Informes Técnicos Ciemat

811
Diciembre 1996

Estimación del movimiento Interframe y su aplicación a la compresión de secuencias de imágenes: una introducción

C. Crémy

Asociación EURATOM/CIEMAT para Fusión - 37

Instituto de Investigación Básica

Toda correspondencia en relación con este trabajo debe dirigirse al Servicio de Información y Documentación, Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas, Ciudad Universitaria, 28040-MADRID, ESPAÑA.

Las solicitudes de ejemplares deben dirigirse a este mismo Servicio.

Los descriptores se han seleccionado del Thesauro del DOE para describir las materias que contiene este informe con vistas a su recuperación. La catalogación se ha hecho utilizando el documento DOE/TIC-4602 (Rev. 1) Descriptive Cataloguing On-Line, y la clasificación de acuerdo con el documento DOE/TIC.4584-R7 Subject Categories and Scope publicados por el Office of Scientific and Technical Information del Departamento de Energía de los Estados Unidos.

Depósito Legal: M-14226-1995
NIPO: 238-96-001-0
ISSN: 1135-9420

Editorial CIEMAT

CLASIFICACIÓN DOE Y DESCRIPTORES

990200, 990300
COMPUTER CALCULATIONS, DATA PROCESSING, IMAGE PROCESSING, IMAGE SCANNERS, INFORMATION THEORY, MOTION, MOTION DETECTION SYSTEMS

"Estimación del movimiento interframe y su aplicación en la compresión de secuencias de imágenes: una introducción"

Crémy, C.
31 págs., 4 figrs. 17 refs.

Resumen

Con el constante desarrollo de las nuevas tecnologías de comunicación como la televisión digital, la teleconferencia, y de las aplicaciones de análisis de imágenes se manejan volúmenes crecientes de información. La transmisión y el almacenamiento de estos datos precisan el uso de técnicas de compresión. El tratar con las imagenes originales conllevaría un coste importante porque se necesitarían canales de comunicación con un amplio ancho de banda y soportes de almacenamiento de gran capacidad. La compresión de secuencias de imágenes, basada en la estimación del movimiento entre tramas o *interframe*, consiste en no considerar la información redundante entre dos tramas, esto es, la información relativa a las zonas de las imágenes de poco movimiento *interframe* como las técnicas de gradiente, *pel-recursive*, *block-matching* y su aplicación en la compresión de secuencias de imágenes.

"Interframe motion estimation and its application to image sequence compression: an introduction"

Crémy, C.
31 págs., 4 figrs. 17 refs.

Abstract

With the constant development of new communication technologies like, digital TV, teleconference, and the development of image analysis applications, there is a growing volume of data to manage. Compression techniques are required for the transmission and storage of these data. Dealing with original images would require the use of expansive high bandwidth communication devices and huge storage media. Image sequence compression can be achieved by means of interframe estimation that consits in retrieving redundant information relative to zones where there is little motion between two frames. This paper is an introduction to some motion estimation techniques like gradient techniques, pel-recursive, block-matching, and its appication to image sequence compression.

ÍNDICE

I- INTRODUCCIÓN

Las tecnologías modernas de comunicación y de análisis de imágenes, tales como la telefonía vídeo, la televisión digital, la teleconferencia, la predicción meteorológica o la tomografía asistida por ordenador, necesitan una capacidad creciente de transmisión y de almacenamiento de datos. Las imágenes que manejan contienen una cantidad importante de información y se necesitan sistemas de comunicación con un amplio ancho de banda. Esto significa un coste importante para los canales de comunicación y del soporte de almacenamiento.

El objetivo de la compresión es la reducción de estos costes mientras se preserva una calidad de las imágenes satisfactoria. La noción de calidad depende fuertemente del contexto en el que se manejan las imágenes. En otros términos, el propósito de la compresión es la disminución del espacio de memoria necesario para el almacenamiento y la reducción del ancho de banda para la transmisión.

La compresión de datos en imágenes corresponde a la minimización de la información necesaria para representarlas. En el procesado de imágenes, cada elemento de una imagen, llamado *pel* (*picture element*), se cuantifica en un número fijo de bits, es decir, que una imagen se almacena y se transmite digitalmente. El procedimiento de compresión se acompaña en general de una distorsión al reconstruir la señal original. La eficiencia de un algoritmo se mide por su capacidad de compresión, por el nivel de la distorsión que supone y por la complejidad de su implementación que es particularmente importante para la definición del material utilizado. Para la transmisión vídeo, las técnicas de compresión están condicionadas fuertemente por consideraciones de tiempo real que tienden a limitar el tamaño y la complejidad del material. Para aplicaciones que sólo necesitan almacenamiento, los requisitos para los compresores son menos rigurosos ya que gran parte del procesado se puede realizar en diferido. Sin embargo, la descompresión y la reconstrucción son tareas que no tienen porque suponer un tiempo de respuesta demasiado grande.

Los métodos de compresión pueden clasificarse en tres categorías. Los métodos de la primera categoría explotan una redundancia estadística presente en los datos y que se relaciona con la predecibilidad, el carácter

aleatorio y la regularidad de estos datos. Con estos métodos se intenta representar una imagen con un arreglo de datos, sin o con poca redundancia, a partir de los cuales se puede recuperar de manera inequívoca a la imagen inicial, acompañándose, en general, de cierta distorsión. En la segunda categoría, la compresión se lleva a cabo mediante una transformación que conserva la energía y que representa una imagen a partir de un arreglo donde el máximo de información esta empaquetado en un número mínimo de muestras. La tercera categoría es la de las técnicas híbridas los cuales combinan codificación por predicción y mediante transformación.

Las técnicas de compresión de imágenes también se pueden clasificar en dos grupos según exploten una redundancia espacial o temporal de los datos. En el primer caso, se reduce la correlación que existe entre las muestras de una misma imagen; en este caso se habla de compresión *intraframe*. El segundo grupo corresponde a una reducción de la correlación que existe entre las tramas sucesivas de una secuencia de imágenes; en este caso se habla de compresión *interframe*.

