

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

ESTIMACION Y COMPENSACION DE
MOVIMIENTO PARA LA CODIFICACION DE
SECUENCIAS DE IMAGENES DIGITALES

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN TELECOMUNICACIONES

P R E S E N T A N :
RODOLFO ARIAS VILLAVICENCIO
SAMUEL KOBELKOWSKY SOSA

DIRECTOR DE TESIS:
DR. FRANCISCO GARCIA UGALDE

MEXICO, D. F. 1996

TESIS CON
FALLA DE ORIGEN

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIAS

A esos grandes emprendedores que fundaron mi fe y fueron mi fortaleza, a ellos que cuidaron mis sueños y me brindaron su amor, a ellos, mis mejores maestros, a mis padres, Rodolfo y María.

Rodolfo

A mis padres Ishaac y Ma. de Jesús por haberme formado y a quienes gracias a su ejemplo y cariño he tenido un camino que seguir.

A mi hermana Rebeca por estar siempre conmigo.

A mis tíos, primos y toda mi familia.

A los amigos que siempre me han apoyado y que no menciono por que no quisiera omitir a ninguno.

Samuel

AGRADECIMIENTOS

Agradecemos al Dr. Francisco García Ugalde por el apoyo brindado para desarrollar esta investigación.

Al Dr. Gustavo Ayala Milián por darnos la oportunidad de hacer nuestro trabajo usando los recursos del Laboratorio de Cómputo Avanzado.

A todos los profesores de la primera generación de la carrera de Ingeniero en Telecomunicaciones, en especial al Ing. Jesús Reyes García por su constancia y esfuerzo en la enseñanza.

Al Dr. José Alberto Escobar y al M. en I. Daniel Gatica por sus invaluable opiniones a nuestro trabajo.

Al equipo de colaboradores del Laboratorio de Cómputo Avanzado de la División de Estudios de Postgrado de la Facultad de Ingeniería por sobrellevar nuestros procesos.

A nuestra *alma mater*, la Universidad Nacional Autónoma de México por todas las enseñanzas recibidas.

A *The University of Western Ontario* por toda la información que para nuestra investigación nos proporcionó.

Este trabajo fue desarrollado con apoyo de Fundación UNAM y del proyecto PAPIIT IN-501095 de DGAPA UNAM

Gracias

Resumen

La estimación y compensación de movimiento tiene como objetivo la reducción de ancho de banda para la transmisión y almacenamiento de señales de video digital por medio del aprovechamiento de la redundancia temporal y espacial de las imágenes que componen las secuencias de video. Existen dos grandes grupos para los métodos de estimación de movimiento, los métodos *pel recursivos* y los de *apareamiento de bloques*.

En el presente trabajo se analizan, desarrollan e implementan algoritmos de ambos grupos e incluso se propone un método que los combina. Del mismo modo, se hace un estudio de la codificación por medio de compensación de movimiento utilizando los algoritmos de estimación mas representativos como parte de dicho codificador. Posteriormente se evalúan los resultados de las simulaciones y se realiza un estudio comparativo a fin de emitir un juicio sobre el desempeño particular de cada algoritmo.

Los algoritmos de simulación están implementados bajo una filosofía orientada a objetos utilizando el lenguaje de programación C++ bajo ambiente UNIX. La biblioteca de funciones desarrollada puede servir como base para futuras aplicaciones por la gran facilidad de reutilización de código que brinda este lenguaje de programación.

Índice

1. Introducción

1.1 Justificación.....	1
1.2 Estructura del documento	2

2. Conceptos básicos

2.1 Filtrado digital.....	5
2.1.1 Introducción.....	5
2.1.2 Fundamentos.....	6
2.2 Procesamiento digital de imágenes en movimiento.....	10
2.2.1 Introducción.....	10
2.2.2 Conceptos básicos.....	11
2.3 Estructuras piramidales.....	14
2.3.1 Introducción.....	14
2.3.2 Fundamentos.....	15
2.3.3 Pirámides paso bajas: la pirámide Gaussiana.....	16
2.3.4 Pirámides paso banda: la pirámide Laplaciana.....	20

3. Estimación de movimiento

3.1 Introducción.....	23
3.2 Estimación de movimiento, caso monoresolución.....	23
3.2.1 Métodos por apareamiento de bloques.....	24

3.2.2 Métodos por transformadas..... 24

3.2.3 Métodos diferenciales..... 24

4. Técnicas pel-recursivas

4.1 Introducción..... 29

4.2 Métodos "steepest descent" 29

4.3 "Steepest descent", versión simplificada..... 31

4.4 "Steepest descent", usando múltiples pels..... 32

4.5 Modelos de luminancia variable 34

4.6 Aproximación de Walker y Rao..... 36

5. Técnicas por apareamiento de bloques

5.1 Introducción..... 37

5.2 Modelos de movimiento de bloques..... 37

5.2.1 *Movimiento traslacional*..... 37

5.2.2 *Deformación de bloques*..... 38

5.3 Método de correlación de fase 39

5.4 Método por apareamiento de bloques 40

5.4.1 *Funciones de máxima similitud*..... 41

5.4.2 *Estrategias de búsqueda*..... 42

5.4.2.1 *Búsqueda de los tres pasos* 43

5.4.2.2 *Búsqueda cruzada*..... 44

6. Estimación de movimiento, casos multiresolución y multirestricciones

6.1 Introducción..... 47

6.2 Métodos multirestricciones	47
6.2.1 <i>Búsqueda de funciones invariantes de movimiento</i>	48
6.2.2 <i>Aproximación multirestricción - multiresolución para la estimación de movimiento</i>	49
6.3 Tratamiento de los métodos multirestricciones	49
6.3.1 <i>Solución iterativa - recursiva</i>	50
6.4 Estimación de movimiento en multiresolución.....	50
6.4.1 <i>Estimación de movimiento sobre pirámides paso bajas</i>	51

7. Compensación de movimiento

7.1 Introducción.....	53
7.2 Algoritmo del codificador.....	53
7.3 Algoritmo del decodificador	55

8. Implantación y comparación de los algoritmos.....57

8.1 Introducción.....	57
8.1.2 <i>Criterios de evaluación</i>	57
8.2 Estructura general de los algoritmos pel-recursivos	60
8.2.1 <i>Cálculo del término de corrección para los diferentes algoritmos</i>	62
8.2.2 <i>Resultados de las simulaciones</i>	64
8.3 Estructura general de los algoritmos de apareamiento de bloques.....	74
8.3.1 <i>Criterios de máxima similitud</i>	74
8.3.2 <i>Método de los tres pasos</i>	74
8.3.3 <i>Método de búsqueda cruzada</i>	75

8.3.4 Resultados de las simulaciones.....	76
8.4 Métodos propuestos por los autores	85
8.4.1 Método mixto.....	85
8.4.2 Método plano.....	87
8.4.3 Resultados de las simulaciones.....	87
8.5 Estructura general para los algoritmos de compensación de movimiento	91
8.5.1 Codificador de Walker y Rao para compensación de movimiento.....	91
8.5.2 Resultados	93
8.5.3 Codificador de Walker y Rao para métodos de apareamiento de bloques.....	97
8.5.4 Resultados	98
8.5.5 Resultados para el codificador del método mixto	101
9. Conclusiones.....	109
10. Bibliografía.....	113

1. Introducción

1.1 Justificación

Diversas técnicas de estimación y compensación de movimiento son utilizadas para la codificación de señales de video digital con el objeto de minimizar su ancho de banda para su transmisión y almacenamiento en canales y dispositivos con limitación en ancho de banda.

Dicha minimización de ancho de banda es indispensable en las aplicaciones actuales de las telecomunicaciones puesto que algunos de los recursos de que se disponen como pueden ser el espectro electromagnético, son escasos y próximos a la saturación.

Una señal de video monocromática se compone de una secuencia de imágenes fuertemente correlacionadas entre sí, por lo que conllevan una redundancia temporal que es utilizada por los algoritmos que se estudian y proponen a lo largo del presente trabajo para reducir el número de símbolos requeridos para su representación.

Para lograr una buena compresión de las señales hay que tomar en cuenta un número de factores que incluyen la relación señal a ruido, la velocidad en que se realiza el proceso de codificación y decodificación y la calidad de la señal reconstruida. Es necesario recalcar que dicha calidad por lo general es inversamente proporcional a la tasa de compresión que se obtiene en el proceso y al tiempo en que se realiza, por lo que la elección de un método eficiente de codificación es muy importante.

Los métodos de compresión de video usados en el presente trabajo son llamados métodos de compensación de movimiento. En la compensación de movimiento intervienen dos procesos importantes: la estimación de movimiento y la codificación de la llamada señal de error resultante.

La estimación de movimiento consiste en obtener un campo vectorial que representa el movimiento aparente entre los elementos de dos imágenes consecutivas de la secuencia. Hay dos grupos de algoritmos de estimación de movimiento, los algoritmos *pel recursivos* y los *algoritmos de apareamiento de bloques*.

Los primeros se fundamentan en una o más funciones llamadas restricciones de movimiento que relacionan las derivadas espaciales y temporales de una señal con un campo de vectores de movimiento aparente. Los segundos dividen las imágenes consecutivas en pequeños bloques para buscar con la ayuda de funciones de máxima similitud un vector de desplazamiento que describa el movimiento de los elementos de la escena.

La codificación se realiza por medio de un lazo de tipo DPCM que utiliza un predictor y un estimador de movimiento para codificar el error de predicción, que es transmitido por el canal de comunicaciones. El decodificador, que es también de tipo DPCM, utiliza la señal del error codificada para reconstruir la secuencia original. Para un buen método de estimación, el error debe ser pequeño, con lo que se reduce la cantidad de información a transmitir.

