

COMPRESIÓN DE VÍDEO MEDIANTE TÉCNICAS DE POST-PROCESAMIENTO

Uno de los requisitos fundamentales en el almacenamiento y transmisión de información multimedia es el disponer de técnicas eficaces de compresión y descompresión de datos en el emisor y receptor del canal de comunicación. Para el caso particular de dispositivos móviles de última generación, los cuales transmiten a través de canales inalámbricos caracterizados por bajos anchos de banda y alta probabilidad de introducción de errores, resulta aún más evidente la necesidad de disponer de altas tasas de compresión así como de técnicas que proporcionen mayor inmunidad de la información ante el ruido.

En este artículo se proponen algoritmos de post-procesamiento de bajo coste computacional que permiten, de manera eficaz, compaginar altas tasas de compresión de vídeo con la obtención de imágenes de gran calidad en el receptor. Estas técnicas están basadas en el análisis previo de las secuencias de vídeo a comprimir, mejorando, independientemente de las características de las mismas, los resultados obtenidos en trabajos realizados previamente.

One of the main constraints found in storing and sending multimedia information is to have efficient compression and decompression techniques available at both sides of the communication channel. For the particular case of mobile multimedia devices, where data are transmitted through wireless links characterized by reduced bandwidths and high bit error rates, it is even more noticeable not only to achieve high compression rates but also to have efficient techniques in order to protect the information against the errors introduced by the channel.

Several low-cost post-processing algorithms are presented in this paper in order to obtain high compression rates and optimal quality of the decoded signal at the receiver. These techniques are based on the previous analysis of the video sequences to be compressed, improving, with independence of their characteristics, the results presented in previous works.

Sebastián López

Gustavo M. Callicó

José Fco. López

Roberto Sarmiento

PRESENTACIÓN

En este artículo se presentan nuevas soluciones para aumentar la tasa de compresión de vídeo en aplicaciones de telefonía móvil de nueva generación mediante la introducción de mejoras en la etapa clave dentro de este tipo de sistemas: la estimación de movimiento.

INTRODUCCIÓN

La sociedad del conocimiento se basa en la importancia que se le da

a la transformación de la información en conocimiento. Es un hecho innegable que la información supone un privilegio para quien la posee (siempre se ha dicho que la información es poder) si bien su almacenamiento, gestión y transmisión se hacen cada vez más complicadas debido al gran volumen de datos que en sí mismo representa. En el escenario de los dispositivos multimedia de última generación, representa un gran reto el buscar soluciones que permitan el manejo de tan vasta cantidad de información

En el escenario de los dispositivos multimedia de última generación, representa un gran reto el buscar soluciones que permitan el manejo de tan vasta cantidad de información en forma de bits.

en forma de bits, y se hace necesario el desarrollar técnicas que permitan, de forma cómoda y fiable, su gestión.

Para el caso particular de estas aplicaciones, estos problemas se pueden apreciar con un ejemplo muy sencillo. Supongamos que se desea transmitir una secuencia de vídeo (formada por la sucesión de fotogramas, a razón de 30 fotogramas por segundo) en la cual cada fotograma consta de 176x144 píxeles, y a su vez, cada píxel viene representado por 12 bits. La transmisión de un vídeo de 2 minutos de vídeo a través de un canal de comunicación típico que acepta un máximo de 64 Kbits cada segundo, tardaría... ¡¡¡4 horas y 45 minutos!!! . Por lo tanto es incuestionable que, para realizar el almacenamiento y transmisión de vídeo digital de manera eficiente, es absolutamente imprescindible contar con algoritmos de compresión en el emisor que permitan reducir el volumen de datos a transmitir. De igual manera, es completamente necesario disponer de técnicas de descompresión que, en sintonía con el compresor utilizado, permitan visualizar la secuencia original en el receptor con la menor pérdida de información posible, tal y como se muestra en la Figura 1.