Entre las técnicas de compresión *intraframe*, se destaca la codificación por predicción donde se quita la redundancia mutual en muestras sucesivas de una imagen y se cuantifica únicamente la nueva información o la "innovación", es decir, para un pixel, la diferencia entre su valor real y la mejor predicción de este valor, en general obtenida por mínimos cuadrados. Dentro de estas técnicas, entran la técnicas de predicción tales como la modulación delta y el DPCM (*Differential Pulse Code Modulation*), que se introducirá en el párrafo III-A. El método de cuantificación de área constante (*Constant Area Quantization*), explota la propiedad visual del ojo humano que distingue más detalles en la regiones de mayor contraste de tal forma que se pueden transmitir imágenes con menos resolución en las zonas de menor contraste. En la técnica de codificación arborescente (*tree encoding*), el nivel de cuantificación de un pixel se basa en los siguientes pixeles en la tramas.

En lo que concierne las técnicas de compresión *interframe*, la redundancia ocurre en las zonas donde hay poco movimiento entre tramas. Este movimiento es el resultado del desplazamiento de objetos relativamente a un fondo, del movimiento de la cámara, y de las posibles operaciones realizadas con esta cámara tales como un barrido o un zoom. El conocimiento de este movimiento permite realizar una codificación

eficiente de la secuencia de imágenes, es decir, una reducción de la cantidad de información necesaria para la codificación. Se habla entonces de compensación de movimiento.

Este trabajo se estructura en dos partes. En una primera parte, se presenta de manera detallada algunas técnicas empleadas para la estimación del movimiento *interframe* en una secuencia de imágenes. En una segunda parte, se tratan las aplicaciones de la estimación de movimiento en la compresión de imágenes mediante el salto de tramas, la codificación híbrida y la transformación 3D acompañados de la compensación de movimiento.

II- ESTIMACIÓN DE MOVIMIENTO EN UNA SECUENCIA DE IMÁGENES

A) Introducción

Existe un amplio rango de aplicaciones que encuentran un interés particular en la interpretación y la descripción de movimientos a partir de una secuencia de imágenes. Aplicaciones orientadas al análisis son, por ejemplo, el reconocimiento y el seguido de blancos, el cómputo y la caracterización del movimiento humano, el análisis de imágenes científicas u obtenidas por satélite, etc. Por otra parte, la reducción de la banda de frecuencias, obtenida por compensación de movimiento, permite la compresión de imágenes para una codificación y una transmisión más eficientes.

El movimiento relativo entre objetos en una escena y una cámara conduce a un movimiento aparente de objetos en una secuencia de imágenes. En otros términos, una secuencia de imágenes digitales se obtiene por proyección de un espacio continuo 4D (espacio más tiempo) sobre un espacio discreto 3D. La estimación del movimiento a partir de esta secuencia puede llevarse a cabo siguiendo dos enfoques.

El primero corresponde a la extracción de formas 2D en la imagen, que corresponden a características geométricas de los objetos 3D en la escena, como ángulos, líneas y superficies de oclusión, líneas de demarcación entre

zonas de diferente reflectividad, etc., y eso, para cada imagen de la secuencia. Después, se establece una correspondencia entre las formas de las sucesivas tramas. Con la hipótesis de que los objetos moviéndose en la escena son cuerpos rígidos, es decir, que la distancias 3D entre sus formas son constantes, se llega a un sistema de ecuaciones no lineales que se pueden resolver a partir de los desplazamientos observados en la imagen 2D. Este enfoque concierne más bien las aplicaciones de análisis de secuencias de imágenes.

El otro enfoque, se basa en el cómputo del flujo óptico, o del campo 2D de movimiento. El flujo óptico corresponde a la variación espacio temporal de la intensidad, mientras que el campo de movimiento se relaciona con la proyección del movimiento en la escena 3D sobre el plano 2D de la imagen. En el caso ideal, el flujo óptico corresponde al campo de movimiento. En la práctica este hecho no se garantiza. Un objeto en movimiento puede dar lugar a un motivo de luminosidad constante, mientras que una escena inmóvil puede presentar variaciones en su luminosidad. En lo que concierne la codificación vídeo, las técnicas de estimación de movimiento estiman la trayectoria de un pixel a través de imágenes sucesivas para expresar la intensidad de una imagen a partir de la información previa lo que equivale a estimar el flujo óptico.

B) Estimación del movimiento basada en el flujo óptico

A continuación se presentan tres grupos de técnicas de estimación del movimiento : las técnicas de gradiente, que tienen aplicación en análisis de imágenes, las técnicas *pel-recursive* y las técnicas *block-matching*, que se utilizan en codificación de secuencias de imágenes. Un cuarto grupo se compone de las técnicas en el dominio de frecuencias de uso restringido.

1 -/ Técnicas del gradiente

El análisis de los cambios instantáneos, en los valores de luminosidad de una imagen, conduce a un mapa de velocidad denso llamado flujo de imagen o flujo óptico. Este análisis no requiere el establecimiento de correspondencias entre las imágenes sucesivas de la secuencia.

Sea $I(x,y,t)$ la intensidad en el punto de coordenadas (x,y) en el tiempo t. La hipótesis primordial consiste en decir que la intensidad o luminosidad de un pixel es constante a lo largo de su trayectoria, es decir, que

$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t) \tag{1}$$

donde Δx, Δy y Δt son pequeños.

Por aproximación del término de la derecha con una serie de Taylor se puede escribir

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + I_x\Delta x + I_y\Delta y + I_t\Delta t + \text{términos de mayor orden} \tag{2}$$

Ignorando los términos de mayor orden y haciendo $\Delta t \to 0$ se llega a

$$I_x u + I_y v + I_t = 0 \tag{3}$$

donde I_x, I_y e I_t son la derivadas parciales de I respecto a x, y y t que se estiman a partir de la imagen, y donde

$$u = \frac{dx}{dt} \quad \text{y} \quad v = \frac{dy}{dt} \tag{4}$$

son las componentes de las velocidad en el punto (x,y) en las direcciones x e y. La colección de vectores solución de esta ecuación constituye el flujo óptico de la imagen.

La ecuación (3) tiene dos incógnitas u y v. Por consiguiente no es suficiente para especificar el flujo óptico. Se tiene que incorporar una condición suplementaria.

Varias condiciones son posibles :

- que el flujo óptico sea suave y que los puntos que sean vecinos tengan velocidades similares.
- que el flujo óptico sea constante en segmentos enteros de la imagen.
- una restricción sobre el movimiento de flujo; que sea plano por ejemplo.

