UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

PROYECTO FIN DE CARRERA

APROXIMACIÓN AL ANÁLISIS DE SECUENCIAS DE VÍDEO CODIFICADAS EN H.264

Diego Sarasúa Álvarez

Julio 2010

Aproximación al análisis de secuencias de vídeo codificadas en H.264

AUTOR: Diego Sarasúa Álvarez

TUTOR: Jesús Bescós Cano

Vídeo Processing and Understanding Lab

Dpto. de Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio de 2010

Palabras clave:

H.264, cuadro, *frame*, macrobloques, vectores de movimiento, redundancia espacial, redundancia temporal, cambios de toma, predicción temporal, predicción espacial, modos de predicción, estimación de movimiento.

Resumen:

Este proyecto consiste en el estudio del estándar H.264, así como de las técnicas ya desarrolladas para la extracción de parámetros para el análisis de secuencias de vídeo en dicho dominio. Trabajar en el dominio comprimido tiene la utilidad de extraer conclusiones tras el análisis de un vídeo en tiempo real. Se observan así mismo las diferencias existentes entre el estándar H.264 y sus predecesores MPEG-1 y MPEG-2. Posteriormente al estudio del estándar y comparación con estándares anteriores, se procede a la creación de una serie de herramientas con el objetivo de extraer dichos parámetros con los que podemos trabajar en tiempo real, sin descomprimir el vídeo. Una vez extraídos, se realiza un análisis de dichos resultados para intentar obtener la mayor información disponible a partir de estos datos, generando una opinión acerca de su importancia y si se pueden sacar conclusiones en aplicaciones concretas y con qué exactitud o fiabilidad.

Abstract:

This Project consists on working in the compressed domain of H.264 and has the objective of studying H.264 standard and its predecessors MPEG-1 and MPEG-2. If we work in the compressed domain, we can then analyze videos on real-time. After studying from which parameters we can get valuable information, we create different kind of tools for extracting those parameters, without decompressing video. Then, we realize analysis of this information for obtaining all information me can, making an opinion about the importance and if we could get valuable and faithful information from different applications.

AGRADECIMIENTOS

En primer lugar, me gustaría dar las gracias especialmente a mi tutor Jesús Bescós, quien confió en mí desde el primer día y, casi sin pedírselo, me dio la oportunidad de poder entrar en el grupo y poder realizar el Proyecto Fin de Carrera con ellos.

También agradecer a todo mi grupo del *VPUlab* el tratamiento que me han dado y la facilidad que tienen para mantener un buen ambiente a diario. Para la posteridad quedarán esas grabaciones en el laboratorio y esos manjares típicos de cada sitio a donde iba cada miembro del grupo. Y cómo no, de esos partidos de fútbol sala: ¡Bayona, deja de protestar!

En especial dentro del grupo, mi más sincero agradecimiento a Luis Herranz, por su entera disponibilidad y su ayuda incondicional en el día a día. Sin él, el camino hubiera sido mucho más arduo.

No puedo olvidar mencionar a todos los profesores que hemos tenido desde primer curso, incluyendo a aquellos que tuvimos en el primer cuatrimestre de primer curso y no volvimos a tener. Si algo hay que destacar de manera positiva en esta facultad es el trato del profesor hacia el alumno, siempre preocupado por su aprendizaje. No quiero dejar de mencionar a una profesora que no he tenido, pero la he conocido por diversas circunstancias y me ha ayudado mucho y me ha enseñado que, aún quitándote injustamente eso por lo que se ha luchado y conseguido merecidamente, sólo hay un camino: el de continuar trabajando, y el del compromiso. Gracias a Susana Holgado.

En estos años de universidad he conocido a mucha gente, mucha más de la que podía suponer antes de entrar y tengo que decir que, aunque todos siempre conocemos a esa gente con la que no nos hubiera sido posible afrontar la carrera, he pasado grandes momentos con todos y cada uno de mis compañeros, y eso no me parece tan fácil. Cómo voy a echar de menos estos años universitarios…

Quiero mencionar aquí también a los culpables de la ocupación mayoritaria de mi tiempo libre. Esa gente de toda la vida, desde el colegio y 6 años después de separarnos ahí siguen. Porque sois muchos (afortunado me siento de poder decir esto) no voy a mencionaros uno por uno, ya sabéis quiénes sois. ¡Muchas gracias!

A mi familia, que al fin y al cabo son los que me han permitido hacer esto, y los que han aguantado mis cambios de humor en épocas de exámenes o de laboratorios, siempre tan difíciles de llevar. Y porque sé que siempre estaréis ahí, incluso con esos emails futboleros estemos donde estemos, hermanitos. A mi abuela, que sé que le hace incluso más ilusión que a mi y a mi primo Juanqui, el tío más fuerte y luchador de todos los Sarasúa, que ya es decir.

Y, por último, no me olvido de los Ruiz Albacete, por su apoyo en estos últimos años. Arancha, ya sabes dónde celebraremos esto cuando vuelvas. Y, cómo no, a la chica que me ha marcado en mi carrera y en mi vida, sin ti nada hubiera sido lo mismo, y mejor imposible. Este proyecto también es tuyo Vir, pero no me dejan ponerle tu nombre. Ya sabes que "*Everything I do…*".

GRACIAS

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 INTRODUCCIÓN

1.1 MOTIVACIÓN

Actualmente el análisis de vídeo es un tema muy importante en nuestra sociedad, donde ha quedado claro que el sector de las telecomunicaciones y la informática va a dar mucho que hablar en el siglo XXI.

En los auges de esta era digital, donde lo analógico está desapareciendo progresivamente (apagón de la TV analógica en abril de 2010), la compresión se vuelve algo fundamental. La digitalización de las señales de vídeo y audio proporciona una gran cantidad de ventajas respecto a la señal analógica que pronto desaparecerá: resoluciones superiores, calidad, robustez frente al ruido y a errores en la propagación, etc. Su mayor inconveniente es la cantidad de información generada en el proceso de digitalización. Así, por ejemplo, una señal de vídeo con resolución 720x480, a una velocidad de 30 *frames*/seg (30 Hz), con 24 bits/píxel, hacen falta aproximadamente unos 250 Mbps.

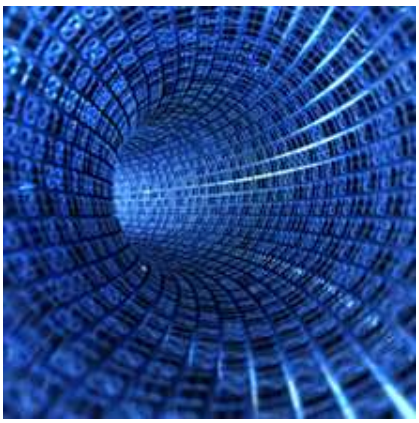


FIG. 1-1 ERA DIGITAL

Es por ello que la compresión de vídeo se vuelve un tema fundamental en la digitalización. Aquí es donde entra en juego la importancia de la estandarización de la compresión de vídeo. MPEG aparece con su primera versión MPEG-1, enfocada principalmente para el almacenamiento en CD y la reproducción de señales de vídeo digital, con velocidad mínima de 1’5 Mbps y calidad comparable a VHS. Posteriormente, surge MPEG-2, cuyo objetivo principal es transmitir señales de televisión digital a través de cualquier medio (terrestre, satélite…), con calidad de HDTV inclusive. MPEG-2 fue elegido como estándar de transmisión de televisión digital para DVB (Digital Video Broadcasting). Hoy en día, MPEG-4 (una de cuyas partes es comparable a H.264) está centrado en aplicaciones multimedia, tales como videoconferencias.

La motivación de este Proyecto Fin de Carrera es abarcar todos los ámbitos posibles de análisis de secuencias de vídeo codificadas siguiendo el estándar H.264, analizar en profundidad todas las posibilidades que este estándar nos ofrece, y ver cuáles son los parámetros extraíbles en el dominio comprimido. Una vez deducidos dichos parámetros, nos

centraremos en un estudio completo de sus aplicaciones basándonos en la documentación previa y los trabajos existentes en los distintos terrenos a analizar.

A partir de un estudio individual de los parámetros y viendo las posibilidades que nos ofrecen (en términos de análisis de vídeo), realizaremos unas pruebas a partir del codificador/decodificador JM para comprobar o rebatir argumentos, tanto a favor como en contra, de la utilidad de cada uno de los parámetros. Se quiere dejar claro que no se busca una optimización de la funcionalidad de cada parámetro, sino de una opinión objetiva acerca de su verdadera utilidad en temas de análisis de vídeo.

1.2 OBJETIVOS

El objetivo de este Proyecto Fin de Carrera es realizar un análisis exhaustivo de todas las características del estándar de compresión H.264 como base para el análisis de secuencias de vídeo en este dominio, para poder trabajar en tiempo real (cualidad fundamental en análisis de vídeo).

Esta memoria se desarrollará suponiendo unos conocimientos mínimos en el área de compresión de vídeo y sobre los estándares MPEG-1 y MPEG-2.

Los diferentes objetivos de que consta este Proyecto son:

1. Estudio de las características del estándar H.264.
2. Análisis de las semejanzas y diferencias con respecto a estándares anteriores: MPEG-1 y MPEG-2.
3. Estudio de la información disponible en un vídeo comprimido según el estándar H.264.
4. Creación de herramientas para la extracción de parámetros en el dominio comprimido de H.264 mediante el codificador/decodificador JM.
5. Análisis de los resultados obtenidos, resaltando la utilidad en temas de análisis de vídeo y su eficiencia y fiabilidad.
6. Presentación de conclusiones tras el análisis de los resultados obtenidos.

1.3 ORGANIZACIÓN DE LA MEMORIA

La memoria de este "Proyecto Fin de Carrera" quedará organizada de la siguiente manera:

Capítulo 1: *Introducción*. Se incluirán las motivaciones, objetivos y organización de la memoria.

Capítulo 2: *Antecedentes*. Descripción de las características más importantes de los estándares de vídeo MPEG (incluido H.264). Se hablará también de las distintas aplicaciones en las que es útil el estándar H.264 y por qué cada día tiene más acogida. También se hará un análisis de los dominios transformados previos a H.264, es decir, MPEG-1 y MPEG-2.

Capítulo 3: *Análisis inicial comparativo*. Se profundizará en los cambios producidos en H.264 con respecto a estándares anteriores (MPEG-1, MPEG-2) en los terrenos de nuestro interés de cara al análisis de vídeo. Adicionalmente se comentarán las diferencias existentes con determinados parámetros que proporcionaban una gran utilidad en la extracción de información de un vídeo y que, tras los cambios aplicados en el estándar H.264, dejan de tener dicha utilidad.

Capítulo 4: *Selección de parámetros extraíbles*. Selección de parámetros a extraer, definiendo la utilidad y aplicaciones que podemos dar a cada uno y referenciando trabajos realizados anteriormente sobre estos puntos. Breve resumen de los parámetros a extraer en el apartado siguiente. Finalmente se explica el desarrollo software implementado para dicha extracción de parámetros.

Capítulo 5: *Pruebas y resultados*. Análisis de los resultados obtenidos, ventajas e inconvenientes de cada uno de ellos, haciendo especial hincapié en la eficiencia de dichos parámetros y en la funcionalidad prevista previamente a la extracción de pruebas y análisis de posibles nuevas aplicaciones. En los casos en que corresponda, se hablará en términos de "*Recall*" y "*Precision*" como características para medir la eficiencia.

Capítulo 6: *Conclusiones*. Comentarios acerca de los resultados obtenidos en el apartado anterior.

2 ANTECEDENTES

2.1 INTRODUCCIÓN

Es importante comenzar este trabajo con una breve descripción de las características fundamentales del estándar MPEG, el cual es la base de todo este trabajo. Dicha base ha sido adquirida en gran parte a través de dos libros de referencia en este campo[1][2].

FIG. 2-1 MPEG

El equipo MPEG (*Moving Picture Experts Group*) se encarga del desarrollo de estándares para elementos de vídeo y audio. Pertenecen al organismo de estandarización ISO/IEC y su designación oficial es ISO/IEC JTC1/SC29 WG11.

Con el paso del tiempo MPEG ha ido evolucionando e introduciendo mejoras en sus estándares dando lugar a los estándares MPEG-1, MPEG-2, MPEG-4…. Este trabajo se centrará en vídeo MPEG-4, concretamente en la parte 10 de este estándar: estándar H.264, códec de vídeo digital con el principal objetivo de conseguir una importante compresión de datos. Este estándar proporciona una compresión considerablemente mayor que la de estándares anteriores, reduciéndose la tasa binaria hasta un 50% respecto a la de MPEG-2.

El estándar H.264 está asumiendo una creciente popularidad debido a que portales Web como *Youtube* o *software* como *Adobe Flash Player 9* ya lo han incorporado y soportan este formato. Las emisiones por satélite, los HD-DVD y los *Blu-ray* usan también H.264.

FIG. 2-2 H.264

Nuestro trabajo consiste en saber cuáles son los parámetros extraíbles en el dominio comprimido de H.264. Buscamos comprender su funcionamiento y su utilidad en distintos ámbitos de análisis de secuencias de vídeo ya abordados sobre estándares anteriores, como la segmentación de objetos, la detección de cambios de toma, marcar los contornos de personas

u objetos en una imagen, etc. Nos ponemos la restricción de trabajar en el dominio comprimido para exprimir al máximo las oportunidades de trabajar con un vídeo en tiempo real.

Intuitivamente, el vídeo digital se comprime con el objetivo de ahorrar espacio. Un códec se encarga de realizar la codificación y decodificación (Codificador-DECodificador). Así pues, una mejora sustancial de los estándares de compresión en los que se basan dichos códecs, permitiría transmitir vídeo de mayor calidad usando la misma cantidad de ancho de banda. A partir de H.264 conseguimos reducir la transmisión de información necesaria para reproducir un vídeo. Estos codificadores no buscan enviar la secuencia de vídeo con pérdidas nulas, sino reducir dichas pérdidas a cantidades aceptables que permitan una visualización de la secuencia de vídeo lo más fiel a la secuencia original posible. Aquí nos encontramos con un compromiso entre compresión y calidad de vídeo que H.264 adapta tras sus mejoras. El objetivo es encontrar los medios adecuados para no perder esa calidad visual a la vez que se logra la mayor compresión posible. Eligiendo los valores adecuados, se pueden llegar a compresiones de 100:1 con diferencias de calidad respecto al original imperceptibles.

2.2 CONCEPTOS BÁSICOS SOBRE H.264

A continuación, veremos ciertos conceptos que son importantes tener en cuenta antes de empezar a trabajar en el estándar H.264. Recordar que H.264 es una mejora respecto a estándares anteriores y, como tal, se basa en ellos añadiendo o modificando ciertas características que veremos más adelante.

2.2.1 MACROBLOQUES

Antes de definir un macrobloque (MB), deberíamos definir lo que es un bloque, pues es la unidad mínima de tratamiento de imágenes constituido por un grupo de 8x8 muestras de Y, Cr o Cb (ver los cuatro macrobloques de la figura "*Fig. 2-3*"). Podemos referirnos a la crominancia simplemente a través de Cr o Cb; a partir de ahora, utilizaremos la notación Cb. YC_bC_r es un espacio de color, el cual lo encontramos en vídeo digital y en sistemas digitales de fotografía. Tiene tres componentes: un componente de luminancia (Y) y dos de crominancia (C_b y C_r). Estos últimos representan los colores azul y rojo respectivamente.

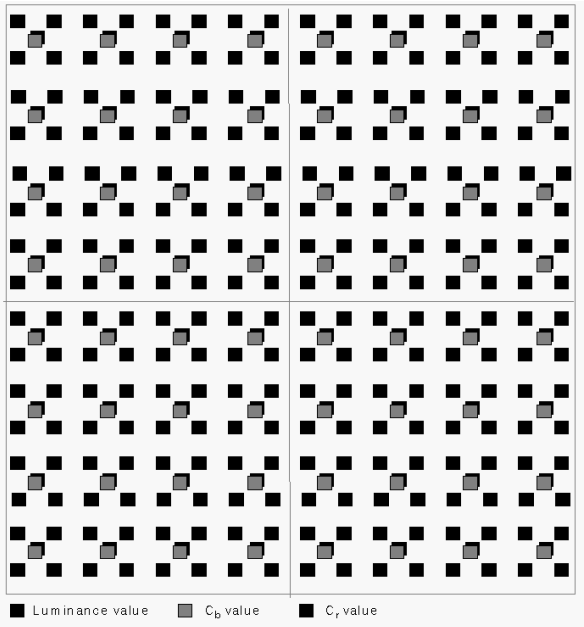


FIG. 2-3 MACROBLOQUES

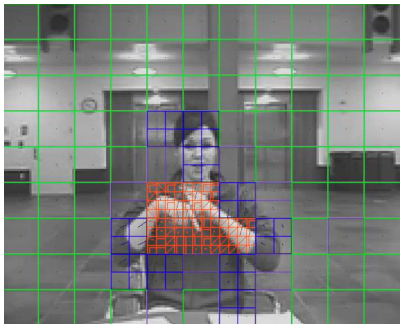


FIG. 2-4 DISTINTAS ESTRUCTURAS DE MACROBLOQUES

Un macrobloque consiste en el mínimo conjunto de bloques enteros de Y (luminancia) y Cr (crominancia) que ocupan la misma posición espacial en el cuadro (imagen). Es la mínima unidad sobre la que se tratan los vectores de movimiento determinándose así el desplazamiento respecto a macrobloques con una información parecida. Estos macrobloques pueden estar divididos en bloques de manera diferente, desde 16x16 hasta 4x4, bloques verdes y rojos respectivamente en la figura "*Fig. 2-4*".

2.2.2 SLICES

Un *slice* ("tira" en español, aunque utilizaremos el término "*slice*") se trata de un conjunto de macrobloques consecutivos del cuadro. Estos macrobloques consecutivos siguen un recorrido de izquierda a derecha y de arriba a abajo, siendo la longitud del *slice* variable y a gusto del codificador. Los *slices* son la unidad fundamental de sincronización en el posible caso de errores. Habitualmente, como veremos más adelante, se suele tratar con un *slice* por cuadro, es decir, que un solo *slice* constituye el cuadro entero.

Existen *slices* tipo I, P y B (en el apartado siguiente, el 2.2.3, veremos con detalle en qué consisten los tipos I, P y B). Y en H.264 aparecen dos nuevos tipos de *slices*: SP, SI, que se explicarán en el apartado *3.1.2 Slices"*.

2.2.3 TIPOS DE IMÁGENES

Como resultado de la compresión MPEG, aparecen diferentes tipos de imágenes:

Imágenes tipo I: Intra

Imagen codificada sin estar referida a ninguna otra imagen, es decir, no utiliza ningún tipo de predicción temporal. Se codifica y se refiere exclusivamente a sí misma (se suele decir que están intracodificadas). Lógicamente, requieren mayor número de bits para codificarse aunque menos tiempo de codificación. Se lleva a cabo una codificación muy similar a la que se usa en JPEG (*Joint Photographic Experts Group*), aunque con algunas diferencias como las tablas de cuantificación.

De cara a la decodificación, las imágenes de tipo I se utilizan generalmente como imágenes de referencia para otras imágenes, generando así períodos de refresco, importantes en videoconferencias a aplicaciones de "broadcast".

Imágenes tipo P: Predicted

Imagen que realiza una predicción temporal tanto en el codificador como en el decodificador utilizando como referencia una imagen almacenada en un buffer *DPB (Decoded Picture Buffer*) las imágenes, ya decodificadas, que se usan como referencia para imágenes P son siempre de instantes anteriores en el tiempo a las del vídeo en cuestión, a diferencia de las imágenes de tipo B que se verán a continuación). Una vez codificadas las imágenes de referencia, las imágenes tipo P se reconstruirán a partir de ellas añadiéndoles una estimación de movimiento. Dicho movimiento se calculará mediante comparación por bloques con la imagen P. Tras dividir ambas imágenes en bloques de igual tamaño, los bloques de la imagen tipo P se comparan con los de la imagen de referencia para encontrar el bloque más parecido. Se considerará óptimo el que resulte en menor diferencia entre ambos. Aunque la información a enviar es mucho menor que en imágenes I, es también menos fiable.

Cabe destacar una característica especial de este tipo de imágenes, y es que es posible omitir la codificación de algunos macrobloques dentro de la imagen (*skip* cuando las diferencias con respecto al mismo bloque de la imagen anterior son mínimas. Esto se puede entender como una predicción con vector de movimiento nulo (sin movimiento) referenciándose al bloque de la imagen anterior. Así, se sustituiría el bloque por los mismos píxeles de la imagen anterior.

Otra característica de las imágenes P es que sirven de referencia de otras imágenes P y también de las imágenes B, que utilizan predicción temporal bidireccional.

Imágenes tipo B: Bi-directional, Bi-predicted

Las imágenes tipo B son similares a las imágenes tipo P, con la diferencia que pueden referenciarse también a imágenes posteriores en el tiempo, aunque codificadas y decodificadas previamente para poder llevar a cabo dicha predicción. En H.264, las imágenes B se pueden utilizar o no como referencias de otras imágenes B (a gusto del codificador), y también existe la opción de utilizar una, dos o más imágenes como referencia de imágenes B, generando así un orden de reproducción completamente variable.

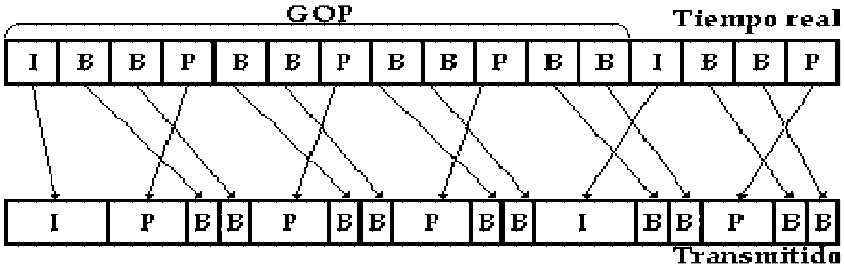


FIG. 2-5 SECUENCIAS DE CUADROS REAL Y TRANSMITIDA

En la figura "*Fig. 2-5*" podemos observar un ejemplo de secuencia de cuadros en tiempo real y la secuencia transmitida. Dado que los cuadros B tienen predicción bidireccional (utilizan como referencia cuadros anteriores y/o posteriores) se ha de codificar/decodificar previamente el cuadro P del que dependen, el cual sólo utiliza como referencia el cuadro I anterior, por lo que puede codificarse antes que los cuadros B.

Hay que tener en cuenta que las imágenes P y B no se podrán utilizar si, por cualquier causa, se pierden las imágenes decodificadas previamente (de las que dependen).

Una imagen I, junto con todas las imágenes hasta la siguiente imagen I, forman un GOP (*Group Of Pictures*). No hay ninguna norma que indique la dimensión ni la estructura de un GOP, ya que pueden variar en función de las condiciones impuestas por el codificador. Obviamente, un GOP más largo (con más imágenes P ó B intercaladas entre dos imágenes I) resultará en una tasa de transmisión mucho menor. Sin embargo, un GOP muy largo retrasará la recuperación debida a un error en la transmisión.

Estas imágenes B requieren menos tasa de transmisión que las imágenes I y P, aunque las imágenes I siguen teniendo mayor fiabilidad, pues son resultantes de una intra-codificación y no de una estimación de movimiento con respecto a otro bloque. A su vez, dada una mayor complejidad en la codificación de este tipo de imágenes, también cabe destacar que el tiempo de codificación es mayor.

2.2.4 REDUNDANCIA

Muy importante y fundamental en todo sistema de compresión es aprovechar la redundancia de la señal para conseguir reducir la tasa de transmisión. Dicha redundancia puede ser temporal o espacial, lo que permite realizar una predicción "inter-frame" (entre cuadros distintos) o codificación "intra-frame" (dentro del mismo cuadro) respectivamente.

Codificación temporal

La redundancia temporal hace referencia a que en un vídeo, y más concretamente en una escena, el paso de un *frame* al siguiente (habitualmente 25, 30 *frames* por segundo) conlleva muy pocas diferencias de píxeles, sobre todo cuando se trata de imágenes en cámara fija.

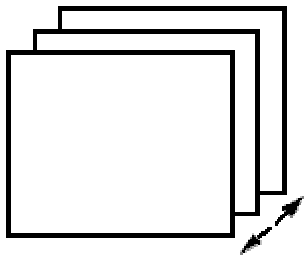


FIG. 2-6 REDUNDANCIA TEMPORAL

Por tanto, es posible codificar no la información de la imagen en sí, sino el error de predicción con respecto a la imagen anterior, guardando previamente dichas imágenes en el buffer. Es necesario que dicho almacenamiento se realice tanto en el codificador como en el decodificador para asegurar su sincronización, pues así ambos podrán realizan la misma predicción. Con una nueva imagen disponible para codificar, hallamos la imagen resultante de la predicción y hallamos en el mismo codificador el error con respecto de la original, y es este error lo que enviamos al decodificador. El decodificador, sabiendo sobre qué bloques se realiza dicha predicción, realiza la misma predicción que el codificador y, con ese mismo resultado, suma el error que recibe y reconstruye la imagen original. Ganamos sobre todo en tasa de transmisión, debido al cambio de enviar el error de una predicción en lugar de la imagen original en sí.

Es por esto que, si se pierden las imágenes del buffer, se pierde también la información fundamental para poder reconstruir las siguientes imágenes y se perdería toda posibilidad de reconstruirlas.

Estimación de movimiento

Hablar de estimación de movimiento, o compensación de movimiento, es hablar de los vectores de movimiento (MV - "Motion Vectors"), ya utilizados en MPEG-1. Tras un análisis dentro de una ventana de búsqueda, estos vectores de movimiento resultan en la localización del bloque más similar al que se está codificando, de entre los posibles dentro de los cuadros previamente codificados.

La elección de los vectores de movimiento puede realizarse de una manera más o menos precisa a través de un análisis de similitud entre el macrobloque a codificar y los macrobloques de imágenes previamente codificadas. Se marcarán como áreas de potenciales coincidencias

posibles las que estén más cerca del macrobloque que estemos codificando en imágenes previas, conocidas como ventanas de búsqueda. También se puede apelar directamente a la búsqueda exhaustiva, la cual buscará en todos los macrobloques de la imagen de referencia, previamente codificada. Obviamente, con la búsqueda exhaustiva se logrará una predicción necesaria para codificar el error de predicción a partir del cual el decodificador podrá reconstruir la imagen original. Debido a esta búsqueda exhaustiva, las predicciones serán mejores y, por lo tanto, se obtendrá una menor tasa de transmisión.

Lamentablemente, existen posibilidades de transmitir errores al utilizar una secuencia compuesta exclusivamente de imágenes con predicción (imágenes P ó B). Por ello desde sus primeros estándares, MPEG siempre envía una imagen sin pérdidas e idéntica a la original; este tipo de imágenes son conocidas como las IDR.

Además de utilizar una ventana de búsqueda (de tamaño menor que el tamaño de la imagen original) para reducir la cantidad de tiempo de codificación de la secuencia de imágenes, también se pueden utilizar otras técnicas encaminadas a la búsqueda del bloque más similar al que se está codificando, como se pueden ver en [3].

Adicionalmente, mencionar que los macrobloques pueden ser divididos a su vez en bloques de 16x8, 8x16… hasta tamaños de 8x8 y, a su vez, estos bloques de 8x8 se pueden dividir en bloques más pequeños de 8x4, 4x8 ó 4x4. Esto favorece que cada sub-bloque tenga su propia predicción sobre otro bloque previamente codificado incrementando la precisión, llegando a ser de ¼ de píxel en H.264.

Codificación espacial

En H.264 (como novedad respecto a estándares anteriores), la redundancia espacial se refiere al análisis de la semejanza de cada bloque en relación a los bloques ya codificados del cuadro en cuestión.

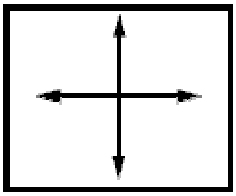


FIG. 2-7 REDUNDANCIA ESPACIAL

Generalmente, dentro de una misma imagen los píxeles adyacentes tienden a estar muy correlacionados. Es por ello que se trabaja con la DCT (*Discrete Cosine Transform*), explicada a continuación, aunque ahora en H.264 también se utilizan los residuos, diferencias entre unos bloques y otros, con lo cual perdemos la información que contenían los coeficientes DC y AC de bloques intracodificados en MPEG-1 y MPEG-2.

2.2.5 DCT

La DCT extrae una serie de puntos resultantes de la suma de distintas señales sinusoidales, cada una con distintas amplitudes y frecuencias. Es similar al funcionamiento de la DFT (*Discrete Fourier Transform*) pero, en vez de utilizar exponenciales complejas, la DCT usa cosenos. Se trata de una función lineal de números reales sobre números reales. Básicamente, los píxeles adyacentes de una imagen tienden a estar altamente correlacionados. Es esta característica la que se encarga de explotar al máximo la DCT.

Concretamente, para el procesamiento de imagen, se necesitan las transformadas de dos dimensiones. Así pues, para frecuencias horizontales, se buscan todas las posibles frecuencias verticales.

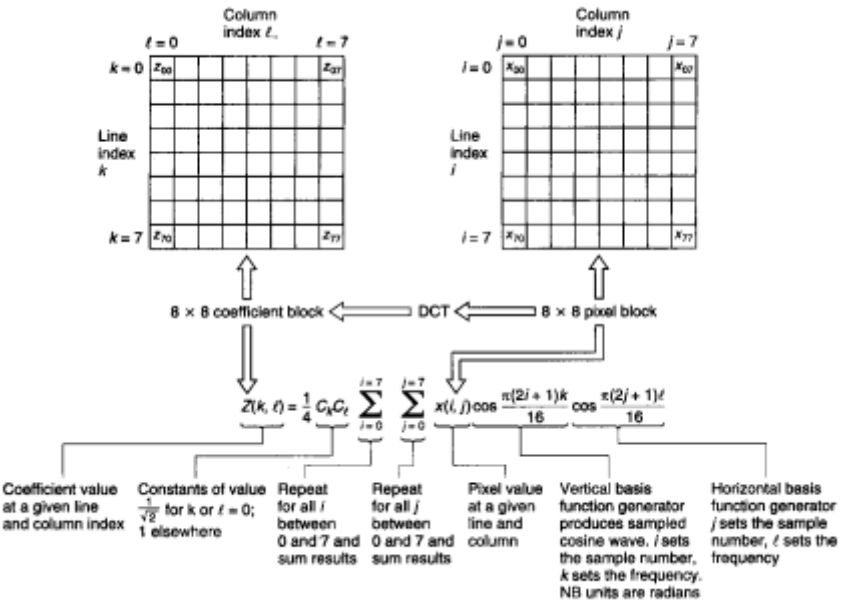


FIG. 2-8 DCT

La DCT de dos dimensiones ("*Fig. 2-8*") se puede obtener operando por separado con cada una de las dos dimensiones.

Además tiene otra serie de características que son de gran utilidad en la compresión de imágenes, tema que se trata aquí.

- La transformación llevada a cabo no depende de los datos recibidos, funcionando sin variación alguna para cualquier tipo de datos recibidos.
- El hecho de poder analizar los componentes en el dominio frecuencial, permite aprovechar al máximo las técnicas de compresión.
- Logra una compactación de la energía de los errores de predicción de manera muy eficiente.
- Se puede calcular dicha transformada de una manera rápida, pues existen algoritmos que lo permiten, produciendo pocos errores en los bloques de la imagen. Algunos de estos algoritmos los podemos ver en [5].

