



# Técnicas para la Codificación Escalable de Vídeo

M. F. López, S. G. Rodríguez, J. P. Ortiz, V. G. Ruiz, I. García

*Resumen*—El objetivo principal de la codificación de vídeo ha consistido tradicionalmente en la optimización de la calidad del mismo a un bit-rate determinado. Sin embargo, esto ha cambiado por la aparición de aplicaciones de vídeo en redes de computadores tales como el *streaming* de vídeo por Internet, porque en el momento de la codificación del vídeo normalmente no se conoce ni la capacidad del receptor ni el ancho de banda disponible para la transmisión. El nuevo objetivo para estas aplicaciones es realizar una codificación escalable de vídeo. Es decir, que el vídeo comprimido resultado de esta codificación tenga propiedades de escalabilidad.

En este artículo revisamos las distintas técnicas de escalabilidad que usan los estándares MPEG y las estructuras de codificación escalable basadas en la transformada wavelet. Por último, presentamos las primeras ideas de un nuevo codificador de vídeo altamente escalable.

*Palabras clave*— Escalabilidad en vídeo, MPEG, transformada wavelet discreta, JPEG 2000.

## I. INTRODUCCIÓN

La finalidad de un algoritmo de compresión de vídeo digital es eliminar la redundancia espacial y temporal de una secuencia de vídeo de forma que se pueda representar con una distorsión visual y un ratio de compresión aceptables. En un sistema de comunicación de vídeo tradicional, el codificador comprime la señal de vídeo de entrada a un bit-rate que es menor que, y cercano, a la capacidad del canal; además, el decodificador reconstruye la señal de vídeo usando todos los bits recibidos por el canal. Por lo tanto, en este modelo, i) el codificador conoce la capacidad del canal y ii) el decodificador es capaz de decodificar todos los bits recibidos a una velocidad suficiente como para reconstruir el vídeo.

Estas dos suposiciones no tienen por qué cumplirse en aplicaciones tales como el *streaming* de vídeo (*streaming media* si incluye sonido). El *streaming* de vídeo es una técnica para la transmisión de vídeo por Internet que permite la visualización del mismo mientras es recibido. Por lo tanto, no es necesario descargarlo completamente antes de reproducirlo. Los datos se transmiten de forma que pueden ser procesados como un flujo continuo, sin pausa; si el usuario recibe los datos con mayor velocidad que la requerida para la reproducción, necesita guardar el exceso en un buffer. Si los datos no llegan lo suficientemente rápido la visualización será de mala calidad.

El vídeo que se transmite en las aplicaciones de *streaming* ha sido comprimido previamente. Por lo tanto, cuando se realiza la codificación, normalmente

no se conoce ni la capacidad del canal ni los recursos computacionales del receptor. En general, estas aplicaciones se enfrentan a la difícil tarea de proveer vídeo a diferentes niveles de resolución temporal, resolución espacial, calidad y/o bit-rate. Estos parámetros dependen de la capacidad de cada receptor, de las opciones preferidas por los usuarios y de la capacidad variable del canal. Una solución a este problema es comprimir y almacenar cada secuencia de vídeo a distintos bit-rates. De esta forma, el servidor será capaz de entregar el vídeo en la forma requerida. Esta solución tiene dos problemas principales: i) introduce una gran sobrecarga de almacenamiento y ii) no es viable para aplicaciones en tiempo real por la sobrecarga computacional del codificador.

Una solución alternativa consiste en utilizar un tipo de codificación que permita al receptor seleccionar dinámicamente estos parámetros. La secuencia de vídeo se codifica una vez y luego se puede decodificar a distintos i) bit-rates de datos, ii) imágenes por segundo, iii) calidades de imagen y iv) resoluciones espaciales. Esta es una solución muy atractiva por la flexibilidad que ofrece y se denomina “escalabilidad de vídeo”. Podemos clasificar los tipos de escalabilidad de vídeo en los siguientes:

- **Escalabilidad de bit-rate.** El receptor puede pedir un bit-rate de datos determinado, elegido de un conjunto o de un intervalo continuo.
- **Escalabilidad temporal.** El receptor puede elegir cuántas imágenes (frames) se mostrarán por segundo.
- **Escalabilidad espacial.** El receptor puede requerir que las imágenes estén a una resolución determinada.
- **Escalabilidad SNR.** Para unas determinadas resoluciones espacial y temporal, el receptor puede requerir un nivel de calidad determinado, calculado como el grado de fidelidad entre las imágenes original y decodificada. La calidad se mide normalmente con la métrica SNR (*Signal-to-Noise Ratio*).

