UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DEPARTAMENTO DE COMUNICACIÓNES

PROGRAMA DE DOCTORADO EN TELECOMUNICACIONES

TÉCNICAS DE MEJORA DE LA EFICIENCIA DE CODIFICACIÓN DE VÍDEO

TESIS DOCTORAL

Presentada por Pau Usach Molina

Dirigida por el Dr. Jorge Sastre Martínez

Valencia, 2015

A mi madre

AGRADECIMIENTOS

A mis padres, por su cariño, por su paciencia, por su apoyo, por haber sido siempre un ejemplo y un espejo donde mirarme. Pero sobre todo, por su amor incondicional.

A mi hermano, porque ser un espejo en el que alguien se mira es también una gran responsabilidad.

A mis amigos, porque la vida sin amigos no es vida.

A mis compañeros en la Clementina, por su amistad y porque sin ellos este viaje no sólo habría sido más aburrido, sino que habría sido imposible.

A Alberto, Antonio, Josep y Valery, por su ayuda y soporte durante todo el tiempo que trabajamos juntos.

Quiero agradecer especialmente a mi director de Tesis, Dr. Jorge Sastre Martínez, su increíble trabajo, su paciencia, su apoyo y su confianza en mí durante todo este tiempo. Gracias por no dejar que me rindiera y por acompañarme en este bonito viaje.

Pero sobre todo, gracias a Ana por ser mi Constante.

RESUMEN

Esta tesis presenta un conjunto de herramientas que permiten mejorar la eficiencia de codificación de vídeo mediante la explotación de los fundamentos en los que se basan los principales estándares de codificación actuales.

El trabajo se ha orientado tanto a la investigación como a la aplicación de los resultados a la codificación de vídeo en tiempo real en entornos móviles.

En primer lugar se ha definido un algoritmo de detección automática de cambios de plano para entornos de tiempo real integrado en el proceso de codificación. Este algoritmo está basado en la monitorización del modo de codificación de los macrobloques de la secuencia y la correcta definición de un conjunto de parámetros consigue unas tasas de detección, una precisión y una eficacia superiores a otros métodos similares existentes en la literatura.

Los resultados muestran también una mejora en la calidad del vídeo codificado al aplicar estas técnicas de detección, lo que lleva a la definición de un algoritmo de selección de imágenes de referencia (*keyframes*) basado en el contenido. Así se pueden obtener las posiciones óptimas para las imágenes de referencia utilizadas por el codificador para realizar predicciones temporales que aumentan la calidad tanto objetiva como subjetiva del vídeo codificado, lo que constituye a su vez el objetivo principal de esta tesis.

Por último, se ha diseñado un algoritmo de control de tasa capaz de obtener un *bitstream* que se adapta rápidamente a los cambios tanto de *bitrate* como de tasa de imágenes por segundo producidos en el canal móvil.

Paralelamente, se ha obtenido un conjunto de secuencias de entrenamiento y test que proporcionan un entorno óptimo para el diseño, desarrollo, configuración, optimización y prueba de los algoritmos aquí descritos.

ABSTRACT

This Thesis presents a set of tools that allows the improvement of the digital video coding efficiency by exploiting the fundamentals of the state of the art video coding standards.

This work has been focused both on research and on the application of the results to the encoding of digital video in real time mobile environments.

The first contribution is an automatic shot change detection algorithm integrated in the encoding process. This algorithm is based on the monitoring of the coding mode of the macroblocks of the sequence, and the proper definition of a set of parameters provides excellent detection rates, precision and recall.

The results also indicate an improvement on the encoded video quality when these detection techniques are used, which triggers the definition of a content-based *keyframe* selection algorithm. With this method, the optimal position of reference pictures can be determined. These *keyframes* are then used by the encoder to perform temporal prediction of the subsequent frames, which improves the compression rate and the encoded video quality (both objective and subjective). This quality improvement is the main objective of this Thesis.

In the last part of this work, a rate control algorithm for variable bitrate and frame rate environments has been defined, being able to generate a *bitstream* that quickly follows the varying conditions of the mobile channel.

In parallel to all this work, a set of training and test sequences has been obtained, providing an optimal environment for the design, development, configuration, optimization and test of the algorithms described here.

RESUM

Aquesta tesi presenta un conjunt de ferramentes que permeten millorar la eficiència de codificació de vídeo digital mitjançant l'explotació dels fonaments en els que es basen els principals estàndards de codificació actuals.

El treball ha estat orientat tant a la investigació com a l'aplicació dels resultats a la codificació de vídeo en temps real en entorns mòbils.

En primer lloc s'ha definit un algoritme de detecció automàtica de canvis de plànol integrat en el propi procés de codificació. Aquest algoritme s'ha basat en la monitorització del mode de codificació dels macroblocs de la seqüència, i la correcta definició d'un conjunt de paràmetres de configuració permet aconseguir unes taxes de detecció, una precisió i una eficàcia superiors a altres mètodes similars presents a la literatura.

Aquests resultats també indiquen una millora en la qualitat del vídeo codificat al aplicar aquestes tècniques de detecció la qual ens porta a la definició d'un algoritme de selecció d'imatges de referència (*keyframes*) basada en el contingut. Amb aquest algoritme es poden obtenir les posicions òptimes per a les imatges de referència utilitzades pel codificador per a realitzar prediccions temporals òptimes que augmenten la qualitat tant objectiva com subjectiva del vídeo codificat. Amb esta millora s'assoleix l'objectiu principal d'aquesta tesi.

Per últim, s'ha dissenyat un algoritme de control de taxa capaç d'obtenir un *bitstream* que s'adapta ràpidament als canvis tant de *bitrate* com de taxa d'imatges per segon requerits per les condicions canviants del canal mòbil.

Paral·lelament s'ha obtingut un conjunt de seqüències d'entrenament i test que permet disposar d'un entorn òptim per al disseny, desenvolupament, configuració, optimització i prova dels algoritmes descrits en aquestes fulles.

ORGANIZACIÓN

Esta tesis se ha dividido en cuatro capítulos y un apéndice, cuya descripción general se presenta a continuación:

1. **Capítulo 1**: en el primer capítulo se presenta una introducción a la codificación digital de vídeo, que sirve como marco para el resto de desarrollos presentados en la tesis. Se analizan las características básicas del paradigma de codificación híbrida, así como un breve resumen de los distintos estándares de codificación que se han utilizado históricamente. Finalmente, se hace hincapié en el estándar H.264/MPEG-4 AVC sobre el cual se han implementado las técnicas aquí descritas.

2. **Capítulo 2**: en este capítulo se introduce la primera de las herramientas de codificación desarrolladas. En concreto, se trata de un detector de cambios de plano integrado en el proceso de codificación de las imágenes en un codificador H.264.

3. **Capítulo 3**: la segunda de las herramientas de codificación, un detector de imágenes de referencia basado en el contenido, se presenta en este capítulo. El selector se basa en el diseño del detector de cambios de plano anterior y produce una mejora tanto objetiva como subjetiva en la calidad del vídeo codificado.

4. **Capítulo 4**: el último capítulo se centra en el control de tasa con *bitrate* y número de imágenes por segundo variable, destinado a su utilización en entornos móviles donde las características del canal varían constantemente. El algoritmo se basa en algoritmos existentes

en la literatura, orientados a entornos con tasa de imágenes por segundo constante, que han sido adaptados a las nuevas condiciones.

5. **Anexo**: en el anexo se describen las secuencias utilizadas para desarrollar y probar los algoritmos anteriormente descritos, constituyendo una base sólida para determinar las bondades de dichos sistemas. Las secuencias aquí descritas cubren gran cantidad de escenarios y tienen un conjunto de características controladas para medir distintos parámetros de los algoritmos propuestos.

ÍNDICE

Capítulo 1: Fundamentos de la Codificación Digital de Vídeo

Capítulo 2: Detección Automática de Cambios de Plano

Capítulo 3: Inserción de Keyframes Basada en el Contenido

Capítulo 4: Control de Tasa con Bitrate y Frame Rate Variable

Anexo

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ECUACIONES

Capítulo 1

Fundamentos de la Codificación Digital de Vídeo

I. Introducción

La codificación digital de vídeo es una tecnología que ha evolucionado exponencialmente durante las últimas décadas, exprimiendo las prestaciones del hardware de procesamiento, de los medios de almacenamiento, de las capacidades de transmisión y de los medios de presentación del vídeo codificado. Por lo tanto, las técnicas de mejora de la eficiencia de codificación constituyen un campo de máximo interés para la industria y la investigación, abarcando diferentes líneas de trabajo destinadas a la mejora de la calidad del vídeo codificado.

Este capítulo pretende fijar el marco en el que se desarrollan las diferentes técnicas descritas en esta tesis.

Para definir un conjunto de herramientas para mejorar la eficiencia de codificación de vídeo se empezará por presentar las bases que fundamentan esta tecnología.

En concreto, la codificación digital de vídeo se desarrolla a partir de los años ochenta. Históricamente, el vídeo se almacenaba como una señal analógica en cinta magnética, pero tras la aparición del CD como sustituto del almacenamiento de audio analógico se hizo posible también la conversión del vídeo a un formato digital.

Sin embargo, debido a la gran cantidad de información que contiene la señal de vídeo y a las limitaciones de almacenamiento y ancho de banda, surgió la necesidad de reducir la cantidad de información de dicha señal de vídeo.

Así, matemáticos e ingenieros desarrollaron desde entonces un conjunto de técnicas de compresión y codificación de la información contenida en la señal

de vídeo, explotando sus particularidades y los últimos avances en hardware y software de codificación.

Uno de los objetivos de esta tesis es desarrollar un conjunto de herramientas de mejora de la eficiencia de codificación que se inserten en los distintos procesos llevados a cabo por el codificador de vídeo. Por lo tanto, para entender las implicaciones, ventajas e inconvenientes que supone la inserción de estos algoritmos dentro del proceso de codificación, es necesario comprender primero la estructura básica del codificador y los principios en los que se basa su funcionamiento.

II. Fundamentos

El volumen de información que incluye la señal de vídeo sin codificar es inmanejable para la mayoría de aplicaciones existentes, por lo que es necesario comprimir el vídeo original, eliminando en lo posible la gran redundancia (espacial y temporal) existente en la señal.

Las técnicas de compresión de vídeo consisten en reducir y eliminar datos redundantes del vídeo. Con técnicas de compresión eficaces se puede reducir considerablemente el tamaño del vídeo codificado, disminuyendo lo mínimo posible (opcionalmente nada) la calidad de la imagen.

Existen diferentes técnicas de compresión, tanto propietarias como estándar, siendo las últimas las más habituales en la mayoría de aplicaciones actuales. Los estándares son importantes para asegurar la compatibilidad y la interoperabilidad, y tienen un papel especialmente relevante en la compresión de vídeo, puesto que éste se puede utilizar para varias finalidades con distintos requisitos.

En el proceso de compresión se aplica un algoritmo al vídeo original para crear un archivo comprimido y listo para ser transmitido o guardado. Para reproducir el archivo comprimido se aplica el algoritmo inverso y se crea un vídeo que incluye prácticamente el mismo contenido que el vídeo original (idealmente, el mismo). El tiempo que se tarda en comprimir, enviar, descomprimir y mostrar un archivo es lo que se denomina latencia. Cuanto más avanzado sea el algoritmo de compresión, mayor será la latencia.

El par de algoritmos que funcionan conjuntamente se denomina códec de vídeo (codificador/ decodificador).

Los diferentes estándares de compresión utilizan métodos distintos para reducir los datos y, en consecuencia, los resultados en cuanto a tasa de bits (*bitrate*) y latencia son diferentes. Existen dos tipos de algoritmos de compresión:

- **Compresión de imágenes**: utiliza la tecnología de codificación *intra-frame*. Los datos se comprimen fotograma a fotograma con el fin de eliminar la información innecesaria que puede ser imperceptible para el ojo humano.
- **Compresión de vídeo**: la compresión de vídeo (*inter-frame*) explota la redundancia temporal de la secuencia para aumentar el grado de compresión.

III. Evolución histórica de la codificación de vídeo

Los inicios de la teoría de la compresión y codificación de vídeo se remontan a 1929, cuando Ray Davis Kell patentó una primera forma de compresión de vídeo. Citando este primer trabajo: "*Ha sido costumbre en el pasado transmitir*

imágenes sucesivas completas. […] Según este invento, esta dificultad se evita transmitiendo únicamente la diferencia entre imágenes sucesivas del objeto.". Aunque fue mucho tiempo antes de que estas técnicas pudieran llevarse a la práctica, todavía constituyen la base de la mayoría de compresores de vídeo actuales.

En los siguientes apartados se describen los principales hitos en la historia de los la codificación de vídeo que han seguido a esta primera idea [33][34], incluyendo tanto estándares de codificación como implementaciones significativas.

Son principalmente dos organizaciones las que se han encargado durante los últimos 35 años de estandarizar los sistemas de codificación de vídeo:

- **ITU-T Video Coding Experts Group (VCEG)**: Telecommunications Standardization Sector, Study Group 16, Question 6.
- **ISO/IEC Moving Picture Experts Group (MPEG)**: International Standardization Organization and International Electrotechnical Commission, Joint Technical Committee Number 1, Subcommittee 29, Working Group 11.

III.1. Primeros pasos

Los primeros sistemas de vídeo evolucionaron desde los osciloscopios. La televisión en blanco y negro se generalizó, y los primeros sistemas en color aparecieron a finales de la década de 1940.

En 1935 el gobierno británico definió la televisión de alta definición (HDTV) como aquélla que tuviera al menos 240 líneas. Actualmente, la definición de facto de la HDTV tiene como mínimo 720 líneas.

Una primera forma de compresión la constituye el entrelazado, donde los campos de líneas pares e impares se transmiten alternativamente. Esto es útil en dos aspectos complementarios: es posible reducir a la mitad el ancho de banda necesario o se puede doblar la resolución vertical.

III.2. H.120

En 1984, la Unión Internacional de Telecomunicaciones (ITU) estandarizó la recomendación H.120 [44], la primera tecnología de codificación de vídeo estándar. Usaba DPCM (*Differential Pulse Code Modulation*), cuantificación escalar y codificación de longitud variable para transmitir NTSC o PAL a través de líneas dedicadas punto a punto. A partir de este momento, las recomendaciones ITU H.XXX se orientaron a la videoconferencia.

H.120 está obsoleto en la actualidad.

III.3. H.261

En la práctica, la compresión de vídeo digital empieza en 1990 con el estándar H.261 de la ITU [45], cuyo objetivo era transmitir vídeo a través de líneas RDSI, con múltiplos de 64 kbps y una resolución CIF (352x288 píxeles) ó QCIF (176x144 píxeles).

Este estándar constituye un primer esfuerzo en el uso de esquemas de codificación híbrida, que son la base de los codificadores actuales.

Como se verá más adelante, la codificación híbrida combina dos métodos:

1. Estimación del movimiento entre imágenes consecutivas y compensación mediante una predicción obtenida a partir de imágenes anteriores.
2. Codificación de la diferencia entre imágenes mediante la decorrelación de la información en el dominio espacial. Esta decorrelación se consigue mediante una transformación 2D al dominio de la frecuencia. La información transformada se cuantifica mediante un proceso en el que se pierde parte de la información. Posteriormente, esta información se comprime sin pérdidas mediante códigos de Huffman o codificación aritmética.

H.261 utiliza un muestreo 4:2:0, que implica que hay el doble de muestras de luminancia que de crominancia. Esto aprovecha el hecho de que el ojo humano es más sensible a la intensidad de la luz que al color.

Este estándar usa macrobloques de 16x16 píxeles con compensación de movimiento, una transformada DCT, cuantificación escalar, escaneado en zig-zag y codificación de Huffman (codificación entrópica mediante códigos de longitud variable).

H.261 fue sustituido por H.263 (publicado en 1995), pero sus principios de funcionamiento siguen siendo la base de las siguientes generaciones de codificadores de vídeo digital.

III.4. Motion JPEG

JPEG [46] es un método muy extendido de codificación de imágenes estáticas, estandarizado en 1992.

Motion JPEG codifica el vídeo como una secuencia de imágenes independientes, con lo que no utiliza compensación de movimiento. Esto hace que no se pueda reducir la redundancia temporal y compromete en gran medida la capacidad de compresión de este método.

A cambio, este tipo de codificación permite el acceso aleatorio a todas las imágenes de la secuencia, así como el avance y retroceso imagen a imagen.

III.5. MPEG-1

El estándar MPEG-1 [47] fue diseñado en 1992 para proporcionar vídeo de calidad aceptable a 1.52 Mbps y con una resolución de 352x288 píxeles. Soporta únicamente vídeo progresivo (no entrelazado), mientras que la mayoría de estándares de difusión son entrelazados. Este hecho lanzó el diseño de MPEG-2 (publicado en 1993), que incluye el soporte de vídeo entrelazado.

Añadió la predicción de movimiento bidireccional, la estimación de movimiento con precisión de medio pixel, las matrices de ponderación de la cuantificación, etc.

III.6. MPEG-2 / H.262

El estándar MPEG-2/H.262 [48] fue diseñado conjuntamente por la ISO y la ITU y aprobado en 1993. El estándar soporta definición estándar (SD, 720x576 píxeles) y alta definición (HD, 1920x1080 píxeles).

En el diseño se llegó a compromisos entre la calidad y la complejidad computacional para la codificación, con márgenes entre 3 y 10 Mbps para SD.

Además, un decodificador MPEG-2 es compatible con un codificador MPEG-1 estándar e introduce varias formas de escalabilidad (SNR, espacial, etc.).

III.7. Digital Video (DV)

El estándar DV [49], desarrollado por IEC en 1994, se enfoca principalmente al almacenamiento del *bitstream* de vídeo en cinta. No utiliza compensación de movimiento, codificando las imágenes individuales a 25 Mbps. Junto con la codificación del sonido y otra información auxiliar da una tasa binaria muy alta, de 36 Mbps.

Estos datos lo hacen mejor que Motion JPEG y similar a la codificación Intra de MPEG-2/H.262.

III.8. H.263

El estándar H.263 [50], desarrollado por la ITU en 1995, es un gran paso adelante en los estándares de codificación orientados a la videoconferencia en entornos móviles.

Especialmente en el caso del vídeo progresivo, la calidad de H.263 es muy superior a los estándares anteriores. En bajas tasas binarias, incluso dobla en calidad a MPEG-2/H.262.

H.263 se usa en H.324, H.323 y H.320, que son estándares para videoconferencia, así como en vídeo distribuido a través de la web.

El proyecto de estandarización de teléfonos móviles del 3GPP incluía H.263 para la transmisión de vídeo desde y hacia teléfonos móviles.

Algunas novedades introducidas por las diferentes versiones de H.263 incluyen la codificación de longitud variable de los coeficientes de la DCT, imágenes PB bidireccionales, codificación aritmética entrópica, etc.

Al H.263 le siguieron H.263+ y H.263++. El primero introduce resistencia frente a errores, mejora la eficiencia de compresión, añade escalabilidad e información suplementaria.

Por último, H.263++ introduce varios anexos con mejoras como la selección de imagen de referencia, la eliminación de las variaciones en la transformada inversa y otros métodos de mejora de la fiabilidad.

III.9. RealVideo

RealNetworks fue una de las primeras empresas en vender herramientas para el *streaming* de audio y vídeo a través de Internet. La primera versión del códec RealVideo [51], introducida en 1997, estaba basada en H.263. Las siguientes versiones ya utilizaron códecs propietarios, hasta llegar a la versión 10 en 2010.

III.10. MPEG-4 SP / ASP

La estandarización de MPEG-4 [52] comenzó en 1995 y ha evolucionado incluyendo nuevos perfiles y otros conceptos relacionados con el vídeo (gráficos en 3D, codificación escalable, etc.).

Los esfuerzos de estandarización posteriores se han enfocado en mejorar la capacidad de compresión del vídeo.

El estándar MPEG-4 se diseñó para permitir un amplio rango de relaciones entre calidad y tasa binaria, lo que se consigue mediante los distintos perfiles:

- El perfil *Simple* es muy similar a H.263.
- El *Advanced Simple Profile* (ASP) añade soporte para vídeo en definición estándar, vídeo entrelazado y algunas herramientas de compresión extra, como estimación de movimiento con resolución de un cuarto de píxel y compensación de movimiento global.

III.11. DivX / XviD

El popular estándar de formato de fichero de codificación propietario DivX [53][54] está basado en el perfil MPEG-4 ASP.

III.12. On2 VPx

On2 es una empresa que diseña tecnología de codificación y decodificación de vídeo, licenciándola a sus clientes. Los códecs diseñados por On2 son conocidos como VP3 a VP7.

En 2004, Macromedia seleccionó VP6 para ser usado como códec de vídeo en Flash 8, utilizado posteriormente en servicios de vídeo como *Youtube* o *Google Vídeo*. También empresas como *Skype* o *XM* han utilizado esta tecnología para diversos propósitos.

III.13. WMV9 / VC-1

Microsoft desarrolló Windows Media Video 9. Este códec fue inicialmente propietario, pero posteriormente fue estandarizado por SMPTE, que lanzó formalmente VC-1 [55] en 2006.

Se trata de un códec híbrido que utiliza la compensación de movimiento tradicional, la transformada DCT y la codificación de Huffman. Por lo tanto, es similar a H.263 y MPEG-4. Sin embargo, WMV9/VC-1 utiliza bloques de 4x4 píxeles, más pequeños que los bloques de 8x8 utilizados en los otros estándares.

Inicialmente se consideró VC-1 como una alternativa al códec de la ITU-T, H.264/MPEG-4 AVC, pero se ha demostrado que la capacidad de compresión de VC-1 es menor. En cualquier caso, este códec es computacionalmente menos exigente que el de la ITU, y fue adoptado por HD DVD y Blu-ray Disc.

Así como otros estándares (RealVideo, DivX ó On2) han sido diseñados para evitar el pago de costosas tecnologías de terceros, WMV9 no proporciona este beneficio extra.

III.14. H.264 / MPEG-4 part 10 / AVC

Mientras los comités de estandarización de MPEG-4 se centraron en nuevas técnicas de codificación, H.264 se centró en técnicas más tradicionales.

El objetivo fue comprimir vídeo al doble de la tasa compresión de los estándares anteriores, a la vez que se mantuviera la misma calidad de imagen.

Inicialmente, el estándar era conocido como H.26L ó JVT (*Joint Video Team*, unión de la ISO y la ITU). Finalmente, H.264 es el nombre del estándar

de la ITU mientras que MPEG-4 part 10, Advanced Video Coding (AVC) es el nombre de la ISO [9].

Debido a sus mejoras en la compresión y calidad, rápidamente se convirtió en el estándar dominante, siendo adoptado por gran cantidad de aplicaciones, fabricantes y estándares (incluyendo DVB-H, o emisión de televisión para dispositivos portátiles).

Las aplicaciones orientadas al entorno móvil utilizan principalmente el perfil *Baseline*, con resoluciones hasta SD, mientras que las aplicaciones de codificación de alto nivel utilizan el perfil *Main* o el *High*, que llegan hasta resoluciones de HD. El perfil Baseline no soporta entrelazado, mientras que los perfiles superiores sí que lo soportan.

III.15. H.265 / MPEG-H part 2 / HEVC

El estándar HEVC (*High Efficiency Video Coding*) es el proyecto más reciente de la colaboración entre la ITU-T y la ISO para la codificación de vídeo [35][56]. Esta colaboración, plasmada en el *Joint Collaborative Team on Video Coding* (JCT-VC), dio como resultado el estándar de codificación de vídeo más avanzado hasta la fecha.

Este estándar se ha diseñado para soportar todas las aplicaciones actuales de H.264/MPEG-4 AVC y para poner el foco en dos aspectos importantes: aumentar la resolución y las arquitecturas de procesamiento en paralelo.

La capa de codificación de vídeo está basada en los conceptos propios de la codificación híbrida (compensación de movimiento basada en bloques), pero con un conjunto de mejoras sustanciales que le permiten conseguir un ahorro

del 50% de la tasa binaria para una calidad perceptual equivalente con respecto a los estándares anteriores (especialmente para vídeo de la más alta resolución).

Además, la complejidad computacional de la decodificación no es sustancialmente superior a la de H.264.

IV. Codificación de vídeo híbrida

Como se ha visto en el apartado anterior, todos los estándares de codificación de vídeo surgidos a partir de los años 90 se basan en el mismo principio de funcionamiento: la codificación híbrida.

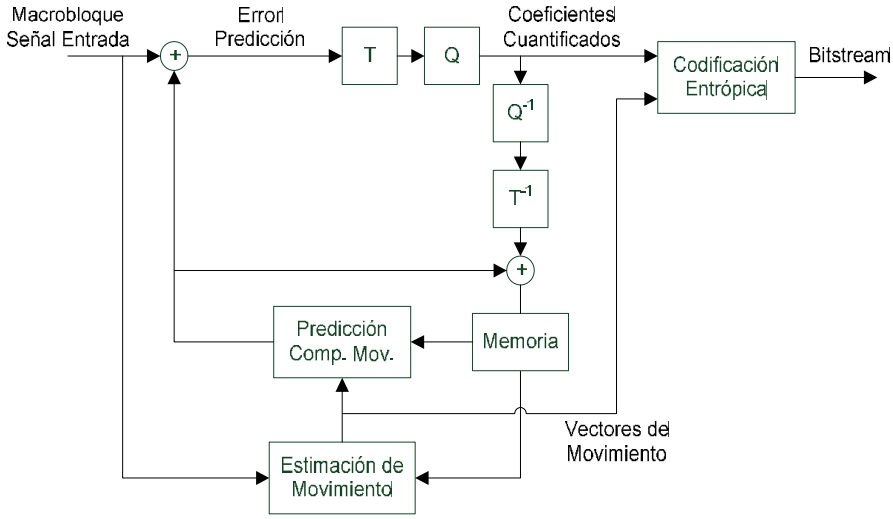


Fig. 1: Diagrama de bloques de un codificador de vídeo digital genérico (codificador híbrido)

La mayoría de los estándares de codificación de vídeo digital se basa en el mismo modelo genérico [2], representado en la Fig. 1. Este modelo, denominado híbrido, incorpora una primera etapa de estimación y

compensación de movimiento (eliminación de la redundancia temporal, o DPCM), una segunda etapa de transformación, que reduce la redundancia espacial y un paso final de codificación entrópica.

Estos codificadores dividen cada imagen de la secuencia en bloques (generalmente de 16x16 píxeles), denominados macrobloques, y aplican las diferentes herramientas de codificación a estas unidades básicas.

Los algoritmos de compresión de vídeo utilizan técnicas de codificación diferencial (predicción *inter-frame*) para reducir la información redundante entre una serie de fotogramas. Estas técnicas de codificación, en las que un fotograma se compara con un fotograma de referencia y sólo se codifica la información que ha cambiado con respecto al fotograma de referencia, son la base de los estándares de codificación de vídeo actuales.

Por lo tanto, la reducción de la redundancia temporal se consigue mediante la codificación de la diferencia entre el macrobloque actual y una predicción del mismo, obtenida mediante técnicas de compensación de movimiento (buscando en imágenes codificadas anteriormente la región más parecida al macrobloque actual). A la información de este error de predicción codificado se añade el vector que permite obtener el desplazamiento necesario para localizar el macrobloque predicción en la imagen de referencia (a este vector se le llama vector de movimiento del macrobloque).

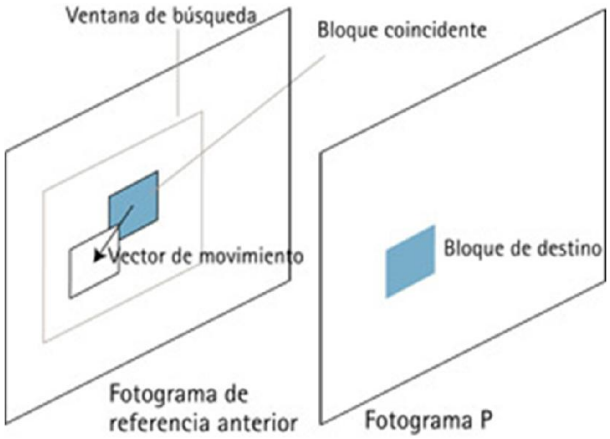


Fig. 2: Vector de movimiento en predicción *inter-frame*

En la Fig. 2 se puede observar el concepto de vector de movimiento. Para predecir el bloque destino se utiliza el bloque coincidente en la imagen anterior junto con el vector que permite encontrar la posición relativa del bloque coincidente respecto al bloque destino en la imagen actual. Como se puede ver, la búsqueda del bloque coincidente se realiza dentro de una ventana de búsqueda para limitar la complejidad computacional de buscar en toda la imagen previa.

Por su parte, la redundancia espacial se reduce mediante una transformación de bloque (T en la Fig. 1) que compacta la energía del macrobloque en unos pocos coeficientes, cuya cuantificación proporciona la mayor parte de la compresión. Finalmente, los índices de cuantificación (Q) de los coeficientes transformados, junto con la información de los vectores de movimiento se codifican entrópicamente, utilizando códigos de longitud variable o codificación aritmética.

Esta estructura, denominada de codificación *inter-frame*, es muy eficiente cuando la redundancia entre imágenes consecutivas es grande, pero si las técnicas de compensación de movimiento no son capaces de obtener buenas predicciones para codificar los macrobloques de la imagen, la eficiencia de codificación disminuye. En estos casos es más eficiente utilizar codificación *intra-frame*, que explota únicamente la redundancia espacial del macrobloque a codificar. Por lo tanto, cuando una imagen se puede representar utilizando básicamente información procedente de la imagen anterior, el número de macrobloques codificados con predicción *intra-frame* (macrobloques *intra* o IMB) es muy reducido. Por el contrario, cuando la correlación entre imágenes disminuye, el número de IMB necesarios para codificar la imagen aumenta. Así, el número de macrobloques *intra* de una imagen proporciona una medida indirecta de la correlación entre imágenes consecutivas de la secuencia.

El principal problema del modelo de codificación diferencial radica en su comportamiento ante la aparición de errores en las imágenes decodificadas. Debido al uso de predicciones, un error en el *bitstream* o en el proceso de decodificación se propagará indefinidamente en los siguientes *frames* decodificados. Para solucionar este problema, se definen dos tipos básicos de imágenes codificadas: las imágenes *inter* (*P-frames* ó *B-frames*) y las imágenes *intra* (*I-frames*) [4]. Las primeras hacen uso del esquema de codificación diferencial anterior, basado en la predicción y la compensación de movimiento, permitiendo el uso de macrobloques *intra* cuando dicha predicción no sea posible o no sea eficiente. Por su parte, las imágenes *intra* no utilizan predicción temporal, sino que todos sus macrobloques se codifican en modo *intra*, eliminando su dependencia de imágenes anteriores, estableciendo puntos de sincronización y de acceso aleatorio, y cortando la propagación indeseada de errores procedentes de imágenes anteriores. Por este motivo, las imágenes *intra*

reciben el nombre de *keyframes*, ya que constituyen puntos clave en la secuencia codificada. La contrapartida de esta solución es que la codificación *intra* es menos eficiente que la *inter*, por lo que las imágenes *intra* necesitan generalmente una mayor cantidad de bits para su codificación.

La combinación de estos tres tipos de imágenes en una secuencia codificada recibe el nombre de *Group Of Pictures* (GOP).

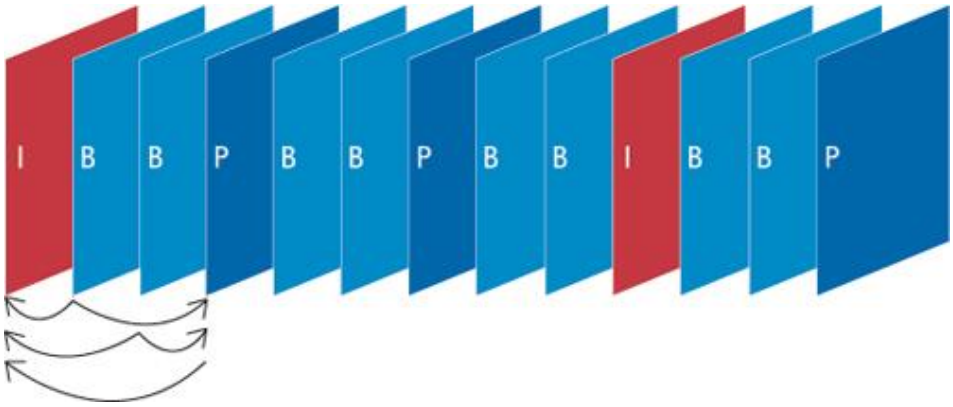


Fig. 3: Ejemplo de *Group of Pictures* (GOP)

En la Fig. 3 se puede apreciar un ejemplo de GOP en el cual, tras una imagen I se repite una sucesión de dos imágenes B y una imagen P. Además, se puede apreciar el sentido de la predicción utilizada para codificar cada tipo de imagen (B ó P).

Un GOP puede presentar distintas configuraciones: si el número y orden de imágenes *inter* entre cada dos imágenes *intra* consecutivas es fijo, el GOP es cerrado, mientras que si la distancia entre imágenes I consecutivas es variable, el GOP es abierto. La mayoría de codificadores de este tipo utilizan un esquema

fijo de inserción periódica de imágenes *intra*, dando como resultado una estructura de GOP cerrado, pero esta solución es poco eficiente, porque no tiene en cuenta las características de las imágenes que se codifican en modo *intra*. Como se verá más adelante, la selección de las posiciones apropiadas para la inserción de imágenes *intra* puede mejorar la eficiencia de compresión de la señal de vídeo [5].

V. El estándar H.264/MPEG-4 AVC

De entre todos los estándares de codificación de vídeo descritos anteriormente, el H.264 se ha escogido como base para los desarrollos descritos en la presente tesis.

En esta sección se presenta una descripción más detallada de las características de este estándar y de las novedades que presenta respecto a los paradigmas de codificación existentes anteriormente [4].

A principios de 1998, el *Video Coding Experts Group* (VCEG) ITU-T SG16 Q.6 publicó una petición de propuestas para un proyecto denominado H.26L, con el objetivo de doblar la eficiencia de codificación de los estándares anteriores en cualquier aplicación (conseguir la misma calidad con la mitad de tasa binaria).

El primer borrador fue aprobado en Octubre de 1999, y en Diciembre de 2001 el VCEG y el *Moving Picture Experts Group* (MPEG ISO/IEC JTC 1/SC 29/WG 11 formaron el *Joint Video Team* (JVT), cuyo objetivo fue aprobar el estándar H.264/AVC en Marzo de 2003.