La condición de suavidad impone que el campo de movimiento varíe suavemente en la mayor parte de la imagen. Horn & Schunck [1] imponen esta condición por minimización del error en el flujo óptico que se expresa como

$$\begin{aligned} E(x,y) &= \text{error en (3)} \ + \lambda^2(\text{desviación por suavidad}), \\ E(x,y) &= \left(I_x u + I_y v + I_t\right)^2 + \lambda^2\left\{\left(u_x^2 + u_y^2\right) + \left(v_x^2 + v_y^2\right)\right\}, \end{aligned} \tag{5}$$

donde λ es una constante. El problema es entonces de encontrar u y v que minimicen el término R dado por

$$R = \iint \left\{\left(I_x u + I_y v + I_t\right)^2 + \lambda^2\left[\left(u_x^2 + u_y^2\right) + \left(v_x^2 + v_y^2\right)\right]\right\} \cdot dx \cdot dy \tag{6}$$

La ecuación (6) puede resolverse por un método de cálculo variacional. Derivando (6) respecto a u y a v, igualando $\partial R/\partial u$ y $\partial R/\partial v$ a cero, y escribiendo

$$\begin{aligned} \left(u_x^2 + u_y^2\right) &= u - u_{ave} \\ \left(v_x^2 + v_y^2\right) &= v - v_{ave} \end{aligned} \tag{7}$$

se llega a

$$u = u_{ave} - I_x \frac{P}{D}, \qquad v = v_{ave} - I_y \frac{P}{D} \tag{8}$$

donde

$$P = \left(I_x u_{ave} + I_y v_{ave} + I_t\right), \quad \text{y} \quad D = \lambda^2 + I_x^2 + I_y^2 \tag{9}$$

Las ecuaciones (8) pueden resolverse de manera iterativa, es decir, obteniendo $u(t)$ y $v(t)$ a partir de $u_{ave}(t-1)$ y $v_{ave}(t-1)$.

Horn y Schunck [1] han mostrado que el método converge cuando el flujo óptico es estático, es decir, cuando los vectores de velocidad no varían con el tiempo, por ejemplo, en el caso de una esfera girando alrededor de un eje estacionario. Cuando se viola esta condición, existen bordes donde no se conserva la suavidad local del flujo óptico. Una solución posible consiste entonces en detectar estos bordes y limitar el método a las regiones donde se mantiene la suavidad. Algunos métodos para determinar estos bordes aparecen en Schunck [2].

La aproximación de primer orden realizada en (2) no es satisfactoria para considerar los bordes y los ángulos en la imagen. Snyder et al. [3] han utilizado una aproximación de segundo orden, para obtener una ecuación no lineal de las variables u y v. Prazdny [4] ha resuelto el problema para el caso donde se considera una translación pura de la cámara asumiendo que se conoce la expansión de foco, es decir, la intersección entre el eje de translación de la cámara con el plano de la imagen en la parte positiva del eje.

Otra variante, introducida por Yachida [5], extiende el método iterativo de cómputo del flujo óptico, considerando la condición de suavidad no solo para pixeles vecinos en el espacio, sino para los pixeles correspondientes en las previas y siguientes tramas.

2-/ Técnicas *pel-recursive*

Las técnicas *pel-recursive* consisten en estimar el desplazamiento entre tramas para cada *pel* de una imagen, clasificándolos entre segmentos predicables y segmentos no predicables. El transmisor duplica el procedimiento de predicción del receptor y transmite el error cuando con no se predice con éxito el desplazamiento de un *pel*. También transmite las posiciones de los *pels* no predicables. Para los *pels* que pueden predecirse, no se mandan datos. El método llamado *"pel recursive motion compensation"*, desarrollado por Netravali y Robbins [6], actualiza la estimación del movimiento *pel* por *pel* de manera que la estimación converge hacia el verdadero desplazamiento. Una vez adquirido el desplazamiento de un *pel*, este se utiliza para la predicción de los valores de los *pels* siguientes.

Sea $I(z,n)$, la intensidad del *pel* de coordenadas $z=(x,y)$ en la trama n. La diferencia entre tramas (*Frame Difference* o *FD*) se define como

$$FD = I(z,n) - I(z,n-1) \tag{10}$$

Si se mueve un objeto, la estimación del desplazamiento entre tramas puede representarse por $\hat{D}$. Una diferencia entre tramas desplazadas (*Displaced Frame Difference* o *DFD*) puede definirse como

$$DFD\left(z,\hat{D}\right) = I(z,n) - I\left(z-\hat{D},n-1\right) \tag{11}$$

Si no hay error en la estimación $\hat{D}$, es decir, si $D=\hat{D}$, la *DFD* debería ser nula ya que los valores de los *pels* son idénticos. Entonces, un desplazamiento $\hat{D}$ debería hacer que la *DFD* tienda a cero.

Algoritmos para estimar $\hat{D}$ se pueden obtener a partir de una descomposición en series de Taylor, con necesaria aproximación.

La aproximación lineal de las series de Taylor, en el caso 2D permite escribir

$$I\left(z-\hat{D}^{i},n-1\right)=I\left(z-\hat{D}^{i-1},n-1\right)+\left(\hat{D}^{i-1}-\hat{D}^{i}\right)\cdot\vec{\nabla}I\left(z-\hat{D}^{i-1},n-1\right) \qquad (12)$$

donde $\vec{\nabla}I\left(z-\hat{D}^{i-1},n-1\right)$ es el gradiente de I en la posición $\vec{\nabla}I\left(z-\hat{D}^{i},n-1\right)$ y en la trama n. $\hat{D}^{i-1}$ y $\hat{D}^{i}$ son el antiguo y el nuevo estimado del vector de movimiento. Dado la serie de Taylor para los valores de *pel* de una trama, una nueva función puede obtenerse simplemente restando esta función de una constante que es el valor del siguiente *pel*. Se puede entonces utilizar el método de Newton para encontrar ceros. Cuando la DFD es nula, $D=\hat{D}$, y se ha localizado un *pel* de la previa trama de valor igual al presente *pel*. El procedimiento se repite entonces para el *pel* siguiente.

A partir de (10) y (12), se obtiene

$$DFD\left(z,\hat{D}^{i}\right)=DFD\left(z,\hat{D}^{i-1}\right)-\left(\hat{D}^{i-1}-\hat{D}^{i}\right)\cdot\vec{\nabla}I\left(z-\hat{D}^{i-1},n-1\right) \qquad (13)$$

Si $\hat{D}^{i}$ es exacto, entonces la DFD es nula. En práctica, la DFD no siempre llega a cero de tal forma que se define un umbral debajo del cual se considera nula.