El objetivo general de este trabajo es analizar los diferentes métodos de estimación y compensación de movimiento y mejorarlos para obtener una mayor tasa de compresión. El trabajo presenta un método que conjuga las dos técnicas de estimación y propone su uso en compensación de movimiento.

1.2 Estructura del documento

Para efectos de estudio y desarrollo, el presente trabajo se encuentra dividido en los siguientes capítulos:

Capítulo 1. Presenta una introducción general a los conceptos de video digital y su imperiosa necesidad de búsqueda de algoritmos adecuados para la minimización de ancho de banda, así como de los métodos estudiados y desarrollados a lo largo de este trabajo. También describe la estructura del presente trabajo.

Capítulo 2. Contiene una introducción a los conceptos generales del procesamiento digital de imágenes, de filtrado digital bidimensional y de estructuras piramidales en imágenes.

Capítulo 3. Introduce los diferentes métodos de estimación de movimiento comparando sus principales características. Expone además la base teórica para los métodos pel recursivos.

Capítulo 4. Basadas en el estudio realizado en el capítulo anterior, este capítulo presenta el desarrollo de diferentes técnicas pel recursivas entre los que se incluyen las propuestas por Netravali y Robbins y por Walker y Rao.

Capítulo 5. Describe los métodos de apareamiento de bloques y sus variantes en el dominio temporal y espectral.

Capítulo 6. Referidos a los conceptos desarrollados en los capítulos anteriores, se llega al uso de modelos más complejos para la estimación de movimiento como son el uso de estructuras piramidales y la utilización de más de una función invariante al movimiento.

Capítulo 7. Una vez estudiados los métodos de estimación se procede en este capítulo al análisis de la compensación de movimiento, desarrollando el esquema para la codificación y decodificación.

Capítulo 8. Presenta la implantación de los algoritmos seleccionados para la estimación y compensación de movimiento evaluando su desempeño y respuesta a la variación de parámetros.

Capítulo 9. En este capítulo se presentan las conclusiones finales del trabajo de tesis realizado.

2. Conceptos básicos

2.1 Filtrado digital

2.1.1 Introducción

El campo de procesamiento digital de señales bidimensionales está creciendo rápidamente. Ejemplos de señales bidimensionales son: fotografías satelitales, mapas de radar y sonar, placas de rayos X, radiografías, micrografías electrónicas, datos de registros magnéticos, sísmicos y gravitacionales.

Típicamente el proceso que se le da a las señales es con el objeto de extraer o resaltar alguna de sus características: ruido, detección de bordes, patrones, etc.

Una señal bidimensional (2-D) continua es una función de dos variables independientes continuas, como puede ser la intensidad luminosa, en el caso de una fotografía, como función de distancias en los ejes coordenados x , y .

Una señal bidimensional discreta es una versión muestreada de una señal bidimensional continua y se representa normalmente como un arreglo en forma matricial de números. También se puede representar por un espectro frecuencial que es modificable por medio de filtros en dos dimensiones.

Los filtros bidimensionales son, al igual que los filtros unidimensionales, sistemas discretos que pueden ser lineales o no lineales, invariantes o dependientes al desplazamiento, causales o no causales y estables o no estables. Pueden ser caracterizados por ecuaciones en diferencias o variables de estado en dos diferentes variables y por funciones o matrices de transferencia, que son funciones racionales de dos variables. Sin embargo, en los filtros bidimensionales, la variable tiempo que es usada en los filtros monodimensionales es reemplazada por la variable espacio (análisis en el dominio del espacio), y la representación frecuencial es conocida como frecuencia espacial.

Los filtros bidimensionales pueden ser recursivos y no recursivos, en cuanto si la salida del filtro depende o no de valores previos de ella. Alternativamente, el filtro puede clasificarse como filtro de respuesta al impulso infinita, *IIR* o de respuesta al impulso finita, *FIR*. Las propiedades de los filtros *FIR* e *IIR* son análogas a las propiedades de los filtros *FIR* e *IIR* unidimensionales. Por ejemplo, los filtros *FIR* pueden ser diseñados para tener fase lineal y los filtros *IIR* pueden ser diseñados para disminuir la dificultad computacional del proceso de filtrado. El

diseño de filtros, ya sea 2-D ó 1-D involucra varios pasos que son: aproximación, realización, implantación y estudio de efectos de cuantización.

Aproximación es el proceso de generar una función de transferencia estable a partir de la máscara deseada del filtro. La *realización* consiste en convertir la función de transferencia en una representación gráfica, diagrama de flujo o variables de estado. En la fase de *implantación* dicho diagrama, representación de variables de estado o red de filtrado digital es traducido en un programa de computadora. Al implantar los filtros digitales, se utilizan registros de longitud finita para representar los coeficientes, por lo que se introducen errores de cuantización que producen inexactitudes en la amplitud de la respuesta. De igual modo, al procesar las señales, los resultados de las multiplicaciones de coeficientes inherentes al proceso de filtrado tienen que volver a ser cuantizados, por lo que se introduce un error de cuantización en la señal, que tiene como resultado ruido en el filtro y oscilaciones parásitas que pueden llegar a causar inestabilidad.

Aunque la caracterización, propiedades y diseño de un filtro 2-D es similar a la de un filtro 1-D, en ocasiones hay excepciones. Por ejemplo, para analizar la estabilidad de un filtro 2-D, se deben tomar en cuenta las intersecciones de contornos en el plano complejo en que el denominador de la función de transferencia es cero, con los contornos en el plano complejo para los puntos en que el numerador es cero. Debido a que polinomios de dos variables no son generalmente factorizables en polinomios de primer orden y que el teorema fundamental del álgebra no es extensible a polinomios de dos variables, se hace de esto una tarea sumamente compleja.

Otro problema es el relacionado con la complejidad computacional del filtro, ya que con una dimensión añadida a los datos, la cantidad de cómputo requerida aumenta a aproximadamente el cuadrado del orden del filtro. Por esto, es extremadamente importante diseñarlos del menor orden posible.

Adicionalmente, al contrario que en señales audibles, los filtros diseñados para el procesamiento digital de imágenes se enfrentan al problema de que el ojo humano es sensible a las distorsiones de fase de las imágenes por lo que los filtros 2-D diseñados para procesamiento de imágenes tienen que prestar particular atención a la linealidad de la fase.

2.1.2 Fundamentos

La región de soporte de una señal discreta 2-D es la región conectada en el plano, en el que la señal es diferente de cero.

Una señal discreta 2-D es separable si se puede expresar como producto de dos señales 1-D:

$$x(n, m) = x_1(n)x_2(m) \quad \dots 2.1$$

Cuando la señal es separable, el problema del diseño del filtro se puede simplificar en el del diseño de dos filtros unidimensionales. Muchas señales 2-D no son separables, sin embargo, a veces es posible expresar una señal 2-D como una combinación lineal de señales 1-D:

$$x(n, m) = \sum_{i=N_L}^{N_H} \alpha_i x_{1i}(m)x_{2i}(n) \quad \dots 2.2$$

donde:

$$\alpha_i = \text{cte.}$$

En el caso de una señal 2-D de soporte finito, siempre se cumple dicha combinación.

Para el caso de filtros 2-D, la causalidad solo tiene significancia si las variables n o m representan al tiempo. Sin embargo, el concepto sigue siendo útil.

Un filtro 2-D no recursivo es aquel en el que la salida $y(m, n)$ es una función de todos los valores posibles en la entrada $x(m, n)$:

$$y(m, n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} a(i, j)x(m-i, n-j) \quad \dots 2.3$$

Para un filtro causal la señal de entrada es cero para valores negativos de i y j por lo que los límites inferiores de la sumatoria son cero. Si además el soporte del filtro es finito, la salida del filtro no recursivo se convierte en:

$$y(m, n) = \sum_{i=0}^M \sum_{j=0}^N a(i, j)x(m-i, n-j) \quad \dots 2.4$$

donde M y N son las dimensiones del soporte, y el filtro es llamado de orden (M, N) . Cuando los datos son registrados en una memoria antes de procesarlos, el filtro no necesita ser causal, pues el origen del plano de referencia puede ser desplazado arbitrariamente.

Un filtro digital 2-D recursivo difiere del no recursivo en que la salida es función de las entradas, así como de los valores en la salida.

Un filtro lineal, invariante al desplazamiento, causal y recursivo puede representarse en términos de la ecuación en diferencias:

$$y(m,n) = \sum_{i=0}^M \sum_{j=0}^N a(i,j)x(m-i,n-j) - \sum_{i=1}^M \sum_{j=1}^N b(i,j)y(m-i,n-j) \quad \dots 2.5$$

Los filtros con respuesta al impulso finita, FIR, son de soporte finito. Tienen la característica de ser de fase lineal si su respuesta al impulso es simétrica alrededor del cero:

$$h(n) = h(N-1-n) \text{ para } 0 \leq n \leq N-1 \quad \dots 2.6$$

Al expresar una entrada arbitraria en términos de una suma de impulsos, es posible expresar la salida de un filtro 2-D lineal, invariante al desplazamiento en términos de la respuesta al impulso. La señal de entrada puede representarse como:

$$x(m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} x(i,j)\delta(m-i,n-j) \quad \dots 2.7$$

La salida del filtro puede representarse como:

$$y(m,n) = \mathfrak{R}x(m,n) \quad \dots 2.8$$

y su respuesta al impulso como:

$$h(m,n) = \mathfrak{R}\delta(m,n) \quad \dots 2.9$$

Entonces si el filtro es lineal, invariante al desplazamiento:

$$y(m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(i,j)x(m-i,n-j) \quad \therefore \quad \dots 2.10$$

$$y(m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h(m-i,n-j)x(i,j) = h(m,n) \otimes x(m,n)$$

A esta relación se le llama suma de convolución y a los coeficientes $h(i, j)$ se les conoce también como máscara del filtro.