Para permitir un amplio grado de aplicabilidad, el estándar define dos capas distintas:

- **VCL (*Video Coding Layer*)**: diseñada para representar eficientemente el contenido del vídeo.
- **NAL (*Network Abstraction Layer*)**: diseñada para formatear la representación del vídeo y para proporcionar cabeceras con información necesaria y suficiente para distintos tipos de sistemas de transporte y almacenamiento del vídeo codificado.

En la Fig. 4 se puede observar el diagrama de bloques de un codificador H.264 genérico, donde se aprecia su estructura, con algunas pequeñas diferencias respecto al codificador genérico mostrado en el apartado anterior.

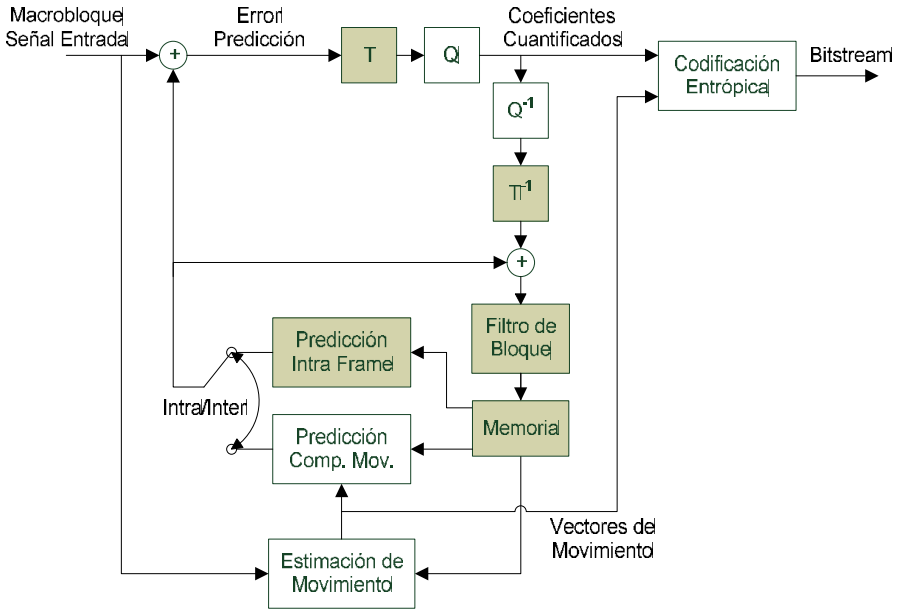


Fig. 4: Diagrama de bloques de un codificador H.264

En la figura se puede apreciar el esquema básico de transformación y cuantificación del error de predicción, seguido de la codificación entrópica del resultado. Así mismo, también aparece el bucle cerrado de predicción, junto con la estimación y compensación de movimiento.

En general, se pueden identificar mejoras en tres campos fundamentales con respecto a los estándares de codificación anteriores (esencialmente MPEG-2): mejoras en la predicción, mejoras de la eficiencia de codificación y mejoras en la robustez y la flexibilidad.

V.1. Mejoras en la predicción

Las siguientes mejoras destacan a la hora de proporcionar una predicción óptima de los macrobloques codificados:

a) Compensación de movimiento con tamaño de bloque variable y tamaño pequeño de bloque:

Este estándar soporta un tamaño de bloque para compensación de movimiento de 4x4 píxeles, más pequeño que todos los estándares anteriores. Además, permite una gran variedad de combinaciones de formas y tamaños para realizar la compensación de movimiento.

b) Compensación de movimiento con precisión de un cuarto de píxel:

Hasta este momento, los estándares utilizaban una precisión de medio píxel para los vectores de movimiento. Esta precisión se duplica hasta el cuarto de

píxel para la compensación de movimiento, a la vez que se reduce la complejidad del procesado de interpolación requerido por versiones anteriores.

c) Vectores de movimiento más allá de los límites de las imágenes:

Los bordes de las imágenes decodificadas previamente se extrapolan para permitir a los vectores de movimiento apuntar fuera de los límites reales de las imágenes. Esta técnica aparece por primera vez en H.263.

d) Compensación de movimiento con múltiples imágenes de referencia:

Tanto para la codificación de imágenes P como de imágenes B, H.264 permite utilizar múltiples imágenes de referencia, en comparación con estándares anteriores que sólo permitían una imagen anterior (I ó P) para la predicción de imágenes P y una imagen anterior y otra posterior para la codificación de imágenes B.

En este nuevo esquema se dispone de un buffer de imágenes decodificadas de entre las que seleccionar la mejor predicción posible.

e) Separación del orden de referencia y el orden de representación:

En los estándares anteriores el orden en el que se utilizaban las imágenes de referencia y el orden de representación estaban muy encorsetados. En este nuevo estándar se eliminan casi todas las restricciones, de forma que el codificador puede escoger con mayor libertad la ordenación de las imágenes para la codificación.

La única limitación la introduce el decodificador, con lo que dependiendo de las características de la aplicación a la que se destine el vídeo esta libertad podrá ser mayor o menor.

f) Separación de los métodos de representación de imágenes de las capacidades de referencia de imágenes:

En este nuevo estándar es posible utilizar todas las imágenes como referencia, eliminando restricciones anteriores que impedían utilizar imágenes B como referencia.

g) Predicción ponderada:

La predicción ponderada permite ponderar y añadir un offset a la señal de predicción compensada en movimiento. Esta ponderación la controla el codificador y es muy útil a la hora de codificar situaciones excepcionales como fundidos o disoluciones.

h) Inferencia de movimiento mejorada (skipped y direct):

En estándares anteriores, las áreas "*skipped*" no podían contener movimiento, lo que era muy negativo en escenas con movimiento global. Esta limitación se ha eliminado.

Además, en imágenes bidireccionales se añade la inferencia directa de movimiento, heredada de H.263+ y MPEG-4 Visual.

i) Predicción espacial direccional para codificación intra:

En este nuevo estándar se realiza la predicción incluso para macrobloques *intra*. En dichos macrobloques se utiliza como información de predicción los macrobloques previamente codificados de la misma imagen a la que pertenece el macrobloque actual. En la Fig. 5 se muestran los 9 modos de predicción *intra* de bloques 4x4 disponibles. Además, se definen 4 modos de predicción *intra* para bloques de 16x16 píxeles (Fig. 6).

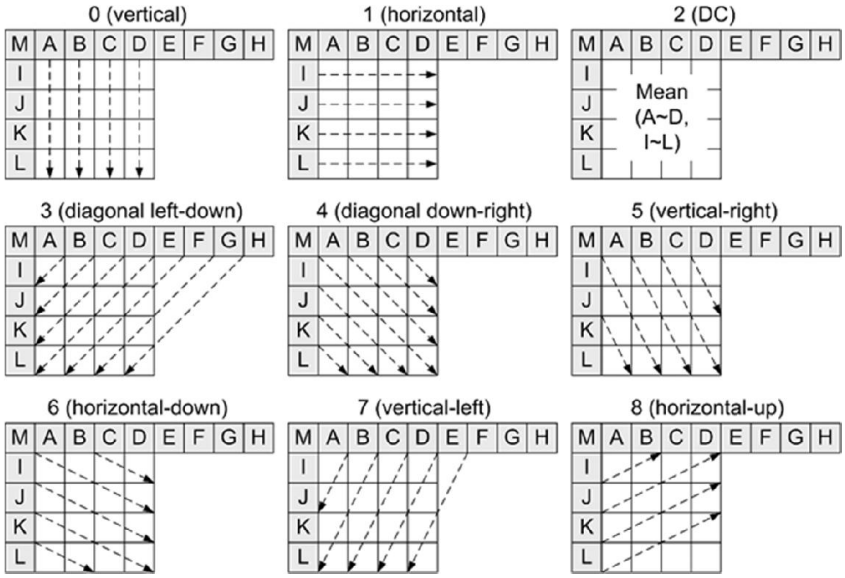


Fig. 5: Modos de predicción Intra-Frame 4x4 en H.264

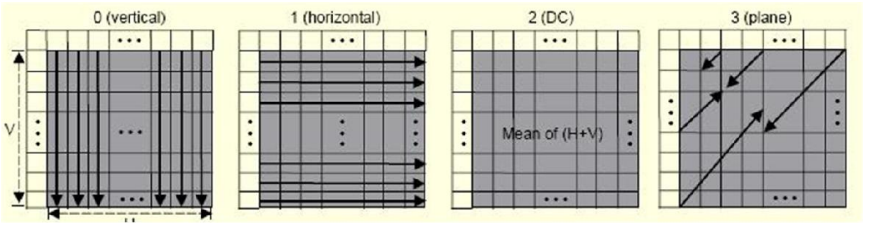


Fig. 6: Modos de predicción Intra-Frame 16x16 en H.264

j) Predicción Inter-Frame con particiones múltiples:

La predicción compensada en movimiento permite el uso de un tamaño adaptativo de bloque, proporcionando una gran versatilidad a la hora de maximizar la compresión de la señal. En concreto, se pueden aplicar distintas particiones al macrobloque según varios patrones para generar la mejor predicción posible del macrobloque objetivo. Esto se consigue mediante particiones mixtas de 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 y 4x4. Las distintas particiones posibles se muestran en la Fig. 7.

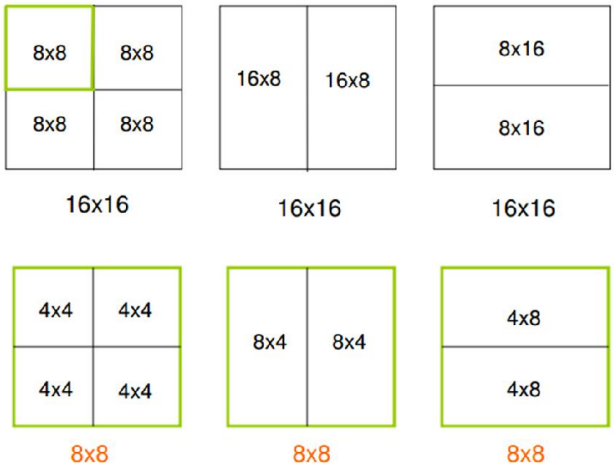


Fig. 7: Partición de macrobloques 16x16 para predicción *inter-frame*

k) Filtro de eliminación del efecto de bloque en el bucle:

La codificación basada en bloques produce artefactos en la señal codificada. Estos artefactos se pueden producir tanto por la predicción como por la codificación del residuo.

El filtro de eliminación del efecto de bloque mejora la calidad tanto objetiva como subjetiva del vídeo codificado, y en el caso del filtro de H.264 nos encontramos ante una evolución de la funcionalidad opcional presente en H.263+.

Este filtro se introduce en el bucle de predicción *inter-frame*, de forma que se mejora también la calidad de la predicción en imágenes anteriores, como se puede ver en la Fig. 4.

V.2. Mejoras en la eficiencia de codificación

Las siguientes mejoras ayudan a ofrecer una mayor eficiencia de codificación:

a) Transformada de bloque de pequeño tamaño:

Todos los estándares anteriores utilizaban una transformada de bloque de tamaño 8x8, mientras que H.264 introduce una transformada con tamaño de bloque de 4x4 píxeles.

Esto permite la representación más precisa de detalles locales, reduciendo los artefactos (*ringing*). Además, la utilización de un menor tamaño de bloque permite aprovechar mejor las ventajas introducidas por las mejoras en la predicción descritas anteriormente.

b) Transformada de bloque jerárquica:

La transformada 4x4 es beneficiosa en la mayor parte de las situaciones, pero en ciertos casos es más útil utilizar un tamaño mayor. Este es el caso de las imágenes que contienen suficiente correlación como para utilizar unas funciones base más grandes.

Esto se consigue mediante dos métodos: utilizando una transformación jerárquica para extender el tamaño de bloque efectivo para la información de baja frecuencia de la crominancia (8x8) y permitiendo al codificador utilizar un modo de codificación espacial para *intra*, que permite la extensión del tamaño de bloque para zonas de baja frecuencia en la luminancia (tamaño 16x16).

c) Transformación de palabra corta:

La aritmética requerida para realizar todas las operaciones necesarias en el proceso de codificación es de sólo 16 bits, en comparación con los 32 bits necesarios en estándares anteriores.

Esto reduce la complejidad computacional y los requisitos hardware para realizar la codificación.

En general, sólo se utilizan sumas y desplazamientos para llevar a cabo todas las operaciones necesarias en la transformación.

d) Transformada inversa con ajuste exacto:

En estándares anteriores no era posible asegurar que la transformada inversa fuese exactamente igual a la señal original, lo que producía una desviación de la señal decodificada que introducía nuevos artefactos. Además, esta desviación era distinta en cada implementación del decodificador, lo que hacía el problema todavía mayor.

En H.264 se consigue proporcionar un método de cálculo de la transformada inversa sin pérdidas, con lo que la reconstrucción de la señal transformada es exacta.

e) Codificación aritmética entrópica:

La codificación aritmética se introdujo en H.263, pero H.264 introduce un nuevo método de codificación, denominada entrópica, que es mucho más efectiva. En concreto, se trata de los códigos CABAC (*Context-Adaptive Binary Arithmetic Coding*).

f) Codificación entrópica adaptativa al contexto:

H.264 presenta dos métodos de codificación entrópica. El ya mencionado CABAC y el nuevo CAVLC (*Context-Adaptive Variable-Length Coding*), que mejoran el comportamiento respecto a anteriores estándares.

V.3. Mejoras en la robustez y la flexibilidad

El último grupo de mejoras introducido por H.264 se refiere a la robustez frente a la pérdida o corrupción de los datos codificados y al funcionamiento sobre una amplia variedad de entornos.

a) Estructura de grupos de parámetros:

La información de gestión de la codificación, especialmente crítica ya que permite configurar correctamente el decodificador, se gestiona de forma específica en este estándar, mejorando la robustez y permitiendo una mayor protección de esta información.

b) Estructura de unidad sintáctica NAL:

Cada unidad sintáctica se introduce en un paquete de datos denominado unidad NAL. Esta estructura es muy flexible, y permite que se adapte a cada aplicación concreta, incluyendo distintos tipos de información y empaquetado.

c) Tamaño de slice flexible:

El tamaño de los *slices* (fragmentos de imágenes) es totalmente flexible, lo que aumenta la eficiencia de codificación. Esto es así debido a que se puede configurar de forma más independiente la forma en que se codifica cada uno de los *slices*, que puede contener características de vídeo diferenciadoras que pueden ser explotadas por el codificador.

d) Ordenación flexible de macrobloques (FMO):

Cada *slice* es una entidad que se puede decodificar de forma independiente. Esto se puede explotar a la hora de gestionar la pérdida de datos mediante la correcta relación espacial entre regiones codificadas en cada *slice*.

e) Ordenación arbitraria de slices:

El hecho de que cada *slice* sea prácticamente independiente del resto para su decodificación hace que se puedan enviar desordenados para mejorar la latencia en el proceso de decodificación.

f) Imágenes redundantes:

Para mejorar la robustez frente a la pérdida de datos se introduce la posibilidad de enviar fragmentos de las imágenes codificadas de forma redundante, normalmente con una calidad menor, para permitir sustituir fragmentos perdidos de la imagen original con los fragmentos redundantes.

g) Particionado de datos:

Para permitir una codificación especial de la información más crítica (vectores de movimiento, información de predicción, etc.) que condiciona la decodificación, se particiona la información transmitida en tres particiones distintas. Estas particiones incluyen distintos tipos de información sintáctica.

a) Imágenes de sincronización/alternancia (SP/SI):

Se trata de dos nuevos tipos de imágenes que permiten la sincronización exacta del proceso de decodificación entre varios decodificadores, cuando algunos de los cuales ya se encuentran en el proceso de decodificación y otros necesitan engancharse a dicho proceso. En estos casos, sin la necesidad de enviar una costosa imagen *Intra*, se puede enviar una imagen SP/SI y permitir el funcionamiento simultáneo de ambas cadenas de decodificación.

Este procedimiento se puede utilizar también para alternar entre varias versiones del mismo vídeo a distintas calidades, para recuperarse ante un escenario de error o pérdida de datos, o para mejorar el funcionamiento de las herramientas como el *fast-forward* o el rebobinado.

VI. El estándar H.265/MPEG-H HEVC

H.264 ha desplazado a todos los estándares anteriores en todos sus ámbitos de aplicación. Se usa de forma masiva en todo tipo de aplicaciones, incluyendo la transmisión de televisión de alta definición (HD), transmisión por satélite, cable y televisión terrestre, sistemas de captación y edición de vídeo, cámaras, aplicaciones de seguridad, internet y vídeo a través del móvil, Bluray,

aplicaciones conversacionales en tiempo real como video chats, video conferencias y sistemas de telepresencia [34][35].

Sin embargo, el rango de aplicaciones ha aumentado en los últimos años, lo que requiere un nuevo conjunto de características a cubrir por estos estándares. Entre estas aplicaciones se pueden incluir mayores resoluciones por encima del HD (4k, 8k, etc.), aplicaciones 3D, captura estereoscópica o multi-vista, vídeo bajo demanda, etc.

Para cubrir estas necesidades se ha desarrollado el estándar HEVC, enfocándose en el campo ya cubierto por H.264 y ampliándolo en dos aspectos clave: mayor resolución de vídeo y mayor uso de arquitecturas de procesado en paralelo.

Tanto en el caso de H.264 como en el de HEVC, el estándar sólo define la estructura del flujo binario y la sintaxis, así como los requisitos a cumplir por el flujo binario y su mapeado para la generación de imágenes decodificadas. De esta forma, cualquier decodificador que se adapte al estándar producirá la misma salida al decodificar un flujo que cumpla los requisitos anteriores.

Esta definición permite optimizar las implementaciones para adaptarse a cualquier aplicación, pero no garantiza la calidad del vídeo reconstruido tras la decodificación.

Para ayudar al desarrollo de estas aplicaciones, además del estándar, se proporciona una implementación de referencia del decodificador. Además, un conjunto de vídeos de test para comprobar la codificación/decodificación también está disponible.

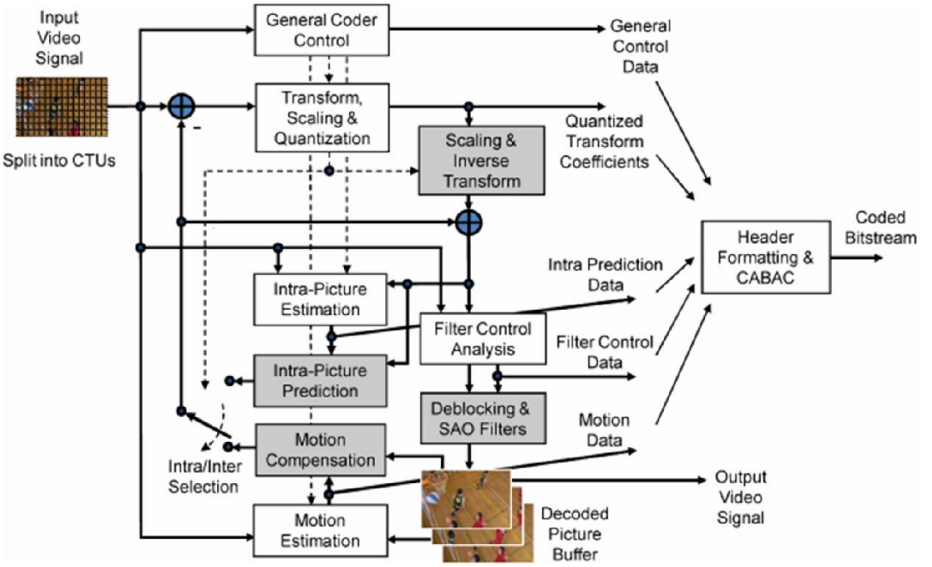


Fig. 8: Diagrama de bloques de un codificador HEVC genérico

En la Fig. 8 se puede observar un diagrama de bloques del codificador HEVC genérico. En dicha figura se aprecia cómo HEVC utiliza el mismo esquema de codificación híbrida que H.264 (predicción *inter/intra*-imagen y codificación de la transformación 2-D). En este modelo siempre se trabaja con la información residual de la predicción (espacial y/o temporal), donde el codificador y el decodificador generan exactamente las mismas predicciones entre imágenes utilizando compensación de movimiento y decisión del modo de codificación, que se envía junto con el flujo de información del vídeo codificado.

El residuo de estas comparaciones se transforma mediante una transformación lineal espacial, donde los coeficientes son escalados,

cuantificados y codificados entrópicamente para ser transmitidos junto a la información de la predicción.

En HEVC el macrobloque es sustituido por otra unidad de codificación: el *Coding Tree Unit* (CTU), cuyo tamaño escoge el codificador, pudiendo ser mayor que el macrobloque. Consiste en un CTB de luminancia y los correspondientes CTBs de crominancia, junto con los elementos sintácticos necesarios. El tamaño LxL puede ser de 16, 32 ó 64 píxeles, con tamaños mayores permitiendo mejores compresiones, especialmente a altas resoluciones. Además, se soporta la partición de los CTBs en tamaños más pequeños en una estructura en árbol.

En la Fig. 9 se puede observar una comparación entre las particiones de macrobloques utilizadas en H.264 y la nueva estructura en CTUs de H.265.

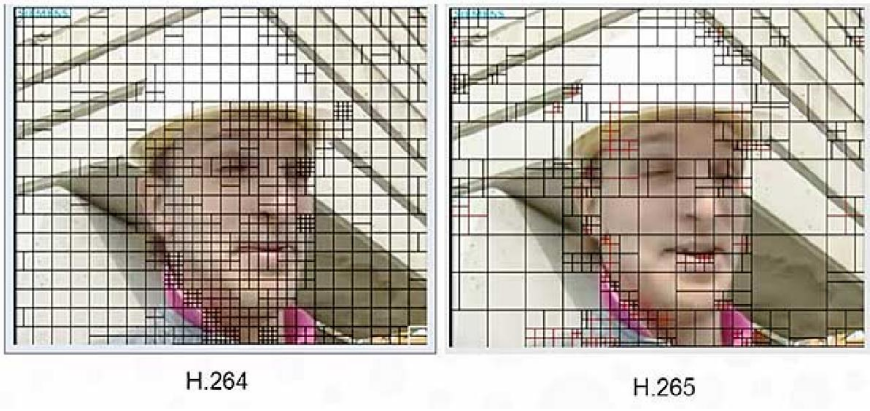


Fig. 9: Comparación entre macrobloques en H.264 y CTU en H.265

En este estándar existen además unidades y bloques de codificación (CBs y CUs) y unidades y bloques de predicción (PUs y PBs). La decisión de codificar una sección de imagen en modo *intra* o *inter* se realiza a nivel de CU, que

pueden ser a su vez sub-particiones de CTUs. Por otra parte, los bloques de codificación (CBs) de luminancia y crominancia se dividen y predicen a partir de los bloques de predicción (PBs). El tamaño de estos PBs va de 4x4 a 64x64 píxeles, de forma que un CU se puede dividir en una, dos o cuatro bloques de predicción, como se puede ver en la Fig. 10.

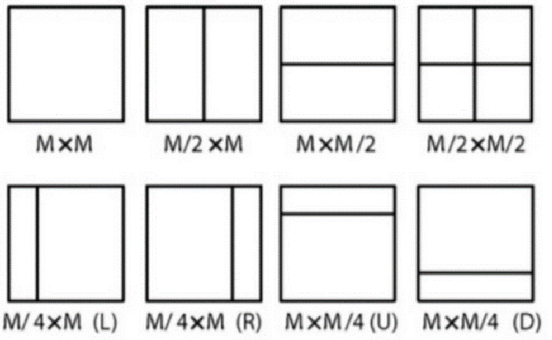


Fig. 10: Partición de CB en PBs

La transformación también se realiza en bloques (TUs), de nuevo a nivel de CU, donde puede haber distintos tamaños de transformación para la luminancia y la crominancia. Los bloques van desde 4x4 a 32x32 píxeles.

Se utiliza predicción avanzada de vectores de movimiento, utilizando los PBs candidatos más probables basados en la información co-localizada en la imagen anterior.

La compensación de movimiento se realiza con una precisión de un cuarto de píxel, utilizando filtros de 7 u 8 coeficientes (en comparación con los filtros de 6 coeficientes en H.264).

Existen 33 modos de predicción *intra-frame* en comparación a los 8 de H.264, y además se deriva el modo más probable a partir de los bloques co-localizados en imágenes anteriores, lo que mejora la eficiencia de codificación de esta información. En la Fig. 11 se puede ver una comparación entre los modos de predicción en H.264 y H.265.

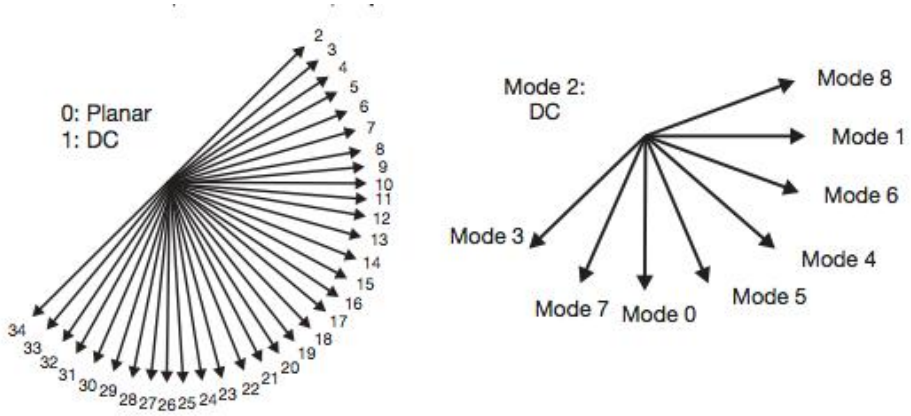


Fig. 11: Modos de predicción intra-frame en HEVC (izquierda) y AVC (derecha)

Como en el caso de H.264, se utiliza la reconstrucción uniforme de la cuantificación, con matrices de cuantificación para varios tamaños de transformada.

La codificación entrópica utiliza codificación aritmética binaria adaptativa al contexto (CABAC), con diversas mejoras respecto a H.264. Estas mejoras radican en la paralelización y la mejora de la tasa binaria, reduciendo también los requisitos en cuanto a la memoria necesaria.

También se usa el filtro de eliminación del efecto de bloque, pero se ha simplificado para hacerlo en una arquitectura paralelizada.

Después del filtro de eliminación del efecto de bloque se incluye un offset adaptativo por muestra (*Sample Adaptive Offset*, SAO), que constituye un mapeo no lineal de la amplitud para reconstruir mejor la amplitud de la señal original. Utiliza tablas de look-up descritas por el codificador mediante el análisis de histogramas.

HEVC incluye nuevos aspectos de diseño que mejoran la flexibilidad y la robustez ante pérdida de datos, aunque la mayoría de funcionalidades se mantienen respecto al estándar H.264.

Entre estas funcionalidades destacan:

- **Conjuntos de parámetros**: *Picture Parameter Set*, *Sequence Parameter Set* y el nuevo *Video Parameter Set*. Proporcionan un mecanismo robusto para compartir datos comunes a lo largo del proceso de decodificación de cada elemento de la secuencia (imágenes, secuencias o vídeos).

- **Unidades Sintácticas NAL**: cada estructura sintáctica se coloca en un paquete lógico de datos denominado unidad NAL (NAL *Unit, Network Abstraction Layer*), mediante las cuales, conociendo únicamente la cabecera, se puede conocer el propósito y el contenido de toda la unidad.

- **Porciones (*Slices*)**: son estructuras de datos que pueden ser decodificados independientemente de otras porciones de la misma imagen. Puede ser una región de una imagen o una imagen completa. Permiten la resincronización en casos de pérdida de datos, ya que son independientes también en términos de predicción, etc.

Se intenta minimizar la sobrecarga de cabeceras en la división de imágenes en porciones.

- **Metadatos SEI (*Supplemental Enhancement Information*) y VUI (*Video Usability Information*)**: incluyen información de temporización, interpretación del espacio de color, información del empaquetado de datos 3-D, etc.

Por último, la capacidad de procesado en paralelo de HEVC incluye cuatro nuevas funcionalidades que también afectan a la capacidad de empaquetado para su transmisión:

- ***Tiles***: Se divide la imagen en rectángulos para ser procesados en paralelo (independientemente de la resincronización ante errores). Se codifican compartiendo información de cabeceras y pueden utilizarse para el acceso aleatorio a distintas regiones del vídeo. No necesitan una sincronización especial para su uso.
- ***Wavefront Parallel Processing* (WPP)**: las porciones se dividen en filas de CTUs, de forma que la primera fila se procesa normalmente, pero la segunda puede empezar a procesarse en cuanto se han procesado dos CTUs de la primera fila. Esto es posible porque la información de contexto para la codificación entrópica es suficiente con sólo dos CTUs de la fila anterior. Esto añade un nivel de granularidad más fino a la paralelización (dentro de una porción). Proporciona mejor compresión que los *tiles* y evita artefactos visuales generados por dichos *tiles*.
- ***Dependent Slice Segments***: permiten encapsular parte de una porción o *wavefront* en una NAL diferente para así hacer que dicha

información esté disponible lo antes posible. Para su decodificación es necesario que parte de la información de la que depende haya sido previamente decodificada.

Las mejoras en compresión proporcionadas por este estándar vienen dadas por los pequeños incrementos en compresión proporcionados por las distintas técnicas aplicadas en cada uno de los procesos de codificación y representación de la información codificada.

Capítulo 2

Detección Automática de Cambios de Plano

I. Introducción

Los codificadores de vídeo actuales intentan mejorar la eficiencia de compresión por distintos medios: explotando la correlación tanto espacial como temporal existente en la señal de vídeo sin codificar; aprovechando las características del sistema visual humano (SVH) para eliminar la información a la cual el SVH es menos sensible; o finalmente, explotando las características de la señal de vídeo derivadas de su contenido y de la forma en que la señal es capturada.

En concreto, la detección automática de cambios de plano es un campo de investigación y aplicación que tiene una relación indirecta con las técnicas anteriores, ya que, pese a no constituir por sí misma una técnica de mejora de la eficiencia de codificación, la correcta aplicación de estos algoritmos de detección puede proporcionar una mejora sustancial de la calidad final obtenida.

En este capítulo se presenta un método de detección automática de cambios de plano basado en la medida de la correlación existente entre imágenes de la secuencia codificada. Este algoritmo proporciona una ganancia en tiempo de procesamiento y es válido para aplicaciones en tiempo real.

Para llevar a cabo la presentación de este novedoso método, el capítulo se organiza de la siguiente forma: en primer lugar se presentan las nociones básicas sobre las que se apoya la detección automática de cambios de plano; a continuación se describe el algoritmo de detección y el entorno de desarrollo y test del mismo. Posteriormente, la sección V presenta la optimización de los valores de los distintos parámetros del algoritmo; y finalmente, los resultados obtenidos son analizados con precisión y se comparan las prestaciones obtenidas con otros métodos de reciente publicación. Dichos resultados llevan a

unas prometedoras conclusiones que dan pie al Capítulo 3 de esta tesis, orientado a la inserción dinámica de *keyframes* en las secuencias de vídeo.

II. Detección automática de cambios de plano

En el lenguaje audiovisual las unidades narrativas principales son las escenas, que están formadas por una sucesión de secuencias. A su vez, una secuencia consiste en un conjunto de planos ordenados formados por fragmentos consecutivos de la escena.

Así, un plano es la parte de una película rodada en una sola toma, de duración arbitraria, y normalmente filmada con una única cámara. Es, por tanto, una secuencia de imágenes que se sitúan entre las operaciones de inicio y final de la grabación de una cámara, que determinan la posición de los límites del plano.

Debido a las características derivadas de la forma en la que se compone un plano, las imágenes que lo componen tienden a tener una elevada correlación entre ellas, mientras que la correlación disminuye entre imágenes pertenecientes a planos distintos. Esto es así debido a la continuidad de dos parámetros físicos fundamentales:

- Continuidad de la escena física que se filma.
- Continuidad de los parámetros físicos de la cámara utilizada (movimiento, foco, zoom, etc.).

En este sentido, la segmentación automática de vídeo en escenas es virtualmente intratable debido al carácter personal y subjetivo de su creación.

Sin embargo, la segmentación en planos se define de forma precisa gracias a las características distintivas de continuidad de los mismos.

Por lo tanto, los planos son la unidad fundamental de organización de secuencias de vídeo y las primitivas para las tareas de recuperación y de anotación y clasificación semántica de vídeo. Como se verá más adelante, las características de los límites entre los cambios de plano definen el tipo de cambio: los cortes son los cambios de plano donde la transición entre planos consecutivos es abrupta, mientras que en las transiciones graduales el cambio de plano sucede durante un número arbitrario de imágenes. Entre las transiciones graduales cabe destacar los desvanecimientos, los fundidos, los difuminados, las cortinillas, los barridos, etc.

En la literatura existe una amplia variedad de métodos de detección de cambios de plano, así como varios artículos basados en la revisión de distintos conjuntos de estos métodos [25]-[28].

II.1. Clasificación de métodos de detección de cambios de plano

Los métodos descritos en estos artículos dividen el proceso de detección de cambios de plano en tres fases:

1. Extracción de características visuales
2. Medida de similitud
3. Detección

Además, estos métodos se pueden clasificar en función del dominio en el que se ejecutan los tres procesos anteriores:

- **Vídeo descomprimido**: requiere del análisis de las imágenes descomprimidas, por lo que deben aplicarse bien a la versión original del vídeo antes de su compresión o bien tras un proceso de codificación y decodificación de la secuencia, lo que puede generar una mayor complejidad computacional de estos métodos. En general, los métodos más habituales, basados en histogramas, detección de bordes, etc. forman parte de este grupo de algoritmos.
- **Vídeo comprimido**: evita la costosa decodificación del vídeo comprimido para realizar la detección. Por lo tanto, pueden aplicarse tanto durante el proceso de codificación como posteriormente. Sin embargo dependen mucho del tipo de compresión y normalmente son menos precisos.

II.1.1. Extracción de características visuales

Distintos tipos de información pueden ser extraídos de la señal de vídeo para realizar diferentes procesados orientados a distintas aplicaciones. En el caso de la detección de cambios de plano, algunas características útiles para determinar la continuidad de las imágenes de un plano son:

a) Intensidad de los píxeles

Los métodos más sencillos utilizan únicamente los valores de intensidad de los píxeles de cada imagen. Estos métodos se aplican en el dominio del vídeo descomprimido.

b) Histograma de color o histograma de color por bloques

Se calcula un histograma con el número de píxeles de cada imagen en cada valor de intensidad, pudiendo ser tanto un histograma en escala de grises como un histograma de color.