- Tras aplicar la DCT, los coeficientes resultantes están muy decorrelados (lo contrario que antes de realizar la DCT), permitiendo una fácil codificación de estos coeficientes.

En la codificación intra, es decir, la que usa la redundancia espacial dentro del mismo cuadro, se utilizan algunos métodos para aprovechar esta redundancia. El método más importante es el uso de la transformada DCT, usando bloques de 8x8. Esto se utiliza desde MPEG-1, que resultó en diversos avances y mejoras de dicha transformada, como bien se ve en [4], donde se habla una nueva transformada (Int-DCT – Integer DCT), la cual es una aproximación entera ortogonal a la clásica DCT.

2.2.6 PERFILES Y NIVELES (PROFILES AND LEVELS)

¿Qué queremos decir con perfiles y niveles de H.264? ¿Para qué se usan?

El gran motivo que ha inspirado la creación de diferentes tipos de perfiles y niveles en este estándar es que H.264 (desde MPEG-2) abarca muchas capacidades y permite al usuario distintos niveles de restricción a la hora de codificar. Así pues, para una orientación más sencilla y una organización adecuada, todas estas capacidades que ofrece se dividieron en perfiles y niveles, que pasamos a explicar a continuación:

- H.264 proporciona, al igual que otros estándares (incluidos sus predecesores, exceptuando MPEG-1), sus capacidades en forma de perfiles y niveles. Los principales puntos de distinción de los perfiles son las características algorítmicas y acerca de los niveles son las clases de rendimiento.

- Hay siete perfiles en H.264, teniendo cada perfil un tipo concreto de aplicaciones. Así, los siete perfiles son:

  - Baseline Profile: Enfocado principalmente a aplicaciones de bajo coste que requieren robustez frente a la pérdida de datos. Generalmente para aplicaciones móviles y de videoconferencia.

  - Main Profile: Encaminado a aplicaciones de almacenamiento y broadcast. Decae la importancia de este perfil cuando aparece el “High Profile”.

  - Extended Profile: Perfil para vídeo streaming. Tiene capacidades para alta compresión y algunas características adicionales para la robustez frente a errores de datos.

  - High Profile: Principal perfil para el almacenamiento de disco y broadcast y para aplicaciones en televisiones de alta definición (HD). Por ejemplo, es el perfil adoptado para aplicaciones de almacenamiento en discos Blu-Ray.

  - High 10 Profile: añade soporte al “High Profile” para 10 bits por muestra en la precisión de imágenes decodificadas.

  - High 4:2:2 Profile: Para aplicaciones profesionales que usan vídeo entrelazado. Añade características nuevas al “High 10 Profile” soportando el formato 4:2:2 de las submuestras de crominancia.

  - High 4:4:4 Predictive Profile: Mejora del “High 4:2:2 Profile” soporta el muestreo de crominancia en formato 4:4:4 con 14 bits por muestra, y codifica de cada imagen como tres planos de color diferentes.

- De estos, los 3 perfiles más utilizados son:

  - Baseline Profile

    - Slices de tipo I/P

    - Codificación entrópica CAVLC

  (Estas dos características son comunes a “Main Profile” y a “Extended Profile”)

- Grupos de Slice y ASO (Arbitrary Slice Ordering - orden de grupos de slice arbitrario)

(Común a "Extended Profile")

➢ Main Profile

Añade a las dos primeras características de "Baseline Profile":

- Slices tipo B
- Weighted prediction
- Codificación entrópica CABAC
- Codificación entrelazada.

➢ Extended Profile

Incluye todo el "Baseline Profile" y las dos primeras características del "Main Profile", además de:

- Slices tipo SP y SI (como explicaremos más adelante en el apartado 3.1.2).
- Partición de datos

- Los niveles hacen referencia a la frecuencia de bits y de codificación en macrobloques por segundo para todo tipo de resoluciones, desde QCIF hasta HDTV. Como se ve en la figura "*Fig. 2-9*", una mayor resolución se traduce en un mayor nivel. Con un menor nivel lo que hacemos es limitar ancho de banda, memoria y exigir menos requisitos de rendimiento.

Level Number	Typical Picture Size	Typical frame rate	Maximum compressed bit rate (for VCL) in Non-FRExt profiles	Maximum number of reference frames for typical picture size
1	QCIF	15	64 kbps	4
1b	QCIF	15	128 kbps	4
1.1	CIF or QCIF	7.5 (CIF) / 30 (QCIF)	192 kbps	2 (CIF) / 9 (QCIF)
1.2	CIF	15	384 kbps	6
1.3	CIF	30	768 kbps	6
2	CIF	30	2 Mbps	6
2.1	HHR (480i or 576i)	30 / 25	4 Mbps	6
2.2	SD	15	4 Mbps	5
3	SD	30 / 25	10 Mbps	5
3.1	1280x720p	30	14 Mbps	5
3.2	1280x720p	60	20 Mbps	4
4	HD Formats (720p or 1080i)	60p / 30i	20 Mbps	4
4.1	HD Formats (720p or 1080i)	60p / 30i	50 Mbps	4
4.2	1920x1080p	60p	50 Mbps	4
5	2kx1k	72	135 Mbps	5
5.1	2kx1k or 4kx2k	120 / 30	240 Mbps	5

FIG. 2-9 NIVELES MPEG

2.3 APLICACIONES DE H.264

El códec de vídeo de H.264 tiene un amplio rango de aplicaciones que cubren todas las formas de compresión de vídeo digital, desde tasas bajas para aplicaciones de *streaming* en Internet, a aplicaciones de *broadcast* en HDTV, permitiéndonos una compresión de hasta el 50% respecto a estándares anteriores, como podemos ver en "H.264 Joint Video Surveillance Group Compression Research Data: 2008" [6].

Hay que reseñar que MPEG-2 ocupaba 15-20 Mbps para transmitir una calidad aceptable de un vídeo de Alta Definición (HD) mediante broadcast o DVD. En H.264 la ocupación es de 8 Mbps, permitiendo utilizar ese ancho de banda ganado en transmitir más canales o en mejorar la calidad de vídeo de la propia emisión. Pero mayor importancia cobra el hecho de que una película en HD puede grabarse en un DVD convencional gracias a esta reducción, evitando así la necesidad de adoptar un formato de DVD de una mayor densidad.

El formato de Blu-Ray, cada vez con mayor aceptación en el mercado, ya incluye el H.264 High Profile como uno de los 3 códecs de vídeo por defecto. Sony también incluye este formato en sus dispositivos [7]. De la misma manera lo hace Panasonic [8].

FIG. 2-10 LOGOS COMPAÑÍAS DESTACADAS

A partir de 2004, DVB (Digital Video Broadcasting) aprobó el uso de H.264/AVC para la emisión en broadcast de televisión, así como lo hizo ATSC en 2008 (Advanced Television System Committee) [9].

Las redes móviles 3G presentan una serie de retos tecnológicos que conducen directamente a varias características de H.264. Las aplicaciones incluyen videoconferencia, *streaming* de vídeo bajo demanda, servicios de mensajes multimedia y broadcast de baja resolución. Algunas de estas características son importantes en aplicaciones de vídeo por los siguientes motivos:

- Para aplicaciones de vídeo, las retransmisiones para paquetes perdidos o con retraso son impracticables; así algunas características de H.264 permiten afrontar estos problemas, tales como: FMO (Flexible Macroblock Ordering), Data Partitioning, etc.

- Los nuevos tipos de Slices de H.264 (los slices SP y SI) permiten un cambio más dinámico entre múltiples streams para acomodar la variabilidad del ancho de banda.

- Tendencia del despliegue de 3G a empezar con H.263 y trasladarse a H.264 cuando éste madurara, creciera. Las redes 3G sólo permitían 57'6 kbps inicialmente. A

medida que estas tasas se incrementaran, móviles y redes tendrán que trasladarse (ya lo están haciendo) a H.264, que ofrece 2 veces el rendimiento de H.263, y resultará en una reducción a la mitad de tasa de transmisión para transmitir la misma calidad de imagen.

2.4 ANÁLISIS DE VÍDEO EN DOMINIOS TRANSFORMADOS

Desde los inicios del estándar de compresión MPEG se trabaja en el dominio comprimido. Como ya se ha comentado, el hecho de trabajar en dicho dominio permite disminuir la complejidad, el coste computacional y los costes de almacenamiento del sistema sin sacrificar la calidad de las imágenes al evitar una continua decodificación y codificación en pasos intermedios. Sin embargo, existen otros motivos que nos llevan a estar interesados en el dominio comprimido:

- Hoy en día, la mayoría del contenido multimedia disponible está en el formato comprimido. Usar directamente las características en dominio comprimido hace posible trabajar eficientemente en análisis e indexación de vídeo en tiempo real.

- Algunas características, como la información de movimiento, son más fáciles de extraer en este dominio comprimido, sin suponer un coste computacional adicional. No olvidemos que el dominio descomprimido seguirá disponible y resultará en una precisión incluso mayor en la extracción de datos, aunque será a costa de unos costes computacionales desproporcionados.

Las características a analizar en un vídeo en el dominio comprimido, las podemos englobar en cuatro grandes grupos: espacial, movimiento, codificación y audio. El objetivo de este apartado es comentar brevemente las técnicas y métodos de cada una de estas características, en los dominios comprimidos de MPEG-1 y MPEG-2, predecesores de H.264.

Esta conocida como “era digital” trae consigo el desarrollo de una amplia variedad de contenidos multimedia con una altísima calidad. Dichos contenidos implican una enorme cantidad de datos, con su consecuente complejidad de distribución. Esto propicia errores en el acceso, dificultades de manejabilidad y necesidad de recuperación de errores. Por ello desde el principio se han propuesto métodos y técnicas para hacer frente a cada uno de estos problemas.

Mucho trabajo de investigación se ha centrado en analizar las características de vídeo y audio, así como los métodos de extracción y sus aplicaciones en varios dominios. Dado el objetivo de este proyecto, nos interesan especialmente los estudios que se centran en la las características de vídeo. Diferentes investigadores han estado trabajando en el resumen y síntesis de estos métodos de extracción y aplicaciones encaminadas a un uso eficiente del dominio comprimido. Otros ejemplos se pueden ver en [10] y [11].

A continuación veremos varias de estas características extraíbles en el dominio comprimido y sus métodos. Se pueden dividir en tres grandes grupos: espaciales, de movimiento y de codificación. Ni que decir tiene que también hay mucho trabajo en parámetros extraíbles y sus respectivos métodos en temas de audio (el cuarto grupo de los mencionados anteriormente), que no trataremos en el presente documento por no abarcar el tema del que trata.

2.4.1 CARACTERÍSTICAS ESPACIALES

En este apartado vamos a ver las características espaciales que se pueden extraer de un vídeo comprimido en MPEG. Dichas características pueden referirse a *frames* por separado o a una secuencia de ellos.

- **Imagen DC:** MPEG incluye una imagen idéntica a la original pero con menor resolución, ocupando por tanto menos espacio. Esta imagen DC mantiene el contenido clave del frame, lo que nos permite extraer la misma información que obteníamos de la original, convirtiéndose así en un parámetro muy eficiente para la extracción de características visuales. Todos los coeficientes DCT, incluido el valor DC, se pueden conseguir fácilmente de la imagen DC para los *frames* I de una secuencia, pero no tan fácilmente para secuencias P y B, dado que se transforma y codifica el error después de la compensación de movimiento, siendo esto lo que finalmente se transmite. Se han publicado muchos trabajos con diferentes algoritmos para reconstruir lo más eficientemente posible esta imagen DC, especialmente utilizando la particularidad de que la transformada DCT es lineal. Hay que tener en cuenta que dada su característica de filtrado paso bajo, la imagen DC a veces da resultados más robustos en la extracción de algunos parámetros concretos que en otros.

  Algunos trabajos a destacar sobre la imagen DC en dominio comprimido son el de Chang y Messerschmit [12] y el de Yeo y Lui [13], que buscan algoritmos eficientes para extraer de *frames* P y B una imagen DC. Otros trabajos destacables son los orientados a la utilidad de esta imagen DC, como por ejemplo para la detección de cambios de toma, de Chen et al. [14] o el de Yeo y Lui [15].

- **Color:** Una de las posibles aplicaciones de la imagen DC anteriormente presentada, es la de extraer las características de color en una secuencia de imágenes de un vídeo. El color de una imagen puede venir dado en varios formatos (RGB, YUV o YCbCr) aunque MPEG siempre los convertirá a YCbCr. Trabajar con la información de color de un vídeo permite identificar un cambio de toma, observando diferencias importantes entre histogramas de color de *frames* consecutivos. El estudio de histogramas de intensidad, niveles de crominancia, etc., para distintas aplicaciones aparece en trabajos como Tan et al. [16] o Won et al. [17], ambos en 1999.

- **Texturas y bordes:** Hallar las diferentes texturas y bordes que aparecen en los diferentes *frames* de un vídeo requiere un procesamiento a nivel de píxel. No es posible su extracción en el dominio comprimido sin una importante decodificación. Sin embargo, la información sobre texturas y bordes de una imagen corresponde a los coeficientes DCT de media-alta frecuencia. De esta manera, si estudiamos los componentes de frecuencia adecuados, podremos sacar valiosa información sobre estos parámetros. Encontramos así algunos trabajos que utilizan estos coeficientes DCT para diferentes aplicaciones. Algunos ejemplos son Song y Ra en 1999 [18] y Shen y Sethi en 1996 [19].

A continuación veremos algunas posibles aplicaciones que se pueden dar a partir de la extracción de los coeficientes DCT de los que venimos hablando:

- **Correlación entre coeficientes DCT de dos *frames*:** Se utiliza para la detección de cambios de toma. Un trabajo sobre este tema lo encontramos en el de Arman et al. en 1993 [20].

- **Diferencia de bloques de la DCT:** Se compara la diferencia relativa de todos los coeficientes en un bloque DCT. Se utiliza para medir la semejanza entre dos bloques DCT. Este método requiere menor gasto computacional que el de la correlación entre dichos coeficientes. Aparece en el trabajo de Zhang et al. de 1995 [21].

- **Variación de los coeficientes DCT DC:** Se mide la intensidad del nivel de grises en *frames* I y P. Se pueden detectar transiciones graduales de cambios de toma. En esto han trabajo Meng et al. en 1995 [22].

2.4.2 CARACTERÍSTICAS DE MOVIMIENTO

La información de movimiento en las distintas imágenes de un vídeo es capturada en forma de vectores de movimiento. En general no representan el movimiento exacto de un bloque, por lo que hay que tener especial cuidado a la hora de usarlos. Los vectores de movimiento son más sensibles al ruido a la hora de calcular la magnitud del movimiento producido que a la hora de calcular la dirección del movimiento aunque estos vectores de movimiento son procesados por el Códec para minimizar los efectos del ruido. Infinidad de métodos han surgido en relación a los vectores de movimiento desde los comienzos de MPEG. A continuación ponemos algunos ejemplos de la información que se puede extraer a partir de los vectores de movimiento:

- **Movimiento global:** Consiste en separar el *frame* en 4 o 16 cuadrantes, cada uno con una característica de movimiento compuesta por magnitud y dirección. Cada cuadrante se corresponde con un número concreto de macrobloques, para una zona específica de la imagen. La magnitud vendrá dada por el valor medio de las magnitudes de los vectores de movimiento pertenecientes a los macrobloques del cuadrante en el que estemos trabajando. Para la dirección se utiliza un valor medio o un vector de movimiento dominante (el valor más alto de los que aparecen en dicho cuadrante). Destacar sobre este tema el trabajo de Ardizzone et al. de 1996 [23].

- **Segmentación basada en movimiento:** Los métodos que estudian la segmentación se basan en movimiento de objetos o personas en escenas de background fijo u homogéneo. Hay diferentes trabajos que optimizan los vectores de movimiento tras su extracción y así mejorar sustancialmente el resultado de su análisis para este fin. Por ejemplo, el trabajo de Eng and Ma de 1999 [24].

- **Análisis de movimiento a nivel de bloque:** Busca utilizar los vectores de movimiento de *frames* P y B para aproximar el movimiento de los macrobloques. Los diferentes métodos son desarrollados a bajo, medio o alto nivel, utilizando técnicas agrupamiento/*clustering* en los dos últimos niveles. Acerca de esto, encontramos trabajos como los de Kobla et al. de 1997 [25].

- **Movimiento acumulado:** Ha sido utilizado en muchos ámbitos, como por ejemplo para detectar eventos importantes en aplicaciones específicas como secuencias de vídeos deportivos. Aquí aparecen algunas investigaciones como las de Saur et al. de 1997 [26], que utilizaron este movimiento acumulado para detectar quiebros rápidos en partidos de baloncesto.

- **Operaciones comunes de la cámara:** Tras la aparición del vídeo, se convirtió en un asunto urgente la estandarización de todo lo referente al trabajo con cámaras. Era necesario definir de manera universal algunos conceptos como *zooms*, traslaciones, *booms*, etc., para trabajar más fácilmente, con bases comunes y entender el trabajo de otras personas. Estos términos definían movimientos de personas u objetos en las diferentes escenas teniendo en cuenta los diferentes movimientos de la cámara. Si se trata de una cámara en movimiento, nos interesa poder obviar dicho movimiento de la cámara y trabajar como si no se hubiese movido dicha cámara, dado que el movimiento que nos interesa es el

general de los elementos de la imagen. Trabajos referidos a este tema incluyen el de Akutsu et al. [27] de 1992.

- **Estimación de movimiento de la cámara:** Para detectar el movimiento de la cámara, se propusieron varios métodos. Algunos de ellos han utilizado directamente los vectores de movimiento de los diferentes macrobloques, como Zhang et al. en 1995 [21], Akutsu et al. en 1992 [27] o Kobla et al. en 1996 [28]. Otros trabajos se basan en modelos físicos en el espacio tridimensional, como los de Tan et al. en 1995 [29] o Tse and Baker de 1991 [30].

- **Acumulación de actividad de movimiento en *frames*:** Usa una media de la magnitud de los vectores de movimiento. Este valor medio se usa para umbralizar todos los vectores de movimiento del *frame*, analizando el número y longitud de las carreras de ceros existentes en la codificación. Esta información permite tener una buena idea acerca del tamaño, forma y número de objetos en movimiento presentes en la escena. Con este método es fácil distinguir *frames* en los que aparece un objeto grande moviéndose, frente a *frames* con varios objetos pequeños moviéndose. Acerca de esto trabajaron Divakaran and Sun en el año 2000 [31].

- **Histograma de movimiento:** Representación compacta de movimiento global o por regiones en un *frame*. Kobla et al. [28] y Ardizzone et al. [32] han usado histogramas de vectores de movimiento para trabajar en el enfoque y otras características de la cámara.

2.4.3 CARACTERÍSTICAS DE CODIFICACIÓN

Características muy útiles debido a su simpleza de extracción. Puesto que MPEG estandariza solamente la decodificación, las características de codificación están sujetas únicamente al codificador que se esté usando en la aplicación en concreto. Existen distintas técnicas que utilizan este tipo de parámetros:

- **Tipo de macrobloque:** Como ya hemos mencionado, existen tres tipos de *frame*: I, P y B. Cada tipo de *frame* tiene unos tipos de macrobloque concretos. Por ejemplo, los *frames* P tienen macrobloques con predicción *forward*, macrobloques intra-codificados, y macrobloques skip, añadiéndose la posibilidad de macrobloques con predicción *backward* y bidireccional para los *frames* B. Se puede obtener valiosa información analizando el número de cada tipo de macrobloques, así como las proporciones entre ellos, muy importante y fiable para aplicaciones como cambios de toma [28], o para detectar la presencia o no de una repetición a cámara lenta en un vídeo MPEG [33]. Estos dos trabajos son de Kobla et al.

  Vemos que hay muchos trabajos sobre los cambios de toma, cada uno utilizando los parámetros de una manera u otra (esto se mantiene hoy en día en el estándar H.264 como veremos en capítulos posteriores).

- **Tasa de transmisión:** Este parámetro también resulta útil para detectar cambios de toma. El primer *frame* de una nueva escena tendrá gran cantidad de macrobloques intra-codificados, los cuales ocupan más, provocando de esta manera que el primer *frame* de una nueva escena tenga una tasa de transmisión mucho mayor que los *frames* anteriores. Diversos estudios se han realizado sobre este parámetro, como el de Feng et al. en 1996 [34]. Además, este parámetro se puede utilizar en combinación con otros métodos, como hacen Divakaran et al. en 1999 [35] y Boccignone et al. en 2000 [36], logrando una detección de cambio de toma de manera más robusta utilizándolo de forma conjunta con la imagen DC para una detección de cambio de toma de una manera más robusta.

3 ANÁLISIS INICIAL COMPARATIVO

3.1 CAMBIOS EN H.264

3.1.1 INTRODUCCIÓN

En este apartado se presentan las principales diferencias que aparecen en el estándar H.264 con respecto a sus predecesores MPEG-1 y MPEG-2. Aunque los cambios son cuantiosos, en general se enfocan a una mayor flexibilidad para el usuario, la posibilidad de elegir más opciones de configuración, y lograr una mejora importante en términos de compresión. Cabe destacar también que, debido a esta mejora en la tasa de transmisión, algunas técnicas de análisis vistas en el capítulo anterior dejan de tener utilidad, como por ejemplo observar los cambios de la tasa de transmisión de los bloques para detectar un cambio de toma. Con el nuevo estándar esto no es tan evidente, ya que las diferencias, incluso en cambios de toma, son mucho más uniformes.

FIG. 3-1 COMPARACIÓN CALIDAD DE VÍDEO

H.264 hace uso tanto de la codificación temporal como de la espacial de cara a mejorar la eficiencia en la codificación de vídeo. La posibilidad de controlar la tasa de transferencia con H.264, permite enviar la información de manera flexible de tal manera que se pueda adaptar a los distintos dispositivos que la reciban. Por ello da la posibilidad de ofrecer vídeo de alta calidad a un amplio rango de dispositivos, desde telefonía móvil hasta dispositivos Blu-Ray (actualmente con las mayores prestaciones en calidad de vídeo). Ésta es una de las grandes razones por las que H.264 está sustituyendo a los estándares actuales de compresión de vídeo.

De forma general, la compresión en el ámbito del vídeo digital busca el ahorro de bits. Los códec (COdificador-DECoficiador) se encargan de la codificación y decodificación del stream de bits. Gracias a una mejora sustancial de los estándares de compresión en los que se basan dichos códecs se consigue transmitir vídeo de mayor calidad usando la misma cantidad de ancho de banda. A partir de la implementación de esta mejora en H.264, se logra reducir los datos necesarios para poder reproducir un vídeo. Además, los codificadores se encargan de

procesar cada *frame* (fotograma) y a su vez cada bloque dentro de cada *frame*. Este procesamiento se puede aprovechar para implementar una *estimación de movimiento*, buscando al mismo tiempo similitudes (sobre todo en relación a la textura) en *frames* anteriores o posteriores a cada bloque. Si encuentra una referencia buena, solo será necesario codificar el correspondiente *vector de movimiento*, que apuntará al bloque del *frame* correspondiente donde la similitud sea importante. Si no se logra una referencia adecuada con respecto a *frames* cercanos al que se está codificando, se buscarán semejanzas con bloques ya codificados en el mismo *frame* con anterioridad. Aunque esta posibilidad es más costosa que la de estimación de movimiento, se trata de una forma mucho más eficaz que codificar la textura directamente. El objetivo de estos codificadores no es reproducir la secuencia de vídeo con pérdidas nulas, sino con que las pérdidas sean aceptables y permitan una visualización de la secuencia de vídeo lo más similar posible a la secuencia original. Aquí aparece el compromiso entre compresión y calidad de vídeo con el que H.264, tras sus mejoras, permite trabajar. Se trata de encontrar los medios adecuados para que la calidad visual no se pierda pero conseguir la mayor compresión posible. Eligiendo los valores adecuados para los distintos parámetros que H.264 permite adaptar, se pueden llegar a compresiones de 100:1 con diferencias prácticamente imperceptibles.

En los subapartados siguientes se presentan los cambios más reseñables que se han introducido en H.264 respecto a los estándares anteriores de la familia MPEG.

3.1.2 SLICES

Toda imagen tiene un proceso de codificación. Esta imagen codificada tiene un número de *frame*, pero no tiene por qué ser el mismo que el orden de reproducción.

En inter-predicción se usan imágenes codificadas previamente, las cuales están almacenadas en un buffer (DPB – Decoded Picture Buffer). Se puede acceder a ellas a partir de dos listas: lista 0 y lista 1. Estas listas sirven para referenciar a *frames* codificados previamente, los cuales no tienen por qué ser *frames* anteriores en el tiempo (en el sentido de la presentación de los mismos). Así pues, un *frame* B se deberá codificar después del *frame* P al que esté referenciado aunque sea con predicción *backward* (predicción hacia *frames* posteriores). En la lista 0 se referencian a *frames de reproducción* anterior al actual (*forward*) y en la lista 1 se referencian a *frames* de reproducción posterior al actual (*backward*), como norma general en la práctica (en la teoría ambas listas referencian a frames tanto anteriores como posteriores).

Como vemos en "*Fig. 3-2*", un *frame* puede dividirse en varios *slices, a*unque en la práctica, un *frame* casi siempre está compuesto por un único *slice*. Un *slice* es un conjunto de macrobloques que se procesa en orden de escaneado, es decir, de izquierda a derecha y de arriba a abajo. Un *slice* se puede decodificar de forma independiente.

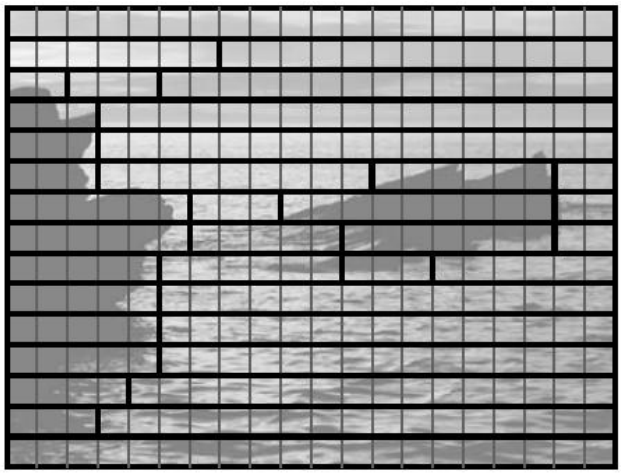


FIG. 3-2 SLICES

- Los *slices* de tipo I contienen solo macrobloques de tipo I.
- Los *slices* tipo P tienen macrobloques tipo I y P.
- Los *slices* tipo B tienen macrobloques I y B.

La predicción de H.264 se produce a nivel de macrobloque:

- Macrobloques tipo I: generalmente usan predicción intra a partir de las muestras decodificadas correspondientes del propio *slice*. La predicción se crea para el macrobloque completo o bien para cada bloque de 4x4 de las muestras de luminancia en el macrobloque. Una alternativa a la predicción intra, es la predicción I_PCM, que permite al codificador transmitir los valores de la imagen directamente, sin predicción ni transformada. En casos especiales (como imágenes anómalas o parámetros muy

bajos de cuantificación), este modo puede ser más eficiente que el procedimiento usual de intra predicción, transformada, cuantificación y codificación entrópica.

- Los macrobloques tipo P y B usan predicción inter entre *frames* próximos en el eje temporal, usando las listas de referencia 0 y 1. Una novedad con respecto a estándares anteriores es que se permite referenciar a más de un *frame* dentro del mismo *frame* que se está codificando (como se explica en el siguiente subapartado "3.1.3-Codificación Temporal").

Otra mejora que se introduce en H.264 es la aparición de dos nuevos tipos de slices: SP y SI. Estos tipos de *slices* se utilizan en el perfil extendido de H.264, conocido como "Extended Profile". El uso de dichos *slices* permite un flujo eficiente entre *streams* de vídeo y un acceso aleatorio eficiente para los decodificadores de vídeo. Por ejemplo, un vídeo a transmitir es codificado a diferentes tasas de transmisión y enviado a través de Internet; el decodificador siempre intentará decodificar el *stream* de vídeo a la mayor tasa posible siempre, con la condición de poder cambiar (switch) automáticamente a una tasa menor si la tasa de transmisión disminuye.

- Los *slices SP* facilitan la transición entre flujos o *streams* codificados. Contiene macrobloques P y/o I y están diseñados para soportar cambios entre secuencias codificadas similares

- Los *slices SI* son similares a los *slices* SP pero contienen macrobloques SI (un tipo especial intra-*frame*). Han sido ideados para conmutar de manera eficaz entre diferentes flujos de vídeo para un acceso aleatorio en los decodificadores de vídeo más eficiente.

Para pasar de una secuencia dada a otra equivalente pero codificada de diferente manera, se deben cumplir una serie de requerimientos. Si los *frames* de dichas secuencias son *frames* P, para pasar del segundo frame codificado de la secuencia A, A_1 (recordamos que el subíndice siempre corresponde a un frame mayor dado que la numeración comienza por cero) al tercer frame codificado de la secuencia B, B_2 (tercer *frame* codificado de la secuencia B) ("*Fig. 3-3")* no sería posible una conversión inmediata pues los *frames* de la secuencia B se referencian a los *frames* anteriores de la propia secuencia, y lo mismo ocurre con los de la secuencia A. De este modo, no se podría obtener B_2 a partir de A_0 y A_1. La solución más sencilla, aunque la más costosa, es la de intercalar *frames* I en las secuencias de manera periódica para crear puntos de acceso, por donde se pueden producir cambios (*switching points).* Sin embargo, esto conlleva picos en la tasa de transmisión, que siempre son poco deseables. Aquí se refleja la ventaja del uso de slices SP y SI, permitiendo una mayor eficiencia en términos de codificación y un cambio de secuencia de una manera sencilla, pues usan predicción de compensación de movimiento. La clave es el Slice SP AB_2 ("*Fig. 3-4"*).

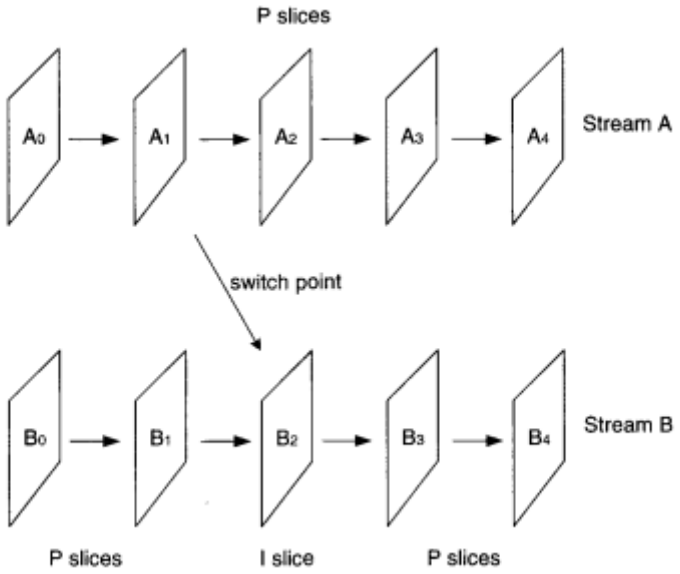


FIG. 3-3 PUNTOS DE ACCESO USANDO SLICES

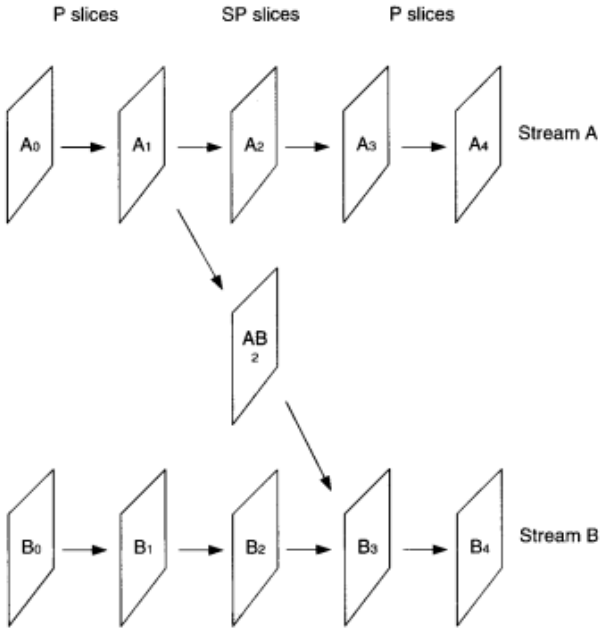


FIG. 3-4 PUNTOS DE ACCESO USANDO SLICES SP

Un análisis de la utilización de los slices SP y SI y sus ventajas en términos de tasa de transmisión, se puede encontrar en diferentes trabajos. En [37] se analiza de manera tanto teórica como experimental el uso de este tipo de slices.