Netravali y Robbins [6] resuelven la ecuación (13) con una técnica iterativa y numérica llamada "*steepest descent*", o de pendiente más abrupta. Esta técnica se basa en el hecho de que el vector opuesto del gradiente de una función apunta en la dirección donde la función disminuye más rápidamente. En cada iteración, se obtiene una nueva estimación $\hat{D}^{i}$ por la cual el gradiente puede ser diferente del anterior. Si $|DFD|$ es menor que el umbral, entonces se suspenden las iteraciones hasta encontrar un *pel* por el cual $|DFD|$ es mayor que el umbral.

Existen varios algoritmos que se basan en esta técnica para la estimación del desplazamiento.

El primer algoritmo busca el mínimo de $|DFD|^2$

$$\hat{D}^i = \hat{D}^{i-1} - \underbrace{\varepsilon \cdot DFD\left(z, \hat{D}^{i-1}\right) \cdot \vec{\nabla} I\left(z - \hat{D}^{i-1}, n-1\right)}_{\text{Término de corrección}} \tag{14}$$

donde ε es una constante de convergencia.

El segundo algoritmo es una versión simplificada se da con

$$\hat{D}^i = \hat{D}^{i-1} - \underbrace{\varepsilon \cdot signo\left\{DFD\left(z, \hat{D}^{i-1}\right)\right\} \cdot signo\left\{\vec{\nabla} I\left(z - \hat{D}^{i-1}, n-1\right)\right\}}_{\text{Término de corrección}} \tag{15}$$

Dado que la función signo sólo toma los valores $\{0, -1, 1\}$, los vectores de corrección solo puede tener ángulos múltiples de 45°. Netravali y Robbins [6] han mostrado que este algoritmo es efectivo. Sin embargo, $\hat{D}$ sólo se puede actualizar con un factor ε, en cada iteración de forma que se podría necesitar un número importante de iteraciones para llegar a la convergencia. Entonces, por razones de implementación de *hardware*, se limita el número de iteraciones a una por *pel*. A pesar de esta imposición, el método funciona ya que una vez obtenido un vector de movimiento correcto, el desplazamiento no varía mucho, excepto en los bordes, donde se requiere entonces un nuevo periodo de convergencia.

Walker y Rao [7] proponen un algoritmo mejorado que reconsidera las ecuaciones (14) y (15). Si se considera la intensidad en el borde de un objeto, las condiciones que requieren un vector de corrección grande se encuentran cuando $|DFD|$ es grande y $\left|\vec{\nabla} I\right|$ es pequeño. Y al contrario, si $|DFD|$ es pequeña y $\left|\vec{\nabla} I\right|$ grande, como podría ser el caso en el borde de un objeto, entonces el vector de corrección es pequeño. Para que funcione el algoritmo, ε debe elegirse de manera que se realice la convergencia en el caso de la corrección pequeña. Las correcciones mayores, se obtienen con muchas iteraciones. Este problema no se resuelve de manera satisfactoria con (13) y

(14) donde se puede ver que el factor de corrección varia inversamente a lo deseado. Una analogía al método de Newton conduce a una variante de (13) donde la constante de convergencia es una variable.

$$\hat{D}^i = \hat{D}^{i-1} - \underbrace{\varepsilon' \cdot DFD\left(z, \hat{D}^{i-1}\right) \cdot \vec{\nabla} I\left(z - \hat{D}^{i-1}, n-1\right)}_{\Delta D = \text{Término de corrección}} \tag{16}$$

donde

$$\varepsilon' = \frac{1}{2} \cdot \frac{1}{\left|\vec{\nabla} I\left(z - \hat{D}^{i-1}, n-1\right)\right|^2} \tag{17}$$

El algoritmo es entonces el siguiente :

si $|DFD| \leq$ umbral

entonces el término de corrección es nulo; $\Delta D = 0$

sino

si $\left|\vec{\nabla} I\right| \neq 0$

entonces se calcula el término de corrección ΔD con (15)

si $|\Delta D| < \frac{1}{16}$ **entonces** $\Delta D = \pm\frac{1}{16}$

sino

entonces el término de corrección es nulo; $\Delta D = 0$

si $|\Delta D| > 2$ **entonces** $\Delta D = \pm 2$

El resultado de este método es que el término de corrección disminuye a medida que aumenta $\left|\vec{\nabla} I\right|$ y viceversa.

3-/ Técnicas de *block-matching*

En estas técnicas, se aproxima el movimiento *interframe* por translación de una o varias áreas de una trama, relativamente a una trama

de referencia. Para esto, se tiene que realizar, primero, una segmentación de la imagen en áreas. Cafforio y Rocca [8] proponen una segmentación y medida del desplazamiento de un único objeto en movimiento, en un fondo estacionario. Sin embargo, este método se vuelve más complejo a medida que se incrementa el número de áreas que se mueven y que aumenta el tamaño de las imágenes. Por otra parte, la codificación de segmentos de bordes arbitrarios aumenta la complejidad y el tamaño del código.

Un método más simple consiste en dividir una imagen en pequeños bloques rectángulares, de tamaño fijo, y suponer que cada uno realiza una translación independiente. Si estos bloques son suficientemente pequeños, se pueden aproximar movimientos como la rotación de objetos grandes, o el zoom de la imagen, a partir de translaciones de estas áreas. este método evita la codificación relativa a la segmentación y sólo es necesario codificar el vector de desplazamiento de cada bloque.

Otro método para medir el desplazamiento entre dos imágenes consiste en calcular la función de correlación cruzada entre estas dos imágenes. La localización del pico de la función indica entonces el vector de desplazamiento. Esta función suele calcularse mediante FFT. Sin embargo, se ha mostrado que la precisión de la correlación es mediocre cuando el tamaño de bloque es pequeño y si los bloques no siguen translaciones puras.