Los histogramas por bloques se obtienen dividiendo la imagen original en bloques y calculando el histograma de cada bloque, adaptándose de esta forma a las características locales de cada plano.

Esta característica es sencilla de calcular, y se obtiene a partir de la imagen original o decodificada. Por lo tanto, este método se aplica en el dominio del vídeo descomprimido.

c) Bordes

Se aplica un detector de bordes y el resultado de dicha detección se utiliza como conjunto de características a monitorizar a lo largo de la secuencia. Se aplica sobre las imágenes originales o decodificadas (vídeo descomprimido).

d) Estadísticos

Diferentes medidas estadísticas obtenidas a partir de los valores de los píxeles de los fotogramas de cada plano se pueden utilizar para extraer características de cada imagen.

Estas medidas estadísticas pueden ser la media, la desviación estándar, la mediana, etc., basándose en los valores de intensidad de los píxeles o en algún pre-procesado de los mismos (filtros, etc.).

e) Vectores de Movimiento

Los vectores de movimiento resultantes del proceso de codificación de cada imagen se utilizan como característica significativa de la continuidad del plano. Estos métodos se aplican en el dominio del vídeo comprimido.

f) Otros

Algunos métodos más actuales utilizan métricas más complejas y avanzadas como pueden ser:

- *Scale Invariant Feature Transform*
- *Corner Points*
- *Information Saliency Map*

Sin embargo, parece claro que los métodos basados en características más complejas no son capaces de producir resultados de detección ni siquiera comparables a los métodos más sencillos, que generalmente ofrecen mejores tasas de detección. En general, los métodos más extendidos por su compromiso entre complejidad y efectividad son los basados en la comparación de histogramas.

II.1.2. Medida de similitud entre imágenes

Utilizando las características extraídas anteriormente se aplica una medida para comprobar la continuidad de la métrica seleccionada entre las imágenes de la secuencia.

En general, los algoritmos se dividen en dos grandes grupos, en función de si la comparación se realiza entre parejas de imágenes consecutivas o mediante una ventana deslizante que tiene en cuenta un conjunto de imágenes consecutivas. Estos últimos métodos incorporan información contextual para reducir la influencia del ruido local, pero son más complejos que los basados en la comparación de dos imágenes.

Algunas de estas medidas de similitud son:

a) Diferencias entre píxeles

Estos métodos se aplican en el dominio del vídeo descomprimido, pudiéndose medir la diferencia entre dos imágenes:

- acumulando la diferencia píxel a píxel entre dos imágenes
- contando el número de píxeles cuya intensidad varía por encima de un umbral

b) Diferencias estadísticas

Se divide la imagen en regiones, calculando medidas estadísticas de cada una de dichas regiones: media, desviación estándar, etc. Estas medidas se comparan entre imágenes consecutivas y se analiza la diferencia. Estos métodos se aplican en el dominio del vídeo descomprimido.

c) Diferencias en la compresión

Se utilizan las diferencias en la transformada discreta del coseno (DCT) de las imágenes comprimidas como medida de similitud, evitando la necesidad de descomprimir las imágenes para realizar la detección. Dado que dicha información codifica las diferencias existentes entre imágenes consecutivas, se puede utilizar como medida de la similitud.

d) Diferencias en los bordes

Se comparan los bordes que aparecen o desaparecen entre imágenes consecutivas, con lo que este método se aplica en el dominio del vídeo descomprimido. El *Edge Change Ratio* (ECR) compara el número de bordes que entran y salen en cada imagen, de forma que en los cambios de plano dicha relación adopta unos patrones determinados:

- **Cortes**: picos aislados en el valor del ECR.
- **Desvanecimientos**: los bordes entrantes o salientes van aumentando en función de si es un *fade-in* o un *fade-out*.
- **Disoluciones**: en primer lugar aumentan los bordes salientes del primer plano, y posteriormente aumentan los bordes entrantes del segundo plano.

e) Diferencias en los vectores de movimiento

Se calculan las diferencias entre los vectores de movimiento de imágenes consecutivas, aplicándose en el dominio del vídeo comprimido.

f) Diferencia de histogramas

Se acumula la diferencia clase a clase entre los histogramas de cada pareja de imágenes consecutivas. En este caso, una clase representa uno de los posibles valores que puede tomar un pixel. Así, en una imagen donde cada pixel puede tomar 255 valores distintos (8 bits), el histograma tendrá 255 clases, cada una de las cuales acumulando el número de píxeles de la imagen que toman el valor correspondiente a la clase.

Se aplican en el dominio del vídeo descomprimido y se pueden utilizar los histogramas de gris o de color, RGB, etc.

II.1.3. Detección

Los métodos de detección de cambios de plano mediante el análisis de la similitud entre imágenes de la secuencia se pueden agrupar en dos grandes conjuntos:

a) Detección basada en umbrales

Compara la medida de similitud entre cada par de imágenes con un umbral, detectando el cambio cuando la similitud está por debajo del umbral. El umbral puede ser:

- **Global**: estos métodos utilizan siempre el mismo umbral, normalmente obtenido de forma empírica. Sin embargo, este umbral no tiene en cuenta las variaciones locales del contenido del vídeo, lo que disminuye la precisión de la detección.

- **Adaptativo**: estos métodos utilizan una ventana deslizante para calcular el umbral, con lo que tienen en cuenta el contexto de cada momento de detección. La contrapartida radica en que es necesario conocer las características del vídeo para determinar el tamaño de la ventana deslizante.

- **Combinación de global y adaptativo**: estos algoritmos ajustan un conjunto de umbrales locales utilizando otro conjunto de umbrales globales, que a su vez se obtienen mediante un compromiso entre la precisión y la eficacia (en el siguiente apartado se proporcionan más detalles sobre la medida de prestaciones de detección). Los valores de las funciones locales son complejos de estimar.

b) Detección basada en aprendizaje estadístico (statistical learning)

Afrontan el problema de la detección de cambios de plano como un problema de clasificación en el cual las imágenes son clasificadas como cambio de plano o no en función de un conjunto de características. Se utilizan los siguientes tipos de métodos:

- **Clasificadores basados en aprendizaje supervisado**:

No necesitan establecer los valores de los umbrales y permiten combinar distintos tipos de características para mejorar la precisión de la detección. Sin embargo, dependen en gran medida de la correcta selección del conjunto de entrenamiento para el aprendizaje. Los métodos más utilizados son:

1. *Support Vector Machine (SVM)*: estos métodos son ampliamente usados ya que pueden aprovechar correctamente la información del proceso de entrenamiento y pueden adaptarse eficientemente a gran cantidad de

características mediante las funciones de *kernel*. Además, existen múltiples implementaciones de SVM. A continuación se enumeran varias aproximaciones:

a) Separar los cortes en dos clases (corte y no corte). Se mapean las características en un espacio de muchas dimensiones para disminuir la influencia de los cambios de iluminación y el movimiento rápido de objetos.

b) Utilizar una ventana deslizante para detectar mediante dos clasificadores SVM cortes y transiciones graduales.

c) Extraer varias características de cada imagen y clasificarlas en cortes, transiciones graduales y otros.

d) Combinar SVM con un método basado en umbrales. Obtener los candidatos mediante los umbrales y verificar las detecciones mediante SVM.

2. *Adaboost*: estos métodos pueden lidiar con gran cantidad de características para la detección mediante la clasificación de imágenes en distintas clases (corte/no corte, corte/no corte/transición gradual, etc.).

3. *Otros*: existen otras aproximaciones:

a) K *Nearest-Neighbour* (kNN)

b) *Hidden Markov Model* (HMM)

- **Clasificadores basados en aprendizaje no supervisado**:

No necesitan un conjunto de entrenamiento, pero son poco eficientes en la detección de distintos tipos de transiciones graduales. Se clasifican a su vez en:

1. *Basados en la similitud entre imágenes*: se dividen las medidas de similitud en dos grupos: el grupo con valores bajos de similitud corresponde con los cortes mientras los valores altos se agrupan para formar las imágenes que no contienen cambios de plano.
2. *Basados en imágenes*: tratan cada plano como una agrupación de imágenes que tienen un contenido visual similar.

II.2. Medida de prestaciones

Las medidas más básicas en este tipo de técnicas consisten en el cálculo de tres tasas:

- **Tasa de detección (*hit rate*)**: relación entre el número de entidades detectadas y el número total de entidades a detectar.
- **Tasa de error (*miss rate*)**: relación entre el número de entidades que no se han detectado y el número total de entidades a detectar. Es el valor inverso de la tasa de detección (1 – *hit rate*).
- **Tasa de falsas detecciones (*false hit rate*)**: relación entre el número de falsas detecciones y el número total de entidades a detectar.

Sin embargo, las medidas más utilizadas en aplicaciones de detección y recuperación (*retrieval*) son la precisión y la eficacia, más utilizadas en su nomenclatura inglesa: *precisión* (precisión) y *recall* (eficacia). La eficacia cuantifica la proporción de entidades correctas (cambios de plano en este caso) que son detectadas, mientras que la precisión cuantifica la proporción de las entidades detectadas que son correctas. En una formulación más matemática, los

conceptos fundamentales de precisión (*precision*, *P*) y eficacia (*recall*, *R*) se expresan de la siguiente manera:

$$P = 100 \times \frac{D}{D + D_F}$$

Ec. 1: Precisión (Precision)

$$R = 100 \times \frac{D}{D + D_P}$$

Ec. 2: Eficacia (Recall)

En estas ecuaciones *D* es el número total de detecciones correctas, D_F es el número total de detecciones falsas (detectar un cambio de plano cuando realmente no existe), y D_P es el número total de detecciones perdidas (no detectar un cambio de plano cuando realmente existe uno). Estas magnitudes se presentan en forma de porcentaje y, dada su definición, el comportamiento óptimo se obtiene cuando ambos parámetros toman valores lo más cercanos posible al 100%.

Sin embargo, debido a la relación existente entre ambas métricas, el comportamiento habitual de estos parámetros hace que al aumentar la precisión disminuya la eficacia y viceversa. Esto es así debido a que, al intentar detectar el mayor número posible de cambios de plano correctos (mayor precisión), aumente también el número de falsas alarmas (disminución de eficacia).

Como se verá más adelante, este es el comportamiento habitual de estas medidas cuando se comparan entre ellas.

II.3. Principales métodos de detección de cambios de plano

En esta sección se presenta un breve resumen de algunos métodos de detección de cambios de plano que han demostrado tener unas prestaciones reseñables en esta tarea.

II.3.1. Diferencias entre píxeles

Los primeros métodos que aparecen en la literatura son los más sencillos, basándose exclusivamente en la comparación de imágenes mediante el cómputo acumulado de la diferencia entre los píxeles de imágenes consecutivas. Estos métodos suman la diferencia píxel a píxel y comparan la suma con un umbral.

Una segunda aproximación añade un segundo umbral, de forma que se detecta un cambio de plano contando el número de píxeles que cambian su valor más de un umbral y comparando este número con un segundo umbral para detectar si hay un cambio de plano.

Algunas variaciones de estos métodos utilizan un filtrado 3x3 para suavizar la imagen y disminuir el ruido, de forma que se reduce el efecto del movimiento de la cámara. Estos métodos no son muy prácticos y no producen resultados reseñables.

Otras aproximaciones dividen la imagen en regiones y tratan de encontrar el *match* para cada región, lo que duplica el coste computacional pero permite detectar transiciones graduales.

Por último, el uso de imágenes cromáticas permite detectar transiciones graduales, pero es muy sensible al movimiento de cámara.

II.3.2. Diferencia de histogramas de color

A lo largo de la literatura existen gran cantidad de variantes de los métodos basados en la diferencia de histogramas. Entre ellos, la diferencia de histogramas de color es una de las variantes más fiables. La idea básica es que el contenido en color no cambia rápidamente más que entre planos distintos. Por lo tanto, los cortes (y otras transiciones rápidas) se detectan como picos individuales en la diferencia temporal entre los histogramas de color de imágenes contiguas o situadas a corta distancia temporal.

La métrica considerada consiste en la suma de las diferencias en el número de píxeles de cada color entre dos imágenes consecutivas tomada a lo largo de todos los valores posibles de color.

Esta métrica se describe mediante la siguiente ecuación:

$$CHD_i = \frac{1}{N} \times \sum_{r=0}^{2^B-1} \sum_{g=0}^{2^B-1} \sum_{b=0}^{2^B-1} \left| p_i(r,g,b) - p_{i-1}(r,g,b) \right|$$

Ec. 3: Diferencia de Histograma de Color

En la Ec. 3, $p_i(r,g,b)$ es el número de píxeles de color (*r,g,b*) en la imagen *i* de la secuencia. *B* es el número de niveles en cada componente de color y *N* es el número de píxeles de cada imagen. Finalmente, CHD_i la diferencia de histogramas de color para la imagen *i*.

Partiendo de esta ecuación, se tiene en cuenta un umbral para la diferencia acumulada (CHD_i), y un intervalo de análisis para la detección de los picos (ventana temporal alrededor de la imagen analizada). También se pueden utilizar umbrales locales en vez de un único umbral.

Sin embargo, el umbral óptimo depende del tipo de contenido del vídeo, por lo que es difícil determinar un umbral medio válido para una secuencia completa, siendo este un problema común a todos los métodos encontrados en la literatura.

En general se trata de un método sencillo pero costoso de calcular, ya que al aplicarse en el ámbito del vídeo decodificado requiere un proceso de decodificación de la secuencia antes de poder aplicarlo. Por otra parte, al aplicarse sobre las imágenes completas es menos sensible al movimiento de cámara o de los objetos del plano.

Los métodos basados en la comparación de histogramas de color siempre se encuentran entre los mejores para una eficiencia de detección dada, siendo capaces de alcanzar una precisión del 96% para la máxima eficiencia.

Para este tipo de métodos existen múltiples extensiones, que calculan los histogramas sobre distintas regiones de las imágenes, o aplican distintas métricas de distancia. Por ejemplo, [26] calcula la diferencia entre los histogramas de regiones de la imagen y descarta las 8 diferencias mayores para reducir los efectos del ruido y del movimiento. Con esta aproximación se detecta además el 82% de las transiciones graduales.

II.3.3. Desviación Estándar de la Intensidad de los Píxeles

Como ejemplo de los métodos basados en medidas estadísticas sobre las imágenes cabe destacar un algoritmo diseñado específicamente para la detección de transiciones graduales, y más concretamente para la detección de desvanecimientos.

Los desvanecimientos se producen mediante el escalado monótono y normalmente lineal de la intensidad de los píxeles de las imágenes de la transición. Esta transición de la intensidad se puede ver claramente en la variación temporal de la desviación estándar de las intensidades de los píxeles.

La detección de desvanecimientos (*fade-in* y *fade-out*) se realiza calculando la línea de regresión entre la intensidad de imágenes consecutivas, y comparando la correlación y la diferencia de pendiente entre dichas líneas de regresión consecutivas. Si la correlación y la pendiente aumentan o disminuyen de forma constante hasta alcanzar una duración mínima de la transición se pueden detectar los *fade-ins* y *fade-outs* respectivamente.

Lo que se hace realmente es detectar la pendiente de la recta usada para generar la transición en la edición del vídeo original.

La detección de desvanecimientos es bastante precisa con un conjunto estándar de parámetros de configuración, según se describe en [28] .

II.3.4. Edge Change Ratio (ECR)

El principio en el que se basa este método [27][28] es diferente a los métodos anteriores. En este caso no se analizan diferencias de color entre imágenes, sino que se fija la atención sobre las diferencias entre los bordes detectados en imágenes adyacentes.

Así, los bordes de los objetos en la última imagen de un plano normalmente no se encuentran en la primera imagen del siguiente plano después del corte. Del mismo modo, los bordes de la primera imagen del nuevo plano no se encuentran en la última imagen del plano anterior al corte.

Cada imagen de la secuencia se transforma a escala de grises y se le aplica un filtrado de Sobel para detectar los bordes. El método busca bordes similares en imágenes adyacentes para detectar los extremos de los planos, con lo que puede evitar los problemas que sufren los métodos basados en comparaciones de color cuando se enfrentan a transiciones graduales. En casos como los desvanecimientos, aunque cambie la luminosidad, siempre habrá un conjunto de bordes que desaparezcan del plano anterior y un conjunto de bordes que aparezcan en el nuevo plano.

El ECR se define partiendo del número total de bordes en imágenes consecutivas, así como el número de bordes que salen de la primera imagen (bordes no encontrados en la segunda imagen) y el número de bordes que entran en la segunda imagen (bordes no encontrados en la primera imagen).

La ecuación que define este comportamiento es la siguiente:

$$ECR_i = \max(X_i^{in}/_{\sigma_i}, X_{i-1}^{out}/_{\sigma_{i-1}})$$

Ec. 4: Edge Change Ratio

La ecuación anterior define el *Edge Change Ratio* para la imagen *i* (ECR_i). X_i^{in} son los bordes que aparecen en la imagen *i*, mientras que X_{i-1}^{out} son los bordes que salen de la imagen *i-1*. σ_i es el número de bordes encontrados en la imagen *i*.

El valor de ECR varía entre 0 y 1, de forma que los máximos aislados se interpretan como cortes abruptos en la secuencia. Por otra parte, en un *fade-in* el número de bordes entrantes crece de forma continua, mientras que en un *fade-out* el número de bordes salientes aumenta de la misma forma. Finalmente, durante una disolución se produce un efecto conjunto, en el cual inicialmente

los bordes salientes del primer plano predominan, posteriormente disminuyen de forma progresiva a la vez que empiezan a aumentar los bordes entrantes del segundo plano hasta que finalmente éstos son los predominantes.

Esta aproximación permite también evitar falsos positivos debidos a cambios bruscos de iluminación, como pueden ser los flashes, en los cuales se sigue detectando el mismo conjunto de bordes.

Sin embargo, según la comparación ofrecida por [28], los resultados obtenidos por este método son sorprendentemente negativos, dando lugar a tasas de detección de cortes abruptos significativamente peores que métodos más sencillos como los basados en histogramas. Además, la detección transiciones (*fades, dissolves*) también es peor que en casos como la desviación estándar de las intensidades de los píxeles.

Por lo tanto, se puede ver que una mayor complejidad computacional no siempre lleva consigo un mejor comportamiento en términos de tasa de detección, tasa de falsas alarmas, eficiencia o eficacia.

II.3.5. GIST

Se ha demostrado que el GIST [31] es una medida capaz de caracterizar correctamente la estructura de las imágenes, siendo además resistente a cambios en la luminancia y a transiciones ligeras.

En inglés, la palabra "*gist*" se refiere a la parte básica o esencial de una cosa, por lo que aplicado al análisis de imágenes esta representación trata de obtener la esencia de una imagen.

El descriptor GIST fue propuesto inicialmente por Oliva y Torralba en [31] y consiste en una representación de una imagen en un conjunto limitado de dimensiones sin requerir ningún tipo de segmentación. Se definen una serie de dimensiones perceptuales (*naturalness, openness, roughness, expansion* y *riggedness*) que representan la estructura espacial dominante de la imagen.

La representación GIST trata la imagen como un objeto que se puede caracterizar mediante unas estructuras globales y locales consistentes. En vez de localizar los objetos dentro de la imagen utiliza la imagen completa como objeto. Se ha demostrado que el GIST funciona bien en el reconocimiento de categorías de escenas y en el proceso de facilitar las tareas de reconocimiento de objetos.

En el caso que nos ocupa de la detección de cambios de plano, el movimiento, aparición o desaparición de objetos en imágenes consecutivas puede hacer que métricas como los histogramas de color u otras no ofrezcan resultados consistentes [32]. Sin embargo, el GIST captura la textura general del fondo de las imágenes, ignorando los pequeños cambios de los objetos en primer plano. Por lo tanto, este método puede servir para detectar cambios en el contenido esencial de imágenes consecutivas, ayudando a la detección de cambios de plano.

Estos métodos suelen reducir el tamaño de las imágenes a procesar, ya que al requerir un conjunto limitado de dimensiones no necesitan toda la información espacial. Las imágenes se dividen en 16 bloques mediante una rejilla de 4x4 bloques y sobre cada bloque se aplica un conjunto de 24 filtros (combinaciones de tres filtros de escalado y 8 de orientación). Este proceso resulta en un conjunto de 384 dimensiones, que se reducen mediante un proceso de PCA (*Principal Component Analysis*). Las 40 componentes principales se utilizan

como representación GIST de la imagen, a la que se añaden otros conjuntos de características extraídas mediante otros procesos. Por ejemplo, el color se puede representar mediante histogramas de color tanto en RGB como en HSV.

Así, se calcula la distancia entre los vectores definidos por la representación GIST de imágenes consecutivas para medir la diferencia entre dichas imágenes y se observa que los cambios abruptos de plano se detectan como cambios bruscos en las características *gist* y de color.

Sin embargo, el cambio absoluto producido en estas características depende fuertemente de las secuencias concretas analizadas, por lo que se hace complicado obtener umbrales fijos para generalizar el método de detección. Por lo tanto, una comparación sencilla entre imágenes consecutivas no se puede utilizar para detectar los cambios de plano [27].

II.3.6. Detección de cambios de plano basada en macrobloques

Este tipo de métodos de detección de cambios de plano, al contrario que los anteriores, se basa en procesar la versión codificada del vídeo. Estos métodos se aplican ya desde los primeros codificadores MPEG-1 que utilizan un esquema de predicción compensada en movimiento. El principio de estos métodos radica en la forma de predicción utilizada, que divide la imagen en bloques (denominados macrobloques), y realiza la codificación de cada uno de ellos utilizando un método de predicción u otro en función de sus características. En concreto, se basa en la similitud del macrobloque entre la imagen actual e imágenes anteriores y posteriores.

Como se ha comentado en el primer capítulo, existen tres métodos de predicción que determinan el tipo de codificación de los macrobloques:

- **Macrobloques I**: utilizan para la predicción únicamente macrobloques de la misma imagen. Se usan cuando no hay similitud entre imágenes consecutivas.
- **Macrobloques P**: utilizan para la predicción imágenes anteriores. Se usan cuando hay similitud con imágenes anteriores.
- **Macrobloques B**: utilizan para la predicción imágenes tanto anteriores como posteriores, lo que lleva al término de bidireccionalidad.

Como se puede ver, cuantos más macrobloques I contenga una imagen, menor similitud con imágenes de su entorno, y por lo tanto mayor probabilidad de encontrarnos ante un cambio de plano.

La principal ventaja de estos métodos radica en que se aplican sobre el vídeo codificado, lo que simplifica y acelera el proceso de detección [27].

Por este y otros motivos, y dados los objetivos del proyecto que se describirán más adelante, este enfoque de la detección de cambios de plano será la utilizada en el método de detección descrito en esta tesis.

II.4. Objetivo

Según lo visto anteriormente, la detección de un cambio de plano en una secuencia se puede realizar a partir del cálculo de una medida de continuidad y su variación [1]. Pero este cálculo depende a su vez de varios factores: encontrar una medida que sea insensible a las variaciones en los parámetros de la escena y de la cámara (iluminación, etc.) y que sea suficientemente discriminante; establecer los valores de dicha medida que corresponden con cambios de plano;

y finalmente, tener en cuenta la gran variabilidad de situaciones que se dan en los cambios de plano, como pueden ser transiciones abruptas, graduales, con efectos, etc.

Por lo tanto, las características de un algoritmo de detección de cambios de plano vienen determinadas por el objetivo final de dicha detección. Existen multitud de aplicaciones para estos algoritmos, como por ejemplo la indexación automática de vídeo, la edición de vídeo, la transcodificación o la restauración de películas antiguas, entre otras.

En el caso que nos ocupa, el principal objetivo del algoritmo de detección de cambios de plano es la optimización de la codificación del vídeo digital en tiempo real, con lo que el método debe aplicarse durante el proceso mismo de codificación. Con esta premisa, el detector debe añadir la mínima carga posible al codificador y, para ello, aprovechará la información generada por el propio codificador durante su funcionamiento para mejorar la eficiencia de codificación.

Así, el objetivo del presente capítulo es explotar las propiedades estadísticas de las imágenes de los cambios de plano, junto con las características intrínsecas del proceso de codificación, para detectarlos con precisión, ayudando así a mejorar la eficiencia global de codificación de la secuencia completa. El fundamento del método de detección que se propone se basa en la medida de la correlación existente entre imágenes consecutivas, para explotar la similitud existente entre las imágenes que forman parte del mismo plano. Con esta medida será posible la detección de los cambios de plano como aquellas imágenes que suponen una ruptura en la continuidad de la correlación entre las imágenes del plano.

Al mismo tiempo, al explotar la estructura esencial de los codificadores de vídeo de última generación, se demuestra que utilizando esta medida de correlación entre imágenes es posible mejorar el proceso de codificación de la secuencia completa.

Por último, la mayoría de los algoritmos presentes en la literatura describen unos resultados basados en el análisis de las prestaciones de detección obtenidas sobre un conjunto de secuencias de test muy reducido. Este hecho limita la validez de dichos métodos al disponer de un conjunto de escenarios que no resulta suficientemente amplio para generalizar los resultados. Por tanto, un objetivo adicional de esta tesis consiste en determinar un conjunto de secuencias suficientemente amplio como para contener información suficiente para entrenar y validar los algoritmos desarrollados.

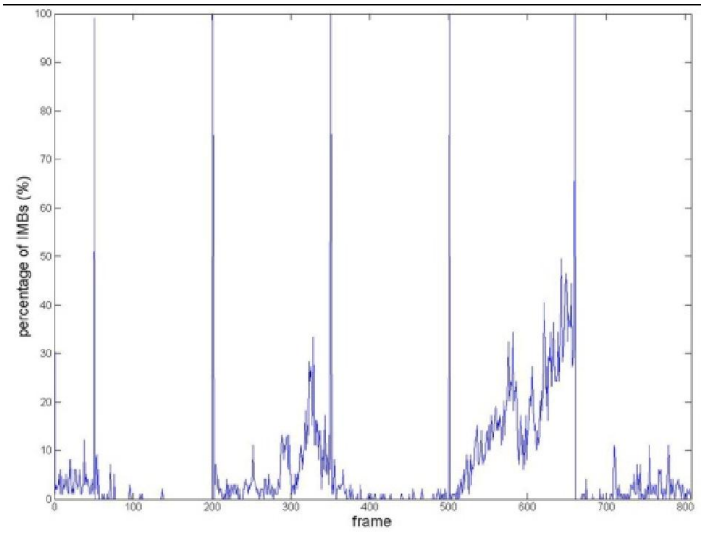
III. Diseño del detector automático de cambios de plano

En esta sección se presenta el algoritmo de detección automática de cambios de plano, describiendo los fundamentos en los que se basa el algoritmo, y ofreciendo una descripción precisa del método y de los parámetros que definen su funcionamiento.

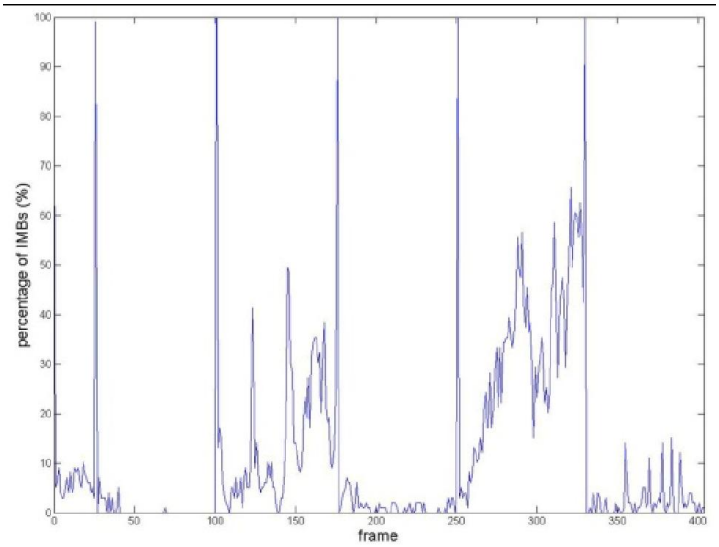
III.1. Fundamentos de la detección

Como se ha visto anteriormente, el número de macrobloques *intra* utilizados para codificar una imagen P puede ser un buen indicador de la correlación existente entre dicha imagen y las anteriores (utilizadas como referencia).

Para ilustrar este comportamiento, en la Fig. 12 se puede observar el número de macrobloques *intra* utilizados para codificar cada uno de los *frames* de una secuencia compuesta por distintos planos [6]. En concreto, nos encontramos ante la concatenación de 6 fragmentos de algunas de las más conocidas secuencias de test estándar: *Flower Garden*, *Akiyo*, *Stephan*, *Hall*, *Snow* y *Container Ship*. En este ejemplo, únicamente la primera imagen ha sido codificada como imagen *intra* (I), mientras que el resto de frames son de tipo P.



(a)



(b)

Fig. 12: Porcentaje de macrobloques *intra* para una secuencia codificada a 12.5 imágenes por segundo (a) y a 6.25 fps (b)

Se puede demostrar, a la vista de la Fig. 12, que el número de macrobloques *intra* depende del contenido de la imagen codificada, y en particular de las características de movimiento de la secuencia. En este sentido, planos con mucho movimiento en escena (como *Stephan* o *Snow*) requieren de gran cantidad de macrobloques *intra* para su codificación, debido a que el movimiento de cámara o de objetos en las imágenes dificulta la obtención de predicciones válidas. Por el contrario, escenas con poco movimiento (como *Akiyo* o *Hall*) utilizan menos macrobloques *intra* porque prácticamente toda la imagen se puede predecir a partir de imágenes anteriores.

Por otra parte, el número de macrobloques *intra* también depende de la tasa de imágenes por segundo de la secuencia codificada, ya que a mayor separación temporal entre imágenes, menor correlación existe entre ellas, aumentando el número de macrobloques *intra* necesarios para su codificación. Por lo tanto, utilizando el número de macrobloques *intra* como medida de continuidad de la correlación entre imágenes es posible detectar con mucha precisión los cambios de plano abruptos de una secuencia.

La potencialmente infinita variabilidad de situaciones que se pueden dar en las escenas codificadas hace que un detector basado en un único umbral fijo no sea suficientemente preciso a la hora de detectar los cambios de plano. Esto es así porque las técnicas basadas en un único umbral no tienen en cuenta el contenido de las imágenes de la secuencia, ni la continuidad de los parámetros medidos, a la hora de realizar la detección. Para solucionar este problema se ha planteado la combinación de dos umbrales: un primer umbral adaptativo, basado en una medida de la continuidad del número de macrobloques *intra* de las imágenes que forman parte del mismo plano; y un segundo umbral fijo, utilizado como mecanismo de seguridad que evita la codificación de imágenes P con un número demasiado elevado de macrobloques *intra*. De esta forma, el umbral dinámico se adapta a las condiciones puntuales del vídeo, estableciendo un umbral más restrictivo en situaciones con mucho movimiento o baja tasa de imágenes por segundo. Por el contrario, el umbral disminuye en situaciones más estables con un número medio de IMBs más reducido.

Uno de los objetivos del método desarrollado consiste en realizar la detección durante el proceso de codificación de cada imagen de la secuencia, permitiendo su utilización por el propio codificador para optimizar el proceso de codificación tanto de la imagen en la que se ha detectado el cambio de plano

como de las siguientes. En concreto, las imágenes detectadas como cambios de plano, debido a su escasa correlación con las imágenes del plano anterior, son las mejores candidatas para la inserción de imágenes *intra* en la secuencia codificada. De esta forma, se abandona el esquema tradicional de inserción periódica de *keyframes* para realizar una inserción basada en el contenido, lo que permite optimizar las referencias utilizadas para codificar el resto de imágenes que pertenecen al plano que comienza en la posición detectada.

III.2. Algoritmo de detección

El algoritmo de detección de cambios de plano abruptos consta de tres componentes fundamentales: los umbrales, la memoria del número de macrobloques *intra* y el intervalo de guarda o *span*, los cuales pasamos a describir a continuación.

En primer lugar, es necesario definir dos intervalos de funcionamiento del algoritmo: el intervalo de guarda y el intervalo de funcionamiento normal, cuya diferencia radica en la utilización de distintos umbrales y en la consideración de diferentes parámetros de medida.

Un análisis estadístico de la distancia existente entre cambios de plano consecutivos muestra que la probabilidad de encontrar dos cortes muy próximos en el tiempo es muy baja [7], agrupando la mayoría de cambios en un intervalo acotado entre un mínimo y un máximo, que dependen en gran medida del tipo de secuencia en consideración. En la Tabla 1 se muestra un ejemplo de la distancia media medida entre cambios de plano en distintos tipos de secuencias.

Tipo de secuencia	Distancia media entre cambios de plano (frames)
Noticias	90.87
Drama	54.67
Acción	44.08
Anuncios	36.76

Tabla 1: Distancia media entre cambios de plano (a 25 imágenes por segundo)

La distancia media entre cambios de plano es mucho menor en secuencias de acción o en anuncios que en noticias o en secuencias dramáticas. Además, no se encuentran cambios de plano a una distancia menor de 40 imágenes para las secuencias habituales, o 30 para el caso especial de anuncios o tráileres, que por sus características presentan cambios de plano mucho más frecuentes.

Por lo tanto, el algoritmo de detección de cambios de plano debe descartar los cortes detectados demasiado cerca de la última detección, para lo que se establece un intervalo de guarda. Durante este intervalo las condiciones de detección son más restrictivas, permitiendo la detección únicamente si prácticamente todos los macrobloques de una imagen se codifican en modo *intra*.

Este intervalo de guarda se puede definir en términos de número de imágenes codificadas desde el último cambio de plano detectado:

$$N_{span} = F \times T_{span}$$

Ec. 5: Intervalo de guarda

En esta ecuación, N_{span} es el número de imágenes incluidas en el intervalo de tiempo T_{span} (medido en segundos) cuando se codifican F imágenes por segundo (fps). En este sentido, el algoritmo se ejecuta para cada nueva imagen codificada, considerando una tasa de imágenes por segundo fija, de forma que para cada imagen, el contador k indica el número de imágenes codificadas desde el último cambio de plano detectado.