3.1.3 CODIFICACIÓN TEMPORAL

También llamada codificación inter-frame. H.264 permiten nuevas particiones de los bloques, desde 16x16 hasta 4x4. Una mayor variabilidad de las particiones permite obtener más información acerca del vídeo a analizar y una mayor exactitud en la estimación de movimiento. También cabe destacar que este estándar permite una precisión de hasta ¼ de píxel, la cual era de ½ en estándares previos.

Una de las claves del éxito de H.264 es la optimización de la cantidad de información residual. Con intención de disminuir al máximo dicha información, se asignan bloques de diferentes tamaños a la imagen, según la zona esté más o menos texturizada. Así, zonas en las que existe una mínima textura (zonas homogéneas de la imagen) tendrán macrobloques del mayor tamaño posible (16x16 píxeles). En caso contrario se pueden dividir estos macrobloques en sub-bloques de 16x8, 8x16, 8x8… que a su vez estos últimos pueden dividirse en 8x4, 4x8 y 4x4 como vemos en "*Fig. 3-5*". Esto permite optimizar la cantidad de información codificada, comprimiendo las zonas homogéneas (habitualmente background de un *frame*) y utilizando más bits, para las zonas con más información (áreas con resolución espacial baja), cuya predicción no sea inmediata debido, por ejemplo, a movimientos.

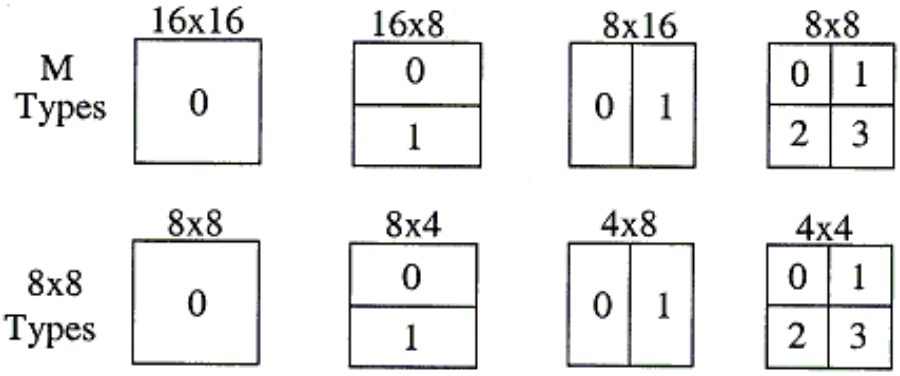


FIG. 3-5 PARTICIONES MACROBLOQUES

Particiones macrobloques H.264 mantiene una de las herramientas más utilizadas en estándares anteriores, los *vectores de movimiento*. La misión de dichos vectores es apuntar al bloque más parecido al que estamos codificando en ese momento. Esto permite codificar la diferencia entre ambos bloques, lo cual supone mucha menor información que enviar el bloque en sí. Puesto que en el decodificador también se cuenta con los *frames* previamente codificados, se puede repetir la misma predicción que en el codificador y, con los pasos inversos, reconstruir el *frame* de forma fiable.

Otra de las principales novedades de H.264 es la de la *"multirreferencia"*, que permite a bloques de un cierto cuadro referenciar a bloques de distintos cuadros. Anteriormente en MPEG-1 y 2, se podía utilizar como referencia de predicción al *frame* anterior (predicción *forward*) o al *frame* posterior (predicción *backward*) con respecto al *frame* que estaba siendo codificado, pero únicamente a los frames contiguos (inmediatamente anterior o posterior). Existían los *frames* y *slices* de tipo I, P y B, con predicciones intra, *forward* y *backward* y/o *forward* respectivamente. Por ejemplo, dentro de un *frame* tipo P, sus macrobloques P utilizarían predicción *forward* utilizando únicamente los bloques del frame anterior. La multirreferencia de H.264 permite que cada partición dentro de un macrobloque tenga su

propia predicción y su propio vector de movimiento y que dichos bloques (dentro del mismo macrobloque) puedan apuntar a distintos bloques de cualquier macrobloque de cualquier *frame* que esté en el DPB (*Decoded Picture Buffer*, donde se almacenan los *frames* que serán usados como referencia).

Es decir, que un macrobloque puede estar referenciado a un macrobloque de un *frame* determinado, y otro macrobloque dentro del mismo *frame* puede estar referenciado a otro macrobloque de un *frame* distinto al anterior. Esto otorga una mayor flexibilidad a la elección del bloque más parecido al bloque codificado, permitiendo una *estimación de movimiento* mucho más eficaz. Sin embargo, esta mejora propicia una mayor complejidad en términos de codificación.

3.1.4 CODIFICACIÓN ESPACIAL

También llamada codificación intra-frame. Como novedad respecto a MPEG-1 y 2, H.264 utiliza no sólo la diferencia con respecto a bloques previamente codificados, sino además la dirección de la predicción. Se trata de un proceso similar a la codificación inter-frame pero entre bloques del mismo cuadro. Sin embargo, al contrario que en inter, en codificación intra sólo se utilizan tamaños de macrobloque de 16x16 o 4x4 en luminancia. Existen distintos modos de predicción según sea el tamaño de dicho macrobloque: nueve modos de predicción para los bloques 4x4 y cuatro modos de predicción para los bloques 16x16, como vemos en la figuras "*Fig. 3-6*" y "*Fig. 3-7*" respectivamente:

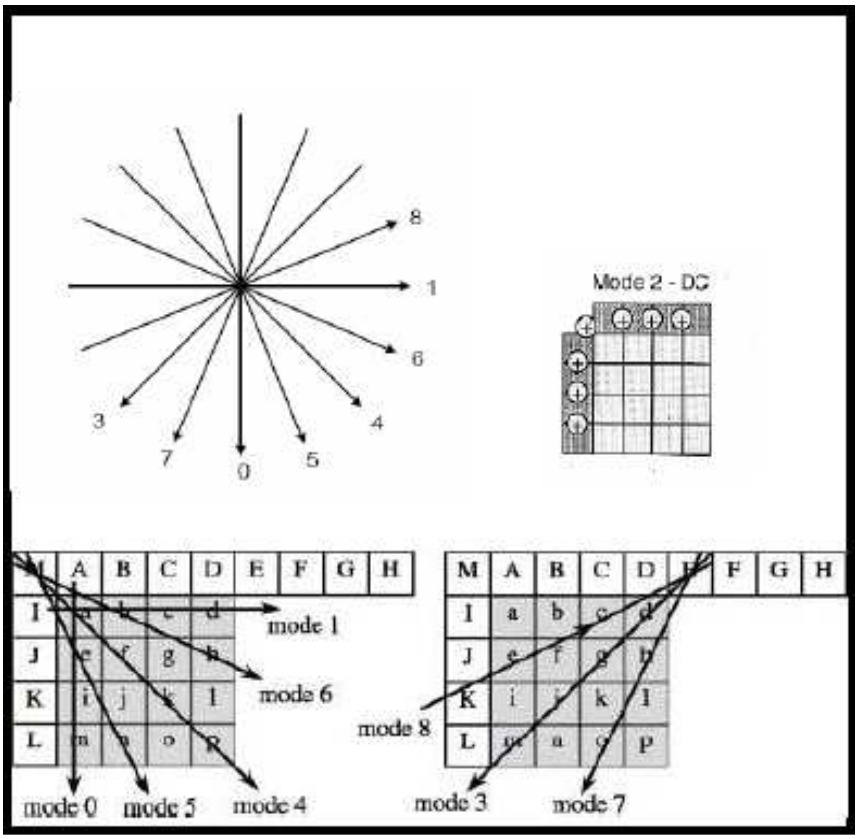


FIG. 3-6 MODOS DE PREDICCION MACROBLOQUES INTRA 4X4

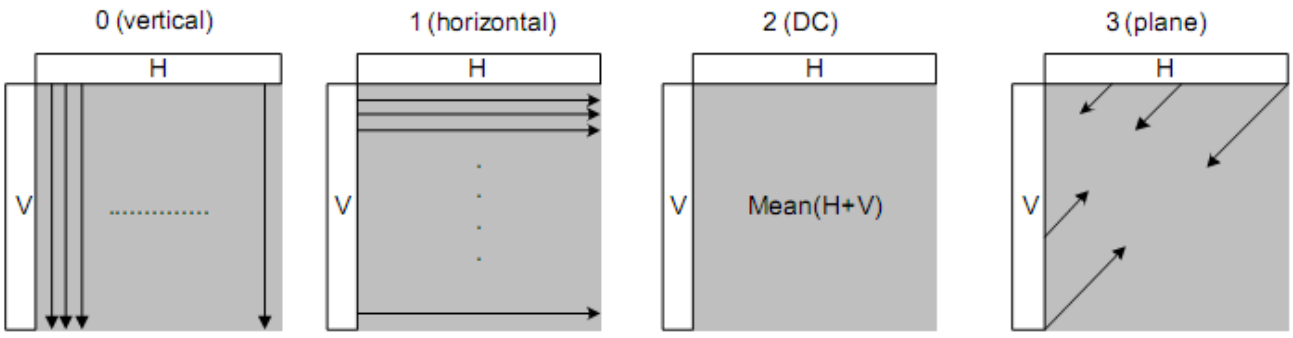


FIG. 3-7 MODOS DE PREDICCIÓN MACROBLOQUES INTRA 16X16

Los modos de predicción parten de bloques ya codificados, concretamente de los bloques que están situados encima y a la izquierda del que se está codificando en ese momento. Como vemos, hay varios modos de predecir el valor de un píxel a partir de los píxeles de los bloques anteriormente codificados. El objetivo es encontrar el mínimo SAE (*Sum of Absolute Errors*) resultante de aplicar los diferentes modos de predicción. El cálculo comienza con el bloque situado en la esquina superior izquierda, desplazándose hacia la parte inferior derecha de la imagen para calcular los siguientes bloques según el modo de predicción óptimo para cada caso. Siguiendo este recorrido se explican a continuación la aplicación de cada modo.

- El orden correcto de los nueve modos de predicción para los bloques 4x4 de luminancia se corresponde con el siguiente:

Modo 0 (Vertical): Las muestras superiores son extrapoladas verticalmente. Un ejemplo práctico de uno de los modos se puede seguir con la figura "*Fig. 3-6*". Para calcular el valor de los píxeles "a-p" con el modo 0, se deben extrapolar los valores de las muestras que están encima ("A-D") de manera vertical, es decir, en los píxeles 'a' (valor de 'A'), 'e' (valor de 'B'), 'i' (valor de 'C'), 'm' (valor de 'D') y también 'b', 'f', 'j', 'n' (otra vez con los valores de "A-D" respectivamente) y repetir el proceso en las cuatro columnas que aparecen.

Modo 1 (Horizontal): Las muestras situadas a la izquierda son extrapoladas horizontalmente.

Modo 2 (DC): Todas las muestras en el bloque resultante de la predicción ("a-p"), se predicen a partir de la media de las demás muestras superiores y a la izquierda.

Modo 3 (Diagonal inferior izquierda): Las muestras son interpoladas con 45 grados de ángulo entre las esquinas inferior izquierda y la esquina superior derecha.

Modo 4 (Diagonal inferior derecha): Las muestras son interpoladas con 45 grados de ángulo hacia abajo a la derecha.

Modo 5 (Vertical derecha): Extrapolación de aproximadamente 26.6 grados a la izquierda de la vertical.

Modo 6 (Horizontal abajo): Extrapolación de aproximadamente 26.6 grados debajo de la horizontal.

Modo 7 (Vertical izquierda): Extrapolación de aproximadamente 26.6 grados a la derecha de la vertical.

Modo 8 (Horizontal arriba): Extrapolación de aproximadamente 26.6 grados encima de la horizontal.

- Los cuatro modos de predicción para los bloques 16x16 de luminancia son:

Modo 0 (Vertical): Las muestras superiores son extrapoladas verticalmente.

Modo 1 (Horizontal): Las muestras situadas a la izquierda son extrapoladas horizontalmente.

Modo 2 (DC): Todas las muestras en el bloque ("a-p") resultante de la predicción, se predicen a partir de la media de las demás muestras superiores y a la izquierda.

Modo 3 (Plano): Una función lineal plana es aplicada a las muestras superiores y a la izquierda para extrapolar las muestras. Este modo es conveniente para trabajar en áreas de suave variación de luminancia.

Cada componente de crominancia 8x8 de un macrobloque intracodificado se predice a partir de muestras de crominancia superiores o a la izquierda de la anterior, ya codificadas, y ambas componentes de crominancia usan siempre el mismo modo de predicción. Los 4 modos de predicción son muy similares a los de luminancia de 16x16 vistos previamente, aunque en este caso cambia el orden de los modos. Además, en los modos de predicción para los componentes de crominancia aparecen *slices* tipo SI y SP, junto a los I, P y B de estándares anteriores, como ya se ha explicado.

3.1.5 TRANSFORMADA

En análisis de vídeo MPEG-1 y 2 se utilizan continuamente las imágenes DC, que recordamos son el resultado de aplicar la transformada discreta del coseno (DCT) a una imagen original, y almacenar posteriormente la parte continua de cada bloque o coeficiente DC. De esta forma se logra una imagen sub-muestreada de la original (imagen DC) muy útil en diversas aplicaciones ya mencionadas.

H.264 utiliza esta misma transformada pero aplicada sobre residuos, es decir, sobre las diferencias entre cada bloque y los contiguos. Esto implica que algunas técnicas que hacían uso de estas imágenes DC dejen de tener utilidad. Es importante señalar que, aunque se aplique sobre residuos, cada bloque sigue teniendo sus coeficientes DC y AC. El coeficiente DC ("*Direct Current*") es la frecuencia más baja y se corresponde con el valor medio. Los coeficientes AC ("Alternative Current") se corresponden con los detalles del bloque. A pesar de perder gran importancia práctica tras no poder obtener la imagen DC como réplica sub-muestreada de la original, los coeficientes DC y AC siguen siendo utilizados por el codificador y el decodificador.

La DCT convierte los datos de una imagen o un residuo de un dominio al de su transformada. Los datos en el nuevo dominio deben cumplir una serie de características:

- Decorrelados: con poca relación entre sí.
- Compactos: logran acumular la mayor parte de la energía en pocos valores.
- Reversibles: para permitir volver a trabajar en el dominio anterior.
- Algoritmo computacionalmente eficaz: utilizar un algoritmo que requiera pocas operaciones y limite la ocupación en memoria.

Cada muestra o píxel de una imagen tiene componentes de luminancia y crominancia, siendo la luminancia la componente que contiene la información de la cantidad e intensidad de brillo, y la crominancia la componente que tiene información sobre el color. Cada macrobloque, con sus correspondientes bloques, ha de estar transformado, cuantificado y codificado. En los estándares anteriores se utilizaba la DCT 8x8. En H.264 se utiliza la transformada Hadamard tanto para los coeficientes DC de luminancia como para los coeficientes DC de crominancia, y la transformada DCT para todos los demás bloques 4x4 de datos residuales.

El hecho de que se puedan utilizar bloques de diferentes tamaños (de 16x16 a 4x4) sólo implica pequeños cambios en las transformadas para cada uno de ellos.

En la "*Fig. 3-8*" se puede ver el orden de escaneo de los bloques residuales en un macrobloque. En primer lugar, el coeficiente DC, etiquetado como -1, de cada bloque 4x4 de luma. Luego, los bloques residuales del 0 al 15 se transmiten en el orden que aparece en la figura (de izquierda a derecha y de arriba a abajo). Los bloques 16 y 17 son los coeficientes DC de los bloques 2x2 de crominancia y los bloques 18-25 son los bloques residuales de crominancia.

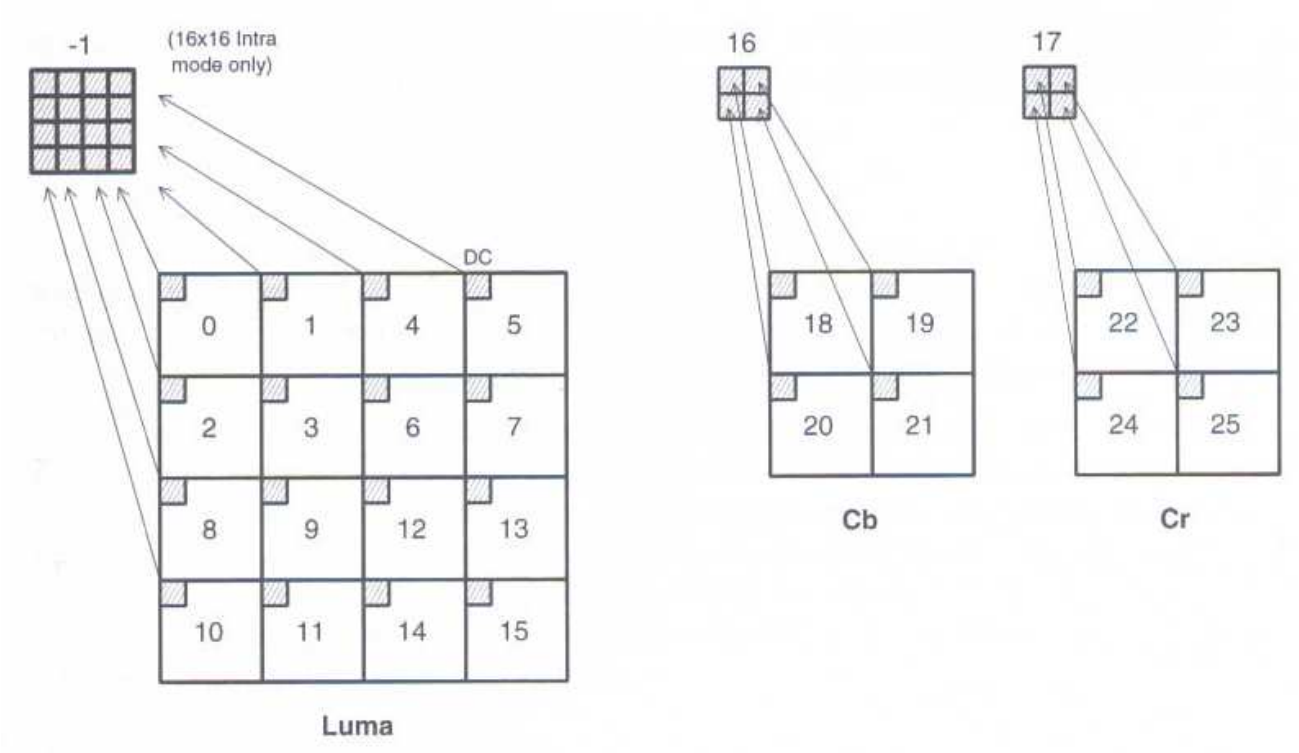


FIG. 3-8 ORDEN DE ESCANEO EN BLOQUES RESIDUALES DE MACROBLOQUES

3.2 CAMBIOS EN PARÁMETROS EXTRAÍBLES

Este apartado pretende resumir los cambios de H.264 acerca de los parámetros extraíbles con respecto a estándares anteriores. En estándares previos se utilizaban métodos no extrapolables al dominio comprimido. Entre estos métodos, destacaban: coeficientes DC, vectores de movimiento y tasa binaria. Procedemos a explicar un poco en qué consistían y por qué no es posible utilizar dichos métodos en el dominio comprimido de H.264:

- ***Coeficientes DC*****:** se obtenían aplicando la transformada discreta del coseno, DCT, y almacenando la parte continua de cada bloque.

Esta imagen DC representaba una imagen sub-muestreada de la original. En H.264 se aplica la DCT sobre residuos (diferencias entre bloques), ya que usa una codificación intra. Es por esto por lo que no podemos utilizar dicha imagen DC como imagen sub-muestreada de la original.

- ***Vectores de movimiento*****:** Los vectores de movimiento apuntan al área más similar al bloque que se quiere codificar de entre los cuadros previamente codificados. H.264 sigue utilizando los vectores de movimiento, pero permitiendo multirreferencia (apuntar a cuadros B como cuadros de referencia o a más de un cuadro dentro del cuadro que estemos codificando) y una precisión de ¼ de píxel. Adicionalmente, los vectores de movimiento no corresponden a tamaños fijos de macrobloques, variando tanto en *frames* I como en frames P ó B, como se ha expuesto previamente.

- ***Tasa binaria*****:** Uno de los principales objetivos del estándar H.264 es reducir la tasa de transmisión. La intra-codificación consigue gran parte de este trabajo, codificando únicamente la diferencia y la dirección en la que se efectuó la predicción en vez de todo el frame. Además, H.264 permite diferentes tamaños de macrobloque, lo que resulta en macrobloques de 16x16 para zonas más homogéneas, de 4x4 para zonas más texturizadas o tamaños intermedios en *frames* P ó B (como 8x4, 16x8…) para zonas mixtas o bordes. La disminución de la tasa de transmisión y su homogenización entre unos *frames* y otros complica operaciones como la detección de cambios de toma a través de variaciones bruscas de la tasa binaria.

Con la introducción de H.264, muchos trabajos destacables sobre los dominios de MPEG-1 y 2, como [31],[38],[39] entre otros, dejan de tener utilidad. Por ello, es necesario un estudio a fondo de los nuevos cambios en el dominio comprimido y una investigación que permita encontrar nuevos métodos y vías para igualar y mejorar los resultados obtenidos con los estándares previos.

4 SELECCIÓN DE PARÁMETROS EXTRAÍBLES

Al trabajar en el dominio comprimido aparecen aplicaciones que pueden utilizar más de un parámetro para realizar el mismo trabajo. Uno de los ejemplos más comunes son los cambios de toma, para lo cual existen diversos parámetros con información útil. Por este motivo, es preferible hablar en primer lugar de los requerimientos de las distintas aplicaciones, concluyendo esta sección con un breve resumen de los parámetros extraíbles que satisfacen estas necesidades. Dichos parámetros serán con los que se trabajará en el siguiente capítulo de resultados.

4.1 CAMBIOS DE TOMA

Una aplicación importante de la información extraíble en el dominio comprimido es la **detección de cambios de toma.** Esta aplicación es fundamental en el análisis de vídeo, pues generalmente un cambio de *background* permite separar unas tomas de otras, especialmente si dichos cambios son abruptos. Esto da la posibilidad de analizar vídeos de manera coherente, sin recurrir a información de imágenes referentes a otras tomas como, por ejemplo, vectores de movimientos, modos de predicción, etc.

Para la detección de cambios de toma en H.264, se utilizan diferentes métodos:

1. En [40] propusieron analizar los **tipos de macrobloque** que se utilizaban en los cuadros Intra, jugando con la nueva particularidad de H.264 que permite codificar dichos macrobloques con diferentes tamaños, 16x16 o 4x4, en función de la textura del cuadro. Será declarado un cambio de toma cuando el número de macrobloques que sufren un cambio en el tipo de codificación es lo suficientemente elevado. El punto débil de este algoritmo es precisamente la medición de cuánto es suficientemente elevado, ya que requiere que el umbral se fije empíricamente.

2. En [41] se define un nuevo concepto, el de **"Tipos de Predicción Temporal"** (TPT), en el cual se combinan dos de las nuevas características de H.264, como son los diferentes tipos de macrobloque y la dirección de los *frames* referenciados. Dependiendo del tipo de predicción de la partición de un macrobloque y de la dirección de los *frames*, la partición pertenecerá a uno de los siguientes cuatro tipos de predicción temporal:

   -Predicción temporal Intra: en caso de que se use la predicción Intra.

   -Predicción temporal *Forward*: en caso de que el número del *frame* actual sea posterior al de los *frames* referenciados.

   -Predicción temporal *Backward*: en caso de que el número del *frame* actual sea anterior al de los *frames* referenciados.

-Predicción temporal Bi-direccional: en caso de que la *frame* actual esté entre dos *frames* referenciadas.

Es importante resaltar que los cambios de toma no siempre ocurren en *frames* de tipo I. Existe un tipo de *frame* que se usa frecuentemente como punto de acceso aleatorio en un video, conocido como IDR. Estos *frames* pertenecen a los *frames* I que limpian el DPB (Decoded Picture Buffer, buffer donde se almacenan las imágenes ya decodificadas que se pueden usar como referencia) y, por tanto, impiden que *frames* posteriores se referencien a *frames* previos a dicho IDR.

Además, los cambios de toma pueden ocurrir en *frames* de tipo P o B en función de las características del codificador o del área de aplicación. Los macrobloques en *frames* P y B usan predicción temporal para aprovechar al máximo la similitud entre dos *frames* consecutivos en una escena. Para estos tipos, si estamos en el primer *frame* correspondiente a una nueva escena, éste va a tener poco que ver con los *frames* anteriores y, por tanto, deberían tener predicción intra y *backward* según lo que hemos visto. Por el contrario, el último *frame* de una secuencia, por el mismo razonamiento, tendrá predicción intra y *forward*. Así, si el porcentaje de predicciones intra y *forward* para el *frame* anterior y el porcentaje de predicciones intra y *backward* para el *frame* actual superan un cierto umbral, se puede decir que estamos en un cambio de toma.

Por otro lado, si el primer *frame* de una escena es de tipo I, no representará correlación temporal con *frames* anteriores ni posteriores. Aún así, si el *frame* anterior es de tipo P o B, la distribución de los TPT's de las particiones de los MB en ese *frame* puede proporcionar bastante información acerca de la relación entre dos *frames* consecutivos. Adicionalmente, si un *frame* I y otro *frame* I anterior pertenecen a distintas escenas, su contenido va a variar sustancialmente. Por tanto, es posible observar la distribución y cantidad de tipos de MB en dichos *frames* (que solo pueden ser 4x4 o 16x16 para *frames* I) y, si la diferencia es grande, asumir que puede tratarse de un cambio de escena. Sin embargo, esta característica no es concluyente, por lo que se recomienda dividir el *frame* en sub-bloques de, por ejemplo, 5x5 macrobloques y ver si el número de macrobloques que varían es muy grande entre el mismo sub-bloque del *frame* I anterior y el actual. A continuación es necesario repetir este procedimiento con todos los sub-bloques y decidir finalmente, mediante un umbral hallado de manera empírica, si la variación de sub-bloques es lo suficientemente grande como para considerar cambio de toma. Ya que esta característica no es concluyente, lo más recomendable es usarla junto a otras características como el tipo de predicción del *frame* anterior. Si a partir de la diferencia en la distribución y cantidad de los tipos de MB 4x4 y 16x16 en el *frame* I actual respecto al anterior se observa un posible cambio de toma, pero el *frame* B anterior tiene gran cantidad de MB con predicción *backward*, muy posiblemente no se trate de un cambio de toma.

Otro tipo de aproximación a la detección de cambio de toma es la presentada en III[42]. Si el primer *frame* de una nueva escena es de tipo I, es posible unir las dos formas de detectar un cambio de toma recién vistas. Esto implica que se observaría por un lado la diferencia en la cantidad y distribución de MB entre el *frame* I y el anterior I y, por otro lado, si el *frame* anterior (P o B) tiene un porcentaje alto de predicciones Intra y *forward* (pues supuestamente debería ser el último *frame* de la

escena anterior). Si se superan ambos umbrales podemos apostar que ese *frame* corresponde a un cambio de escena.

La mayor desventaja de este algoritmo es que no tiene en cuenta los cambios graduales de toma. Para su detección, el algoritmo debería ser ampliado, pues dichos cambios de toma abarcan varios *frames*, no ocurren de un *frame* al siguiente. Además, los umbrales utilizados se calculan empíricamente atendiendo a la Precisión y el *Recall, que* vienen definidas de la siguiente manera:

$$\mathrm{Re}\,call = \frac{Detects}{Detects + MDs} \qquad \mathrm{Pr}\,ecision = \frac{Detects}{Detects + FAs}$$

Siendo:

-Detects: detecciones correctas.

-MDs : Missed Detections, cambios de toma no detectados.

-FAs : False Alarms, detecciones de cambio de toma falsas.

El objetivo es que ambas variables, *Recall* y Precisión, sean lo más altas posibles. Aunque prácticamente imposible de alcanzar, el máximo valor que pueden tomar ambas es 1, lo que supondría un 100% de eficacia en la variable en cuestión. Todo esto viene más detallado en [42].

3. Otro modo de detectar cambios de toma es a partir de los **modos de predicción** en macrobloques Intra. Generalmente se realizan primero histogramas de la cantidad de veces que aparece cada modo y de ellos se extrae una medida acerca de su variabilidad (ya hemos visto que cada tipo de macrobloque Intra, 4x4 o 16x16, tiene varios modos distintos de predicción).

Estos modos de predicción apuntan a una zona del *frame* de referencia donde aparece el bloque más parecido al actual que se está codificando, con la consecuente disminución de codificación del error de predicción. En una misma escena, dichos modos de predicción tenderán a ser muy parecidos en los 25 o 30 *frames* por segundo que se utilizan en un vídeo. Sin embargo, un cambio de escena probablemente producirá un cambio drástico en la elección de dichos modos, modificando también, con casi toda seguridad, la cantidad de cada uno de los modos de un tipo u otro entre dos *frames* I consecutivos. Hay varios trabajos sobre este tema, entre ellos el de Wei Zeng [43], en el cual realiza pruebas de eficiencia marcando un umbral de decisión para la detección automática de cambios de toma en un vídeo. El objetivo de este proyecto es observar dicha variabilidad y ver si efectivamente tiene sentido trabajar sobre este parámetro, aunque no se determinará el valor umbral que optimice los recursos.

En el capítulo de resultados, se analizarán las distintas formas de extraer información para detección de cambios de toma y, posteriormente, se analizará y se sacarán conclusiones.

4.2 CONTORNOS DE OBJETOS

En el terreno de la segmentación espacial aparece con especial relevancia la detección de bordes de objetos, o detección de los propios objetos en sí.

- En [44] se utilizan los **tipos de macrobloque Intra**. La idea principal en la que se basa este *paper* es en que las zonas más texturizadas del cuadro tendrán una codificación 4x4 y las zonas más homogéneas tendrán una de 16x16. En las diferentes pruebas realizadas se pueden observar resultados aceptables en la detección de los bordes de objetos.

