)

[drivers32]
VIDC.???? = ????.DLL (depende del desarrollador)

El MPEG-4 es un estándar ISO que contempla procedimientos de compresión y manipulación de contenidos muy avanzados. Existen distintas versiones simplificadas de este estándar. Las más importantes son las versiones 1, 2 y 3 de Microsoft y el DIVX, que se ha popularizado como formato de alta compresión para películas de vídeo en calidad media.

Sorenson Video

Es un códec de baja velocidad disponible únicamente para QuickTime. Está considerado como uno de los que proporciona mejor calidad a tasas de transmisión muy bajas. Está basado en la codificación vectorial.

VxTreme

Es un códec basado en Wavelets y compensación de movimiento y tiene muy buenas prestaciones en aplicaciones de videoconferencia con movimientos lentos. La empresa fue adquirida por Microsoft en 1997 y se ha utilizado parte de su tecnología para las nuevas versiones de Windows Media.

Windows Media

Familia de códecs de Microsoft de excelente calidad y orientados a aplicaciones de transferencia de materiales multimedia a través de Internet. Disponen de varias tec-

nologías de compresión que se eligen en función de la tasa final de bits que desea obtenerse. Admiten la escalabilidad y pueden ser reproducidos en tiempo real durante el proceso de descarga. Analizaremos los principios de funcionamiento de estos sistemas en el módulo siguiente.

Resumen de tecnologías de compresión y códecs de vídeo

Finalmente, a modo de resumen proporcionamos un listado de las principales tecnologías de compresión de vídeo y de los códecs que las utilizan.

- **Run Lenght Encoding**: Microsoft RLE, Coeficientes de la DCT (H.261, H.263, ect)
- Codificación vectorial: Indeo 3.2, Cinepack, Sorenson.
- **DCT**: Motion JPEG, Editable JPEG, MPEG-1, MPEG-2, MPEG-4, H.261, H.263, H.263+, H.26L
- **Transformada Wavelet**: VDOWave, VxTreme, Intel Indeo 5
- **Codificación de imagen basada en identificación de contornos**: SFM (Crystall Net's). MPEG-4 sugiere ideas
- **Diferencia de cuadros**: Cinepack
- **Compensación de movimiento**: ClearVideo, RealVideo, VDOWave, VxTreme, MPEG's, H.26x.

Etapa 2: Estándares de compresión para videoconferencia

Introducción

Uno de los primeros problemas prácticos en los que se introdujeron los sistemas de compresión de vídeo digital fue la videoconferencia. En 1990, la ITU (International Telecomunications Union) presentó un conjunto de estándares para la transmisión de señales de videoconferencia orientados a redes de servicios digitales (red digital de servicios integrados –RDSI–). Los estándares contemplaban varias recomendaciones para la compresión de audio y vídeo, el envío de comandos de control y la multiplexación de todas estas componentes en una trama de datos. La parte de compresión de vídeo se conoce como el estándar H.261 y puede considerarse como el verdadero precursor de sistemas más avanzados como el MPEG-1 o el MPEG-2, ya que se introdujeron distintos conceptos para la extracción de la redundancia espacial y temporal que posteriormente se han utilizado en otros estándares de compresión de vídeo.

El estándar H.261 obtiene resultados aceptables para tasas de transmisión de datos propias de servicios digitales directos a través de redes de velocidad media-alta (redes de área local, Ethernet, ATM, RDSI de banda ancha), pero la calidad del vídeo se degrada de forma considerable cuando se pretende trabajar con canales de baja velocidad como redes telefónicas analógicas (máximo de 56 kbps). Los resultados obtenidos utilizando canales de acceso básico RDSI (128 kbps) son aceptables pero susceptibles de mejora. El ITU propuso la creación de un nuevo estándar que sustituyera al H.261 en aplicaciones con velocidades de transmisión bajas. El nuevo estándar se denominó H.263 e incorpora nuevos modos de compresión avanzados que permiten mejorar la calidad de imagen a velocidades bajas (en torno a los 64 kbps). El estándar H.263 se introdujo en 1995 y tuvo en cuenta toda la experiencia obtenida en el desarrollo de los estándares MPEG-1 y MPEG-2. El H.263 incorpora algunas técnicas de compresión y compensación de movimiento más avanzadas que las de estos últimos. Posteriormente, el estándar H.263 se ha mejorado para la codificación de vídeo en redes de muy baja velocidad, lo que ha dado lugar a una versión avanzada que recibe el nombre de H.263+. Existe una versión que se prevé que se apruebe durante el año 2002 que incorpora avanzados algoritmos de análisis de imagen que recibe el nombre de H.26L.

Los estándares para videoconferencia H.261, H.263 y H.263+ están también disponibles como codificadores de vídeo para aplicaciones multimedia en entornos de ordenador personal. Los códecs de estos estándares pueden instalarse en cualquier ordenador y utilizarlos para la transmisión de vídeo en tiempo real, para la creación de ficheros de vídeo (AVI, MOV) o incluso convertirlos a formatos de *streaming* (ASF) para su difusión en Internet.

En esta etapa veremos algunas de las características generales de estos compresores de vídeo desde el punto de vista de las tecnologías y los métodos de compresión que utilizan. Muchas de las estrategias son comunes con estándares que ya hemos analizado con detalle como el JPEG o los MPEG-1 y MPEG-2, por lo que sólo analizaremos las diferencias más significativas. El objetivo fundamental es proporcionar una idea básica de las capacidades y características de cada uno de los procedimientos sin entrar en los detalles concretos de cómo se realiza la codificación de los datos en una trama de bits.

El estándar H.261

La transferencia de señales de vídeo y audio en sistemas de videoconferencia impone ciertas restricciones en el proceso de codificación. Una de las más importantes es la necesidad que el proceso de codificación y decodificación pueda ser realizado en tiempo real y a un coste reducido. Además, el ancho de banda que se dispone en algunas redes de datos es limitado, de manera que es imprescindible utilizar técnicas de compresión de vídeo que aprovechen la redundancia temporal existente entre las diferentes imágenes de la secuencia.

El estándar H.261 se definió para poder trabajar con diferentes velocidades de transmisión que fueran múltiplos enteros de 64 kbps (kbits/s), y está especialmente diseñado para trabajar con la RDSI (red digital de servicios integrados) utilizando uno o más canales de acceso básico.

Las posibles velocidades de transmisión son p x 64kbps donde "p" es un número entero que puede tomar valores entre 1 y 30. Para el máximo p = 30 se obtiene una velocidad de transmisión de 1,92 Mbps que se corresponde con un acceso primario RDSI. La calidad proporcionada cuando se trabaja con este ancho de banda es excelente.

Las principales características del estándar desde el punto de vista de las tecnologías de compresión utilizadas se detallan en los siguientes subapartados.

Tamaño de las imágenes

Se puede trabajar con los modos CIF (*Common Intermediate Format*) y QCIF (*Quarter* CIF). El tamaño de la imagen CIF es de 352×288 (columnas por filas) y el QCIF tiene la mitad de filas y columnas (176×144), por lo que el número de píxeles de este formato es de una cuarta parte del formato CIF. El QCIF forma parte de los requisitos mínimos que debe cumplir cualquier descodificador o codificador H.261, aunque la mayoría de sistemas también implementan el formato CIF. Debe tenerse en cuenta que la resolución del CIF es una cuarta parte de la del estándar de TV digital ITU-601 (la mitad de filas y la mitad de columnas), por lo que la calidad final de la imagen proporcionada por el H.261 es relativamente pobre. Como criterio orientativo de la

calidad obtenida puede tenerse en cuenta que el formato CIF presenta una calidad equivalente al sistema de reproducción de vídeo VHS.

Existe un tercer modo de imagen que permite la transmisión de gráficos estáticos (pensado para la visualización de documentos durante el transcurso de la videoconferencia). Este modo gráfico estático tiene una resolución doble al CIF (704×576 píxeles).

Modo de vídeo y número de imágenes por segundo

El H.261 siempre trabaja con modos de vídeo no entrelazados (progresivos). En este sentido es equivalente al MPEG-1 y las imágenes se toman generalmente sólo de uno de los dos campos que proporciona una cámara PAL o NTSC. Tened en cuenta que el número de líneas en el formato CIF es el mismo que el número de líneas de uno de los campos de la señal de vídeo analógica, por lo que la conversión puede ser directa. En el caso de trabajar con el formato QCIF sólo se muestrea una de cada dos líneas de uno de los campos. Previamente, la señal analógica debe ser filtrada paso bajo para evitar los posibles problemas de *aliasing*.

El número de imágenes por segundo cuando se utilizan cámaras PAL es de veinticinco. En el caso del sistema NTSC es de treinta imágenes por segundo. El H.261 permite trabajar con estas dos frecuencias de imagen. No obstante, debido a las bajas velocidades de transmisión con las que debe trabajar el sistema, es habitual reducir el número de imágenes por segundo, omitiendo la codificación de algunos *frames*. Es habitual utilizar entre diez y quince imágenes por segundo. Con ello se permite que el número de *frames* que hay que codificar sea menor, de manera que pueden dedicarse más bits a los *frames* que realmente se codifican, con lo que se obtiene una mejora considerable en la calidad de cada *frame*.

Tipos de imágenes comprimidas

Se utilizan dos tipos de imágenes que tienen características parecidas a las del MPEG-1: imágenes tipo **intra** (I) e imágenes tipo **inter** (P).

Las imágenes intra (I) están codificadas sin tener en cuenta otros *frames* anteriores y sólo se realiza una compresión espacial. La forma de codificar las imágenes I es totalmente equivalente a como se hace con MPEG-1 o JPEG. La imagen se divide en bloques de 8×8 píxeles a los que se aplica la transformada coseno. El resultado de la transformada coseno se cuantifica mediante tablas de cuantificación y los resultados se codifican mediante códigos de longitud variable que también están tabulados.

Las imágenes tipo P (predicción) se usan para extraer la redundancia temporal existente en la secuencia de vídeo. Las imágenes se predicen utilizando los *frames* anteriores que actúan como imágenes de referencia y se permite el envío de vectores de compensación de movimiento para mejorar la predicción. La diferencia entre los valores reales del *frame* y los valores de predicción obtenidos mediante la compensa-

ción de movimiento se transmite al receptor sólo para aquellos bloques en los que el error sea significativo. De esta forma, el receptor puede recomponer una imagen más exacta al original que la que puede obtenerse mediante la simple compensación de movimiento. La transmisión de los errores de predicción se realiza utilizando la transformada coseno que nuevamente se aplica a bloques de 8×8 píxeles. Los coeficientes transformados del error se cuantifican y se codifican mediante códigos de longitud variable.

Diferencias entre el H.261 y el MPEG-1

Las diferencias más significativas de todo el proceso de codificación H.261 con respecto al MPEG-1 son:

- Las imágenes tipo B no se utilizan en el H.261.

- En el H.261 los vectores de compensación de movimiento sólo se calculan con la precisión de 1 píxel (en los estándares MPEG-1 y MPEG-2 es posible realizar la compensación de movimiento con vectores de precisión de medio píxel).

Debe tenerse en cuenta que este estándar es anterior a los MPEG y que durante su definición era preciso garantizar que todo el proceso de codificación y descodificación debía poder realizarse en tiempo real mediante *software* genérico que se ejecuta en un ordenador personal o mediante un *hardware* específico de bajo coste. Actualmente, tanto los ordenadores personales como los circuitos integrados específicos permiten realizar operaciones más complejas, de modo que el estándar más actual H.263 ya incluye modos avanzados de compensación de movimiento bidireccional. No obstante, el H.261 continúa teniendo interés tanto desde el punto de vista histórico como desde el punto de vista práctico, puesto que existen numerosas aplicaciones y estándares de videoconferencia que lo utilizan. Además, cuando las tasas de transmisión de datos son elevadas (superiores a 1,5 Mbps) consigue excelentes resultados.