Durante el intervalo de guarda, el algoritmo de detección de cambios de plano consiste en un único umbral de seguridad, denominado T_S. Una vez transcurrido dicho intervalo de guarda (cuando $k > N_{span}$), el algoritmo de detección entra en su etapa de funcionamiento normal, donde entra en juego un segundo grupo de umbrales de detección: el umbral adaptativo T_A y el umbral fijo T_L, que en conjunto forman el umbral dinámico T_D. En concreto, el umbral T_A se calcula según la siguiente ecuación:

$$T_A = m_k + T_a$$

Ec. 6: Umbral adaptativo

Donde $T_a = a \times N_{MB}$ es un porcentaje del número total de macrobloques de una imagen completa (N_{MB}) y $a \in [0,1]$ identifica el porcentaje concreto. Por su parte, m_k es la media ponderada del número de macrobloques *intra* que se han utilizado para codificar las imágenes que pertenecen al mismo plano, actualizando la media mediante un proceso con memoria:

$$m_k = m_{k-1} \times \alpha + IMB_k \times (1-\alpha)$$

Ec. 7: Media ponderada del número de macrobloques intra

En la ecuación anterior IMB_k es el número de macrobloques *intra* de la imagen k y α es el parámetro de memoria. El valor de m_k es una ponderación del valor de m_{k-1} anterior y del número de macrobloques *intra* de la imagen recién codificada (k), con lo que, en función del valor de α, se puede variar el comportamiento de la media ponderada, permitiendo obtener una media más estable o una media más sensible.

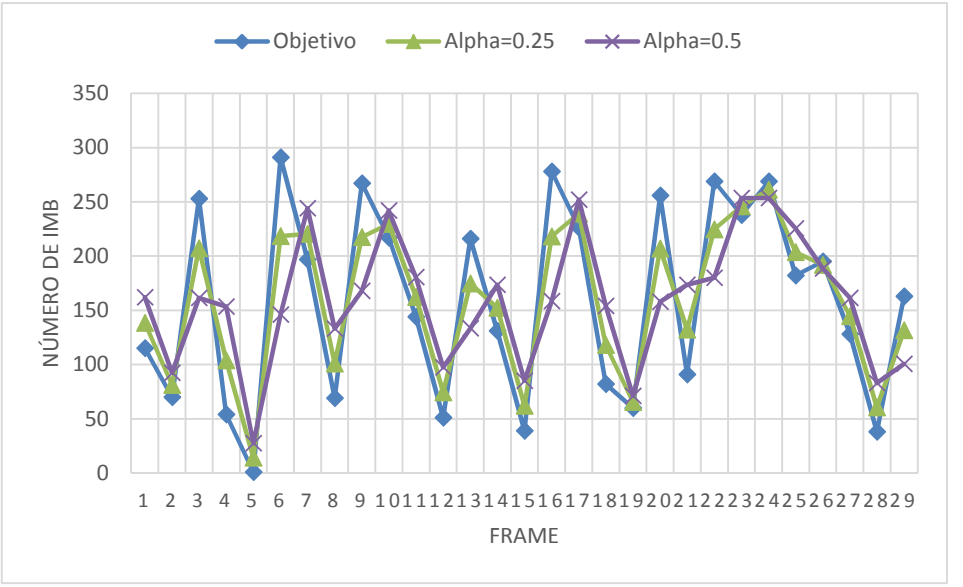


Fig. 13: Comportamiento de la memoria

En la Fig. 13 se puede observar el comportamiento del parámetro de memoria para dos valores distintos de α. Con $\alpha = 0.5$ se obtiene un compromiso entre el valor anterior y el actual, mientras que con $\alpha = 0.25$ el umbral sigue más rápidamente las oscilaciones del número de macrobloques *intra* (IMB) objetivo.

Por último, el umbral dinámico se calcula aplicando un límite superior al umbral adaptativo:

$$T_D = Min(T_A, T_L)$$

Ec. 8: Umbral dinámico

En la Fig. 14 se puede observar un diagrama de bloques que muestra el funcionamiento del algoritmo aquí descrito: cuando una nueva imagen está preparada para ser codificada, se actualizan los parámetros del detector: se calcula el valor de la media de macrobloques *intra* de las imágenes que pertenecen al plano actual (m_k), se actualiza el umbral dinámico T_D para la imagen k y se inicializa el contador de macrobloques *intra* de la imagen k ($IMB_k = 0$). A continuación se realiza la codificación de la imagen macrobloque a macrobloque, de forma que, tras la codificación de cada uno de ellos, se actualiza el valor del contador de macrobloques *intra*. Si el valor de IMB_k no supera el umbral activo en función del intervalo de actividad, la codificación de la imagen actual termina, y el codificador está preparado para recibir la siguiente imagen de la secuencia. Por el contrario, si durante el proceso de codificación IMB_k supera el umbral correspondiente (T_D o T_S, en función del valor de k), se produce la detección de un cambio de plano, señalizándolo de la manera oportuna según la aplicación en la que se utilice el detector.

En concreto, en la implementación realizada, la detección de un cambio de plano implica abortar la codificación de la imagen actual como imagen P para recodificarla en modo I, de forma que sirva como referencia para las imágenes del nuevo plano.

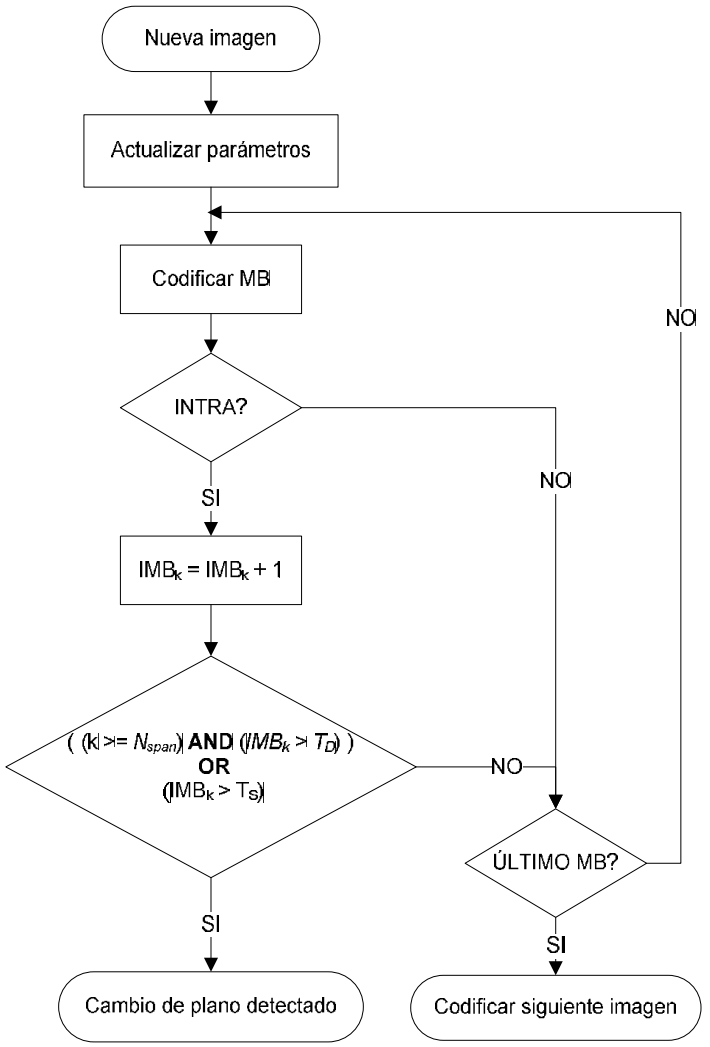


Fig. 14 Diagrama de flujo del algoritmo de detección de cambios de plano

En la Fig. 15 se puede observar un ejemplo de funcionamiento del algoritmo en un entorno real, obtenida como resultado de la codificación de una secuencia de vídeo a 750 kbps y 25 fps. En línea continua se representa el porcentaje de

macrobloques *intra* utilizados para codificar cada una de las imágenes de la secuencia. Por otra parte, en línea discontinua, se muestra el valor porcentual del umbral final aplicado en cada imagen. En la figura aparecen cinco cortes reales y las correspondientes detecciones llevadas a cabo por el algoritmo, representadas mediante una cruz que se superpone al punto que representa el corte real. Como se puede ver, la detección es precisa en todos los casos que se muestran en la figura. En el caso del cambio de plano situado en el frame 4563, la detección se produce cuando se han codificado un 50% de los macrobloques de la imagen, ya que se trata de una escena con poco movimiento, y el cambio de plano produce un aumento brusco que se detecta fácilmente gracias al umbral adaptativo. Por el contrario, el cambio de plano situado en el frame 4667 corresponde a un plano con mucho movimiento, por lo que el número medio de macrobloques *intra* de las imágenes del plano es muy elevado. En este caso, es el umbral fijo T_L el que produce la detección, cuando se supera el 96% de macrobloques *intra*, limitando por debajo el valor del umbral adaptativo T_A. Finalmente, en la figura también se puede observar el comportamiento del umbral de seguridad, T_S, definiendo el intervalo de guarda después de la primera imagen de cada nuevo plano.

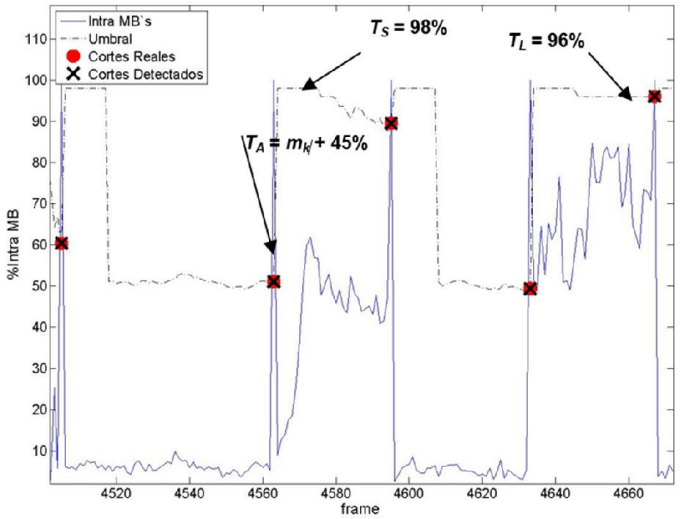


Fig. 15: Ejemplo de funcionamiento real del algoritmo de detección de cambios de plano

IV. Entorno de pruebas

Una vez diseñado el detector automático de cambios de plano, es necesario definir el entorno de funcionamiento utilizado para la optimización de los valores de sus parámetros (entrenamiento) y para la medida de los resultados de detección una vez optimizado (test). Dicho entorno de pruebas comprende las secuencias empleadas para entrenar y probar el algoritmo de detección, el *ground truth* de las secuencias, el codificador utilizado, las configuraciones de calidad consideradas y las definiciones necesarias para la medida de las prestaciones del algoritmo.

El hardware utilizado para el desarrollo de este proyecto ha consistido en un PC con procesador Pentium IV a 3.2 GHz con *HyperThreading* y 1 GB de memoria RAM.

IV.1. Secuencias

La obtención de un conjunto de secuencias de test con unas características determinadas incluye la fuente de la que se extraen, el formato de vídeo en el que se almacenan, la determinación de sus características y su posterior organización en distintas categorías en función de ciertos criterios.

IV.1.1. Fuentes, resoluciones y formato

a) *Fuentes:*

Las secuencias de test estándar tienen unas características bien definidas en cuanto a movimiento en escena, movimiento de cámara, detalle espacial y temporal, crominancia, etc. Todas estas características son positivas a la hora de medir las prestaciones de determinados sistemas, pero, en el caso de la detección de cambios de plano no se adaptan a las necesidades planteadas: se trata de secuencias demasiado cortas, de pocos segundos de duración, que no incluyen cambios de plano y no disponen de la suficiente variabilidad a la hora de representar fielmente las secuencias reales que un sistema de codificación se encuentra en su funcionamiento habitual. Sin embargo, la concatenación artificial de estas secuencias para formar secuencias más largas con cambios de plano controlados constituye una primera toma de contacto para comprobar el funcionamiento de los algoritmos implementados. Este es el caso de la secuencia utilizada para obtener los datos plasmados en la Fig. 12.

Este tipo de concatenación es útil para las primeras etapas de desarrollo del algoritmo, pero para comprobar su funcionamiento una vez implementado es necesaria una mayor variedad de secuencias, extraídas preferiblemente de grabaciones reales con la mayor variabilidad posible.

Para tener en cuenta estas consideraciones se han utilizado secuencias procedentes de dos fuentes principales:

- **Películas en DVD**: el grueso del conjunto de secuencias utilizadas proviene de películas extraídas del DVD original. Las películas escogidas intentan cubrir un abanico lo suficientemente amplio tanto de géneros como de estilos. Esta variedad se refleja en la diversidad de características de las escenas seleccionadas, obteniendo una muestra significativa de las secuencias existentes en el mundo real.
- **Emisiones de Televisión Digital Terrestre**: para completar el conjunto de secuencias, se han grabado emisiones de TDT, procedentes de anuncios comerciales, tráileres de películas, etc. Estas escenas presentan mayor frecuencia de cambios de plano, gran variedad de estilos, de imágenes y de efectos, junto con diferentes tipos de transiciones entre planos: cambios abruptos, transiciones graduales, fundidos, cortinillas, etc.

Cada uno de los mencionados fragmentos está constituido por 5000 *frames*, codificados a 25 imágenes por segundo, dando como resultado secuencias con una duración de 3 minutos y 20 segundos. Esta duración es suficiente para que cada fragmento contenga un número de cambios de plano que permita una medida fiable del comportamiento del algoritmo de detección correspondiente. En total se han seleccionado 14 fragmentos, con un total de 70000 *frames* y más

de 1500 cambios de plano. Se puede encontrar una descripción detallada de las características de estas secuencias en [24] y en el Anexo de esta tesis.

b) *Resoluciones:*

La resolución original de estas secuencias es de 720x576 píxeles, que constituye la resolución nativa del DVD y de las emisiones televisivas digitales de la Televisión Digital Terrestre (TDT) en Europa. Esta resolución se ha adaptado posteriormente, escogiendo tres tamaños distintos de imágenes, para acomodar aplicaciones de baja, media y alta resolución, como se aprecia en la Tabla 2.

Formato	QCIF	CIF	SDTV
Resolución	Baja	Media	Alta
Nº Píxeles	176x144	352x288	720x416

Tabla 2: Resoluciones de imagen utilizadas

El formato CIF (*Common Intermediate Format*) es un formato definido para estandarizar la resolución horizontal y vertical de secuencias de vídeo, utilizado comúnmente en sistemas de videoconferencia. Fue propuesto inicialmente en el estándar H.261.

CIF fue diseñado para ser fácilmente convertible a los formatos PAL o NTSC, definiendo una resolución de 352x288 (la misma que el formato de fuente de entrada de PAL), una tasa de imágenes por segundo de 30000/1001 (aproximadamente 29.97 fps, como en NTSC), y un espacio de color YCbCr 4:2:0.

Por su parte, el formato QCIF (*Quarter CIF*) tiene una cuarta parte de resolución que CIF (la mitad de píxeles horizontales y la mitad de píxeles verticales).

Los tamaños de imágenes CIF fueron escogidos de forma que contengan un número entero de macrobloques. Esto facilita la codificación y decodificación de las imágenes debido a la forma en la que la transformada discreta del coseno trabaja en la mayoría de los codificadores de vídeo híbridos actuales.

c) Formato:

Las secuencias extraídas deben ser formateadas correctamente para coincidir con el formato de entrada del codificador de vídeo en el cual se ha implementado el detector. En concreto, este formato de entrada es YUV420, con una componente de luminancia y dos componentes diezmadas de crominancia, todas ellas sin codificar, para cada imagen de la secuencia a procesar.

El formato YUV es un espacio de color usado típicamente en la cadena de procesado del vídeo codificado. Codifica las imágenes o el vídeo teniendo en cuenta la percepción del sistema visual humano. Esto permite reducir el ancho de banda de las componentes de crominancia con respecto a la luminancia.

Históricamente, el término YUV se utilizaba para identificar la codificación analógica de la información del color en sistemas de televisión, mientras que el término YCbCr se utilizaba en su vertiente digital. Actualmente, el término YUV se utiliza para describir los formatos de fichero que se codifican utilizando YCbCr.

YUV define un espacio de color con una componente de luminancia (Y) y dos de crominancia (UV). Las componentes de crominancia son diferenciales con respecto a la Y, de forma que la conversión de RGB a YUV se puede aproximar por:

$$Y = 0.299R + 0.587G + 0.114B$$
$$U = 0.492(B - Y)$$
$$V = 0.877(R - Y)$$

Ec. 9: Conversión de RGB a YUV

Una de las principales ventajas de los sistemas basados en la luminancia y la crominancia es que son compatibles con los sistemas en blanco y negro originales. En estos sistemas se ignoran las componentes de crominancia y se representa únicamente la luminancia.

Otra ventaja radica, como se ha comentado anteriormente, en que parte de la información puede ser descartada para reducir el ancho de banda sin una pérdida significativa de calidad percibida. Es lo que se conoce como submuestreo de la crominancia, que consiste en diezmar la resolución horizontal y/o vertical de las señales de crominancia con respecto a la luminancia.

Por lo tanto, si no se diezma la crominancia se obtienen esquemas de tipo 4:4:4; si se diezma la resolución horizontal de la crominancia obtenemos los esquemas 4:2:2, mientras que si se aplica también el diezmado a la dimensión vertical de la crominancia se obtiene el esquema 4:2:0.

Así, el formato YUV420 representa las imágenes mediante una componente de luminancia y dos componentes de crominancia diezmadas en horizontal y vertical.

Es curioso comentar que la nomenclatura 4:2:0 no parece seguir un formato lógico sino histórico, ya que no representa directamente la proporción entre el número de muestras en cada dimensión ni en cada componente de color [8].

Finalmente, cada una de estas componentes se representa mediante valores codificados en 8 bits, por lo que este tipo de representación se suele conocer como de 12 bits por píxel. Este valor viene del hecho de que, para codificar 4 píxeles, son necesarias 6 muestras (4 de luminancia y 2 de crominancia). Esto da un total de 48 bits, que repartidos entre los cuatro píxeles proporcionan los mencionados 12 bits por píxel.

IV.1.2. Categorías de movimiento

El siguiente paso es la clasificación de las secuencias anteriormente obtenidas en función del movimiento tanto de cámara como en escena que incluyen las escenas analizadas. De esta forma, se han definido cuatro categorías diferentes:

- **Poco Movimiento y Poco Movimiento de Cámara (LM&LC)**: secuencias con poco movimiento en escena, como en diálogos con fondo homogéneo. La cámara está fija o realiza movimientos suaves. Los cambios de plano son poco frecuentes y abruptos.
- **Movimiento Moderado y Movimiento Moderado de Cámara (MM&MC)**: escenas con más movimiento, tanto de personajes y objetos como de cámara, que realiza *pannings* y zooms de mayor velocidad que en el caso anterior, mientras el escenario y los personajes realizan movimientos moderados. Ejemplos de este tipo de escenas pueden ser secuencias de carretera, o escenas de lucha

filmadas a cierta distancia. Los cambios de plano son más frecuentes que en el primer caso.

- **Mucho Movimiento y Mucho Movimiento de Cámara (HM&HC)**: escenas de acción con mucho movimiento, tanto en escena como de cámara, filmadas con cámara al hombro, con desplazamientos bruscos, oclusiones, etc. Ejemplos de este grupo son escenas de combate filmadas a corta distancia, con múltiples personajes en el plano y movimientos caóticos. Los cambios de plano son mucho más frecuentes y bruscos.
- **Anuncios Comerciales (COM)**: secuencias que contienen todas las configuraciones anteriores simultáneamente, con variación en el tipo de movimiento de cámara y del movimiento en escena, con cortes bruscos y transiciones graduales, con cortes frecuentes y largas secuencias continuas, etc. Estas secuencias ponen a prueba al algoritmo, proporcionando el escenario más desfavorable posible por la gran variabilidad de las características que presentan. Ejemplos de este tipo de secuencias pueden ser anuncios o tráileres de películas, clips musicales, etc.

IV.1.3. Conjunto de Entrenamiento y Conjunto de Test

A la hora de comprobar el funcionamiento del algoritmo de detección de cortes hay que establecer dos etapas claramente diferenciadas: el entrenamiento y el test. La primera etapa tiene como objetivo encontrar los valores óptimos de los distintos parámetros del algoritmo para conseguir el funcionamiento deseado. En la segunda etapa se comprueba el comportamiento obtenido cuando

el algoritmo se aplica a un conjunto de secuencias distintas de las utilizadas para su optimización.

Por este motivo, el conjunto original de secuencias analizadas anteriormente se ha dividido en dos grupos, el conjunto de entrenamiento (*Training Set*) y el conjunto de test (*Test Set*), cada uno formado por secuencias pertenecientes a las cuatro categorías de movimiento.

IV.2. Ground Truth

Una vez definidas las secuencias a utilizar, el siguiente paso consiste en analizar dichas secuencias, para establecer el *ground truth* del que partir a la hora de medir las prestaciones de detección. La obtención del *ground truth* consiste en determinar la posición exacta de los cambios de plano existentes en las secuencias, de forma que dicha información pueda ser comparada con los resultados de detección, permitiendo el análisis estadístico de las prestaciones del algoritmo.

De este modo, cada secuencia se ha analizado *frame* a *frame*, estableciendo las imágenes que constituyen los cambios de plano, teniendo en cuenta tres tasas de imágenes por segundo distintas: la tasa de imágenes por segundo original, la mitad de dicha tasa y un cuarto de la tasa original. Con estas consideraciones, se comprueba que el *ground truth* es ligeramente diferente para cada valor de fps, lo cual es debido a que, al disminuir el número de imágenes codificadas, la distancia temporal existente entre imágenes consecutivas es mayor, disminuyendo la correlación existente entre una imagen y la anterior. Este hecho puede hacer que, a bajas tasas de imágenes por

segundo aparezcan cambios de plano donde no existían a la tasa original, como se puede comprobar en los siguientes ejemplos:

- **Transición gradual a cambio abrupto**: una transición gradual puede aparecer como un cambio de plano abrupto al disminuir la tasa de imágenes por segundo. Un ejemplo de este tipo de situación se muestra en la Fig. 16, donde aparece un fundido entre dos secuencias de test (*Akiyo* y *Stephan*) codificadas a 25 imágenes por segundo (a-e). Esta misma secuencia, codificada a 8.3 fps, incluye únicamente las imágenes (a) y (e), apareciendo como un cambio de plano abrupto.

(a) (b) (c) (d) (e)

Fig. 16: Transición gradual a 25 imágenes por segundo (a-b-c-d-e) que, al reducir a 8.3 el número de imágenes por segundo, aparece como un cambio de plano abrupto (a-e)

- **Movimiento extremo que genera cambio abrupto**: un movimiento brusco de cámara o de objetos en el plano puede sobrepasar la capacidad de los vectores de movimiento para estimar correctamente el movimiento entre imágenes consecutivas. En estos casos, la disminución del número de imágenes por segundo puede producir el mismo resultado que un cambio de plano abrupto, como se puede

apreciar en la Fig. 17, donde se observan cinco fotogramas de la secuencia *Foreman* codificados a 12.5 fps. En la escena se produce un movimiento brusco de cámara, que hace que la imagen (d) sea considerada como un cambio de plano por la discontinuidad introducida.

(a) (b) (c) (d) (e)

Fig. 17: Ejemplo de movimiento brusco que supera el tamaño máximo de los vectores de movimiento

IV.3. Codificador

Una de las principales ventajas del algoritmo de detección de cambios de plano desarrollado es su simplicidad, que lo hace apto para su implementación en cualquier codificador de vídeo basado en la decisión *inter/intra* macrobloque a macrobloque. La gran mayoría de codificadores actuales mantienen esta filosofía de funcionamiento, según se ha visto en el primer capítulo.

El codificador finalmente seleccionado para la implementación del algoritmo descrito en esta tesis es el H.264/AVC [8][9], que era el esquema de codificación más avanzado en el momento en el que se desarrolló este trabajo. En este apartado se presenta una descripción de las principales características de este códec, junto con las implicaciones que tiene esta elección en el funcionamiento del detector de cambios de plano.

a) Consideraciones sobre el codificador utilizado:

Para la implementación del algoritmo de detección de cambios de plano se ha escogido un codificador de código libre (*open source*) denominado x264 [11], ya que proporciona todas las herramientas de codificación disponibles en el estándar, así como una implementación eficiente, robusta y fiable.

En este caso, los elementos del codificador que más afectan a la hora de integrar el algoritmo de detección de cambios de plano en el codificador son el control de tasa del mismo y el algoritmo de selección del modo de codificación de cada macrobloque (*intra*, inter, tipo de partición, etc.).

En el primer caso, el control de tasa es clave en aplicaciones de tiempo real o bajo retardo a la hora de asignar a cada imagen la misma cantidad de bits para mantener el *bitrate* constante a lo largo de toda la secuencia codificada. Esto influye directamente en la calidad de las imágenes codificadas, y por tanto, en la calidad global de la secuencia completa [13][14].

Por otra parte, la implementación del x264 incluye un algoritmo de selección de modo que utiliza como métrica la suma de las diferencias absolutas transformadas (SATD). Este parámetro es mayor cuanto menor es la correlación entre el macrobloque original y su predicción, por lo que depende directamente de la correlación existente entre imágenes cercanas y de la cantidad de movimiento que contenga la escena. Así, el algoritmo *Rate Distortion Optimization* utiliza el modo de predicción que obtiene un menor SATD a la hora de codificar cada macrobloque.

b) Configuración del codificador utilizado:

La necesidad de una implementación válida para aplicaciones de tiempo real o bajo retardo tiene distintas implicaciones a la hora de configurar el codificador utilizado. En particular, es necesario definir el número de imágenes de referencia que se utilizan para la predicción y la estructura y tipo de GOP que se genera.

El enfoque del codificador a aplicaciones de bajo retardo implica una reducción forzosa de la complejidad computacional del proceso global de codificación: añadir una nueva imagen a la lista de imágenes de referencia implica un aumento de la complejidad, ya que todo el proceso de búsqueda de predicciones en todos los modos *intra* e *inter* se tiene que realizar en cada imagen de la lista de referencia. En este caso se ha limitado el tamaño de dicha lista a dos imágenes, de forma que se establece un compromiso entre la complejidad computacional y la calidad en las predicciones obtenidas.

Por otra parte, el escenario de bajo retardo limita el tipo de imágenes codificadas a utilizar, permitiendo únicamente *frames* de tipo I y P, y descartando el uso de imágenes B, ya que éstas implican un retardo fijo que en este tipo de aplicaciones no es tolerable. De este modo, el GOP seleccionado toma la forma IP…P.

Finalmente, el hecho de codificar las imágenes detectadas como cambios de plano en modo *intra* obliga a utilizar un tipo de GOP abierto, ya que no se conoce a priori la duración del plano, y por lo tanto no se puede predecir la posición de la siguiente imagen *intra* en la secuencia.

c) Estadísticas de codificación:

Al código original del codificador x264 se le han añadido diversas funcionalidades estadísticas, utilizadas para el estudio del comportamiento del algoritmo implementado. Estas funcionalidades consisten en la creación de archivos de estadísticas que contienen la información utilizada por el codificador a la hora de procesar cada imagen, así como los resultados de dicha codificación.

Entre la información recopilada se incluye el número de macrobloques de cada tipo que se ha utilizado para codificar cada imagen, las predicciones utilizadas, el número y tipo de imagen, las imágenes de referencia, el número de bits empleados para su codificación, el tiempo de codificación y la PSNR de cada uno de los *frames* codificados. Así es posible la comparación de los resultados obtenidos por el detector de cortes tanto en consideraciones de precisión en la detección como en calidad objetiva de la codificación obtenida.

IV.4. Configuraciones

El enfoque original de la primera etapa de desarrollo del algoritmo de detección de cambios de plano va dirigido a la codificación de vídeo de alta calidad, lo que limita el tipo de formato de vídeo (resolución estándar, SDTV), y las configuraciones de imágenes por segundo y *bitrate*. Se han utilizado cuatro configuraciones distintas, con dos tasas de imágenes por segundo diferentes:

Tasa de imágenes (fps)	Bitrate (kbps)
12.5	750
25	750
	1500
	2000

Tabla 3: Configuraciones de bitrate y frame rate

Se han escogido tres configuraciones de *bitrate* para secuencias a 25 imágenes por segundo, coincidiendo con aplicaciones de bajo (750 kbps), medio (1500 kbps) y alto (2000 kbps) *bitrate*. Para el caso de 12.5 fps se ha optado por una única configuración de alta calidad a 750 kbps.

IV.5. Medida de prestaciones

Como se ha mencionado anteriormente, las medidas más usadas para determinar las prestaciones en algoritmos de detección y recuperación son la precisión (*Precision*) y la eficacia (*Recall*).

En función de la aplicación concreta a la que se apliquen, el objetivo de los métodos de detección puede ser distinto. En algunos casos se prima la precisión sobre la eficacia, mientras que en otros es más importante minimizar el número de detecciones fallidas.

En el caso que nos ocupa, el objetivo es conseguir un compromiso en el cual los valores de precisión y eficacia sean lo más altos posibles simultáneamente, maximizando la tasa de detección y minimizando el número de falsos positivos.

Esto es especialmente importante, ya que es sencillo aumentar el número de entidades detectadas (aumentando la precisión) de forma que se minimice el

número de detecciones perdidas. Sin embargo, esto hace que se dispare el número de falsas detecciones, empeorando la eficacia.

Por otra parte, para la medida objetiva de la calidad de vídeo, el parámetro más utilizado es la PSNR, o relación señal a ruido de pico (*Peak Signal to Noise Ratio*), calculada según:

$$PSNR = 10\log\left(\frac{255^2}{MSE}\right)$$

Ec. 10: PSNR

En la Ec. 10 *MSE* es el error cuadrático medio entre los píxeles de la imagen codificada y la imagen sin codificar, y 255 es el valor máximo que pueden tomar los píxeles de la imagen. Esta magnitud se mide en dB, y toma el valor máximo cuando no existe diferencia entre las imágenes comparadas.

V. Entrenamiento del detector de cambios de plano

V.1. Justificación teórica

El algoritmo de detección de cambios de plano se puede ver como un proceso de votación, familiar en la teoría de detección descentralizada o distribuida [15][16]. En este caso, se debe decidir si una determinada imagen es o no un cambio de plano. En primer lugar, se debe tomar una decisión local sobre cada macrobloque del *frame* que se está procesando, decidiendo si es *intra* o no, de forma que el número total de macrobloques *intra* (IMB_K) se compara con un umbral. Así, se puede considerar cada macrobloque *intra* de la imagen como un voto a favor de que la imagen sea un cambio de plano, de

forma que cuando el número de votos es suficiente, la imagen se señaliza como un cambio de plano.

La optimización de los procesos de votación ha sido estudiada tanto en presencia como en ausencia de correlación entre las decisiones locales [16]. Dicha optimización se puede conseguir desde dos perspectivas diferentes: maximización de la probabilidad de detección para una determinada probabilidad de falsa alarma (detector de Neyman-Pearson), o minimización de una función de coste (detector de Bayes). En nuestro caso, la segunda perspectiva es más apropiada, considerando que la función de coste es la maximización conjunta de los valores de *Precision* y *Recall* del algoritmo. Sin embargo, es muy difícil definir modelos analíticos que relacionen el valor del umbral con el comportamiento del algoritmo en términos de eficacia y precisión, por lo que la optimización del detector consiste en una fase de aprendizaje experimental o entrenamiento, seguida de una fase de test para comprobar la validez de los resultados de la fase experimental.

V.2. Selección de parámetros

a) Umbrales:

Los parámetros que mayor influencia tienen en el resultado final son los tres umbrales de que consta el algoritmo: el umbral adaptativo, el fijo y el de seguridad. Así, se ha realizado una optimización conjunta del umbral adaptativo y del umbral fijo, probando diferentes combinaciones de T_a y T_L en una malla lo suficientemente densa como para conseguir un resultado satisfactorio. En este caso, se ha mantenido un valor fijo para el intervalo de guarda y para el

parámetro de memoria, aislando el comportamiento de los umbrales del resto de parámetros.

Esta malla, junto con el resultado de *Precision* y *Recall* se muestra en la Fig. 18, donde se pueden apreciar los valores utilizados para probar los umbrales: T_a entre 25% y 75% y T_L entre 85% y 100%.

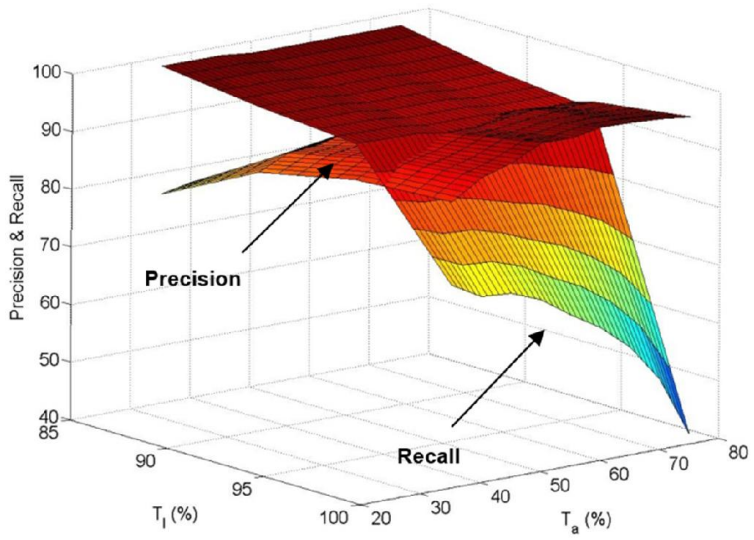


Fig. 18: *Precision* y *Recall* para distintos valores de Ta y TL

En la Fig. 18 se observan dos mallas tridimensionales, correspondientes a los valores de *Precision* y de *Recall* respectivamente, de forma que se puede comprobar el comportamiento conjunto de ambos parámetros al variar los valores de los umbrales. Al aumentar dichos umbrales aumenta la precisión (disminuye el número de falsas alarmas), pero disminuye la eficacia (aumenta el número de detecciones perdidas, lo que disminuye *Recall*).

Con este comportamiento, se puede establecer el mejor punto de funcionamiento como aquél que obtiene los mejores valores posibles de precisión y eficacia simultáneamente, seleccionando como punto de funcionamiento el mejor punto de intersección entre ambas mallas (mayor valor de cruce entre las mallas de precisión (*Precision*) y eficacia (*Recall*)).

Un análisis más pormenorizado del comportamiento de *Precision* y *Recall* se presenta en Fig. 19 y Fig. 20. En particular, Fig. 19 representa la relación entre precisión y eficacia para distintos valores del umbral adaptativo T_a, manteniendo fijo el valor de $T_L = 95\,\%$. En esta figura se puede observar cómo el rango de valores óptimos para el umbral adaptativo se sitúa alrededor del codo que se aprecia en la zona de valores altos de precisión y eficacia (correspondiente a valores de $T_a = 40\% - 55\%$).