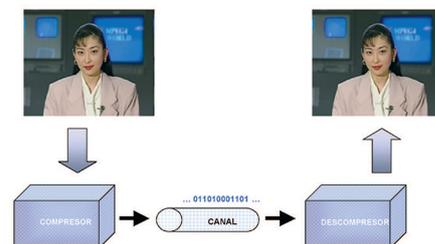


Figura1. Proceso general de compresión, transmisión y descompresión de una secuencia de vídeo.

En este artículo se presentan nuevas soluciones para aumentar la tasa de compresión de vídeo digital particularizada para el caso de transmisión de vídeo entre dis-



Figura 2. Ejemplo de dispositivo móvil multimedia.

positivos multimedia de nueva generación a través de canales inalámbricos, las cuales han sido desarrolladas en la División de Diseño de Sistemas Integrados (DSI) del Instituto Universitario de Microelectrónica Aplicada (IUMA) adscrito a la Universidad de Las Palmas de Gran Canaria.

TRANSMISIÓN DE VÍDEO SOBRE CANALES INALÁMBRICOS

Las comunicaciones multimedia a través de canales inalámbricos posee tres características principales:

- Muy bajo ancho de banda. Esta característica determina que, cuando se están manejando grandes cantidades de información, sea absolutamente necesario contar con algoritmos eficientes de compresión y descompresión de información en los extremos emisor y receptor respectivamente, para de esta forma acelerar la transmisión.
- Alta tasa de errores de bit. Este hecho hace que la información digital que se manda a través de un canal inalámbrico desde el emisor hacia el receptor llegue con errores a este último.
- Naturaleza móvil de los dispositivos. Los terminales con los que se suele trabajar en aplicaciones inalámbricas son normalmente de carácter móvil, y por lo tanto, dichos dispositivos deberán presentar un tamaño reducido en su implementación final, así como un bajo consumo de potencia cuando se encuentran transmitiendo o recibiendo información.

Cuando la información que se desea transmitir es vídeo digital, los dos primeros factores cobran aún mayor relevancia. En primer lugar, este tipo de información multimedia al que nos referimos, suele venir representada por un gran volumen de datos, siendo ésta la principal razón por la cual durante estos últimos años se han desarrollado numerosas técnicas de compresión, como son los estándares JPEG y JPEG2000 para la compresión de imágenes estáticas y los estándares MPEG-1, MPEG-2, MPEG-4, H.261, H.263 y H.264 para la compresión de vídeo. En segundo lugar, cuando se transmite información de naturaleza visual, los errores en la transmisión son absolutamente intolerables, pues se pueden obtener efectos tan devastadores como los mostrados en la Figura 3. En dicha Figura se presenta el resultado de haber comprimido la imagen original (izquierda) con el estándar JPEG, siendo enviada por un canal con una *tasa de error de bit (BER, Bit Error Rate)* relativamente baja, obteniéndose en el receptor, tras el proceso de descompresión, la imagen mostrada a la derecha de la Figura 3, en la cual se aprecian claramente los efectos de los errores presentes en el canal.

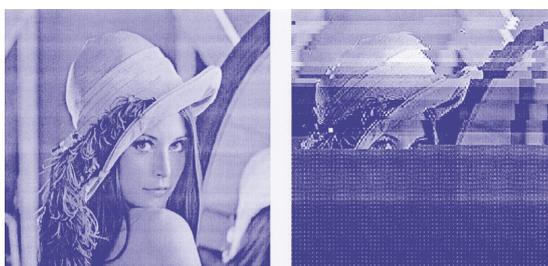


Figura 3. Efecto de los errores en la transmisión de imágenes comprimidas.

Llegados a este punto, es importante destacar que los estándares de vídeo anteriormente mencionados consiguen altas tasas de compresión basándose en esquemas altamente predictivos, eliminando para ello las redundancias espaciales y temporales presentes

en una señal de vídeo digital (Oliver, 2001), tal y como se aprecia en la Figura 4. Se entiende por redundancia espacial en una secuencia de vídeo al hecho de que, normalmente, los píxeles que ocupan posiciones vecinas dentro de un mismo fotograma poseen valores similares. Por otro lado, la redundancia temporal se refiere al hecho de que píxeles de fotogramas consecutivos en el tiempo poseen valores muy similares para las mismas posiciones espaciales.

