Conceptos Tècnicos del DivX

Es complicado escoger siempre la resolución y los parámetros de DivX para la peli que vamos a hacer. Hay que tener en cuenta ciertos patrones de funcionamiento del códec (y de la potencia de tu ordenador) para escoger ambos, lógicamente también el espacio que queramos para nuestro AVI final.

Keyframes y VKI:

Antes de nada, hay que explicar qué son los keyframes y los delta frames. DivX para ahorrar espacio, entre otras cosas cosas, se basa en que de un fotograma al siguiente, la variación es muy poca, así que entre fotogramas sólo se codifica la diferencia. Un keyframe es una imagen completa, de la que deriva el resto. Los fotogramas que se forman a partir de este, son los delta frames, ni más ni menos. Es obvio que un keyframe ocupa bastante más que un delta frame, de ahí la necesidad del VKI.

Hasta hace poco, la inserción de keyframes se hacía cuando se cumplía un tiempo determinado (fijo), lo que podía provocar efectos raros (como que en una escena muy estática y suficientemente larga se produjeran cambios bruscos de calidad de imagen). Básicamente, el VKI funciona de dos maneras:

1.- Cuando detecta que el delta frame ocuparía más que un keyframe, inserta un keyframe.

2.- Cuando detecta un cambio de plano.

Actualmente hay dos maneras de hacer VKI en DivX: mediante el parche VKI (sección descargas) o desde un motor VKI en un codificador. La ventaja añadida de usar un codificador con motor propio de VKI es que también podría insertar keyframes en fundidos, y una detección mejor de cambios de plano, o también evitar demasiada distancia entre keyframes, lo que daría lugar a problemas de búsqueda y sincronización. Si vas a usar un codificador con esta característica (VirtualDub/Nandub por ejemplo) conviene no tener instalado el parche ya que podrías tener problemas, aunque si usas mpeg2avi, FlasKMPEG o similares deberías tenerlo instalado. Para hacer que funcione el parche, al configurar el intervalo de keyframe pon un número alto, como 9999. La instalación es sencilla, simplemente sustituir dos ficheros en c:\windows\system o c:\winnt\system32, normalmente puede hacerse sin reiniciar.

Al reproducir la película, si saltas de un punto a otro de la película el reproductor tiene que calcular los delta frames desde el keyframe anterior al punto que elegiste (sin visualizarlos), por ello no conviene demasiada distancia entre keyframes.

Bitrate:

La traducción literal es tasa de bits, por segundo, con lo que del bitrate depende directamente el tamaño que va a ocupar nuestro avi. Desde luego, mientras más alto sea, mejor va a ser la calidad de imagen (aparentemente), pero más va a ocupar la película.

En DivX normalmente usamos dos códecs, Low Motion y Fast Motion. Estos códecs no son CBR (bitrate constante), sino que más o menos son ABR (bitrate adaptativo). Ambos códecs son básicamente lo mismo, lo que ocurre es que varían en su parámetro de compresión, que puede variar desde 1x hasta 32x. Low motion usa desde 2x hasta 4x, generalmente (1x no se puede utilizar), y fast motion, desde 5x. Fast motion dedica una mayor parte que low del flujo de datos en movimiento de vectores, lo que hace que a bitrates más bajos funcione mejor con secuencias rápidas. En realidad lo que ocurre es que a medida que el parámetro de compresión aumenta disminuye el flujo de datos para la imagen en sí, pudiendo aprovechar algo más para codificar el movimiento, de ahí que Fast motion funcione mejor en general en escenas rápidas. Además, el códec varía adaptativamente este parámetro en función del movimiento detectado, con lo que la compresión se mueve en un rango de valores.

Este motivo hace que predecir el tamaño del archivo sea difícil, sobre todo para Fast motion que se mueve en un rango de valores mucho mayor que Low motion. De todas maneras, en la sección de descargas tienes algunas calculadoras muy buenas que, eso sí, sólo pueden utilizarse con Low motion. Lo he advertido.