Un filtro digital 2-D es estable si:

$$\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} |h(i, j)| \leq K < \infty \quad \dots 2.11$$

Lo cual es una condición necesaria y suficiente.

Para realizar un filtro, se pueden seguir varias aproximaciones, sin embargo cuando se trabaja con coeficientes de precisión finita, algunas diferencias pueden surgir. La aproximación más directa para realizar un filtro consiste en convertir la ecuación de diferencias en la red del filtro digital sin modificarla.

La estructura paralela de un filtro se puede obtener si la respuesta al impulso se puede expresar como suma de dos funciones

$$h(n, m) = h_1(n, m) + h_2(n, m) \quad \dots 2.12$$

entonces:

$$y(n, m) = y_1(n, m) + y_2(n, m) \quad \dots 2.13$$

donde:

$$y_1(n, m) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_1(i, j) x(n-i, m-j)$$

$$y_2(n, m) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_2(i, j) x(n-i, m-j) \quad \dots 2.14$$

Esto quiere decir que la respuesta al impulso se puede obtener sumando la salida filtrada de la entrada a través de dos diferentes filtros. De igual modo, la estructura en cascada se da cuando se puede expresar la salida del filtro como:

$$h(n, m) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_2(i, j) h_1(n-i, m-j) = h_1 \otimes h_2 \quad \dots 2.15$$

Y la salida del filtro es:

$$y(n, m) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} h_2(i, j) \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x(i, j) h_1(n-k-i, m-l-j) \quad \dots 2.16$$

De aquí se desprende que la salida del filtro puede ser equivalente a la salida de un filtro tomando la salida de otro filtro. De particular importancia es el caso en que la respuesta al impulso de un filtro es separable:

$$h(n, m) = h_1(n)h_2(m) \quad \dots 2.17$$

Entonces la señal de salida es:

$$y(n, m) = \sum_{i=-\infty}^{\infty} h_1(i) \sum_{j=-\infty}^{\infty} h_2(j)x(n-i, m-j) \quad \dots 2.18$$

Esto es, la señal de salida es equivalente a las respuestas de dos filtros digitales unidimensionales, cada uno en una dirección del plano de soporte. Esta relación es muy importante por que permite diseñar los filtros 2-D, diseñando filtros 1-D y conectándolos posteriormente en cascada.

El análisis y diseño de los filtros por medio de métodos de transformadas puede ser simplificado enormemente puesto que la ecuación en diferencias puede ser convertida en una ecuación algebraica más simple de manipular, como en el caso de la transformada z bidimensional. Utilizando la transformada 2-D de Fourier se pueden establecer relaciones matemáticas entre señales discretas y continuas que hacen posible procesar digitalmente señales continuas.

2.2 Procesamiento digital de imágenes en movimiento

2.2.1 Introducción

Como se mencionó en la sección anterior, una imagen es una función bidimensional $I(x, y) = I(p)$ cuya amplitud para cada punto $p=(x, y)$ denota la intensidad de la luz reflejada por los objetos sobre los que incide.

Con el objeto de representar una imagen en un medio de comunicaciones, se realizan varias funciones básicas: la adquisición, el almacenamiento, el procesamiento y el despliegue. En la fase de adquisición se convierte la señal continua en su equivalente digital y discreto. Para imágenes en movimiento, los sistemas de adquisición generan secuencias de imágenes a una velocidad generalmente mayor a 10 cuadros por segundo.

Estas imágenes adquiridas deben ser almacenadas en memorias lo suficientemente rápidas para su procesamiento. Debido a la gran cantidad de puntos (muestras) de que se compone cada imagen, llamados *pixeles* o *pels* que pueden llegar a ser algunos cientos de miles, la velocidad de transferencia a la memoria y *buses* de datos es un factor crítico.

Debido a los tremendos requerimientos en memoria, velocidad de procesamiento y de transmisión, los procesadores de imágenes requieren así mismo de una gran velocidad. Es por esto que los algoritmos de procesamiento tienden a diseñarse de la más baja complejidad computacional posible en procesadores especializados, como son los procesadores digitales de señales y utilizando muy frecuentemente arquitecturas paralelas de diversas topologías contándose entre los más populares, los arreglos matriciales de procesadores, los procesadores en cascada (tipo *pipeline*), arreglos sistólicos y los procesadores vectoriales. También se emplean procesadores ópticos.

2.2.2 Conceptos básicos

Teorema de muestreo

Establece que una señal de banda limitada con componentes frecuenciales no mayores a W Hz, puede ser completamente recuperada si se conocen sus muestras tomadas a una tasa mayor a $2W$ por segundo. A la tasa de muestreo de $2W$ se le conoce como tasa de muestreo de Nyquist.

Aliasing

Cuando la frecuencia de muestreo es menor a la de Nyquist, aparecen réplicas de la señal desplazadas en frecuencia que se traslapan unas con otras, ocasionando distorsión de la señal. Con el fin de tener control sobre el ancho de banda de una señal, antes de muestrear la señal, generalmente se le aplica un filtro paso bajas para excluir las frecuencias mayores a W .

Cuantización

Las señales continuas tienen un rango infinito de valores. Sin embargo los sentidos humanos sólo captan diferencias de intensidad finitas [HAY86], por lo que la señal continua puede aproximarse por medio de una señal construida por amplitudes discretas. Si los niveles de amplitud discretos se hacen suficientemente cercanos unos a otros, la señal discreta se hará indistinguible de la continua. Al proceso de convertir una señal analógica continua en su forma discreta, digital, es llamado *cuantización*. El error de cuantización, también conocido como ruido de cuantización, es la diferencia entre la señal cuantizada y la original. El *cuanto* o

paso de cuantización, es la distancia entre dos niveles sucesivos de intensidad de la señal discreta. Esta no necesariamente tiene que ser uniforme.

Entropía

Es una medida cuantitativa de la información contenida en una señal. Si se asume que los símbolos, estadísticamente independientes, emitidos por una fuente de alfabeto finito la definen completamente, $S = \{s_1, s_2, \dots, s_K\}$, con probabilidades $P(s_1), P(s_2), \dots, P(s_K)$, al recibir un símbolo s_k , se dice que se han recibido

$$I(s_k) = \log \frac{1}{P(s_k)} = -\log P(s_k) \quad \dots 2.20$$

unidades de información. Si la base del logaritmo es dos, las unidades de información se representan por *bits*.

La información promedio de una señal por símbolo de información es entonces,

$$H(S) = \sum_{k=1}^K P(s_k) I(s_k) = -\sum_{k=1}^K P(s_k) \log P(s_k) \quad \dots 2.21$$

llamada entropía de la señal.

Error cuadrático medio.

Es una medida de la disimilitud de una imagen con otra. Esta comparación se realiza muestra por muestra:

$$EQM = \frac{1}{K} \sum_{k=1}^K [I_1(k) - I_2(k)]^2 \quad \dots 2.22$$

Relación señal a ruido

La relación señal a ruido relaciona el error cuadrático medio con los niveles máximos y mínimos de la imagen con el objeto de dar una medida más objetiva que sirva de punto de comparación para señales de naturaleza diferente:

$$SNR_{db} = 10 \log \frac{(I_{MAX} - I_{MIN})^2}{EQM}$$

...2.23

Aunque la relación señal a ruido da una medida de la similitud de una imagen con otro, lo que es muy útil al momento de evaluar la calidad de las imágenes reconstruidas, no es necesariamente una medida precisa de la calidad con la que el sistema de visión humana evalúa dicha imagen, para lo cual se necesitan escalas subjetivas derivadas de métodos estadísticos e investigación de la fisiología humana, en lo que se conoce como codificación perceptiva.

Ruido

Es una señal de naturaleza aleatoria indeseable que se añade a la señal de información cuando esta es procesada o transmitida.

El desempeño de un sistema de codificación es influenciado por dos fuentes típicas; El ruido de transmisión que puede ser introducido en cualquier punto entre el transmisor y el receptor y el ruido de cuantización introducido al momento de la cuantización.

Codificación

Es el proceso de convertir una señal discreta a una forma en que sea más apropiada para transmitirse por los medios de comunicación.

Codificación perceptiva

La codificación perceptiva se inspira en el sistema de percepción humana con el fin de codificar únicamente los aspectos de la señal que son relevantes para el sentido de la vista. Fundamentalmente, el estudio de la sensibilidad a la frecuencia permite codificar, con mayor densidad de muestras aquellas bandas frecuenciales relevantes al ojo. Otros modelos se basan en la sensibilidad al contraste, al cambio de fase e inmunidad al ruido.

Codificación predictiva (DPCM)

Cuando una señal de voz o video es muestreada a una frecuencia un poco mayor a la de Nyquist, la señal resultante suele tener una correlación muy grande entre muestras adyacentes, lo que significa que en promedio, la señal no cambia muy rápidamente de una muestra a la otra, con lo que resulta que la varianza entre muestras adyacentes es menor a la varianza de la señal, así que la señal contiene información redundante, por lo que no todos los símbolos no son absolutamente necesarios para la transmisión; eliminando esta redundancia, se obtiene una señal más eficientemente codificada.