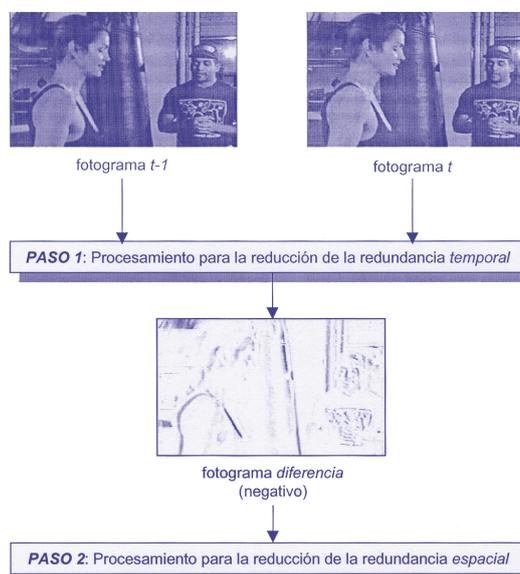


Figura 4. Reducción de redundancias espaciales y temporales en una secuencia de vídeo.

Por lo tanto, y debido fundamentalmente a esta última razón, si se produce un error en la transmisión se obtendrá el efecto conocido como propagación espacial y temporal de los errores (Sadka, 2002). Además se debe considerar que, sin duda alguna, la explotación de las redundancias temporales anteriormente mencionadas es el factor más determinante a la hora proporcionar una elevada tasa de compresión con un nivel de calidad óptimo. Dicha tarea es realizada por un bloque funcional claramente diferenciado dentro del esquema general de un compresor de vídeo: el estimador de movimiento. Dicho bloque presenta un

Se presentan nuevas soluciones para aumentar la tasa de compresión de vídeo digital particularizada para el caso de transmisión de vídeo entre dispositivos multimedia de nueva generación a través de canales inalámbricos.

coste computacional mucho más elevado que el resto de los elementos que conforman un codificador de vídeo, por lo que el objetivo de este proyecto de investigación se centra en optimizar esta etapa para obtener mejores prestaciones en el caso particular de comunicaciones inalámbricas.

LA ETAPA DE ESTIMACIÓN DE MOVIMIENTO

Los algoritmos más utilizados para la implementación de la etapa de estimación de movimiento en los codificadores de vídeo actuales son los denominados *algoritmos de ajuste de bloques (BMA, Block Matching Algorithm)* que destacan por su simplicidad y robustez.

Estos algoritmos se basan en comparar cada bloque de $N \times M$ píxeles -en la mayoría de los estándares se cumple que $N=M=16$ - de la imagen actual (bloque de referencia) sobre un área reducida de una imagen anteriormente codificada. El resultado de la estimación es un vector, denominado vector de movimiento, que identifica el desplazamiento entre el bloque de referencia y la predicción. La Figura 5 muestra el proceso de obtención del vector de movimiento de coordenadas (u,v) utilizando un algoritmo de *block-matching*, donde el parámetro p define las dimensiones del área de búsqueda.

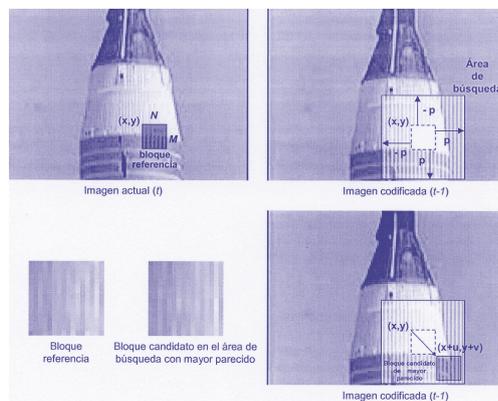


Figura 5. Funcionamiento de los algoritmos de ajuste de bloques.