Como todos los estándares de compresión de vídeo, el H.261 sólo especifica las posibles técnicas que pueden emplearse en la codificación y la estructura sintáctica y semántica de la trama de bits que proporciona la información de vídeo (cómo deben insertarse los bits dentro de la trama y cómo deben ser interpretados por el descodificador). Las decisiones de cómo debe codificarse un bloque determinado debe tomarlas el propio codificador e informar al descodificador para que pueda recomponer correctamente la secuencia de imagen. Esto significa que la carga computacional del codificador es mucho más elevada que la del descodificador (algoritmos de compresión no simétricos). Los algoritmos para decidir si un determinado bloque de imagen debe codificarse como intra o utilizando la compensación de movimiento, los algoritmos para determinar los vectores de movimiento, las decisiones de si resulta conveniente transmitir el error de predicción o no, etc. no están especificadas por el estándar. En consecuencia, es posible que dos codificadores que cumplen con la sintaxis del H.261 presenten resultados muy distintos en función de los algoritmos que tienen implementados para tomar las distintas decisiones. Esta filosofía de especifi-

cación del estándar se ha mantenido para todos los compresores posteriores (MPEG-1, MPEG-2, H.263, H.263+, MPEG-4, etc) y seguramente se seguirá manteniendo en el futuro. El éxito principal es que el estándar está abierto a distintas implementaciones, de forma que los desarrolladores pueden competir para conseguir codificadores que, utilizando las reglas definidas por el estándar, proporcionen una mejor calidad de imagen y unos factores de compresión más elevados que los productos de sus competidores.

El estándar H.263

Este estándar es el resultado de aplicar varias mejoras parciales sobre el estándar H.261. Estas modificaciones, consideradas de forma conjunta, permiten obtener unos factores de compresión y calidad de vídeo muy significativos. El estándar fue propuesto en 1995, por lo que es posterior al MPEG-1 y simultáneo con el MPEG-2; incorpora distintas opciones y herramientas de codificación comunes con estos dos.

El objetivo principal del H.263 era obtener un sistema que permitiera la compresión de vídeo utilizando líneas de transmisión de datos de baja velocidad, principalmente la RDSI de banda estrecha y las comunicaciones digitales a través de redes telefónicas analógicas (POTS –*Plain Old Telephone System*–). Este objetivo significa que el flujo máximo de bits debe estar situado sobre los 56 kbps y los 64 kbps, lo cual sólo es posible cuando se utilizan imágenes de reducido tamaño (formatos más pequeños que el QCIF) y se reduce el número de imágenes por segundo.

No obstante, debido a las excelentes prestaciones del algoritmo, el H.263 también se utiliza para codificar vídeo a más altas resoluciones, con lo que se obtienen buenas relaciones entre calidad y factor de compresión. Existen varias versiones de códecs para PC basados en el H.263 que a menudo se utilizan para obtener ficheros AVI con resoluciones de pantalla completa. En la mayoría de los casos la compresión no puede realizarse en tiempo real, pero sí la descodificación.

Una de las potenciales ventajas de trabajar con imágenes de baja resolución y un menor número de *frames* es que el número total de bloques por segundo que deben procesarse disminuye considerablemente. Esto permite que el procesador pueda realizar algunas operaciones y cálculos adicionales que no son posibles en estándares como el MPEG-2.

En las siguientes secciones se examinan los distintos modos de codificación que proporciona el H.263. Únicamente se discuten las diferencias más significativas desde un punto de vista sistemático con respecto a los estándares H.261 y MPEG-1. Los detalles concretos de la codificación de los vectores de movimiento y bloques de imagen se omiten, puesto que sólo resultan de interés para los desarrolladores.

Tamaños de imagen

En principio el H.263 está orientado a los mismos tamaños de imagen que el H.261, es decir, los formatos QCIF y opcionalmente el QCIF. No obstante, para facilitar que pueda trabajarse con tasas de bits más bajas se introdujo un nuevo subformato, de menor tamaño, denominado sub-QCIF. Además, las buenas prestaciones del algoritmo para trabajar con imágenes de mayor tamaño (aunque en la mayoría de las implementaciones no sea en tiempo real) facilitó la introducción de tamaños mayores (de pantalla completa) dentro del estándar, aunque sólo de forma opcional.

En la siguiente tabla se proporciona un resumen de los tamaños de imagen que se contemplan en el estándar. Los campos obligatorios significan que cualquier descodificador compatible con el estándar debe incluirlos. En todos los casos sólo se proporciona el tamaño de la imagen en píxeles de luminancia. Ambos estándares utilizan el formato 4:2:0, por lo que las componentes de croma tienen la mitad de filas y columnas que las de luminancia.

Tabla de tamaños de imagen contemplados en los estándares de videoconferencia H.261 y H.261

Formato de imagen	Píxeles luminancia	Compatibilidad H.261	Compatibilidad H.263
sub-QCIF	128×96	No definido	Obligatorio
QCIF	176×144	Obligatorio	Obligatorio
CIF	352×288	Opcional	Opcional
4CIF	704×576	No definido	Opcional
16CIF	1408×1152	No definido	Opcional

Tipos de imágenes

El estándar H.263, como el MPEG-1, contempla imágenes de tipo I, tipo P y tipo B en su modo básico (compensación de movimiento bidireccional). La inclusión de imágenes del tipo B permite una mayor capacidad de compresión de la secuencia de vídeo.

También están definidos modos opcionales dentro del H.263 que utilizan técnicas de compresión más avanzadas. En los modos opcionales, el H.263 puede utilizar imágenes de un nuevo tipo denominado PB que se analizará brevemente en secciones posteriores.

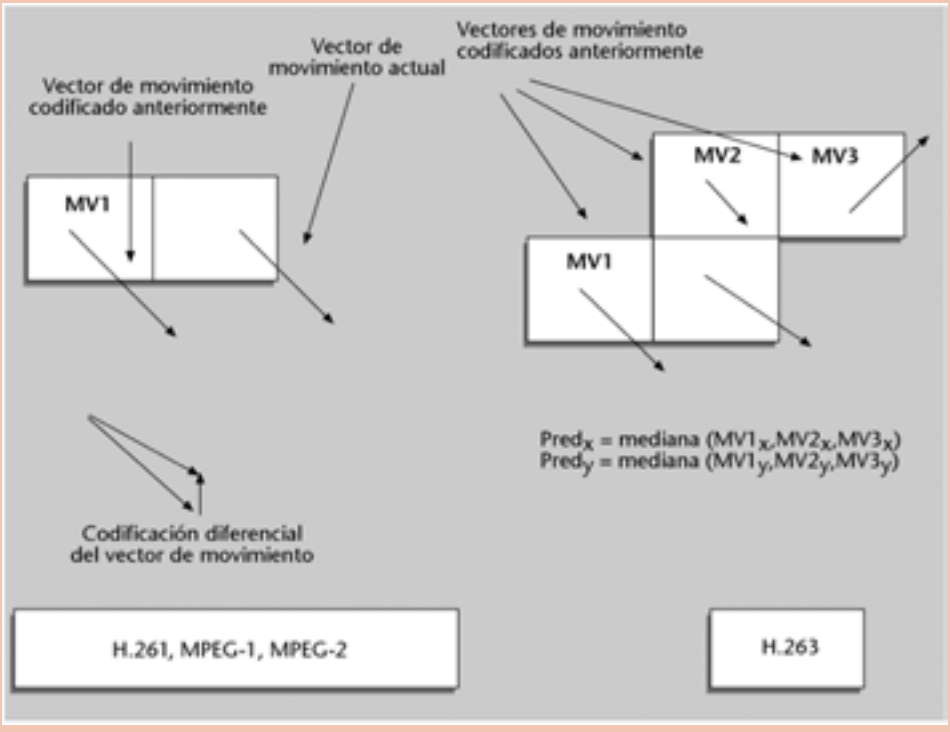
Precisión de los vectores de movimiento

La precisión con la que se realizan las estimaciones de compensación de movimiento en el H.263 es de medio píxel. Esta característica coincide con la del MPEG-1 y representa una mejora con respecto al H.261, en el que la precisión en la compensación de movimiento estaba limitada a un píxel.

Codificación diferencial del vector de movimiento

La codificación del vector de movimiento en los estándares H.261, MPEG-1 y MPEG-2 siempre es diferencial con respecto al vector de movimiento del macrobloque transmitido o codificado anteriormente. La ventaja de utilizar una codificación diferencial es que los vectores de movimiento entre macrobloques próximos suelen ser muy parecidos. De este modo, si sólo enviamos las diferencias con respecto al vector anterior, los valores serán, con una gran probabilidad, muy próximos a cero. Al utilizar códigos de longitud variable para codificar los vectores de movimiento se obtiene una reducción considerable del número de bits necesarios gracias a la codificación diferencial.

El codificador H.263 añade una mejora adicional a la codificación diferencial de los vectores de movimiento al permitir que puedan utilizarse como vectores de predicción no sólo los del macrobloque anterior, sino los de otros macrobloques situados en la proximidad del que actualmente se está codificando. La idea general se representa en la figura en la que se compara la codificación tradicional utilizada en el H.261 y MPEG con la del H.263.



Codificación diferencial del vector de movimiento en el H.263 comparado con H.261 y MPEG

La diferencia básica es que la predicción del vector de movimiento se realiza teniendo en cuenta los vectores de movimiento de los macrobloques situados a la izquierda, arriba y arriba a la derecha. Todos estos macrobloques ya habrán sido codificados anteriormente, por lo que sus vectores de movimiento se pueden aprovechar para mejorar la estimación del vector de movimiento actual.

La predicción de las componentes x e y del vector de movimiento se realiza ordenando las tres componentes de cada uno de los vectores y seleccionando la central (filtro de mediana). Estadísticamente, la predicción obtenida con esta estrategia es mejor que la que se obtiene con el método clásico que se utiliza en el MPEG-1 o MPEG-2, ya que se tiene en cuenta un mayor volumen de información. Considerad como ejemplo una región de la imagen en la que el vector de movimiento del macrobloque de la izquierda tiene una dirección completamente distinta de la del macrobloque actual. En esta situación, la codificación diferencial clásica producirá un vector de movimiento con un módulo elevado que requerirá muchos bits al codificarlo mediante los códigos de Huffman. En cambio, la estrategia utilizada por el H.263 puede producir todavía una buena estimación si los vectores de movimiento de los dos bloques que están situados en la fila anterior están correlados con el vector de movimiento actual.

Cuando los macrobloques de referencia se codifican en el modo intra (no se utiliza compensación de movimiento), se supone que los vectores de movimiento asociados a estos macrobloques tienen componentes nulas. Esta suposición es idéntica a la que se utiliza en el resto de los codificadores. Además, cuando los macrobloques de referencia están situados fuera de los límites de la imagen, también se supone que los vectores de movimiento serán nulos. Así, cuando se codifican los macrobloques situados en el límite izquierdo de la imagen MV1 serán nulos. De forma equivalente, cuando se codifican los del límite derecho MV3 se supondrá nulo. Cuando se codifica la primera fila de la imagen, son MV2 y MV3 los que se suponen nulos.

Es importante observar que esta estrategia de codificación de los vectores de movimiento introduce desde el punto de vista estadístico una pequeña reducción de la tasa de bits asignada a los vectores de movimiento. H.263 consigue obtener una mejora considerable en el factor de compresión debido al efecto combinado de un elevado número de optimizaciones pequeñas. Esta técnica de compresión es un claro ejemplo del tipo de mejoras parciales que introduce este estándar. El coste adicional es que tanto el compresor como el descompresor deben mantener un *buffer* de memoria con varios vectores de movimiento anteriores y deben de realizar la operación de ordenación de los tres vectores de referencia.

Estimación de vectores de movimiento no restringida (modo opcional)

En el modo básico del H.263, todas las estimaciones de los vectores de movimiento están restringidas a que los píxeles de referencia estén siempre dentro de la imagen. En este modo opcional se permite que los vectores de movimiento tomen valores en los que parte de los píxeles del macrobloque queden fuera de los límites de la imagen. Los píxeles que se sitúan en el exterior de la imagen se aproximan por los del límite de la imagen. La posibilidad de utilizar vectores de movimiento no restringida introduce una cierta reducción del error de predicción de los píxeles en algunos casos.

Codificación aritmética (modo opcional)

En el modo opcional, el H.263 permite sustituir el uso de códigos de longitud variable (Huffman) por codificadores aritméticos, de mayor complejidad, pero con una

mejor eficiencia en la codificación de datos. Los resultados experimentales sobre la codificación de Huffman y la aritmética establecen que puede conseguirse entre un 5% y un 10% de reducción del flujo de datos cuando se utiliza esta última estrategia.