Esta figura se ha obtenido a partir de los resultados de la codificación de una secuencia de la categoría *HM&HC* en una configuración de *bitrate* intermedio (1500 kbps a 25 imágenes por segundo).

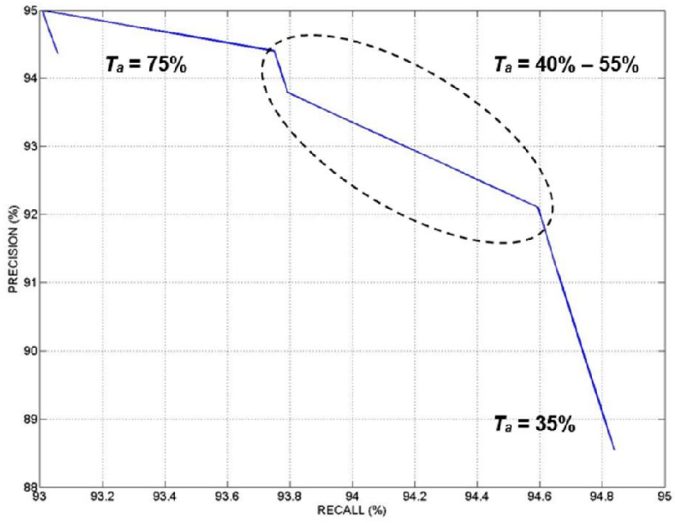


Fig. 19: *Precision* frente a *Recall* para distintos valores de Ta

Por otra parte, en la Fig. 20 se observa la superposición de gráficas similares a la de la Fig. 19 obtenidas al aplicar distintos valores del umbral fijo T_L.

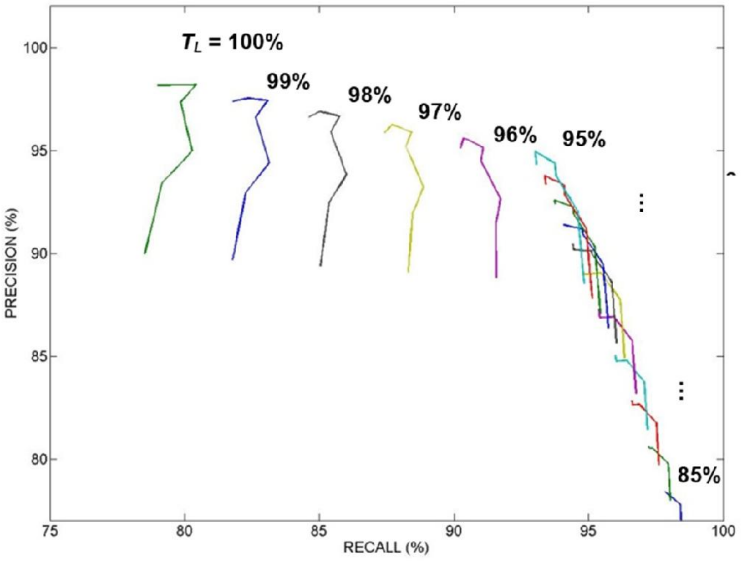


Fig. 20: *Precision* frente a *Recall* para distintos valores de Ta y TL

Se puede comprobar cómo, al aumentar el valor del umbral T_L aumenta la precisión pero disminuye la eficacia, obteniendo el comportamiento deseado en la región cercana al codo que se produce alrededor de los valores $T_L = 94\,\% - 96\,\%$.

En la Tabla 4 se presenta un resumen de los valores óptimos para los distintos parámetros del algoritmo:

FPS (fps)	Bitrate (kbps)	T_a (%)	T_L (%)	T_S (%)
12.5	**750**	40	96	98
25	**750**	45		
	1500	50		
	2000	55		

Tabla 4: Resumen de los valores óptimos para los umbrales

En la tabla se puede apreciar como los valores óptimos del umbral adaptativo aumentan con el *bitrate* y con la tasa de imágenes por segundo, manteniendo valores cercanos al 50%, mientras que los valores del umbral fijo y del umbral de seguridad se mantienen constantes en el 96% y el 98% del número total de macrobloques de una imagen, respectivamente. Finalmente, cabe resaltar nuevamente que el umbral de seguridad se sitúa ligeramente por encima del umbral fijo.

b) Parámetro de memoria:

Según se puede observar en la expresión de la Ec. 7, el valor del parámetro α controla la forma en la que m_k sigue el número de macrobloques *intra* de las imágenes P de la secuencia. Si α es demasiado bajo, la contribución de las imágenes pasadas es más importante que el valor de IMB de la última imagen codificada, obteniendo un valor de m_k suavizado que no sigue correctamente las variaciones abruptas de IMB, produciendo un umbral adaptativo demasiado estable y detectando un número demasiado elevado de cambios de plano, lo que aumenta la eficacia pero también el número de falsas alarmas. Por el contrario, un valor demasiado elevado de α provoca un seguimiento demasiado rápido de

los cambios bruscos de IMB, generando un umbral T_A demasiado alto e inestable y detectando menos cambios de plano de los deseados.

El valor óptimo de α se ha obtenido basándose en simulaciones del algoritmo [17] desarrolladas previamente a su implementación final en el codificador. Los resultados de dichas simulaciones han sido validados en distintas pruebas del codificador, y se presentan en la Tabla 5.

QCIF	CIF	SDTV
0.25	0.35	0.45

Tabla 5: Valores óptimos del parámetro de memoria

En esta tabla, los valores de α se presentan en una escala entre 0 y 1, y se observa que el valor del parámetro de memoria depende directamente de la resolución de la secuencia codificada, aumentando cuando el número de macrobloques de cada imagen se hace mayor. Sin embargo, el valor final seleccionado para α se sitúa siempre por debajo del 50%, asignando mayor peso en m_k al número de macrobloques *intra* de las imágenes anteriores que al de la última imagen codificada.

c) Intervalo de guarda:

La duración óptima del intervalo de guarda se ha establecido basándose en resultados experimentales con los vídeos del grupo de entrenamiento. En la Tabla 6 se puede apreciar su valor en milisegundos para los distintos grupos de vídeos en función de su movimiento.

LM&LC	MM&MC	HM&HC	COM
500			350

Tabla 6: Intervalo de guarda (en milisegundos) para los distintos grupos de vídeos

El intervalo de guarda general tiene una duración de 500 ms, lo que equivale a 12 imágenes codificadas cuando se utilizan 25 *frames* por segundo (6 imágenes a 12.5 fps, y así sucesivamente). Por otra parte, cuando la frecuencia de los cambios de plano es muy grande y aparecen otro tipo de efectos (transiciones graduales, flashes, etc.), es conveniente reducir la duración del intervalo de guarda, permitiendo la aparición de cambios de plano con menor separación temporal. Este tipo de situación se da especialmente en secuencias de vídeo como las de la categoría de Anuncios Comerciales (COM), para la que se ha fijado un intervalo de guarda de 350 ms.

VI. Resultados

En esta sección se presentan los resultados obtenidos como medida del comportamiento del mismo en distintos aspectos: capacidad de detección, mejora objetiva de calidad (PSNR) y tiempo de procesamiento. Del mismo modo, se presenta una comparación de los resultados de este algoritmo con otro método de reciente publicación.

VI.1. Precisión y eficacia

Para medir el comportamiento del algoritmo en cuanto a su capacidad de detección, se han utilizado como métricas la precisión y la eficacia, bajo la

forma de los parámetros *Precision* (P) y *Recall* (R), respectivamente. Con estos resultados se ha obtenido la Tabla 7, donde se aprecian los valores de precisión y eficacia para las cuatro categorías de movimiento, incluyendo además todas las configuraciones de *bitrate* y tasa de imágenes por segundo.

		LM&LC		**MM&MC**		**HM&HC**		**COM**	
FPS	***Bitrate***	**P(%)**	**R(%)**	**P(%)**	**R(%)**	**P(%)**	**R(%)**	**P(%)**	**R(%)**
12.5	**750**	100	100	93.14	99.05	96.16	96.25	82.60	96.86
25	**750**	99.42	99.42	99.44	96.19	98.43	93.47	81.34	97.81
	1500	99.43	100	98.89	98.93	98.48	96.05	82.11	98.03
	2000	98.90	100	99.43	98.92	98.72	97.54	82.89	98.30

Tabla 7: Precisión (P) y Eficacia (R) para las secuencias del Conjunto de Test, con todas las configuraciones y todas las categorías de movimiento

En la Tabla 7 se puede apreciar el buen comportamiento del algoritmo en términos de precisión y eficacia: el valor medio de precisión en la detección es superior al 94%, mientras que en la eficacia se sitúa por encima del 95%. En el caso mejor, ambos parámetros se sitúan por encima del 98.9%, mientras que en el peor caso, correspondiente a las secuencias con mucho movimiento (HM&HC), los resultados medios son superiores al 96%. En este sentido, los resultados correspondientes a las secuencias del grupo COM presentan un comportamiento más modesto en cuanto a precisión, ya que aumenta el número de falsas alarmas debidas a las múltiples transiciones graduales, flashes y otros eventos que incluyen estas secuencias. Esto empeora los valores de la precisión, pero el algoritmo consigue mantener la eficacia en valores medios superiores al 97%.

A la hora de comparar los resultados obtenidos con este algoritmo con los proporcionados por otros métodos es necesario seleccionar aquéllos que tengan una estructura similar. En este sentido, se ha llevado a cabo una comparación con el algoritmo de detección de cambios de plano presentado en [18], cuyo método presenta unas bases teóricas muy similares al método descrito en esta tesis. En concreto, se basa en la utilización de dos umbrales, uno fijo y el otro adaptativo, siguiendo la diferencia en el número de macrobloques *intra* entre imágenes consecutivas. En la Tabla 8 se pueden apreciar los resultados de precisión y eficacia del método de [18] para las mismas secuencias que las utilizadas para obtener Tabla 7.

		LM&LC				MM&MC			
FPS	***Bitrate***	**P(%)**	**Δ**	**R(%)**	**Δ**	**P(%)**	**Δ**	**R(%)**	**Δ**
12.5	**750**	98.43	-1.57	100	0	94.48	+1.34	92.22	-6.83
25	**750**	96.92	-2.5	100	+0.58	83.43	-16.01	93.77	-2.42
	1500	97.58	-1.85	100	0	83.18	-15.71	92.97	-5.96
	2000	98.28	-0.62	100	0	83.05	-16.38	92.31	-6.61
		HM&HC				**COM**			
FPS	***Bitrate***	**P(%)**	**Δ**	**R(%)**	**Δ**	**P(%)**	**Δ**	**R(%)**	**Δ**
12.5	**750**	89.83	-6.33	92.98	-3.27	77.28	-5.32	92.27	-4.59
25	**750**	96.62	-1.81	94.07	+0.6	75.8	-5.54	95.23	-2.58
	1500	95.71	-2.77	87.58	-8.47	74.92	-7.19	93.84	-4.19
	2000	96.37	-2.35	87.5	-10.04	74.66	-8.23	93.13	-5.17

Tabla 8: Precisión (P) y Eficacia (R) para el método presentado en [18] y comparación con el método de la UPV

En esta tabla se muestra también la diferencia de precisión y eficacia del método de [18] con respecto a la UPV. En la columna Δ, un valor positivo indica que el método de [18] es mejor que el de la UPV, mientras que un valor negativo indica la mejora proporcionada por el método de la UPV.

Como se puede ver en la tabla anterior, el detector desarrollado en la UPV obtiene unas tasas de detección notablemente mejores que el algoritmo [18], consiguiendo una mejora media en la precisión del 7% y una mejora media de la eficacia del 3%. Por otra parte, bajo ciertas condiciones, el nuevo algoritmo supera en un 16% al método de [18] en la comparación realizada (para el grupo de secuencias MM&MC).

VI.2. Tiempo de procesamiento

Una de las principales características de este algoritmo de detección de cambios de plano es su simplicidad, especialmente en lo que se refiere a su coste computacional. En la implementación realizada el algoritmo no introduce ninguna complejidad computacional añadida al proceso de codificación mientras no se produce una detección de cambio de plano. Durante su funcionamiento habitual, el algoritmo consiste únicamente en la actualización de un contador de macrobloques *intra* y su comparación con un umbral, que se actualiza *frame* a *frame*. En el caso de que la imagen que se está codificando sea detectada como un cambio de plano el detector debe realizar un procedimiento extraordinario, consistente en abortar la codificación de la imagen actual como P-*frame* y codificar dicha imagen como *intra*. En este apartado se presenta un análisis del comportamiento del algoritmo en términos de tiempo de procesamiento requerido, tanto para la detección local de cambios de plano como para el proceso global de codificación de una secuencia.

En primer lugar, para analizar el coste computacional introducido por la detección de un cambio de plano y su posterior codificación en modo *intra*, llamamos t_P al tiempo medio necesario para codificar una imagen en modo P, y t_I al tiempo medio necesario para codificar dicha imagen en modo I. Así, podemos llamar $t_{P\text{-}I}$ al tiempo necesario para detectar un cambio de plano durante la codificación *inter* de la imagen, abortar dicha codificación y recodificar la imagen en modo *intra*:

$$t_{P-I} = \gamma \times t_P + t_I$$

Ec. 11: Tiempo de recodificación

En esta ecuación $\gamma \in [0,1]$ indica el porcentaje de la imagen detectada como cambio de plano que había sido codificada en modo *inter* cuando se ha producido la detección.

	LM&LC	MM&MC	HM&HC	COM
$\frac{t_{P-I}}{t_P}$	0.9	1.03	1.10	1.05

Tabla 9: Tiempo de re-codificación frente a tiempo de codificación en modo P

En la Tabla 9 se puede apreciar la relación entre $t_{P\text{-}I}$ y t_P. Pese a que el tiempo requerido para codificar parcialmente una imagen, abortar dicha codificación y recodificar la imagen en modo *intra* es mayor que el tiempo de codificación de la imagen completa en modo *inter*, el aumento de tiempo observado es inferior al 10%, debido a que la codificación *inter* de las imágenes que se detectan como cambios de plano es muy poco eficiente, debido a la

escasa correlación con las imágenes de referencia (pertenecientes al plano anterior). Este aumento de tiempo es mayor cuanto más complejas son las imágenes de la secuencia, mientras que en escenas con poco movimiento se puede producir incluso una ganancia en tiempo al recodificar los cambios de plano en modo *intra*.

	LM&LC	MM&MC	HM&HC	COM
$\frac{(T_{ND}-T_D)}{T_{ND}}$	3.02%	2.36%	1.7%	2.2%

Tabla 10: Relación entre el tiempo de codificación de la secuencia completa con y sin detección de cambios de plano

Por otra parte, en la Tabla 10 se presenta un segundo análisis del coste computacional introducido por el algoritmo de detección. En este caso se analiza el tiempo total de codificación de una secuencia completa, comparando el tiempo de codificación cuando se utiliza el algoritmo de detección (T_D) y cuando no se utiliza ningún tipo de detección (T_{ND}). En el segundo caso no se inserta ninguna imagen *intra*, por lo que el formato de GOP es de tipo IP…P.

En la codificación de la secuencia completa se puede conseguir una ganancia en tiempo de procesamiento de hasta un 3%. Esto choca con el hecho de que la detección del cambio de plano y la recodificación en modo *intra* son más costosas que la codificación en modo *inter*, pero se puede explicar por la mejora proporcionada por la codificación *intra* de las imágenes detectadas como cambios de plano. Estas imágenes optimizan las referencias de los siguientes *frames*, mejorando su tiempo de codificación y con ello el tiempo de procesamiento de la secuencia completa.

Por este motivo, se puede asegurar que este algoritmo de detección de cambios de plano se puede utilizar en aplicaciones en tiempo real, ya que mantiene las condiciones de bajo retardo si el codificador original las cumplía.

VI.3. PSNR

Al analizar los resultados obtenidos por el algoritmo de detección en el proceso global de codificación de secuencias de vídeo se ha observado un efecto colateral que proporciona una mejora de la calidad global de la codificación, tanto en términos objetivos como subjetivos. Esto es debido a que, al insertar imágenes *intra* en las posiciones correspondientes a los cambios de plano, las referencias utilizadas por las siguientes imágenes tienen una mayor calidad, ya que la PSNR de las imágenes *intra* es mayor que la de imágenes *inter* cuyas referencias tienen una baja correlación por su pertenencia a distintos planos. El aumento de la calidad objetiva, medida por la PSNR, se corresponde en este caso con un aumento de la calidad subjetiva de las secuencias codificadas utilizando un codificador que incluye el algoritmo de detección de cambios de plano aquí descrito.

Este aumento de calidad obtenido al aplicar el algoritmo de detección constituye la base para el desarrollo posterior de dicho método, produciendo un cambio en su enfoque para orientarlo a la optimización de la codificación de vídeo y a la mejora de la eficiencia de codificación, como se verá en el siguiente capítulo de esta tesis.

VII. Conclusiones

En este capítulo se ha presentado un algoritmo rápido y robusto de detección de cambios de plano que ha demostrado tener unas buenas prestaciones en términos de tasa de detección, precisión y eficacia, superando en estos aspectos a otros métodos de reciente publicación. El algoritmo está basado en el uso de un umbral fijo y otro adaptativo, aplicados sobre el número de macrobloques *intra* utilizados para codificar cada imagen de la secuencia.

El desarrollo del algoritmo se ha orientado hacia la mejora en la eficiencia de codificación, por lo que se ha modificado el esquema de codificación original para introducir imágenes *intra* en las posiciones donde se detecta un cambio de plano, mejorando con ello las referencias utilizadas por las siguientes imágenes del nuevo plano. El algoritmo se ha optimizado para su aplicación a secuencias de definición estándar (SDTV), desarrollando un entorno de pruebas consistente en un conjunto de secuencias que permite realizar un entrenamiento del algoritmo y comprobar su funcionamiento con un total de más de 1500 cambios de plano. El método es válido para aplicaciones en tiempo real, consiguiendo una ganancia en tiempo de codificación con respecto a un codificador sin detección de cambios de plano.

Por último, la optimización de las referencias utilizadas en las regiones alrededor de los cambios de plano ha permitido obtener una ganancia en la calidad tanto objetiva (PSNR) como subjetiva de la secuencia codificada, lo que supone el punto de partida para la siguiente fase de este proyecto de investigación, destinado a mejorar la calidad del vídeo codificado y la eficiencia de codificación.

VIII. Líneas futuras

Como principales líneas de investigación para la evolución del algoritmo presentado en este capítulo se destacan las siguientes:

- Estudiar la implementación del detector automático de cambios de plano en un codificador H.265: dado que el algoritmo se basa en la monitorización del modo de codificación de cada macrobloque de las imágenes codificadas y que el diseño del codificador H.265 se basa en los mismos principios que el codificador H.264, es posible implementar el presente método en el nuevo estándar de codificación. La selección de los umbrales en el nuevo códec y el estudio de las variaciones necesarias en el algoritmo constituyen uno de los principales puntos de trabajo futuro obtenidos de los resultados presentados en este capítulo.
- Ampliar el espectro de formatos y resoluciones de las secuencias utilizadas: en este capítulo se han utilizado secuencias de resolución estándar SDTV. Sin embargo, cada día aumenta el número de aplicaciones que hacen uso de resoluciones mayores, como HDTV o incluso los nuevos formatos 4K (UHD). Los resultados de este proyecto se pueden extender a estas nuevas resoluciones mediante el estudio y la obtención de los umbrales y las configuraciones correspondientes, siguiendo un enfoque similar al utilizado en el actual proyecto.

Capítulo 3

Inserción de *Keyframes* Basada en el Contenido

I. Introducción

En el capítulo 2 de esta tesis se ha presentado un algoritmo para la detección automática de cambios de plano, orientado a mejorar la eficiencia de codificación, insertando el algoritmo de detección en la propia cadena de codificación. Con el esquema propuesto, los cambios de plano detectados se codifican en modo *intra*, ya que la primera imagen del nuevo plano tiende a tener una correlación baja con las imágenes del plano anterior, por lo que su codificación en modo *inter* es poco eficiente, ya que no se pueden aprovechar las herramientas de predicción compensada en movimiento.

El resultado de dicho capítulo ha sido un algoritmo de detección con buenas tasas de precisión y eficacia, y aplicable en entornos de codificación en tiempo real, ya que consigue ganancias en el tiempo total de codificación. Por otra parte, el esquema de codificación *intra* adoptado (GOP abierto con inserción de *intras* basada en el contenido) ha proporcionado como resultado paralelo la mejora de la eficiencia de codificación, al proporcionar una ganancia local en la calidad objetiva del vídeo codificado.

Esta ganancia viene dada por la mencionada baja correlación que las imágenes correspondientes a los cambios de plano tienen con sus imágenes de referencia (pertenecientes al plano anterior). Pero los cambios de plano no son las únicas situaciones en las que se da esta baja correlación entre imágenes consecutivas, por lo que este factor se puede explotar para mejorar todavía más la eficiencia de codificación y la calidad objetiva del vídeo codificado. En particular, existen distintos tipos de situaciones que tienden a disminuir la correlación entre imágenes, como pueden ser los movimientos abruptos, las oclusiones, flashes, transiciones graduales, etc.

Por lo tanto, el objetivo de este capítulo ha consistido en la modificación del algoritmo de detección de cambios de plano para adaptarlo al nuevo enfoque planteado: seleccionar las posiciones óptimas para la inserción de *keyframes* en la secuencia para maximizar la calidad objetiva del vídeo codificado, manteniendo las características de tiempo real del sistema de codificación. En concreto, dichas posiciones óptimas coinciden con los puntos de la secuencia donde la correlación entre imágenes consecutivas es suficientemente baja.

En este sentido, es necesario comprobar si, al igual que en el caso de los resultados de precisión y eficacia de detección, la ganancia local en PSNR alrededor de cada candidato a la inserción de *keyframe* detectado depende también de los parámetros del algoritmo de detección. Por este motivo, el algoritmo presentado en la primera parte de este documento puede ser modificado y es susceptible de un entrenamiento como el descrito en el apartado 2-V, destinado en este caso a maximizar la calidad objetiva de la secuencia codificada. Por lo tanto, el parámetro a tener en cuenta a la hora de analizar las prestaciones del nuevo algoritmo ya no es la tasa de detección, sino la PSNR obtenida en la codificación de la secuencia.

Este capítulo se organiza de la siguiente forma: en primer lugar se presentan las bases de la inserción de *keyframes* basada en el contenido; a continuación se describe el algoritmo propuesto, seguido de las puntualizaciones necesarias sobre el entorno de pruebas utilizado para el entrenamiento, desarrollo y test de este método. Finalmente se presenta la aplicación comercial del método aquí descrito junto a las conclusiones obtenidas en este análisis.

II. Inserción de *keyframes* basada en el contenido

En la indexación de vídeo, las *Key Frames* o imágenes de referencia son imágenes que representan de forma concisa un plano [25]. Estas imágenes deben descartar toda la redundancia posible para determinar las características más importantes de cada plano.

Las características consideradas incluyen colores, formas, bordes, flujo óptico, descriptores de movimiento y actividad de movimiento (MPEG-7), coeficientes de la DCT, vectores de movimiento, movimiento de cámara, etc.

En general, la extracción de imágenes de referencia se clasifica en seis categorías:

1. **Basada en comparación secuencial**: compara secuencialmente cada imagen de la secuencia con el último *keyframe* extraído hasta que una imagen es suficientemente diferente de dicho *keyframe*. Se puede utilizar la diferencia de histogramas de color u otras métricas. Estos métodos son simples, intuitivos y requieren baja complejidad computacional. Además, se adapta el número de imágenes de referencia a la duración de la secuencia. Las contrapartidas incluyen que las imágenes de referencia representan propiedades locales del plano más que globales; también son inconvenientes la distribución irregular de imágenes de referencia a lo largo de la secuencia o la aparición de redundancia cuando hay contenido que aparece repetidamente a lo largo de la secuencia.
2. **Basada en comparación global**: estos métodos distribuyen las imágenes de referencia minimizando funciones predefinidas que dependen de la aplicación. Pueden usar una varianza temporal justa,

máxima cobertura, mínima correlación o mínimo error de reconstrucción. En estos métodos las imágenes de referencia representan las características globales de la secuencia, el número de imágenes de referencia es controlable y el conjunto de imágenes de referencia es más conciso y menos redundante que en el caso secuencial. Las limitaciones incluyen su mayor complejidad computacional.

3. **Basada en imágenes de referencia**: estos métodos generan una imagen de referencia sintética y comparan las imágenes de cada plano con dicha imagen. Se detectan los *keyframes* como picos en la curva de distancia entre cada imagen y la de referencia. Estos métodos son sencillos, pero si la imagen de referencia no representa correctamente el plano se puede perder parte de la información buscada.

4. **Basada en *clustering***: estos algoritmos agrupan imágenes y seleccionan las imágenes más cercanas al centro del grupo como imágenes de referencia. Pueden usar algoritmos genéricos de agrupación y extraen correctamente las características globales del vídeo. Sin embargo, dependen de los resultados del *clustering* y la agrupación correcta es muy difícil de precisar. Además, la agrupación de conjuntos secuenciales no es muy precisa y es necesario utilizar trucos para asegurar que imágenes consecutivas sean agrupadas juntas con la mayor probabilidad posible.

5. **Basada en simplificación de curvas**: representan cada imagen como un punto en el espacio de características, formando una trayectoria en la que se buscan puntos que definan correctamente la forma de la curva. La información secuencial se preserva en la extracción de *keyframes*,

pero la optimización de la mejor representación de la curva es computacionalmente compleja.

6. **Basada en objetos/eventos**: consideran de forma conjunta la extracción de *keyframes* y la detección de objetos/eventos, asegurando que las imágenes de referencia contienen información de estos objetos/eventos. Las imágenes extraídas son semánticamente representativas y reflejan los objetos o los patrones de movimiento de los mismos. Sin embargo, dado que la detección de objetos se basa en métodos heurísticos, está muy ligada a la aplicación y por tanto a la configuración del sistema.

En general, dada la subjetividad del concepto *keyframe*, no hay un método uniforme de evaluación del funcionamiento de estos métodos. Normalmente se utiliza la tasa de error y el ratio de compresión como medidas. Así, las imágenes de referencia que dan bajas tasas de error y alta tasa de compresión son preferibles. Hay que tener en cuenta, sin embargo, que normalmente una baja tasa de error lleva asociada una baja tasa de compresión, por lo que se debe buscar un compromiso.

En nuestro caso el concepto de *keyframe* se limita a la determinación de las posiciones donde introducir imágenes *intra*, debido a que tienen un contenido diferente a las codificadas previamente. Por lo tanto, de entre los métodos anteriores, este desarrollo se enmarca principalmente dentro de los de la primera categoría. Sin embargo, el encaje no es total, ya que las imágenes de la secuencia no se comparan con la última imagen de referencia, sino con la imagen inmediatamente anterior, que a su vez se basa en la anterior y así sucesivamente hasta llegar efectivamente a la imagen de referencia original (*keyframe*).

Por lo tanto, según lo descrito anteriormente, la determinación de las mejores imágenes de referencia se reduce a la determinación de las posiciones óptimas para las imágenes codificadas en modo *intra-frame*.

III. Descripción del algoritmo

El algoritmo de selección de imágenes *intra* (*keyframes*) está basado directamente en el algoritmo de detección de cambios de plano presentado en el apartado 2-III.2, añadiendo ciertas modificaciones para optimizar su funcionamiento. Las características básicas permanecen inalteradas, basándose en el uso de dos umbrales, uno fijo y el otro adaptativo, aplicados sobre el número de macrobloques *intra* utilizados para codificar cada imagen P de la secuencia.

En concreto, el umbral adaptativo (T_A) y el umbral fijo (T_L) tienen la misma formulación que en el caso del algoritmo original, según las expresiones Ec. 6, Ec. 7 y Ec. 8, incluyendo el proceso con memoria para la actualización del número medio de macrobloques *intra* de las imágenes del GOP actual. Nuevamente, la combinación de estos umbrales mediante una operación de mínimo, da lugar al umbral dinámico T_D.

La principal diferencia que se ha introducido ha sido la sustitución del intervalo de guarda fijo por una función de decrecimiento exponencial adaptativa que hace menos restrictivas las condiciones de inserción de *keyframes* con poca separación temporal entre ellos, dando lugar a un intervalo de guarda adaptativo. Este cambio lleva consigo la eliminación del umbral de seguridad (T_S), y la creación de un único umbral universal *T(k)* que incluye tanto el intervalo de guarda como el funcionamiento en modo normal del

algoritmo. En este sentido, basándonos en [19][20], se ha optado por introducir un offset $\Delta T(k)$ al umbral dinámico $T_D(k)$:

$$T(k) = \min(T_D(k) + \Delta T(k), T_L)$$

Ec. 12: Umbral universal

El offset introducido decrece progresivamente tras cada inserción, de forma que el umbral resultante se aproxima gradualmente al valor marcado por el umbral dinámico correspondiente:

$$\Delta T(k) = A_{K_{LK}} \times f(k - k_{LK})$$

Ec. 13: Offset sobre el umbral dinámico

En la ecuación anterior k_{LK} es la posición del último *keyframe* insertado, $A_{k_{LK}}$ controla la amplitud del factor añadido, y *f(t)* es la función de decrecimiento aplicada al offset, en función de la distancia entre la imagen actual y la última imagen *intra* insertada ($k - k_{LK}$).

La función de decrecimiento puede tomar distintos aspectos en función del comportamiento deseado, habiendo escogido un decrecimiento exponencial [20], con la siguiente expresión:

$$f(t) = \exp\left(\frac{t}{\tau}\right)$$

Ec. 14: Función de decrecimiento exponencial

En la Ec. 14, τ controla la velocidad de decrecimiento y t es el número de imagen dentro del GOP actual.

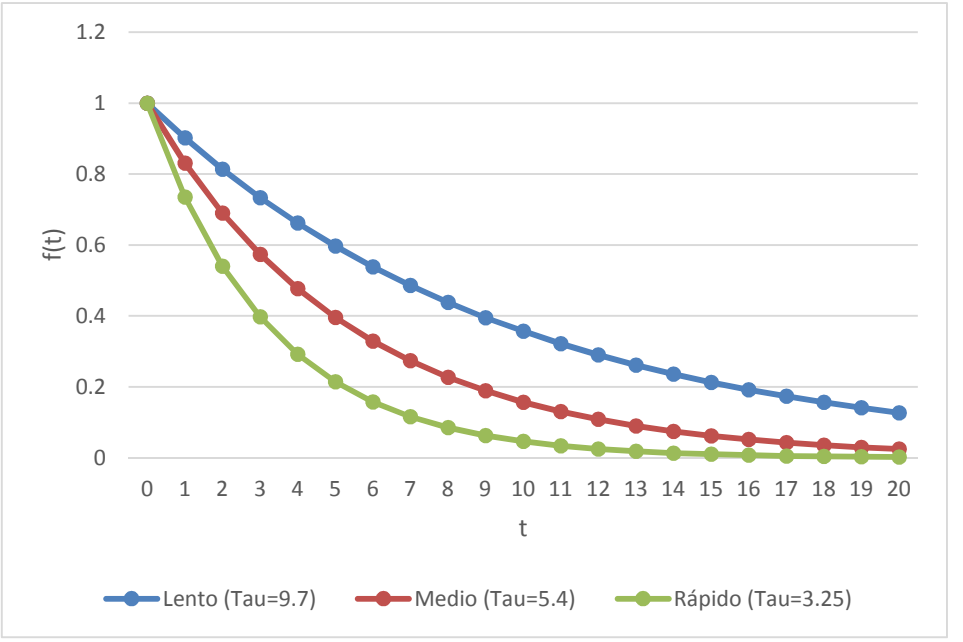


Fig. 21: Comportamiento de *f(t)*

En la Fig. 21 se pueden observar tres ejemplos distintos del comportamiento de esta función para un decrecimiento lento, medio y rápido, correspondientes a valores de τ de 9.7, 5.4 y 3.25 respectivamente.

Con estas definiciones, el funcionamiento en el intervalo de guarda queda definido: tras cada inserción de *keyframe*, tenemos que $k - k_{LK} = 0$; con este valor, el offset $\Delta T(k)$ toma el valor máximo $A_{k_{LK}}$, elevando el umbral y estableciendo una condición más estricta para insertar un nuevo *keyframe* a poca distancia del anterior. A continuación, durante el intervalo definido por τ, la función de decrecimiento disminuye el valor del offset, haciendo que el umbral $T(k)$ se aproxime progresivamente a $T_D(k)$, ya que:

$$\lim_{t}\big(f(t)\big) = 0$$

Ec. 15: Límite de la función de decrecimiento exponencial

Por lo tanto, al aumentar la distancia temporal disminuye exigencia del umbral aplicado.

Por otra parte, el algoritmo de inserción de imágenes *intra* se ha integrado completamente en el proceso de codificación, por lo que también se tiene en cuenta la monitorización del tamaño del GOP abierto para evitar la propagación indeseada de errores en entornos de transmisión problemáticos. En este caso, se ha introducido un intervalo máximo entre *keyframes*, limitando el tamaño máximo de GOP a K_{max} imágenes. Así, este límite se aplica si antes de codificar la imagen *k* su distancia al último *keyframe* (k_{LK}) cumple:

$$k - k_{LK} > K_{max}$$

Ec. 16: Tamaño máximo de GOP

En este caso, la imagen *k* se codifica en modo *intra*, independientemente de su contenido y de su correlación con las imágenes anteriores.

En la Fig. 22 se puede observar un diagrama de flujo correspondiente al funcionamiento completo de este algoritmo de inserción de *keyframes*. En dicho diagrama aparece el proceso de codificación de una secuencia completa: la primera imagen de la secuencia se codifica forzosamente en modo *intra* ($k_{LK} = k = 0$), estableciendo los valores iniciales para los parámetros del algoritmo; cuando la siguiente imagen ($k+1$) está preparada para ser codificada, se actualiza el valor de los distintos umbrales que forman el umbral universal *T(k)*,

basándose en la información recogida durante la codificación de las imágenes anteriores (m_k). En este momento, si se ha superado el intervalo máximo entre *keyframes* (K_{max}), se fuerza la codificación de la imagen k en modo *intra*, mientras que si no se ha superado el límite, se inicia su codificación en modo *inter*. En este caso, el contador de macrobloques *intra* (IMB_k) se inicializa a 0 y se actualiza cada vez que un macrobloque de la imagen k se codifica en modo *intra*, comparando su valor con el umbral universal. Si durante la codificación de dicha imagen se supera el umbral, la codificación en modo *inter* se aborta, recodificando la imagen k en modo *intra* y actualizando los parámetros correspondientes del nuevo GOP que se inicia. Por el contrario, si todos los macrobloques de la imagen se han codificado sin superar el umbral de macrobloques *intra*, se finaliza la codificación en modo *inter* y se actualiza la métrica de continuidad m_k.