- Además de la codificación 4x4 o 16x16 de los distintos macrobloques en un *frame* Intra, aparecen los **modos de predicción** de cada macrobloque (o sub-bloques en caso de ser 4x4). La elección de los modos tiene que ver con la estructura espacial del cuadro que se está codificando, buscando siempre la optimización de recursos como la minimización de la tasa binaria

  La predicción Intra reduce la redundancia espacial aprovechando la correlación entre bloques adyacentes dentro del mismo *frame*. Cada *frame* se divide en macrobloques de 16x16 píxeles y cada macrobloque está formado por componentes de luminancia y crominancia. Para los componentes de luminancia, los macrobloques 16x16 se pueden dividir a su vez en un tamaño de hasta 4x4 bloques. Los componentes de crominancia se predicen mediante bloques de 8x8 con una técnica de predicción similar que la de luminancia de 16x16. Como ya hemos visto, hay nueve modos de predicción para los bloques de luminancia 4x4 y cuatro modos de predicción para los bloques de luminancia 16x16. Para los componentes de crominancia, hay cuatro modos de predicción que se aplican a los dos bloques 8x8 de crominancia (U y V).

  El codificador selecciona el modo de predicción para cada bloque de forma que minimice la diferencia entre el bloque a codificar y los bloques codificados y reconstruidos previamente. Por tanto, se puede observar el comportamiento en la selección de los modos de predicción. Los bloques pertenecientes al mismo objeto dentro de un *frame* tienden a intra-codificarse siguiendo los modos la dirección en la que el objeto continúa. Por ejemplo, en un programa de televisión los modos de predicción irán siguiendo la dirección de la chaqueta del presentador, apuntando a los bloques más parecidos previamente codificados, que no son otros que la propia chaqueta. Además de los bordes de un objeto, también aparecerán zonas de macrobloques 4x4 en zonas muy texturizadas, como puede ser una camisa a rayas, una valla, etc. Aún así, el codificador sólo busca reducir el número de bits, sin seguir un patrón ni una detección de objetos. Por ello, es muy complicado conseguir información determinante en este aspecto desde el dominio comprimido. Sí se pueden observar casos concretos en los que parece que el codificador sigue el contorno de un objeto, pero no es una regla y pueden aparecer tramos en los que esto no se cumpla. Además, un macrobloque Intra 4x4 tiene un modo de predicción por cada uno de los 16 bloques en los que está dividido dicho macrobloque. Algunos de ellos parecen seguir un patrón pero otros no, cambiando de repente la dirección. Esto se debe a que puede haber encontrado otro bloque cuya predicción es mejor, a pesar de no seguir una trayectoria aparentemente óptima.

Una posible solución es analizar, mediante un histograma, el número de veces que salen los distintos modos de predicción en los macrobloques Intra 4x4 y 16x16. Si están en rangos similares, se podría apostar por que de un *frame* Intra al consecutivo no se ha producido un cambio de toma. Sin embargo, el problema vuelve a ser el mismo: la aparición o desaparición repentina de objetos en la misma toma, o movimientos rápidos. Además, recordar que entre un *frame* Intra y el siguiente pueden aparecer 10 o 15 *frames* P ó B entre medias, con lo cual puede haber muchos cambios entre dos *frames* Intra consecutivos.

- Falta hablar aún de la detección de objetos mediante **vectores de movimiento**. Con el estudio ya presentado de la función y el procedimiento de los vectores de movimiento, no es difícil asumir que, si un objeto o una persona se mueve, se podrán observar diferencias en los vectores de movimiento del background y del *foreground. Con esto es posible distinguir* al objeto o persona en cuestión a partir de dichos vectores de movimiento. Desde MPEG-1, este es un importante tema de trabajo, que se continúa hoy en día con H.264. Encontramos múltiples referencias a este tema, como por ejemplo en [45],[46],[47],[48].

Al igual que con los cambios de toma, en la sección de resultados se analizará la información extraída y se procederá a sacar conclusiones acerca de estos resultados, determinando la fiabilidad de los mismos y sus aplicaciones prácticas.

4.3 RESUMEN PARÁMETROS EXTRAÍBLES

Finalmente, pasamos a resumir brevemente los parámetros extraíbles en el dominio comprimido de H.264 con los que vamos a trabajar:

- **Tipo de macrobloque en *frames* intra (I)**: estos macrobloques pueden tener tamaño 16x16 o 4x4.

Como ya hemos visto, si se produce un cambio considerable del número de macrobloques de tipo 16x16 o 4x4 entre dos *frames* I consecutivos, es posible suponer que se ha producido un cambio de toma entre estos dos *frames* consecutivos. Un cambio de toma puede producirse en un *frame* tipo I, P ó B. Lo que se calcula al advertir un cambio de toma de esta manera es el GOP (*Group Of Pictures*) donde se ha producido, pues al observar cambios considerables en un parámetro que sólo podemos detectar en *frames* Intra, no es posible calcular con precisión exacta dónde se ha producido el cambio de toma.

Adicionalmente, este parámetro puede utilizarse para marcar contornos de objetos o personas dentro de un *frame*. La partición 16x16 (sin partición, un único bloque) se utiliza para zonas homogéneas de la imagen, y la partición del macrobloque 4x4 (macrobloque partido en 16 bloques) se utiliza para zonas más texturizadas de la imagen. Con esto, se logra distinguir algunos objetos o personas dentro de una imagen. No es 100% fiable ya que si dicho objeto o persona contiene rasgos poco homogéneos (por ejemplo, una persona con camisa de rayas o con algún dibujo en la misma) no será posible distinguir dicho contorno.

- **Número de macrobloques I y P ó B en un *frame***: un *frame* I sólo tiene macrobloques I, un *frame* P tiene macrobloques I ó P, y un *frame* B tiene macrobloques I ó B.

Este parámetro permite detectar cambios de toma. Consiste en observar dentro de los *frames* P ó B cuántos macrobloques son I y cuántos son P ó B respectivamente (no habrá macrobloques P en *frames* B y viceversa). Si el primer *frame* de una escena no es de tipo I, se observará con casi toda seguridad que el número de macrobloques I en el *frame* P ó B correspondiente será muy elevado (mínimo 90%).

- **Dirección de predicción (*forward* o *backward*) de los MB**: los macrobloques P ó B pueden utilizar predicción temporal *forward* o *backward*.

Atendiendo a la dirección que asumen los vectores de movimiento de los distintos macrobloques, es posible detectar un cambio de toma. Los distintos parámetros para la detección de cambios de toma no son completamente fiables, y por lo tanto es recomendable complementarlos entre sí para una detección más fiable.

Es lógico pensar que el último *frame* de una escena no tenga predicción *backward*, y que el primero de una escena no tenga predicción *forward*. Si observamos las predicciones de los distintos macrobloques de un *frame* y se cumplen estas condiciones con un porcentaje alto, será posible determinar un cambio de toma con mayor certeza.

- **Modos de predicción en un MB Intra**: como ya se ha mencionado, en un *frame* Intra, los MB pueden ser 16x16 o 4x4. Los primeros tienen 4 modos de predicción y los segundos tienen 9 modos de predicción, los cuales se eligen buscando minimizar la tasa de transmisión.

La optimización resulta cuando bloques pertenecientes a un mismo objeto acaban intra-codificándose, permitiendo seguir, con los modos de predicción, contornos de objetos o personas.

Sin embargo, este parámetro no proporciona una información demasiado cuantiosa o fiable, pues es el codificador el que finalmente realiza la elección del modo y su finalidad no es la de buscar contornos, sino la de buscar la menor tasa de transmisión.

- **Cantidad de los distintos modos de predicción en un *frame* intra**: la utilidad de este parámetro reside en observar la cantidad de veces que sale en un *frame*, y analizando mediante un histograma las diferencias entre dicho histograma en un *frame* I y el siguiente *frame* I. Si las diferencias son muy grandes, podemos entender que estamos en un cambio de toma. Aún así, es necesario tratar con los problemas de los parámetros anteriores para cambios de toma. Puesto que el análisis se realiza sólo sobre los *frames* I, pueden producirse muchos cambios aún estando en la misma escena de un *frame* I al siguiente.

- **Vectores de movimiento**: aunque las nuevas características de H.264 aumentan considerablemente la complejidad de trabajo con dicho parámetro, sigue siendo de mayúscula importancia el análisis de vectores de movimiento en el dominio comprimido.

Una de las principales utilidades de este parámetro es la de la segmentación de objetos o personas a partir del movimiento.

4.4 DESARROLLO SOFTWARE

El desarrollo software utilizado para la extracción de todos estos parámetros comentados anteriormente tiene lugar en el mismo punto de decodificación del decodificador JM.

Para ello, se ha utilizado el códec JM, que aparece de manera gratuita junto con el manual de referencia en [49]. Concretamente, se ha trabajado con la versión 12.4, también disponible en el link anterior.

El proceso de extracción consiste en detectar el momento en que la decodificación de cada *frame* llegaba a su punto final, pero aún no se habían 'limpiado' los datos de cara a la decodificación de un nuevo *frame*. Una vez detectado este punto, se buscaba abordar dichos datos de la manera más adecuada en función del parámetro a extraer. Los datos de todos los parámetros se podían deducir del mismo punto. Dichas extracciones se han obtenido a través de ficheros de texto (.txt) de cara a su análisis posterior.

En algunas ocasiones estos ficheros de texto no eran suficientes para el análisis requerido y fue necesario un procesado con Matlab para observar gráficamente los resultados.

Por último, mencionar también que se ha hecho uso del software VISUALmpegAVC, el cual permitía una visualización rápida de qué estaba ocurriendo realmente para poder comparar con los datos extraídos. Sin embargo, esta aplicación no permitía el uso de ningún parámetro, sino simplemente su visualización.

5 PRUEBAS Y RESULTADOS

5.1 INTRODUCCIÓN

Una vez extraídos dichos parámetros a través del codificador/decodificador JM, en este capítulo del proyecto se llevará a cabo la presentación de resultados para comprobar la utilidad práctica de dicha extracción.

Los parámetros a extraer son los que se han resumido brevemente al final de la sección anterior (en el capítulo 4). Conviene recordar que el objetivo de este Proyecto no es que los resultados sean óptimos, sino lo más objetivos posible, asesorando sobre la utilidad y eficacia de trabajar con dichos parámetros en las aplicaciones de análisis de vídeo que se consideren oportunas para cada uno de estos parámetros que se han destacado en nuestro estudio del apartado anterior.

Así, se procederá directamente con el estudio de cada uno de estos parámetros en las siguientes subsecciones, analizando los resultados de cada uno de ellos.

5.2 METODOLOGÍA

En este apartado se quiere mencionar brevemente cómo se ha procedido a la hora de evaluar los parámetros obtenidos, así como hacer una breve reseña sobre los vídeos utilizados y la manera de presentar los resultados.

En cada uno de los parámetros que se extraen, lo que se hace es presentar los resultados de la manera más gráfica y representativa, generalmente tras un breve procesamiento de los datos obtenidos directamente del decodificador JM, procediendo a continuación a las conclusiones que se pueden deducir de lo visto anteriormente en la teoría y en otros trabajos. La misión no es encontrar un algoritmo, con su consecuente umbral, para obtener resultados directamente, sino más bien observar qué es lo que ocurre y si realmente tiene la utilidad que debería, si resulta fiable y en qué proporción. A su vez, tanto si los datos extraídos revelan que dicho parámetro proporciona la información adecuada como si no, se intenta buscar explicaciones de por qué ocurre esto con el apoyo de la herramienta de trabajo VISUALmpegAVC, observando qué ocurre en la secuencia de vídeo en el punto concreto del mismo. Este procedimiento se lleva a cabo con todos los parámetros.

Los resultados no se presentan de la misma manera para todos los parámetros, pues se entiende que son parámetros muy distintos y la forma de observarlos puede ser conveniente de una u otra manera en función del parámetro en cuestión, pudiendo presentarlos gráficamente, en forma de tabla, mediante imágenes, etc.

Los vídeos que se van a analizar son vídeos que se encontraban en la base de datos del grupo de investigación 'VPUlab' de la Universidad Autónoma de Madrid. Algunos ya venían en formato .H264, otros estaban en formato .YUV o .MPG. Para estos últimos vídeos se utiliza la herramienta ffmpeg para convertirlos al formato deseado .H264. También se ha usado esta herramienta para cambios de resolución de los mismos vídeos.

A continuación, en la "Tabla 5-1" se expone una breve descripción de todos los vídeos que se han utilizado en el presente trabajo:

TABLA 5-1 VÍDEOS CAPÍTULO 5

	Archivo	Resolución	Duración (seg)	Nº Frames
1	news.h264	368x288	7	182
2	fragment0.h264	720x576	59	1496
3	fragment0.h264	368x288	59	1496
4	fragment1.h264	720x576	59	1504

A su vez, de cada vídeo que se usado la estructura de GOP considerada más oportuna para cada caso. Así, del vídeo "fragment0.h264" se utilizan diferentes GOPs como IPPP...IPPP…, IBPBPB…IBPBPB…, IBBPBBP…IBBPBBP…

De cualquier modo, antes de la presentación de resultados en cada apartado, se comentará el vídeo utilizado, así como los parámetros del mismo y su estructura.

Se han utilizado estos vídeos por encontrarse en la base de datos disponible y por una cierta comodidad a la hora de manejarlos debido a un tamaño adecuado de los mismos, permitiendo sacar suficientes pruebas y resultados de cada vídeo sin consumir demasiados recursos y tiempo.

5.3 TIPOS DE MACROBLOQUE EN FRAMES INTRA (I)

Como se ha comentado anteriormente, uno de los parámetros extraíbles del formato de compresión H.264 en dominio comprimido, y que puede ser de mucha utilidad en la detección de cambios de toma abruptos, es el tipo de macrobloques en *frames* intra. Hay dos tipos de macrobloque intra, con tamaño 4x4 para zonas más texturizadas, y 16x16 para zonas más homogéneas, tales como el fondo de una imagen, que además no conlleva mucho cambio de un *frame* al siguiente.

Así, analizar la variación del número de macrobloques de tamaño 4x4 y 16x16 en los diferentes *frames* intra puede dar una idea acerca de un posible cambio de toma en la secuencia de vídeo. Asimismo, según se verá más adelante, también puede ofrecer información sobre detección de contornos de objetos o personas en la imagen.

Para ello se han usado vídeos en formato .h264 y extraído el número de macrobloques 4x4 y 16x16 de todos los *frames* intra de las distintas secuencias de vídeo a analizar a través del codificador/decodificador JM. De este modo, no observaremos qué ocurre en *frames* P ó B intermedios entre estos *frames* I consecutivos. A continuación se exponen los resultados de dicho análisis y qué relación tienen con la localización de los verdaderos cambios de toma. Para esta tarea también se ha usado la aplicación VISUALmpegAVC.

Se observan dichos resultados de manera gráfica, pudiendo analizar con detalle dicha información en el "*Apéndice A*".

5.3.1 CAMBIOS DE TOMA

Se describen a continuación los vídeos utilizados para la detección de cambios de toma.

Vídeos utilizados:

TABLA 5-2 VÍDEOS CAPÍTULO 5.3.1

	Archivo	Resolución	Duración (seg)	Nº Frames	GOP
1	new.h264	368x288	7	182	IPPP…
2	fragment0.h264	720x576	59	1496	IPPP…
3	fragment1.h264	720x576	59	1504	IPPP…

Resultados:

Ejemplo 1: news.h264

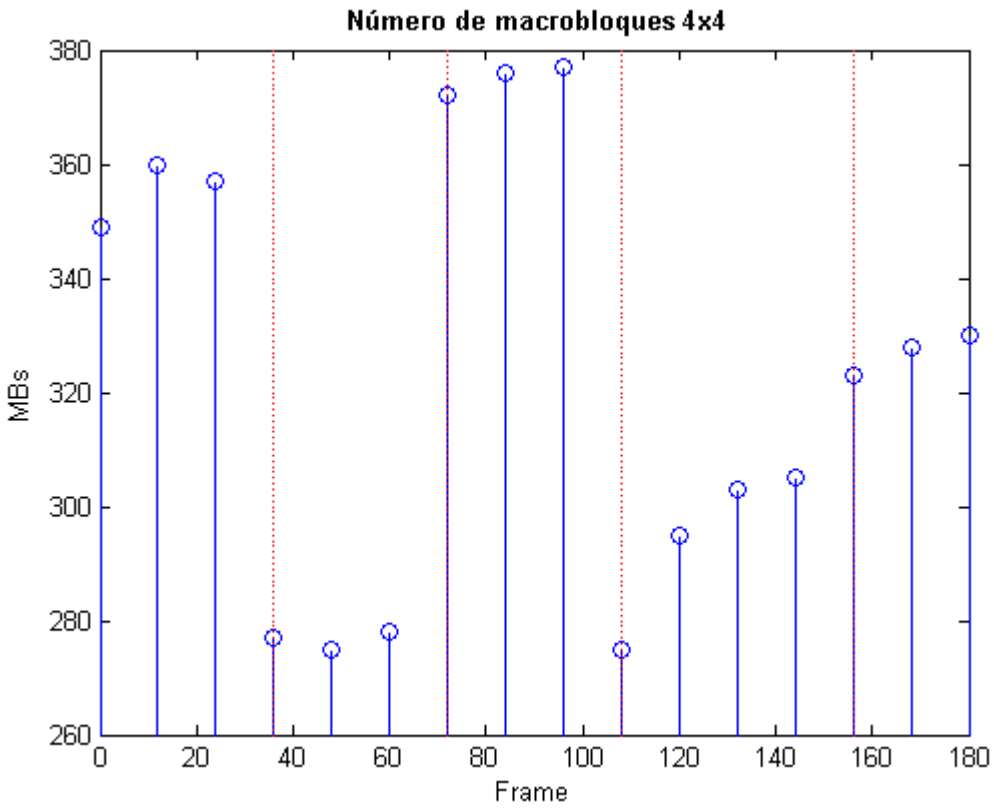


FIG. 5-1 NÚMERO MB'S INTRA 4X4 VÍDEO "NEWS.H264"

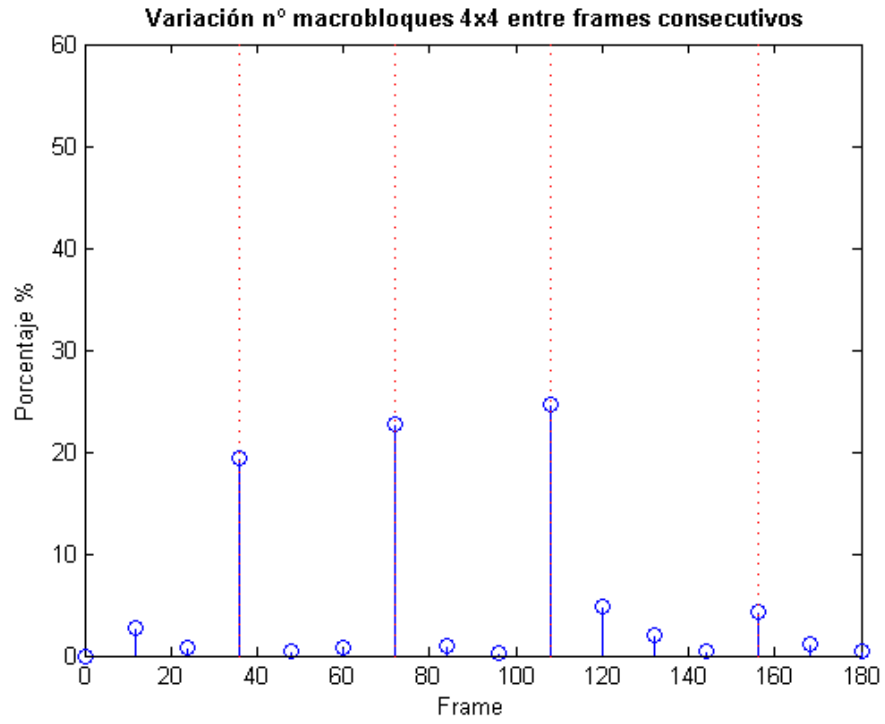


FIG. 5-2 VARIACIÓN MB'S INTRA 4X4 VÍDEO "NEWS.H264"

En primer lugar, decir que la resolución de este vídeo es de 368x288 píxeles. Sabiendo que cada macrobloque es de 16x16 píxeles, y realizando la cuenta, obtenemos que el tamaño de cada *frame* del vídeo es de: (368/16)x(288/16) = 23x18 = 414 macrobloques. Éste es el número total de macrobloques que hay en cada *frame*. Dado que estamos analizando los *frames* I, tendrán estructura 4x4 ó 16x16, y la suma de unos y otros macrobloques de distinto tipo deberá ser siempre 414.

Hay que observar variaciones significativas en el número de macrobloques con estructura 4x4 ó 16x16, si hay más de un tipo, habrá menos del otro. Esta relación es directa, si hay más macrobloques con estructura 4x4 habrá menos en la misma proporción con estructura 16x16. Hay que tener en cuenta que el número de macrobloques en total es 414, así que un cambio de 80, 100 macrobloques en cualquier tipo es ya un cambio significativo. En las figuras "*Fig. 5-1*" y "*Fig. 5-2*" se representan el número de macrobloques Intra con estructura 4x4 y las variaciones del número de macrobloques 4x4 entre frames I consecutivos respectivamente. En ambas figuras se han representado gráficamente, mediante líneas verticales discontinuas de color rojo, los frames en los que se produce un cambio de toma. A simple vista, en la "*Fig. 5-2*" se aprecian cambios de toma en los *frames* 36, 72, 108, y el 156. Este último con alguna duda, pues el cambio no es muy grande con respecto al frame I anterior. Los valores exactos del número de macrobloques Intra con estructura 4x4 de este vídeo se pueden ver en el "*Apéndice A*".

Si analizamos con el VISUALmpegAVC se observan que los cambios de toma se producen en los *frames*: 36, 72, 108 y 156. Se trata de un vídeo corto y muy sencillo de analizar, con cambios de toma abruptos y en este caso no hay ningún cambio de toma que lleve a confusión (como sí ocurrirá en los siguientes ejemplos). En el último cambio de toma apenas se observa gran diferencia en el número de macrobloques 4x4 o 16x16 debido a que, tras el cambio de toma, la secuencia permanece en el mismo escenario, con poca iluminación.

Ejemplo 2: fragment0.h264

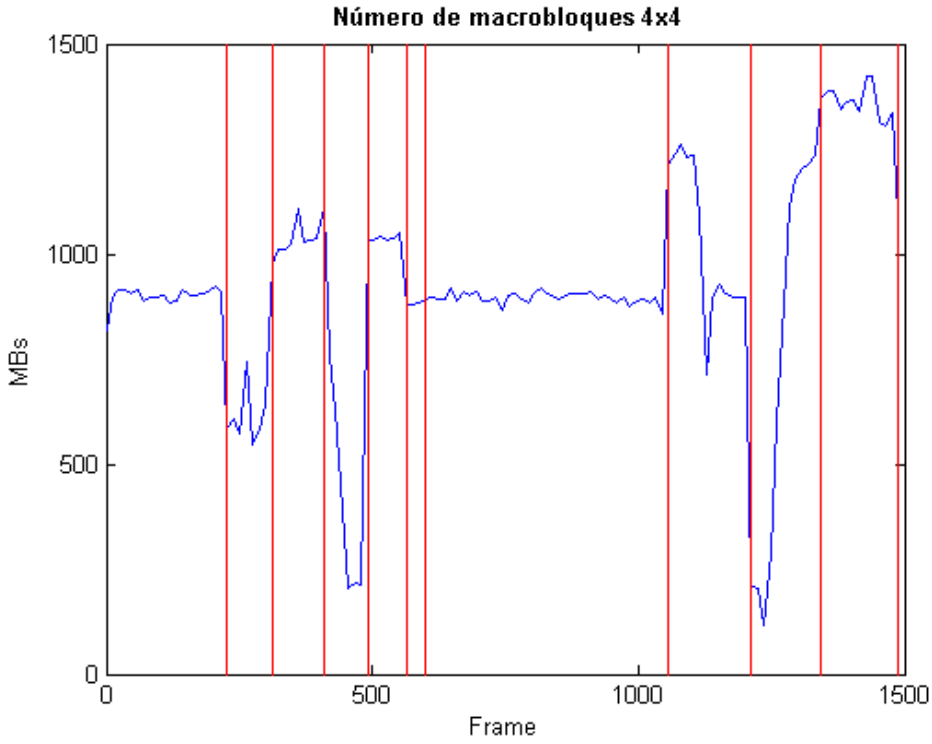


FIG. 5-3 NÚMERO MB'S INTRA 4X4 VÍDEO "FRAGMENT0.H264"

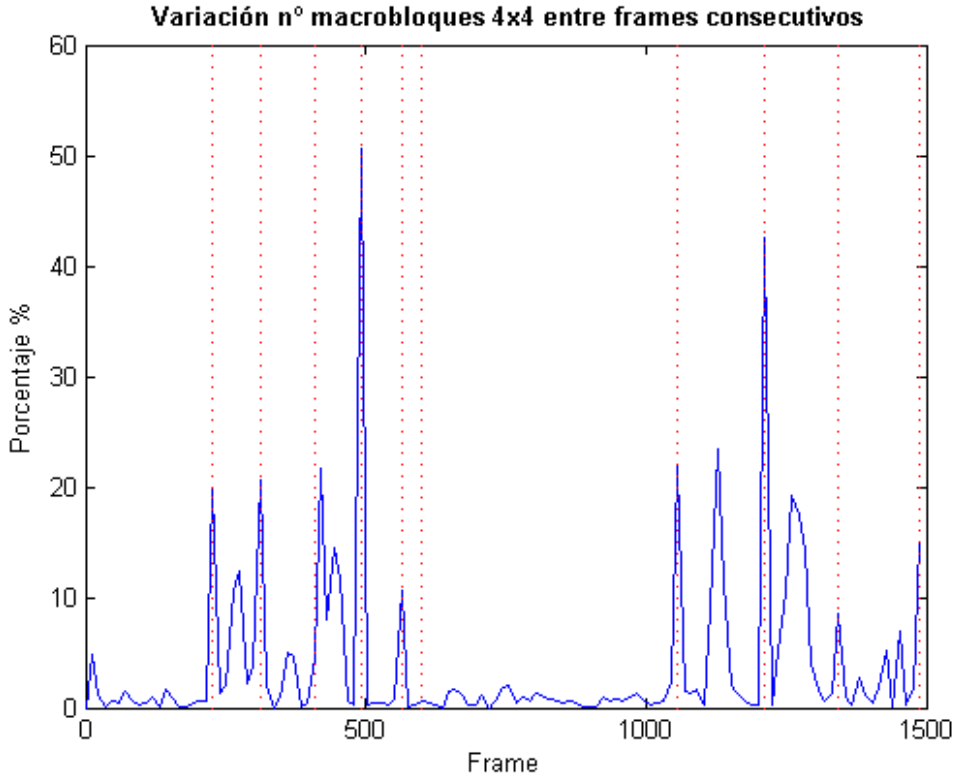


FIG. 5-4 VARIACIÓN MB'S INTRA 4X4 VÍDEO "FRAGMENT0.H264"

Analizando con la aplicación VISUALmpegAVC se puede observar que los cambios de toma se producen en los *frames* I: 228, 312, 408, 492, 564, 600, 1056, 1212, 1344, 1488. Estos cambios de toma no tienen por qué producirse justo en los *frames* I, sino que pueden producirse en algún *frame* P situado entre dos *frames* I consecutivos. Se representan mediante las líneas verticales discontinuas de color rojo.

Mencionar también para este ejemplo que todos los *frames* de esta secuencia de vídeo tienen una resolución de 720x576. Haciendo la misma cuenta que en el ejemplo anterior, se deduce que la dimensión de estos *frames* en macrobloques es de: 45x36 macrobloques, es decir, 1620 macrobloques, los cuales serán 4x4 o 16x16. Por tanto, la suma de ambas cifras siempre será 1620.

En las figuras "*Fig. 5-3*" y "*Fig. 5-4*", aunque se lleve a cabo una representación continua de los valores, es conveniente no olvidar que los valores son discretos, los frames I aparecen de manera periódica cada 12 frames en este ejemplo. Se ha hecho de esta manera por conseguir una visualización más sencilla de los resultados.

A partir de estos resultados, se ha podido llegar a las siguientes conclusiones:

- La mayoría de los cambios de toma se detectan correctamente tras un cambio considerable en el número de ambos tipos de macrobloque. Por considerable se entiende un cambio mínimo de 200, 300 macrobloques de cada tipo, dado que la variación entre *frames* I consecutivos correspondientes a la misma escena oscila entre 0 y 100.

- Si se observa el *frame* 264, se produce la aparición de un rótulo. Esto se ve reflejado en que el número de macrobloques aumenta/disminuye en casi 200 para cada tipo, y el siguiente *frame* I vuelve al número anterior.

- Se produce un cambio de toma en el *frame* 408, cuando se creía que iba a realizarse en el *frame* 420 (el siguiente I). Esto puede deberse a que, aunque se produzca un cambio de toma, el número de macrobloques 4x4 y 16x16 apenas varíe debido a un mismo número de zonas muy texturizadas que requieran codificación 4x4 y un mismo número de zonas homogéneas que requieran codificación 16x16, a pesar de que dichas zonas no se parezcan en nada. Lo mismo ocurre en el cambio de toma que se produce en el *frame* 1344. Aquí es donde se puede ver una de las limitaciones de observar cambios de toma sólo observando este parámetro. Por este motivo, se deben analizar más parámetros.

- Parece claro que en el *frame* 1128 se tiene que producir un cambio de toma porque el cambio es bastante brusco, pero no sucede así. Esto se debe a que la toma hasta el *frame* 1116 consiste en un primer plano de una jarra de cerveza con una mano sujetando dicha jarra. Para el *frame* posterior, el 1128, dicha cerveza ya no se encuentra en el plano, desaparece por completo un objeto que ocupaba gran parte de la imagen ("*Fig. 5-5*"). Es comprensible que pueda llevar a confusiones creyéndose que ahí se pueda producir un cambio de toma.

Frame 1116

Frame 1128

FIG. 5-5 FRAMES I: 1116 Y 1128 VÍDEO "FRAGMENT0.H264"

- Se puede ver en la figura "*Fig. 5-6*" cómo entre los *frames* 1248 y 1284 (cuatro frames I) se producen cambios que, en condiciones normales, intuiríamos por un cambio de toma. Lo que llamaba la atención era que se producían consecutivamente y es raro porque el paso de 12 *frames* (los que hay entre dos *frames* I consecutivos en este vídeo) en el tiempo es muy pequeño. Analizando qué ocurre realmente se puede encontrar una explicación. La escena se encontraba en una panorámica del cielo, con una zona muy homogénea (por eso el número de macrobloques 16x16 es muy amplio en la escena 1236), y a partir del *frame* 1248 va descendiendo para acabar viéndose en la escena un edificio que, obviamente, produce zonas más texturizadas que las del cielo y se incrementa considerablemente el número de macrobloques 4x4.