En definitiva, las coordenadas (u,v) definen qué bloque dentro del área de búsqueda es el “más parecido” al bloque de referencia. Para tal propósito, el criterio más utilizado para la comparación entre un bloque de referencia situado en la posición (x,y) con un bloque candidato situado en la posición (i,j) del área de búsqueda se denomina *SAD (Sum of Absolute Differences)* y viene dado por:

$$SAD(i, j) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} |r(x, y) - s(x+i, y+j)|$$

donde $r(x,y)$ es el valor del píxel en la posición (x,y) del bloque base, y $s(x+i, y+j)$ es el valor del píxel en la posición $(x+i,y+j)$ del bloque candidato. Básicamente se trata de acumular la diferencia entre los píxeles del bloque de referencia y el bloque candidato. Cuanto más parecidos sean los bloques, los píxeles van a tener valores similares, y el coeficiente *SAD* tenderá a ser mínimo. El proceso de búsqueda del vector de movimiento óptimo se basa pues en encontrar qué bloque candidato en el área de búsqueda tiene el mínimo *SAD*.

ESTIMACIÓN DE MOVIMIENTO POR BÚSQUEDA EXHAUSTIVA

Dentro de los algoritmos de estimación de movimiento por ajuste de bloques, el más popular es, sin duda, el *algoritmo por búsqueda exhaustiva (FSBMA, Full Search Block Matching Algorithm)*, el cual permite, dentro de los de su género, obtener las mayores tasas de compresión a costa de evaluar absolutamente todas las posiciones posibles dentro del área de búsqueda (Jain, 1981). Además, el flujo de datos en los algoritmos de búsqueda exhaustiva es extremadamente regular, lo cual facilita enormemente su implementación hardware (López, 2002). Sin embargo, el principal problema que presenta este tipo de algoritmo, aparte de su elevado coste computacional, es que se basa exclusiva-

mente en minimizar una métrica de error (*SAD*), sin tener en cuenta la consistencia de los vectores de movimiento calculados. En el contexto de la estimación de movimiento, la consistencia del campo de vectores de movimiento se refiere al hecho de que no existan grandes diferencias entre vectores de movimiento de bloques vecinos, o dicho de otra manera, que el conjunto de vectores de movimiento obtenido sea “suave”. Esta suposición está completamente justificada, pues, lo que no parece lógico es que bloques de píxeles que pertenecen a un mismo objeto dentro de una misma imagen presenten movimientos con velocidades y direcciones diametralmente opuestas.

Sin embargo, una situación muy común que se produce tras una estimación de movimiento por búsqueda exhaustiva es la obtención de vectores de movimiento de bloques vecinos completamente dispares. Debido a que en los estándares de compresión de vídeo los vectores de movimiento se codifican de forma diferencial con respecto a los vectores de bloques vecinos, este hecho provoca que se emplee un número excesivo de bits para la transmisión de todos los vectores de movimiento correspondientes a un frame.

El trabajo de numerosos investigadores dentro de este área ha demostrado que un suavizado de los vectores de movimiento puede resultar altamente beneficioso para reducir el ancho de banda necesario en la transmisión de señales de vídeo (Jing, 2003). Estas investigaciones están basadas en la obtención, en la medida de lo posible, de vectores de movimiento verdaderos (Chen, 1997) entendiéndose por vector de movimiento verdadero aquel que reproduce el movimiento real de la escena que se analiza. Es bien sabido que, determinados algoritmos de estimación de movimiento como son los basados en