Si se conoce una parte de la señal $m(k)$, como puede ser su comportamiento pasado, es posible estimar una aproximación del resto de ella, $\hat{m}(k)$, donde la entrada al cuantizador es una señal $e(k)=m(k)-\hat{m}(k)$ llamada error de predicción. El valor reconstruido consiste en la versión cuantizada de la señal.

En la ausencia de ruido en el canal de transmisión, la señal reconstruida difiere únicamente de la original en el ruido de cuantización.

Codificación de longitud variable

Debido a que los codificadores de longitud fija codifican más bits que los requeridos por la cantidad de información contenida en la señal, se recurre a los codificadores de longitud variable, entre los que se encuentran los *run length*, *Shannon Fano* y *Huffman*. Generalmente estos codificadores transmiten a una tasa promedio ligeramente mayor o igual a la entropía de la señal. Esto se logra asignando a cada símbolo una secuencia de bits, única y completamente decodificable, de longitud variable, dependiendo esta longitud de la probabilidad de cada símbolo dentro del flujo de datos.

Codificación por transformada

Consiste en dividir la imagen en bloques a los que se les aplica una transformada de tipo convolucional en la que los coeficientes resultantes están más descorrelacionados, codificándose con estos nuevos valores.

Codificación piramidal

Subdivide a la imagen en bandas espectrales contiguas, eliminando la correlación entre pels adyacentes. Primero se aplica un filtrado de tipo paso bajas a la imagen original. La señal de error resultante, de restar la imagen original con su versión paso bajas, es codificada y transmitida junto a la versión paso bajas. Debido a que la versión paso bajas se puede submuestrear y a que la señal de error, llamada versión paso banda tiene una entropía reducida debido a su rango dinámico menor, este proceso se puede repetir utilizando como punto de partida la versión paso bajas.

2.3 Estructuras Piramidales

2.3.1 Introducción

Una aproximación del procesamiento digital de imágenes en los últimos años, ha sido el considerar las características en las que el sistema de visión humano

presta más importancia: color, contornos, ruido, movimiento, etc. Conociendo estos aspectos, es posible desarrollar un sistema de análisis, mejoramiento y codificación que extraiga y preserve la información más relevante al ojo, a expensas de la menos relevante.

La investigación llevada a cabo por C.R. Carlson, E.H. y R.W. Cohen y por A.P. Ginsburg¹ indica que el sistema de visión humano separa la información en bandas de frecuencias espaciales contiguas que se traslapan, del mismo modo como lo hace un analizador de espectros. Hay evidencia que el número de subbandas oscila aproximadamente entre 5 y 7.

Cada banda del sistema de visión humana responde aproximadamente a una octava, esto es, la frecuencia central de cada banda difiere de su predecesora por un factor de dos. Al parecer cada banda es procesada independientemente de las demás. Incluso dentro de estas, la información es procesada en pequeñas subregiones espaciales; la información de frecuencias espaciales en un punto dado en una imagen es determinada usando áreas de la imagen muy localizadas en vez del campo total de la imagen.

Si se pretende hacer un procesamiento de imágenes inspirado en el que realiza el sistema de visión humano, se necesita descomponer la imagen en bandas pasantes y el algoritmo usado no necesariamente debe incluir el campo entero de la imagen. Para calcular la información se puede usar filtrado dentro de una vecindad. Algunas aplicaciones en las que se puede aplicar el procesamiento piramidal incluyen la reducción de ruido, reducción de entropía y análisis de movimiento, que es el tópico que se desarrollará en el presente trabajo.

2.3.2 Fundamentos

La pirámide es una técnica de procesamiento de señales que separa una imagen dada en imágenes de espectro de frecuencias espaciales contiguas. Estas señales son imágenes que contienen una octava de información que se descompone en componentes ortogonales, en las direcciones x e y . Las imágenes que representan la información de más baja frecuencia contienen un número reducido de muestras en relación con las bandas de frecuencia central, debido al teorema de muestreo. El conjunto entero de imágenes contiene la representación completa de la imagen original como lo hace el sistema de visión humana.

¹C.R. Carlson, E.H. y R.W. Cohen, "A simple Psychophysical Model for Predicting the Visibility of Displayed Information", *proc SID*, 21(3) p. 229, 1980

A.P. Ginsburg, "Specifying Relevant Spatial Frequency Information For Image Evaluation and Display Design: An Explanation of How We Can See Certain Objects" *Proc. SID*, 21 (3), p. 219 (1980).

Al eliminar muestras en las bandas superiores se obtiene un cálculo más rápido y eficiente. Es importante asegurar que el proceso de descomposición sea completamente reversible, seleccionando una tasa de submuestreo adecuada.

La estructura piramidal es usada en las técnicas de estimación y compensación de imágenes en movimiento para calcular el movimiento aparente de los pels en los niveles altos de la pirámide, propagando estos a los niveles más bajos, donde se encuentran los detalles más finos del movimiento.

2.3.3 Pirámides paso bajas: La pirámide Gaussiana

Al filtrar pasobajas la imagen g_0 para obtener la imagen g_1 , se dice que la imagen g_1 es la versión reducida de g_0 puesto que su resolución es menor. Debido al teorema de muestreo, la densidad de muestras de la imagen g_1 puede ser reducida. El filtrado se realiza por medio de la convolución bidimensional de la imagen con una función de peso simétrica, $w(n)=w(-n)$, y de soporte finito. Puesto que una de estas funciones de peso recuerda a la función de densidad de probabilidad Gaussiana, al conjunto de imágenes g_0, g_1, \dots, g_n se le llama pirámide Gaussiana, donde g_{k+1} es la versión reducida de g_k .

Generación de la pirámide Gaussiana

A partir de la imagen g_k , se puede obtener la imagen reducida g_{k+1} aplicando la convolución, o promedio pesado de la función de peso que es la respuesta al impulso de un filtro paso bajas. El tamaño de la función de peso no es crítico y generalmente se usa una de dimensiones 5×5 , ya que provee un buen filtrado a bajo costo computacional.

La función *REDUCE* puede ser enunciada de la siguiente manera, para el caso en que el filtro pasobajas es de soporte 5 por 5 para cada pel $g_k(i, j)$ de la imagen g_k :

$$g_k(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) g_{k-1}(2i+m, 2j+n) \quad \dots 2.24$$

El núcleo generador

La misma función pesada generadora w es usada para cada nivel de la pirámide. A esa función también se lo conoce como núcleo generador.

El núcleo generador puede hacerse separable de tal manera que:

$$w(m,n) = \hat{w}(m)\hat{w}(n) \quad \dots 2.25$$

Donde la función $\hat{w}(i)$ es unidimensional, simétrica y normalizada.

$$\sum_{m=-2}^2 \hat{w}(m) = 1$$

$$\hat{w}(l) = \hat{w}(-l) \quad \dots 2.26$$

Por último, todos los pels en cada nivel de la pirámide deben contribuir con el mismo peso total a los pels del siguiente nivel. Para el núcleo de longitud 5, esto se cumple con:

$$\begin{aligned} \hat{w}(0) &= a \\ \hat{w}(-1) &= \hat{w}(1) = 1/4 \\ \hat{w}(-2) &= \hat{w}(2) = 1/4 - a/2 \end{aligned} \quad \dots 2.27$$

Funciones de peso equivalentes

La generación iterativa piramidal es equivalente a convolucionar la imagen g_0 con una serie de funciones de peso equivalentes h_l tales que:

$$\begin{aligned} g_l(i, j) &= \sum_{m=-M_l}^{M_l} \sum_{n=-M_l}^{M_l} h_l(m,n) g_0(i2^l + m, j2^l + n) \\ M_l &= 2 M_{l-1} \end{aligned} \quad \dots 2.28$$

La forma sin embargo, depende del núcleo generador de la función, particularmente cuando $a=0,4$ en el núcleo generador, la figura de las funciones de peso es de forma de una función de distribución Gaussiana.

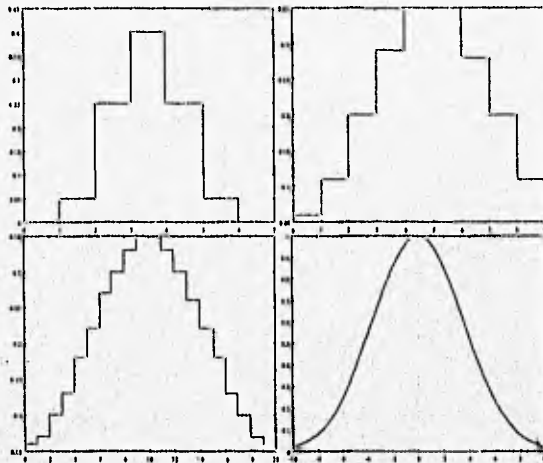


Figura 2.1 Funciones de peso equivalentes para los niveles 1, 2, 3 e infinito de la pirámide Gaussiana, con $a=0.4$

El efecto de convolucionar la imagen con la función de peso equivalente es la de filtrar paso bajas la imagen, reduciendo la banda de paso a una octava de nivel a nivel. Sin embargo el método piramidal también reduce la frecuencia de muestreo por el mismo factor, resultando en algoritmos más rápidos.