FRAME 1236

FRAME 1248

FRAME 1260

FRAME 1272

Frame 1284

FIG. 5-6 FRAMES I: 1236-1284 VÍDEO "FRAGMENT0.H264"

Ejemplo 3: fragment1.h264

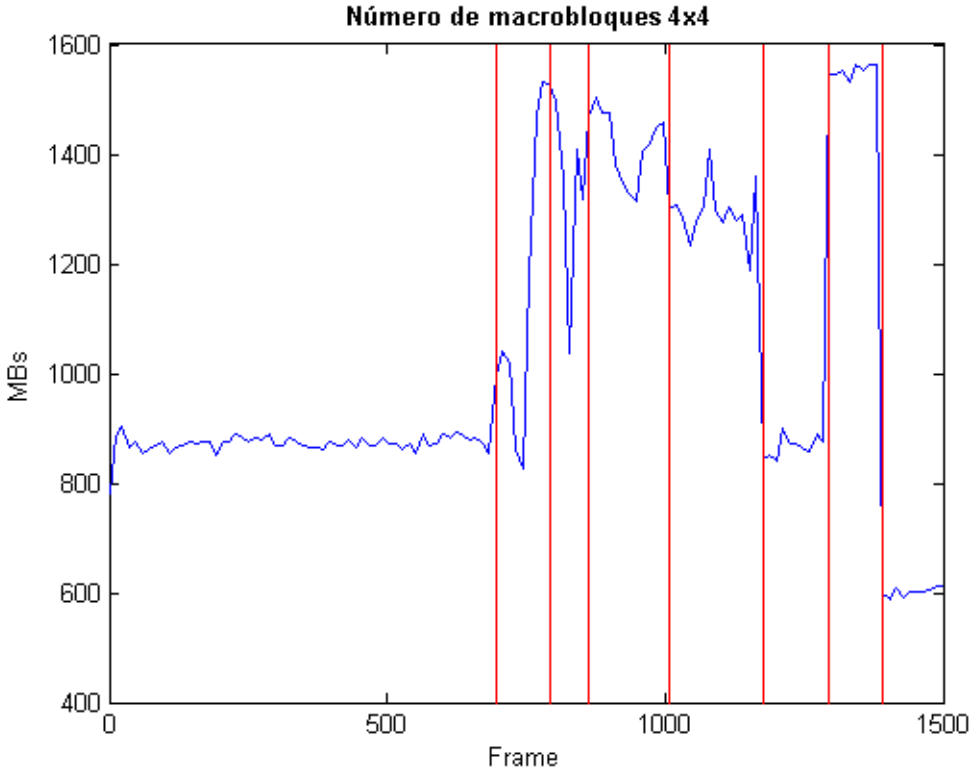


FIG. 5-7 NÚMERO MB'S INTRA 4X4 VÍDEO "FRAGMENT1.H264"

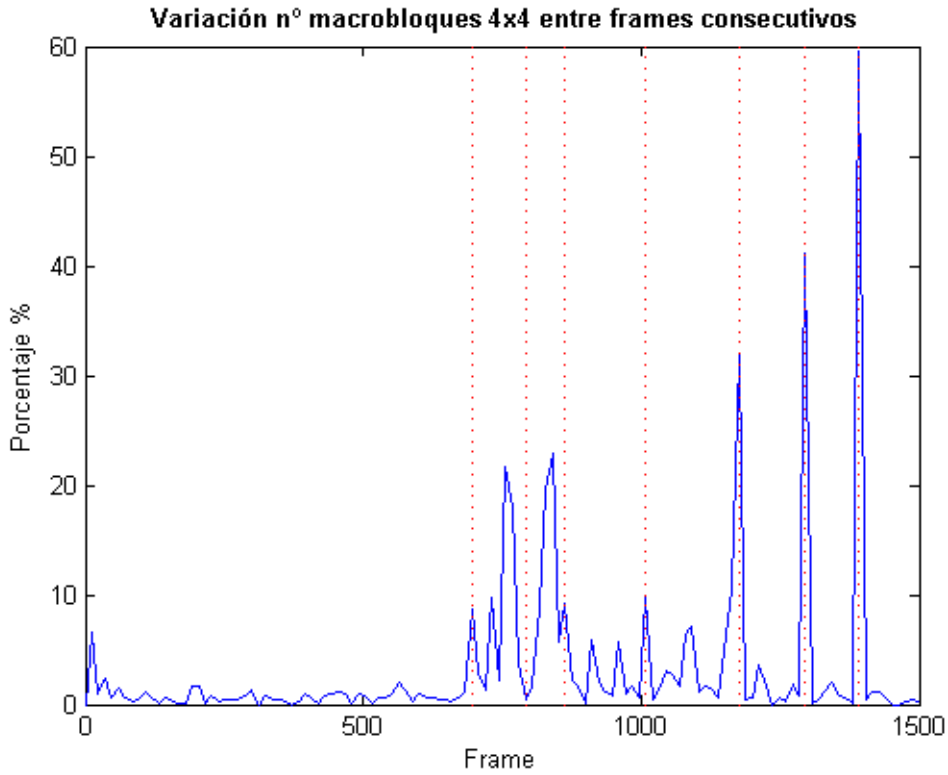


FIG. 5-8 VARIACIÓN MB'S INTRA 4X4 VÍDEO "FRAGMENT1.H264"

Al igual que en el ejemplo 2, en las figuras "*Fig. 5-7*" y "*Fig. 5-8*", se visualizan los resultados de una manera continua aunque dichos valores sean discretos, vuelven a aparecer los frames I de manera periódica cada 12 frames. Se ha hecho de esta manera por tratarse de una visualización más sencilla de los resultados. Vuelven a aparecer los cambios de toma mediante las líneas verticales discontinuas de color rojo.

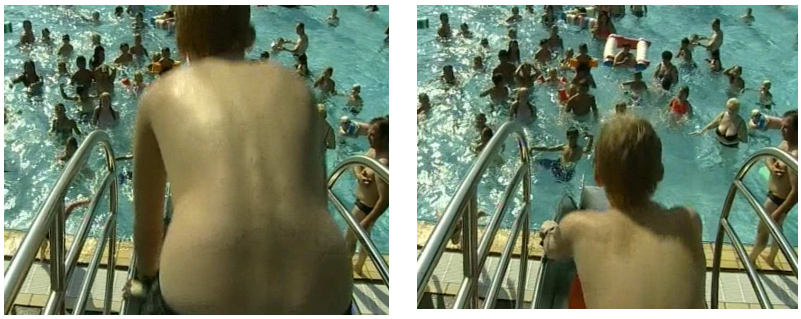
El tamaño de los *frames* en macrobloques es de 45*36 = 1620, igual que en el ejemplo anterior.

Se ve, a través de la cantidad de macrobloques de cada tipo en los distintos *frames* de la secuencia de vídeo, que los posibles cambios de toma se podrían producir en los frames: 696, 768, 828, 1008, 1176, 1296, 1392.

Si se analizan los *frames* de dicho vídeo, se observan que los cambios de toma se producen en los *frames*: 696, 792, 864, 1008, 1176, 1296 y 1392.

A continuación se intenta deducir por qué no coinciden con exactitud lo que se preveía viendo el número de macrobloques en cada *frame* y las variaciones con respecto al frame anterior y los verdaderos cambios de toma:

- En el *frame* 768 se produce un cambio algo brusco porque sale la espalda de un niño en gran parte del cuadro del vídeo anteriormente y, en dicho *frame*, el niño se agacha en la misma toma pero cambiando gran parte del cuadro. Se aprecia en la figura "*Fig. 5-9"* que aparece a continuación:



Frame 756 **Frame 768**

FIG. 5-9 FRAMES I: 756 Y 768 VÍDEO "FRAGMENT1.H264"

- No se detecta ni mucho menos un cambio de toma en el *frame* 792. Se puede pensar que una piscina llena de niños da muchas zonas texturizadas (y no homogéneas). De este modo, el cambio de toma apenas se nota, pues tras dicho cambio de toma el escenario sigue siendo la piscina, donde sigue habiendo muchas zonas no homogéneas.

- En el *frame* 828 se observa algo raro, porque sólo cambia en dicho *frame* el número de macrobloques de un tipo u otro, volviendo en el siguiente a unos números parecidos al *frame* anterior al 828. Un rótulo suele permanecer en escena durante más de un *frame* de tipo I, dado que en el paso de un *frame* I al siguiente hay muy poco tiempo transcurrido. Se procedió a ver qué ocurría exactamente y era la aparición momentánea de un niño ocupando gran parte de la imagen.

- En el *frame* 864 (“*Fig. 5-10*”) se produce un cambio de toma pero la localización de una toma y otra es la misma. Por el mismo razonamiento anterior, se cree que puede ser el motivo fundamental de que no se percibiera un cambio de toma a partir de la información extraída.

Frame 852 **Frame 864**

FIG. 5-10 FRAMES I: 852 Y 864 VÍDEO “FRAGMENT1.H264”

- En el *frame* 1008 efectivamente se produce un cambio de toma, aunque no es un cambio tan brusco el que se produce en relación al número de macrobloques de ambos tipos. Pero sí lleva a pensar eso el hecho de que se produce un cambio y se mantiene en los siguientes.

- En los *frames* 1176, 1296 y 1392 se producen cambios de toma fácilmente perceptibles a partir del número de macrobloques 4x4 y 16x16.

Conclusiones

Este parámetro puede ayudar a decidir si se produce o no un cambio de toma pero no es ni mucho menos concluyente. En primer lugar, no da el punto exacto donde se produce el cambio de toma, sino el intervalo de *frames* donde se produce (en este caso entre un *frame* I y el siguiente hay 12 *frames* entre medias, pudiéndose haber producido el cambio de toma al pasar a cualquiera de esos *frames*). Además, no tiene toda la fiabilidad que se quisiera, un movimiento rápido (como la espalda del niño) o la desaparición rápida de un objeto (como la cerveza) da lugar a conclusiones erróneas.

Sí servirá como información adicional para la detección de cambios de toma, junto a otras que se verán a continuación.

5.3.2 DETECCIÓN DE CONTORNOS

Este mismo parámetro puede ser utilizado con otra finalidad, y no es otra que la detección de contornos de objetos o personas en una imagen. Como ya se ha dicho, en cuadros Intra sólo se utilizan dos tamaños de macrobloque, 16x16 ó 4x4, los primeros para zonas más homogéneas del cuadro y los segundos para zonas más texturizadas. A continuación se verán un par de ejemplos demostrando que esto es así.

Pero también se puede concluir que no es tan eficiente, pues basta con una imagen un poco cargada con muchos objetos juntos, o con objetos con forma irregular, o una persona con camiseta muy recargada de símbolos, dibujos, frases, etc., para comprobar que los resultados no son muy buenos. Sí sirve, por ejemplo, en casos muy concretos, como un presentador con un fondo relativamente homogéneo detrás. Por ejemplo, si el fondo consiste en una redacción trabajando (imagen típica últimamente en los telediarios), o si el presentador va con un traje o camisa de rayas, el resultado puede salir con poca claridad. En cualquier caso, se verán a continuación un par de ejemplos que ilustren esto.

De otro modo, también comentar un tema adicional respecto a estas mismas imágenes resultantes. Se trata de la resolución. Si la imagen tiene mayor resolución, más nítida, con más píxeles, el detalle será mayor y se tendrán más bloques sobre los que ver si tienen tamaño 4x4 ó 16x16 y, de esta manera, distinguir mejor el cambio de tipo de macrobloques.

La estructura de esta subsección consistirá en ver primero la diferencia entre imágenes con un fondo más recargado e imágenes con más zonas homogéneas, para ver la diferencia. Y después de ver y comentar dichas diferencias, se procederá con ejemplos de imágenes iguales (las vistas en la primera parte) con distinta resolución observando los efectos señalados recientemente.

Se describen previamente los vídeos utilizados en este apartado.

Vídeos utilizados:

TABLA 5-3 VÍDEOS CAPÍTULO 5.3.2

	Archivo	Resolución	Duración	Nº Frames	GOP
1	fragment0.h264	720x576	59	1496	IPPP…
2	fragment0.h264	368x288	59	1496	IPPP…
3	fragment1.h264	720x576	59	1504	IPPP…

Resultados:

Ejemplo 1: Frame 0 del vídeo fragment0.h264, 720x576, IPP…

Imagen original **Imagen resultante**

FIG. 5-11 DETECCIÓN CONTORNOS FRAME 0 VÍDEO "FRAGMENT0.H264"

En primer lugar, comentar que se está trabajando con una resolución inicial del vídeo de 720x576. Posteriormente se cambiará la resolución para ver las diferencias a raíz de lo recién comentado. Y también comentar que en la imagen resultante los macrobloques 16x16 tienen color negro, y los macrobloques 4x4 tienen color blanco. Los macrobloques con estructura 4x4 siempre están agrupados de cuatro en cuatro formando bloques del mismo tamaño que un macrobloque de 16x16.

En la figura "*Fig. 5-11*" se puede ver que se trata de una imagen con un fondo más o menos homogéneo. Conforman el cuadro la presentadora y un rótulo (imagen) que aparece en la zona superior izquierda de la imagen. En la imagen derecha de la figura "*Fig. 5-11*" se puede distinguir sin mayor dificultad al presentador sin haber visto inicialmente la imagen original.

Ejemplo 2: Frame 876 del vídeo fragment1.h264

Imagen original **Imagen resultante**

FIG. 5-12 DETECCIÓN CONTORNOS FRAME 876 VÍDEO "FRAGMENT1.H264"

Se ve en la figura "*Fig. 5-12*", cómo en una imagen como la dada, donde no hay casi ninguna zona homogénea, en la cual es difícil distinguir el *foreground* del background, la imagen resultante apenas da información alguna. La única información que se puede sacar es que la gran mayoría de macrobloques son 4x4, pero resulta imposible poder obtener un solo contorno de ningún objeto o persona presente en la imagen.

Este es el inconveniente al que se hacía referencia al principio de este apartado cuando se hablaba de que sólo serviría para imágenes concretas con una clara distinción entre *foreground* y background.

A partir de ahora se darán un par de ejemplos en los se ha cambiado la resolución, tratándose de la misma imagen y así poder ver las diferencias entre las imágenes resultantes para distinta resolución.

Ejemplo 3: Frame 0 del vídeo fragment0.h264 (cambio de resolución)

Imagen original **Imagen resultante (720x576)** **Imagen resultante (368x288)**

FIG. 5-13 DETECCIÓN CONTORNOS FRAME 0 VÍDEO "FRAGMENT0.H264" CAMBIO RESOLUCIÓN

En la figura "*Fig. 5-13*" se ve de manera clara la diferencia de una resolución a otra (aproximadamente la mitad en la segunda respecto de la primera). De entrada, se aprecia la diferencia de resolución en la propia imagen, resultando muy pixelada (granular) la segunda imagen resultante (imagen de la derecha), la de menor resolución. Por ello, sacar conclusiones sobre que el contorno de la derecha es el presentador y el de la izquierda un rótulo está más complicado, pues hay zonas en las que incluso llegan a unirse. Así pues, si uno ve la imagen sin haber visto la original puede pensar que se trata de cualquier cosa.

Ejemplo 4: Frame 432 del vídeo fragment0.h264 (cambio de resolución)

Imagen original **Imagen resultante (720x576)** **Imagen resultante (368x288)**

FIG. 5-14 DETECCIÓN CONTORNOS FRAME 432 VÍDEO "FRAGMENT0.H264" CAMBIO RESOLUCIÓN

En este segundo ejemplo de cambio de resolución, se alcanza a ver en la figura "*Fig. 5-14*" cómo también sale bastante más distorsionada la segunda imagen respecto de la primera, por los mismos razonamientos que en el ejemplo anterior. Aquí se ve un poco mejor, dado que el helicóptero está alejado de la zona de árboles de la parte inferior de la imagen.

Con estos dos últimos ejemplos, se ha querido remarcar la importancia de la resolución de la imagen a la hora de trabajar también con ciertas características de los estándares de compresión.

5.4 NÚMERO DE MACROBLOQUES I/P-B

Este parámetro se puede extraer de manera muy sencilla mediante el codificador/decodificador JM. Antes de nada, vendría bien recordar un pequeño detalle acerca de los *frames* I, P ó B.

Los *frames* I sólo están compuestos de macrobloques I, los *frames* P están compuestos de macrobloques I y P, y los *frames* B de macrobloques I y B. Como se ha mencionado anteriormente, los macrobloques P utilizan predicción *forward* y los *frames* B predicciones *forward* y *backward*.

Así, se puede jugar con esta característica de este formato de compresión de una manera aparentemente muy lógica. Los *frames* P utilizarán como referencias *frames* P anteriores o el *frame* I más cercano ya decodificados. La libertad es absoluta y, a no ser que aparezca un IDR (el cual borra el buffer DPB, *Decoded Picture Buffer*), también pueden referenciarse a *frames* I o P anteriores incluso al último I decodificado. Aquí es donde se encuentra la posibilidad de una detección fiable de cambios de toma. La tendencia en *frames* P es a codificar la mayoría de sus macrobloques con predicción *forward* y no con codificación Intra (se busca una optimización en la compresión), permitiéndose además la multirreferencia de H.264 comentada anteriormente, lo cual permite encontrar predicciones más fiables y, por ende, un mayor número de macrobloques P. De esta manera, se intentará que el número de macrobloques que tengan que ser codificados con codificación Intra sea mínimo. Es una limitación que se encuentra cuando un cambio de toma tiene lugar en un *frame* P: supuestamente ya no va a encontrar referencias fiables de *frames* anteriores y la mayoría de sus macrobloques tendrán que usar codificación Intra, resultando un número mucho mayor de macrobloques I con respecto a anteriores *frames* P o con respecto al número de macrobloques P del propio *frame*. Esto tiene, a priori, un pequeño problema: si el cambio de toma se produce justo en un *frame* I, los siguientes *frames* P encontrarán similitudes con dicho *frame* I y la diferencia de número de macrobloques I con respecto a los anteriores al cambio de toma no será tan grande. Por ello, es conveniente tener otras bazas en la mano con las que poder jugar como son los otros parámetros que se están viendo para detectar también cambios de toma.

En cualquier caso, se analizarán una serie de ejemplos con secuencias de vídeo adecuadas con cambios de toma abruptos, de momento sólo vídeos codificados con *frames* I o P, prescindiendo de los *frames* B. El objetivo es encontrar atractivo dicho parámetro (el número de macrobloques Intra en *frames* consecutivos) y observar si se produce una variación importante a partir de la cual se pueda deducir que se está ante un cambio de toma. Para ello, se extraerá dicha información del codificador/decodificador JM. Dichos resultados se comprobarán con la aplicación VISUALmpegAVC, observando dónde se producen dichos cambios de toma en realidad.

En los siguientes ejemplos, los vídeos tienen resolución 720x576, por lo tanto tienen 45x36 = 1620 macrobloques cada *frame*, de los cuales dichos macrobloques pueden ser I ó P/B según el *frame* sea P ó B respectivamente (obviamente si el *frame* es I, los 1620 macrobloques serán de tipo I). El último ejemplo es el del vídeo con resolución 368x288, así que este vídeo tendrá 23x18 = 414 macrobloques en total cada *frame*.

En las gráficas que se presentan en los siguientes subapartados se representa solamente el número de macrobloques I en *frames* P ó B (según corresponda), pues el número de macrobloques P ó B del mismo *frame* será el resto de macrobloques, dando poca información adicional una vez se dispone del número de macrobloques I. Además, se remarcará con una línea roja discontinua los *frames* en los que se produce realmente el cambio de toma. Se ha dividido este apartado, en diferentes subsecciones de acuerdo a los ejemplos utilizados.

Vídeos utilizados:

TABLA 5-4 VÍDEOS CAPÍTULO 5.4

	Archivo	Resolución	Duración	Nº Frames	GOP
1	fragment0.h264	720x576	59	1496	IPPP…
2	fragment0.h264	720x576	59	1496	IBPBP…
3	fragment0.h264	720x576	59	1496	IBBPBBP…
4	fragment1.h264	720x576	59	1504	IPPP…
5	fragment0.h264	368x288	59	1496	IPPP…

5.4.1 CAMBIOS DE TOMA – SECUENCIAS IPP…IPP…

Ejemplo 1: fragment0.h264

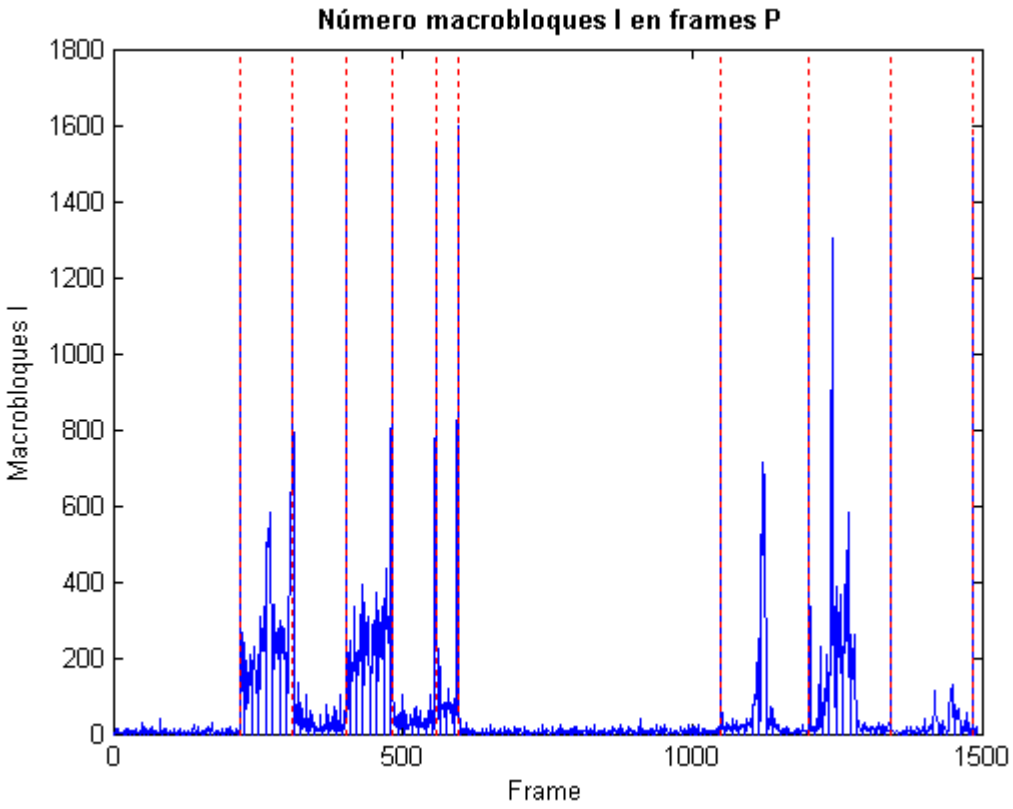


FIG. 5-15 NÚMERO MB'S I VÍDEO "FRAGMENT0.H264"

En los *frames* I que aparecen a lo largo del vídeo, los 1620 macrobloques son de tipo I, pero para poder observar mejor los resultados gráficamente se han puesto a 0 los macrobloques en los *frames* I en los que no se producía cambio de toma (en nuestro ejemplos nunca se producía un cambio de toma en dichos *frames*, así que en todos los *frames* I). Éste es el mayor problema de este parámetro, pues si el cambio de toma se produce en un *frame* I no tendremos manera de, con esta información, deducir si se trata o no de un cambio de toma. Es por motivos como estos por los que se considera muy importante no tener en cuenta solamente un parámetro para la detección de cambios de toma o de cualquier otra aplicación.

Se observa en la figura "*Fig. 5-15*" cómo en la mayoría de los *frames* P, el número de macrobloques P es muy inferior al número de macrobloques I cuando se produce un cambio de toma.

Los *frames* donde se producen realmente los cambios, analizado con el VISUALmpegAVC, son: 219, 311, 402, 481, 557, 595, 1049, 1201, 1343, 1484. Se aprecian mediante líneas verticales discontinuas de color rojo.

Sólo se produce un error, y es en el *frame* 1242 (pico azul más alto sin tener línea roja discontinua encima), en el cual no se produce ningún cambio de toma. Si se fija uno detenidamente ocurre algo raro en los *frames* adyacentes, a partir de 3 *frames* anteriores (es decir, desde el 1239) empieza a incrementarse el número de macrobloques I en el *frame*. Si se

analiza con detenimiento el video en cuestión (ver figura "*Fig. 5-16*"), se ve que la toma es la misma pero desapareciendo un rótulo en el *frame* 1238. Aún así, la imagen es muy parecida (en ese momento consiste en una toma del cielo, en un día más o menos despejado) y, sin embargo, se producen cambios muy bruscos del número de macrobloques I o P en dichos *frames*. Es muy posible que se deba a una mayor iluminación del sol en los últimos *frames*. A continuación se adjuntan las imágenes:

FIG. 5-16 FRAMES: 1234-1241 VÍDEO "FRAGMENT0.H264"

Al margen de este error, el mecanismo parece bastante fiable. Acierta en todos los cambios de toma, y añade este último que realmente no es un cambio de toma.

Resumiendo, se pueden ver los resultados en función del *Recall* y la Precisión, como he visto anteriormente:

Recall = 100%

Precision = 90'90 %

Ejemplo 2: fragment1.h264

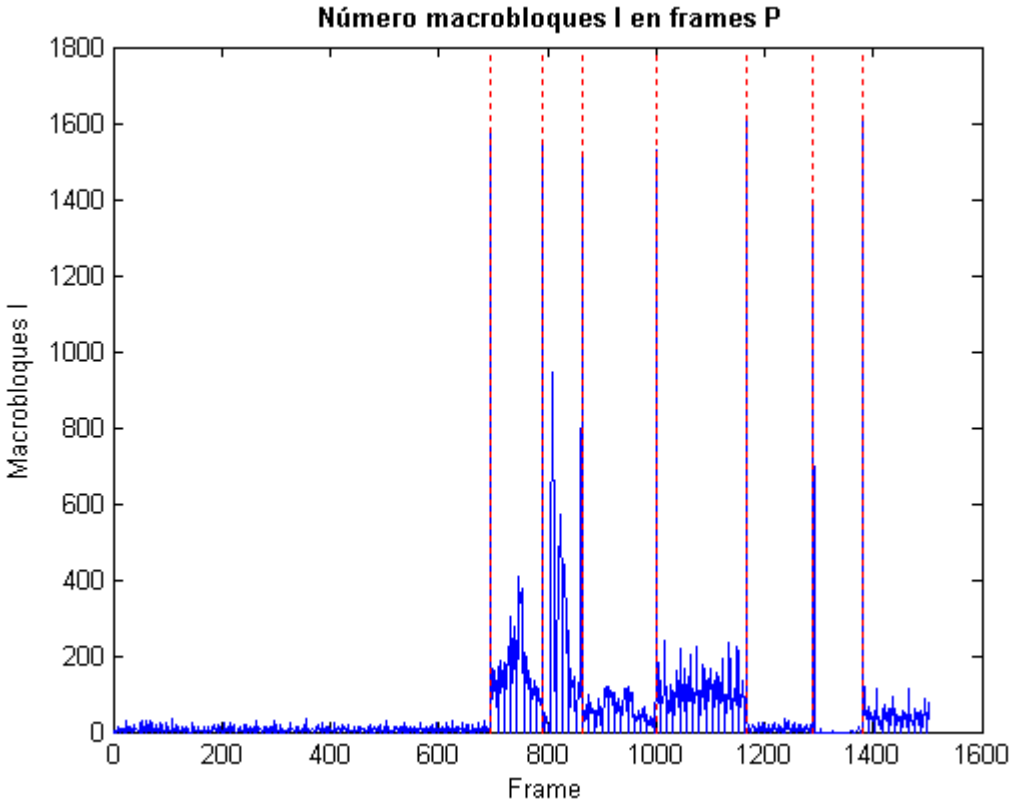


FIG. 5-17 NÚMERO MB'S I VÍDEO "FRAGMENT1.H264"

Al igual que en el ejemplo anterior, el tamaño en macrobloques de todos los *frames* de esta secuencia es: 45x36 = 1620.

Realmente se producen los cambios de toma en los *frames*: 695, 791, 863, 1001, 1166, 1290, 1381. Esto se puede apreciar en la figura "*Fig. 5-17*", donde se ve el número de macrobloques I en dichos *frames*, poniendo a 0 los macrobloques I en los *frames* I y remarcando los cambios de toma mediante líneas verticales discontinuas de color rojo, como se veía en el ejemplo anterior. En este ejemplo acierta en un 100% de los casos.

Como se pudo ver en el ejemplo 3 del apartado "*5.3.1 Cambios de toma*", en el *frame* 828 se produce la aparición repentina de un niño en la escena, propiciando un posible error de detección de cambio de toma. Aún así, el número de macrobloques I en dicho *frame* está lejos del valor que coge cuando realmente se produce un cambio de toma, que suele ser del 90% o más del número total de macrobloques (en este caso 1620).

Resumidamente,

Recall = 100%

Precision = 100 %

5.4.2 CAMBIOS DE TOMA – SECUENCIAS IBPBP…IBPBP…

A partir de ahora introduciremos una novedad respecto a las pruebas que se han venido realizando: se trata de añadir *frames* de tipo B (predicción Bi-direccional). Así, ahora habrá *frames* tanto de tipo P como de tipo B entre los *frames* I. Así mismo, se harán pruebas insertando 1 o 2 *frames* tipo B entre dos *frames* I ó P. No se llevan a cabo pruebas con más *frames* B, porque en la realidad no existen vídeos de este tipo, por su poca utilidad práctica.

Recordar que los *frames* de tipo I sólo pueden tener macrobloques I, *frames* P tendrán macrobloques tipo I y P, y los *frames* B tendrán macrobloques I y B. A su vez, los macrobloques B pueden referenciarse a *frames* anteriormente y/o posteriormente codificadas. De esta manera, si sólo utilizan como referencia un *frame* previo en el tiempo, en el orden de reproducción (predicción *forward*) tendrá la misma función que un *frame* P.

Ejemplo: fragment0_1b.h264

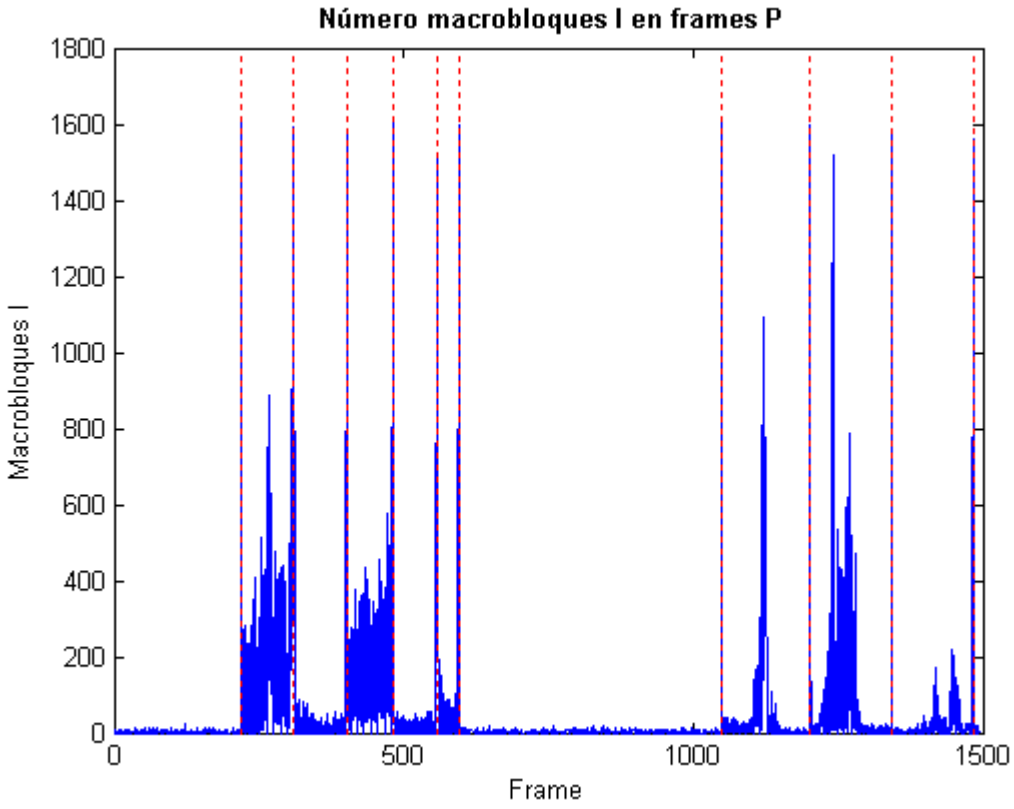


FIG. 5-18 NÚMERO MB'S I VÍDEO "FRAGMENT0_1B.H264"

Al igual que en el ejemplo análogo anterior, se dispone del mismo tamaño en macrobloques, siendo un total de 1620 macrobloques. Así, se vuelve a analizar lo mismo que en ejemplos anteriores. Se buscan cambios radicales del número de macrobloques I en *frames* P o B, impidiéndonos esto encontrar un cambio de toma que se produzca en un *frame* I por lo mencionado anteriormente. Dicho de otra manera, se busca que el número de macrobloques I en un *frame* P o B esté cerca del 100%.