Por lo tanto, un codificador escalable será aquel que reciba como entrada un vídeo digital y produzca, como salida, un vídeo comprimido que posea una o varias de las características de escalabilidad antes mencionadas.

## II. CODIFICACIÓN ESCALABLE DE VÍDEO EN MPEG

Las primeras técnicas de escalabilidad que se incluyeron en MPEG son las de escalabilidad por capas (SNR, temporal y espacial). Posteriormente apare-

cieron técnicas más complejas y eficientes que conseguían mejorar las propiedades de escalabilidad, como son FGS y PFGS.

#### A. Técnicas básicas de codificación escalable por capas

La técnica de escalabilidad SNR (Signal-to-Noise Ratio) incluida en MPEG-2 o MPEG-4 es una técnica para codificar una secuencia de vídeo en dos capas, cada una a igual número de frames por segundo e idéntica resolución espacial. Los coeficientes DCT de alta calidad se consiguen sumando los coeficientes reconstruidos de la capa base y el residuo obtenido de la capa de mejora. En esta técnica de codificación escalable la información de la capa de mejora se usa en el lazo de la predicción del movimiento. Por lo tanto, si el usuario sólo recibe la capa base, aparece un efecto no deseable que deteriora la calidad de las reconstrucciones y que se denomina error de *drift*.

La técnica de escalabilidad temporal de MPEG-2 codifica una secuencia de vídeo en dos capas a la misma resolución espacial, pero a diferente número de frames por segundo. La capa de mejora proporciona los frames que faltan para formar un vídeo con número de frames por segundo más alto. La eficiencia de codificación de esta técnica es alta y se aproxima bastante a la de la codificación no escalable. En la capa base sólo se usa predicción de tipo P; en la capa de mejora la predicción puede ser de tipo P o B con la capa base, o de tipo P con frames de la propia capa de mejora.

Por último, la técnica de escalabilidad espacial codifica una secuencia de vídeo en dos capas al mismo número de frames por segundo, pero a diferente resolución espacial. La capa base se codifica a una resolución espacial menor. La imagen reconstruida de la capa base es sobremuestreada para formar la predicción de la capa de mejora.

Si no se incluye la información de la capa de mejora en el lazo de predicción se evita la aparición del error de drift. La eficiencia de codificación en esta clase de codificadores escalables de “lazo-abierto” se degrada porque la información de mejora del frame previo no se utiliza para la predicción del frame actual.

Una característica común de las técnicas de codificación escalable vistas en este apartado es que si la capa de mejora no se transmite/recibe/descodifica completamente no se produce ninguna mejora en la calidad del vídeo visualizado por el usuario. Para mejorar este comportamiento surge la técnica de escalabilidad de grano fino FGS (Fine Granularity Scalability).

#### B. Codificación FGS (Fine Granularity Scalability)

La diferencia sustancial entre las técnicas anteriores y la técnica de codificación escalable de grano fino FGS (Fine Granularity Scalability) es que, aunque la técnica FGS también codifica una secuencia en dos capas, la capa de mejora puede ser truncada a la cantidad de bits que se quiera en cada frame y, de esta forma, se obtiene la mejora proporcional al