flujo óptico (Horn, 1981) o los algoritmos recursivos a nivel de píxel (Walker, 1984), son capaces de producir vectores de movimiento verdaderos, pero su elevadísimo coste computacional hace que su uso no resulte conveniente para aplicaciones en tiempo real con bajo consumo de potencia, como son las que nos ocupan en este caso. Además, las aportaciones realizadas hasta la actualidad relacionadas con el suavizado de vectores de movimiento en algoritmos de ajuste de bloques presentan un inconveniente común: los resultados son extremadamente dependientes de la secuencia de vídeo que se desea comprimir. Este hecho hace que los resultados obtenidos tras la aplicación de algoritmos de suavizado sean satisfactorios para algunas secuencias pero no para otras, llegando a situaciones en las que incluso se puede obtener una peor tasa de compresión tras el proceso de suavizado que la obtenida con el algoritmo de ajuste bloques original.

Debido a las razones comentadas en esta sección, el objetivo principal de la presente investigación es idear nuevas técnicas de suavizado que se adapten a las características de la secuencia a comprimir, con el objetivo de mejorar la tasa de compresión independientemente de la secuencia a analizar. Para ello, y como primer paso, es muy importante obtener un conjunto de parámetros característicos de la imagen que nos permitan extraer conclusiones acerca del grado de suavidad y veracidad de los vectores de movimiento obtenidos tras aplicar un proceso de búsqueda exhaustiva.

OBTENCIÓN DE PARÁMETROS

Para la consecución del objetivo propuesto se desarrolló el esquema detallado en la Figura 6. La metodología seguida se explica a continuación. Primero, se toma un fotograma de una secuencia de vídeo cualquiera y se crea a partir de

El trabajo de numerosos investigadores dentro de este área ha demostrado que un suavizado de los vectores de movimiento puede resultar altamente beneficioso para reducir el ancho de banda necesario en la transmisión de señales de vídeo

Esta etapa de post-procesamiento se caracteriza por su bajo coste computacional, además de por ser capaz de obtener resultados satisfactorios independientemente de la secuencia de vídeo a comprimir.

este fotograma una secuencia de diez fotogramas mediante la introducción de nueve vectores de movimiento diferentes. Para ello se utiliza el bloque MOVER, el cual desplaza cada píxel de la imagen de referencia según lo indicado por el vector de movimiento correspondiente. Una vez que se ha creado una secuencia en la cual el movimiento está perfectamente controlado, se aplica sobre ella un algoritmo de ajuste de bloques por búsqueda exhaustiva, para finalmente comparar los vectores de movimiento obtenidos con los introducidos para la generación de la secuencia de vídeo, determinándose si los vectores de movimiento obtenidos eran verdaderos o falsos. Como se comentó en el apartado anterior, lo que se pretende realizar es una caracterización de los vectores de movimiento obtenidos con algoritmos de ajuste de bloques a través de parámetros que nos puedan dar una idea acerca de la veracidad de dichos vectores de movimiento. Para ello, se han introducido dos nuevos parámetros denominados *Intra_SAD* y *SAD_deviation* que pasan a ser explicados a continuación.

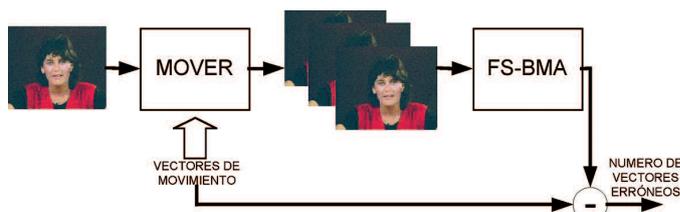


Figura 6. Experimento diseñado para la obtención de parámetros.

En primer lugar, es lógico pensar que los bloques que presentan una alta varianza espacial tienen una alta probabilidad de presentar vectores de movimiento verdaderos. Esto es debido a que este tipo de bloques presentarán un buen *matching* en un solo lugar del área de búsqueda. Según esta idea, el primer parámetro introducido para el proceso de caracterización es el que se ha denominado *Intra_SAD*, definido de la siguiente forma:

$$Intra_SAD = \sum_{i,j} |p(i,j) - m|$$

donde $p(i,j)$ representa el valor del píxel en la posición (i,j) dentro del bloque que se está analizando y m representa la media de todos los valores de los píxeles que se encuentran dentro de dicho bloque.