*bitrate_video (en kbps)=[tamaño_final_MB-(bitrate_audio*duracion_segs/8192)]*8192/duracion_segs*. 8192 nos convierte de kbits a megabytes al dividir, o de megabytes a kbits al multiplicar (8bits/byte*1024kbytes/mbyte)

Ejemplo: queremos un avi con sonido a 128 kbps en 700 MB, la duración es de 93 minutos (5580 segundos).

bitrate_video=[700MB-(128kbps*5580s/8192)]*8192/5580s = 612.8125MB*8192/5580s = 899.67 kbits/s ~ **900 kbits/s**

Compression control:

Visto lo anterior, esto es fácil de entender. Es una tolerancia a la variación del parámetro de compresión. Estableces con este parámetro un compromiso entre nitidez de imagen (parámetros menores de compresion-cercanos a crispness) y fluidez de movimiento. Cuidado con este parámetro, porque a bitrates bajos es fácil que el códec sacrifique fotogramas por calidad de imagen. Normalmente, 80-85 es un valor ajustado para bitrates mayores de 600 kbits/s.

Resolución:

Elegir la resolución adecuada no siempre es fácil. Hay que tener presentes un varias cosas sobre el funcionamiento del códec.

Para comprimir la imagen, ésta se divide en bloques de 8x8 pixels (conocidos como macrobloques, los cuadros gigantes que se ven a veces en DivX), y el valor de estos 64 pixels se calcula como una media, más o menos. El bitrate se divide entre los macrobloques que tiene la imagen, de tal manera que a mayor bitrate, mayor detalle por macrobloque y mayor calidad de imagen resultante. Si el bitrate es demasiado bajo, el contenido del macrobloque se tiene que simplificar haciendo una especie de media por así decir, lo que puede provocar que el contenido de los macrobloques adyacentes no coincida demasiado, y por ello se notan tanto. Además, el códec distingue áreas con mayor necesidad de bitrate que otras, con lo que a veces ves una cara en primer plano completamente detallada y el fondo completamente macrobloqueado.

Lo aplicado a los frames se aplica aquí en los macrobloques. Los keyblocks son los macrobloques que se mantienen del keyframe original, y los delta blocks son los añadidos como distintos.

Lógicamente, mientras menor sea la resolución, menor número de macrobloques habrá, y tendrán más información para repartirse entre sí. Por tanto, mientras más bajo sea el bitrate, menor debe ser la resolución de vídeo. Si usas menos de 700 kbits/s de bitrate, debes de usar casi que mejor resoluciones de 512x*** para compensar, a partir de ahí debes ir aumentando.

Una característica del códec es que para cada resolución tiene un tope de calidad. Para 640x*** más o menos está en 1500 kbits/s, no te molestes en subir más el bitrate porque no vas a apreciar un aumento de calidad (aunque sí de tamaño xD), si subes el bitrate debes de subir la resolución consecuentemente para poder aprovecharla.

Otra cosa que debes de tener en cuenta al elegir la resolución es la máquina que tienes. A menor resolución, menos micro se consume. Por ejemplo, el paso de 720x576 a 640x480 supone procesar un 26% menos de pixels, con el aumento de rendimiento que ello conlleva.

Por cierto, que al estar codificada la imagen en bloques de 8x8, necesariamente tanto la resolución vertical como la horizontal tienen que ser múltiplos de 8 para conseguir una correcta reproducción (probablemente el programa codificador te mande a tomar por culo si le metes por ejemplo 639xloquesea). Eso como mínimo. Tenemos otro escollo. Es de gran ayuda para la reproducción que se pueda usar el overlay de la tarjeta gráfica, y en muchas se activa cuando la resolución horizontal es múltiplo de 16, o bien la vertical, o bien ambas... en concreto la mía, he comprobado que salta cuando ambas son múltiplos de 8 (más que nada, viene en el manual xD), por tanto os recomiendo que siempre que podáis escojáis resoluciones múltiplos de 16 (para los cenutrios crónicos, al ser 16 múltiplo de 8, cualquier múltiplo de 16 es también múltiplo de 8... creo que esto se daba en descomposición numérica como en 5º de EGB, pero algún kpullo habrá que me lo pregunte).