Los cambios de toma se detectan en los *frames*: 219, 311, 401, 481, 557, 595, 1049, 1201, 1343, 1483.

Como se puede ver en la figura "*Fig. 5-18*" se produce además una detección falsa de un cambio de toma en el *frame* 1241. Este error es exactamente el mismo que el producido en el mismo ejemplo pero sin *frames* B (*Ejemplo 1* del apartado "*5.4.1 Cambios de toma – secuencias IPP…IPP…*"). A continuación se habla de dicho error.

Tras analizar los puntos donde se producen los cambios de toma, a partir del número de macrobloques I en todos los *frames*, se pueden sacar algunas conclusiones:

- Se siguen detectando los mismos cambios de toma que en el mismo ejemplo sin *frames* B, incluida la falsa detección que se producía en el *frame* 1240 que habíamos concluido que se podía deber a un cambio de iluminación.

- Aparece un pequeño inconveniente, el cual no afecta en mayor medida a los resultados a buscar. Se trata del número de macrobloques I detectados en los *frames* P, los cuales se ven muy incrementados cuando no hay cambios de toma. En ocasiones, llegan a tener algo más de 1000 MB's de tipo I en dichos *frames* de un total de 1620 (aproximadamente un 63%), lo cual es un número bastante alto, aunque todavía lejos de los casi 1600 que aparecen cuando se trata de un cambio de toma (en torno al 98%). Se puede suponer que esto se debe a que, al haber predicción *forward* y *backward*, los bloques serán resultantes de una predicción más fuerte y, las muestras son reconstrucciones menos fiables de las originales. Así, aparecen más macrobloques intra-codificados.

- La aparición de *frames* B en la secuencia genera un cambio en el orden de codificación de los diferentes *frames*, debido a poder utilizar dichos *frames* predicción *backward*. De esta manera, los *frames* que usen como referencia frames que se presentan después, harán que estos frames que se presentan después se codifiquen antes para poder realizar una predicción adecuada. El hecho de cambiarse el orden de codificación no afecta al orden de presentación. Los cambios de toma siguen produciéndose en el mismo *frame*, pero no aparecen en el mismo número de *frame* en los resultados que se han expuesto debido a que se representa el orden de codificación, pues es como trabaja el codificador/decodificador JM en el punto donde se ha creído óptimo sacar la información.

5.4.3 CAMBIOS DE TOMA – SECUENCIAS IBBPBBP…IBBPBBP…

Ejemplo: fragment0_2b.h264

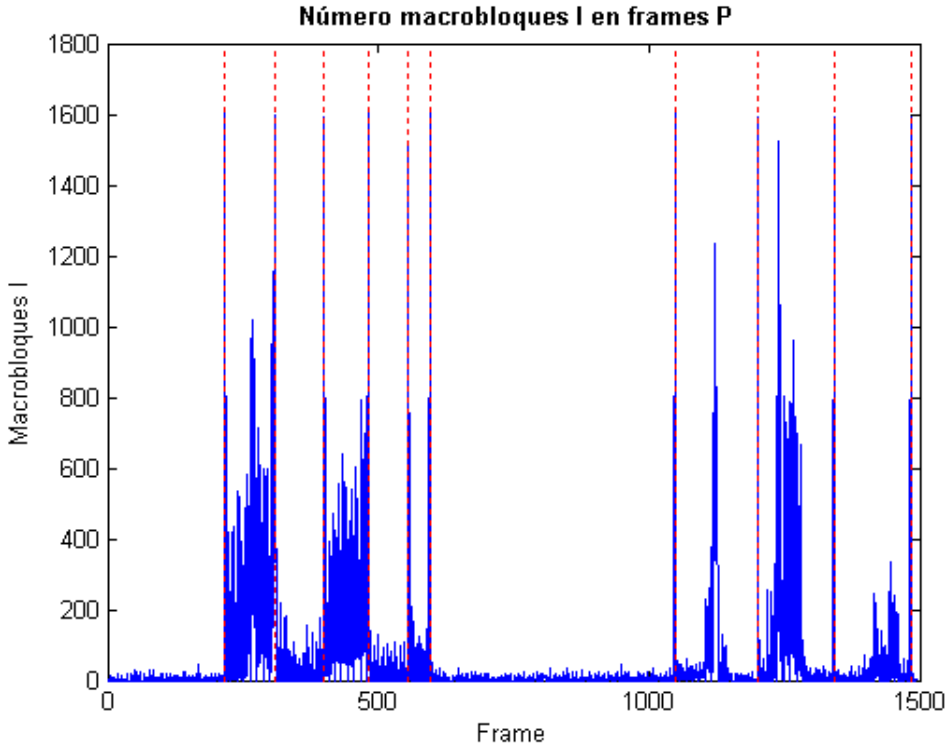


FIG. 5-19 NÚMERO MB'S I VÍDEO "FRAGMENT0_2B.H264"

A partir de ahora se realizará exactamente lo mismo, pero intercalando 2 *frames* B entre *frames* I ó P consecutivos. Los resultados de este ejemplo se pueden ver gráficamente en la figura "*Fig. 5-19*".

Cambios de toma detectados en *frames*: 217, 310, 400, 481, 556, 595, 1048, 1201, 1240, 1342, 1483.

Cambios de toma reales en *frames*: 217, 310, 400, 481, 556, 595, 1048, 1201, 1342, 1483.

En relación a estos resultados, se observa:

- Los cambios de toma no suceden exactamente en el mismo número de *frame* debido al distinto orden de codificación, como se mencionó en el ejemplo del subapartado anterior.

- Vuelven a aparecer muchos *frames* P con un número alto de macrobloques I, aunque nunca se llega a la confusión sobre si se trata o no de un cambio de toma porque nunca se acerca a 1500, 1600 macrobloques I (~90% o más de los 1620 macrobloques de los *frames*).

- Otra vez se produce el error en el *frame* 1240 debido al cambio de iluminación. Lógicamente, si se producía en vídeos sin *frames* B, con *frames* B no se van a conseguir mejores resultados en este aspecto (sí quizás en otros aspectos).

5.4.4 CAMBIOS DE TOMA – INFLUENCIAS DE LA RESOLUCIÓN

Ejemplo: fragment0.h264 (cambio de resolución)

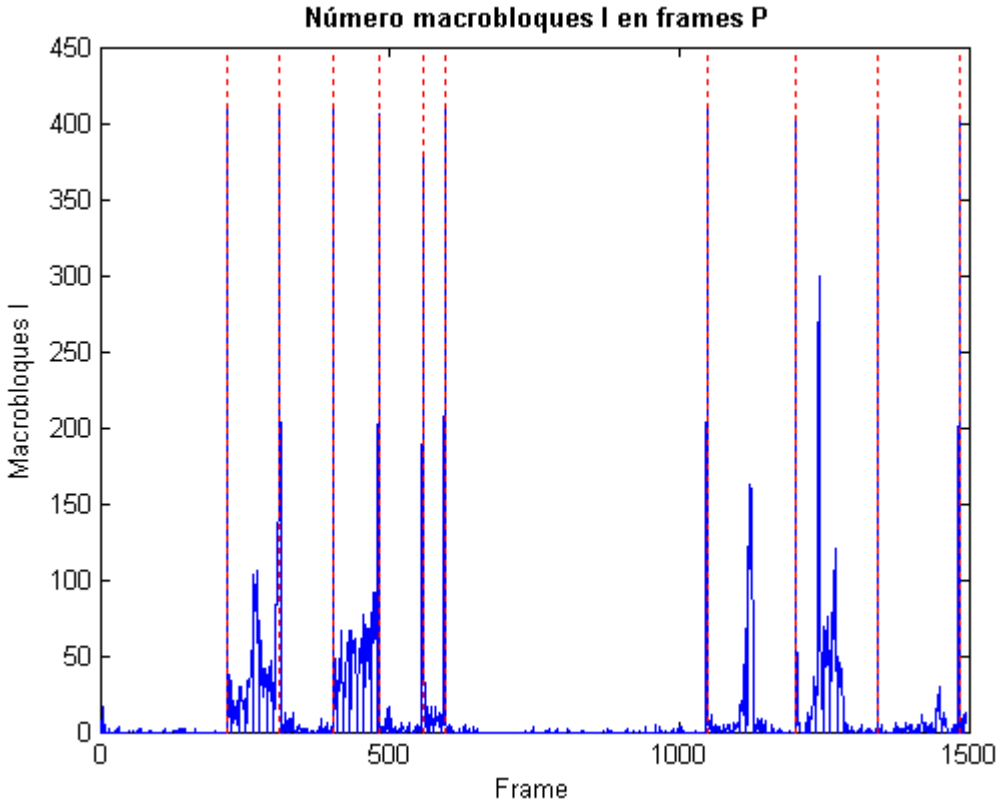


FIG. 5-20 NÚMERO MB'S I VÍDEO "FRAGMENT0.H264" CAMBIO DE RESOLUCIÓN

En este ejemplo se procederá a hacer lo mismo pero con un cambio de resolución. Pasando de 720x576 a 368x288, es decir, reduciendo la resolución a la cuarta parte.

Como se ha visto ya en el caso en "5.3.2 Detección de contornos", cambiar la resolución afecta en gran medida a la consecución de nuestros objetivos. De esta manera, se exponen los resultados obtenidos con el mismo vídeo que en el *ejemplo 1* del apartado "*5.4.1 Cambios de toma – secuencias IPP…IPP…*" con el mencionado cambio de resolución.

Primero, comentar que las nuevas dimensiones del vídeo en macrobloques son 23x18, es decir, que hay 414 macrobloques en total en cada *frame*, cuando anteriormente teníamos 1620 macrobloques. Esto, a priori, lo único que puede hacer es dificultar la distinción entre la presencia o no de un cambio de toma en la secuencia de vídeo, pues hay menos macrobloques para comparar. Aún así, las proporciones solían estar en el 90%, así que, a pesar de este inconveniente, se cree que a partir de los resultados se debería todavía poder distinguir los cambios de toma presentes. Como se ve en la gráfica de la figura "*Fig. 5-20*", se pueden seguir distinguiendo los cambios de toma con relativa facilidad.

Veamos diferentes casos, del mismo modo que en ejemplos anteriores, destacando los *frames* que parezcan relevantes.

Los cambios de toma se producen en los *frames*: 219, 311, 402, 481, 557, 595, 1049, 1201, 1343, 1484.

- A partir del cambio de toma producido en el *frame* 311, se ven varios *frames* P en los que la cuarta parte o incluso la mitad son macrobloques I, pasando de valores en torno a 20, 30 (sobre 1620 en total), a valores próximos a 150, 200 (sobre 414 en total). Aún así, sigue lejos del 90% de macrobloques I en *frames* P. Esto se debe al tipo de escena y a haber menor resolución. Al verse reducida la resolución a la cuarta parte es difícil encontrar zonas homogéneas en *frames* como el 377, que presentamos en la figura "*Fig. 5-21*".

FIG. 5-21 FRAME 337 VÍDEO "FRAGMENT0.H264" RESOLUCIÓN 368X288

- Después del cambio de toma que se produce en el *frame* 311, se vuelve a los valores iniciales habiendo como mucho 10 macrobloques I en *frames* P.

- En los *frames* 1120-1124 la cantidad de macrobloques I asciende a 150 aproximadamente, sin llegar tampoco al 50% de macrobloques I (por tanto, se puede deducir que no se trata de un cambio de toma) aunque se llega a un valor significativo. Se debe al caso de la jarra de cerveza presente en el vídeo, por eso hay tanta zona homogénea en el vídeo ("*Fig. 5-22*").

FIG. 5-22 FRAME 1118 VÍDEO "FRAGMENT0.H264" RESOLUCIÓN 368X288

- En los *frames* 1239-1243 ocurre lo que se ha visto en ejemplos anteriores: el problema de la luminosidad de la imagen. Ya con valores próximos al 70% de macrobloques I, sigue sin llegar al 90% pero ya son datos que podrían interpretarse como cambio de toma.

Resumidamente, se puede ver que una reducción de la resolución provoca que la extracción de información pierda fiabilidad, pero en este caso sigue siendo una forma muy adecuada de detectar cambios de toma. Afectar en menor grado que en la detección de contornos.

5.5 DIRECCIÓN DE PREDICCIÓN DE LOS MACROBLOQUES

Como se ha mencionado en apartados anteriores, el análisis de la dirección sobre la que se realiza la predicción de los vectores de movimiento en los distintos macrobloques de un *frame,* puede dar información muy valiosa acerca de la detección de cambios de toma abruptos.

Se puede llegar a esta conclusión, sabiendo que el último *frame* antes de producirse un cambio de toma no tendría apenas predicción *backward* porque no encontraría correspondencia o bloques muy parecidos con respecto a *frames* posteriores, de la nueva toma o escena. De la misma manera, el primer *frame* de una nueva escena apenas tendrá similitudes en bloques para predecirse con *frames* anteriores que pertenecen a la escena anterior, predicción *forward*.

Para verlo con la mayor claridad posible, se ha utilizado un GOP con la estructura IBBPBBP…IBBPBBP… con aparición de *frames* I de manera periódica. Se ha usado la herramienta *VISUALmpegAVC* para observar la utilidad de este parámetro de una manera global.

Sabemos por el estándar H.264 que, generalmente, los vectores de movimiento en cada *macrobloque* se refieren a los macrobloques de los *frames* almacenados en el DPB (*Decoded Picture Buffer*) a los que se apuntan a partir de las listas 0 y 1 que se veía en el apartado "3.1.2 Slices". La lista 0 apunta a los *frames* almacenados del DPB anteriores en el orden de presentación al que se encuentra codificándose, y la lista 1 apunta a *frames* posteriores en dicho orden de presentación. De esta manera, los vectores de movimiento de un macrobloque resultantes de la predicción de la lista 0 implican predicción *forward*, y los de la lista 1 implican predicción *backward*, por norma general.

Es lógico pensar que, a pesar de cambiar la estructura del GOP (*Group of Pictures*) de un vídeo, el cambio de toma se seguirá produciendo en el mismo número de *frame* cuando se habla de orden de presentación, de reproducción. De este modo, si la estructura de un vídeo es IPPP…IPPP, este parámetro deja de tener validez, al menos en gran parte. El último *frame* de una escena, en caso de ser de tipo P, seguirá teniendo predicción *forward* pero no se podría observar si ha dejado de tener o no predicción *backward* (en este *frame* ya no debería tener por lo recién mencionado) porque los *frames* P no utilizan este tipo de predicción. Sí se podrá seguir viendo que el primer *frame* de una escena deja de tener predicción *forward* y, dado que tampoco puede tener predicción *backward*, con seguridad aparecerá un *frame* P con su mayoría macrobloques intracodificados, tal y como se ha visto en la extracción de dicho parámetro en el apartado "*5.4 Número de macrobloques I/P-B*".

No todos los cambios de toma se detectan igual con este parámetro, pues algunos se producen en *frames* I, P ó B. Siempre se podrá observar algo porque consiste en ver qué ocurre en el mismo *frame* y los contiguos. Pero también es cierto que, si ocurre un cambio de toma en un *frame* I, la información de dicho *frame* no dirá nada (al menos en relación a este parámetro). Sí puede decir si el *frame* anterior tiene o no predicción *backward*, el cual no debería tener en caso de ser un *frame* de tipo B, si es P tampoco tendríamos información útil.

Dicho de otro modo, la información que puede proporcionar la extracción de este parámetro puede ser más o menos eficaz, dependerá del momento en el que se produzca el cambio de toma y de la estructura de GOP del vídeo en cuestión.

Se verán a continuación algunos ejemplos en los que se resaltan todos estos casos de una manera gráfica. De un marcado interés será lo que ocurre antes y después del cambio de toma en relación a la dirección de predicción de los vectores de movimiento de los diferentes macrobloques de dichos *frames*.

Vídeos utilizados:

TABLA 5-5 VÍDEOS CAPÍTULO 5.5

	Archivo	Resolución	Duración	Nº Frames	GOP
1	fragment0.h264	720x576	59	1496	IPPP…
2	fragment0.h264	720x576	59	1496	IBPBP…
3	fragment0.h264	720x576	59	1496	IBBPBBP…
4	fragment1.h264	720x576	59	1504	IPPP…
5	fragment0.h264	368x288	59	1496	IPPP…

Ejemplo 1: Frames 1480-1487 vídeo "fragment0.h264"

El número de *frame* se corresponde con el orden de codificación, así un número de *frame* mayor significa que se ha codificado/decodificado posteriormente, pero esto no se corresponde con el orden de presentación cuando aparecen *frames* B, como en este caso. Por supuesto, los *frames* B pueden tener predicción *forward* y/o *backward* y los *frames* P solamente predicción *forward*. Se añaden un par de *frames*, los contiguos a donde se produce el cambio de toma (en el frame 1485), para observar lo que ocurre en condiciones normales en el resto de frames. En los siguientes ejemplos solo se señalarán lo que ocurre exactamente en los cambios de toma.

TABLA 5-6 ORDEN CODIFICACIÓN FRAMES 1480-1487 VÍDEO "FRAGMENT0.H264"

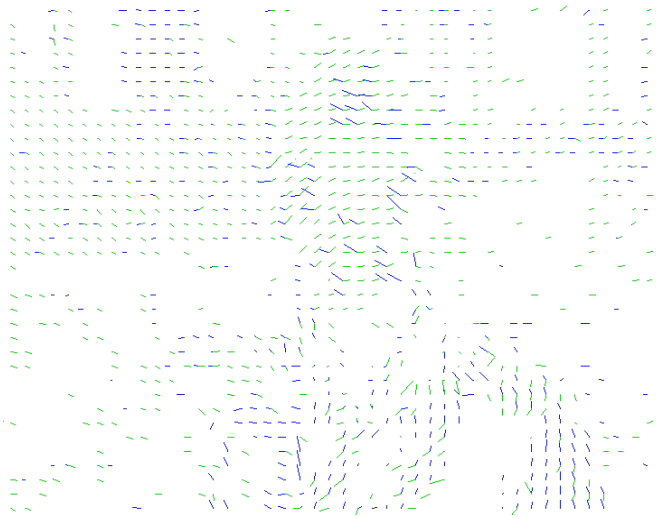
Orden Codificación		
Nº Frame	Tipo Frame	Etiqueta
1480	P	P1
1481	B	B1
1482	B	B2
1483	P	P2
1484	B	B3
1485	**B**	**B4**
1486	P	P3
1487	B	B5

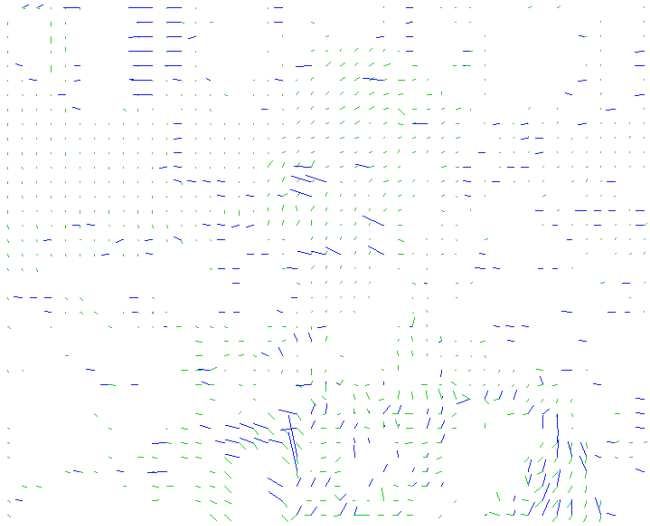
* En negrita donde se produce el cambio de toma

Siendo el orden de presentación, según las etiquetas:

B1 → B2 → P1 → B3 → B4 → P2 → B5 → P3

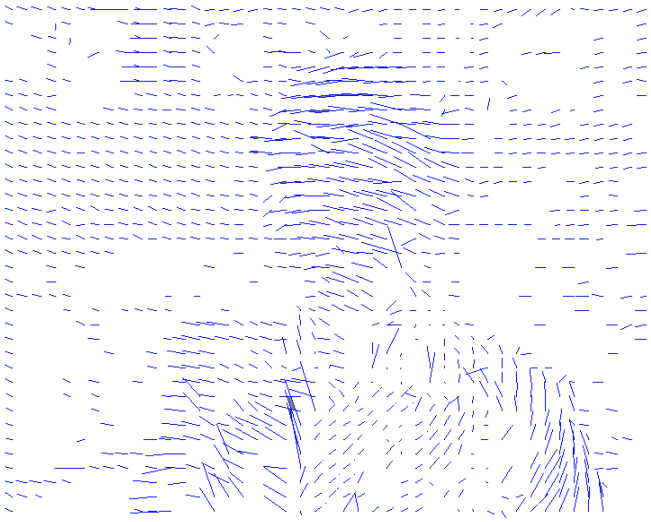
Se muestran a continuación los diferentes *frames* (en el orden de presentación) con sus respectivos vectores de movimiento (*forward* en azul, *backward* en verde).

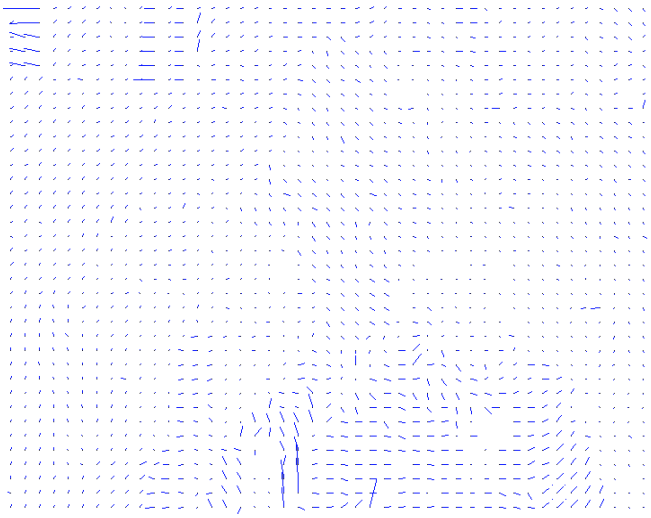




B1

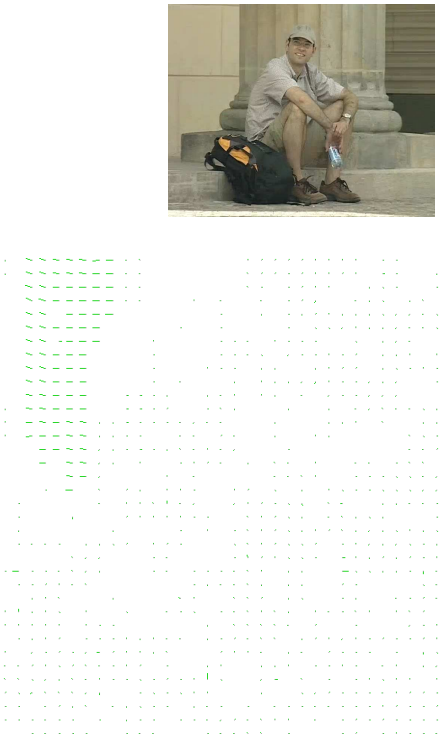
B2



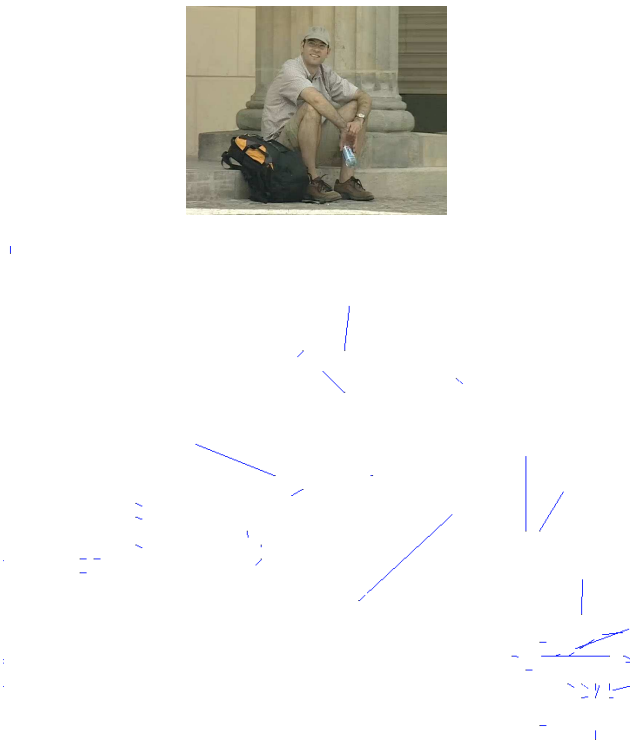


P1

B3



B4



P2

B5

P3

FIG. 5-23 VECTORES MOVIMIENTO FRAMES: 1480-1487 VÍDEO "FRAGMENT0.H264"

En las imágenes de la figura "*Fig. 5-23*" se ve cómo en los *frames* P solamente aparecen vectores de movimiento con predicción *forward*. En el *frame* con etiqueta P2 se aprecia que apenas tiene vectores de movimiento, esto se debe a que justo anteriormente se producía el cambio de toma y, de hecho, sirve como referencia del *frame* con etiqueta B3, el cual se presenta antes pero se codifica después por las características de codificación de H.264. Estos vectores de movimiento apuntan a bloques de *frames* anteriores, los cuales pertenecen a otra escena, así pues introducirán un pequeño error, imperceptible para el SVH (Sistema Visual Humano).

Por otro lado, B1 y B2 son *frames* tipo B y aparecen efectivamente vectores de movimiento con predicción *forward* y *backward*. El problema llega en B3, el cual es el último *frame* de la escena. Así, no tiene predicción *backward* como se ha explicado anteriormente. Lo mismo ocurre con B4, primer *frame* de la nueva escena, el cual no tiene predicción *forward*. El siguiente frame de este tipo, B5, ya vuelve a tener ambos tipos de predicción.

Ejemplo 2: Frames 307-311 vídeo "fragment0.h264"

TABLA 5-7 ORDEN CODIFICACIÓN VÍDEO "FRAGMENT0.H264"

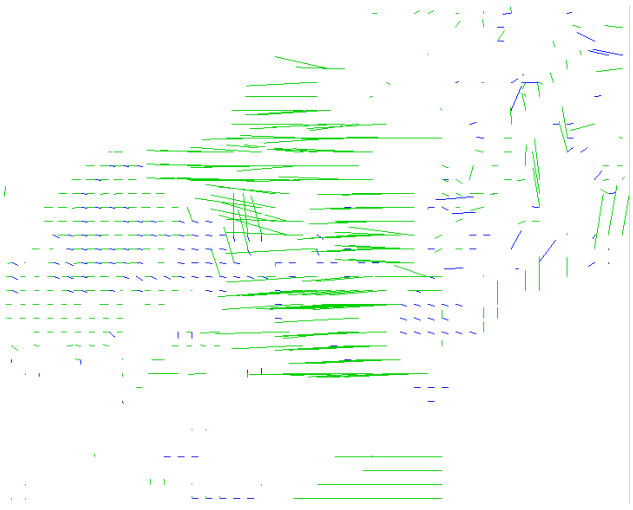
Orden Codificación		
Nº Frame	Tipo Frame	Etiqueta
307	P	P1
308	B	B1
309	B	B2
310	**P**	**P2**
311	B	B3

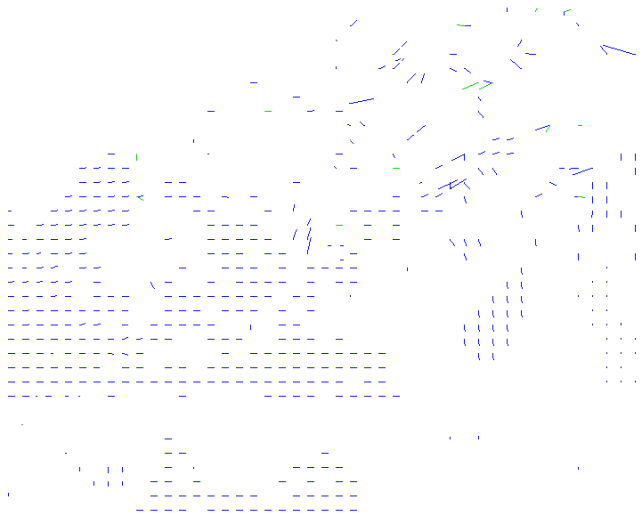
*En negrita donde se produce el cambio de toma

Siendo el orden de presentación, según las etiquetas:

B1 → B2 → P1 → B3 → P2

En este caso solo se presentan *frames* que consideramos representativos, por no hacer muy densa la presentación de resultados.





B2

B3

P2

FIG. 5-24 VECTORES MOVIMIENTO FRAMES 307-311 VÍDEO "FRAGMENT0.H264"

En P2 ("*Fig. 5-24*") se produce el cambio de toma, por tanto apenas aparecen macrobloques con predicción *forward*, solamente aparecen en el rótulo a la altura del rótulo de la imagen, dado que también aparece en la escena anterior. Se ve cómo en B3, el *frame* anterior al cambio de toma, apenas aparece predicción *backward*, y sí aparece en B2, el cual no estaba ni antes ni después de un cambio de toma.