Un método que ha mostrado buenos resultados en gran parte de los casos de estimación de movimiento entre tramas, consiste en encontrar la dirección del mínimo de distorsión (DMD). La imagen se divide en pequeñas áreas rectángulares, llamadas sub-bloques. Sea U un sub-bloque de tamaño $M \times N$ y U_R un sub-bloque de tamaño $(M+2p)\times(N+2p)$, donde p es el desplazamiento máximo permitido en todas las direcciones, y expresado en número de pixeles. Se define la función de distorsión media entre U y U_k como

$$D(i,j)=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} g\big(u(m,n)-u_k(m+i,n+j)\big), \quad -p \le i,j \le p \tag{18}$$

donde $g(x)$ es una función de distorsión positiva y creciente, $g(x)=x^2$ corresponde a $D(i,j)$ igual a la función de error cuadrático mínimo. La dirección de la distorsión mínima se da con (i,j) que minimiza $D(i,j)$.

La dificultad de este método es que requiere la evaluación de $D(i,j)$ para $(2p+1)^2$ direcciones, es decir que, para un movimiento de cinco pixeles, se llega a 121 direcciones. Una solución para superar esta dificultad se encuentra con el desarrollo siguiente :

Si se define

$$D_0(q,l)=\min_{i,j}\{D(i,j)\} \tag{19}$$

Entonces, para $m=i-q, \quad n=j-l$, las funciones

$$\begin{aligned} D_1(|m|,|n|) &= D(i,j)-D_0(q,l), \quad m\geq 0, \quad n\geq 0 \\ D_2(|m|,|n|) &= D(i,j)-D_0(q,l), \quad m\geq 0, \quad n\leq 0 \\ D_3(|m|,|n|) &= D(i,j)-D_0(q,l), \quad m\leq 0, \quad n\leq 0 \\ D_4(|m|,|n|) &= D(i,j)-D_0(q,l), \quad m\leq 0, \quad n\geq 0 \end{aligned} \tag{20}$$

son funciones creciente de ambos $|m|$ y $|n|$, es decir que, para $1\leq k\leq 4$,

$$D_k(|m|,|n|) < D_k(|m'|,|n'|),$$

$$\text{si} \quad |m|<|m'| \quad \text{y} \quad |n|\leq|n'| \quad \text{o} \quad |m|\leq|m'| \quad \text{y} \quad |n|<|n'| \tag{21}$$

Esto significa que la función de distorsión es monótona creciente a medida que uno se aleja de la DMD, según cualquier dirección en cada uno de los cuadrantes. En el caso de una función de distorsión $g(x)=x^2$ se satisface la condición si la función de covarianza de las imágenes es una función creciente del desplazamiento en cada uno de los cuadrantes, lo que es el caso general, al menos en la cercanía de varios pixeles.

Luego, mediante un método de búsqueda en 2D, se reduce sucesivamente el área de búsqueda. En cada paso, se buscan cinco posiciones que contienen el centro del área y los puntos medio entre este centro y los

cuatro vértices a lo largo de los ejes que pasan por el centro. Este procedimiento continua hasta que el plano de búsqueda se reduzca a 3 por 3. En el paso final, la DMD se calcula con las nueve posiciones que quedan.

Las técnicas de *block-matching* tienen algunos problemas, relativos al campo de movimiento cuando se trata de estimar el movimiento real en una escena, a la compensación de movimiento al nivel de los bordes en movimiento, y a la distorsión creada por el hecho de considerar que todos los pixeles de un bloque tiene el mismo vector de desplazamiento; se habla entonces de *block artifacts*.

Para obtener un campo de movimiento más fiable, existen algoritmos basados en una representación multiresolución [9], [10] que realizan una estimación burda pero robusta, del campo de movimiento, en las zonas de menor resolución que se va afinando iterativamente a medida que se aumenta la resolución.

El problema de los *block artifacts* se puede resolver mediante varias técnicas. Una consiste, consiste en especificar el campo de movimiento para un número reducido de puntos en las sucesivas tramas, y obtener el desplazamiento de los demás puntos por interpolación.

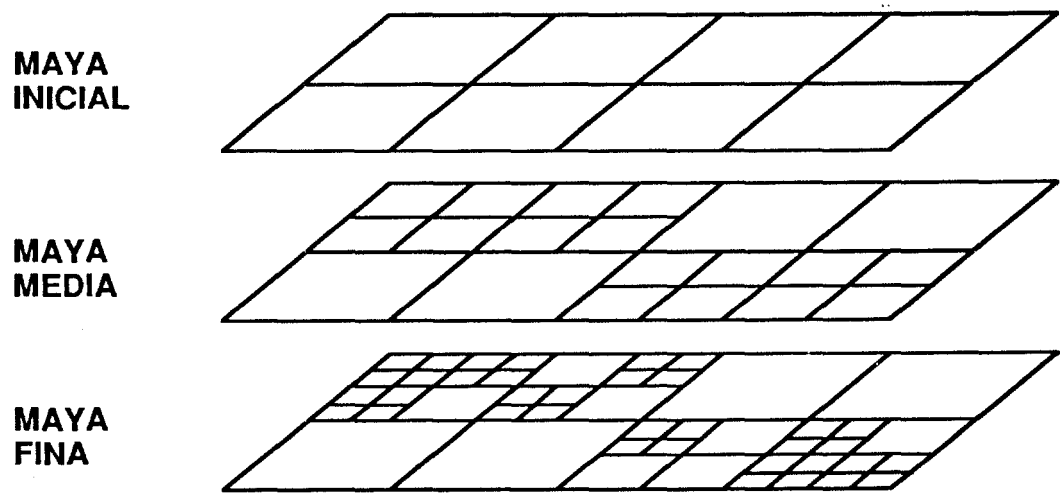

Figura nº 1: Estructura multimaya

Dufaux y Moscheni [11] han propuesto un método adaptativo, basado en una estructura multiníveles, construida a partir de mayas de tamaño

diferente destinadas a la estimación de movimientos en varias escalas. Esta estructura permite obtener un campo de movimiento suave, robusto y preciso que incluso reduce la complejidad del cómputo utilizado en las técnicas de *block-matching*. Esta técnica es localmente adaptativa y toma en cuenta el contenido de la escena. La figura nº 1 muestra un ejemplo de estructura multimaya donde la segmentación se lleva a cabo mediante una descomposición *quad-tree*. El algoritmo empieza por estimar el campo de movimiento de manera burda. Entonces, la maya inicial se subdivide solamente en las regiones donde la precisión de la estimación no resulta satisfactoria. Los vectores de movimiento se proyectan en la maya de mayor resolución. Este proceso se repite hasta obtener una precisión satisfactoria o llegar a un tamaño mínimo de bloque. Este algoritmo consta entonces de tres partes principales que se introducen a continuación.