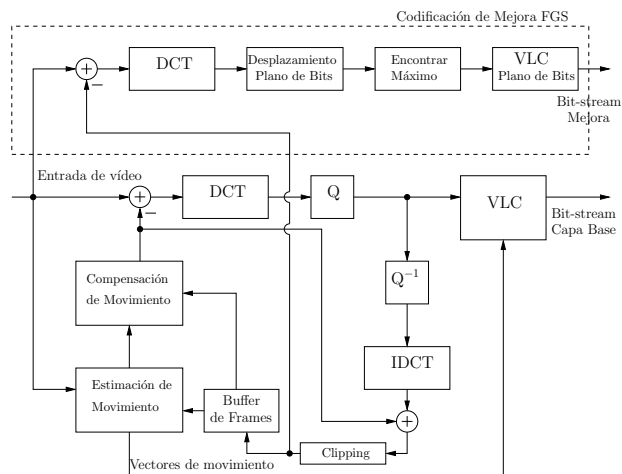


Fig. 1. Estructura del codificador FGS de MPEG-4.

número de bits descodificados de cada frame.

#### B.1 Codificación por planos de bits de los coeficientes DCT

En la codificación DCT convencional, los coeficientes DCT cuantificados se codifican usando codificación *run-level*. El número de ceros consecutivos antes de un coeficiente DCT distinto de cero se denomina “run” y al valor absoluto del coeficiente distinto de cero se le llama “level”. La codificación por planos de bits considera cada coeficiente DCT cuantificado como un número binario con un número determinado de bits, cada uno de los cuales pertenece a una capa de bits distinta [4]. Para cada bloque DCT de  $8 \times 8$ , los 64 valores absolutos son ordenados en zigzag en un array. Un plano de bits de un bloque se define como un array de 64 bits extraídos de la misma posición significativa de distintos coeficientes DCT. La codificación de cada uno de los planos de bits es similar a la codificación *run-level* de los coeficientes DCT. Esta técnica posee cualidades apropiadas para las técnicas de codificación escalable, ya que se puede truncar la descodificación en la capa de bits deseada.

#### B.2 FGS con codificación por planos de bits

La codificación FGS se identifica en MPEG-4 como una funcionalidad deseada para el streaming de vídeo por Internet. Es bien conocido que la codificación wavelet basada, por ejemplo, en codificación aritmética *zero-tree* consigue la funcionalidad FGS [5]. Inicialmente fueron propuestas tres técnicas para FGS en MPEG-4: codificación por planos de bits de los coeficientes DCT, codificación wavelet de la imagen residuo y codificación *matching pursuit* de la imagen residuo. Después de los experimentos realizados, la codificación por planos de bits de los coeficientes DCT fue elegida debido a su simplicidad de implementación.

La idea básica de FGS es codificar una secuencia de vídeo en una capa base y en una capa de mejora. La capa base usa codificación no escalable convencional. El bit-stream de la capa de mejora contiene la diferencia entre la imagen original y la reconstruida de

la capa base. Esta diferencia (o error) es comprimida usando codificación por planos de bits de los coeficientes DCT. De esta forma, se puede considerar que la capa de mejora está formada por múltiples capas, cada una correspondiente a un plano de bits de los coeficientes DCT. La Figura 1 muestra la estructura del codificador FGS.

La codificación por planos de bits de los coeficientes DCT es una forma eficiente y sencilla para conseguir los requerimientos de cambios visuales agradables en la calidad del vídeo mientras ocurren variaciones en el ancho de banda del usuario. Después de que se haya realizado la compresión, el bit-stream de la capa de mejora FGS puede ser truncado a cualquier número de bits en cada frame. El descodificador reconstruye el vídeo de mejora con la capa base y la capa de mejora truncada, y la calidad es proporcional al número de bits descodificados.

### C. Codificación PFGS (*Progressive Fine Granularity Scalable*)

Para mejorar la eficiencia de codificación del esquema de la técnica escalable FGS, en MPEG-4 fue propuesto un marco de trabajo llamado *Progressive Fine Granularity Scalable* (PFGS) [3].