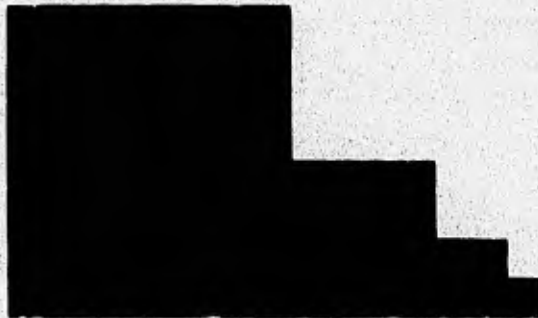


Fig. 2.2 Primeros cuatro niveles de una pirámide Gaussiana, $a=0.4$

Interpolación de la imagen Gaussiana

La función *EXPANDE* es el inverso de la función *REDUCE*. Produce a partir de una imagen g_l de dimensiones $M+1$ por $N+1$ una imagen $g_{l,1}$ de dimensiones $2M+1$ por $2N+1$ interpolando nuevos valores entre los ya existentes, que es del mismo tamaño que la imagen $g_{l,1}$.

Expandiendo n veces la imagen g_l se obtiene $g_{l,n}$.

$$g_{l,n} = \text{EXPANDE}(g_{l,n-1}) = \sum_{m=-2n+2}^2 \sum_{n=-2}^2 w(m,n) \cdot g_{l,n-1} \left(\frac{i-1}{2}, \frac{j-n}{2} \right) \quad \dots 2.29$$

Donde únicamente se incluyen en la suma los términos $(i-m)/2$ y $(j-n)/2$ cuando son enteros.

Si se aplica la función *EXPANDE*, l veces a la imagen g_l , se obtiene la imagen $g_{l,l}$ que es del tamaño de la imagen original g_0 .



Fig. 2.3 Primeros cuatro niveles de una pirámide Gaussiana, $a=0.4$ con las imágenes expandidas por medio de interpolación Gaussiana

2.3.4 Pirámides paso banda: La pirámide Laplaciana

Para obtener una compresión de la imagen se codifica únicamente el error resultante de restar la imagen expandida g_j de la imagen original g_0 . Repitiendo el proceso con las imágenes g_k se genera la pirámide Laplaciana.

La pirámide Laplaciana es la secuencia de señales de error L_0, L_1, \dots, L_N . Cada imagen es la diferencia entre dos niveles de la pirámide Gaussiana.

$$L_i = g_i - g_{i+1} \quad \dots 2.30$$

Para el último nivel $L_N = g_N$.

Funciones de peso equivalentes.

La generación iterativa piramidal Laplaciana es equivalente a la diferencia de las convoluciones de la imagen g_0 con dos funciones de peso equivalentes h_i y h_{i+1} que de nuevo, tienen un costo computacional mayor.

La pirámide Laplaciana se puede ver como un conjunto de imágenes paso banda de la imagen original. La escala del operador Laplaciano se dobla y la frecuencia central es reducida una octava de nivel a nivel.

A veces es conveniente interpolar las imágenes por medio de la interpolación Gaussiana. Así la expansión de la imagen $L_{i,n}$ es el resultado de expandir L_i n veces y $L_{i,1}$ es del tamaño de la imagen original.

Se puede observar que la pirámide Laplaciana sobresalta los contornos; más finamente en los niveles bajos que en los altos.



Primeros cuatro niveles de la pirámide Laplaciana de la imagen Lena.

Decodificación

La imagen original se puede obtener a partir de la pirámide Laplaciana por medio de la expansión y suma de todos los niveles de ella.

$$g_0 = \sum_{l=0}^N L_{\tau,l} \quad \dots 2.31$$

O bien se puede expandir L_l una vez y sumar el resultado con L_{l-1} hasta que se alcance g_0 .

Entropía

Si se asume que los valores de los píxeles son estadísticamente independientes, entonces el mínimo número de bits por pel para representar la imagen está dado por la entropía de la distribución de intensidades de la imagen y este mínimo se puede aproximar usando técnicas de codificación de longitud variable.

La técnica de sustraer un valor predicho de cada pel de la imagen como en la pirámide Laplaciana, remueve mucha de la correlación entre pels que también resulta en una concentración de los valores alrededor del cero con una reducción de la entropía y de la varianza. El grado de reducción de la entropía depende del valor de n elegido para generar la pirámide.

Cuantización

La entropía también puede ser reducida cuantizando los valores de los pels en cada nivel de la pirámide Laplaciana, que aunque introduce errores de cuantización la degradación puede no ser notoria para los observadores. Usando cuantización lineal, los valores para cada pel cuantizado $C_l(i,j)$ de una imagen de la pirámide $L_l(i,j)$ son

$$C_l(i,j) = mn \quad \text{si} \quad (m - 1/2)n < L_l(i,j) \leq (m + 1/2)n \quad \dots 2.32$$

Donde m y n son enteros, m representa el nivel de cuantización más cercano a $C_l(i,j)$ y n es el tamaño del intervalo entre niveles.

Para la reconstrucción de la imagen se usa la expansión del conjunto de imágenes C_l en lugar del conjunto L_l . El tamaño del intervalo n entre niveles se escoge: cuando el error de cuantización es apenas perceptible y cuando la imagen es observada a 5 veces la distancia de su ancho, lo que significa una separación de pels de aproximadamente 3 minutos de arco. Debe hacerse notar que n es menor a mayor nivel en la pirámide (menores frecuencias espaciales). Esto significa que el

sistema de visión humano es más sensible a perturbaciones en el contraste a altas frecuencias espaciales que a bajas frecuencias espaciales por lo tanto se deben usar más niveles de cuantización en los niveles altos de la pirámide; afortunadamente la densidad de muestras en estos niveles es menor que en los niveles bajos donde un paso de cuantización mayor puede ser usado.

Transmisión progresiva.

La pirámide Laplaciana es susceptible de ser transmitida progresivamente, esto significa que una versión burda de la imagen se transmite primero, para después ser transmitidos los detalles de resoluciones más finas. Con esto se permite terminar la transmisión de la imagen tan pronto como se reconoce que no es de interés, con el consecuente ahorro de tiempo de transmisión.

Conclusión

Si se escogen apropiadamente los parámetros de cuantización y codificación, la pirámide Laplaciana puede reducir sensiblemente la entropía de las imágenes sin exceder los límites de distorsión impuestos por el sistema de visión humana.

La generación de las pirámides Gaussianas y Laplacianas requieren de muy poca capacidad computacional, que además son susceptibles de implantarse con una estructura paralela e iterativa por lo que pueden ser generadas en tiempo real con procesadores en "pipeline" o arreglos sistólicos. Además uno puede acceder a copias casi paso banda o paso bajas de la imagen, para tareas subsiguientes de procesamiento de imágenes.

3. Estimación de movimiento

3.1 Introducción

La existencia de objetos relativamente inmóviles y de fondos invariantes entre las escenas de una secuencia de imágenes digitales, dan la idea de la existencia de información temporal redundante. Bajo este enfoque la reducción o eliminación de esta información, es la clave para minimizar el ancho de banda en una transmisión de señales de TV o vídeo digital. Así, la información a transmitir sería solo aquella que concierne a los objetos que están en movimiento.

En este estudio, el proceso de estimación de movimiento pretenderá la obtención de un campo de vectores de desplazamiento aparente, a través de las variaciones espaciales y temporales de la intensidad luminosa, con el fin de utilizarlos posteriormente para la extracción de la información no redundante.

Dos casos serán tratados, la estimación de movimiento aplicado al caso monoresolución, y la estimación de movimiento aplicado al caso multiresolución.

El tratamiento monoresolución opera directamente sobre las imágenes de la secuencia mientras que el tratamiento multiresolución opera de manera jerárquica sobre varios niveles de resolución de las imágenes de la secuencia a través de una estructura piramidal.

3.2 Estimación de movimiento, caso monoresolución

Los métodos que abordan la estimación de movimiento están clasificados según tres aproximaciones[BAA91]:

- Métodos por apareamiento de bloques
- Métodos por transformadas
- Métodos diferenciales

De estos, los métodos por apareamiento de bloques y los diferenciales se caracterizan por su gran simplicidad, un grado de complejidad factible para su implantación y la posibilidad de asegurar una buena calidad de reconstrucción.

Se dará en seguida un bosquejo general de estos métodos, siendo por su facilidad de implantación, los métodos diferenciales y por apareamiento de bloques, el objeto de estudio del presente trabajo.

3.2.1 Métodos por apareamiento de bloques

En los métodos por apareamiento de bloques, la imagen se divide en un número finito de pequeños bloques en los que se supone un movimiento puramente traslacional y a cada uno de ellos se le asigna un vector de desplazamiento obtenido con una función de similitud o correlación al comparar cada bloque dentro de una vecindad de la escena siguiente en la secuencia con el objeto de encontrar la nueva posición que ocupa. Es claro que es muy difícil encontrar un bloque exactamente igual al de referencia, por lo que se elige al que guarde un mayor parecido. Esta técnica es de gran utilidad cuando los movimientos son de gran amplitud, sin embargo el método en sí impone un alto número de cálculos, los cuales pueden ser reducidos por medio de algoritmos de búsqueda de caminos óptimos o bien por la aplicación de procesamiento paralelo y mediante la restricción del área de búsqueda.

3.2.2 Métodos por transformadas

Estos métodos trabajan en el dominio de la transformada, la cual puede ser: de Fourier, Coseno, Wavelet o descomposiciones de Gabor. Se basan en la observación de los efectos de un movimiento bidimensional sobre sus características o componentes frecuenciales y/o escalares. Ya que una translación espacial puede ser traducida como un desfaseamiento frecuencial, las medidas de variación en la fase de una imagen a otra en una secuencia, permiten la medida de una translación espacial. La aplicación de estos métodos requiere de procesos de transformación y comparación muy complicados, por lo que se presentan menos atractivos.