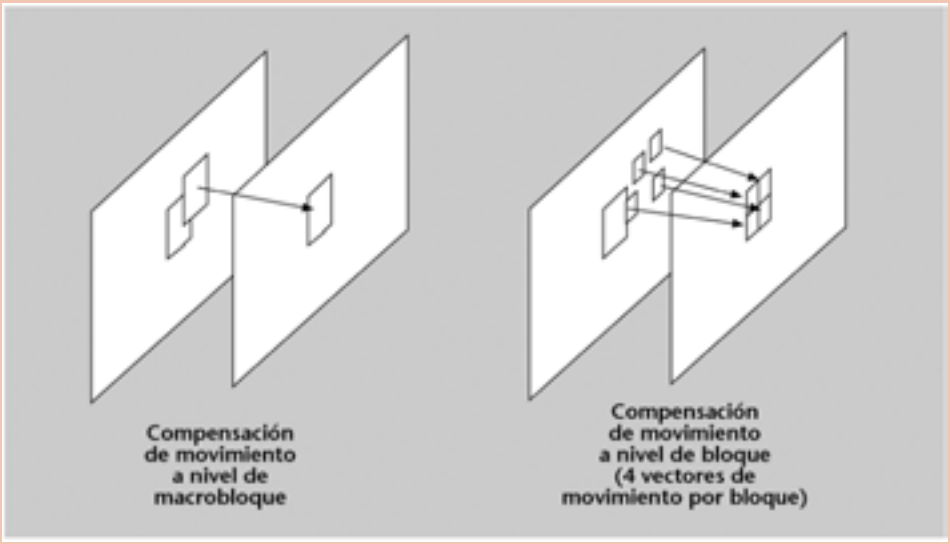
Modo de predicción avanzado (modo opcional)

El H.263 contempla un modo de predicción avanzado que permite el uso de hasta cuatro vectores de movimiento para cada macrobloque. Además, la compensación de movimiento puede realizarse solapando diferentes bloques de las imágenes de referencia.

Cuatro vectores de movimiento por macrobloque

En el modo de predicción avanzado, cada macrobloque se divide en cuatro bloques de 8×8 píxeles, de manera que se determina un vector de movimiento diferente para cada uno de los bloques. El uso de vectores de movimiento en el bloque (y no en el macrobloque como en el resto de los estándares estudiados hasta este momento) permite obtener una mayor precisión en la compensación de movimiento, lo que mejora la predicción final. La mejora es especialmente importante cuando existen objetos de pequeño tamaño que se desplazan en direcciones opuestas o con movimientos aleatorios. El coste adicional es un aumento considerable de la carga computacional asociada al cálculo de los vectores de movimiento. Además, el compresor debe considerar la decisión de codificar el macrobloque con un único vector de movimiento global o cada uno de los cuatro bloques con vectores de movimiento independientes. Generalmente, esta última estrategia produce un error menor de predicción de la imagen, con lo que se requieren menos bits para codificarla, pero también requiere que se codifiquen los tres vectores de movimiento adicionales, circunstancia que aumenta el número de bits dedicado a los vectores de movimiento. Como siempre, los algoritmos de decisión del modo de codificación no son considerados por el estándar, que deja que sea el desarrollador del producto quien decida los parámetros con los que se selecciona el modo de codificación.

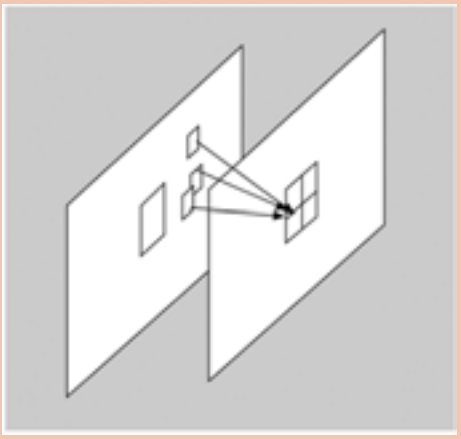
Los vectores de movimiento asociados a cada uno de los bloques también se codifican de forma diferencial teniendo en cuenta tres vectores de movimiento de bloques que ya han sido previamente codificados, de forma equivalente a como se realiza en el modo básico del H.263 que ya hemos descrito anteriormente. Los vectores de movimiento que se utilizan de referencia dependen de la posición del bloque dentro del macrobloque, y aunque omitimos aquí los detalles concretos de cuáles se utilizan en cada caso, la filosofía general es la misma que la utilizada en el modo básico.



Comparación entre la compensación de movimiento en el nivel de macrobloque y en el de bloque

Compensación de movimiento con solapamiento entre bloques

Un modo opcional adicional que permite el H.263 es que la predicción de un bloque se realice teniendo en cuenta no sólo su propio vector de movimiento, sino también los vectores de movimiento de dos de sus bloques vecinos. Este modo sólo se utiliza cuando se realiza la compensación de movimiento en el nivel de bloque. Aunque se omiten los detalles exactos de este modo de codificación, la idea general se representa en la siguiente figura.



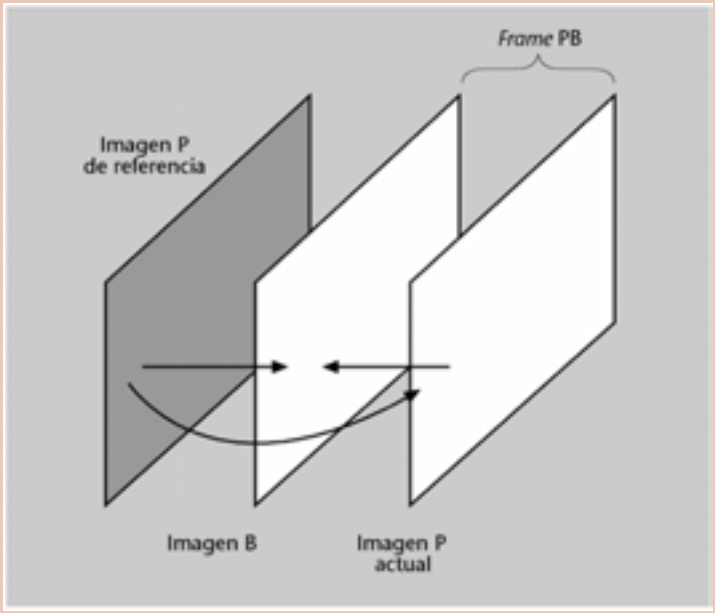
Compensación solapando distintos vectores de movimiento

La predicción de los píxeles del bloque actual se obtiene combinando tres bloques de la imagen de referencia. Uno de los bloques corresponde con el que se obtiene al utilizar el vector de movimiento asociado al bloque que estamos codificando. Los otros dos bloques se obtienen a partir de los vectores de movimiento que ya se han calculado para dos de los bloques vecinos (los que se considere que están más correlados). La predicción final se obtiene ponderando los tres bloques de la imagen de referencia, aunque otorgando un mayor peso al bloque obtenido a partir del vector de movimiento del que se está codificando.

La ventaja principal que proporciona esta estrategia de codificación es que, al utilizar distintas zonas de la imagen de referencia solapadas, disminuye notablemente el efecto de bloque que se observa cuando se utilizan factores de compresión elevados.

Imágenes tipo PB (modo opcional)

El estándar contempla un nuevo tipo de imágenes que combina los conceptos de imágenes P e imágenes B del MPEG-1, pero que son codificadas como un único bloque o tipo de imagen. En la siguiente figura se representa la estructura básica de este tipo de imagen.



Estructura general del modo *frame PB*

La diferencia fundamental de este modo es que la unidad de codificación es un macrobloque que contiene información combinada sobre la imagen P y la B. El macrobloque combinado proporciona información de los bloques de la imagen P y de los bloques de la imagen B. En primer lugar, se envía la información correspondiente a los bloques de la imagen P, que se corresponden con el formato utilizado en el modo por defecto del H.263, y posteriormente se envía la información correspondiente a los bloques B. El codificador puede definir distintos modos de interpretar la información, pero en cualquier caso, la ventaja principal es que los vectores de movimiento *forward* y *backward* para los macrobloques B pueden deducirse de los vectores transmitidos para la parte P. Esto generalmente representa una reducción significativa del flujo de datos asociado a la transmisión del vídeo debido a la reducción de carga en el envío de vectores de movimiento.

Comparativa entre H.261, MPEG-1 y H.263

En la tabla siguiente se representan los valores de PSNR obtenidos al codificar una secuencia de vídeo de tamaño CIF con cuatro configuraciones distintas de estándares

de compresión. En todos los casos, la tasa de compresión de vídeo se ha fijado a 256 kbps, por lo que la PSNR da una idea global de la calidad relativa que es posible obtener con cada codificador. Los codificadores utilizados son:

- H.261
- MPEG-1 con una estructura de GOP de doce imágenes con la secuencia IBBPBBPBBPBB. (MPEG-1 (a))
- MPEG-1 con una estructura de GOP IPPPPPPP... (MPEG-1(b))
- H.263 utilizando el modo avanzado pero sin el codificador aritmético.

Los valores de PSNR de la tabla son el resultado de promediar la PSNR medida en cada uno de los *frames* de la secuencia de vídeo.

Tabla comparativa de prestaciones de compresores

Compresor	PSNR media
H.261	38 dB
MPEG-1 (a)	37,2 dB
MPEG-1 (b)	40,1 dB
H.263	43,6 dB

Los resultados obtenidos muestran que para esta tasa de compresión el estándar H.263 presenta los mejores resultados de todos los métodos comparados.

Es interesante observar que el MPEG-1 (a) presenta los peores resultados de todos los métodos comparados. La principal razón de esta baja calidad es que se invierten prácticamente todos los bits en la codificación de las imágenes tipo I, y queda un número de bits muy reducido para la codificación de las imágenes P y B. Por este motivo, la calidad que se consigue en estos *frames* es muy baja. Si la tasa de bits objetivo estuviera en torno a los 1,5 Mbps, este método sería uno de los que ofrecería mejores resultados.

Otro resultado de interés es que el H.261 y el MPEG-1 (b) utilizan prácticamente la misma tecnología de compresión (no están definidas imágenes tipo B en este modo del MPEG-1). La única diferencia importante es que el MPEG-1 (b) utiliza la compensación de movimiento con una precisión de medio píxel, lo que se traduce en una mejora de la codificación de unos 2 dB.

La mejora del H.263 con respecto al MPEG-1 (b) se debe fundamentalmente al uso de los modos de codificación avanzados, que representan una mejora de aproximadamente 3,5 dB. Si el H.263 hubiera utilizado el modo de codificación aritmético, la mejora hubiera sido aún superior (entre 1 dB y 2 dB adicionales).

El estándar H.263+ (H.263 versión 2)

El H.263+ es una nueva versión del H.263 que incorpora nuevas características y métodos de compresión avanzados adicionales. Los nuevos modos (hasta un total de doce) introducen mejoras considerables en la tasa de compresión y permiten el uso de *bitstreams* escalables. Además, está especialmente diseñado para facilitar su uso en redes de conmutación de paquetes (tipo Internet). Los tamaños de imagen son libres y no están restringidos a los modos impuestos en el H.263. La nueva versión es totalmente compatible para poder descodificar secuencias comprimidas con el H.263.

El estándar está pensado para poder trabajar con tasas de bits inferiores a los 64 kbps. El objetivo principal es conseguir una codificación de vídeo aceptable con tasas próximas o incluso por debajo de los 24 kbps. Por este motivo, puede utilizarse en redes de comunicación de datos de muy baja velocidad como redes de telefonía móvil digital (GSM, UMTS). El espectro de aplicación es relativamente extenso y puede utilizarse con imágenes de cierta calidad con velocidades de transmisión más altas.

El H.263+ mantiene todos los modos de codificación avanzados que se introducen en el H.263 e incorpora además nuevas tecnologías de compresión y métodos para ocultar los errores. Las principales técnicas avanzadas que se introducen en el H.263+ son:

- Nuevo modo de predicción en imágenes intra. Se permiten utilizar bloques adyacentes al bloque que se está codificando como una estimación del bloque actual (predicción) en imágenes del tipo intra. Esta estrategia es totalmente nueva y resulta útil en zonas de la imagen que sean muy uniformes o que tengan una textura parecida.

- Para los bloques en los que se utiliza la predicción intra se permiten dos formas distintas adicionales de realizar la exploración de los coeficientes transformados de la DCT. Puede utilizarse el método en zigzag convencional o dos variantes (una horizontal y otra vertical) del método de exploración alternado que también se utiliza en el MPEG-2 para bloques entrelazados.

- Posibilidad de conmutar las tablas de Huffman para la codificación de los coeficientes correspondientes a coeficientes intra y coeficientes inter.

- Mejora del modo de *frames* PB con respecto al H.263. La principal diferencia es que la primera versión sólo permite utilizar la compensación bidireccional para la parte B de la imagen. El H.263+ permite realizar una compensación de movimiento *forward*, *backward* o bidireccional.