Ejemplo 3: Frames 862-866 vídeo "fragment1.h264"

TABLA 5-8 ORDEN CODIFICACIÓN FRAMES 862-866 VÍDEO "FRAGMENT1.H264"

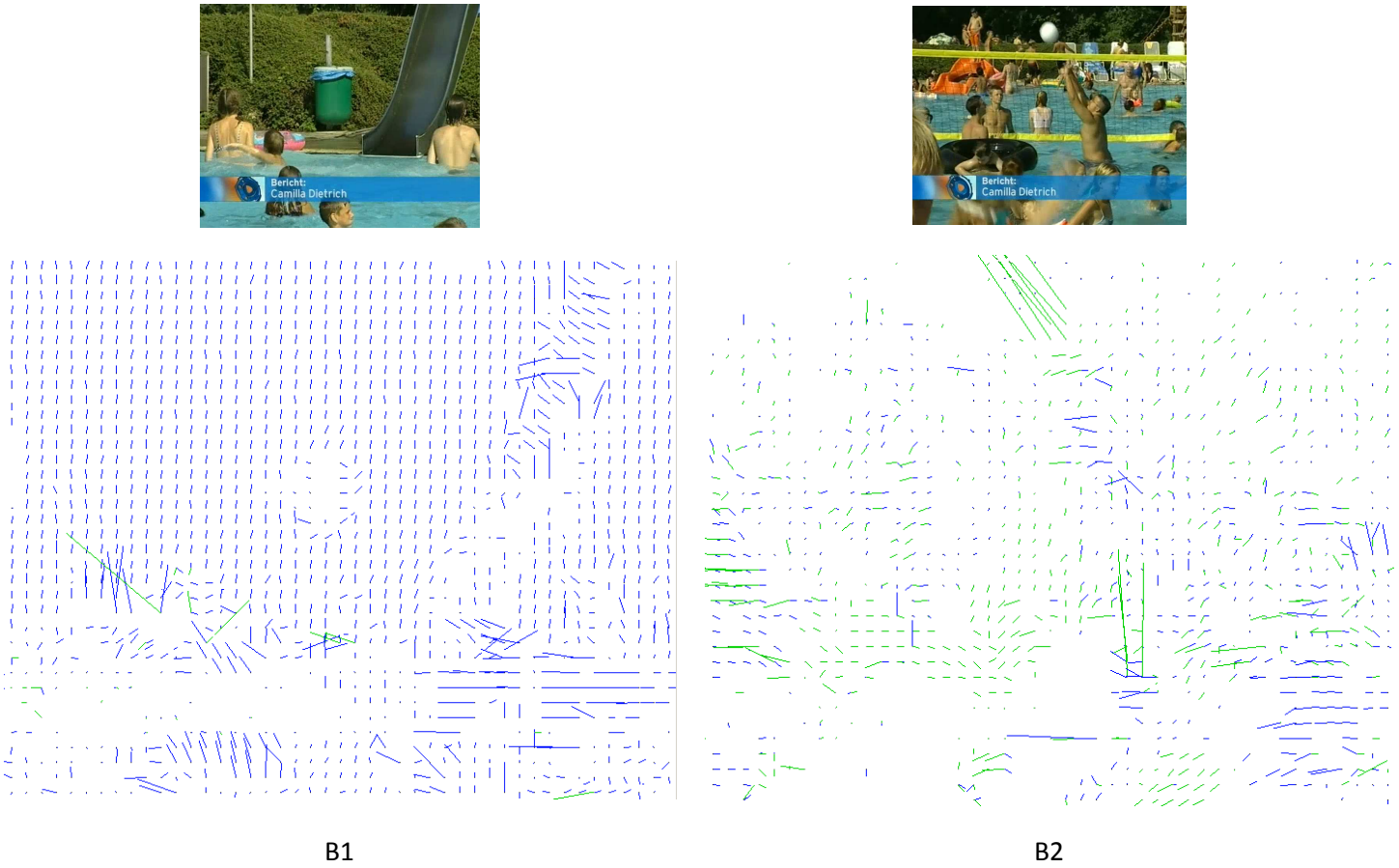
Orden Codificación		
Nº Frame	Tipo Frame	Etiqueta
862	**P**	**P1**
863	B	B1
864	I	I
865	P	P2
866	B	B2

*En negrita donde se produce el cambio de toma

Siendo el orden de presentación, según las etiquetas:

B1 → P1 → I → B2 → P2

El *frame* B1 apenas tiene predicción *backward* al ser el último de la primera toma. Sin embargo, P2 y B2 no tienen ya ningún problema en encontrar bloques sobre los que predecir, puesto que hay un *frame* I antes de su presentación y justo después del cambio de toma. Se ve continuación en la figura "*Fig. 5-25*".



B1 B2

FIG. 5-25 VECTORES MOVIMIENTO FRAMES 862-866 VÍDEO "FRAGMENT1.H264"

5.6 VECTORES DE MOVIMIENTO

Como se ha visto en secciones anteriores, el método de trabajo con los vectores de movimiento no ha variado con respecto a estándares anteriores. Sí ha cambiado, sin embargo, la propiedad de multirreferencia (no existente en estándares anteriores), y el hecho de los tamaños de macrobloque como se comentó en el apartado "3.1.3 Codificación Temporal".

Las funcionalidades que se pueden dar a la extracción de los vectores de movimiento son las mismas que en trabajos ya realizados acerca de la segmentación de objetos.

Adicionalmente, comentar que la funcionalidad en este aspecto a través de la extracción de una imagen junto con sus vectores de movimiento también tiene sus restricciones. Y existen una cantidad prácticamente ilimitada de trabajos con sus correspondientes algoritmos para superar todas estas restricciones.

De esta manera, para poder hacer una buena segmentación de objetos se necesita que la cámara esté fija. Si no es así, los vectores de movimiento aparecerán en todas partes de la imagen pues, por ejemplo, el árbol que formaba parte del background de la imagen ya no se encuentra exactamente en el mismo sitio, ha variado ligeramente su posición (debido al movimiento de la cámara), provocando así la aparición de un vector de movimiento para una óptima codificación posteriormente, pues se busca el bloque más parecido al que se esté codificando. En cuanto a esto, existen algoritmos para detectar el movimiento global de la cámara y poder así realizar una discreción correcta entre el vector de movimiento debido al movimiento de la cámara y el debido al movimiento real de lo que haya presente en la imagen, consiguiendo distinguir con mucha mayor facilidad los objetos o personas que se hayan movido realmente en la escena.

Si, afortunadamente, la cámara está fija, los únicos vectores de movimiento que aparecerán serán los de aquellas personas u objetos de la escena que estén moviendo y es ahí donde se puede hacer una segmentación de objetos.

A continuación se verán algunas imágenes extraídas del VISUALmpegAVC donde se podrán analizar dos situaciones: uno con poco movimiento entre el mismo *frame* y el anterior, y otro con un mayor movimiento. Obviamente se verá una clara diferenciación en el número de vectores de movimiento con valor no nulo (con algo de movimiento con respecto al bloque más parecido del *frame* al que hace referencia) y en la magnitud de dichos vectores de movimiento (la distancia entre el bloque actual y el bloque al que hacen referencia sus vectores de movimiento).

Adicionalmente, se incluye el mapa de vectores para el mismo *frame* pero para la secuencia correspondiente en formato .mpg (MPEG-1). Salvando las diferencias insalvables entre el vídeo con un formato de compresión MPEG-1 y otro con formato H.264, se ha procurado que las propiedades comunes fueran iguales, tales como la longitud del GOP (longitud 12), el mismo tamaño de la ventana de búsqueda y una búsqueda exhaustiva del bloque más parecido. Representar estos dos mapas de vectores sobre el mismo *frame* pero en distintos formatos de compresión tiene la única intención de poder observar ciertas diferencias destacables de las que ya se han hablado en apartados anteriores y su utilidad.

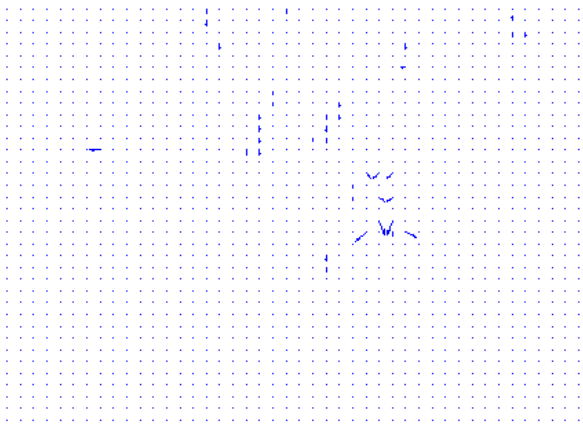
Los vectores de movimiento se extraen directamente del *stream* de vídeo con extensión .mpg mediante una aplicación desarrollada en C. Los vectores de movimiento se extraen de los vídeos con extensión .h264 mediante la aplicación de vídeo *VISUALmpegAVC*. Comentar también que en la primera representación de vectores (vídeos .mpg) se representan todos los vectores de movimiento, incluidos los que tienen valor nulo (estos se representan mediante un punto). No ocurre así con la segunda representación de los mismos (.h264), donde no se representan los vectores de movimiento nulos.

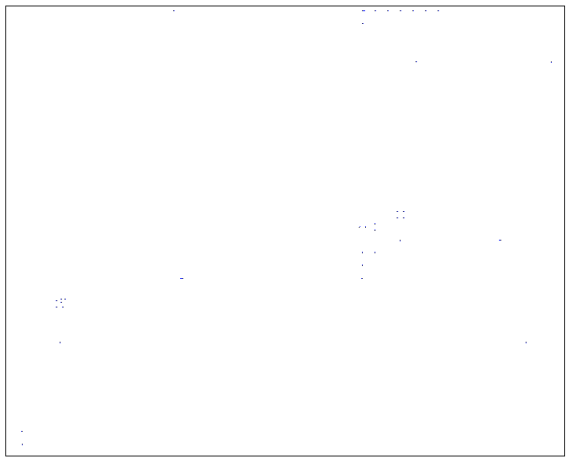
Vídeo utilizado:

TABLA 5-9 VÍDEO CAPÍTULO 5.6

	Archivo	Resolución	Duración	Nº Frames	GOP
1	fragment0.h264	720x576	59	1496	IPPP…

Caso 1: Frame 816 vídeo "fragment0"





.mpg .h264

FIG. 5-26 VECTORES DE MOVIMIENTO MPEG Y H.264, FRAME 816, VÍDEO "FRAGMENT0"

En la figura "*Fig. 5-26*", se ve cómo los vectores de movimiento tanto en un mapa de vectores como en el otro apenas tienen valores no nulos y, en caso de haberlos, su valor no es demasiado alto. Esto se debe a un movimiento general muy pequeño con respecto al *frame* anterior. En este ejemplo es muy complicado observar diferencias entre los vectores de movimiento de MPEG-1 y H.264.

Caso 2: Frame 268 vídeo "fragment0"

.mpg .h264

FIG. 5-27 VECTORES DE MOVIMIENTO MPEG Y H.264, FRAME 268, VÍDEO "FRAGMENT0"

Esta imagen (figura "*Fig. 5-27*") consiste en las vistas de un paisaje grabadas desde un coche. Así, el movimiento implícito de la cámara es el del coche. Se ha seleccionado esta imagen porque se ve muy claramente la influencia del movimiento de la cámara en los vectores de movimiento correspondientes a los objetos del paisaje. En primer lugar, se aprecia que el rótulo, como acaba de aparecer en la escena, se intracodifica (macrobloques de tipo I dentro de un *frame* P), de este modo no tiene vectores de movimiento correspondientes. Se puede suponer que ni los arbustos, ni el árbol, de la imagen se mueven. En todo caso se podrían mover debido al viento, pero dicho movimiento sería mínimo comparado con el del coche. Así, en esta imagen sería muy fácil detectar un movimiento global del coche pudiendo, posteriormente, suprimir dichos vectores debido a este movimiento.

Resumidamente, se pueden ver unos vectores de movimiento que dan información sobre el movimiento de lo presente en la escena, pudiendo así distinguir cualquier cosa siempre que sea fácil diferenciar el background del *foreground*. Además, se pueden detectar de alguna manera la diferencia entre los vectores de movimiento que se extraen del vídeo con extensión .mpg del que tiene extensión .h264. Si se observa con detenimiento, las zonas más texturizadas de la imagen se corresponden con una mayor afluencia de vectores de movimiento en el mapa correspondiente a .h264. En un simple macrobloque pueden aparecer varios vectores de movimiento, como en algunos puntos a derecha e izquierda del árbol, que presumiblemente corresponden a los árboles del fondo de la imagen. Ésta es una característica, una novedad, del estándar H.264 con respecto a los estándares anteriores.

5.7 MODOS DE PREDICCIÓN

Como se ha podido ver en secciones anteriores, también es posible sacar cierta información de interés a partir de los modos de predicción en macrobloques Intra. Para ello, se analizan los *frames* Intra que aparecen en un vídeo.

Hay dos tipos de macrobloques Intra: 16x16 o 4x4, teniendo cada uno de ellos distintos tipos de modos de predicción, cuatro distintos para los macrobloques 16x16 y nueve para los macrobloques 4x4, tal y como se ha visto más detalladamente en el apartado *"3.1.4* Codificación Espacial*"*.

En el apartado "5.3 Tipos de macrobloque en frames Intra (I)" se ha visto cómo detectar, bien cambios de toma, bien contornos de objetos/personas, a través simplemente del número de macrobloques de un tipo u otro en un *frame* o de la disposición de dichos macrobloques en el *frame*, respectivamente.

Ahora, se analizan los modos que utiliza el codificador/decodificador, y a partir de ellos se intentará ver si esto puede dar una pista o no sobre la detección de cambios de toma.

5.7.1 CAMBIOS DE TOMA

Estos modos de predicción se pueden utilizar con cierta eficacia para detectar cambios de toma. Lo que se hace es analizar la cantidad de veces que salen los distintos modos de cada tipo de macrobloque, así como su variación con respecto al *frame* I anterior.

Para ello, se han calculado y representado gráficamente dos tipos de parámetros como se ve a continuación:

- El primero consiste en ver la cantidad de veces que salen los nueve modos de predicción de los macrobloques Intra con estructura 4x4. Se ha decidido utilizar estos modos, y no los cuatro modos pertenecientes a los macrobloques con estructura 16x16, debido a que salen en una cantidad mayor los primeros, puesto que de cada macrobloque 4x4 salen cuatro modos. En cambio, de los macrobloques con estructura 16x16 solamente sale un modo por macrobloque.

- En el segundo lo que se hace es calcular una diferencia total aproximada (sobre todos los modos de predicción) también con respecto al *frame* I anterior. Este cálculo es el que refleja Wei Zeng en [43]. Este parámetro se representa gráficamente, y se le denominado "Diferencia Wei Zeng", que es la diferencia total que se deduce de la fórmula del trabajo referenciado anteriormente. Es importante que esta diferencia sea superior a cierto umbral (aquí no se trabaja con dicho umbral, simplemente se observa qué ocurre y cuáles son estos valores) y que sea un pico, es decir, que los dos *frames* adyacentes al *frame* que se está analizando no superen este umbral, que no tengan un valor tan alto como el del *frame* a analizar. Se podrá apreciar que esta diferencia suele oscilar entre los valores 0 y 40 aproximadamente. Así, picos con valores a partir de 80, 90 empiezan a ser relevantes, aunque hay que tener cuidado con los valores de los *frames* contiguos. A la hora de representar en la gráfica, dado que se trata de una línea continua, no será tan evidente observar si los picos que observamos son tal o tienen

los valores de los *frames* contiguos también altos y luego bajan. Se remarcarán dichos valores donde no se vea claro.

Se exponen un par de ejemplos para observar qué es lo que ocurre, con sus consecuentes gráficas. Ambas gráficas deben tomarse como complementarias la una de la otra. Es decir, para poder intentar extraer conclusiones a partir de dichas gráficas, conviene que se observen ambas gráficas.

Vídeos utilizados:

TABLA 5-10 VÍDEOS CAPÍTULO 5.7

	Archivo	Resolución	Duración	Nº Frames	GOP
1	fragment0.h264	720x576	59	1496	IPPP…
2	fragment1.h264	720x576	59	1496	IPPP…

Ejemplo 1: fragment0.h264

A continuación se exponen gráficamente, en primer lugar, la cantidad de veces que salen cada uno de los nueve modos correspondientes a los macrobloques I con estructura 4x4 en *frames* I y, en segundo lugar, la diferencia "Wei Zeng" que mencionada anteriormente. Se remarcan las líneas verticales, rojas y discontinuas dónde se producen los cambios de toma para verlo con mayor facilidad.

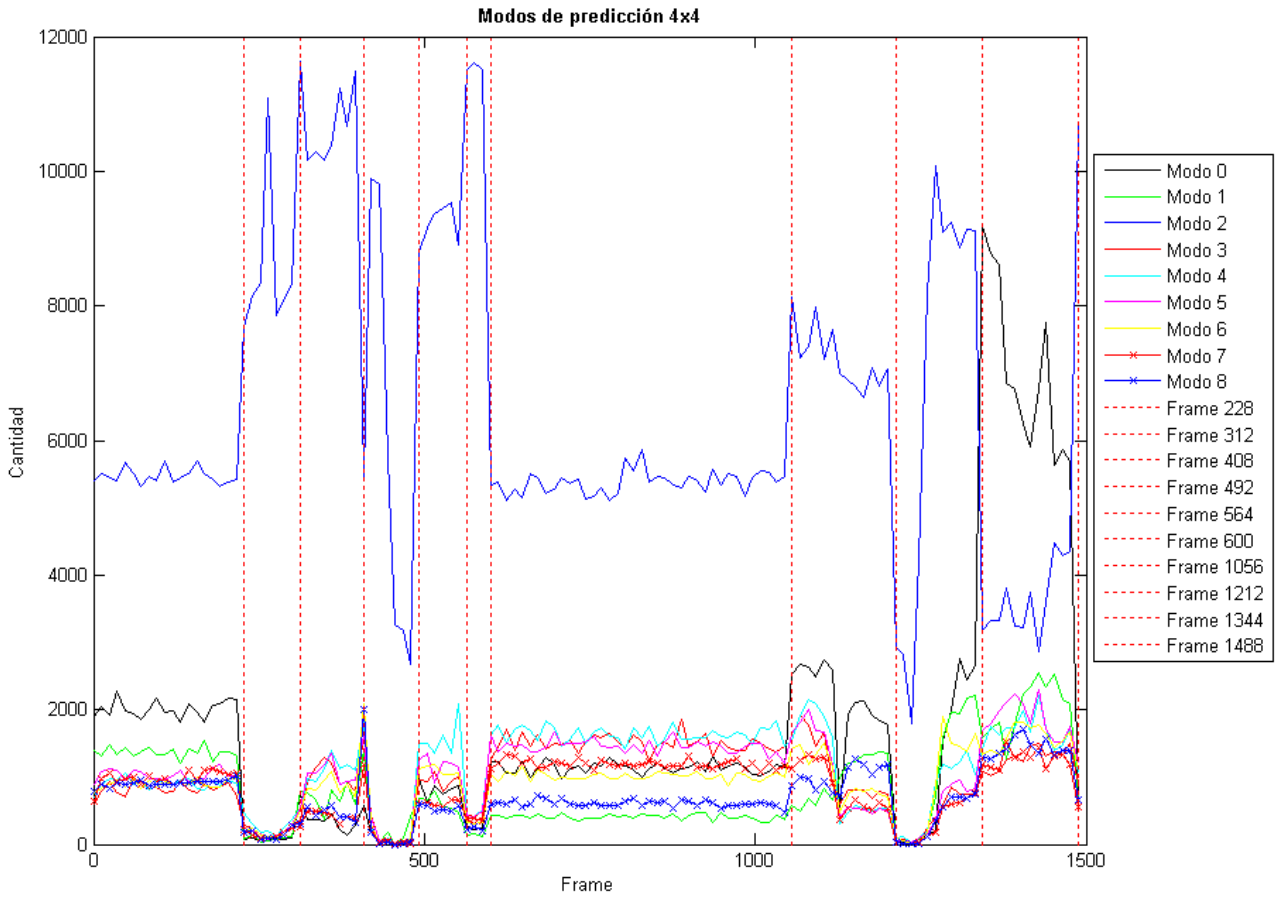


FIG. 5-28 MODOS DE PREDICCIÓN 4X4 VÍDEO "FRAGMENT0.H264"

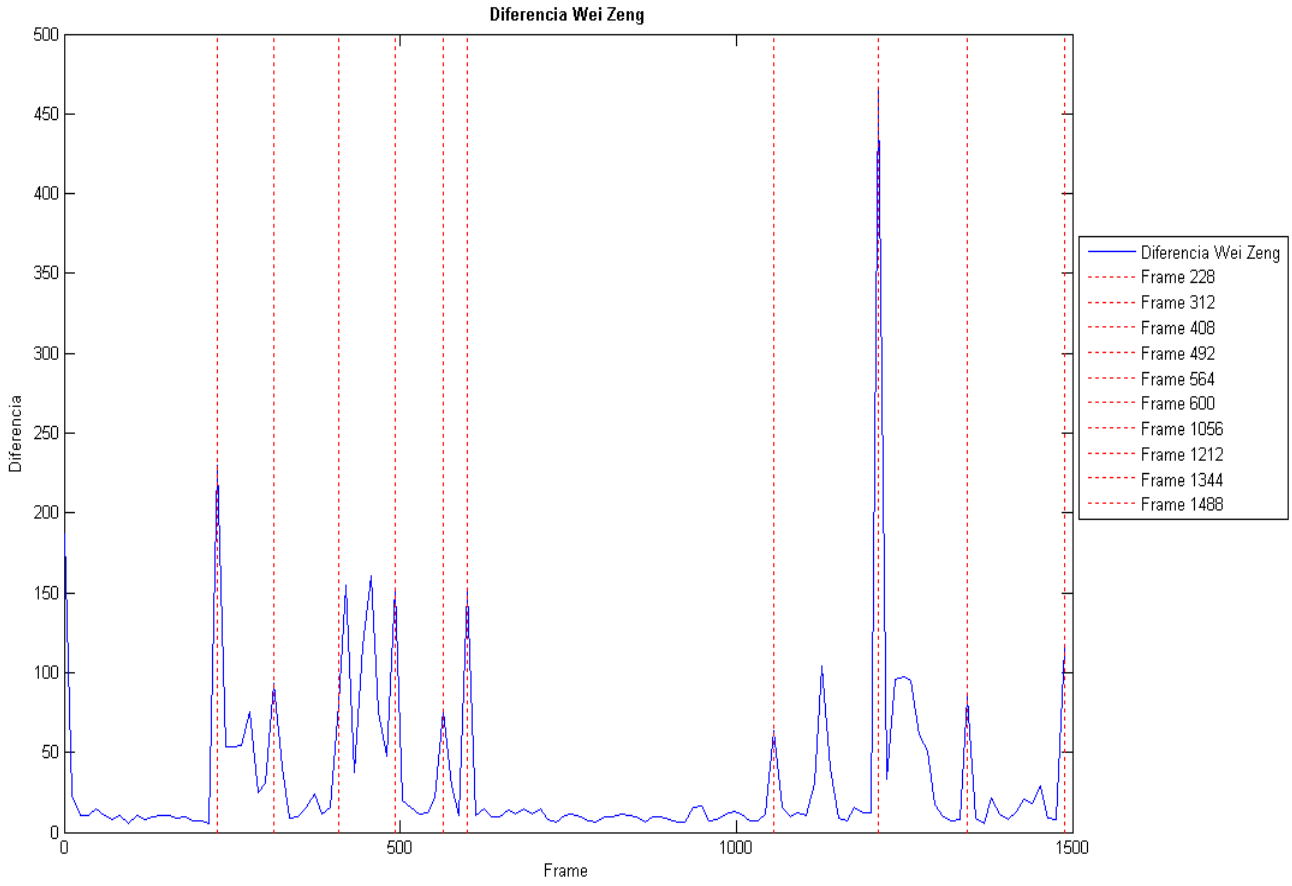


FIG. 5-29 DIFERENCIA "WEI ZENG" VÍDEO "FRAGMENT0.H264"

Los *frames* donde se han observado que se producen los cambios de toma a través de la aplicación *VISUALmpegAVC* son: 219, 311, 402, 481, 557, 595, 1049, 1201, 1343, 1484.

Pero estos *frames* no siempre se corresponden con *frames* I, que aparecen periódicamente cada 12 *frames*. De este modo, donde se deberían detectar los cambios de toma con este parámetro es en los *frames*: 228, 312, 408, 492, 564, 600, 1056, 1212, 1344, 1488. En estos puntos se ha elegido representar los cambios de toma en las gráficas.

Por otro lado, se verá que siempre en la figura "*Fig. 5-28*" el modo 2, DC, es el predominante. Aún así se atenderán especialmente las variaciones de todos los modos. Es en este modo donde se ven las variaciones más drásticas, aunque para suponer un cambio de toma, los demás modos también deberían sufrir alguna variación, sin olvidar de la diferencia "Wei Zeng" ("*Fig. 5-29*"). A partir de ahora, cuando se hable de cambios en la cantidad de veces que se usan los diferentes modos o de la diferencia "Wei Zeng", se hará referencia a las figuras "*Fig. 5-28*" y "*Fig. 5-29*" respectivamente.

Comentarios acerca de los datos obtenidos:

- No prestar atención al valor inicial de la diferencia "Wei Zeng" debido a que se utiliza los valores del *frame* anterior para hacer la comparación y en el primer caso, al no existir el *frame* anterior, coge valor 0 y por eso la diferencia es tan grande.

- En el *frame* 408 debería detectarse un cambio de toma y, sin embargo, a partir de los datos extraídos parece que se produce en el *frame* 420. Esto ya ocurría en el apartado "*5.3.1 Cambios de toma*" y se había concluido que podía deberse a que el número de macrobloques con estructura 4x4 y 16x16 apenas variase a pesar del cambio de toma, siendo también posible que sufra una pequeña variación (en cantidad) el tipo de modo utilizado por los diferentes macrobloques del *frame*. Aquí es donde se puede ver una de las limitaciones de observar cambios de toma sólo observando este parámetro. Es por eso que se deben analizar más parámetros.

- En los *frames* 444 y 456 se observan cambios importantes en el modo DC (modo 2) de los macrobloques Intra 4x4, pero no así en el resto de modos y no se observa tampoco un pico en la diferencia "Wei Zeng", aumentan pero no representan un pico, así que se podría deducir que no hay un cambio de toma, como efectivamente no se produce.

- Los casos de los *frames* 492, 564 y 600 son muy claros en todos los aspectos.

- En el *frame* 1056 los datos no son tan claros como en los casos anteriores. Sí se ve que se produce una diferencia con respecto a los *frames* I anteriores, pero los datos no son tan significativos como en otros ejemplos. También se produce un pico en la diferencia "Wei Zeng", aunque tampoco es un valor tan grande. Se produce un cambio de toma aquí.

- En el *frame* 1128 se produce un caso llamativo. El modo 2, el que más suele variar, apenas varía, y sí lo hacen todos los demás modos, especialmente el modo 1. La diferencia "Wei Zeng" también toma un valor bastante alto, siendo además un pico. Sin embargo, no se produce ningún cambio de toma aquí. Este caso ya se ha visto en el apartado "*5.3.1 Cambios de toma*", se trata del caso de la jarra de cerveza que desaparece de la imagen en un muy breve espacio de tiempo. Así, en el *frame* 1116 consiste en un primer plano de una jarra de cerveza con una mano cogiendo dicha cerveza. Para el *frame* I posterior, el 1128, dicha cerveza ya no se encuentra en el plano, desaparece por completo un objeto que ocupaba gran parte de la imagen ("*Fig. 5-30*"). Es comprensible que este caso pueda llevar a confusiones creyéndose que ahí se pueda producir un cambio de toma.

Frame 1116

Frame 1128

FIG. 5-30 FRAMES I: 1116 Y 1128 VÍDEO "FRAGMENT0.H264"

- En el *frame* 1212 se producen variaciones radicales. Esto se debe especialmente a que ha cambiado de manera muy brusca el número de macrobloques Intra 4x4 respecto a los 16x16, siendo ahora una imagen con abundancia de macrobloques 16x16. De esta manera, se trata de una imagen muy homogénea. Efectivamente, se trata también de un cambio de toma.

- En los *frames* 1248 y 1260 se observan cambios importantes en el modo 2 de los macrobloques Intra 4x4, pero no en el resto de modos, y también se puede ver que la diferencia "Wei Zeng" tampoco aparece como un pico respecto a las diferencias de los *frames* adyacentes. Así que se podría deducir, de manera correcta, que no se produce un cambio de toma en estos *frames*. Este caso también se ha visto en "*5.3.1 Cambios de toma*", era el caso del cambio de luminosidad en la escena ("*Fig. 5-31*").

FRAME 1236 **FRAME 1248** **FRAME 1260**

Frame 1272 **Frame 1284**

FIG. 5-31 FRAMES I: 1236-1284 VÍDEO "FRAGMENT0.H264"

- El *frame* 1344 es otro ejemplo claro, con grandes diferencias ya no solo en el modo 2, sino también en el modo 1 de los macrobloques Intra 4x4. Se cumplen todas las condiciones y, efectivamente, se trata de un cambio de toma. Como también ocurre en el *frame* 1488 con la única salvedad de que, al ser el último *frame*, no hay forma de comprobar que la diferencia "Wei Zeng" es un pico (con respecto al frame I siguiente), aunque se observa que es claramente superior a la diferencia del *frame* anterior, así que no tendríamos problemas para poder suponer que es un cambio de toma.

Ejemplo 2: fragment1.h264

Segundo ejemplo, explicado de la misma manera que en el ejemplo anterior.