Al igual que FGS, el esquema PFGS también codifica los frames del vídeo en una capa base de relativa poca calidad y una capa de mejora, pero esta última está formada por múltiples capas de mejora obtenidas de la codificación por planos de bits de los coeficientes DCT. Sin embargo, en el marco de trabajo PFGS se intenta usar algunas referencias de alta calidad para las predicciones en las capas de mejora en vez de usar siempre la capa base. Al utilizar referencias de alta calidad, la estimación del movimiento es más precisa y se mejora la eficiencia de codificación. PFGS intenta conseguir un buen balance entre la eficiencia de codificación y las propiedades de escalabilidad.

Hay dos puntos clave en el diseño de PFGS [9]:

1. Emplear predicciones usando las capas de mejora. Cuantas más capas utilicemos, mayor será la eficiencia de codificación.
2. Mantener un camino de predicción desde la capa base hasta la capa de mejora más alta. Este camino de predicción es necesario para una buena recuperación de errores y adaptación a cambios en la velocidad del canal de transmisión.

En los siguientes apartados vamos a ver las características básicas del esquema PFGS y algunas variaciones.

- **Esquema PFGS Básico.** La Figura 2 ilustra el esquema PFGS básico para la codificación eficiente de vídeo evitando el error de drift. El frame 2 es predicho a partir de la capa base y de las capas pares de mejora del frame 1. El frame 3 es predicho a de la capa base y de las capas impares del frame 2. El frame 4 es de nuevo predicho a partir de la capa base y de las capas de mejora impares del frame 3, y así en adelante.

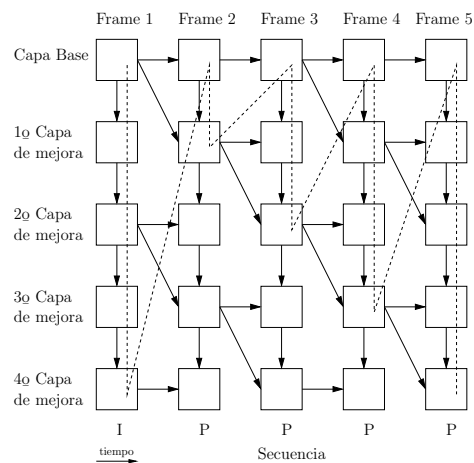


Fig. 2. Marco de trabajo PFGS básico.

Las ventajas de este esquema son obvias cuando se aplica para transmisión de vídeo por Internet o canales *wireless*. El bit-stream codificado puede adaptarse al ancho de banda disponible sin problemas de drift. La Figura 2 muestra también un ejemplo de este proceso de adaptación. Las líneas formadas por trazos dibujan las capas de vídeo transmitidas. Notar que en el frame 2 hay una reducción en el ancho de banda. Después de tres frames el descodificador puede obtener la calidad más alta de vídeo de nuevo. Gracias a este esquema la calidad del vídeo puede recuperarse rápidamente a errores por pérdidas de paquetes o fluctuaciones en el ancho de banda.

### D. Limitaciones

Las limitaciones de las técnicas de escalabilidad que ha ido incluyendo MPEG en sus estándares son todavía sustanciales:

- Las técnicas básicas de codificación escalable por capas limitan su flexibilidad al número de capas. Si codificamos un vídeo con 2 capas, sólo tendremos 2 bit-rates disponibles para las peticiones de transmisión.
- FGS aporta escalabilidad de grano fino. Pero, según incluya o no, la capa de mejora, en el lazo de predicción, padecerá de error de drift o se verá afectada su eficiencia de codificación.
- PFGS mejora la técnica FGS. Sin embargo, sigue manteniendo algunos problemas: i) necesidad de una capacidad mínima en el canal para enviar la información relativa a la capa base del vídeo, ii) el error de drift de FGS es eliminado sólo parcialmente.

## III. CODIFICACIÓN ESCALABLE DE VÍDEO CON WAVELETS

Recientemente se han potenciado líneas de investigación que utilizan una transformada distinta a la DCT, la transformada wavelet, como núcleo del compresor escalable de vídeo [2].