- Modo de selección de la imagen de referencia. En el H.263 se utiliza siempre la imagen anterior para realizar la predicción de la imagen actual. Si debido a la aparición de errores de canal se produce la pérdida de datos, la calidad de las imáge-

nes posteriores se degrada de forma considerable. Con este modo es posible seleccionar la imagen de referencia utilizada para realizar la predicción y suprimir la propagación temporal de los errores. El descodificador tiene que disponer de múltiples *buffers* de memoria para almacenar todas las posibles imágenes de referencia. La información que indica cuál es la imagen de referencia que se está utilizando en cada momento la proporciona el codificador dentro del mismo *bitstream*. Si el codificador y el descodificador están conectados por un canal de control es posible que el segundo comunique al primero la presencia de errores de forma que el codificador modifique las imágenes que está utilizando como referencia.

- Filtrado de *blocking*. Se aplican filtros no lineales en los límites de los bloques para reducir los molestos efectos visuales que se producen en tasas de codificación muy bajas cuando aparece el efecto de la pobre cuantificación de la transformada coseno (aparición de bloques).

- Es posible incorporar códigos correctores de errores en aplicaciones en las que no se incluya la protección de errores en protocolos de comunicación o en las que no resulte suficiente. Esta alternativa está especialmente pensada para comunicaciones móviles en las que la tasa de errores puede ser muy elevada.

- Se introduce el concepto de escalabilidad espacial, escalabilidad SNR y escalabilidad temporal ya utilizado en el MPEG-2. La señal de vídeo puede transmitirse en varias capas que permiten recuperar la señal con diferentes calidades en función del estado de la red. Los conceptos de escalabilidad espacial y escalabilidad SNR son parecidos a los que ya hemos considerado para el JPEG2000. La escalabilidad temporal permite la introducción de imágenes adicionales para mejorar la tasa de refresco de la señal de vídeo. La capa base contiene un número reducido de imágenes por segundo, mientras que el *bitstream* adicional contiene imágenes tipo B suplementarias que pueden descodificarse si se desea. Al tratarse de imágenes del tipo B no contienen información de referencia que resulte necesaria para la codificación de nuevas imágenes. Observad que la inclusión de imágenes de tipo B exige una mayor carga computacional al descodificador y un mayor retardo en la reproducción, por lo que es posible que en algunas implementaciones se descarte esta información auxiliar.

El estándar H26L

Es el estándar de videoconferencia más reciente (se prevé su aprobación final durante el año 2002) y está pensado específicamente para comunicaciones móviles con tasas de transferencia muy bajas. Está basado en las mismas tecnologías que el H.263, H.263+ y el MPEG-4, que combinan la compensación de movimiento con la transformada coseno. Se prevé que este estándar sustituirá los actuales H.263 y H.263+ en un futuro próximo.

Las principales características adicionales que contempla el H26L son:

- Precisión de un cuarto de píxel en la estimación de los vectores de compensación de movimiento.

- Diferentes tamaños de bloque para la realización de la compensación de movimiento: 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 y 4×4.

El H26L tiene una carga computacional media situada en torno a las 3,8 veces la carga asociada al H.263 (versión 2). La mejora en la tasa de compresión es de aproximadamente un 25% para una calidad de vídeo equivalente. El algoritmo puede implementarse en tiempo real mediante un procesador genérico (Pentium III, 733 MHz).

Etapa 3: El estándar MPEG-4

Introducción

Una de las características del desarrollo tecnológico de los sistemas de comunicación actuales es que el usuario dispone de un gran número de posibilidades técnicas para acceder a la información audiovisual: radiodifusión de señales de televisión y radio, DVD, vídeo en CD-ROM, Internet, redes locales corporativas, sistemas de vídeo y audio *on demand* en redes locales, telefonía móvil, etc. Cada uno de estos sistemas de comunicación tiene sus propias particularidades y las velocidades de transmisión de datos que pueden obtenerse de ellas son muy diferentes. Además, algunos de estos sistemas tienen enlaces bidireccionales, lo que en principio debería permitir que el usuario pudiera interactuar de algún modo con la información audiovisual.

Antes de abordar la elaboración del estándar MPEG-4, el grupo MPEG había definido con éxito los estándares MPEG-1 y MPEG-2 que estaban orientados al registro de vídeo en CD-ROM y a la televisión digital, respectivamente. No obstante, estos dos estándares no resultaban lo bastante flexibles como para cubrir todo el margen de posibilidades que ofrecen los sistemas de comunicaciones ni preveían ningún tipo de interacción con los contenidos por parte del usuario. El estándar MPEG-4 surge de esta necesidad e intenta proporcionar suficiente flexibilidad como para integrar los aspectos de producción, distribución y acceso a los contenidos en los siguientes campos:

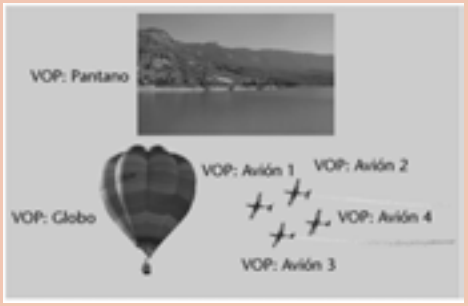
- Televisión digital
- Aplicaciones con gráficos interactivos (imágenes sintéticas)
- Distribución y acceso a contenidos multimedia interactivos (Wide World Web, móviles, etc.)

La primera versión del MPEG-4 aparece en 1999 y representa un cambio radical en el enfoque del problema de la codificación con respecto a los estándares precedentes. El nuevo enfoque está basado en la codificación de los contenidos que representan la escena. Esto significa que la escena se interpreta como un conjunto de objetos que se codifican de forma independiente y que posteriormente se describen sus relaciones. Los objetos pueden ser de vídeo, gráficos, audio, etc. Esta representación de la escena en objetos facilita la posible interactividad del usuario con los contenidos. El estándar es muy amplio e intenta cubrir aplicaciones muy distintas, por lo que, igual que con el MPEG-2, se definen distintos niveles y perfiles para adaptarlo a las necesidades. En los siguientes apartados examinaremos las características básicas de la parte de vídeo. El estándar también incluye herramientas para la codificación de audio en forma de objetos, la descripción de los contenidos y la multiplexación de los objetos en tramas de información cuyos detalles no consideraremos.

Descomposición en objetos

La codificación de vídeo en el estándar MPEG-4 se define como planos de objetos de vídeo (VOP: *Video Object Plane*). Cada VOP representa un objeto de interés que se codificará de forma independiente y que, si el autor así lo desea, el usuario podrá interactuar con él. En la figura adjunta se representa una imagen en la que está definido un fondo y dos objetos de interés. Los objetos de interés (VOP) son el globo y los cuatro aviones que están realizando la demostración aérea. El fondo es un pantano que también puede ser considerado como un VOP independiente. Los seis VOP se envían al usuario multiplexados y si el autor lo permite, el usuario podrá manipular los contenidos (modificando las posiciones de los objetos, modificando su escala, activando que sean visibles o resulten invisibles, etc.). Los VOP por separado se representan en una figura aparte.

Imagen obtenida como la combinación de seis VOPS: el fondo, el globo y cada uno de los aviones



Codificación de los VOP como objetos independientes

Segmentación de objetos

Uno de los problemas previos a la codificación es el análisis de las imágenes y la segmentación de los objetos que la forman. En el caso de imágenes sintéticas, el problema puede ser relativamente simple en función de las características del medio con el que se hayan generado los objetos. En algunos casos, las imágenes sintéticas pueden generarse directamente como objetos y capas de vídeo independientes, de forma que resultan más o menos simples de manipular como elementos independientes.

Sin embargo, cuando las imágenes son naturales, el problema de la segmentación es especialmente complejo; por el momento no se dispone de métodos automáticos que garanticen una segmentación automática eficiente de una imagen. De hecho, la propia naturaleza de los VOP, definidos como objetos de interés en la imagen, nos indica que es necesario algún tipo de participación humana para definir los elementos básicos en los que debe segmentarse la imagen. En algunos casos, durante la producción del material o vídeo es posible realizar segmentaciones de los objetos asistidas por ordenador de forma más o menos fiables. Básicamente, el proceso consiste en que el operador defina un objeto dentro de la imagen a partir de una segmentación previa y que posteriormente un sistema de seguimiento automático obtenga la forma aproximada del objeto en los *frames* sucesivos.

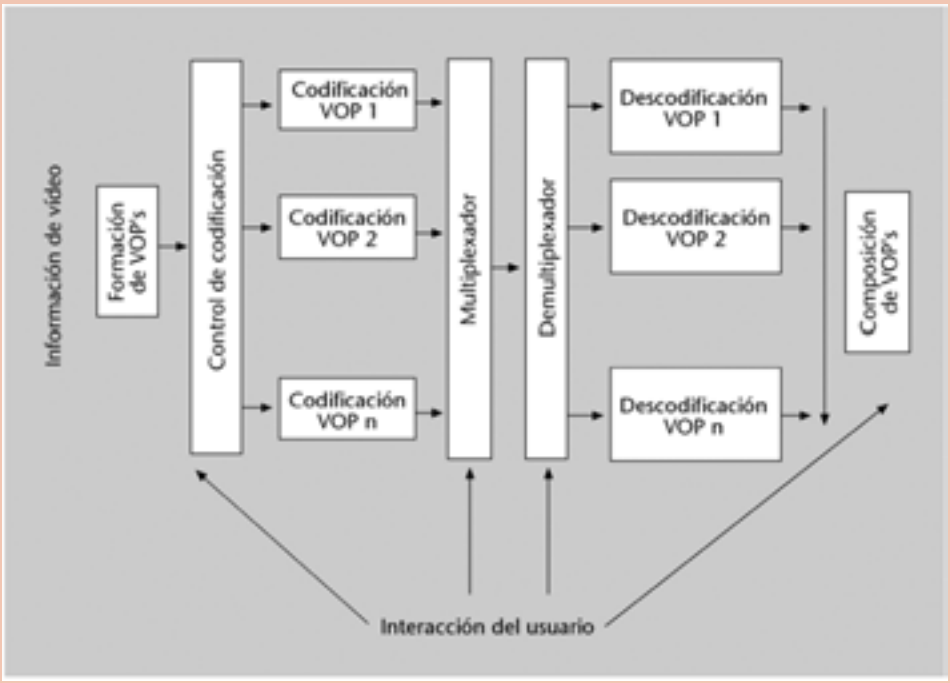
En algunos casos especiales es posible simplificar el proceso de segmentación registrando los objetos por separado, mediante el uso de técnicas como el *croma key*.

Es importante insistir de nuevo en que, como el resto de los estándares MPEG, el MPEG-4 sólo define la sintaxis, la semántica y las herramientas disponibles para codificar objetos. El problema de la segmentación y definición de los VOP se deja para que cada desarrollador lo elabore como desee.

Finalmente, comentaremos que es posible transmitir vídeo con un único objeto con forma rectangular, que no es más que toda la imagen completa. De hecho, ésta es la opción que implementan la mayor parte de los codificadores MPEG-4 comercializados actualmente (MPEG4 versión 1, versión 2 y versión 3 de Microsoft, DIVX). El MPEG-4 dispone de muchísimas herramientas y estrategias de codificación y, aunque su filosofía de base esté basada en el uso de objetos, también es muy eficiente cuando se define un único objeto rectangular. Pensemos además que en muchos casos el autor no está interesado en permitir que el espectador pueda interactuar con el material multimedia. Es este caso, aunque la codificación mediante VOP sigue teniendo sentido, no es especialmente importante disponer de herramientas o tiempo para la segmentación.

Descomposición en VOP y multiplexación

En la figura se muestra un diagrama de bloques de la estructura del codificador y del descodificador. A partir de la información de vídeo debe realizarse la descomposición en VOP del contenido de la imagen y codificar cada uno de los elementos por separado. Cada objeto de plano de vídeo es una secuencia de vídeo propia, con movimiento incorporado que debe ser codificada de forma eficiente. Las técnicas básicas para realizar la codificación son parecidas a las que se utilizan en el MPEG-1 o el H.263. Están basadas en la transformada coseno y la codificación de los coeficientes resultantes. Cada VOP tiene asociado un *stream* elemental que proporciona información de cómo evoluciona este objeto a lo largo del tiempo. Los *streams* elementales se multiplexan en un único *stream* que contiene la información de todos los VOP.



Estructura de descomposición de la información de vídeo en VOP: codificación, descodificación y multiplexación.