La estimación del movimiento en cada nivel de la estructura, mediante una técnica de *block-matching* y una técnica de búsqueda en n pasos modificada.

La regla de segmentación que dicta la subdivisión de los bloques debe ser función de la precisión del vector de movimiento. Un criterio, de implementación simple, consiste en comparar el error absoluto medio o el error cuadrático medio con un umbral. Sin embargo, este criterio no garantiza que la ganancia obtenida en la DFD compense el coste en información aditiva necesaria para la codificación del movimiento. Por su lado el criterio de entropía contempla este problema y llega a un compromiso por minimización de la suma de los costes relativos a la codificación de la DFD, dado por su entropía y del movimiento.

Para realizar un mapeo del campo de movimiento entre dos mayas, durante el proceso de afinamiento, se necesita un operador de proyección. Este operador tendría que evitar la propagación a los niveles de mayor precisión, por una parte, de los *block-artifacts*, y por otra parte, de las estimaciones incorrectas de los vectores de movimiento. También tendría que garantizar un campo de movimiento suave y robusto. Para cumplir con estos requerimientos, el operador de proyección debería añadir una consistencia espacial del campo de movimiento. Si se considera los cuatro sub-bloques, fruto de la división de un bloque de nivel superior, cada uno necesita un vector de movimiento inicial. El método más simple consiste en duplicar cuatro veces el vector de nivel superior. Un operador más eficiente

consiste en seleccionar para cada sub-bloque, la mejor condición inicial entre los cuatro bloques más próximos, en la maya de nivel superior.

III-/ APLICACIÓN DE LA ESTIMACIÓN DEL MOVIMIENTO *INTERFRAME* A LA COMPRESIÓN

La estimación del movimiento *interframe* tiene aplicaciones en la compresión tanto para la transmisión como para el almacenamiento de imágenes. La utilización del conocimiento del movimiento o del desplazamiento de los pixeles en las tramas sucesivas de una secuencia de imágenes, para su codificación, se llama la "compensación de movimiento". Una vez calculada la trayectoria de los pixeles en una secuencia de imágenes, se pueden aplicar varias técnicas de codificación, mediante una adaptación de éstas a lo largo de las trayectorias. A continuación se presentan tres técnicas basadas en la compensación de movimiento.

A) Método del salto de trama.

Sin tener en cuenta la información relativa al movimiento de los pixeles, se puede realizar una compresión de los datos contenidos en una secuencia de imágenes a partir del método siguiente. El salto de trama o *frame-skipping*, consiste en alternativamente considerar una trama y saltar la siguiente. Sin conocimiento de la trayectoria de los pixeles, las imágenes "despreciadas" se pueden reproducir por repetición de la trama anterior, o por interpolación entre tramas anterior y posterior. En los dos casos, se obtienen efectos indeseables en la reproducción del movimiento. En el primer caso, se produce una pequeña vibración de la imagen mientras que la interpolación resulta en un emborronamiento de las áreas en movimiento.

Si u_{2k} representa un bloque de la trama de orden $2k$, cuando se han despreciado las tramas de orden $2, 4, ..., 2k$, etc., el valor reproducido de u_{2k}, notado u_{2k}^* se obtiene, sin compensación de movimiento, como viene a continuación.

En el caso de la repetición de tramas :

$$u_{2k}^{*}(m,n) = u_{2k-1}(m,n) \tag{22}$$

En el caso de la interpolación entre tramas :

$$u_{2k}^{*}(m,n) = \frac{1}{2}\{u_{2k-1}(m,n) + u_{2k+1}(m,n)\} \tag{23}$$

Los inconvenientes de la repetición de tramas o de la interpolación puede superarse mediante una predicción o una interpolación de los pixeles de las tramas despreciadas a lo largo de la trayectoria de su movimiento. Entonces, con la compensación de movimiento las ecuaciones (22) y (23) se reemplazan por :

$$u_{2k}^{*}(m,n) = u_{2k-1}(m+q, n+l) \tag{24}$$

y

$$u_{2k}^{*}(m,n) = \frac{1}{2}\{u_{2k-1}(m+q, n+l) + u_{2k+1}(m+q', n+l')\} \tag{25}$$

donde (q,l) y (q',l') son las coordenadas de los vectores de desplazamiento de u_{2k} relativamente a las tramas anterior y posterior, respectivamente.

La incorporación de la compensación de movimiento conduce a una mejoría del método de codificación. En [12], se obtienen resultados donde esta modificación conduce a un incremento de la relación señal a ruido de alrededor de 10dB.

B) Codificación híbrida con compensación de movimiento

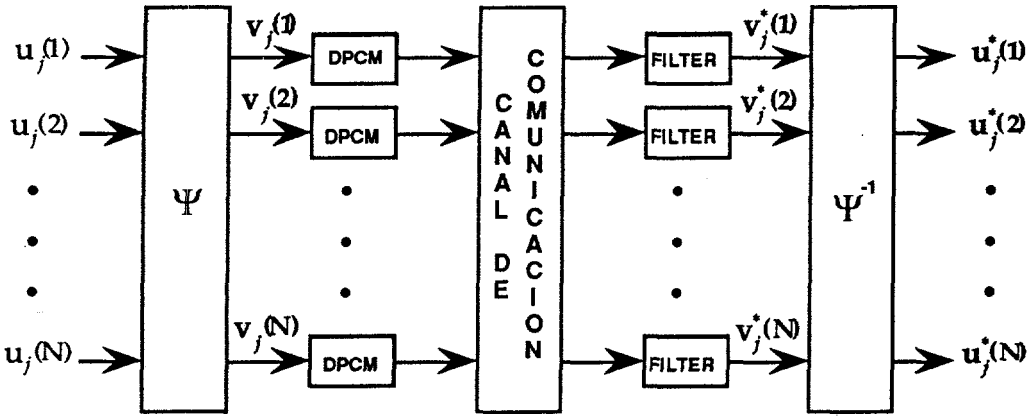

Figura nº 2 : Codificación híbrida

La figura nº 2 ilustra el principio de la codificación híbrida para una imagen de tamaño $M \times N$. Después de realizar una transformación unitaria de la imagen, en una de sus dimensiones espaciales, con el fin de quitar la correlación las muestras en esta dirección, los coeficientes de la transformada Ψ se codifican en la otra dimensión espacial, por una DPCM de primer orden.