Por otra parte, es importante recordar que de entre todas las posiciones evaluadas dentro del área de búsqueda por un algoritmo del tipo *FSBMA*, el vector de movimiento finalmente elegido es calculado a partir de la posición que presenta la menor distorsión (*SAD*) con el bloque de referencia. Parece evidente pensar que si una posición candidata presenta un *SAD* muchísimo menor que el resto de las posiciones evaluadas, el vector resultante presenta una alta probabilidad de ser verdadero. Por ello, el segundo parámetro denominado *SAD_deviation* es definido como:

$$SAD_deviation = \sum_{u,v} |SAD(u,v) - SAD_min|$$

donde $SAD(u,v)$ representa el *SAD* de cada una de las posiciones evaluadas y SAD_min representa el mínimo obtenido y que es, tal y como ya se ha comentado, el que da el vector de movimiento para el bloque de referencia que se está analizando.

Haciendo uso del esquema detallado en la Figura 6 e introduciendo los dos nuevos parámetros comentados anteriormente, se obtuvieron los resultados que se muestran en la Figura 7, en los cuales se analiza la veracidad de la componente horizontal de los vectores de movimiento obtenidos en función de los dos parámetros previamente introducidos. Como se puede observar, se presentan seis gráficas en las cuales aparecen los diferentes bloques de 16 x 16 píxeles para los que se obtuvieron vectores de movimiento, estando representado en el eje horizontal el valor del *in-*

tra_SAD de cada uno de estos bloques y en el eje vertical el SAD_deviation derivado de la búsqueda exhaustiva llevada a cabo para el cálculo del vector de movimiento asociado a ese bloque. En la esquina superior izquierda se presentan los bloques para los que se obtuvieron vectores de movimiento verdaderos (error = 0) mientras que en el resto se presentan los bloques con vectores de movimiento falsos (error = 1, error = 2, error = 3, error = 4, y error ≥ 5).

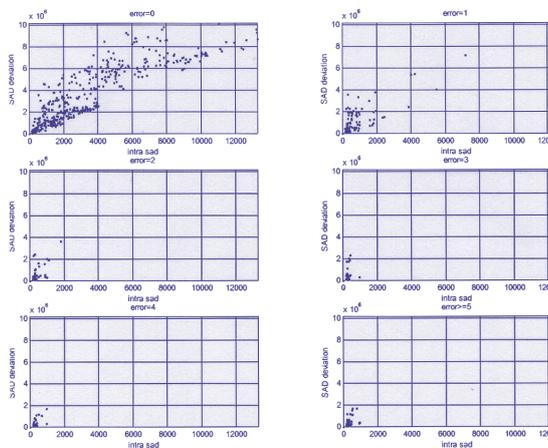


Figura 7. Resultados obtenidos en la caracterización de vectores de movimiento.

Tal y como se esperaba, la inmensa mayoría de los bloques con vectores de movimiento falsos se encuentran en una región caracterizada por bajos valores de Intra_SAD (≤ 2000) y de SAD_deviation ($\leq 2 \cdot 10^6$), mientras que la mayor parte de los vectores de movimiento se encuentran fuera de esta región. Este hecho es más evidente a medida que la cantidad de error en los vectores de movimiento aumenta. Por otra parte, es importante señalar que en esta región de bajos valores de ambos parámetros se encuentra también una cantidad importante de vectores verdaderos, lo cual nos lleva a afirmar que este sencillo juego de parámetros que se ha introducido, si bien representa una potente herramienta en la caracterización de vectores de movimiento, no está todavía “completo” pues no permite distinguir estos vectores verdaderos de los falsos. Sin embargo, tal y como se demostrará en el siguiente aparta-

do, aunque este conjunto de parámetros no es completo, permite obtener algunos resultados interesantes. En particular, ambos parámetros permiten definir una etapa de post-procesamiento de vectores de movimiento a través de la cual se obtiene una mejor relación de compresión de vídeo para canales con muy bajo ancho de banda, como es el caso que nos ocupa.