La transformada wavelet es una herramienta matemática originalmente desarrollada para el análisis

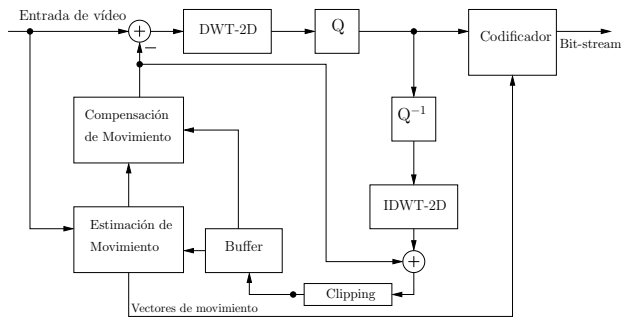


Fig. 3. Esquema de codificación “Wavelet in-the-loop”.

de datos. En la actualidad se aplica exitosamente en compresión de imágenes, utilizando sus propiedades de análisis multirresolución y descorrelación de datos. El nuevo estándar JPEG 2000 [7] para imágenes utiliza una implementación discreta [6] (*DWT Discrete Wavelet Transform*) de esta familia de transformadas. En el campo de la compresión de vídeo, se estudian esquemas de compresión basados en wavelets con compensación de movimiento capaces de proporcionar plena escalabilidad de grano fino en un rango amplio de bit-rates. Estos esquemas se pueden clasificar en las siguientes categorías:

- **Codificación del residuo con Wavelets** (*Wavelet in-the-loop*).

La estructura predictiva de codificación usada en FGS se mantiene, pero el error residual del lazo de la compensación de movimiento se codifica usando la DWT-2D en vez de la DCT.

Dado este esquema, dibujado en la Figura 3, si todo el error residual se incluye en el lazo de estimación del movimiento entonces aparecerá el error de drift si el decodificador no recibe parte de esa información. En caso contrario, disminuye la eficiencia de codificación. Además, las discontinuidades que aparecen, una vez realizada la compensación del movimiento por bloques, provocan altas frecuencias que degradan la eficiencia de codificación por ser difícil su modelización con los filtros de la DWT. Por último, la escalabilidad espacial de este esquema también padece de error de drift y/o ineficiencia de codificación ya que la compensación del movimiento ha de realizarse sólo a una resolución determinada.

- **Compensación del movimiento en el dominio transformado** (*In-band prediction*).

Primero, se realiza la DWT-2D al vídeo de entrada. La redundancia temporal presente en cada banda es explotada usando estimación y compensación de movimiento de lazo cerrado en el dominio wavelet redundante. Se utiliza el dominio redundante porque en el convencional no es posible realizar la estimación del movimiento. Con este esquema, dibujado en la Figura 4, se evita el error de drift sobre la escalabilidad espacial, ya que la estimación y compensación del movimiento se realiza separadamente en cada nivel de resolución. Sin embargo, este esquema su-

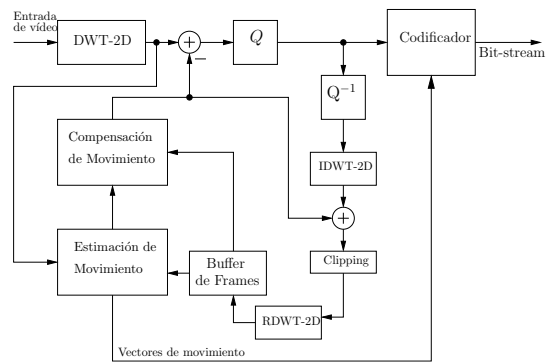


Fig. 4. Esquema de codificación “In-band prediction”.

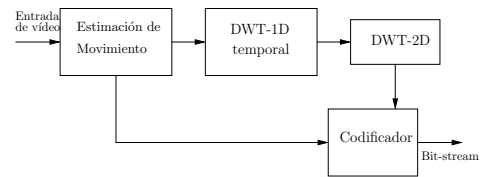


Fig. 5. Esquema de codificación “Wavelet Interframe”.

fre de la aparición del drift causado por el lazo cerrado de predicción entre frames.

- **Wavelet temporal en el dominio de la imagen** (*Wavelet Interframe*).