En el descodificador el proceso es inverso. La trama que contiene todos los objetos de vídeo se descompone en las tramas asociadas a cada uno de los VOP originales, cuyas secuencias de vídeo se reconstruyen formando elementos independientes que finalmente se combinan en una única secuencia de vídeo.

El usuario receptor puede controlar la interacción entre los objetos visuales. Para ello, el transmisor proporciona un *stream* adicional que define la relación entre los diferentes objetos y que se utiliza como descriptor básico de la escena. El operador del transmisor debe definir las relaciones entre los objetos y decidir el grado de interactividad que permite al receptor.

Descripción de la escena

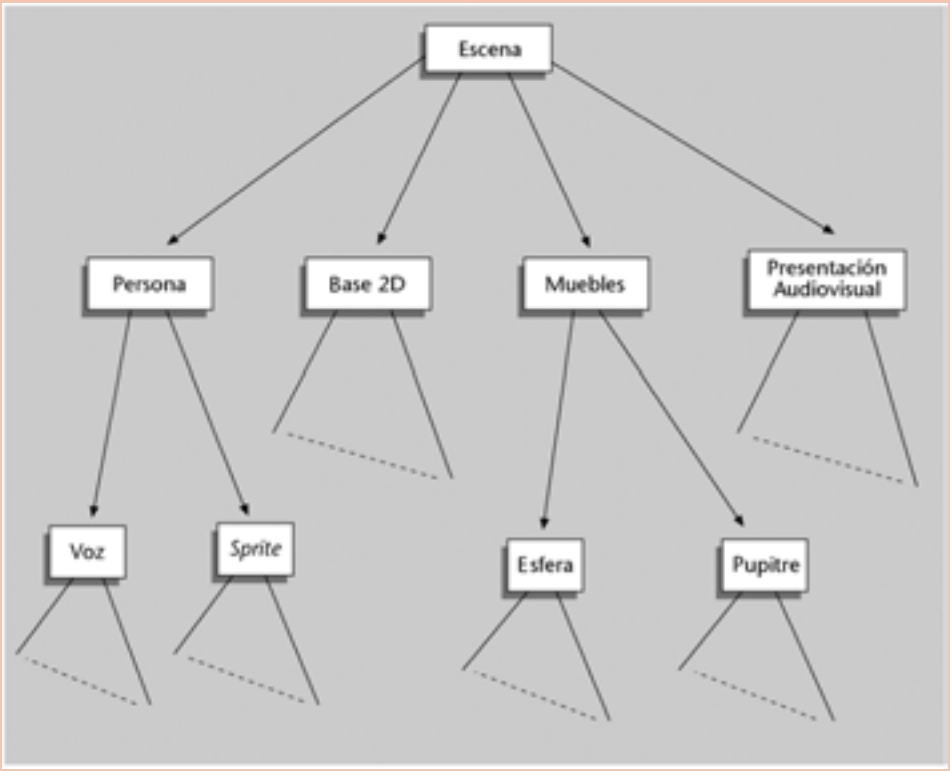
La descripción de una escena en MPEG-4 tiene una estructura jerárquica en la que los objetos quedan representados como un grafo en forma de árbol. Cada nodo del árbol es un objeto. El MPEG ha desarrollado un lenguaje binario para describir la escena denominado BIFS (*Binary Format for Scenes*).

Para facilitar el desarrollo de herramientas de autor, manipulación e interacción, una parte de la información correspondiente a los objetos se codifica en la parte de la descripción de la escena. Así, todos aquellos parámetros con los que el usuario podría interactuar se codifican en la parte de descripción de la escena y no en el *stream* asociado al objeto. Un ejemplo sería la codificación de la posición del objeto en la pantalla. El usuario puede interactuar modificando su posición, por lo que es conveniente que esta información esté incluida en el descriptor de la escena y no en el *stream* de datos asociado al objeto.

La estructura no es necesariamente estática, sino que pueden irse modificando los parámetros de posición, los atributos de los nodos y pueden añadirse o eliminarse algunos nodos.

En las siguientes gráficas se muestra una escena ejemplo, compuesta por múltiples objetos, y la forma que tendrá el grafo asociado a la descripción de la escena.

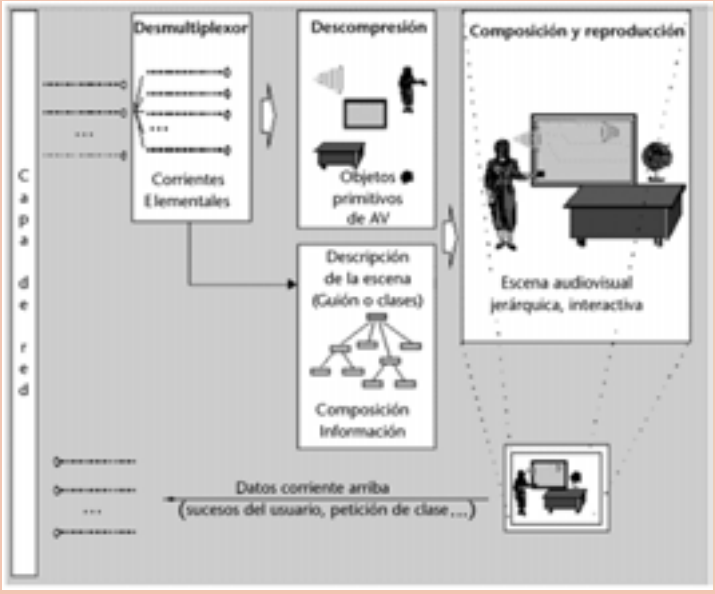
Ejemplo de una escena y su descomposición en planos de objetos



Grafo de relaciones entre objetos en la escena de la imagen anterior

Terminal MPEG-4

En la figura se muestra la estructura de un terminal de MPEG-4 que muestra cómo puede componerse la escena final a partir de la información proporcionada por la descompresión de los objetos y la información de composición.



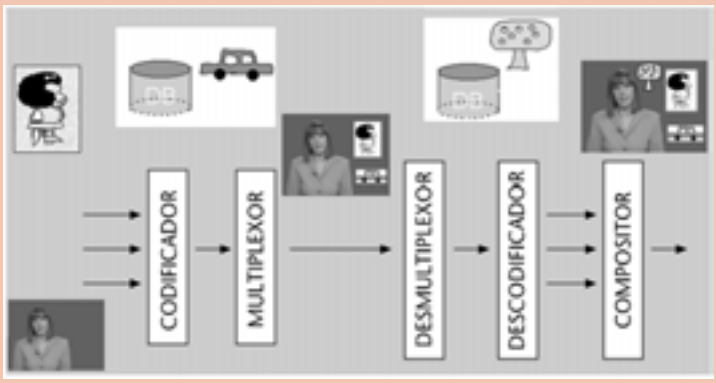
Terminal genérico MPEG-4

Los elementos con los que puede interactuar el usuario en el compositor son los siguientes:

- Posición espacial de los objetos
- Tamaño de los objetos
- Prioridades de *overlay* (orden de fundido de los VOP)
- Selección de los objetos visibles

Reutilización de contenidos

La descomposición de la escena en objetos independientes facilita que puedan reutilizarse contenidos. Una posibilidad es que el receptor combine objetos que tiene almacenados en su disco duro con los objetos que está recibiendo. El generador de contenidos también puede combinar objetos definidos previamente en bases de datos con objetos que se codifican en tiempo real.



Ejemplo de reutilización de contenidos

Planos de objetos de audio

Hasta ahora hemos supuesto que sólo estamos codificando una secuencia de vídeo. Si la secuencia contiene audio y vídeo, la filosofía general es parecida. En primer lugar, debe descomponerse el contenido original en VOP y AOP (*Audio Object Plane*). Los AOP en los que se puede descomponer una escena son también muy variados. En la siguiente figura se representa una escena típica en la que se combinan diferentes objetos de audio.

Ejemplo de una escena con varios AOP's

Los objetos de audio también pueden ser naturales (voz y música) o sintéticos. Los sonidos admiten distintas estrategias de codificación, que van desde tasas extremadamente bajas (2 Kbps) hasta tasas de alta calidad. Todos los modos de audio definidos en estándares anteriores (MPEG-1, MPEG-2) pueden ser utilizados en el MPEG-4. Los modos de baja velocidad están orientados principalmente a la codificación de voz y utilizan técnicas con características parecidas a las de telefonía digital o videoconferencia.

Los sonidos sintéticos incorporan conversores de texto a voz con velocidades de codificación situadas entre los 200 bps y los 1,2 Kbps. También se incluyen herramientas de audio estructurado (SAOL: *Structured Audio Orchestra Language*) en las que se definen distintos modos de síntesis a partir de la partitura y las características de los instrumentos.

Codificación de objetos de vídeo natural

Herramientas para la representación de vídeo natural

El MPEG-4 proporciona distintas herramientas para la codificación de objetos de vídeo con formas arbitrarias. Gran parte de los procedimientos están basados en funcionalidades que anteriormente ya se habían definido para otros estándares a los que se incorporan herramientas adicionales para optimizar la codificación de los objetos con formas arbitrarias.

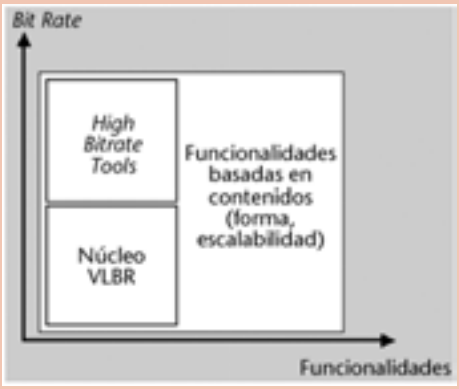
En la siguiente figura se muestra una primera clasificación de todo el conjunto de algoritmos y herramientas que se contemplan en el estándar. Las herramientas y modos de codificación utilizados dependen en última instancia de la aplicación final a la que destina el codificador (vídeo no interactivo para Internet o videoconferencia, vídeo no interactivo de calidad, vídeo interactivo).

- El núcleo VLBR (*Very Low Bit Rate*) proporciona algoritmos para la codificación de vídeo entre tasas que operan entre 5 kbps y 64 kbps. El tamaño de la imagen es típicamente CIF y las frecuencias de imagen suelen ser de hasta unos 15 fps. Los objetos son siempre rectangulares y coinciden con toda la imagen. Los algoritmos tienen poca latencia y pueden ser implementados en tiempo real. Este conjunto de herramientas es en parte compatible con las definidas en el H.263+. Además, incorpora elementos en el *stream* de datos para facilitar el acceso aleatorio a la información y permitir el avance y el rápido retroceso del vídeo (aplicaciones de almacenamiento de vídeo en bases de datos).

- El grupo de altas velocidades incorpora las mismas funcionalidades y algoritmos que el núcleo VLBR, pero puede trabajar con imágenes de tamaño completo y frecuencias de imagen de 25 fps. También pueden codificarse modos entrelazados con procedimientos análogos a los definidos en el MPEG-2. Las tasas de codifica-

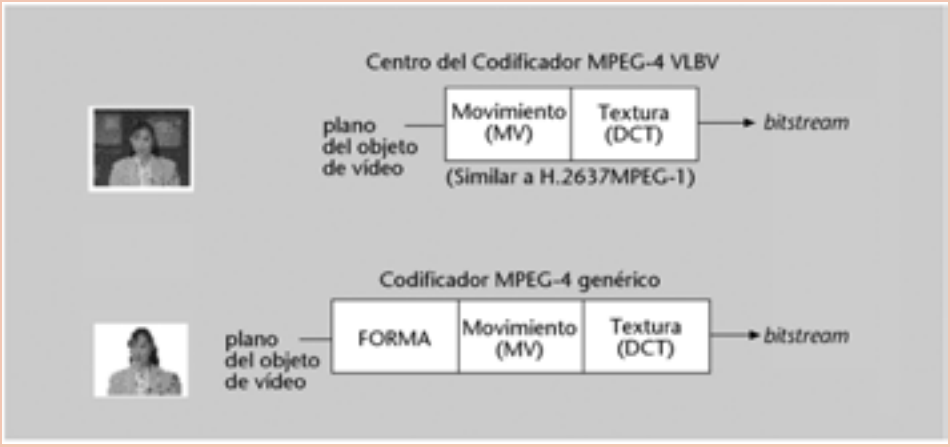
ción típicas en este conjunto de aplicaciones están situadas entre 64 kbps y 10 Mbps.

- El grupo de herramientas de codificación basadas en contenido es el único que permite la definición de VOP y, por lo tanto, introduce los elementos necesarios para la interactividad con estos contenidos.