ETAPA DE POST-PROCESAMIENTO DE VECTORES DE MOVIMIENTO

Los algoritmos de post-procesamiento de vectores de movimiento han sido introducidos en la literatura reciente (Callicó, 2003) con el objetivo de obtener a su salida un mayor ratio de vectores verdaderos que el obtenido tras la aplicación de la estimación de movimiento por búsqueda exhaustiva o de cualquier otro algoritmo de estimación de movimiento por ajuste de bloques. La Figura 8 muestra el esquema general de post-procesamiento de vectores de movimiento a partir de los resultados obtenidos con un estimador de movimiento genérico.

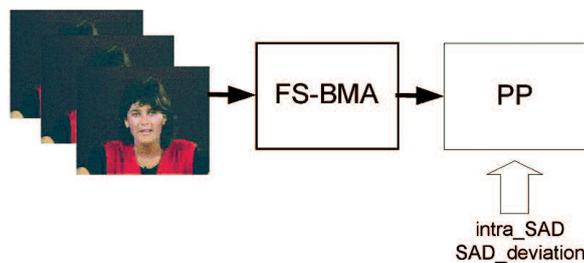


Figura 8. Esquema de post-procesamiento de vectores de movimiento.

Sin embargo, no debemos olvidar que el objetivo en este proyecto de investigación no es otro que obtener un conjunto de vectores de movimiento encaminados a producir la mayor tasa de compresión posible, y por lo tanto, estos vectores no tienen por qué coincidir con los vectores de movimiento verdaderos. Debido a esta razón, la etapa de post-procesamiento (PP) que se ha diseñado responde a la siguiente ecuación:

$$mv^{pp} = \frac{SAD_Deviation_reduced(0) \cdot mv(0) \cdot Intra_SAD(0) + \sum_{i=1}^8 w(i) \cdot mv(i) \cdot Intra_SAD(i)}{SAD_Deviation_reduced(0) \cdot Intra_SAD(0) + \sum_{i=1}^8 w(i) \cdot Intra_SAD(i)}$$

donde $mv(i)$ indica cada uno de los vectores de movimiento dentro de la vecindad del vector correspondiente al bloque de referencia y $w(i)$ un peso específico que se le da a cada vector, con el objetivo de dar mayor prioridad a determinados vectores frente a otros. La notación que se ha seguido para cada uno de los vectores vecinos es la que se indica en la Figura 9.

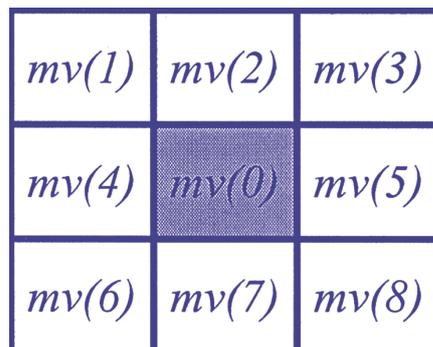


Figura 9. Vecindad de vectores de movimiento.

Con el objetivo de reducir el coste computacional del algoritmo presentado, se ha sustituido el parámetro $SAD_Deviation$ por $SAD_Deviation_reduced$, definido como sigue:

$$SAD_Deviation_reduced = SAD_max - SAD_min$$

donde SAD_max indica el máximo valor de SAD encontrado durante el proceso de búsqueda y SAD_min el mínimo de estos valores.

Como se puede deducir a partir de la ecuación de post-procesamiento, el proceso que se lleva a cabo no es otro que el de realizar una media ponderada de todos los vectores de movimiento que se encuentra dentro de una vecindad de 3x3 bloques alrededor del vector de movimiento que se desea post-procesar. En esta media de vectores de

movimiento tendrán un mayor protagonismo aquellos vectores asociados a bloques de píxeles vecinos con un alto $Intra_SAD$. Por otro lado, el vector central es también pesado por su $SAD_Deviation_reduced$ con el objetivo de calibrar la sensibilidad que presenta el bloque central ante variaciones en el vector de movimiento originalmente calculado por el algoritmo $FSBMA$.