En el esquema de codificación escalable *Wavelet Interframe*, dibujado en la Figura 5, a la secuencia de imágenes de vídeo se le realiza la estimación del movimiento y compensación (o descorrelación) temporal. La descorrelación temporal consiste en aplicar la transformada DWT-1D sobre la dimensión del tiempo en la secuencia de imágenes que forman el vídeo. Esta transformada DWT-1D temporal está guiada por la estimación de movimiento convencional. Posteriormente, la información residual se trata con la DWT-2D.

Esta categoría no emplea una estructura temporal recursiva para eliminar la redundancia temporal. En su lugar, realiza estimación del movimiento + DWT-1D. Sin embargo, la escalabilidad espacial es limitada dado que la descorrelación temporal a resolución completa de las imágenes crea error de drift si descodificamos usando sólo la información de resoluciones más pequeñas. Además, existe aun el problema de las discontinuidades que aparecen tras la descorrelación temporal y que afectan a la eficiencia en la codificación del residuo.

- **DWT-1D temporal en el dominio transformado** (*RDWT-2D + DWT-1D*).

En este esquema, primero se realiza la transformada DWT-2D espacial. Luego, sobre las imágenes en el dominio transformado redundante RDWT-2D, se realiza estimación del movimiento y descorrelación temporal con la transformada DWT-1D.

Las propiedades de este esquema, mostrado en la Figura 6 son las más ideales para obtener cual-

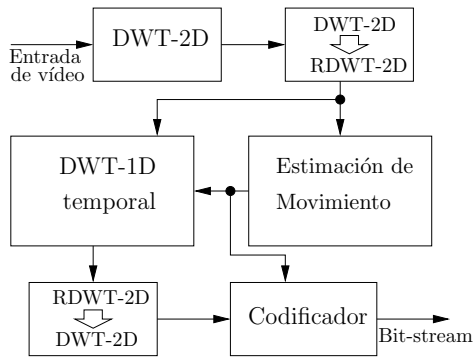


Fig. 6. Esquema de codificación “RDWT-2D + DWT-1D”.

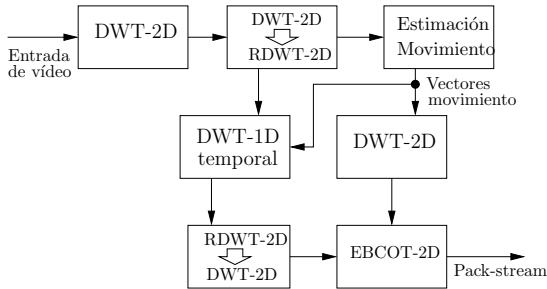


Fig. 7. Esquema de codificación FSVC-PS.

quier clase de escalabilidad de grano fino a cualquier bit-rate de descodificación. Sin embargo, tiene el inconveniente de que los distintos tipos de movimiento y aparición de objetos, que pueden ocurrir en una secuencia de vídeo, determinan la longitud óptima de filtro a la que debe ser aplicada la DWT-1D temporal. Por ejemplo, si en un objeto aparece en el frame  $x$  y desaparece en el frame  $x+10$ , la longitud de la DWT-1D aplicada sobre la posición de dicho objeto debe ser igual a 11, abarcando desde el frame  $x$  hasta el  $x+10$ . El tratamiento de este problema aumenta la complejidad de esta clase de codificadores en comparación con los convencionales.

#### IV. CODIFICACIÓN FSVC-PS (*Fully Scalable Video Codec with Packetized Stream*)

La estructura de codificación escalable de vídeo más prometedora de las vistas hasta ahora es la “DWT-1D temporal en el dominio transformado” (Figura 6). Actualmente muchos esfuerzos se están llevando a cabo en esa dirección [1] [8]. En esta sección presentamos una implementación basada en esta estructura de codificación y cuyo esquema se muestra en la Figura 7. El resultado de aplicar un codificador como éste es un *pack-stream* de vídeo totalmente escalable.