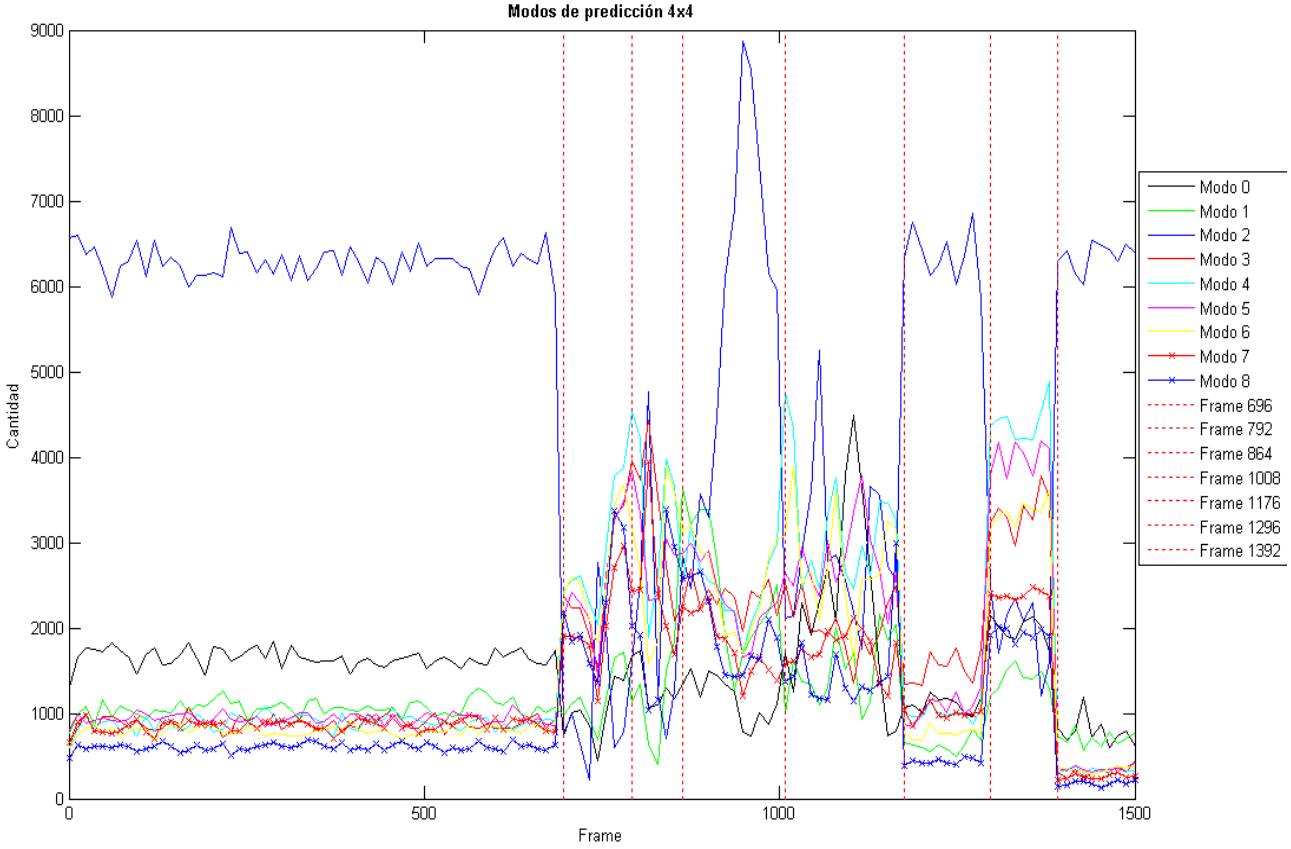


FIG. 5-32 MODOS DE PREDICCIÓN 4X4 VÍDEO "FRAGMENT1.H264"

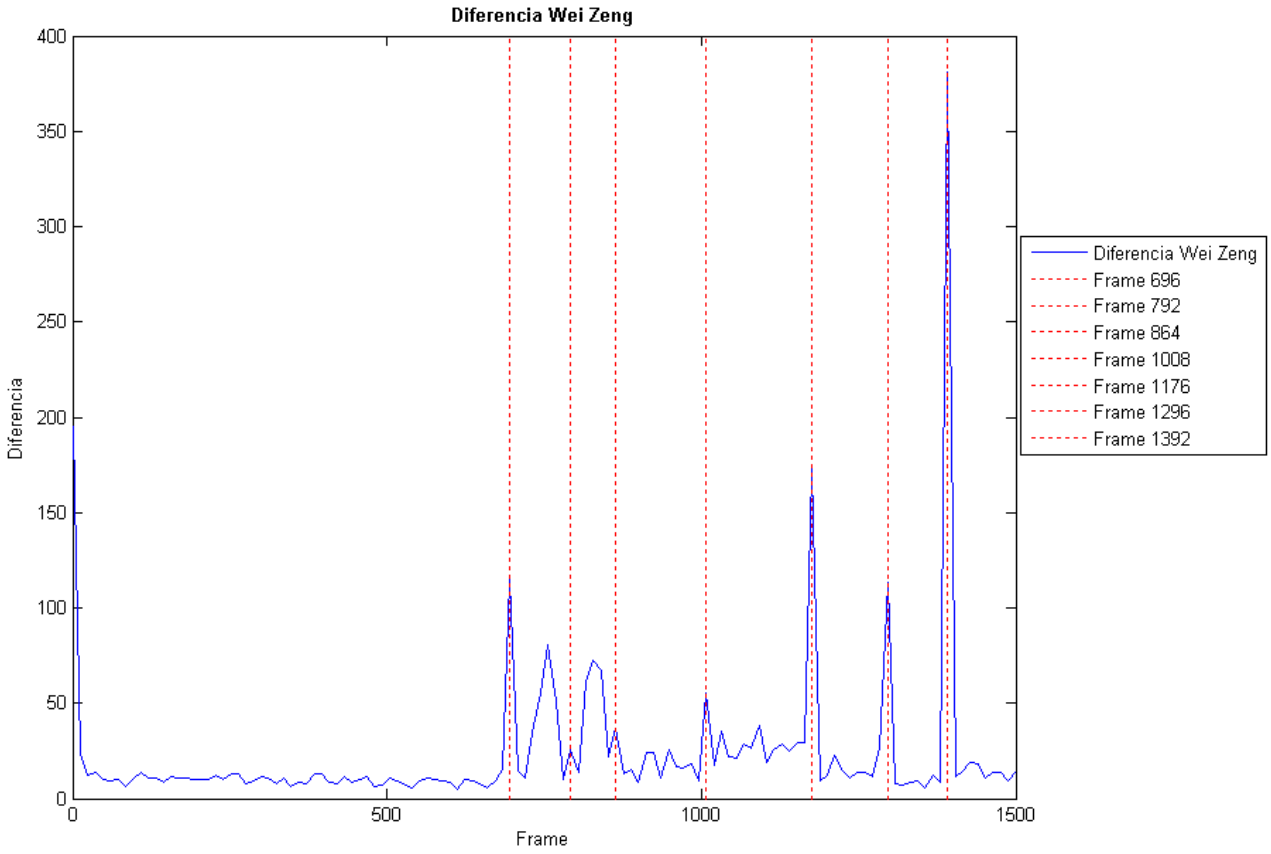


FIG. 5-33 DIFERENCIA "WEI ZENG" VÍDEO "FRAGMENT1.H264"

Los cambios de toma se producen en los *frames*: 695, 791, 863, 1001, 1166, 1290, 1381. De la misma manera que en el ejemplo anterior, ya que los *frames* I aparecen periódicamente cada 12 *frames*, se deberían detectar los cambios de toma en los *frames*: 696, 792, 864, 1008, 1176, 1296 y 1392. Veamos qué ocurre.

De la misma manera que antes, se verá que siempre el modo 2, DC, es el predominante. Pero hay que tener en cuenta la variación de todos y cada uno de los 9 modos, pudiendo suponer cambios de toma cuando se observen cambios significativos en varios de ellos. Se vuelve a prestar especial atención a lo que se ha denominado "Diferencia Wei Zeng". Se exponen los resultados de la misma manera que en el ejemplo anterior, cuando se hable de la cantidad de cada uno de los modos, se hará referencia a la figura "*Fig. 5-32*" y cuando se hable de la diferencia "Wei Zeng", a la figura "*Fig. 5-33*".

Comentarios acerca de los datos obtenidos:

- No prestar atención al pico con el que empieza la diferencia "Wei Zeng", debido a que hace comparación con el *frame* anterior que, al no existir, toma valor 0, por eso la diferencia es tan grande.

- En el *frame* 696 se pueden apreciar valores significativos en todos los modos excepto en el modo 1, observando además un pico alto en la diferencia "Wei Zeng" de 115.02, con valores 15.04 y 14.15 en los *frames* I adyacentes.

- Se observa cómo en el *frame* 756 la diferencia "Wei Zeng" se incrementa bastante pero no se trata de un pico, pues los valores de los *frames* contiguos al *frame* 756 también cogen valores altos. En la gráfica se puede alcanzar a ver, por ejemplo, que en el *frame* 696 el pico era bastante más pronunciado que para este *frame*. Así que, a pesar de una variación en casi todos los modos y un valor alto de la diferencia "Wei Zeng", se puede deducir de manera correcta que no se trata de un cambio de toma por no tratarse esta diferencia de un pico.

- Para los *frames* 792 y 864 los datos no son ni mucho menos remarcables. Se hubieran pasado por alto de no saberse que supuestamente aquí debería verse reflejado un cambio de toma. Sí se observan cambios significantes de este mismo cambio de toma (en el *frame* P 791) en la extracción del parámetro referente al números de macrobloques I ó P de este mismo ejemplo en: *"5.4 Número de macrobloques I/P-B".* Lo mismo ocurre con el cambio de toma que debería detectarse en el *frame* 864. A pesar de esto, se puede ver cómo la diferencia "Wei Zeng" aparece como un pico, cosa muy diferente a la que ocurría en el *frame* 756 o lo que ocurre en el 828, en los que se ven valores altos pero también en sus frames adyacentes.

- En el *frame* 1008 aparece un cambio de toma. Los datos no son tan claros, pero da una mejor pista la distancia "Wei Zeng" que, sin ser muy alta, corresponde a un pico, ya que las distancias adyacentes son: 9.40 y 17.58 (de los *frames* 996 y 1020 respectivamente). Efectivamente se trata de un cambio de toma.

- En el *frame* 1176 sí se ven datos bastante llamativos acerca de la presencia de un cambio de toma en todos los aspectos, alta variación en la mayoría de los modos y una distancia "Wei Zeng" bastante alta y también formando un pico pronunciado. Se trata de un cambio de toma supuestamente fácil de detectar.

- En el *frame* 1296 se da otro caso en el que se observa de forma más o menos clara un cambio de toma, aunque con datos no tan significativos como en el caso anterior. Se cumplen todas las condiciones y se corresponde con un cambio de toma.

- Por último, en el *frame* 1392, se ve el caso más claro de este ejemplo. Los datos son más que evidentes para saber que se trata de un cambio de toma.

6 CONCLUSIONES Y TRABAJO FUTURO

6.1 CONCLUSIONES

En el presente proyecto se ha llevado a cabo un estudio en profundidad de todas las características del estándar de compresión de vídeo H.264, llevando a cabo, de forma paralela, una comparación con los estándares previos de MPEG. Todo esto orientado al análisis de vídeo en el dominio comprimido.

Tras un análisis de todas estas características del mencionado estándar, se ha hecho una selección de los parámetros a extraer en la decodificación de un vídeo en H.264, con el objetivo de alcanzar una finalidad práctica a lo visto en la teoría, así como realizar un análisis de la eficiencia de dicha extracción. No se ha marcado como objetivo unos resultados positivos de dicha extracción, sino un análisis crítico en términos de funcionalidad y eficiencia. Se han comentado ventajas e inconvenientes de cada uno de los parámetros que se decidieron extraer a partir del codificador/decodificador JM y se ha usado la herramienta de análisis de vídeo VISUALmpegAVC como apoyo, pudiendo resumir los resultados obtenidos de la siguiente manera:

- Las funciones sobre las que se han trabajado a través de estos parámetros son los cambios de toma y la segmentación de objetos en una imagen.

- Sería una equivocación tomar sólo un parámetro para el análisis de resultados, sea cual sea la función que se le quiera dar. Tras las diferentes pruebas realizadas, se puede concluir que los diferentes parámetros extraídos pueden ser de mayor o menos eficiencia, pero ninguno puede ser concluyente por sí solo. Es muy recomendable utilizar varios parámetros para la consecución de resultados, dándole la importancia que se considere oportuna a cada uno ellos.

- Siempre se debe saber en qué resolución se está trabajando. Se han visto ejemplos sobre cómo varían los resultados en un análisis sobre el mismo vídeo cambiando la resolución. Obviamente, a mayor resolución, más eficientes son los resultados.

- En referencia a la detección de cambios de toma, el parámetro más destacable en cuanto a buenos resultados es el de “Número de macrobloques I y P ó B en un frame”. Ha resultado ser muy eficiente, y no se ha visto apenas afectado por cambios de resolución. El problema de este parámetro recae en la imposibilidad de detectar un cambio de toma si el mismo se produce en un frame I. Por ello, se recomienda la utilización de más parámetros para el estudio de cualquier función.

- En cuanto a la detección de contornos de objetos, ningún parámetro ha resultado tan fiable como el “Número de macrobloques I y P ó B en un frame” para los cambios de toma. Los “tipos de macrobloque en frames Intra” (4x4 ó 16x16) ha demostrado ser un parámetro que actúa de manera aceptable pero en unas condiciones muy limitadas, como escenas no muy ‘recargadas’ (con muchas zonas texturizadas) o en términos de resolución altos (ha demostrado ser muy vulnerable a bajas resoluciones).

6.2 TRABAJO FUTURO

Se propone un análisis en profundidad sobre algunas de las posibles aplicaciones a analizar en el dominio comprimido de H.264, llevar a cabo diferentes sistemas con la intención de obtener unos resultados de manera automática.

Con el apoyo del presente trabajo y de otros trabajos referenciados aquí sobre H.264, se puede fijar un umbral teórico o empírico y proceder con análisis de resultados, tras el desarrollo de un algoritmo que pueda realizar alguna de las aplicaciones mencionadas aquí en el dominio comprimido.

En el auge de la alta definición, de las aplicaciones multimedia, vídeo en streaming, videoconferencias, etc., es muy importante seguir desarrollando algoritmos y sistemas referentes a la compresión de vídeo. Pero no sólo se busca variedad y abarcar diferentes ámbitos del dominio comprimido, sino también progresos en los mismos algoritmos ya implementados, con la intención de alcanzar mayor robustez frente a ruido, o a la utilización de diferentes sistemas o redes.

REFERENCIAS

[1] Watkinson, John, 2nd Ed, 2004. The MPEG Handbook MPEG-1, MPEG-2, MPEG-4.

[2] E.G. Richardson, Iain, 2003. H.264 and MPEG-4 Video Compression. Video Coding for Next-generation Multimedia.

[3] Phadtare, M. System Architect, Video. NXP Semiconductors India Pvt. Ltd.

[4] K. Wahid, V. Dimitrov, and G. Jullien, "New Encoding of 8x8 DCT to make H.264 Lossless," IEEE Asia Pacific Conference Circuits and Systems (APCCAS), pp. 780 – 783, Dec. 2006.

[5] Ephraim Feig and Shmuel Winograd, "Fast Algorithms for the Discrete Cosine Transform", IEEE Septiembre 1992.

[6] http://www.jvsg.com/

[7] http://www.sony.es/biz/view/ShowContent.action?site=biz_es_ES&contentId=1222694816901&sectiontype=NVM+Whitepapers

[8] http://www2.panasonic.com/webapp/wcs/stores/servlet/prModelDetail?storeId=11301&catalogId=13251&itemId=339747&modelNo=Content03272009032224962&surfModel=Content03272009032224962

[9] http://www.atsc.org/

[10] H. Wang, A. Divakaran, A. Vetro, S.-F. Chang, H. Sun. "Survey of compressed-domain features used in audio-visual indexing and analysis", J. Visual Commun. Image Representation 14 (2) (2003) pp. 50-183.

[11] D. Le Gall, "MPEG:A Video Compression Standard for Multimedia Applications". Communications of the ACM, April 1991, vol.34, (no.4):46-58.

[12] Chang, S.-F., Messerschmit, D.G., 1995. "Manipulation and compositing of MC-DCT compressed video". IEEE J. Selected Areas Commun. Special Issue on Intelligent Signal Processing, January, pp. 1-11.

[13] Yeo, B.-L., Liu, B., 1995a. "On the extraction of DC sequence from MPEG video". In: Proc. IEEE Internat. Conf. on Image Processing, vol. 2, pp. 260–263.

[14] Chen, J.-Y., Taskiran, C., Delp, E.J., Bouman, C.A., 1998. "ViBE: a new paradigm for video database browsing and search". In: Proc. IEEE Workshop on Content-Based Access of Image and Video Databases.

[15] Yeo, B.-L., Liu, B., 1995b. Rapid scene analysis on compressed videos. IEEE Trans. Circuits Systems Video Technol.fs 5 (6), 533–544.

[16] Tan, Y.-P., Kulkarni, S.R., Ramadge, P.J., 1999. "A framework for measuring video similarity and its application to video query by example". In: Proc. IEEE Internat. Conf. on Image Processing, Kobe, Japan, October.

[17] Won, C.S., Park, D.K., Na, I.Y., Yoo, S.-J., 1999. “Efficient color feature extraction in compressed video”. In: Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases VII, San Jose, CA, January, pp. 677–686.

[18] Song, B.C., Ra, J.B., 1999. “Fast edge map extraction from MPEG compressed video data for video parsing. In: Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases”. VII, San Jose, CA, January, pp. 710–721.

[19] Shen, B., Sethi, I.K., 1996a. “Convolution-based edge detection for image/video in block DCT domain”. J. Visual Commun. Image Representation 7 (4), 411–423.

[20] Arman, F., Hsu, A., Chiu, M.-Y., 1993. “Image processing on compressed data for large video databases”. In: Proc. ACM Multimedia 93, Anaheim, CA, pp. 267–272.

[21] Zhang, H.J., Low, C.Y., Smoliar, S.W., 1995. Video parsing and browsing using compressed data. Multimedia Tools Appl. 1 (1), 89–111.

[22] Meng, J., Juan, Y., Chang, S.-F., 1995. “Scene change detection in an MPEG compressed video sequence”. In: IS & T/SPIE Symposium Proceedings, vol. 2419, San Jose, CA, February.

[23] Ardizzone, E., La Cascia,M.,Molinelli, E., 1996. “Motion and color based video indexing and retrieval”. In: Proc. Internat. Conf. on Pattern Recognition.

[24] Eng, H.-L., Ma, K.-K., 1999. “Motion trajectory extraction based on macroblock motion vectors for video indexing”. In: Proc. IEEE Internat. Conf. on Image Processing, Kobe, Japan, October.

[25] Kobla, V., Doermann, D., Lin, K.-I., Faloutsous, C., 1997. “Compressed domain video indexing techniques using DCT and motion vector information in MPEG video”. In: Proc. SPIE Conf. On Storage and Retrieval for Image and Video Databases V, SPIE vol. 3022, pp. 200–211.

[26] Saur, D.D., Tan, Y.-P., Kulkarni, S.R., Ramadge, P.J., 1997. “Automated analysis and annotation of basketball video”. In: Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases V, SPIE vol. 3022, pp. 176–187.

[27] Akutsu, A., Tonomura, Y., Hashimoto, H., Ohba, Y., 1992. “Video indexing using motion vectors”. In: Proc. SPIE Conf. on Visual Communications and Image Processing, SPIE vol. 1818, pp. 1522–1530.

[28] Kobla, V., Doermann, D., Lin, K.-I., 1996. “Archiving, indexing, and retrieval of video in the compressed domain”. In: Proc. SPIE Conf. On Multimedia Storage andArchiving Systems, SPIE vol. 2916, pp. 78–89.

[29] Tan, Y.-P., Kulkarni, S.R., Ramadge, P.J., 1995. “A new method for camera motion parameter estimation”. In: Proc. IEEE Internat. Conf. on Image Processing, vol. 1, pp. 406–409.

[30] Tse, Y.T., Baker, R.L., 1991. “Camera zoom/pan estimation and compensation fore video compression”. In: Proc. SPIE Conf. on Image Processing Algorithms and Techniques II, Boston, MA, pp. 468–479.

[31] Divakaran, A., Sun, H., 2000. “A descriptor for spatial distribution of motion activity for compressed video”. In: Proc. SPIE Conf. on Storage and Retrieval for Media Databases 2000, San Jose, CA, January, pp. 392–398.

[32] Ardizzone, E., La Cascia, M., Avanzato, A., Bruna, A., 1999. “Video indexing using MPEG motion compensation vectors”. In: Proc. IEEE Internat. Conf. on Multimedia Computing and Systems, 1999.

[33] Kobla, V., DeMenthon, D., Doermann, D., 1999. “Detection of slow-motion replay sequences for identifying sports videos”. In: Proc. IEEE Workshop on Multimedia Signal Processing.

[34] Feng, J., Lo, K.T., Mehrpour, H., 1996. “Scene change detection algorithm for MPEG video sequence”. In: Proc. IEEE Internat. Conf. on Image Processing, Lausanne, Switzerland.

[35] Divakaran, A., Ito, H., Sun, H., Poon, T., 1999. “Scene change detection and feature extraction for MPEG-4 sequences”. In: Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases VII, San Jose, CA, January, pp. 545–551.

[36] Boccignone, G., De Santo, M., Percannella, G., 2000. “An algorithm for video cut detection in MPEG sequences”. In: Proc. SPIE Conf. on Storage and Retrieval of Media Databases 2000, San Jose, CA, January, pp. 523–530.

[37] Setton, E. and Girod, B. “Rate-Distortion Analysis and Streaming of SP and SI Frames”, Proc. Of IEEE, Transactions on Circuits and Systems for video technology, Vol. 16, No. 6 pp 733-743, 2006.

[38] P. Kuhn, “Camera motion estimation using feature points in MPEG compressed domain”. In Proceedings 2000 International Conference on Image Processing, 2000, vol. 3, pp. 596-9.

[39] Wei Xiong; Lee, J.C.-M., “Efficient scene change detection and camera motion annotation for video classification” In Computer Vision and Image Understanding. Aug. 1998, vol 71, no 2, pp.166-181.

[40] Kim, S.M., Byun, J. and Won, C.S., “A scene change detection in H.264/AVC compression domain”. In Lecture Notes in Computer Science, vol. 3768. Springer, Berlin, 2005. pp. 1072-1082.

[41] De Bruyne, S. “Video Shot Detection on H.264/AVC Compressed Bitstreams Using Temporal Prediction Types”.

[42] De Bruyne, S., De Neve, W., De Wolf, K., De Schrijver, D., Verhoeve, P. and Van de Walle, R., “Temporal video segmentation on H.264/AVC compressed bitstreams” In Lecture Notes in Computer Science, vol. 4351. Springer, Berlin, 2007. pp. 1-12.

[43] Zeng, W. and Gao, W., “Shot change detection on H.264/AVC compressed video”. In: Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 4. pp. 3459-3462.

[44] Wei, Z., Ngan, King N., Li, Hongliang., “An efficient intra-mode selection algorithm for H.264 based on edge classification and rate-distortion estimation”, Image Communication, 2008. v.23 n.9, p.699-710.

[45] F. Porikli, “Real-time video object segmentation for MPEG encoded video sequences” In: Proc. SPIE Conference on Real-Time Imaging VIII, San Jose, 2004, vol. 5297, pp. 195-203.

[46] De Bruyne, S., Poppe, C., Verstockt, S., Lambert, P., De Walle, R.V., "Estimating motion reliability to improve moving object detection in the H.264/AVC domain". IEEE international conference on Multimedia and Expo, 2009. p.330-333.

[47] G. Yang, S. Yu, and Z. Zhang, "Robust moving object segmentation in the compressed domain for H.264 video stream". In Proc. Picture Coding Symposium, 2006.

[48] W. You, M.S.H. Sabirin, and M.C. Kim, "Moving object tracking in H.264/AVC bitstream". In *MCAM07*, 2007. pp. 483-492.

[49] http://iphome.hhi.de/suehring/tml/

APÉNDICE A

En este apéndice se expondrán las tablas correspondientes a las gráficas del apartado "*5.3.1 Cambios de toma*". Las tablas se corresponden con el respectivo ejemplo de dicho apartado.

Ejemplo 1: news.h264

TABLA A-1 NÚMERO MB'S INTRA VÍDEO "NEWS.H264"

	4x4	**16x16**
Frame 0:	349	65
Frame 12:	360	54
Frame 24:	357	57
Frame 36:	**277**	**137**
Frame 48:	275	139
Frame 60:	278	136
Frame 72:	**372**	**42**
Frame 84:	376	38
Frame 96:	377	37
Frame 108:	**275**	**139**
Frame 120:	295	119
Frame 132:	303	111
Frame 144:	305	109
Frame 156:	**323**	**91**
Frame 168:	328	86
Frame 180:	330	84

*Se ha resaltado en **negrita** los frames I posteriores a un cambio de toma.

Ejemplo 2: fragment0.h264

TABLA A-2 NÚMERO MB'S INTRA VÍDEO "FRAGMENT0.H264"

	4x4	**16x16**
Frame 0:	818	802
Frame 12:	896	724
Frame 24:	915	705
Frame 36:	916	704
Frame 48:	907	713
Frame 60:	913	707
Frame 72:	889	731
Frame 84:	898	722
Frame 96:	895	725
Frame 108:	901	719
Frame 120:	884	736
Frame 132:	886	734
Frame 144:	914	706
Frame 156:	901	719
Frame 168:	903	717
Frame 180:	904	716
Frame 192:	911	709
Frame 204:	922	698
Frame 216:	911	709
Frame 228:	**587**	**1033**
Frame 240:	607	1013
Frame 252:	573	1047
Frame 264:	745	875
Frame 276:	545	1075
Frame 288:	580	1040
Frame 300:	642	978
Frame 312:	**977**	**643**
Frame 324:	1011	609
Frame 336:	1011	609
Frame 348:	1026	594
Frame 360:	1106	514
Frame 372:	1030	590
Frame 384:	1032	588
Frame 396:	1038	582
Frame 408:	**1110**	**510**
Frame 420:	759	861
Frame 432:	629	991
Frame 444:	394	1226
Frame 456:	205	1415
Frame 468:	215	1405
Frame 480:	212	1408
Frame 492:	**1031**	**589**
Frame 504:	1034	586
Frame 516:	1040	580
Frame 528:	1033	587
Frame 540:	1036	584
Frame 552:	1050	570
Frame 564:	**878**	**742**
Frame 576:	879	741
Frame 588:	883	737
Frame 600:	**892**	**728**
Frame 612:	898	722

Frame 624:	894	726
Frame 636:	894	726
Frame 648:	917	703
Frame 660:	889	731
Frame 672:	908	712
Frame 684:	903	717
Frame 696:	908	712
Frame 708:	889	731
Frame 720:	889	731
Frame 732:	898	722
Frame 744:	867	753
Frame 756:	901	719
Frame 768:	907	713
Frame 780:	892	728
Frame 792:	882	738
Frame 804:	904	716
Frame 816:	920	700
Frame 828:	906	714
Frame 840:	897	723
Frame 852:	891	729
Frame 864:	900	720
Frame 876:	906	714
Frame 888:	905	715
Frame 900:	906	714
Frame 912:	908	712
Frame 924:	892	728
Frame 936:	903	717
Frame 948:	891	729

Frame 960:	882	738
Frame 972:	898	722
Frame 984:	876	744
Frame 996:	887	733
Frame 1008:	892	728
Frame 1020:	884	736
Frame 1032:	895	725
Frame 1044:	857	763
Frame 1056:	**1212**	**408**
Frame 1068:	1237	383
Frame 1080:	1259	361
Frame 1092:	1232	388
Frame 1104:	1236	384
Frame 1116:	1090	530
Frame 1128:	711	909
Frame 1140:	893	727
Frame 1152:	927	693
Frame 1164:	906	714
Frame 1176:	895	725
Frame 1188:	899	721
Frame 1200:	896	724
Frame 1212:	**208**	**1412**
Frame 1224:	203	1417
Frame 1236:	118	1502
Frame 1248:	286	1334
Frame 1260:	598	1022
Frame 1272:	880	740
Frame 1284:	1112	508

Frame 1296:	1177	443
Frame 1308:	1203	417
Frame 1320:	1213	407
Frame 1332:	1233	387
Frame 1344:	**1370**	**250**
Frame 1356:	1386	234
Frame 1368:	1389	231
Frame 1380:	1344	276
Frame 1392:	1362	258

Frame 1404:	1368	252
Frame 1416:	1339	281
Frame 1428:	1422	198
Frame 1440:	1424	196
Frame 1452:	1312	308
Frame 1464:	1307	313
Frame 1476:	1334	286
Frame 1488:	**1088**	**532**

* Se ha resaltado en **negrita** los frames I posteriores a un cambio de toma.

Ejemplo 3: fragment1.h264

TABLA A-3 NÚMERO MB'S INTRA VÍDEO "FRAGMENT1.H264"

	4x4	**16x16**
Frame 0:	779	841
Frame 12:	887	733
Frame 24:	903	717
Frame 36:	866	754
Frame 48:	877	743
Frame 60:	854	766
Frame 72:	863	757
Frame 84:	868	752
Frame 96:	874	746
Frame 108:	855	765
Frame 120:	866	754
Frame 132:	868	752
Frame 144:	877	743
Frame 156:	873	747
Frame 168:	875	745
Frame 180:	877	743
Frame 192:	850	770
Frame 204:	877	743
Frame 216:	876	744
Frame 228:	888	732
Frame 240:	884	736
Frame 252:	876	744
Frame 264:	884	736
Frame 276:	878	742
Frame 288:	890	730
Frame 300:	869	751
Frame 312:	869	751
Frame 324:	882	738
Frame 336:	875	745
Frame 348:	869	751
Frame 360:	866	754
Frame 372:	866	754
Frame 384:	861	759
Frame 396:	876	744
Frame 408:	869	751
Frame 420:	868	752
Frame 432:	880	740
Frame 444:	864	756
Frame 456:	883	737
Frame 468:	868	752
Frame 480:	867	753
Frame 492:	883	737
Frame 504:	871	749
Frame 516:	872	748
Frame 528:	861	759
Frame 540:	871	749
Frame 552:	855	765
Frame 564:	888	732
Frame 576:	868	752
Frame 588:	873	747
Frame 600:	890	730
Frame 612:	881	739

Frame 624:	892	728
Frame 636:	885	735
Frame 648:	878	742
Frame 660:	882	738
Frame 672:	872	748
Frame 684:	853	767
Frame 696:	**993**	**627**
Frame 708:	1041	579
Frame 720:	1021	599
Frame 732:	863	757
Frame 744:	827	793
Frame 756:	1178	442
Frame 768:	1469	151
Frame 780:	1531	89
Frame 792:	**1524**	**96**
Frame 804:	1497	123
Frame 816:	1354	266
Frame 828:	1038	582
Frame 840:	1409	211
Frame 852:	1318	302
Frame 864:	**1467**	**153**
Frame 876:	1502	118
Frame 888:	1476	144
Frame 900:	1474	146
Frame 912:	1380	240
Frame 924:	1346	274
Frame 936:	1328	292
Frame 948:	1314	306

Frame 960:	1406	214
Frame 972:	1421	199
Frame 984:	1449	171
Frame 996:	1458	162
Frame 1008:	**1300**	**320**
Frame 1020:	1306	314
Frame 1032:	1284	336
Frame 1044:	1233	387
Frame 1056:	1277	343
Frame 1068:	1304	316
Frame 1080:	1409	211
Frame 1092:	1295	325
Frame 1104:	1276	344
Frame 1116:	1302	318
Frame 1128:	1280	340
Frame 1140:	1289	331
Frame 1152:	1188	432
Frame 1164:	1361	259
Frame 1176:	**843**	**777**
Frame 1188:	851	769
Frame 1200:	842	778
Frame 1212:	900	720
Frame 1224:	872	748
Frame 1236:	872	748
Frame 1248:	863	757
Frame 1260:	859	761
Frame 1272:	889	731
Frame 1284:	877	743

Frame 1296:	**1544**	**76**
Frame 1308:	1545	75
Frame 1320:	1552	68
Frame 1332:	1532	88
Frame 1344:	1564	56
Frame 1356:	1552	68
Frame 1368:	1562	58
Frame 1380:	1563	57
Frame 1392:	**597**	**1023**

Frame 1404:	589	1031
Frame 1416:	608	1012
Frame 1428:	590	1030
Frame 1440:	601	1019
Frame 1452:	601	1019
Frame 1464:	601	1019
Frame 1476:	606	1014
Frame 1488:	613	1007
Frame 1500:	614	1006

*Se ha resaltado en **negrita** los frames I posteriores a un cambio de toma.

PRESUPUESTO

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 2.000 €
- Alquiler de impresora láser durante 6 meses ..... 50 €
- Material de oficina .....150 €
- Total de ejecución material ..... 2.200 €

2) Gastos generales

- 16 % sobre Ejecución Material..... 352 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material..... 132 €

4) Honorarios Proyecto

- 640 horas a 15 € / hora ..... 9600 €

5) Material fungible

- Gastos de impresión ..... 60 €
- Encuadernación..... 200 €

6) Subtotal del presupuesto

- Subtotal Presupuesto..... 12060 €

7) I.V.A. aplicable

- 16% Subtotal Presupuesto..... 1929.6 €

8) Total presupuesto

- Total Presupuesto ..... 13989,6 €

Madrid, Julio de 2010

El Ingeniero Jefe de Proyecto

Fdo.: Diego Sarasúa Álvarez

Ingeniero Superior de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un análisis de secuencias de vídeo codificadas en H.264. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

CONDICIONES GENERALES

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las

condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa otro concepto.

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.