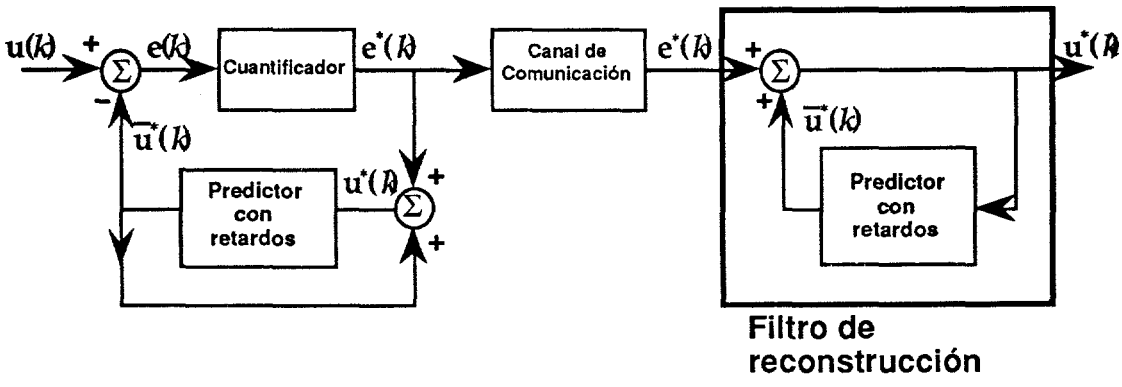

Figura nº3 : Differencial Pulse Code Modulation

La figura nº 3 muestra el principio de la DPCM, que se basa esencialmente en un predictor y un cuantificador. El aspecto importante de este esquema es que la predicción se basa en las muestras de salida. Esto resulta en que el predictor se encuentra en el bucle de realimentación de manera que el ruido de cuantificación en un paso, realimenta la entrada del cuantificador en el paso siguiente, lo que permite evitar una acumulación de errores.

Sea $\mathbf{u}_j$ el vector columna de índice j,

$$\mathbf{u}_j = \left[u(1,j), u(2,j), \ldots, u(N,j)\right]^T \tag{26}$$

La transformación unitaria $\mathbf{v}_j = \Psi \mathbf{u}_j$ se realiza en cada vector $\mathbf{u}_j$ de manera que los elementos de $\mathbf{v}_j$ no sean correlacionados. Además, para cada índice i, la secuencia $\{v_j(i)\}$, se modeliza por un proceso autoregresivo adecuado, como un modelo de Markov del primer orden.

$$v_j(i) = \alpha_i v_{j-1}(i) + e_j(i), \qquad 1 \le i \le N \tag{27}$$

Las ecuaciones de la DPCM, para el canal *i*, son en este caso :

$$\begin{array}{ll} \text{para el predictor} & \overline{v}_j^*(i) = \alpha_i v_{j-1}^*(i), \\ \text{para la entrada del cuantificador} & \tilde{e}_j(i) = v_j(i) - \overline{v}_j^*(i), \\ \text{para el filtro de reconstrucción} & v_j^*(i) = \overline{v}_j^*(i) + \tilde{e}_j^*(i) \end{array} \tag{28}$$

El esquema de codificación requiere, primero, calcular la transformada de cada vector columna $\mathbf{u}_j$. Luego, se realiza para los sucesivos vectores transformados $\mathbf{v}_j$, la codificación por predicción a través de los canales DPCM. Al nivel de la recepción, el receptor reconstruye los vectores transformados y realiza una transformación inversa.

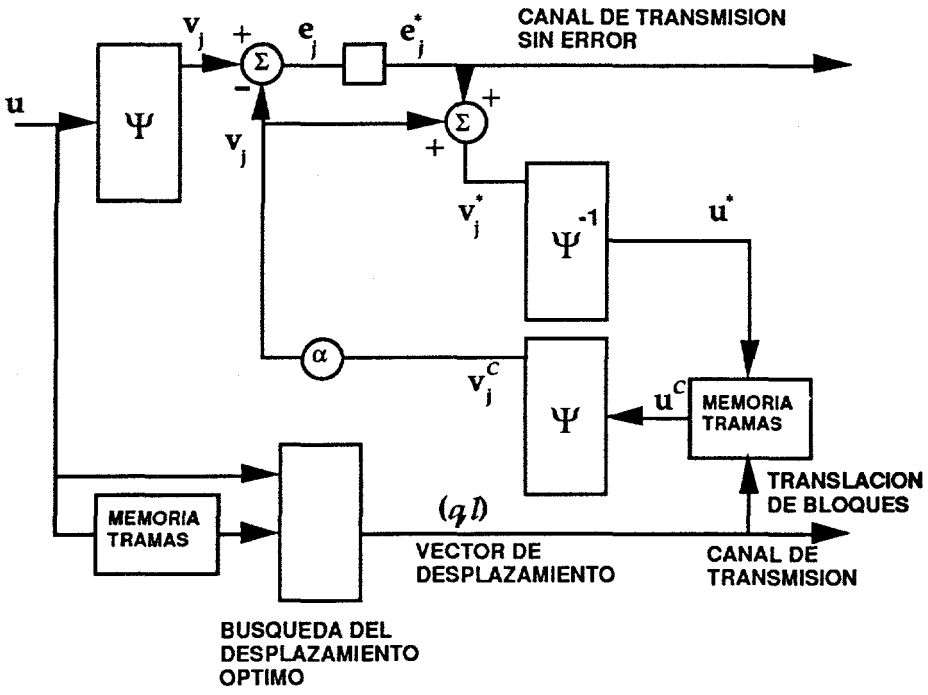

Figura nº 4 : Codificación híbrida con compensación de movimiento.

La figura nº 4, muestra el diagrama bloque de la codificación híbrida con compensación de movimiento. En esta variante, el predictor utiliza los pixeles que siguen la trayectoria del movimiento.

La codificación híbrida acompañada de compensación de movimiento es muy interesante y conduce a una mejor compresión que la obtenida, por ejemplo, a partir de las técnicas predictivas y mediante transformación. En [12], se evalúan las prestaciones de la codificación híbrida sin y con compensación de movimiento, en comparación con la técnica por transformación. Se observa que la ganancia entre técnica híbrida y transformación, que es del orden de 2.5dB (con una tasa de 2 bits/pixel), se aumenta de un factor 7dB al introducir la compensación de movimiento.