3.2.3 Métodos diferenciales

Los métodos diferenciales se basan en premisas de restricción de movimiento y relacionan las variaciones espaciales y temporales de una señal con un campo de vectores de movimiento aparente. Muchos métodos se derivan de la ecuación de restricción de movimiento, explicada a continuación, cuyo origen es la hipótesis de la invarianza de la luminancia.

La hipótesis de la invarianza de la luminancia supone que la luminancia de un punto en una imagen no cambia con el movimiento del punto y que toda modificación temporal de la distribución de intensidades luminosas de una secuencia es debida únicamente al desplazamiento de los pels que la componen. Es decir, que para todo pel p de coordenadas (x, y) y con una intensidad luminosa $I(p, t)$, que ha realizado un desplazamiento $d = (dx, dy)^T$ del instante $t-dt$ al instante t , podemos escribir:

$$I(x, y, t) = I(x - dx, y - dy, t - dt)$$

o bien,

$$I(p, t) = I(p - d, t - dt) \quad \dots 3.1$$

Una medida del cambio de luminancia de un pel que se ha desplazado de una escena a otra, es llamada función de diferencia de escena desplazada, o *DFD*², definida como:

$$DFD(p, d) = I(p, t) - I(p - d, t - dt) \quad \dots 3.2$$

De acuerdo a la hipótesis de la invarianza de la luminancia, si se calcula bien d , se puede llegar a:

$$DFD(p, d) = 0 \quad \dots 3.3$$

Los métodos de estimación de movimiento, estiman un valor de d llamado \hat{d} ; si no hay error en la estimación, $d = \hat{d}$ y la diferencia de escena desplazada estimada, representada como $DFD(p, \hat{d})$ es igual a cero. Una buena estimación de \hat{d} es la que hace que la *DFD* se aproxime a cero.

La ecuación 3.1 genera una ecuación de restricción de movimiento *ECM*³, relacionando así las variaciones espaciales de la luminancia con sus variaciones temporales.

Suponiendo que la función de intensidad es continua y derivable, se puede considerar su desarrollo en serie de Taylor en una aproximación de primer orden como:

²Del inglés Displaced Frame Difference

³Del francés Equation de Constraint de Mouvement

$$I(p-d, t-dt) = I(p, t) - \frac{\partial I(p, t) dx}{\partial x} - \frac{\partial I(p, t) dy}{\partial y} - \frac{\partial I(p, t) dt}{\partial t} + O \quad \dots 3.4$$

donde O contiene los términos de orden superior a 1.

Si tomamos en cuenta la ecuación 3.1 y si despreciamos los términos de orden superior a uno, esta ecuación de gradientes constituye el punto de partida de varios métodos de estimación de movimiento generadores de un campo de vectores $d=(dx, dy)^T$ de tipo *denso*, llamado así por que a cada pel le corresponde un vector.

Sustituyendo la definición de la *DFD* en la ecuación anterior y despreciando los términos de orden mayor a uno:

$$DFD(p, d) = \frac{\partial I(p, t) dx}{\partial x} + \frac{\partial I(p, t) dy}{\partial y} + \frac{\partial I(p, t) dt}{\partial t} \quad \dots 3.5$$

Esta ecuación se puede analizar de dos maneras diferentes:

Si se toma el límite de dt cuando se aproxima a cero y se divide toda la ecuación entre dt , haciendo que $DFD=0$:

$$\frac{\partial I(p, t) dx}{\partial x dt} + \frac{\partial I(p, t) dy}{\partial y dt} + \frac{\partial I(p, t)}{\partial t} = 0 \quad \dots 3.6$$

$$\frac{\partial I(p, t)}{\partial x} u_x(p, t) + \frac{\partial I(p, t)}{\partial y} u_y(p, t) + \frac{\partial I(p, t)}{\partial t} = 0$$

donde $u=[u_x(p, t), u_y(p, t)]^T$ denota el vector de velocidades del punto p al instante t . Esto significa que la estimación de velocidades y de desplazamiento son equivalentes cuando dt tiende a cero, por lo que muchas veces se intercambia d por U . ($dt=1$, $d=U \cdot dt$). Así llegamos a la ecuación de restricción de movimiento usual, ECM:

$$\nabla I \cdot U = -I_t \quad \dots 3.7$$

donde $U = (U_x, U_y)^T = (dx/dt, dy/dt)^T$ es el vector de movimiento aparente, $\nabla I = (I_x, I_y)$ es el gradiente espacial de la intensidad e I_t es el gradiente temporal.

U está definido como la tasa temporal de cambio en las coordenadas del plano imagen, en un punto particular $(p, t) \in \mathbb{R}^3$, como una determinación de las

variaciones espacio-temporales de la intensidad luminosa de la imagen I . Esto es, U representa un vector de velocidad instantánea para ese punto, también conocido como vector de velocidad aparente. En la práctica, se define un campo de vectores de desplazamiento basados en variaciones observables de la intensidad luminosa en imágenes 2-D sobre un campo plano-temporal A^3 . El campo de vectores es también conocido como campo de desplazamiento aparente.

Aunque la ECM es una restricción lineal de un espacio (U_x, U_y) , que relaciona los componentes de movimiento aparente, es insuficiente para estimar estos. Esto es por que solamente la proyección de U sobre el gradiente de la luminosidad puede ser determinada, es decir su componente paralela al gradiente de la intensidad. Esto es por que existen dos variables a estimar (U_x, U_y) , y solo una ecuación para calcularlas (ecuación 3.5).

Así, la determinación de los componentes de movimiento está sub-determinada, en el sentido de que la solución no es única, de aquí la necesidad de añadir restricciones adicionales. Estas pueden ser añadidas de dos maneras:

Explícitamente, mediante la formulación de restricciones sobre la naturaleza del movimiento o criterios de alisamiento del campo. Otros métodos proceden a la búsqueda de funciones invariantes con el tiempo, los cuales serán tratados en el capítulo referente a métodos multirestricciones.

Implícitamente, mediante la búsqueda de un campo de movimiento que satisfaga mejor la ECM, a partir de la información de un punto vecino apropiado. La explotación de tal información puede ser hecha mediante la adopción de aproximaciones por bloques donde el movimiento es constante en un bloque de la imagen o bien por medio de la adopción de métodos de resolución recursivos o iterativos. Este tipo de aproximación es muy utilizado en la codificación para compensación de movimiento.

En el segundo panorama, cuando dt es finita, se puede estimar el vector de desplazamiento por medio de varias maneras:

- Buscando un valor de d que haga la DFD cercana a cero dentro de un bloque de pels, en una estrategia de apareamiento de bloques.
- Calculando \hat{d} por medio de la ecuación 3.5 utilizando una estrategia de optimización basada en gradiente, con técnicas pel-recursivas.
- Resolver la ecuación 3.5 utilizando una serie de ecuaciones lineales obtenidas de un bloque de pels cuando $DFD=0$.

4. Técnicas pel-recursivas

4.1 Introducción

Las técnicas pel-recursivas adoptan en general la ECM para encontrar una solución recursiva de estimación de un campo denso de vectores de desplazamiento mediante la minimización del valor absoluto de la DFD o de la DFD^2 .

En 1975, Limb y Murphy propusieron un algoritmo para medir la velocidad de un objeto en movimiento sobre un fondo fijo. Después Cafforio y Rocca dieron los fundamentos teóricos para la estimación de pequeñas cantidades de movimiento traslacional e introdujeron un algoritmo de segmentación, el cual trata con pels relacionados con diferentes movimientos.

Basados en los anteriores trabajos el primer método pel-recursivo para la estimación de movimiento aparente fue introducido por Netravali y Robbins en 1979. La idea de base es un criterio de minimización de un error de predicción con la DFD^2 en el punto considerado.

4.2 Métodos "Steepest Descent"

Para un pel que tiene un desplazamiento d a partir de la imagen $t-dt$ a la imagen t , su DFD está dada por:

$$DFD(p, t) = I(p, t) - I(p - d, t - dt) \quad \dots 4.1$$

Dada la hipótesis de invarianza de la luminancia, se debe encontrar el valor estimado de d , \hat{d} tal que satisfaga:

$$DFD(p, \hat{d}) = 0 \quad \dots 4.2$$

es decir, el valor d que hace que la DFD sea nula.

Sin embargo, rara vez se alcanza esta condición debido a errores de cuantización, a que los objetos se ocultan detrás de otros, errores de interpolación y variaciones en la iluminación de la escena [MUR95] por lo que se consideran aproximaciones aceptables aquellas que hagan que el valor absoluto de la DFD esté por debajo de un valor umbral ($SDFD$ ó T) previamente establecido.

La solución propuesta por Netravali y Robbins, consiste en minimizar la DFD^2 , mediante un método numérico iterativo de gradiente directo llamado *steepest descent*. Este método clásico de minimización de funciones consiste en buscar el mínimo de una función en la dirección opuesta al gradiente:

Para encontrar un mínimo de una función $f(u_1, \dots, u_n)$ se calculan sus derivadas parciales con respecto a cada variable y se igualan a cero, entonces se resuelven las ecuaciones simultáneas resultantes:

$$\begin{aligned} \frac{\partial f(u)}{\partial u_1} &= 0 \\ \vdots & \\ \frac{\partial f(u)}{\partial u_n} &= 0 \end{aligned} \quad \dots 4.3$$

que también se pueden representar como:

$$\nabla_u f(u) = 0 \quad \dots 4.4$$

Como la DFD^2 no se puede representar en forma cerrada, se recurren a métodos iterativos que resuelvan esta ecuación. El más sencillo es el llamado *steepest descent* y consiste en actualizar el estimado de la posición del mínimo de la función en la dirección contraria al gradiente, puesto que el gradiente apunta hacia la dirección del máximo de la función, así la solución iterativa para encontrar el mínimo u de una función $f(u)$ es:

$$u^i = u^{i-1} - \alpha \nabla_u f(u)_{u^{i-1}} \quad \dots 4.5$$

donde α es una constante positiva que define el tamaño del paso de actualización. Si es demasiado grande, el algoritmo puede ser inestable, si es demasiado pequeño, tomará muchas iteraciones encontrar el mínimo de la función.