Una vez descrito el funcionamiento del algoritmo, en la Fig. 23 se puede apreciar un ejemplo del comportamiento que se obtiene al aplicarlo a la codificación de una secuencia real. En dicha figura se muestra el porcentaje de macrobloques *intra* de un conjunto de *frames* codificados (línea continua), junto con el umbral universal aplicado en cada una de ellas (línea discontinua). Señaladas mediante un punto rojo se aprecian cuatro inserciones de imágenes *intra*, en los *frames* 131, 160, 193 y 238.

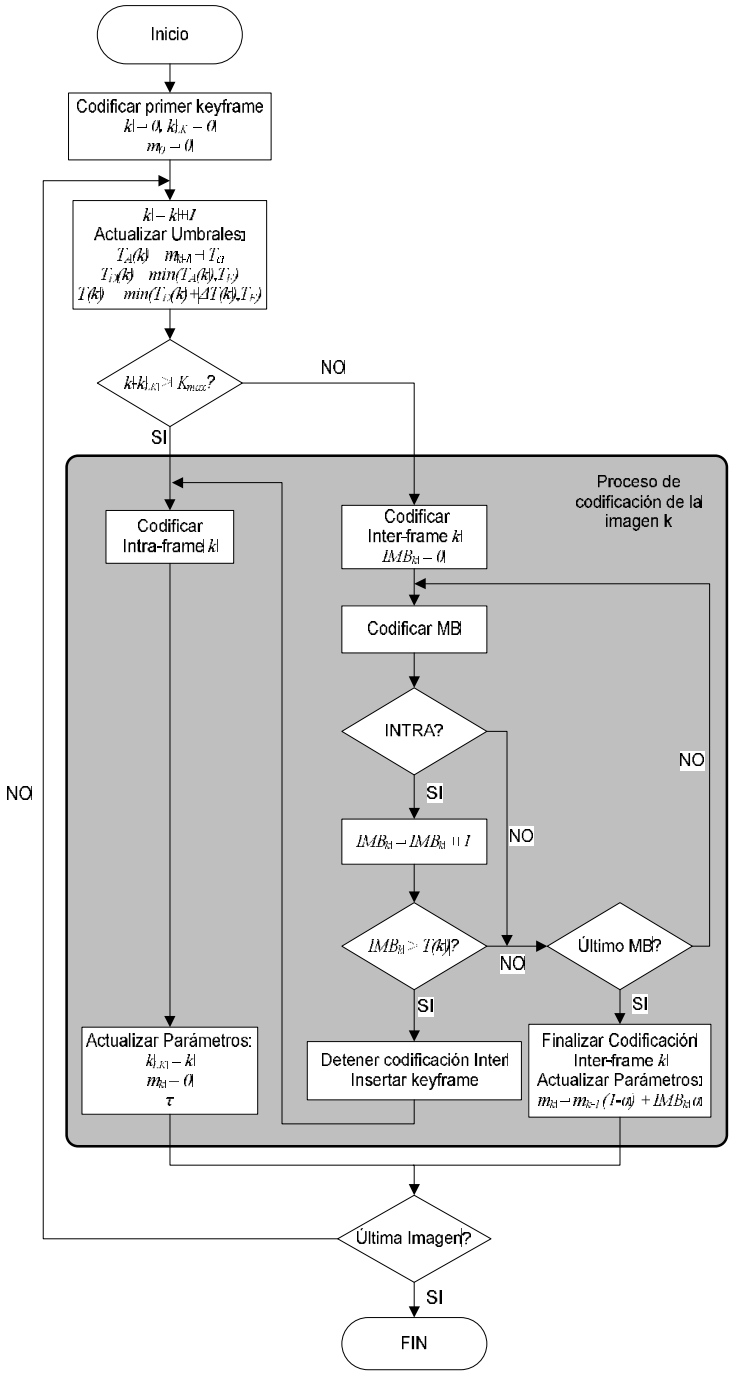


Fig. 22: Diagrama de flujo del algoritmo de inserción de *keyframes*

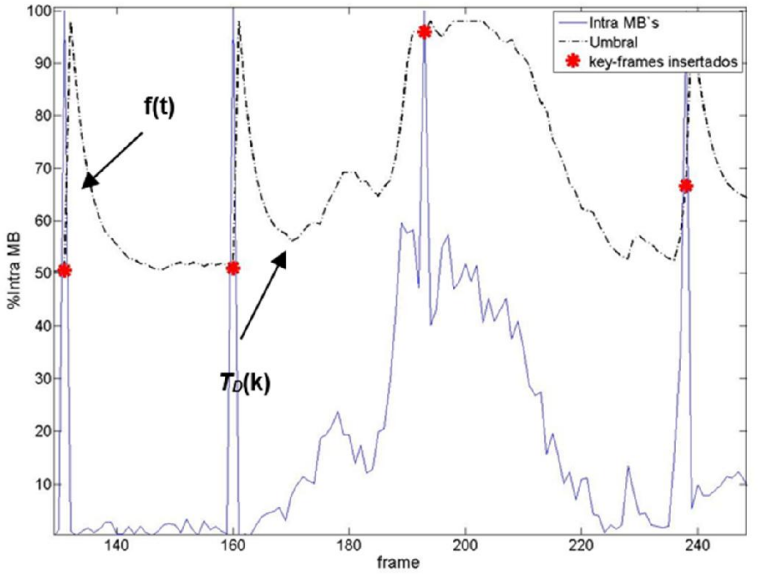


Fig. 23: Ejemplo real de funcionamiento del algoritmo de inserción de *keyframes*

El comportamiento de las distintas etapas de funcionamiento del algoritmo es el siguiente: las dos primeras inserciones se producen en escenas con poco movimiento, por lo que una disminución brusca de la correlación es fácilmente detectada con aproximadamente el 50% de los macrobloques codificados; por otra parte, la tercera inserción se produce en una escena con mayor movimiento en pantalla, por lo que la media de IMB es mayor, y la detección se produce cuando se han codificado más de un 90% de los macrobloques de la imagen. Finalmente, tras cada detección se aplica la función de decrecimiento exponencial *f(t)*, que empieza forzando un umbral muy restrictivo para decrementarlo progresivamente hasta alcanzar el umbral dinámico $T_D(k)$. La

velocidad de decrecimiento depende directamente del parámetro τ de la función exponencial.

IV. Entorno de pruebas

El entorno de pruebas utilizado para medir las prestaciones del algoritmo de inserción de imágenes *intra* es el mismo que se utilizó para el análisis de las prestaciones del algoritmo de detección de cambios de plano, utilizando el mismo codificador y las mismas secuencias (2-IV.1), divididas en categorías de movimiento y agrupadas en los conjuntos de entrenamiento y test.

En este caso no existe un *ground truth* que indique las posiciones correctas para las imágenes *intra*, sino que se trata de encontrar aquellas posiciones que proporcionen una mayor ganancia local en PSNR que aumente la calidad de la secuencia codificada.

Las principales diferencias en el entorno de pruebas entre la primera parte del proyecto y esta segunda fase se centran en las configuraciones de calidad (*bitrate* y tasa de imágenes por segundo) y la medida de prestaciones del algoritmo desarrollado.

IV.1. Configuraciones

El algoritmo de detección de cambios de plano se optimizó para su utilización en aplicaciones de alta calidad (alta tasa de bits por segundo y de imágenes por segundo). En esta segunda parte, el abanico de aplicaciones es mucho más amplio, por lo que se han utilizado configuraciones de calidad alta, media y baja para los tres formatos de vídeo considerados (SDTV, CIF y

QCIF). Para ello se ha llevado a cabo un estudio que relaciona la calidad subjetiva del vídeo codificado (percibida por los espectadores) con la calidad objetiva medida a través de la PSNR.

Calidad Subjetiva (Perceptual)	Calidad Objetiva (PSNR)
Baja (L)	x < 35 dB
Media (M)	35 dB < x < 40 dB
Alta (H)	x > 40 dB

Tabla 11: Relación entre calidad subjetiva (perceptual) y calidad objetiva (PSNR) del vídeo codificado

En la Tabla 11 se puede comprobar cómo valores de PSNR por debajo de 35 dB son considerados de baja calidad subjetiva, mientras que los que se sitúan por encima de 40 dB son de alta calidad. Hay que tener en cuenta que la medida objetiva de la calidad perceptual de vídeo es una de las áreas de investigación más novedosas, por lo que el estudio realizado ha sido meramente empírico, utilizando las secuencias de los grupos de entrenamiento y test como material de prueba, con el objetivo de obtener una primera aproximación a la relación buscada.

Por otra parte, en la Tabla 12 se muestran las configuraciones de *bitrate* y tasa de imágenes por segundo que se han utilizado para entrenar y probar el algoritmo de inserción de *keyframes* basado en el contenido junto con la calidad subjetiva correspondiente a cada configuración.

	QCIF		CIF		SDTV	
FR (fps)	**BR**	**CS**	**BR**	**CS**	**BR**	**CS**
25			100	L	250	L
			200	M	500	M
			300	H	1000	H
12.5	20	L	50	L	150	L
	50	M	100	M	250	M
	100	H	200	H	500	H
6.25	20	L	50	L		
	50	M	100	M		
	100	H	200	H		

Tabla 12: Configuraciones de bitrate (BR) y frame rate (FR) para los distintos formatos de imagen

La columna BR de la Tabla 12 indica el *bitrate* de la configuración correspondiente (kbps), mientras que la columna CS indica la calidad subjetiva asociada a dicho *bitrate*. Se han definido aplicaciones de calidad alta (H), media (M) y baja (L) para todas las configuraciones, con dos excepciones: en aplicaciones con formato QCIF, la baja tasa de bit típica con esta resolución hace que aumentar el número de fps no produzca un aumento de calidad (25 fps, QCIF); por otra parte, en aplicaciones de gran formato (SDTV), la reducción en el número de fps (6.25) proporciona experiencias de visionado pobres, por lo que ambas configuraciones han sido eliminadas.

IV.2. Medida de prestaciones

Para medir las prestaciones del algoritmo de inserción de *keyframes* se ha recurrido a una filosofía totalmente distinta que en el caso del algoritmo de detección de cambios de plano. En este caso no tienen sentido las medidas de precisión y eficacia, ya que no existen unas entidades definidas que tengan que ser detectadas, sino que se trata de escoger las imágenes cuya codificación en modo *intra* puede ayudar a mejorar la eficiencia de codificación de la secuencia completa.

La codificación en modo *inter* de una imagen con muy poca correlación con sus referencias es muy poco eficiente, siendo aconsejable su codificación en modo *intra*. Esta codificación *intra* produce una mejora de la calidad objetiva y subjetiva del *keyframe* seleccionado, lo que tiene consecuencias directas en la PSNR de las imágenes inmediatamente siguientes. Esta influencia se da especialmente en las primeras imágenes del GOP, ya que éstas hacen un uso explícito del *keyframe* insertado para obtener las predicciones en la codificación de sus macrobloques. Sin embargo, cuando avanza la codificación del nuevo GOP, la influencia del *keyframe* inicial se diluye, ya que las predicciones utilizadas provienen de imágenes alejadas de la primera imagen.

Así, se puede concluir que la mejora en la calidad introducida por la inserción de un *keyframe* en una posición óptima se concentra en unas pocas imágenes tras la inserción, mejorando la ganancia cuanto más cerca se encuentre la imagen del inicio del GOP. Por lo tanto, a la hora de medir las prestaciones del algoritmo hay que tener en cuenta este comportamiento, de forma que se ha escogido como medida de calidad la PSNR local en una ventana de cinco imágenes tras cada inserción aplicada por el algoritmo. En concreto, se toma como PSNR local el valor medio de la PSNR de la imagen

intra insertada y las cinco imágenes siguientes. De esta forma, es posible comparar la PSNR local de la secuencia codificada con el codificador que incluye el algoritmo de inserción óptima de *keyframes* con un codificador que no implementa esta característica.

Finalmente, hay que tener en cuenta que el conjunto de mejoras de calidad locales producidas por la codificación en modo *intra* del conjunto óptimo de imágenes de la secuencia producen una mejora global de calidad tanto objetiva como subjetiva de la secuencia codificada completa.

V. Entrenamiento del algoritmo

V.1. Justificación teórica

Nuevamente, la optimización de los valores de los distintos parámetros del algoritmo de inserción de *keyframes* se debe realizar de forma empírica, basándose en un entrenamiento experimental (ver apartado 2-V.1). En este caso, la función de coste a maximizar es la PSNR local de la secuencia obtenida al codificar el vídeo utilizando el nuevo algoritmo. Sin embargo, el establecimiento de un modelo analítico para relacionar la PSNR local con los valores de los parámetros del algoritmo es un problema difícilmente abarcable, por lo que la sintonización de los valores óptimos de los parámetros se ha realizado de forma empírica, utilizando las secuencias de Entrenamiento.

V.2. Selección de parámetros

En este apartado se presenta la selección final de los valores de los parámetros del algoritmo de inserción de imágenes *intra*, divididos en distintas categorías:

a) Umbrales:

Para obtener el valor óptimo del umbral adaptativo T_a se ha aprovechado el comportamiento observado en la ganancia de PSNR local obtenida al aplicar este algoritmo: si el umbral se sitúa demasiado bajo, un número demasiado elevado de imágenes serán señaladas como *keyframes*, consiguiendo menor ganancia debido al mayor consumo de bits de las imágenes *intra*; por el contrario, si el umbral aplicado es demasiado elevado, un número demasiado reducido de imágenes serán codificadas en modo *intra*, disminuyendo también la ganancia local. Por lo tanto, el funcionamiento óptimo del algoritmo se obtiene cuando el umbral se sitúa en una posición que obtiene la cantidad apropiada de *keyframes*, proporcionando la ganancia máxima en PSNR local.

Para obtener este valor óptimo de T_a se ha calculado la ganancia en PSNR local obtenida en un amplio rango de valores, entre el *25%* y el *75%* de los macrobloques de una imagen, manteniendo fijos el resto de los parámetros. El comportamiento observado se describe en la Fig. 24.

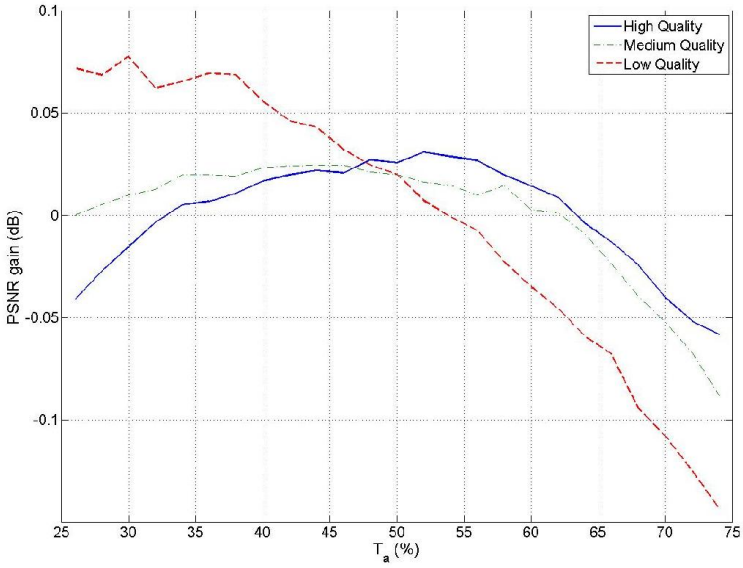


Fig. 24: Relación entre la ganancia en PSNR local y el valor del umbral adaptativo Ta

En esta figura se pueden observar tres líneas, correspondientes a la ganancia en PSNR local obtenida para configuraciones de calidad alta (línea continua), media (línea con puntos y rayas) y baja (línea punteada). Los datos de la figura se han calculado teniendo en cuenta los resultados obtenidos al utilizar todos los vídeos del conjunto de entrenamiento en todas las configuraciones.

Calidad	Baja	Media	Alta
T_a**(%)**	30	50	55

Tabla 13: Valor óptimo del umbral Ta para las distintas calidades

En la Tabla 13 se puede observar el valor óptimo del parámetro T_a para las distintas configuraciones de calidad (L, M, H). El comportamiento del umbral proporciona un punto óptimo de funcionamiento para cada configuración, pero la ganancia obtenida en un intervalo alrededor del óptimo es muy similar al óptimo, por lo que es posible sintonizar un valor común como compromiso para todas las configuraciones de calidad. De esta forma se independiza la sintonización de los parámetros del algoritmo de las características del vídeo codificado. En concreto, el valor óptimo calculado se sitúa en $T_a = 38\%$.

Por otra parte, el valor de T_L se ha escogido basándose en trabajos previos y en los resultados de la primera parte de esta tesis, para establecer un valor final de $T_L = 95\%$. En este sentido, hay que tener en cuenta que la influencia de este parámetro es residual en el funcionamiento global del algoritmo, ya que al tratarse de un mecanismo de seguridad, un número reducido de inserciones de *keyframes* se produce por la actuación de este umbral.

b) Función de decrecimiento exponencial:

La definición de la función de decrecimiento exponencial implica la selección tanto del valor de amplitud $A_{k_{LK}}$ como del parámetro de velocidad de decrecimiento τ.

El valor de la amplitud de la exponencial tras cada inserción se obtiene forzando un valor inicial de $T(k) = 98\%$ a partir de las ecuaciones Ec. 13 y Ec. 14, mientras que el valor del parámetro τ se obtiene de forma dinámica tras cada inserción. Para realizar esta elección se tiene en cuenta la distancia media entre *keyframes* consecutivos (I_{KF}), calculado mediante un proceso con memoria:

$$I_{KF} = I_{KF} \times (1 - \beta) + (k - k_{LK}) \times \beta$$

Ec. 17: Distancia media entre *keyframes* consecutivos

En esta ecuación, $\beta = 0.5$ es el parámetro de memoria y (k– k_{LK}) es la distancia desde el nuevo *keyframe* insertado y el inmediatamente anterior (k_{LK}).

Como se mostró en la Tabla 1, la distancia media entre cambios de plano es un buen indicador de la complejidad de la secuencia, por lo que se ha escogido un esquema de actualización del parámetro de decrecimiento exponencial basado en dicha distancia:

$$\tau = -\frac{I_{KF}}{2 \times \ln(0.01)}$$

Ec. 18: Parámetro de la función de decrecimiento exponencial

Según la Ec. 18 la función *f(t)* pasa de 1 a 0.01 en $I_{KF}/2$ imágenes.

Así, el umbral decrece más rápidamente en escenas de mayor complejidad, alcanzando más rápidamente el valor del umbral adaptativo, mientras que en escenas de complejidad reducida el decrecimiento es más lento y el intervalo de guarda se hace mayor.

c) Intervalo máximo entre keyframes:

El intervalo máximo entre imágenes *intra* consecutivas se calcula de forma que se evite la aparición de intervalos demasiado grandes entre *keyframes*, lo que podría dar lugar a la propagación indeseada de errores de decodificación.

El escenario de aplicación en el cual se ha desarrollado el presente algoritmo está libre de errores en la trama codificada. Sin embargo, en aplicaciones de codificación en entornos reales, el valor del intervalo máximo entre *keyframes* se puede calcular en función de la tasa de error presente en el canal, y basándose en el intervalo medio entre *keyframes*, insertando una imagen *intra* siempre que se supere el umbral determinado:

$$K_{max} = p \times I_{KF}$$

Ec. 19: Cálculo del tamaño máximo de GOP

En esta ecuación p es el número de veces que se tiene que sobrepasar I_{KF} para que se produzca una inserción debida al intervalo máximo aquí descrito.

V.3. Resultados del entrenamiento

Una vez finalizada la etapa de entrenamiento, se ha realizado un análisis de las posiciones donde la inserción de imágenes *intra* produce una mayor ganancia en PSNR. En la Tabla 14 se muestra un resumen de las características principales de las posiciones óptimas detectadas:

- **Cambios de plano abruptos**: la primera imagen de cada nuevo plano presenta una correlación potencialmente muy baja con las imágenes del plano anterior, por lo que constituye el candidato principal para la inserción de *keyframes*.

(a) (b) (c) (d) (e)

Fig. 25: Ejemplo de cambio de plano abrupto (c)

- **Oclusiones**: cuando un objeto aparece en escena o atraviesa la imagen, ocultando el fondo y los objetos del plano, la correlación con el resto de imágenes del plano disminuye, estableciendo puntos en los cuales la inserción de *keyframes* puede proporcionar una ganancia en PSNR. Un ejemplo de este tipo de situación se puede observar en la Fig. 26, donde la oclusión se produce en la imagen (c).

(a) (b) (c) (d) (e)

Fig. 26: Ejemplo de oclusión (c)

- **Movimiento extremo**: un movimiento extremo de cámara o de los objetos puede hacer disminuir la correlación, haciendo que dichas imágenes sean buenos candidatos para la inserción de *keyframes*.

(a) (b) (c) (d) (e)

Fig. 27: Ejemplo de movimiento extremo

- **Otros**: la aparición de transiciones graduales, flashes, cortinillas y otras situaciones también pueden dar lugar a la reducción de la correlación.

(a) (b) (c) (d) (e)

Fig. 28: Ejemplo de transición gradual.

Característica	Porcentaje
Cambio de plano abrupto	94.6%
Oclusión	2.7%
Movimiento Extremo	1.45%
Otros	1.25%

Tabla 14: Porcentaje de aparición de características en los *keyframes* detectados

En la Tabla 14 se puede apreciar como prácticamente el 95% de los *keyframes* insertados corresponde con cambios de plano abruptos, seguidos de oclusiones (2.7% de los casos), y movimientos extremos en escena. Los ejemplos incluidos en la categoría *Otros* constituyen condiciones especiales que pueden ser tratadas de forma particular por parte del codificador correspondiente.

VI. Resultados

En esta sección se presenta una valoración de los resultados del algoritmo, tanto en términos de ganancia en PSNR (local y global), como de tiempo de procesamiento. Finalmente, se ha llevado a cabo una comparación del funcionamiento del nuevo algoritmo con otros métodos similares.

VI.1. Ganancia en PSNR

En primer lugar, se ha comparado la PSNR local obtenida al codificar las mismas secuencias con un codificador que no inserta ninguna imagen *intra* y con un codificador que implementa el algoritmo aquí presentado. En este sentido, en la Tabla 15 se presenta la ganancia en PSNR local (medida en dB) obtenida al aplicar dicho algoritmo en las diferentes configuraciones.

		QCIF		CIF		SDTV	
FR (fps)	**Calidad**	**E**	**T**	**E**	**T**	**E**	**T**
25	L			1.37	1.59	1.32	1.50
	M			1.47	1.63	1.38	1.56
	H			1.43	1.59	1.34	1.51
12.5	L	1.33	1.56	1.14	1.22	0.84	0.97
	M	1.49	1.72	1.09	1.22	0.84	1.00
	H	1.74	1.89	1.04	1.20	0.84	0.97
6.25	L	0.87	1.08	0.62	0.72		
	M	0.97	1.20	0.70	0.85		
	H	1.11	1.33	0.66	0.81		

Tabla 15: Ganancia en PSNR local (dB) para las distintas configuraciones de calidad y formatos

La ganancia para el conjunto de vídeos de entrenamiento (columna E) y de test (columna T) se encuentra siempre dentro del mismo rango dentro de cada categoría y formato, por lo que se demuestra la correcta distribución de las secuencias en ambos conjuntos.

En un análisis más profundo de la Tabla 15, se puede concluir que, para cada valor de *frame rate* (FR), el valor de la ganancia para distintas configuraciones de calidad es muy similar, con una dependencia muy baja en la calidad subjetiva de la secuencia codificada. En general, el algoritmo proporciona mejores resultados cuando la tasa de imágenes por segundo es más elevada, obteniendo una ganancia media de 1.17 dB (considerando la Tabla 15 en su conjunto). Por otra parte, para cada configuración de calidad, la ganancia en PSNR local aumenta cuando la resolución del vídeo disminuye.

Por otra parte, la Tabla 16 presenta un segundo análisis de la ganancia en PSNR local obtenida al aplicar el algoritmo aquí presentado con respecto a un codificador que no inserta ninguna imagen *intra* en la secuencia codificada. En esta tabla, los resultados se han agrupado en función de la categoría de movimiento:

Formato	Calidad	HM&HC	MM&MC	LM&LC	COM
QCIF	L	0.54	1.29	1.82	1.19
	M	0.52	1.31	2.01	1.52
	H	0.53	1.40	2.32	1.81
CIF	L	0.64	1.18	1.63	1.00
	M	0.53	1.08	1.66	1.37
	H	0.51	1.08	1.63	1.17
SDTV	L	0.80	1.20	1.77	0.85
	M	0.63	1.07	1.68	1.36
	H	0.61	1.05	1.78	1.19

Tabla 16: Ganancia en PSNR local (dB) para las distintas categorías de vídeos en función del movimiento

La ganancia en PSNR local es mayor cuando el movimiento en escena es menor (LM&LC), lo que proporciona un aumento de calidad subjetiva todavía mayor. En esta categoría la ganancia mínima es de 1.6 dB, mientras que el máximo es de 2.3 dB. En este caso, el sistema visual humano es más sensible a los detalles de la imagen que en escenas de mucho movimiento, por lo que la ganancia en calidad objetiva proporciona una mejora subjetiva sustancial.

La ganancia media para los vídeos con poco movimiento es de 1.91 dB, mientras que los vídeos con movimiento moderado alcanzan una ganancia de

1.21 dB. Finalmente, para los vídeos con mucho movimiento se consigue una ganancia de 0.54 dB, alcanzando una mejora de 1.39 dB para los vídeos de la categoría COM.

Por último, para finalizar con el análisis de la Tabla 16, cabe constatar que, para cada categoría de movimiento y formato de vídeo, la ganancia en PSNR local es mayor para las configuraciones de alta calidad (H), lo que proporciona una mejora mayor en la calidad visual aportada por el esquema de codificación.

La siguiente parte del análisis de los resultados en ganancia en PSNR se refiere a la comparación del algoritmo desarrollado con un esquema de inserción periódica de *keyframes*, con un GOP fijo de 13 imágenes (una imagen I seguida de 12 imágenes P). En este caso no es posible realizar una comparación basada en la ganancia en PSNR local, ya que la posición donde se insertan los *keyframes* en ambos esquemas es totalmente diferente. Por este motivo, se ha realizado una comparación de la PSNR global de las secuencias codificadas:

Formato	Calidad	HM&HC	MM&MC	LM&LC	COM
QCIF	L	0.26	0.89	1.60	0.95
	M	0.28	0.86	1.43	1.01
	H	0.26	0.74	1.16	0.90
CIF	L	0.26	0.92	1.47	0.90
	M	0.29	0.85	1.32	1.01
	H	0.28	0.72	1.11	0.81
SDTV	L	0.29	1.03	1.57	0.78
	M	0.31	0.90	1.33	0.76
	H	0.29	0.66	1.00	0.55

Tabla 17: Ganancia en PSNR global (dB) para las distintas categorías de vídeos en función del movimiento con respecto a un codificador con GOP fijo (IPPPPPPPPPPPP)

En la Tabla 17 se puede apreciar que la ganancia es mayor cuanto menor es el grado de movimiento. Nuevamente, este hecho proporciona una mayor mejora subjetiva a la hora de visualizar el resultado de la codificación, ya que el sistema visual humano es más sensible al detalle espacial en escenas estáticas que en escenas con mucho movimiento. Por otra parte, para un determinado formato, cuanto menor es la calidad (L), mayor es la ganancia obtenida con respecto al esquema de inserción periódica de *intras*. Así, la ganancia máxima obtenida es de 1.6 dB en la PSNR de la secuencia completa, mientras que la ganancia mínima conseguida es de 0.26 dB, proporcionando siempre una ganancia positiva respecto a la codificación con GOP cerrado.

Algunas investigaciones se han centrado en los últimos años en la inserción de *keyframes* basada en el contenido ([21]), pero dichos algoritmos

proporcionaban unos resultados de ganancia en PSNR global sustancialmente menores, y el preprocesado y análisis del movimiento que involucra aumentan considerablemente su coste computacional. Por lo tanto, para comparar los resultados del presente algoritmo, se ha tenido en cuenta el método más reciente de [22], que se basa en el análisis del valor cuadrático medio de los módulos de los vectores de movimiento de cada macrobloque. Los autores de dicho método realizaron una comparación de su algoritmo con un esquema de inserción periódica de *keyframes*, como la analizada en los párrafos anteriores.

El algoritmo de [22] obtiene una ganancia media de PSNR de alrededor de 0.1 dB. Obtiene los mejores resultados, con una mejora de 0.8 dB respecto al codificador original con inserción periódica de imágenes de referencia, para clips de *trailers* de películas (mucho movimiento y cambios de plano frecuentes).

	UPV	**[22]**
Ganancia PSNR (Avg.)	0.82 dB	0.1 dB
Ganancia PSNR (Max.)	1.6 dB	0.8 dB

Tabla 18: Comparación de ganancia en PSNR entre UPV y [22]

Como se puede apreciar en la Tabla 17, para resoluciones bajas y medias, la ganancia obtenida por nuestro algoritmo es de entre 0.3 y 1.6 dB. Por su parte, en la Tabla 18 se puede observar una comparación de los resultados presentados por [22] y los obtenidos mediante el método presentado aquí. En esta tabla se comprueba cómo la mejora media es hasta 8 veces mejor en el presente caso,

mientras que la mayor ganancia obtenida es el doble en el método desarrollado en la UPV.

Para poner en contexto esta comparación es necesario comentar que el tipo de GOP utilizado por [22] es distinto al utilizado en el presente método, ya que utiliza imágenes B e introduce imágenes P adaptativamente en la secuencia.

Finalmente, se ha llevado a cabo un análisis de la influencia del tamaño del intervalo fijo de inserción de *keyframes*, observando que cuanto más corto es el intervalo, mayor es la ganancia en PSNR global que el algoritmo de inserción basada en el contenido proporciona. Este comportamiento se puede explicar por la condición de *bitrate* constante, ya que un *keyframe* es más costoso en términos de consumo de bits que una imagen codificada en modo *inter*, necesitando mayor número de bits para conseguir la misma PSNR. De esta forma, cuanto mayor es el número de *keyframes*, menor es la cantidad de bits disponibles para codificar el resto de imágenes de la secuencia, disminuyendo así la calidad objetiva final.

VI.2. Tiempo de procesamiento

Los resultados del algoritmo de inserción de *keyframes* en cuanto a tiempo de procesamiento son muy similares a los obtenidos por el algoritmo de detección de cambios de plano:

	LM&LC	MM&MC	HM&HC	COM
$\frac{t_{P-I}}{t_P}$	1.01	1.04	1.19	1.13

Tabla 19: Relación entre el tiempo de re-codificación de una imagen y el tiempo medio de codificación de una imagen tipo P

Como se puede observar en la Tabla 19, el coste de detectar una imagen como candidata a *keyframe*, abortar su codificación en modo *inter* y recodificarla en modo *intra* (t_{P-I}) es entre un 1% y un 20% mayor que el coste de codificar la correspondiente imagen en modo *inter* (t_P). Sin embargo, el hecho de escoger las mejores referencias y optimizar su codificación y su uso, hacen que la codificación del resto de imágenes de la secuencia se optimice de igual modo, consiguiendo una ganancia global en tiempo de codificación, como se puede apreciar en la Tabla 20.

	LM&LC	MM&MC	HM&HC	COM
$\frac{(T_{ND}-T_D)}{T_{ND}}$	4.09%	2.16%	1.07%	2.14%

Tabla 20: Relación entre el tiempo total de codificación de una secuencia con y sin inserción automática de keyframes

Esta ganancia llega a un máximo del 4% en secuencias con poco movimiento, y se sitúa por encima del 1% en escenas con mucho movimiento.

Por otra parte, el tiempo global de codificación cuando se usa el presente algoritmo es menos de un 10% mayor que el tiempo necesario con un esquema de inserción periódica de *intras* con un GOP de tamaño 13. En este sentido,

cuanto mayor es la frecuencia de inserción de *keyframes*, menor es la carga computacional de un algoritmo de inserción periódica de *intras*, ya que el análisis a realizar en cada imagen *inter* es mucho mayor que el necesario para la codificación de *keyframes*.

Finalmente, cuando se compara el presente algoritmo con el descrito en [22], se observa una ganancia en coste computacional, ya que el algoritmo de la UPV obtiene un incremento de tiempo computacional inferior al 4% en media en el peor caso, mientras que el método de [22] informa de unos incrementos de entre el 5 y el 10% respecto a un esquema de inserción periódica de *intras*.

Nuevamente, comentar que en el caso de [22] el tipo de GOP utilizado emplea tanto imágenes P como B, por lo que la comparación en términos de tiempo de procesamiento debe ser ponderada. En cualquier caso, la comparación absoluta de tiempos se mantiene completamente válida, ya que se refiere en todos los casos a una comparación con la versión del codificador con inserción periódica de imágenes de referencia.

VII. Implementación comercial

Las técnicas descritas en esta tesis han sido incorporadas en un codificador H.264 de tiempo real, resultando en la mejora de la eficiencia y de la calidad de la codificación. En concreto, Telefónica I+D ha integrado este codificador dentro de la solución propietaria del cliente de videotelefonía H.324m para PC denominado *Escritorio Movistar*.

La apariencia de la aplicación se muestra en la Fig. 29, y la inclusión del algoritmo aquí descrito mejora de forma notable la calidad percibida del vídeo

transmitido con la mencionada aplicación, que opera en un canal UMTS muy estrecho de 64kbps.

Fig. 29: Apariencia de la aplicación videocliente de telefonía UMTS Escritorio Movistar de Telefónica

VIII. Conclusiones

En este capítulo de la tesis se ha presentado un algoritmo de inserción de *keyframes* basada en el contenido, que se ha desarrollado como continuación del proyecto iniciado para la detección automática de cambios de plano y escena en vídeo digital.

El objetivo original de ambas partes del proyecto ha sido el mismo: aumentar la eficiencia de codificación de los codificadores de vídeo actuales mediante la optimización de las referencias utilizadas y el correcto posicionamiento de las imágenes *intra* (*keyframes*). De esta forma, se han

aprovechado las características del algoritmo de detección de cambios de plano para detectar las imágenes de la secuencia que suponen una ruptura en la continuidad de la correlación entre imágenes consecutivas, escogiendo dichas imágenes como las posiciones óptimas para la inserción de *keyframes*. La codificación de estas imágenes en modo *intra* supone un ahorro en el tiempo de proceso necesario y en el *bitrate* generado, ya que el uso de las herramientas de predicción para la codificación de imágenes escasamente correladas con sus referencias es muy poco eficiente.