Relación de aspecto y su alter ego, el recorte:

Nota: toda la sección marcada va a ser reescrita para hacerla lo más exacta y ajustada a los términos que se emplean el el cine, sobre todo en el aspecto técnico de las relaciones de aspecto, pero como nota práctica puede servir de momento.

La mayoría de películas en DVD suelen venir tal y como se filmaron en cine, en formato panorámico. Decir formato 16:9 es una manera genérica de decir panorámico, puesto que hay muchas películas que no siguen esta relación. En realidad, la imagen de un DVD está siempre en formato 4:3. Si ves franjas negras arriba y abajo (quedando un área visible, que es a lo que se refiere la relación de aspecto) puede ser por dos razones:

-El DVD tiene codificadas esas franjas negras: Las franjas negras forman parte de la imagen.

-El DVD es anamórfico: para aprovechar mejor el bitrate (por algo que luego veremos) el área visible está codificada en formato 4:3, es decir, ocupa toda la pantalla (algo que podemos ver si abrimos directamente un vob de uno de estos DVD), pero es el reproductor el que achata la imagen y añade las franjas negras en reproducción.

La relación de aspecto suele venir en plan 2'35:1 ó 1'77:1, que son las más usadas. Eso significa que la resolución

horizontal es x veces la vertical. Por ejemplo, si queremos poner de resolución 640x***, con un DVD con relación de aspecto 2'35:1, nos quedaría 640x272 que es perfectamente divisible entre 8. Si nos queda un número que no lo es, hay que aproximar al múltiplo de 8 más cercano - estamos ya modificando la relación de aspecto, pero en muy poco, ya que la diferencia será como mucho de 4 pixels, que frente a una resolución de 640 es el 0.625%.

Al estar codificada la imagen en bloques entre los que se reparte el bitrate, si esa línea que separa las franjas negras de la imagen (en el caso de un DVD no anamórfico) cae (con toda probabilidad) en mitad de un bloque, el códec detectará una mayor necesidad de bitrate en esas zonas en detrimento de otras. Con lo cual, tendremos una línea perfectamente definida y una mierda de imagen. Sin embargo, si hacemos el recorte justo ahí, eliminamos el problema. El tema no es por el espacio que ocupan las franjas negras (que lo ocupan), porque en sí se comprimen muy bien, el problema es precisamente esa distinción entre la franja negra y el área de imagen. Y por cierto, que aunque el DVD sea anamórfico o 4:3 siempre hay algo que recortar... arriba, abajo y a los lados.

Cada codificador recorta como le parece. Por un lado, FlasKMPEG primero modifica la resolución (en el caso de que le especificaras una resolución de destino menor que la original, o mayor) y a partir de la imagen modificada, hace el recorte. Además, el recorte sólo lo hace en múltiplos de 16. Por otro lado, VirtualDub hace primero el recorte sobre la imagen original, y luego ajustando la resolución a la resolución destino elegida. Esto hace que VirtualDub sea mucho más preciso a la hora de cortar que FlasK, y, por cierto, en este sentido está más optimizado, ya que no es lo mismo bajar de resolución (cosa que requiere mucha potencia de proceso) a una imagen de 720x576 que a una de 720x306 (una vez recortada).

FlasK y mpeg2avi recortan de la misma manera, primero bajar resolución y a partir de ahí recortar. mpeg2avi lo puede hacer en múltiplos de 8, es más preciso que FlasK. Ten en cuenta que siempre es preferible comerse unos pocos píxeles por alguna parte que mostrar la línea de separación de franja negra-imagen.

Qué significa iDCT?

Recientemente todos hemos visto, en FlasKMPEG, las optimizaciones de su iDCT para distintos procesadores, que nos hacían ahorrar mucho tiempo de proceso, pero en realidad casi nadie sabe qué optimiza exactamente.

En lo que nos ocupa, la iDCT es uno de los algoritmos que se usan para descodificar el flujo mpeg2 que nos interese. Y claro está, para descodificar nuestro avi una vez hecho. Las tarjetas decodificadoras para DVD tienen entre sus características la iDCT por hard, lo que agiliza mucho la reproducción, aunque de momento no es posible aprovechar esta característica (que también tienen muchas tarjetas de vídeo como las ATI) para la reproducción de DivX.