En los métodos por recursivos, la estimación de d se traduce en:

$$d^i = d^{i-1} - \frac{\epsilon}{2} \nabla_u DFD^2(p, d^{i-1}) \quad \dots 4.6$$

donde ϵ es una constante positiva y diferente de cero.

La recurrencia de esta ecuación permite desplazarse llevada de un pel $i-1$ a su vecino i según la dirección de barrido de las líneas de televisión.

Resolviendo el gradiente de la ecuación 4.6

$$d^i = d^{i-1} - \epsilon DFD(p, d^{i-1}) \nabla I(p - d^{i-1}, t - 1) \quad \dots 4.7$$

puesto que:

$$\begin{aligned} \nabla_d DFD(p, d) &= \nabla_d [I(p, t) - I(p - d, t - dt)] \\ &= \nabla_d I(p - d, t - dt) \end{aligned} \quad \dots 4.8$$

4.3 Steepest Descent, versión simplificada

El anterior estimador de desplazamiento requiere una multiplicación en cada iteración, lo cual es indeseable en la realización física, así Netravali y Robbins llegaron a la siguiente simplificación:

$$d^i = d^{i-1} - \epsilon \operatorname{sgn}[DFD(p, d^{i-1})] \operatorname{sgn}[\nabla I(p - d^{i-1}, t - 1)] \quad \dots 4.9$$

donde

$$\operatorname{sgn}(z) = \begin{cases} -1 & \text{si } z < -T \\ 0 & \text{si } |z| \leq T \\ 1 & \text{otro caso} \end{cases} \quad \dots 4.10$$

El anterior algoritmo para actualizar a d^i es llevado a cabo solamente en áreas de movimiento de la escena actual, por ejemplo en todas aquellas regiones donde

$$\sum_{i=0}^t |I(p_{x,y}, t) - I(p_{x,y}, t-1)| \geq T \quad \dots 4.11$$

Así los vectores de corrección tienen únicamente ángulos múltiplos de 45°. La desventaja de este método es que d es únicamente actualizado por el valor de ϵ , si es un valor pequeño, se necesitarán muchas iteraciones para lograr un desplazamiento. Sin embargo, una vez que se obtiene un vector inicial para un pel, es muy común que este no varíe drásticamente de pel a pel, con lo que el algoritmo es eficiente.

4.4 Steepest Descent, usando múltiples pels

El algoritmo anterior puede ser extendido considerando las diferencias de escena desplazada de muchos pels para estimar d . Existen varios algoritmos, en el primero analizado, d puede ser actualizado de muestra a muestra, usando el algoritmo *steepest descent* para minimizar una suma ponderada de la DFD² de algunos pels vecinos previos. Así:

$$d^l = d^{l-1} - \frac{\epsilon}{2} \nabla_d \left[\sum_{j=0}^l W_j \left[DFD(p_{l-j}, d^{l-1}) \right]^2 \right] \quad \dots 4.12$$

donde

$$W_j \geq 0 \text{ y } \sum_{j=0}^l W_j = 1 \quad \dots 4.13$$

La ecuación 4.12 puede ser expandida como:

$$d^l = d^{l-1} - \epsilon \left\{ \sum_{j=0}^l W_j DFD(p_{l-j}, d^{l-1}) \nabla [I(p_{l-j} - d^{l-1}, l-1)] \right\} \quad \dots 4.14$$

Esta puede generalizarse aún más usando diferentes funciones de error (por ejemplo DFD) en vez de una función cuadrática. La diferencia con las ecuaciones 4.7 y 4.12 es que, para estimar el nuevo valor de d , la DFD es evaluada en la vecindad de un pel. Esto tiene un efecto de suavización en el término de actualización.

El segundo algoritmo analizado aquí que utiliza múltiples pels analizado aquí, es un estimador de mínimos cuadrados basado en la intensidad de la escena previa, en la dirección en que lo hace la estimación anterior de desplazamiento. Así, debido a que:

$$DFD(p, \hat{d}) = I(p, t) - I(p - \hat{d}, t - 1) \quad \dots 4.15$$

y

$$I(p, t) = I(p - d, t - 1)$$

la ecuación 4.1 puede ser reescrita como:

$$DFD(p, \hat{d}) = I(p - d, t - 1) - I(p - \hat{d}, t - 1) \quad \dots 4.16$$

La expansión en serie de Taylor de primer orden de la función de intensidad luminosa es:

$$I(p - d, t - 1) = I(p - \hat{d}, t - 1) + (-d + \hat{d})^T \nabla I(p - \hat{d}, t - 1) + O \quad \dots 4.17$$

Donde O representa los términos de orden superior. Sustituyendo la ecuación 4.17 en la ecuación 4.16:

$$DFD(p, \hat{d}') = -(d - \hat{d}')^T \nabla I(p - \hat{d}', t - 1) + O \quad \dots 4.18$$

La cantidad $d - \hat{d}'$ puede ser estimada mediante técnicas estándar de mínimos cuadrados lineales, tomando varias muestras en el entorno de p , y tratando los términos de orden mayor a 1 como ruido, se genera un algoritmo *steepest descent* de la forma [WAN92]:

$$\begin{aligned} d^i &= d^{i-1} - \varepsilon \left[\sum_{j=0}^k \nabla I(p_{x-j}, d^{i-1}, t-1) \nabla I^T(p_{x-j}, d^{i-1}, t-1) \right]^{-1} \\ &\quad * \left[\sum_{j=0}^k DFD(p_{x-j}, d^{i-1}) \nabla I(p_{x-j}, d^{i-1}, t-1) \right] \quad \dots 4.19 \\ &= d^{i-1} - \varepsilon M_i^{-1} \psi_i \end{aligned}$$

La matriz inversa de la ecuación anterior puede ser aproximada [ROB83] por:

$$M_i = \frac{1}{\Delta_i} \left\{ \begin{array}{cc} \sum_{i=0}^n LDIF_j^2 & -\sum_{i=0}^n LDIF_j EDIF_j \\ -\sum_{i=0}^n LDIF_j EDIF_j & \sum_{i=0}^n EDIF_j^2 \end{array} \right\} \quad \dots 4.20$$

con

$$\Delta_i = \sum_{i=0}^n EDIF_j^2 \sum_{i=0}^n LDIF_j^2 - \left\{ \sum_{i=0}^n EDIF_j LDIF_j \right\}^2 \quad \dots 4.21$$

donde

$EDIF_j$ = Diferencia de elemento en $(p_{x,j}, d_{i,j})$ en la escena anterior (aproximación de la componente horizontal de ∇I).

$LDIF_j$ = Diferencia de línea en $(p_{x,j}, d_{i,j})$ en la escena anterior (aproximación vertical de la componente de ∇I).

En la evaluación de la matriz inversa, se debe verificar que la matriz sea no singular. La singularidad puede ser resultado de un número insuficiente de pels vecinos; por otro lado, de un gran número de ellos resultaría un vector de desplazamiento no satisfactorio debido a un efecto de suavización. Para evitar esto, cuando M_i sea singular, es decir, cuando su determinante sea nulo, no se realiza la actualización.

El último término de la ecuación 4.19 es dado por:

$$\psi_i = \left\{ \begin{array}{c} \sum_{i=0}^n DFD(p_{i-1} - d^{i-1}, t-1) EDIF_j \\ \sum_{i=0}^n DFD(p_{i-1} - d^{i-1}, t-1) LDIF_j \end{array} \right\} \quad \dots 4.22$$

4.5 Modelos de luminancia variable

Debido a factores como cambio en la iluminación de la imagen, rotación de los objetos y otros, el modelo de invariabilidad de la luminancia no es completamente válido por lo que se puede añadir un factor de ganancia que describe

la variación temporal de la luminancia, por lo que el modelo de la ecuación 4.1 puede ser generalizado como:

$$\begin{aligned}
 a) \quad I(p,t) &= \rho_1 I(p,t-1) \\
 y \\
 b) \quad I(p,t) &= \rho_2 I(p-d,t-1)
 \end{aligned}
 \qquad \dots 4.23$$

Las ecuaciones anteriores muestran dos modelos alternativos que incluyen la variación de la luminancia entre escenas, por lo que se debe estimar un parámetro adicional, ρ_1 ó ρ_2 .

El algoritmo para la compensación de la primera ecuación es [ROB83]:

$$\rho_1^i = \epsilon_1 [I(p,t) - \rho_1^{i-1} I(p,t-1)] I(p,t-1) \qquad \dots 4.24$$

Para la estimación en la segunda ecuación, ρ_2 y d deben ser estimados. Esto se logra por:

$$\begin{aligned}
 \rho_2^i &= \rho_2^{i-1} + \epsilon_1 DFD(p, \rho_2^{i-1}, d^{i-1}) I(p-d^{i-1}, t-1) \\
 d^i &= d^{i-1} - \epsilon_2 \rho_2^{i-1} DFD(p, \rho_2^{i-1}, d^{i-1}) \nabla I(p-d^{i-1}, t-1)
 \end{aligned}
 \qquad \dots 4.25$$

donde la DFD está definida para estos casos como:

$$DFD(p, \rho_2, d) = I(p,t) - \rho_2 I(p-d,t-1) \qquad \dots 4.26$$

Las iteraciones pueden proceder muestra por muestra a lo largo de una línea si ρ_1 , ρ_2 y d varían suficientemente lento, lo cual en la práctica ocurre frecuentemente.