La etapa de post-procesamiento presentada ha sido introducida dentro un codificador de vídeo H.263, que se caracteriza por utilizar el algoritmo de búsqueda exhaustiva durante la etapa de estimación de movimiento. Se han realizado pruebas para tres secuencias (*Guardacostas*, *Miss América* y *Foreman*), siendo éstas muy diferentes entre sí tanto en sus texturas como en la cantidad de movimiento que presentan. El primer fotograma de cada una de estas secuencias aparece en la Figura 10.



Figura 10. Primer fotograma de las secuencias de test.

Cabe destacar que se han elegido tres secuencias ampliamente utilizadas en el testeo de algoritmos de compresión de vídeo y que además, la secuencia denominada *Guardacostas* se ha dividido en dos subsecuencias con el objetivo de analizar el efecto de la presencia de movimientos de cámara con movimientos tanto locales como globales. Los resultados obtenidos se muestran en la Figura 11 en la cual se aprecia una clara mejora de las prestacio-

de Educación, Cultura y Deporte le otorgó una Mención Especial por su expediente académico en el año 2002. Durante el año 2003 realizó una estancia de investigación en los centros de Milán y Padua (Italia) de la empresa franco-italiana STMicroelectronics, donde contribuyó al desarrollo de soluciones para la codificación de vídeo en dispositivos móviles. Es autor de varias publicaciones en revistas y congresos internacionales de reconocido prestigio dentro del área de compresión de imagen y vídeo y actualmente ejerce como Profesor Ayudante en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Las Palmas de Gran Canaria.

Dirección:

Edificio de Electrónica y Telecomunicaciones (Pabellón A)

Campus Universitario de Tafira, s/n

35017 Las Palmas de Gran Canaria.

Telf: 928 457335

Fax: 928 451083

Email:seblopez@iuma.ulpgc.es

BIBLIOGRAFÍA

Oliver J y Pérez M. (2001): *Compresión de imagen y vídeo: Fundamentos teóricos y aspectos prácticos*. Valencia. Editorial Universidad Politécnica de Valencia.

Sadka A.H. (2002): *Compressed video communications*. West Sussex (Inglaterra). John Wiley & Sons.

Jain J.R y Jain A.K. (1981), "Displacement Measurement and its application in interframe image coding", *IEEE Trans. on Communication*, vol. COM-29, págs. 1799-1808.

López J.F., Cortés P., López S. y Sarmiento R. (2002), "Design of a 270 Mhz / 340 mW processing element for high performance motion estimation system application", *Microelectronics Journal*, ELSEVIER vol. 33, pp. 1123-1134.

Jing X., Zhu C., y Chan L.-P. (2003), "Smooth constrained block matching criterion for motion estimation", *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and signal Processing (ICASSP)*, vol. 3, págs. 661-664.

Chen Y.-K. y Kung, S.Y. (1997), "Rate optimization by true motion estimation", *Proceedings of the IEEE First Workshop on Multimedia Signal Processing*, págs. 187-194.

Horn B.K.P. y Schunck B.G. (1981), "Determining Optical Flow", *Artificial Intelligence*, vol.17, págs. 185-204.

Walker D.R. y Rao K.R. (1984), "Improvedpel-recursive motion estimation" *IEEE Trans. Commun.*, vol. COM-32, págs. 1128-1134.

Callicó, G.M. (2003), *Real-Time and Low-Cost Super-Resolution Algorithms onto Hybrid Video Encoders*, Tesis Doctoral, Universidad de Las Palmas de Gran Canaria.

Patrocinador de esta investigación:

LA CAJA DE CANARIAS