El algoritmo se basa en el uso de dos umbrales, uno fijo y otro dinámico, aplicados sobre el número de macrobloques *intra* utilizados para codificar cada imagen P de la secuencia, lo que ha demostrado ser un buen indicador de la correlación existente entre una imagen sus referencias. Con este método, se identifican diferentes situaciones que suponen una reducción de la correlación entre imágenes: cambios de plano abruptos, oclusiones, movimientos bruscos y situaciones especiales, como transiciones graduales y flashes. La utilización de estas imágenes como *keyframes* permite obtener una ganancia en PSNR, tanto local (alrededor del *keyframe* insertado), como global (en toda la secuencia codificada), proporcionando una mejora de la calidad objetiva y subjetiva.

En este caso, el entorno de pruebas de la primera parte se ha ampliado, para dar cabida a configuraciones de baja, media y alta calidad, incluyendo distintas resoluciones de vídeo, *bitrates* y *frame rates*, y presentando resultados correspondientes a un amplio abanico de aplicaciones.

Así mismo, se han presentado varias mejoras respecto al algoritmo de detección de cambios de plano, destinadas a mejorar el comportamiento del mismo tras cada detección. En este sentido, se ha aplicado un mecanismo de transición exponencial del umbral desde un valor muy restrictivo cerca de la

última detección hacia el valor final del umbral dinámico una vez transcurrido cierto intervalo. Un análisis de las prestaciones del algoritmo ha permitido medir una mejora en la PSNR local de hasta 1.75 dB cuando se aplica el presente algoritmo en comparación con un codificador que no utiliza ninguna imagen *intra* en la secuencia codificada. Del mismo modo, cuando se compara con un esquema de inserción periódica de *keyframes*, el algoritmo aquí descrito proporciona una ganancia global de hasta 1.6 dB. Por otra parte, comparado con otros métodos de reciente publicación, se aprecia un mejor comportamiento, alcanzando mejores cotas de ganancia.

Finalmente, en cuanto al objetivo de mantener las condiciones de tiempo real en el funcionamiento del algoritmo, los resultados han sido nuevamente satisfactorios, consiguiendo incluso ganancias en tiempo de proceso tanto con respecto a una versión del codificador sin inserción de *intras*, como con respecto a esquemas de inserción periódica de *keyframes*.

El funcionamiento del algoritmo se ha optimizado utilizando un conjunto de secuencias de Entrenamiento, y el funcionamiento del algoritmo optimizado se ha probado mediante su aplicación a un conjunto diferente de secuencias de Test. Entre ambos conjuntos se han procesado un total de 70000 imágenes, incluyendo más de 1500 candidatos a la inserción de *keyframes*.

Por lo tanto, el algoritmo de inserción de *keyframes* basada en el contenido desarrollado en este proyecto es rápido y robusto, proporcionando una mejora apreciable tanto en la eficiencia de codificación como en la calidad objetiva y subjetiva del vídeo codificado. Este método ha sido probado exhaustivamente y en el análisis de los resultados se han demostrado unas prestaciones superiores a otros métodos similares de reciente publicación.

Finalmente, la elección del algoritmo desarrollado por parte de Telefónica I+D para su implantación en una de sus aplicaciones comerciales de codificación de vídeo demuestra la validez de los resultados obtenidos, que han sido publicados en la prestigiosa revista *IEEE Transactions on Circuits and Systems for Video Technology*.

IX. Líneas futuras

La mejora de la eficiencia de codificación tiene dos consecuencias complementarias: el aumento de la calidad de vídeo la misma cantidad de información codificada o la disminución de la cantidad de información codificada necesaria para obtener la misma calidad. En este sentido, la medida de la calidad de vídeo es un campo de investigación que se ha desarrollado mucho en los últimos años, presentando dos enfoques distintos: medida de la calidad subjetiva y medida de la calidad objetiva.

Las medidas objetivas tienen en cuenta únicamente los valores de los píxeles de las imágenes codificadas, por lo que no tienen una relación directa con la calidad percibida por el espectador humano. Por su parte, las medidas subjetivas de calidad se realizan mediante costosos métodos estadísticos basados en la presentación de un clip de vídeo a un conjunto de observadores humanos que deben valorar la calidad en una escala dada, realizando posteriormente un promediado de las valoraciones para obtener la medida de calidad subjetiva final. Este método subjetivo es muy costoso, por lo que en los últimos años se ha investigado mucho en el desarrollo de métodos de medida objetiva de la calidad perceptual del vídeo digital, eliminando así la necesidad de disponer de las valoraciones de los observadores para obtener la medida de calidad.

Este es un campo de investigación muy interesante que puede aportar gran cantidad de mejoras a los distintos algoritmos de mejora de la eficiencia de codificación, permitiendo valorar de forma objetiva el aumento de calidad subjetiva obtenido mediante la aplicación de diferentes técnicas como las presentadas en esta tesis [23].

Por otra parte, al igual que en el capítulo anterior, el estudio de la implementación de este algoritmo en un codificador de vídeo H.265 constituye el siguiente paso natural en este desarrollo. De esta forma se adaptan los resultados obtenidos al más novedoso estándar de codificación de vídeo hasta la fecha.

Capítulo 4

Control de Tasa con *Bitrate* y *Frame Rate* Variable

I. Introducción

Distintas formas de comunicación multimedia, como audio y vídeo, han utilizado tradicionalmente redes de comunicaciones dedicadas, generalmente de tipo conmutación de circuitos, para garantizar el ancho de banda y la estabilidad necesarias para proporcionar la calidad de experiencia adecuada. Como ejemplos de estas redes cabe destacar las redes de comunicaciones tradicionales como la red de telefonía fija (*Public Old Telephony System*, POTS), la telefonía móvil 2G o los inicios de la videotelefonía 3G y la más avanzada tecnología móvil 4G (LTE).

Por otra parte, el auge de nuevas redes de comunicaciones basadas en la tecnología IP ha permitido desarrollar canales de comunicación especializados para este tipo de aplicaciones. Sin embargo, la tecnología IP no está basada en conmutación de circuitos, por lo que no proporciona unos mínimos de calidad de servicio por sí misma. Por este motivo, no se pueden garantizar unos niveles mínimos de ancho de banda, estabilidad de la transmisión, etc. Todos estos factores hacen que la transmisión de audio y vídeo sea mucho más complicada.

En lo que respecta a redes fijas de comunicaciones, existen distintas aproximaciones para solventar o aliviar estas limitaciones, pero en el caso de comunicaciones móviles IP estos problemas no sólo son mayores (menor ancho de banda, mayor retardo, etc.), sino que las condiciones del canal pueden variar de forma dramática a lo largo de la comunicación debido a diferentes factores (compartición de recursos con otros usuarios, desvanecimientos, movimiento de los usuarios, interferencias, etc.).

El desarrollo de esta tesis se ha realizado en colaboración con Telefónica I+D, que ha trabajado en el desarrollo de aplicaciones de usuario que permitan una comunicación de vídeo a través de redes P2P sobre redes 3G. En estos

trabajos, la sesión P2P se puede aproximar por un canal dual de tipo cliente/servidor, por lo que cada sentido de la comunicación se puede analizar independientemente.

En la Fig. 30 se muestra un diagrama de bloques que describe la aproximación realizada al comportamiento del canal y sus parámetros fundamentales.

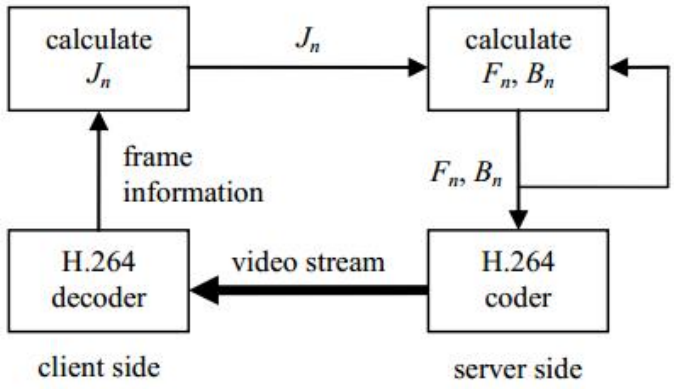


Fig. 30: Aproximación del canal de transmisión

En este escenario de trabajo, el vídeo se codifica utilizando el estándar H.264, que es apropiado para las condiciones de bajo ancho de banda propias de estas redes de comunicaciones móviles.

Así, el servidor se encarga de enviar la información de vídeo codificada a través del canal, por lo que debe adaptar la cantidad de información enviada a las condiciones puntuales y cambiantes de dicho canal sin un conocimiento real de las mencionadas condiciones. La única información disponible al respecto proviene de la estimación del canal realizada por el cliente, según la

recomendación del protocolo RTCP [36]. En este sentido, existen dos métodos principales para adaptar la cantidad de información generada por el codificador:

- Adaptar la tasa de imágenes por segundo, variando la distancia entre imágenes consecutivas.
- Adaptar el *bitrate*, variando la calidad de las imágenes codificadas para una determinada tasa de imágenes por segundo.

Si asumimos que la adaptación del *bitrate* y el *frame rate* se producen en intervalos regulares, podemos describir la situación según el siguiente modelo:

$$F_{n+1} = f(F_n, B_n, J_n)$$

$$B_{n+1} = g(F_n, B_n, J_n)$$

Ec. 20: Tasa de imágenes por segundo y *bitrate* para la imagen n+1

En estas ecuaciones, F_n es la tasa de imágenes por segundo objetivo en el instante *n*, B_n es el *bitrate* objetivo en el instante *n* y J_n es la distorsión del canal en el instante *n* (medida por el cliente).

En Telefónica I+D se han desarrollado modelos para estimar la relación entre f y g, por lo que el paso siguiente consiste en obtener el algoritmo necesario para adaptar la codificación de vídeo al *bitrate* y a la tasa de imágenes por segundo demandadas por este modelo.

Por lo tanto, cuando las condiciones del canal cambian (J_n), el codificador modifica el *bitrate* de salida, excepto cuando el *bitrate* alcanza un nivel mínimo, en cuyo caso es necesario reducir el número de imágenes por segundo para ser capaces de afrontar nuevas disminuciones de la capacidad del canal.

Modificar la tasa de imágenes por segundo es una tarea sencilla, ya que únicamente es necesario adaptar la temporización de los instantes de captura. Sin embargo, la modificación del *bitrate* es más complicada, ya que los parámetros básicos de cuantificación sólo se relacionan indirectamente con la cantidad de bits finalmente generada por la codificación. Además, la modificación del número de imágenes por segundo también afecta a la estrategia de codificación del codificador, como se ha visto en las secciones anteriores de esta tesis.

II. Control de tasa en codificación de vídeo

II.1. Conceptos básicos

Un algoritmo de control de tasa en un codificador de vídeo ajusta dinámicamente ciertos parámetros del codificador para conseguir una tasa de bits objetivo. Básicamente se encarga de adjudicar un presupuesto de bits para la codificación de un conjunto de imágenes, y refina el presupuesto asignando distintas cantidades a cada imagen o subpartición de imagen en función de la proporción del presupuesto restante.

El control de tasa no forma parte del estándar H.264, pero el grupo que ha desarrollado el estándar ha determinado también una guía no normativa para ayudar a su implementación.

En general, los codificadores híbridos son sistemas inherentemente con pérdidas, que consiguen la compresión eliminando información realmente redundante, pero también haciendo pequeños compromisos de calidad que se intenta que sean lo menos perceptibles posible.

El algoritmo de control de tasa del codificador puede adaptar distintos parámetros para obtener la tasa de bits objetivo, entre los que destacan el tamaño de la imagen, el tipo de imagen (I, P ó B), la tasa de imágenes por segundo o, sobre todo, el parámetro de cuantificación QP.

En particular, el parámetro de cuantificación QP regula la cantidad de detalle que se preserva en la codificación. Con QP bajos todo el detalle se mantiene, pero al aumentar se empiezan a perder detalles, lo que proporciona una ganancia en la tasa de bit generada por el codificador (compresión). Además de esta disminución del *bitrate* también se produce un aumento de la distorsión y una pérdida de calidad.

En general, todo algoritmo de control de tasa se divide en dos procesos básicos:

- **Asignación de bits**: consiste en asignar un presupuesto de bits a un segmento de la secuencia a codificar. Como se verá más adelante, este segmento puede ser de varias imágenes o de un pequeño fragmento de una imagen. En un control de tasa escalable, esta asignación se hace a distintos niveles de forma coordinada.
- **Obtención del valor del cuantificador**: en esta etapa se calcula el valor de QP para codificar el segmento definido anteriormente. Normalmente se utiliza un modelo tasa-distorsión para tener en cuenta la complejidad del segmento. Este modelo se puede obtener de dos formas:
    - *Analítica*: la función tasa-distorsión se deriva del modelado de la distribución de los coeficientes transformados y cuantificados. Cuando un segmento de vídeo se ha

codificado se actualizan los parámetros del modelo para el siguiente proceso de estimación de QP.

- *Empírica*: la función de tasa-distorsión se obtiene interpolando un conjunto de puntos tasa/distorsión procedentes de segmentos de vídeo anteriores o siguientes.

II.2. Tasa de bit Variable y Tasa de bit Constante

En un codificador de tasa de bit variable (VBR) el usuario puede seleccionar la calidad de la compresión fijando el valor de QP. Esto produce una salida de calidad constante pero un flujo de bits muy variable, ya que las imágenes de mayor complejidad requerirán de una gran cantidad de bits para su codificación, mientras que en imágenes más sencillas (por ejemplo, con poca textura o en secuencias con poco movimiento) la cantidad de bits generados será menor.

En realidad, tanto el ancho de banda de la red de transmisión como el tamaño del buffer del decodificador imponen una serie de restricciones a la codificación del vídeo y a la cantidad de bits por segundo que puede procesar el sistema. Por lo tanto, es necesario variar dinámicamente el valor de QP basándose en estimaciones de la complejidad de las imágenes a codificar para que cada imagen utilice una cantidad de bits lo más constante posible durante todo el proceso de codificación. En este caso, el usuario, en vez de especificar el valor del QP a utilizar, especifica el *bitrate* objetivo.

En estos casos se asume que tanto la tasa de bit objetivo como la tasa de imágenes por segundo son constantes (CBR y CFR). Sin embargo, otro tipo de aplicaciones requieren que la tasa de imágenes por segundo cambie durante la secuencia. Esto es especialmente útil para crear ciertos vídeos como

presentaciones o contenidos con mucho detalle en muchas imágenes estáticas, de forma que, además, se consigue una mayor compresión de la secuencia. También se pueden aplicar estos conceptos a secuencias donde hay una variabilidad en la tasa de imágenes por segundo, combinando material en 24/25/30/50/60 imágenes por segundo y evitando artefactos producidos por la conversión de FPS.

II.3. Requisitos y restricciones

Uno de los requisitos más importantes a tener en cuenta en el desarrollo de un algoritmo de control de tasa es el compromiso tasa-distorsión producido en todo sistema con pérdidas de este tipo.

Como se verá más adelante, la distorsión es una función decreciente del *bitrate*, que a su vez depende inversamente del valor de QP. En resumen, la función del control de tasa consiste en obtener la mayor calidad posible (la mínima distorsión) dadas una serie de restricciones en el *bitrate* generado.

Finalmente, otro de los requisitos de este tipo de sistemas es la baja complejidad, que depende básicamente de la aplicación objetivo de la codificación. En el caso que nos ocupa, la codificación en tiempo real requiere de una complejidad computacional mínima en todos los puntos de la cadena de codificación. También requiere que el algoritmo realice su función en una única pasada y que el algoritmo de optimización de tasa-distorsión sea lo más sencillo posible.

Una solución óptima de este tipo de problemas consiste en obtener el valor del cuantificador que proporciona la mínima distorsión manteniendo el *bitrate*

bajo unos ciertos límites. Diferentes estrategias permiten obtener estos resultados:

- Minimizar la distorsión media (MINAVE)
- Minimizar la distorsión máxima (MINMAX)
- Minimizar la variación de la distorsión (MINVAR)

Además, la codificación híbrida impone otras restricciones, especialmente por la introducción de la variabilidad temporal (redundancia entre imágenes). Por lo tanto, la tasa y la distorsión ya no sólo dependen del parámetro de cuantificación, sino también del QP utilizado en imágenes anteriores.

Finalmente, un nuevo conjunto de restricciones proviene de la aplicación a la que se orienta la codificación, como pueden ser los requisitos del buffer del decodificador o del canal de transmisión.

Para resolver estos problemas condicionados se han propuesto dos aproximaciones principales:

- **Métodos de relajación de Lagrange**: convierte un problema condicionado en uno sin condicionar mediante un conjunto de parámetros (multiplicadores de Lagrange).
- **Programación dinámica**: intenta resolver un problema multi-variable como múltiples problemas de una única variable.

Sin embargo, el principal problema de estas soluciones óptimas es su elevada complejidad computacional, que los hace inviables para las aplicaciones a las que se dirigen.

Para resolver este conflicto se utilizan los algoritmos de control de tasa basados en modelos que, pese a no obtener la solución óptima, proporcionan

resultados competitivos en términos de tasa-distorsión con una complejidad manejable.

Estos modelos tienen como entradas los valores objetivos (tasa de bit y tasa de imágenes por segundo) y el propio vídeo sin comprimir, además de otros valores realimentados de la salida del proceso, como el número de bits generados por las imágenes anteriores, información de la red, estado del buffer, etc. Como salida, el modelo genera el valor de QP a utilizar para codificar la siguiente imagen (o conjunto de imágenes, o fragmento de imagen).

Además de esta información, también se puede utilizar información procedente de un pre-procesado del vídeo a codificar, como puede ser un detector de cambios de plano como los descritos en el segundo capítulo de esta tesis [40]. Estos detectores permiten seleccionar el método de codificación de las imágenes que proporcionen la mejor relación tasa-distorsión (en este caso, codificar en modo I las imágenes de los cambios de plano).

Otras formas de pre-procesado permiten asignar mayor número de bits a imágenes o zonas más complejas, o tomar en consideración las características del sistema visual humano a la hora de distribuir la distorsión a lo largo de la secuencia codificada [41].

II.4. Tasa-Distorsión

Como se ha visto en secciones anteriores, uno de los elementos claves relacionados con el control de tasa es la relación existente entre la tasa de bit generada y la distorsión introducida en la secuencia codificada dada la complejidad de la secuencia a codificar.

Esta relación se empezó a estudiar en los años 70 [42], y desde entonces se usa para obtener un compromiso entre la distorsión y la tasa de bit generada por un codificador de vídeo. En la teoría de R-D se formula una función para proporcionar un límite inferior a la tasa de bit dado un nivel de distorsión.

Esta teoría se utiliza para asignar recursos de codificación a un determinado segmento de secuencia de vídeo, utilizando un número finito de modos de codificación y otros parámetros. Por lo tanto, estamos ante un problema que define un conjunto finito de pares tasa-distorsión. Sin embargo, es prácticamente imposible encontrar unas fórmulas cerradas para estas relaciones, por lo que se utilizan funciones operacionales (ORD, *Operational Rate-Distortion*). En estas funciones se escoge el límite inferior de la curva definida por todos los puntos posibles dadas todas las combinaciones de parámetros. Sin embargo, pese a obtener potencialmente la mejor solución posible, estos métodos requieren de la evaluación de todos los posibles puntos del espacio de soluciones, lo que los hace computacionalmente inviables.

En su lugar, se utilizan los métodos basados en modelos, que se encargan de aproximar una curva lo más similar posible a la función ORD anterior. Como se ha comentado anteriormente, estos modelos pueden ser analíticos o empíricos.

En los modelos analíticos se asume que una estimación precisa de la distribución de los coeficientes de la DCT de una imagen producirá una estimación precisa de la tasa y la distorsión para un valor dado del parámetro de cuantificación.

En [43] se propone un modelo sencillo para la entropía de la señal cuantificada que separa la estimación en dos bandas, para tasas de bit altas y bajas. A partir de este modelo se puede definir un modelo de la distorsión

utilizando el paso de cuantificación y el número de macrobloques de una imagen:

$$D(Q) = \frac{1}{N}\sum N\frac{Q_i^2}{12}$$

Ec. 21: Modelo de distorsión en función del paso de cuantificación

Generalizando el modelo, se puede llegar a ecuaciones del siguiente tipo para el modelo de tasa:

$$R(Q) = M(aQ^{-1} + bQ^{-2}) + H$$

Ec. 22: Modelo de tasa

En esta ecuación, M es la medida de la complejidad de la imagen basada en el MAD (media de las diferencias absolutas), y H es la sobrecarga de información de las cabeceras y los vectores de movimiento.

$$MAD = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}|x_a(i,j) - x_b(i,j)|$$

Ec. 23: Media de las Diferencias Absolutas (MAD)

En la ecuación anterior se muestra la expresión correspondiente al MAD entre dos imágenes (a y b) de MxN píxeles cada una.

III. Control de tasa en H.264

III.1. Conceptos básicos

El control de tasa utilizado por el codificador H.264 de partida en este trabajo se basa en el implementado por el codificador del JVT (*Joint Video Team*), descrito en [1][2], aplicándose a tres niveles distintos: GOP, imagen y unidad básica.

En la Fig. 31 se puede observar un esquema de los elementos que forman parte del control de tasa diseñado para el codificador H.264. Como se puede ver, este control de tasa sigue el paradigma de los algoritmos basados en modelos (modelo de buffer, estimación de la complejidad o modelo tasa-distorsión).

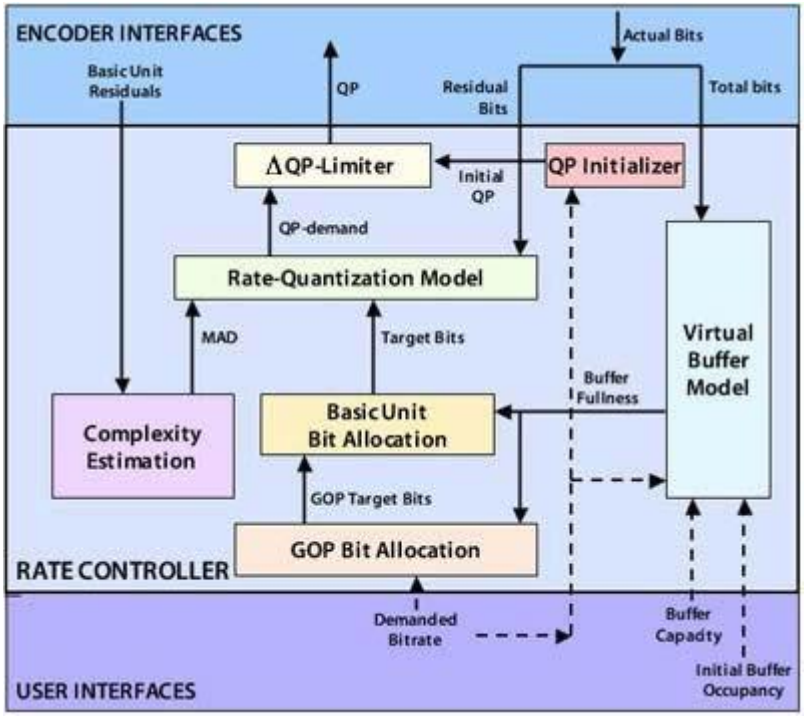


Fig. 31: Elementos del control de tasa de H.264

A continuación se enumeran los principales elementos de este modelo [38]:

- **Modelo tasa-distorsión**: es el corazón del sistema de control de tasa y se encarga de relacionar el valor de QP con los bits generados y la complejidad de la codificación de cada imagen.
- **Estimación de la complejidad**: la complejidad de cada imagen se mide a partir del MAD del error de predicción. Cuanto peor es la predicción, más compleja es la imagen a codificar. Por lo tanto, en el caso de la predicción temporal también proporciona una medida de la similitud entre imágenes adyacentes. Idealmente, el MAD se

debería estimar/calcular tras codificar la imagen actual. Sin embargo, esto requeriría una codificación en dos pasadas, donde en la primera se escogería el QP y en la segunda se realizaría la codificación propiamente dicha. Para evitar este problema, y basándose en la similitud entre imágenes consecutivas, se predice el MAD actual a partir de la codificación de las imágenes anteriores del codificador.

- **Limitación de la variación de QP**: para evitar fluctuaciones demasiado bruscas de la calidad debido a variaciones grandes de QP, la modificación de QP se limita a ±2 entre imágenes consecutivas.
- **Modelo de buffer virtual**: todos los decodificadores que cumplen con el estándar están dotados de un buffer para suavizar las variaciones en la tasa de bit y en el tiempo de llegada de los datos codificados. Por lo tanto, el codificador debe disponer de un buffer virtual para simular el estado del buffer del decodificador para ser capaz de generar una tasa de bits que no sature al decodificador. El buffer virtual se llena con la cantidad de bits generados por el codificador al codificar cada imagen y se vacía a una tasa constante con la salida de la información codificada hacia el decodificador (en entornos con tasa de bits y tasa de imágenes por segundo constantes).
- **Inicialización de QP**: el QP de la primera imagen se debe especificar explícitamente. Para ello se puede fijar un valor o calcularlo a partir del número de bits por píxel, según se explica en [37].

- **Asignación de bits al GOP**: como se verá más adelante, el control de tasa trabaja a tres niveles, el primero de los cuales consiste en asignar un presupuesto de bits para la codificación de un GOP completo. Este nivel también determina el valor de QP para la imagen *intra* del GOP.
- **Asignación de bits a la unidad básica**: la terminología de unidad básica fue introducida por [14]. Partiendo de esta base se define un control de tasa escalable con distintos niveles de granularidad, como se verá más adelante. En la Fig. 31 se asume que la unidad básica coincide con la imagen completa, pero este concepto se puede extender a un conjunto de macrobloques o incluso a un macrobloque individual.

En las secciones siguientes se presenta una descripción más detallada de los distintos niveles que entran en juego en el control de tasa escalable y la relación entre el control de tasa y el algoritmo de optimización de la tasa y la distorsión.

III.2. Control de tasa escalable

En este apartado se presentan los tres niveles a los que se aplica el algoritmo de control de tasa aquí descrito: GOP, imagen y unidad básica.

III.2.1. Control de tasa a nivel de GOP

Un GOP es un conjunto de imágenes consecutivas codificadas por el codificador, de forma que la primera imagen es una IDR (*intra*), mientras que el

resto de imágenes del GOP se codifican mediante técnicas de predicción compensada en movimiento (P ó B).

El control de tasa a nivel de GOP se encarga de calcular el número de bits asignados para la codificación del GOP completo, de forma que dicho presupuesto se asigne en las etapas siguientes a la codificación de cada imagen concreta. Para este cálculo se tiene en cuenta el número de bits sobrantes procedentes de la codificación del GOP anterior, el *bitrate* demandado, y la tasa de imágenes por segundo deseada. Como se verá más adelante, es en este apartado donde se han introducido las modificaciones relacionadas con la utilización de *bitrates* y *frame rates* variables.

Esta parte del proceso también es la encargada de calcular el valor de QP para la primera imagen del GOP (la imagen I) y para la primera imagen P del mismo.

Finalmente, cabe comentar que el tamaño del GOP puede ser tanto fijo como variable, en función del uso deseado del codificador y del posible uso de un algoritmo de detección de cambios de plano, influyendo dicha decisión en el comportamiento del algoritmo de control de tasa.

III.2.2. Control de tasa a nivel de imagen

El control de tasa a nivel de *frame* es el encargado de asignar un número de bits determinado a cada imagen del GOP para cumplir con las restricciones impuestas para el GOP completo, así como para mantener el nivel del buffer bajo control. El objetivo final de esta etapa es la selección del valor del cuantificador a utilizar en la imagen para cumplir con los requisitos anteriores. A su vez, esta etapa se divide en tres partes: una etapa previa a la codificación

en la que se calcula el número de bits objetivo para la codificación de la imagen; una segunda etapa en la que se escoge el valor más adecuado del cuantificador para cumplir con el número de bits objetivo calculado; y una etapa final posterior a la codificación en la cual se actualizan los valores de los parámetros del algoritmo (parámetros de regresión lineal, ocupación del buffer, número de *frames* descartados, etc.).

En esta etapa se tiene en cuenta tanto el estado actual del buffer como el número de bits que quedan para la codificación del resto de imágenes del GOP actual.

III.2.3. Control de tasa a nivel de unidad básica

Por último, el control de tasa a nivel de unidad básica permite refinar el control de tasa, variando el valor del cuantificador dentro de una imagen. En concreto, una unidad básica es un conjunto consecutivo de macrobloques codificados con el mismo valor del cuantificador, de forma que en función del número de macrobloques que constituyan la unidad básica, el control de tasa se realizará de forma más fina o más gruesa.

En este sentido, se definen tres estrategias distintas:

- **Control de tasa a nivel de frame**: si el número de macrobloques de la unidad básica es el total de macrobloques de la imagen, el control de tasa se hace únicamente a nivel de imagen, teniendo menor precisión a la hora de utilizar los bits asignados a cada imagen.
- **Control de tasa a nivel de macrobloque**: si cada unidad básica está formada por un único macrobloque, se puede modificar

individualmente el cuantificador de cada macrobloque, obteniendo mayor precisión en el control de tasa.

- **Control de tasa a nivel de línea de macrobloques**: como situación intermedia, se aconseja el uso de unidades básicas formadas por una línea completa de macrobloques de la imagen, llegando a un término medio entre las dos estrategias anteriores.

III.3. Rate-Distortion Optimization

La optimización de la tasa de bits y la distorsión es un problema largamente estudiado [39] y, pese a estar íntimamente relacionado con la cantidad de bits generados por la codificación de cada imagen, no está directamente ligado al algoritmo de control de tasa.

Por lo tanto, más que formar parte del control de tasa, el algoritmo de RDO es complementario al mismo. Los dos procesos se encuentran desacoplados, ya que el hecho de acoplarlos para obtener una solución conjunta supondría un costoso proceso iterativo.

H.264 proporciona 7 modos de predicción temporal, 9 modos de predicción espacial para bloques de 4x4 píxeles y 4 modos de predicción espacial para macrobloques de 16x16 píxeles, además del modo *Skip*. Cada macrobloque puede ser dividido y codificado de varias formas, por lo que la selección del mejor modo de codificación para cada macrobloque es crítica en el proceso de codificación global. Además, pese a ser uno de los elementos que más carga computacional supone en el proceso, es también uno de los que más ganancia de compresión proporciona (ya que trata de conseguir las mejores predicciones posibles para cada macrobloque).

Precisamente, la selección del mejor modo de predicción para cada macrobloque es uno de los objetivos del algoritmo de *Rate-Distortion Optimization* (RDO). Este proceso consiste en:

1. Cálculo exhaustivo de todos los posibles modos para determinar cuántos bits genera cada uno y cuánta distorsión produce.
2. Evaluación de una métrica que considere conjuntamente la tasa de bit y la distorsión.
3. Selección del método que minimice la métrica.

En este sentido, hay que tener en cuenta que el algoritmo de RDO no modifica el valor de QP, sino que lo recibe como entrada.

Así, la relación entre el control de tasa y el algoritmo de RDO se suele conocer como un problema de tipo "el huevo y la gallina": el objetivo del control de tasa es minimizar la distorsión con la cantidad de bits disponibles, y para conseguir este objetivo trata de seleccionar el valor óptimo de QP. Pero la distorsión generada sólo está disponible después de que el algoritmo de RDO ha utilizado el valor de QP para obtener el modo de codificación óptimo.

Por lo tanto, la relación con el algoritmo de RDO condiciona el diseño del algoritmo de control de tasa, que debe usar una estimación de la distorsión basada en la complejidad de las imágenes anteriores de la secuencia.

En este sentido, el control de tasa en H.264 es más complejo que en los estándares anteriores debido a que en los anteriores las estadísticas de codificación de la imagen actual están disponibles para el control de tasa, mientras que en H.264 esto no sucede. El motivo de este hecho es que en H.264 el parámetro de cuantificación QP está involucrado tanto en el control de tasa como en el algoritmo de RDO.

En general, el objetivo final del algoritmo de control del codificador es conseguir un conjunto de parámetros, y por lo tanto un flujo de bits, de forma que se consiga un compromiso entre la tasa de bits generada y la distorsión introducida en la secuencia codificada [14].

En todos los estándares actuales se utilizan las técnicas de asignación de bits Lagrangianas, debido a su simplicidad y efectividad [39]. En concreto, se trata de minimizar la siguiente ecuación:

$$J(K, M|Q) = D_{REC}(K, M|Q) + \lambda_{MODE} R(K, M|Q)$$

Ec. 24: Ecuación de Lagrange a minimizar

En esta ecuación, K es un macrobloque, M un modo de codificación para el macrobloque y D y R son los valores de distorsión y tasa del macrobloque reconstruido. λ_{MODE} es el parámetro de Lagrange, que depende del valor del cuantificador:

$$\lambda_{MODE} = 0.85 \times 2^{(Q-12)/3}$$

Ec. 25: Parámetro de Lagrange para la selección de modo de codificación

Aquí se puede ver de nuevo el dilema del huevo y la gallina, ya que para calcular el MAD se necesita el valor de Q, pero a su vez éste valor es necesario para calcular el MAD según la Ec. 23.

Para solucionar este problema se recurre a la estimación del MAD mediante la regresión lineal, de la siguiente forma:

$$\widehat{MAD} = a_1 MAD_{PREV} + a_2$$

Ec. 26: Estimación del MAD

En esta ecuación, a_1 y a_2 son los parámetros del modelo de regresión a partir del MAD de la imagen previa.

IV. Control de tasa H.264 con *bitrate* y *frame rate* variable

Las modificaciones introducidas al algoritmo anterior destinadas a permitir el uso de un valor variable dinámicamente tanto para el *bitrate* como para el *frame rate* tienen relación exclusivamente con la etapa de control de tasa a nivel de GOP.

Como se ha comentado anteriormente, el número de bits asignado para la codificación de cada GOP depende, entre otras cosas, del *bitrate* y del *frame rate* utilizados, por lo que su variación afectará necesariamente al cálculo realizado, que tendrá que actualizarse cada vez que se produzca una modificación de cualquiera de los dos parámetros.