##### A. Descripción de FSVC-PS

A continuación se describe, a grandes rasgos, el proceso de codificación:

1. Leer un GOP (*Group of Pictures*) de  $G$  imágenes. Procesar la secuencia de imágenes en GOP’s proporciona las siguientes ventajas:

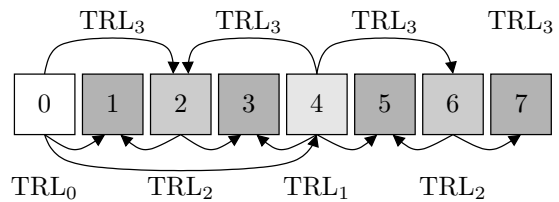


Fig. 8. Ejemplo de los niveles de resolución temporal.

- Permite el avance y el retroceso rápido a lo largo de la secuencia de imágenes.
  - Incrementa la resistencia a los errores de transmisión.
  - Acota los requerimientos de memoria del codec.
2. Calcular la 2D-DWT espacial a cada una de las imágenes del GOP. Aplicar la 2D-DWT en primer lugar aporta las siguientes ventajas:
    - Es posible aplicar filtros conocidos (como los usados en JPEG 2000) que sabemos que dan un excelente resultado sobre el dominio de la imagen (o espacial).
    - Como la 2D-DWT es expansiva, la estimación de movimiento realizada por bloques sobre el dominio wavelet redundante no genera el problema de *blocking*, típico en secuencias de vídeo comprimidas a bajos bit-rates con los estándares MPEG.
    - Gracias a la representación multiresolución de las imágenes, los algoritmos de estimación y compensación de movimiento se pueden aplicar primero sobre los niveles de resolución más bajos con lo que la búsqueda a mayores niveles de resolución puede verse significativamente acelerada.
  3. Aplicar la DWT-1D sobre el dominio temporal con lo que obtenemos  $\log_2(G) + 1$  niveles de resolución temporales por GOP. La transformada wavelet unidimensional aplicada sobre el dominio wavelet es guiada mediante estimación de movimiento. El residuo resultado presenta una menor entropía (es más comprimible) y se genera un conjunto de vectores de movimiento cuyo número depende del nivel de resolución. A pesar de los vectores de movimiento, la reducción en la entropía de los residuos provoca que las tasas de compresión aumenten en comparación con la técnicas sin compensación del movimiento (por ejemplo, Motion-JPEG 2000).  
Dependiendo del tipo de filtro DWT-1D utilizado para la descorrelación temporal (compensación del movimiento) podemos obtener diversas estructuras. Una de ellas podría ser la siguiente: para el primer nivel de resolución utilizamos la primera imagen del GOP (la imagen 0) que no depende de ninguna otra. El siguiente nivel de resolución (el 1) se sitúa en la mitad del GOP (imagen 4 en el ejemplo de la Figura 8) y para encontrarlo es necesario el primer nivel de resolución temporal y los vectores de movimiento utilizados en la compensación de la imagen 4 a

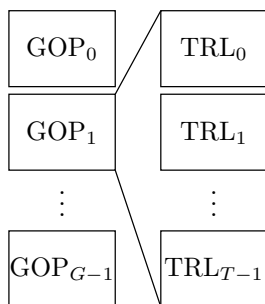


Fig. 9. Estructura del pack-stream propuesto.

partir de la 0. El nivel de resolución 2 está formado por dos imágenes (la 2 y la 6) que pueden obtenerse a partir de las imágenes de los niveles de resolución anteriores tal y como se muestra en la Figura 8. Finalmente, con el nivel de resolución 3 completamos todas las imágenes del GOP. Los campos de movimiento necesarios para el ejemplo mostrado en la Figura 8 serían 11.

4. Utilizar EBCOT (el codificador entrópico de JPEG 2000) para comprimir tanto los residuos como los campos de movimiento. El resultado es un JPEG 2000 pack-stream que presenta, dentro de cada nivel de resolución temporal, escalabilidad espacial y SNR.