Clasificación de las herramientas proporcionadas por el MPEG-4 en función de la aplicación

La siguiente figura compara la codificación de un vídeo con las herramientas proporcionadas por el núcleo VLBR (el único VOP es el *frame*) con un codificador MPEG-4 genérico en el que debe tenerse en cuenta la información de forma del objeto.



Comparación entre la codificación de un vídeo con VLBR y con un codificador genérico.

Esquema para la codificación de vídeo

La codificación de vídeo en los modos en que no están definidos objetos con formas arbitrarias es, como ya hemos mencionado, análoga a la de otros estándares basados en la transformada coseno. Se incluyen todas las herramientas avanzadas que están definidas para el H.263+. No obstante, cuando el objeto tiene una forma arbitraria, existen algunas diferencias significativas con el resto de los estándares.

En la figura siguiente se representa un esquema genérico de cómo se realiza la codificación de un bloque. Notad que, exceptuando la parte de la codificación de la for-

ma del objeto, el resto del diagrama de bloques coincide con el esquema básico utilizado en otros estándares.

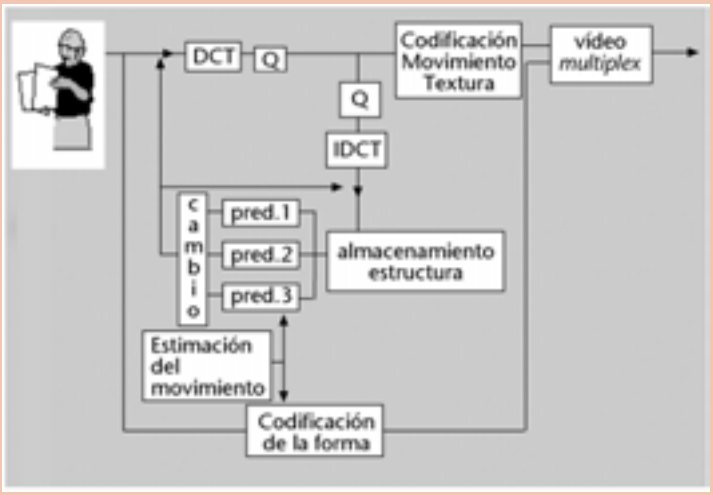
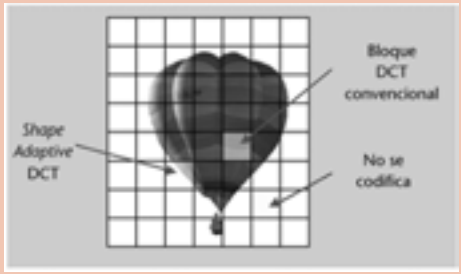


Diagrama del procedimiento de codificación en MPEG-4

Las diferencias más importantes cuando el objeto tiene forma arbitraria es que es necesario identificar los bloques de la imagen que deben ser codificados y los que no. Esto depende de la forma de objeto. El objeto se divide en una retícula rectangular, tal y como se muestra en la figura, y sólo se codificarán los bloques que tienen algún contenido de imagen. El resto se codifica como transparencia. Los bloques que están completamente en el interior del objeto se codifican como en el caso convencional, mientras que los que están en los límites del objeto se codifican utilizando un modo especial de DCT que se denomina *Shape Adaptive* DCT y que tiene en cuenta la relación espacial entre el bloque y el contorno del objeto. Los detalles de este modo de codificación son complejos y se omiten, pero en esencia, de lo que se trata es de optimizar la codificación empleando sólo los bits necesarios.



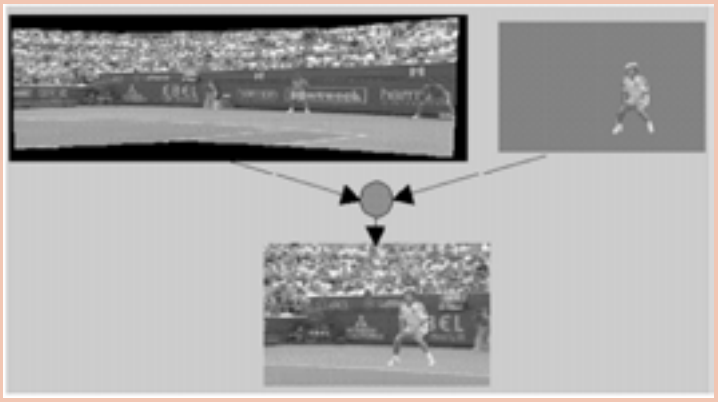
Distintos tipos de bloques en la codificación de objetos con forma arbitraria

Definición de *sprites*

Una de las ventajas que aporta la descomposición en VOP es que pueden definirse *sprites* estáticos. Un *sprite* es una imagen, probablemente de gran tamaño, que proporciona una panorámica global del fondo que debe codificarse. La idea es transmitir el *sprite* (fondo) durante la primera parte del *stream* de manera que el descodificador pueda almacenar esta imagen de fondo en la memoria y utilizarla posteriormente para presentar los diferentes objetos móviles sobre dicho fondo. Generalmente, el

fondo tiene un tamaño muy superior al área visible de la imagen, por lo que sólo se visualiza una parte del mismo. El codificador debe enviar los movimientos de cámara para que el receptor pueda realizar las transformaciones oportunas al fondo antes de presentarla en el *display*. Estas transformaciones pueden incluir el desplazamiento horizontal o vertical (movimientos de la cámara).

En la siguiente figura se muestra un ejemplo de un *sprite*. En este caso suponemos que resulta fácil separar el tenista del fondo y decidimos enviar la imagen del fondo como un *sprite* que el descodificador almacenará en memoria. A partir de este momento sólo es necesario codificar la imagen en tiempo real del tenista y proporcionar los movimientos de cámara para que el receptor pueda ir moviendo la posición del fondo durante la transmisión. Alternativamente, no es necesario enviar todo el fondo en las primeras fases de la transmisión, sino que puede irse enviando de forma progresiva, fotograma a fotograma, y que sea el propio receptor quien reconstruya la información a partir de los fotogramas parciales.



Ejemplo de una codificación de vídeo utilizando *sprites*

Codificación de imágenes fijas

El MPEG-4 tiene definido un modo especial, basado en la transformada Wavelet, para la codificación de imágenes fijas. Las características de este modo son parecidas a las del JPEG2000 y presenta compatibilidad con éste.

Codificación escalable de objetos de vídeo

El MPEG-4 soporta la codificación de imágenes y de objetos de vídeo escalables tanto para los objetos rectangulares convencionales como para los objetos con formas arbitrarias. La escalabilidad supone que el descodificador, en función de su ancho de banda y su capacidad de proceso, puede elegir entre descodificar todo el *bitstream* o sólo una parte del mismo. La capa base es la que proporciona una calidad mínima pero necesaria para poder descodificar el resto de las capas. La calidad con la que se reproducen las imágenes mejora a medida que el descodificador es capaz de utilizar más capas. Los tipos de escalabilidad que contempla el MPEG-4 son los mismos que contemplan el resto de los estándares: espacial, calidad (SNR) y temporal.

Robustez en entornos con errores

Uno de los objetivos del MPEG-4 es que pueda utilizarse en entornos móviles, con tasas de codificación muy bajas y en los que pueden producirse ráfagas de errores con cierta frecuencia. El sistema de codificación incorpora distintas herramientas para atenuar los efectos de la aparición de errores.

- **Resincronización**. Una de las características nuevas del MPEG-4 es que envía códigos de resincronización de forma periódica, cada cierto número de bits del *stream*. Los códigos de resincronización permiten que la descodificación de la secuencia a partir de este código pueda realizarse sin tener en cuenta los valores anteriores. Si se han producido errores en la trama de bits, el sistema puede recuperarse a partir de un código de resincronización.

- **Recuperación de datos**. Aunque no se incorporan códigos correctores convencionales en la secuencia de datos, los códigos de longitud variable que se usan son reversibles. Esto significa que pueden leerse de izquierda a derecha o de derecha a izquierda. Con esta técnica es posible recuperar algunos datos de la secuencia cuando se produce un error.

- **Ocultación de errores**. La información de vectores de movimiento y la información de los coeficientes transformados de la DCT se envían en tramas separadas. Esto permite que si se producen errores en los coeficientes, aún puedan utilizarse los vectores de movimiento para sustituir los bloques del *frame* de referencia en el bloque actual. Esta estrategia es equivalente a que el error de cuantificación obtenido durante la codificación no se hubiese enviado.

Codificación de objetos de vídeo sintético

El MPEG-4 define distintos tipos de objetos de vídeo sintéticos que pueden combinarse en una misma escena con los objetos naturales. La combinación entre ambos tipos de objetos (incluyendo también los objetos de audio naturales y sintéticos) se conoce como SNHC (*Synthetic and Natural Hybrid Coding*).

Los diferentes elementos que pueden combinarse incluyen los siguientes elementos:

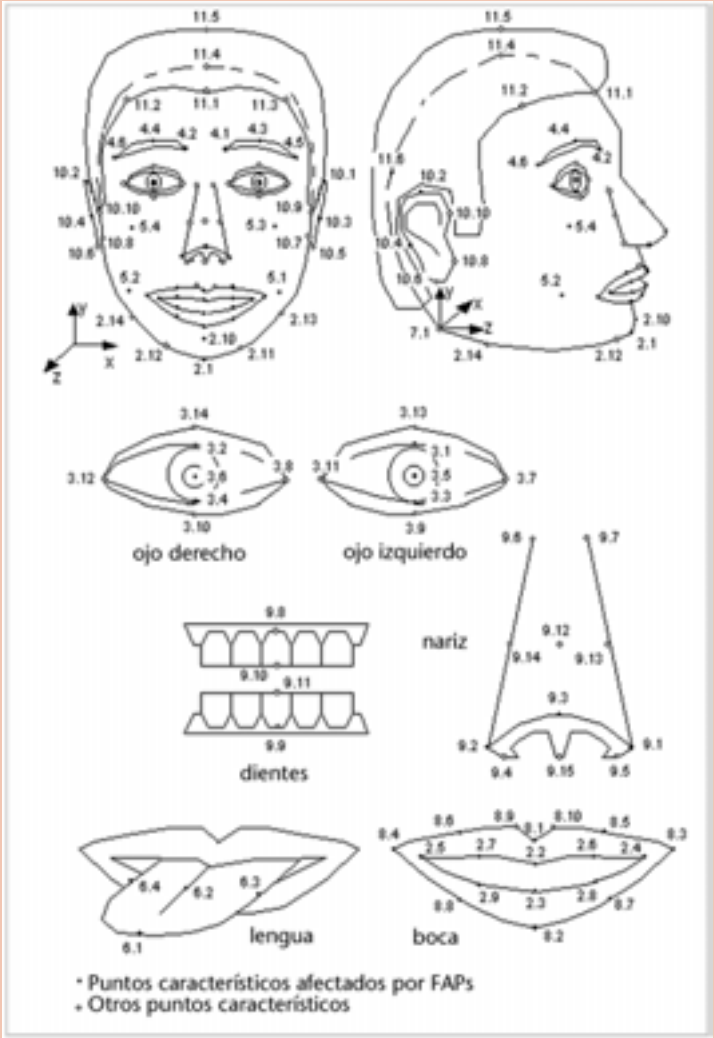
- Objetos de vídeo natural
- Objetos de audio natural
- Objetos de animación facial o animación de cuerpo
- Retículas 2D animadas
- Objetos de audio estructurado
- Conversores de texto-voz
- Gráficos, texto

Consideraremos como ejemplos los objetos de animación facial y las retículas 2D animadas.

Objetos de animación facial

La animación facial puede tener aplicaciones muy diferentes. Probablemente la más importante sea en el soporte a sordos en aplicaciones de telefonía.

En esencia, la animación facial consiste en enviar una serie de puntos que definen los puntos característicos de una cara y posteriormente ir enviando los movimientos que se producen a lo largo del tiempo de estos puntos para producir la animación. Durante la construcción se envía una cara genérica con expresión neutra y después se envía un *bitstream* que realiza la animación de los puntos. La cara genérica definida por el estándar se representa en la siguiente figura.