Esta modificación actualiza el número de bits restantes para la codificación del resto de imágenes del GOP con la diferencia de *bits/frame* generada por las variaciones introducidas tanto en el *bitrate* como en la tasa de imágenes por segundo:

$$\Delta Bits = \frac{B_i - B_{i-1}}{F_i} \times (N - N_C)$$

Ec. 27: Variación en el número de bits restantes al cambiar el *bitrate* entre las imágenes i e i+1

$$\Delta Bits = B_i \times \frac{F_i - F_{i-1}}{F_i \times F_{i-1}} \times (N - N_C)$$

Ec. 28: Variación en el número de bits restantes al cambiar el número de imágenes por segundo entre las imágenes i e i+1

La primera ecuación describe el factor a añadir al número de bits disponibles para la codificación del resto de imágenes del GOP cuando el *bitrate* pasa de B_{i-1} a B_i, mientras que la segunda ecuación indica el factor de variación cuando la tasa de imágenes por segundo pasa de F_{i-1} a F_i. En dichas ecuaciones, *N* es el número de imágenes que forman el GOP, mientras que N_C es el número de imágenes del GOP que ya se han codificado.

Como se puede apreciar en las ecuaciones anteriores, al disminuir la tasa de imágenes por segundo aumenta el número de bits disponibles para codificar las siguientes imágenes, mientras que ocurre lo contrario cuando aumenta el número de imágenes por segundo a codificar.

V. Resultados

En esta sección se presentan algunos resultados obtenidos al aplicar el algoritmo de control de tasa desarrollado en distintas situaciones.

En las gráficas siguientes se presenta, en azul y línea continua, el número de bits por imagen según el cociente B/F, mientras que en verde y línea discontinua aparece el número de bits consumidos para la codificación de cada imagen al utilizar el algoritmo presentado. Finalmente, en rojo se muestra el número de bits por *frame* medio obtenido al aplicar una ventana de 5 imágenes alrededor de cada frame considerado.

En concreto, en la Fig. 32 se observa el resultado de la codificación de una secuencia en formato QCIF, con el *bitrate* variando entre 40 kbps y 200 kbps y el *frame rate* variando entre 3 y 25 imágenes por segundo. Estas tasas de variación dan como resultado tamaños objetivo de imagen que van desde 1600 bits hasta 66000 bits. En particular, en la Fig. 32 el rango dinámico va de aproximadamente 2000 bits por imagen a unos 5000, y, como se puede comprobar, el comportamiento del control de tasa es muy satisfactorio, ya que es capaz de seguir las variaciones demandadas con suficiente precisión y velocidad, adaptándose a las transiciones introducidas.

Este ejemplo se ha obtenido al aplicar un control de tasa a nivel de macrobloque, de forma que al utilizar el algoritmo a nivel de imagen o de línea de macrobloques el comportamiento es más oscilatorio, pero mantiene la media siempre alrededor del valor demandado, como se observa en las siguientes figuras (Fig. 33 y Fig. 34).

Pruebas similares con formatos CIF y SDTV presentan resultados y comportamientos similares, obteniendo un *bitrate* final que se adapta correctamente a la tasa demandada.

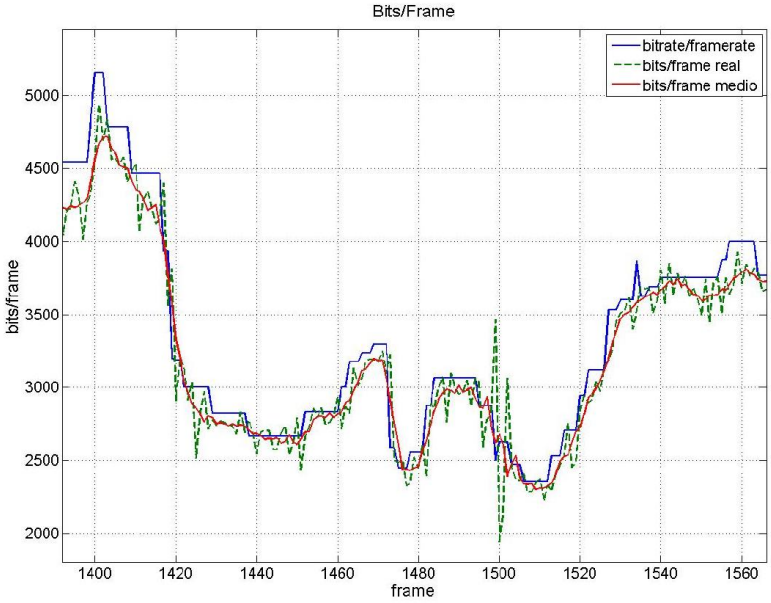


Fig. 32: Comportamiento del control de tasa al variar el *bitrate* y el *frame rate* con control de tasa a nivel de macrobloque. *Bitrate* entre 40 kbps y 200 kbps y tasa de imágenes por segundo entre 3 fps y 25 fps

En la Fig. 32 se ha variado el *bitrate* y la tasa de imágenes por segundo en aproximadamente el 10% de las imágenes de la secuencia. Por su parte, en la Fig. 33 se observa el resultado obtenido al variar el *frame rate* y el *bitrate* en todas las imágenes codificadas. Como se puede apreciar, el comportamiento es nuevamente satisfactorio, de forma que el *bitrate* medio por imagen generado (línea roja) se adapta perfectamente al número de bits por frame deseado (línea azul).

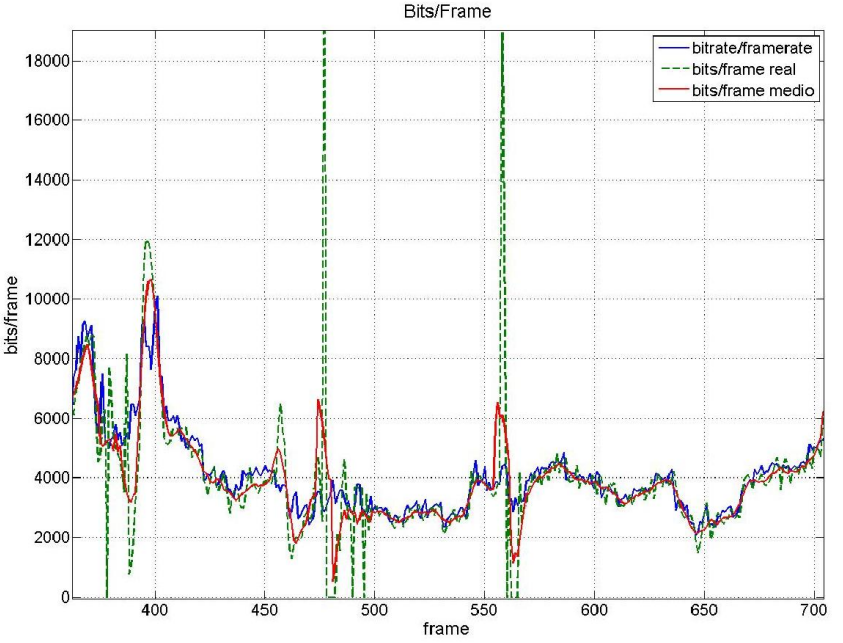


Fig. 33: Comportamiento del control de tasa al variar el *bitrate* y el *frame rate* en cada imagen con control de tasa a nivel de macrobloque

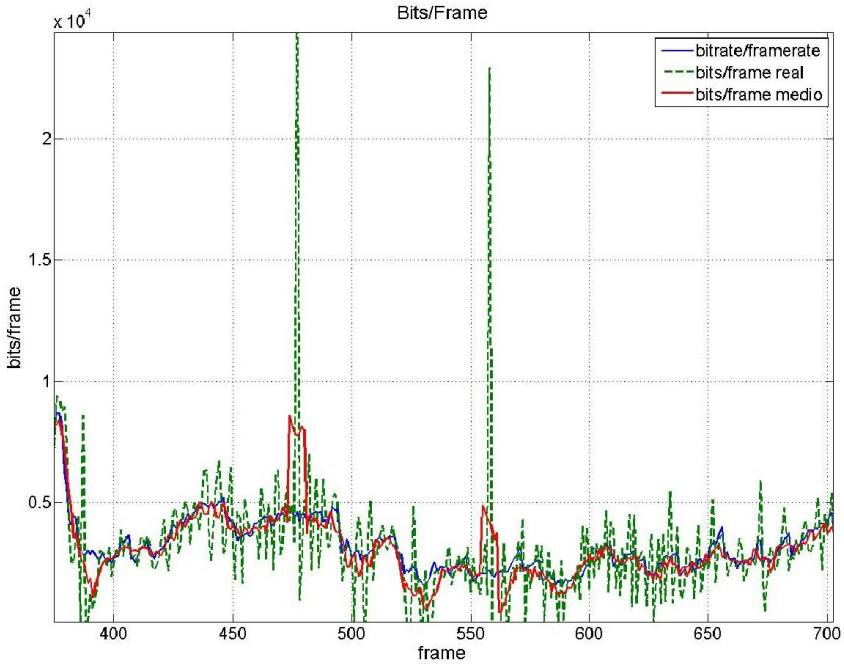


Fig. 34: Comportamiento del control de tasa a nivel de línea de macrobloques

En la Fig. 34 se puede observar el comportamiento del algoritmo de control de tasa cuando se aplica un esquema de control al nivel de cada línea de macrobloques. En este caso, la adaptación es menos precisa que al adaptar el valor del cuantificador cada macrobloque, por lo que se observan más oscilaciones en el *bitrate* puntual generado (línea verde discontinua). Sin embargo, el *bitrate* medio medido en una ventana de 5 *frames* (línea roja) se adapta nuevamente al *bitrate* objetivo.

Estas pruebas se han repetido para distintos valores de tasa de imágenes por segundo, *bitrate*, formato, etc. Del mismo modo, se han utilizado formatos de GOP abierto y cerrado, utilizado codificación CABAC y CAVLC, y se han

probado distintos valores de descarte de imágenes cuando se sobrepasa el tamaño del *buffer*, modificando la correlación existente entre imágenes codificadas consecutivamente.

Por último, y al igual que en el caso del control de tasa original, se ha comprobado su funcionamiento ante GOPs de tipo IBBP, intercalando dos imágenes B entre cada par de I o P. En estos casos, el control de tasa asigna un cuantificador fijo a las imágenes B, obteniendo una distribución de los bits más oscilante, asignando más bits a las imágenes P y menos a las Bs que se codifican más eficientemente.

VI. Conclusiones

En este capítulo, se ha desarrollado un algoritmo de control de tasa de bit para codificadores de vídeo basados en el modelo JVT, modificando el algoritmo de control de tasa original de H.264 para incluir la posibilidad de utilizar tasas de imágenes por segundo variables y *bitrates* variables, de forma que se ha conseguido un algoritmo robusto y apto para su utilización en canales de condiciones inestables y conocidas.

El algoritmo original se aplica a tres niveles distintos: GOP, imagen y unidad básica, trabajando siempre con valores fijos del número de imágenes por segundo.

Las modificaciones introducidas para adaptar este algoritmo a un entorno con tasa de imágenes por segundo variable se han centrado en la asignación del número de bits al GOP. Así, el presupuesto asignado al grupo de imágenes se modifica cuando varía el *frame rate* objetivo (aumenta al disminuir el número de imágenes por segundo y viceversa).

Las pruebas realizadas, con *bitrates* entre 40 kbps y 200 kbps y número de imágenes por segundo entre 3 y 25 proporcionan un rango de funcionamiento que cubre varios tipos de escenarios, desde canales muy restrictivos hasta entornos más favorables que permiten una mayor calidad de imagen.

Además, se han probado varias configuraciones, tanto de funcionamiento del algoritmo como del codificador en sí. En cuanto al algoritmo, se ha comprobado el funcionamiento actuando cada macrobloque, cada línea de macrobloques o cada imagen. En lo que respecta a la configuración del codificador, se han utilizado los dos modos de codificación aritmética disponibles.

Como resultado de estas pruebas se ha comprobado que el algoritmo es capaz de seguir las variaciones demandadas tanto de bits como de imágenes por segundo de forma eficiente.

Por último, la complejidad computacional que añade el método aquí presentado es mínima, por lo que se trata de un algoritmo aplicable a entornos en tiempo real como los que se plantean en los objetivos de esta tesis.

VII. Líneas futuras

El nuevo códec de vídeo H.265 es el más avanzado hasta la fecha, y ya empiezan a aparecer las primeras aplicaciones basadas en este estándar.

Por lo tanto, la dirección normal de la continuación de este trabajo consiste en el estudio y el desarrollo de un algoritmo de control de tasa con *bitrate* y tasa de imágenes por segundo variable adaptado a las características de este nuevo estándar.

Así, basándose en los nuevos algoritmos de control de tasa y en las decisiones de diseño de H.265 se deben analizar los requisitos y las propiedades de un nuevo algoritmo adaptado al nuevo paradigma.

Por otra parte, en el algoritmo de control de tasa presentado en esta tesis se hace hincapié únicamente en la asignación de bits a imágenes P. Sin embargo, la codificación de las imágenes I del GOP se codifican con un valor de QP fijo, lo que hace que la asignación de bits no sea tan estricta. Esto puede llevar a descartar varias imágenes después de cada imagen *intra*, lo que es perjudicial para la calidad perceptual de la secuencia completa.

Por lo tanto, es necesario estudiar un algoritmo de control de tasa que obtenga la mejor configuración posible del codificador para afrontar la codificación de las imágenes de referencia de las secuencias codificadas.

PROYECTOS

El trabajo llevado a cabo durante la elaboración de esta tesis ha sido soportado por:

- El proyecto nacional "*Nuevas técnicas para video vigilancia inteligente*", 2010-2013. Referencia TEC2009-09146, entidad Financiadora Ministerio de Educación
- Los proyectos de investigación con Telefónica I+D:
    - "*Desarrollos Tecnológicos sobre Coders de Vídeo H.263 y H.264*", 2005.
    - "*Desarrollos Tecnológicos sobre Coders de Vídeo H.264*", 2005-2006.
    - "*Codificación y Aplicaciones Multimedia para Redes Móviles*", 2006-2007.
    - "*Tecnologías Disruptivas para Servicios Avanzados en Movilidad*", 2007-2008.
    - "*Tecnologías Avanzadas para Videotelefonía Móvil*", 2008-2009.
- El "*Programa de apoyo a la investigación y desarrollo*" gestionado por la Universidad Politécnica de Valencia:
    - PAID-06-06 "*Detección Automática de Cambios de Plano y Escena en Codificación de Vídeo H.264*",

PUBLICACIONES

El trabajo de investigación realizado durante el desarrollo de esta tesis ha dado lugar a las siguientes publicaciones:

- J. Sastre, P. Usach, A. Moya, V. Naranjo, J. M. López, *Shot Detection Method for Low-Bitrate H.264 Video Coding*. Proceedings of the European Signal Processing Conference 2006. September 2006, Firenze, Italy
- P. Usach, J. Sastre, V. Naranjo and J.M. López. *Fast Shot Detection for High Quality Low Delay H.264 Video Coding*. Proceedings of the Picture Coding Symposium 2007. November 7 – 9, 2007, Lisboa, Portugal.
- P. Usach, J. Sastre and J.M. López. *Variable Frame Rate And GOP Size H.264 Rate Control For Mobile Communications*. Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, ICME 2009, June 28 - July 2, 2009, New York City, NY, USA. IEEE 2009, ISBN 978-1-4244-4291-1.
- P. Usach, J. Sastre, V. Naranjo, L. Vergara and J.M. López. *Content-based Dynamic Threshold Method for Real Time Keyframe Selecting*. IEEE Transactions on Circuits and Systems for Video Technology 20, no. 7 (2010), pp. 982-993.

BIBLIOGRAFÍA

[1] C. Cotsaces, N. Nikolaidis, I. Pitas, *Video Shot Detection and Condensed Representation. A Review*. IEEE Signal Processing Magazine. March 2006, pp. 28–37.

[2] G. Sullivan, T. Wiegand, *Video Compression. From Concepts to the H.264/AVC Standard*. Proceedings of the IEEE. December 2004.

[3] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, *Video Coding with H.264/AVC: Tools, Performance and Complexity*. IEEE Circuits and Systems Magazine. First Quarter 2004.

[4] T. Wiegand, G. Sullivan, G. Bjontegaard, A. Luthra, *Overview of the H.264/AVC Video Coding Standard*. IEEE Transactions on Circuits and Systems for Video Technology. July 2003.

[5] Y. Yuan, D. Feng, Y. Zhong, *A Novel Method of Keyframe Setting in Video Coding: Fast Adaptive Dynamic Keyframe Selecting*. Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing. IEEE Computer Society.

[6] J. Sastre, P. Usach, A. Moya, V. Naranjo, J. M. López, *Shot Detection Method for Low-Bitrate H.264 Video Coding*. Proceedings of the European Signal Processing Conference 2006. September 2006, Firenze, Italy.

[7] S. Youm, W. Kim, *Dynamic Threshold Method for Scene Change Detection*. IEEE International Conference on Multimedia and Expo 2003, pp. 337–340.

[8] I. Richardson, *H.264 and MPEG-4 Video Compression*. Wiley. 2003.

[9] ITU-T Recommendation H.264, *Advanced Video Coding for Generic Audiovisual Services*. 2003-2005.

[10] Joint Video Team of ITU-T and ISO/IEC JTC 1, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)*, document JVT-G050r1, May 2003; technical corrigendum 1 documents JVT-K050r1 (non-integrated form) and JVT-K051r1 (integrated form), March 2004; and Fidelity Range Extensions documents JVT-L047 (non-integrated form) and JVT-L050 (integrated form), July 2004.

[11] VideoLAN x264. Página web: http://www.videolan.org/developers/x264.html.

[12] S. Ma, W. Gao, F. Wu, Y. Lu, *Rate Control for JVT Video Coding Scheme with HRD Considerations*. IEEE International Conference on Image Processing 2003.

[13] S. Ma, W. Gao, Y. Lu, *Rate-distortion Analysis for H.264/AVC Video Coding and its Application to Rate Control*. IEEE Transactions of Circuits and Systems for Video Technology, Vol. 15, No. 12. December 2005. pp. 1533–1544.

[14] Z. Li, F. Pan, K. Lim, X. Lin, S. Rahardja, *Adaptive Rate Control for H.264*. IEEE International Conference on Image Processing 2004, pp. 746–748.

[15] R. Viswanathan, P.K. Varshney, *Distributed Detection with Multiple Sensors: Part I- Fundamentals*. Proc. IEEE, Vol. 85, no. 1, January 1997, pp. 54-63.

[16] L. Vergara, *On the Equivalence between Likelihood Ratio Tests and Counting Rules in Distributed Detection with Correlated Sensors*. IEEE Signal Processing, Vol. 87, July 2007, pp 1808-1815.

[17] E. Albert, *Algoritmos de detección de cambios de secuencia para vídeo H.264*. Proyecto final de carrera, ETSIT UPV. 19 de enero de 2006.

[18] S. Spinsante, E. Gambi and F. Chiaraluce, *An improved error concealment strategy driven by scene motion properties for H.264/AVC decoders*. 14th European Signal Processing Conference 2006.

[19] A. Dimonu, O. Nemethova and M. Rupp, *Scene change detection for H.264 using dynamic threshold techniques*. 5th EURASIP Conference, 2005.

[20] S. Youm and W. Kim, *Dynamic threshold method for scene change detection*. International Conference on Multimedia and Expo, 2003.

[21] A.Y. Lan, *Scene context dependent reference frame placement for MPEG video coding*. M.S. Thesis, University of Washington, Seattle. March 1996.

[22] Y. Yuan, D. Feng and Y. Zhong, *A novel method of keyframe setting in video coding: fast adaptive dynamic keyframe selecting*. International Conference on Computer Networks and Mobile Computing, 2003. IEEE Computer Society.

[23] S. Winkler and P. Mohandas, *The evolution of Video Quality Measurement: from PSNR to Hybrid Metrics.* IEEE Transactions on Broadcasting, Vol. 54, Nº3, September 2008.

[24] P. Usach-Molina, J. Sastre, V. Naranjo, L. Vergara, and J. M.L. Muñoz. 2010. *Content-Based Dynamic Threshold Method for Real-Time Keyframe Selecting*. IEEE Trans. Circuits and Systems for Video Technology 20, 7 July 2010, pp. 982-993.

[25] W. Hu, N. Xie, L. Li, X. Zeng, S.Maybank. *A Survey on Visual Content-Based Video Indexing and Retrieval*. IEEE Transactions on Systems, Man, and Cybernetics. Part C: Applications and Reviews, Vol. 41, No. 6, November 2011, pp. 797-819.

[26] J. S. Boreczky, L. A. Rowe. *Comparison of Video Shot Boundary Detection Techniques*. Storage and Retrieval for Still Image and Video Databases IV, No. SPIE 2664. January 1996.

[27] R. Mishra, S. Kumar. *A Review on Different Methods of Video Shot Boundary Detection*. International Journal of Electrical and Electronics Engineering (IJEEE), Vol. 1, Issue 1, August 2012, pp. 46-57.

[28] R. Lienhart. *Comparison of Automatic Shot Boundary Detection Algorithms*. Proc. SPIE 3656, Storage and Retrieval for Image and Video Databases VII. 17 December 1998.

[29] G. Rascioni, S. Spinsante, E. Gambi. *An Optimized Dynamic Scene Change Detection Algorithm for H.264/AVC Encoded Video Sequences*. International Journal of Digital Multimedia Broadcasting, Vol. 2010, Article ID 864123.

[30] Zhang, Dong, Wei Qi, and Hong Jiang Zhang. *A new shot boundary detection algorithm*. Advances in Multimedia Information Processing—PCM 2001. Springer Berlin Heidelberg, 2001. Pp. 63-70.

[31] A. Oliva and A. Torralba. *Modelling the Shape of the Scene: A Holistic Representation of the Spatial Envelope*. Int. J. Comput. Vision 42, 3. May 2001, pp. 145-175.

[32] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg and C. Schmid. *Evaluation of GIST descriptors for web-scale image search*. Proceedings of the ACM International Conference on Image and Video Retrieval 2009. ACM, New York, NY, USA, Article 19, 8 pages.

[33] M. Jacobs and J. Probell, *A Brief History of Video Coding*. ARC International, 2007.

[34] G. Sullivan, *Overview of International Video Coding Standards (preceding H.264/AVC)*. ITU-T VICA Workshop. 22-23 July 2005. ITU Headquarters, Geneva.

[35] G. Sullivan, J-R. Ohm, W-J. Han and T. Wiegand, *Overview of the High Efficiency Video Coding (HEVC) Standard*. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, December 2012, pp. 1649-1668.

[36] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*. IETF RFC 3550, Julio 2003.

[37] G. Sullivan, T. Wiegand and K.P. Lim, *Joint Model Reference Encoding Methods and Decoding Concealment Methods; Section 2.6: Rate Control*. JVT-I049, San Diego, September 2003.

[38] Pixeltools, *Rate Control and H.264*. http://www.pixeltools.com/

[39] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. Sullivan, *Rate-Constrained Coder Control and Comparison of Video Coding Standards*. IEEE Transactions on Circuits & Systems for Video Technology, 13, #7, July 2003.

[40] Y. Yu, J. Zhou and Y. Wang, *A fast effective scene change detection and adaptive rate control algorithm*. Proceedings of the International Conference on Image Processing, 1998. Volume 2, pages 379-382.

[41] H. Yu, F. Pan, Z. Lin and Y. Sun, *A perceptual bit allocation scheme for H.264*. IEEE International Conference on Multimedia and Expo, 2005, pp. 313-316.

[42] T. Berger, *Rate distortion theory: A mathematical basis for data compression*. Prentice-Hall, Englewood Cli s, NY, 1971.

[43] J. Ribas-Corbera and S. Lei, *Rate control in DCT video coding for low-delay communications*. Circuits and Systems for Video Technology, IEEE Transactions on, 9(1), 1999, pp. 172-185.

[44] ITU-T Recommendation *H.120. Codecs for videoconferencing using primary digital group transmission*. 1984-1993.

[45] ITU-T Recommendation *H.261. Video codec for audiovisual services at p x 64 kbit/s*. 1988-1993.

[46] ISO/IEC 10918-1. JTC 1/SC 29. *Coding of audio, picture, multimedia and hypermedia information*. 1990-2013.

[47] ISO/IEC 11172-1:1993. JTC 1/SC 29. *Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s. Part 1: Systems*. 1993-1998.

[48] ISO/IEC 13818-1:2015. *Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems*. 1998-2015.

[49] IEC 61834-1:1998. *Recording - Helical-scan digital video cassette recording system using 6,35 mm magnetic tape for consumer use (525-60, 625-50, 1125-60 and 1250-50 systems) - Part 1: General specifications*. 1994-1998.

[50] ITU-T Recommendation *H.263. Video coding for low bit rate communication*. 1996-2004.

[51] RealNetworks. *RealVideo version 1.0*. http://www.realnetworks.com/

[52] ISO/IEC 14496-2:1999. *Information technology - Coding of audio-visual objects - Part 2: Visual*. 1995-1999.

[53] DivX. http://www.divx.com/

[54] XviD. https://www.xvid.com/

[55] WMV9/VC-1. http://www.microsoft.com/windows/windowsmedia/howto/articles/vc1techoverview.aspx

[56] ITU-T Recommendation *H.265. High efficiency video coding*. 2013-2015.

[57] *El Rey Arturo*. Dirigida por Antoine Fuqua. Touchstone Pictures, Jerry Bruckheimer Films, Green Hills Production, 2004. 1 DVD: 126 min.

[58] *El Señor de los Anillos. Las dos torres*. Dirigida por Peter Jackson. New Line Cinema, WingNut Films, The Saul Zaentz Company, 2002. 1 DVD: 179 min.

[59] *Hero*. Dirigida por Yimou Zhang. Beijing New Picture Film Co., China Film Co-Production Corporation, Elite Group Enterprises, 2002. 1 DVD: 99 min.

[60] *Destino Final 2*. Dirigida por David R. Ellis. New Line Cinema, Zide-Perry Productions, 2003. 1 DVD: 90 min.

[61] *La pasión de Cristo*. Dirigida por Mel Gibson. Icon Productions, 2004. 1 DVD: 127 min.

[62] *Piratas del Caribe. La maldición de la Perla Negra*. Dirigida por Gore Verbinski. Walt Disney Pictures, Jerry Bruckheimer Films, 2003. 1 DVD: 143 min.

[63] *El Señor de los Anillos. El retorno del Rey*. Dirigida por Peter Jackson. New Line Cinema, WingNut Films, The Saul Zaentz Company, 2003. 1 DVD: 201 min.

[64] *Matrix Reloaded*. Dirigida por Andy y Lana Wachowski. Warner Bros., Village Roadshow Pictures, Silver Pictures, 2003. 1 DVD: 138 min.

[65] *Kill Bill. Volumen 1*. Dirigida por Quentin Tarantino. Miramax, A Band Apart, Super Cool ManChu, 2003. 1 DVD: 111 min.

[66] *Matrix*. Dirigida por Andy y Lana Wachowsky. Warner Bros., Village Roadshow Pictures, Silver Pictures, 1999. 1 DVD: 136 min.

[67] *Pulp Fiction*. Dirigida por Quentin Tarantino. Miramax, A Band Apart, Jersey Films, 1994. 1 DVD: 154 min.

ANEXO

I. Secuencias de entrenamiento y test

En este anexo se describen las secuencias de vídeo utilizadas para entrenar y probar tanto el algoritmo de detección de cambios de plano abruptos como el algoritmo de inserción de *keyframes* basada en el contenido que se han presentado en estas páginas. En los apartados siguientes se presenta un resumen de las secuencias incluidas en los conjuntos de Entrenamiento y Test incluyendo la siguiente información:

- **Nombre**: nombre de la secuencia de vídeo original de la que se ha extraído el fragmento correspondiente.
- **Fotogramas**: posición de los *frames* que se han extraído de la secuencia.
- **Candidatos**: número aproximado de candidatos a la inserción de *intras* que aparecen en la secuencia. Este número se ha calculado como parte del *ground truth* obtenido para la medida de las prestaciones del algoritmo de detección de cambios de plano abruptos, correspondiendo con el número de cambios de plano existentes.
- **Categoría de movimiento**: categoría de movimiento a la que pertenece la secuencia, según las definiciones de la sección 2-IV.1.
- **Descripción**: descripción general del contenido de la secuencia, así como de las características más importantes que aparecen en ella (transiciones abruptas, grado de movimiento, etc.).
- **Muestra**: selección de fotogramas de la secuencia en formato QCIF.

Como se puede comprobar en estos apartados, tanto el conjunto de Entrenamiento como el conjunto de Test constan de 7 secuencias cada uno, con cada secuencia formada por 5000 *frames* consecutivos.

En total el número aproximado de candidatos para la inserción de *keyframes* (cambios de plano) es de 1500 entre ambos conjuntos de secuencias.

II. Conjunto de Entrenamiento

En este apartado se describen las secuencias agrupadas en el conjunto de entrenamiento.

II.1. El Rey Arturo

- **Nombre**: El Rey Arturo [57].
- **Fotogramas**: 10001 – 15000.
- **Candidatos**: 147.
- **Categoría de Movimiento**: HM&HC.
- **Descripción**: Escena de batalla en campo abierto, con movimientos rápidos de cámara, objetos y personajes. Frecuentes oclusiones, cambios de plano y texturas complejas.

- **Muestra**:

Fig. 35: Selección de fotogramas de la secuencia El Rey Arturo

II.2. Las dos torres

- **Nombre**: Las dos torres [58].
- **Fotogramas**: 15001 – 20000.
- **Candidatos**: 108.
- **Categoría de Movimiento**: HM&HC.
- **Descripción**: Escena de batalla con *pannings* constantes, zooms, primeros planos y objetos en movimiento.
- **Muestra**:

Fig. 36: Selección de fotogramas de la secuencia Las dos torres

II.3. Hero

- **Nombre**: Hero [59].
- **Fotogramas**: 10001 – 15000.
- **Candidatos**: 144.
- **Categoría de Movimiento**: MM&MC.
- **Descripción**: Alternancia de escenas de movimiento suave o inexistente (diálogos) y escenas con mucho movimiento (combates). Movimiento moderado de cámara y personajes.
- **Muestra**:

Fig. 37: Selección de fotogramas de la secuencia Hero

II.4. Destino Final II

- **Nombre**: Destino Final II [60].
- **Fotogramas**: 35001 – 40000.
- **Candidatos**: 125.
- **Categoría de Movimiento**: MM&MC.
- **Descripción**: Escena de tráfico con oclusiones, grandes objetos en movimiento, movimiento de cámara, etc.

- **Muestra**:

Fig. 38: Selección de fotogramas de la secuencia Destino Final II

II.5. La pasión de Cristo

- **Nombre**: La pasión de Cristo [61].
- **Fotogramas**: 55001 – 60000.
- **Candidatos**: 65.
- **Categoría de Movimiento**: LM&LC.
- **Descripción**: Escenas de diálogos con cámara fija y algunos *pannings* con oclusiones.
- **Muestra**:

Fig. 39: Selección de fotogramas de la secuencia La pasión de Cristo

II.6. Piratas del Caribe

- **Nombre**: Piratas del Caribe [62].
- **Fotogramas**: 418 – 5417.
- **Candidatos**: 115.
- **Categoría de Movimiento**: LM&LC.
- **Descripción**: Escenas alternadas de diálogos y *travelings* con gran cantidad de cambios de plano abruptos.
- **Muestra**:

Fig. 40: Selección de fotogramas de la secuencia Piratas del Caribe

II.7. Anuncios I

- **Nombre**: Anuncios I.
- **Fotogramas**: 1 – 5000.
- **Candidatos**: 116.
- **Categoría de Movimiento**: COM.
- **Descripción**: Frecuentes cambios de plano combinados con transiciones graduales. Se incluyen anuncios comerciales, *trailers* de películas, etc.

- **Muestra**:

Fig. 41: Selección de fotogramas de la secuencia Anuncios I

III. Conjunto de Test

En este apartado se describen las secuencias agrupadas en el conjunto de test.

III.1. El retorno del Rey

- **Nombre**: El retorno del Rey [63].
- **Fotogramas**: 47501 – 52500.
- **Candidatos**: 78.
- **Categoría de Movimiento**: HM&HC.
- **Descripción**: Escena de combate con una gran multitud en movimiento. Gran cantidad de movimientos de cámara abruptos, oclusiones, etc.

- **Muestra**:

Fig. 42: Selección de fotogramas de la secuencia El retorno del Rey

III.2. Matrix Reloaded

- **Nombre**: Matrix Reloaded [64].
- **Fotogramas**: 121201 – 126200.
- **Candidatos**: 144.
- **Categoría de Movimiento**: HM&HC.
- **Descripción**: Escena de tráfico con movimientos abruptos de cámara, flashes y oclusiones.
- **Muestra**:

Fig. 43: Selección de fotogramas de la secuencia Matrix Reloaded

III.3. Kill Bill

- **Nombre**: Kill Bill [65].
- **Fotogramas**: 107001 – 112000.
- **Candidatos**: 93.
- **Categoría de Movimiento**: MM&MC.
- **Descripción**: Escenas alternadas de diálogos y combates con transiciones abruptas entre ellas.
- **Muestra**:

Fig. 44: Selección de fotogramas de la secuencia Kill Bill

III.4. Matrix

- **Nombre**: Matrix [66].
- **Fotogramas**: 70401 – 75400.
- **Candidatos**: 137.
- **Categoría de Movimiento**: MM&MC.
- **Descripción**: Varios *pannings* con movimiento de objetos y personajes en escena.

- **Muestra**:

Fig. 45: Selección de fotogramas de la secuencia Matrix

III.5. Pulp Fiction (I)

- **Nombre**: Pulp Fiction (I) [67].
- **Fotogramas**: 25101 – 30300.
- **Candidatos**: 48.
- **Categoría de Movimiento**: LM&LC.
- **Descripción**: Escena de diálogos con cámara fija y cambios de plano abruptos.
- **Muestra**:

Fig. 46: Selección de fotogramas de la secuencia Pulp Fiction (I)

III.6. Pulp Fiction (II)

- **Nombre**: Pulp Fiction (II) [67].
- **Fotogramas**: 202001 – 207000.
- **Candidatos**: 50.
- **Categoría de Movimiento**: LM&LC.
- **Descripción**: Escena de diálogos con cámara fija, cambios de plano abruptos y fondo dinámico.
- **Muestra**:

Fig. 47: Selección de fotogramas de la secuencia Pulp Fiction (II)

III.7. Anuncios II

- **Nombre**: Anuncios II.
- **Fotogramas**: 1 – 5000.
- **Candidatos**: 160.
- **Categoría de Movimiento**: COM.
- **Descripción**: Cambios de plano frecuentes, incluyendo transiciones graduales y otros efectos. Se incluyen anuncios de televisión, trailers de películas, etc.

- **Muestra**:

Fig. 48: Selección de fotogramas de la secuencia Anuncios II