Una forma de comprimir cada nivel de resolución (que en general implica la compresión de varias imágenes residuo más los correspondientes vectores de movimiento para reconstruirlas) consistiría en crear una imagen con tantos tiles como imágenes y campos de movimiento existen<sup>1</sup> y a continuación aplicar el algoritmo estándar de JPEG 2000 para comprimir dichas imágenes (eliminando la fase de cálculo de la 2D-DWT). El resultado es, para cada nivel de resolución temporal, un pack-stream compatible con JPEG 2000.

#### B. Estructura del pack-stream

El pack-stream generado por el codificador propuesto es compatible con la maquinaria JPEG 2000 y Motion-JPEG 2000 ya que finalmente se utiliza EBCOT para comprimir los residuos y los campos de movimiento.

La estructura del pack-stream es presentada en la Figura 9. En ella se ha supuesto que los paquetes se ordenan primero por GOP y luego por nivel de resolución espacial. La progresión que se realice de forma más interna, dentro de cada nivel de resolución, debería depender de las necesidades de la aplicación. Por ejemplo, en el caso de una transmisión de vídeo a través de Internet, si los clientes demandan escalabilidad espacial en primer lugar, la progresión interna debería de ser, por ejemplo, RLCP. En cualquier caso, una vez que los paquetes han sido creados todos los tipos de escalabilidad (temporal, SNR, espacial,

<sup>1</sup>Los campos de movimiento, que tienen dos componentes X e Y, pueden procesarse como dos imágenes independientes que ya se encuentran en un dominio multiresolución, como los residuos.

por componente y por precinto) están disponibles en el pack-stream.

## V. CONCLUSIONES

El objetivo principal de la codificación de vídeo ha consistido tradicionalmente en la optimización de la calidad del mismo a un bit-rate determinado. Sin embargo, esto ha cambiado por la aparición de aplicaciones de vídeo en redes de computadores tales como el *streaming* de vídeo por Internet, porque en el momento de codificación del vídeo normalmente no se conoce ni la capacidad del receptor ni el ancho de banda disponible para la transmisión. El nuevo objetivo para estas aplicaciones es realizar una codificación escalable de vídeo. Es decir, que el vídeo comprimido resultado de esta codificación tenga propiedades de escalabilidad.

En este artículo hemos revisado las distintas técnicas de escalabilidad que utilizan los estándares MPEG y las estructuras de codificación escalable basadas en la transformada wavelet. Los vídeos codificados con características de escalabilidad permiten ser reproducidos por decodificadores con distintos recursos y ser transmitidos por canales con ancho de banda variable o no predecible en el momento de la codificación. Por último, hemos presentado las primeras ideas de un nuevo codificador de vídeo altamente escalable.

## REFERENCIAS

- [1] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis. Fully-Scalable Wavelet Video Coding using In-Band Motion-Compensated Temporal Filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2003.
- [2] S. Cho and W. A. Pearlman. A full-featured, error resilient, scalable wavelet video codec based on the set partitioning in hierarchical trees (spiht) algorithm. *IEEE Transactions on circuits and systems for video technology*, 12:151–170, 2002.
- [3] S. Li, F. Wu, and Y. Q. Zhang. Study of a New Approach to Improve FGS Video Coding efficiency. ISO/IEC JTC1/SC29/WG11, MPEG99/M5583, December 1999.
- [4] W. Li, F. Ling, and H. Sun. Bitplane Coding of DCT Coefficients. ISO/IEC JTC1/SC29/WG11, MPEG97/M2691, October 1997.
- [5] K. Shen and E. J. Delp. Wavelet Based Rate Scalable Video Compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 9:109–122, February 1999.
- [6] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.
- [7] D. Taubman and M. Marcellin. *JPEG2000: Image compression fundamentals, standards and practice*. Kluwer Academic Publishers, November 2001.
- [8] Y. Wang, S. Cui, and J.E. Fowler. 3D video coding using redundant-wavelet multihypothesis and motion-compensated temporal filtering. In *International Conference on Image Processing*, volume 2, pages 755–758, 2003.
- [9] Feng Wu, Shipeng Li, and Ya-Qin Zhang. A framework for efficient progressive fine granularity scalable video coding. *IEEE Transactions on circuits and systems for video technology*, 11(3), March 2001.