Definición de los puntos característicos de una cara en aplicaciones de animación facial

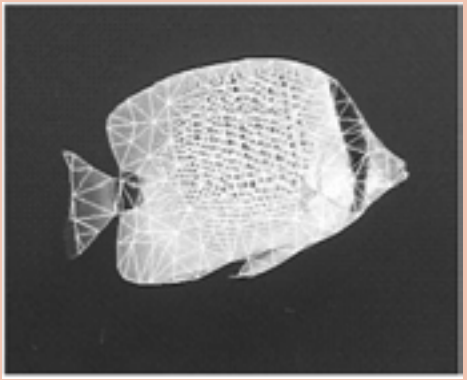
Es posible incorporar texturas o personalizar las posiciones de los puntos para que el objeto facial muestre un parecido directo con el interlocutor.

La principal aplicación se da en telefonía móvil, sector en el que aplicando un sistema de reconocimiento de voz e identificación de fonemas del usuario que habla pueden codificarse las posiciones de los labios y lengua que representan de forma visual el contenido de los fonemas.

Retículas 2D animadas

Las retículas 2D animadas representan una forma muy eficiente de codificar la información y producir animaciones a un coste de codificación muy bajo. Esencialmente consisten en definir una serie de puntos del objeto que debe codificarse y construir una malla formada por polígonos triangulares. Cada polígono se codifica con una textura de forma que puede reconstruirse una imagen muy aproximada a los objetos reales. La animación se consigue enviando la forma como se desplazan cada uno de los puntos de la retícula en el tiempo y modificando las estructuras poligonales en consecuencia.

Los algoritmos para la selección de los puntos de la segmentación original no están definidos por el estándar. En la figura se muestra un objeto codificado como una retícula 2D.



Ejemplo de un objeto codificado como una retícula poligonal 2D

Etapa 4: El estándar MPEG-7

Introducción

A medida que aumenta la cantidad de información audiovisual disponible en formato digital, resulta más complejo el proceso de gestión de los contenidos y su búsqueda y acceso eficiente. Es posible disponer de materiales audiovisuales de gran interés, pero antes de utilizarlos deben ser localizados. Cuanto más aumenta el volumen de información multimedia, más compleja resulta su búsqueda.

Actualmente existen motores de búsqueda de gran utilidad para documentos de tipo texto. La mayor parte de tales documentos están disponibles por medio de páginas web que actúan como buscadores y que, sin duda, se encuentran entre las más visitadas. Sin embargo, la búsqueda de materiales audiovisuales no resulta tan simple y es difícil establecer criterios de descripción de estos materiales que resulten intuitivos para el público en general. Por el momento, no existe ningún buscador que introduciendo la frase "Hombre y mujer detrás de un piano con un pianista en primer plano" nos proporcione directamente el siguiente fotograma correspondiente a la película *Casablanca*.

Ejemplo de búsqueda de contenidos audiovisuales

Actualmente existen algunas aplicaciones de bases de datos multimedia que permiten la búsqueda de fotografías a partir de informaciones como el color, la textura y la forma de los objetos que aparecen en la imagen.

Disponer de un estándar que normalice y proporcione herramientas para describir los contenidos de materiales audiovisuales permitirá desarrollar mecanismos eficientes para la búsqueda de contenidos. Las aplicaciones no se limitan a la búsqueda de material en bases de datos o librerías digitales en otras áreas. Así, si se incorporan descriptores de contenido en la cabecera de los propios materiales audiovisuales, será posible sintonizar un programa de televisión de entre varios paquetes de programas por satélite o cable a partir de la descripción del contenido que queremos visualizar. En los estudios de televisión existe una gran necesidad de poder indexar el material

audiovisual de forma eficiente para buscar posteriormente acontecimientos o reutilizar el material para nuevas posproducciones.

El MPEG-7 se conoce formalmente como la interfaz de descripción de contenidos multimedia y es el nuevo estándar aprobado por la ISO para facilitar y estandarizar los procesos de descripción en el ámbito sintáctico y semántico de contenidos multimedia. Estos contenidos engloban vídeo, audio, voz, síntesis y efectos, gráficos, imágenes fijas, modelos tridimensionales, etc.

El estándar incluye varios aspectos de representación del conocimiento y extracción de características de alto y bajo nivel que requieren conocimientos profundos en diferentes áreas tecnológicas. En el apartado siguiente veremos los principios básicos y algunas particularidades concretas del estándar. El lector que quiera profundizar sobre estos temas deberá acudir a la extensa documentación que se referencia en la bibliografía, que en gran parte puede encontrarse en www.mpeg.org.

Una cuestión que despierta la curiosidad de mucha gente es la extraña numeración de los estándares MPEG. El MPEG-3 fue originalmente un grupo de trabajo que debía considerar la extensión de los estándares MPEG-2 a la televisión de alta definición. El trabajo realizado por el MPEG-2 fue tan bueno que ya se incorporaron todos los aspectos relacionados con la alta definición, por lo que el MPEG-3 desapareció. La descripción de contenidos multimedia debería haber tenido asignado el número 5. No obstante, se consideró que existía un salto cualitativo importante en los objetivos y aspectos temáticos del nuevo estándar y finalmente se adoptó el número 7, que según algunos puede interpretarse que es la suma de los anteriores estándares (1 + 2 + 4 = 7) y según otros es, simplemente, un número que representa la suerte.

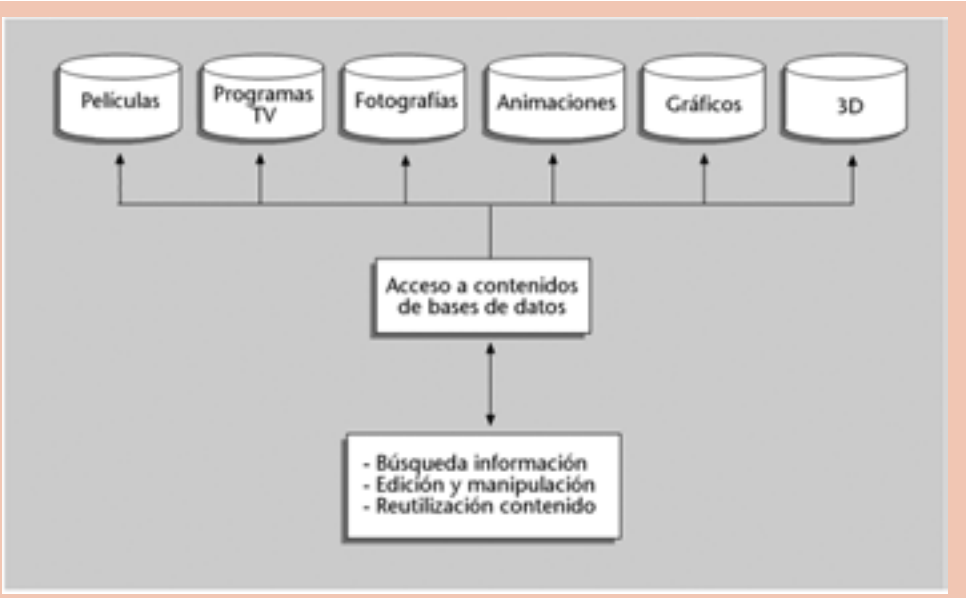
Conceptos básicos

El grupo de trabajo del MPEG-7 se inició en octubre de 1996 y la primera versión del estándar está disponible desde septiembre de 2001. Las herramientas de descripción de contenidos son independientes del formato en que está registrado el material audiovisual. Debe ser posible crear una descripción MPEG-7 tanto de una película en formato analógico como de una fotografía impresa en papel o de un vídeo codificado en estándares como MPEG-1, MPEG-2 o MPEG-4. No obstante, el MPEG-7 puede aprovechar el análisis previo realizado por el proceso de compresión de otros estándares. Esto significa que un descriptor de forma utilizado durante la codificación en MPEG-4 de una fuente puede resultar útil en el contexto de descriptores MPEG-7. Lo mismo puede ocurrir con los vectores de movimiento calculados por el MPEG-1 o el MPEG-2 para describir el movimiento de una escena.

Objetivos

El objetivo fundamental del MPEG-7 es proporcionar un estándar que defina un conjunto de descriptores que puedan ser utilizados de forma eficiente para describir distintos tipos de información multimedia. Para ello se definen estructuras o esquemas

de descriptores que pueden relacionarse entre sí. La combinación de los descriptores y los esquemas de descripción se asocia al material audiovisual para permitir que sea indexado o puedan realizarse búsquedas eficientes.



Esquema de un sistema de acceso a bases de datos con contenidos multimedia

Ejemplos de búsqueda

Para entender mejor los objetivos es conveniente poner unos ejemplos de búsquedas avanzadas de contenidos multimedia de deberían poder realizarse en una base de datos:

- Tocar unas notas en el teclado del ordenador o un instrumento y recibir un listado de todas las piezas musicales que contienen un fragmento parecido. La misma función podría realizarse silbando la melodía en un micrófono.

- Dibujar unas líneas en la pantalla y obtener un conjunto de imágenes que contengan gráficos o logos parecidos.

- Definir movimientos y relaciones espaciales entre objetos y obtener animaciones que contengan las relaciones deseadas.

- Describir acciones como por ejemplo: "Señor con traje azul y maleta subiendo a un tren" y obtener una lista de los materiales audiovisuales que contengan estas acciones.

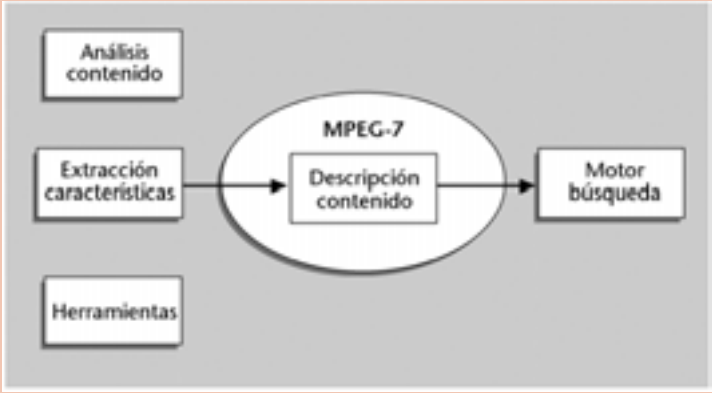
- Utilizar un fragmento con la voz de Josep Carreras y obtener una lista de todos sus discos, videoclips, pósters, etc.

Algunos de estos ejemplos pueden parecer auténtica ciencia ficción desde el punto de vista del tratamiento de señal necesario para extraer todas las características de

forma automática. Sin embargo, el estándar no se ocupa de los procedimientos utilizados para realizar la extracción de las características, simplemente define cómo deben describirse objetos con diferentes niveles de abstracción y la interrelación entre tales objetos. La introducción de ciertas características en los descriptores se puede realizar de forma automática en algunos casos y en otros requiere la participación humana para la correcta interpretación.

Perspectiva global

En la siguiente figura se representa un esquema de muy alto nivel en el que se define el contexto general que engloba el estándar MPEG-7. Para obtener la descripción de los contenidos es necesario utilizar varias herramientas que permitan extraer las características que se utilizaran como descriptores. El MPEG-7 no contempla ni define estas técnicas, simplemente se ocupa de establecer los esquemas para organizar los descriptores, los tipos de descriptores para diferentes materiales, su organización, estructura, etc. El MPEG-7 tampoco se ocupa de técnicas de motores de búsqueda y acceso a la información. Los aspectos de extracción de características y motores de búsqueda se dejan para desarrolladores externos, con lo que se facilita la libre competencia entre las empresas para obtener productos avanzados y compatibles dentro del contexto de la descripción de los contenidos. Esto significa que un motor de búsqueda desarrollado por una determinada empresa deberá ser capaz de gestionar de forma eficiente la base de datos de contenidos multimedia independientemente de cuáles hayan sido las aplicaciones que se hayan utilizado para realizar la extracción de las características.



Contexto general de aplicación del MPEG-7

Aplicaciones

El ámbito de posibles aplicaciones en el que puede usarse el MPEG-7 es muy variado. En los siguientes puntos se describen algunas posibilidades:

- Arquitectura, interiorismo (búsqueda de ideas y motivos arquitectónicos).
- Comercio electrónico (anuncios personalizados, catálogos *on-line*).
- Compras (búsqueda de vestidos, artículos deportivos, etc.).

- Educación (cursos multimedia, búsqueda de materiales multimedia para soporte docente).
- Entretenimiento (manejo de colecciones multimedia personales).
- Librerías digitales (catálogos de imágenes, diccionarios musicales, catálogos de imágenes biomédicas, archivos de películas, vídeo, etc.).
- Navegación (control de tráfico).
- Periodismo (búsqueda de discursos de personajes a partir de su nombre, su voz o su cara).
- Selección de programas de radiodifusión (radio, televisión).
- Servicios culturales (museos, galerías de arte).
- Servicios de directorios multimedia (páginas amarillas, guías de turismo, sistemas de información geográfica).
- Servicios de investigación (reconocimiento de caras, aplicaciones forenses).