C) Codificación por transformación 3D y compensación de movimiento

La codificación *interframe* de secuencias vídeo corresponde al manejo de datos en 3D o más. En este contexto, se puede aplicar una codificación por transformación y llegar a una compresión de estos datos. Una

transformación, separable, en tres dimensiones de una secuencia $u(i,j,k)$ de tamaño $L \times M \times N$ se define como

$$v(l,m,n) = \sum_{i=1}^{L} \sum_{j=1}^{M} \sum_{k=1}^{N} u(i,j,k) a_L(l,i) a_M(m,j) a_N(n,k) \tag{29}$$

$$1 \le l \le L, \quad 1 \le m \le M, \quad 1 \le n \le N \tag{30}$$

donde $\{a_L(i,j)\}$ son los elementos de una matriz unitaria A_L de dimensiones $L \times L$.

El algoritmo de codificación es el mismo que el de la codificación por transformación, introducido previamente, excepto en que se trabaja con variables de tres índices. La integración de la compensación de movimiento consiste entonces en seleccionar bloques espaciales, que siguen la trayectoria del movimiento.

IV- CONCLUSIONES

En este trabajo, después de una introducción general sobre los métodos empleados para la compresión de imágenes, se han presentado algunas técnicas de estimación del movimiento *interframe*. Las técnicas del gradiente, conducen a un campo de movimiento denso y se orientan más al análisis de secuencias de imágenes. La utilización de las técnicas *pel-recursice* y *block-matching* permite la supresión de la redundancia temporal, presente en las secuencias de imágenes, y por consiguiente conducen a una reducción de la cantidad de información necesaria para su codificación.

A las técnicas *block-matching* han resultado más apropiadas en el campo de la codificación de primera generación, utilizadas en televisión digital. La definición del estándar MPEG (*Moving Picture Experts Group*) [13], un estándar para la compresión de señales digitales vídeo y audio, especifica el formato de representación de la información relativa al movimiento *interframe*, donde se asocian un o varios vectores a cada sub-bloques de una imagen. Aunque el estándar no especifique las técnicas que tienen que emplearse para el manejo de estos vectores, las técnicas *block-*

matching aparecen las más indicadas. Dufaux y Moscheni [11] proponen un variante de estas técnicas, basada en una estructura multimaya adaptativa, que evita sus desventajas, como el fenómeno de *block artifacts* o la mediocre compensación al nivel de los bordes.

El uso del conocimiento relativo al movimiento *interframe*, para una codificación efectiva de las secuencias vídeo se llama compensación de movimiento. En la segunda parte de este trabajo, se han presentado tres técnicas de compresión, el salto de tramas, la codificación híbrida y la codificación mediante transformación 3D, donde la incorporación de la compensación de movimiento conduce a un aumento de sus prestaciones.

El lector encontrará otros datos de interés relacionados con la compresión de secuencias de imágenes en las referencias [14] a [17].

BIBLIOGRAFÍA

[1] . Horn and B.G. Schunck, "Determining optical flow," *Artificial Intelligence*, Vol. 17, pp. 185-203, 1981.

[2] B.G. Schunck, "Image flow : Fundamentals and Algorithms," in *Motion Understanding : Robot and human vision*, N.N. Martin and J.K. Aggarwal, Eds Norwell, MA : Klumer Academic Publishers, 1988.

[3] W.E. Snyder, S.A. Rajala and G. Hirzinger, "Image modelling, the continuity assumption and tracking," in *Proc. Int. Conf. of Pattern Recognition*, pp. 1111-1114, 1980.

[4] V. Prazdny, "A simple method for recovering relative depth map in the case of a translating sensor," in *Proc. of Int. Joint Conf. on Artificial Intelligence*, pp. 698-699, 1981.

[5] M.Yachida, "Determining velocity map by 3-D iterative estimator," in *Proc. of Int. Joint Conf. on Artificial Intelligence*, pp. 716-718, 1981.

[6] A.N. Netravali and J.D. Robbins, "Motion compensated television coding: Part I," *Bell syst. Tech. J.*, Vol. 58, pp. 631-670, Mar. 1979.

[7] D.R. Walker and K.R. RAO, "Improved Pel-Recursive Motion Compensation", *IEEE Trans. on Comm.*, Vol. COM-32, No. 10, October 1984.

[8] C.-Cafforio and F. Rocca, "Method for measuring small displacements of television images", *IEEE Trans. Inform. Theory*, Vol. IT-22, pp. 573-579, Sept. 1976.

[9] P. Anandan, "A unified perspective on computational techniques for the measurement of visual motion," in *IEEE Proc. Int. Conf. Computer Vision*, pp. 219-230, London, UK, 1987.

[10] M.Bierling, "Displacement estimation by hierarchical block matching," in *SPIE Proc. Visual Commun. and Image Process. '88*, Cambridge, MA, Nov. 1988, Vol.1001, pp. 942-951.

[11] F.Dufaux and F.Moscheni, "Motion Estimation Techniques for Digital TV : A Review and a new contribution," *Proc. of IEEE*, Vol. 83, No. 6, pp. 858-875, June 1995.

[12] J.R. Jain and A.K. Jain, "Displacement Measurement and Its Application in Interframe Image Coding," *IEEE Trans. Comm.* Vol. COM-29, No. 12, pp. 1799-1809, December 1981.

[13] D. Le Gall, "MPEG : A video Compression Standard for Multimedia Applications," *Communications of the ACM* / April 1991/ Vol. 34, No. 4, pp. 47-58.

[14] A.K. Jain , "Image Data Compression : A Review," *Proc. IEEE*, Vol. 69, No. 3, pp. 349-389, March 1981.

[15] A.K. Jain, P.M. Farelle and V.R. Algazi, "Image Data Compression," in *Digital Image Processing Techniques*, Academic Press Inc.

[16] J.K. Aggarwal, N. Nandhakumar, "On the Computation of Motion from Sequences of Images - A Review," *Proc. IEEE*, Vol. 76, NO. 8, pp. 917-935, August 1988.

[17] M. Kunt, M. Benard, R. Leonard, "Recent Results in High-Compression Image Coding", *IEEE Trans. Circ. Sys*, Vol. CAS-34, No. 11, pp.1307-1336, November 1987.