Diversidad de características y descriptores

Las características que describen un determinado material deben resultar útiles en el contexto de diferentes aplicaciones en las que puede ser utilizado. Esto significa que el material debe describirse con diferentes tipos de características si debe ser utilizado en distintos contextos. Para ello se utilizan distintos niveles de abstracción en la descripción del material. Así, un material de vídeo puede describirse a bajo nivel utilizando características como la forma, el tamaño, la textura, el color, el movimiento o trayectoria, la posición, etc. Un objeto de audio puede ser descrito a bajo nivel utilizando características como la clave, el tempo, los cambios de tempo, la tonalidad, etc.

La abstracción de alto nivel es una descripción semántica entre las abstracciones de bajo nivel. Así, la descripción "Señora con sombrero rojo subiendo a un taxi" sería una descripción de alto nivel que contiene distintos elementos de bajo nivel como la identificación de una persona, la identificación del color rojo, la identificación del sombrero, los sonidos elementales del tráfico de la escena, el color del taxi, etc.

El nivel de abstracción está directamente relacionado con las características y sus procedimientos de extracción. Seguramente es posible encontrar métodos eficientes para la extracción de características de bajo nivel de forma totalmente automática, pero las características de alto nivel requerirán, por el momento, una alta interacción humana.

Componentes del estándar

Los principales elementos en los que se sustenta el estándar MPEG-7 son los siguientes:

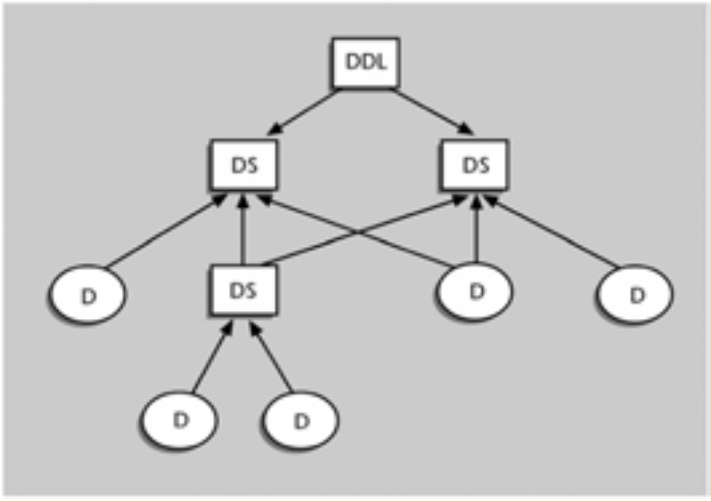
- Descriptores (D). Representación de las características. Se define la sintaxis y la semántica de cada representación.

- Esquemas de descripción (DS: *Description Schemes*). Definen la estructura y la semántica de las relaciones entre sus componentes que pueden ser descriptores (D) o esquemas de descripción (DS).

- Un lenguaje de definición de descripción (DDL: *Description Definition Language*). Se permite la creación de nuevos elementos DS y D. Se permite la extensión y modificación de esquemas de descripción (DS) ya definidos.

- Herramientas en el nivel del sistema para soportar la multiplexación de descripciones y su sincronización con los contenidos, los procedimientos de transmisión, las representaciones codificadas, el almacenamiento eficiente y la protección de la propiedad intelectual.

El MPEG-7 también utiliza el lenguaje XML *Schema* para las representaciones textuales de descriptores de contenidos. En la siguiente figura se muestra un diagrama de las relaciones:



Representación gráfica de la relación entre los diferentes elementos del estándar.

Otros tipos de información

Además de las descripciones referentes al contenido, también se incorporan otros tipos de información adicionales.

- La forma. Un ejemplo sobre la información de forma es el tipo de codificación utilizado (JPEG, MPEG-2, MPEG-4). También puede incluirse el tamaño total de los datos. Esta información permite que el usuario pueda determinar si es capaz de leer el contenido.

- Condiciones de acceso al material. Se incluyen enlaces al registro de la propiedad intelectual para información sobre derechos del material y precios.

- Clasificación. La clasificación del material incluye aspectos como el control paterno (*parental control*) y clasificación del contenido en categorías predefinidas.

- Enlaces a otros materiales de relieve. Esta información puede permitir que el usuario acelere la búsqueda del material.

- El contexto. En el caso de que se trate de materiales registrados en situaciones reales (materiales no ficticios) es importante disponer de información del contexto en el que se ha realizado la grabación (por ejemplo, Juegos Olímpicos 1992. Barcelona. Final de la maratón).

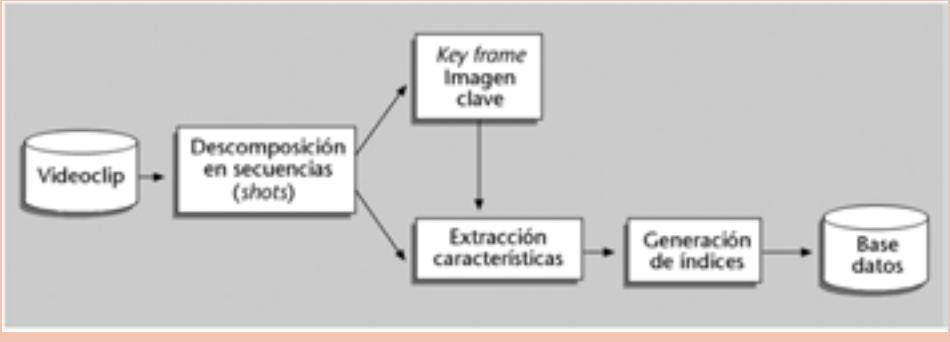
Aunque en algunos casos la adición de información textual puede resultar útil, debemos tener en cuenta que los descriptores deberían ser independientes del lenguaje. Un buen ejemplo es proporcionar los nombres de películas, actores, directores, etc. Sin embargo, proporcionar descripciones basadas únicamente en documentos textuales no es uno de los objetivos del MPEG-7.

Extracción de características de vídeo

Como ejemplo básico de una de las aplicaciones del MPEG-7, examinaremos con algo más de detalle el proceso de análisis y extracción de características de un vídeo. Se pueden definir dos problemas básicos: la indexación y la búsqueda (*index and query*).

Indexación de vídeo

En la indexación el problema consiste en analizar el contenido de un vídeo y descomponerlo en los planos e imágenes clave más significativas (*shots and key frames*). Después se aplica un procedimiento de extracción de características sobre cada uno de los *key frames* que permitan representar y describir la parte más significativa de su contenido y faciliten su búsqueda posterior. En la figura siguiente se muestra un esquema genérico del proceso de análisis y segmentación de la información de un vídeo.

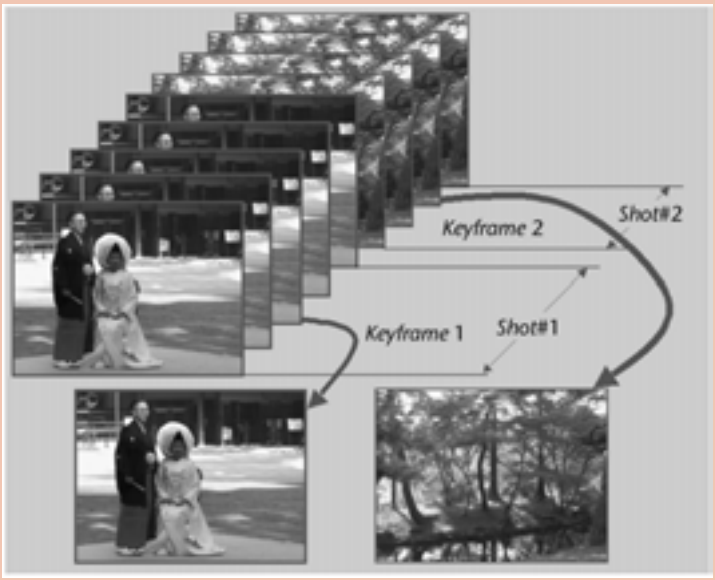


Esquema para la generación de índices sobre el contenido de un vídeo

Descomposición en secuencias (*shots*)

La primera etapa del proceso de análisis del vídeo es descomponerlo en las secuencias básicas, también denominadas planos, tomas o, en inglés, *shots*. El vídeo original

puede tener una duración aproximada entre una hora y una hora y media, y se trata de simplificar toda esta secuencia de fotogramas en una secuencia más simple y que pueda ser representada de forma compacta y sin que ello represente una pérdida de eficiencia para posteriormente encontrar el material que buscamos. Una toma o *shot* es un fragmento de vídeo en el que la información entre los *frames* sucesivos es muy parecida. En general, no existe ningún corte dentro de la toma y la correlación entre los diferentes *frames* es muy elevada. En la imagen se muestra un fragmento de secuencia de vídeo formado por dos *shots*, de los que se extrae un *frame* clave de cada uno de ellos.



Descomposición de una secuencia de vídeo en *shots* y *key frames*

Para separar la secuencia en *shots* pueden utilizarse algoritmos que comparen las imágenes sucesivas y realicen varias medidas de distancia entre dos *frames* sucesivos. Las medidas de distancia más habituales son las siguientes:

- Diferencia acumulada de píxeles entre las dos imágenes.
- Diferencias de histogramas entre las dos imágenes.
- Medidas de movimiento.

Generalmente, cuando la secuencia está editada por corte, todas las medidas de distancia aumentan considerablemente cuando se cambia de plano, por lo que la detección de inicio y final de un *shot* suele ser bastante evidente. Cuando se realiza un fundido entre los dos planos, la detección del cambio de plano es algo más compleja, ya que las medidas de distancia varían lentamente. Es posible detectar esta circunstancia cuando se detecta una diferencia importante en los píxeles de las imágenes sucesivas durante cierto periodo y la detección de movimiento produce resultados erráticos. En la siguiente figura se muestra un cambio de plano por corte directo y un cambio de plano por fundido.

Cambios de *shot* por corte y por fundido

La selección del *key frame* representativo de cada plano puede realizarse de varias formas. Un tratamiento óptimo consiste en seleccionar el *frame* que presenta un máximo parecido con todos los de su plano y una diferencia máxima con los de los otros planos. Existen otros criterios más sencillos que a veces simplifican notablemente los cálculos. Una solución trivial consiste en elegir el *frame* central de la secuencia.

Extracción de características y generación de índices

Una vez extraído el *frame* clave de cada secuencia, se procede a la extracción de características, que puede ser por medio de procedimientos totalmente automáticos o semiautomáticos. En estos últimos, parte de la información de alto nivel es proporcionada por un operador humano. La extracción de características de bajo nivel incluye operaciones como las siguientes:

- Detectar o describir el espacio de color de la imagen.
- Color dominante.
- Cuantificación del color.
- Tipo de textura (homogénea, histograma con picos).
- Detección de formas básicas.
- Movimiento de la cámara. Detección de movimiento de los objetos.
- Reconocimiento de objetos básicos.
- Reconocimiento de caras, etc.
- Sonidos de fondo.

Una vez determinadas las características de la secuencia y definidas las posibles relaciones de alto nivel entre los elementos de nivel inferior, se indexan los datos obtenidos y se almacenan junto con los *key frames* obtenidos en la base de datos. La información obtenida sobre la secuencia se denomina Metadata.

Para facilitar el posterior acceso y navegación por el material indexado pueden generarse sumarios que definan las interrelaciones y la secuencia entre los *key frames*. En la figura se muestra un sumario de un partido de fútbol estructurado en forma de árbol.

Sumario de un vídeo utilizando *key frames*

Búsqueda

El proceso de búsqueda (*query*) es el inverso del de indexación. En este caso se explora la base de datos para encontrar un contenido visual específico. Dependiendo de cómo se formule la cuestión al motor de búsqueda, el proceso puede ser muy complejo.

Una forma sencilla de búsqueda suele ser proporcionar una imagen de referencia de lo que estamos buscando en la base de datos. En este caso, el motor de búsqueda puede realizar cálculos previos sobre la imagen de referencia para extraer parte de sus características y compararlas con las de la base de datos. El resultado de la búsqueda son los *frames* clave que presentan características parecidas con la imagen de referencia.

Cuando la información se introduce de forma textual, como por ejemplo "señor con traje azul subiendo a un tren", la búsqueda es, hoy por hoy, un proceso bastante complejo sobre el que se está desarrollando una fuerte actividad investigadora.