Este es un modelo apropiado para la variación de intensidad hecha por movimientos traslacionales de objetos. Para una iluminación uniforme, ρ_2 será unitario y d será igual al desplazamiento. La ecuación 4.23.a también provee descripción de objetos en movimiento. A causa de esto, ρ_1 en algunos casos aproxima variaciones de intensidad hechas por objetos en translación. Esto ocurre si el parámetro ρ_1 varía lo suficientemente lento de pel a pel.

4.6 Aproximación de Walker y Rao

Los anteriores algoritmos propuestos por Netravali y Robbins tienen la característica de tener ganancias constantes. Esto puede resultar una desventaja cuando estas no son fijadas con un valor adecuado, por ejemplo, si se tienen una ganancia pequeña y movimientos grandes se necesitarían muchas iteraciones para actualizar el vector de desplazamiento, y si por el contrario se tiene una ganancia muy alta y movimientos pequeños, el algoritmo no convergerá.

Las condiciones requieren que un gran vector de corrección sea proporcionado cuando la $|DFD|$ es grande y el $|VI|$ es pequeño. Análogamente cuando la $|DFD|$ es pequeña y el $|VI|$ grande (como es el caso cuando existe un borde), el término de corrección debe ser pequeño.

Walker y Rao [WAL84] sugirieron una ganancia adaptativa a la ecuación 4.6 como solución al problema anterior:

$$\epsilon' = \frac{1}{2 \|\nabla l(p-d^{t-1}, t-1)\|^2} \quad \dots 4.27$$

De esta forma la ecuación 4.7 toma la forma:

$$d^t = d^{t-1} - \frac{1}{2} \frac{DFD(p, d^{t-1}) \nabla l(p, d^{t-1}, t-1)}{\|\nabla l(p-d^{t-1}, t-1)\|^2} \quad \dots 4.28$$

Así con esta ganancia adaptativa, los algoritmos propuestos por Netravali y Robbins, no dependerán de una ganancia que la mayoría de las veces se asigna gracias a conocimientos empíricos de la secuencia analizada. De igual modo, Cafforio y Rocca [WAL84] desarrollaron una expresión similar:

$$\epsilon' = \frac{1}{\sigma^2 + \|\nabla l(p-d^{t-1}, t-1)\|^2} \quad \dots 4.29$$

donde $\sigma^2=100$ y se incluye con la finalidad de controlar los casos en que el gradiente de l es pequeño o cero.

5. Técnicas por apareamiento de bloques

5.1 Introducción

La estimación y compensación de movimiento basadas en métodos de bloques, son de las más populares aproximaciones por su simplicidad de *Hardware*, tanto que los estándares internacionales de compresión de vídeo, tales como H.261 y MPEG 1-2 han adoptado estos métodos [MUR95]. Para reducir la complejidad computacional la imagen se divide en bloques de tamaño fijo con el mismo desplazamiento para cada pel incluido en el bloque.

5.2 Modelos de movimiento de bloques.

Partiendo del concepto anteriormente citado, se consideran tres tipos de movimientos en los bloques, la translación, la rotación y la deformación. Debido a la complejidad computacional de los modelos que incluyen rotación y deformación, el enfoque estándar es hacia el modelo traslacional, por lo que las implantaciones en VLSI disponibles están orientadas hacia este modelo.

5.2.1 Movimiento traslacional.

El supuesto básico de este modelo consiste en considerar un bloque β de dimensiones $M \times N$ en la escena t centrado en el pel $p=(x, y)$, que deberá estar presente en la escena $t-1$ pero desplazado una distancia d :

$$\begin{aligned} I(p-d, t) &= I(p, t-1) \\ \forall p \in \beta & \end{aligned} \quad \dots 5.1$$

donde d es el vector de desplazamiento para el bloque. La nueva posición de cada pel del bloque en la imagen t , es:

$$\begin{aligned}x'_1 &= x_1 + d_x \\y'_1 &= y_1 + d_y\end{aligned}\quad \dots 5.2$$

Este modelo considera posibles traslapes de los bloques. En el caso en que no haya traslape, el vector d es el mismo para cada pel del bloque, en caso contrario, se puede asignar un promedio de los vectores estimados para los pels que queden dentro de la región de traslape o bien se selecciona un vector de los ya estimados, esto último con la finalidad de ahorrar cálculos.

Sin embargo, este modelo no es completo puesto que no incluye el movimiento rotacional, los acercamientos, deformaciones locales y traslape de objetos en la escena. Además los bloques pueden generar deformaciones en la imagen reconstruida debido a efectos de frontera.

5.2.2 Deformación de bloques

La transformación espacial del bloque puede ser generalizada para incluir transformaciones de coordenadas más complejas, para manejar rotación y deformación, como la transformada *affine* que modela una proyección ortogonal dada por:

$$\begin{aligned}x'_1 &= a_1x_1 + a_2y_1 \\y'_1 &= a_3x_1 + a_4y_1\end{aligned}\quad \dots 5.3$$

La transformación de perspectiva modela, como su nombre lo indica, proyecciones en perspectiva. Está dada por:

$$\begin{aligned}x'_1 &= \frac{a_1x_1 + a_2y_1 + a_3}{a_7x_1 + a_8y_1 + 1} \\y'_1 &= \frac{a_4x_1 + a_5y_1 + a_6}{a_7x_1 + a_8y_1 + 1}\end{aligned}\quad \dots 5.4$$

y la transformación bilineal, que no modela ningún tipo de movimiento físico puede ser expresada como:

$$x_1' = a_1 x_1 + a_2 y_1 + a_3 x_1 y_1 + a_4$$

... 5.5

$$y_1' = a_5 x_1 + a_6 y_1 + a_7 x_1 y_1 + a_8$$

La figura 5-1 muestra los efectos de estas transformaciones sobre ciertos polígonos.

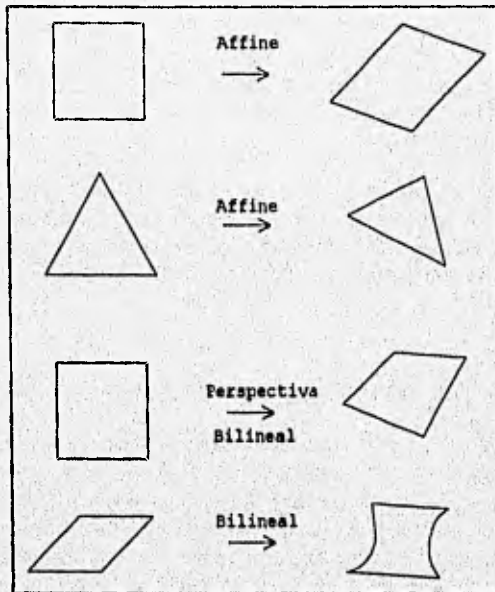


Figura 5-1 Diferentes transformaciones para deformaciones de bloques.

Aunque los métodos de bloques apareados y de correlación de fase están basados en el modelo traslacional, estos pueden generalizarse al uso de bloques deformables.

5.3 Método de correlación de fase.

Tomando la transformada de Fourier en 2-D en ambos lados del modelo de invarianza de la luminancia, dada en la ecuación 5.1, sobre el bloque β , se obtiene:

$$I(f, t) = I(f, t-1) e^{-j2\pi(d, f, +d, f, t)}$$

... 5.6

donde $I(f, t)$ denota la transformada de Fourier en 2-D de la escena t , con respecto a las variables espaciales (x, y) . Esto induce a que, en el caso de un movimiento traslacional, la diferencia de fases de cada uno de los respectivos bloques sea:

$$\arg\{I(t-1, f)\} - \arg\{I(t, f)\} = 2\pi(d_x f_x + d_y f_y) \quad \dots 5.7$$

La diferencia de fases en la ecuación anterior define un plano sobre las variables f_x y f_y . El vector de movimiento entre escenas puede ser estimado por la orientación de este plano. Esta aproximación tiene como problemas que la obtención de la estimación del plano generalmente requiere la obtención de la fase en 2-D y no es fácil identificar el vector de movimiento con dos o más objetos moviéndose dentro del bloque.

El método de correlación de fases estima un cambio relativo entre dos bloques de las imágenes t y $t-1$, por medio de una función normalizada de correlación cruzada. Se basa en el principio de que un cambio de movimiento relativo en el dominio espacial resulta en un término de fase lineal en el dominio de Fourier.

5.4 Método por apareamiento de bloques.

La idea básica del método de apareamiento de bloques consiste en dividir la imagen en bloques y asignar a cada bloque un solo vector de movimiento. Este vector es hallado de la siguiente manera: teniendo un bloque β de $M \times N$ centrado en el punto p de la escena t , se busca el bloque del mismo tamaño que más se le parezca en la escena $t-1$, es decir, el que se aparee mejor. Hecho esto, el vector de desplazamiento será $p-p'$, donde p' es el punto donde se centra el bloque hallado en la imagen t . La búsqueda está usualmente limitada a una ventana de dimensiones $M+2M, \times N+2N$, llamada ventana de búsqueda.

Los algoritmos de apareamiento de bloques, se clasifican en cuanto al criterio de apareamiento y la estrategia de búsqueda. La calidad del resultado dependerá en gran medida del tamaño elegido del bloque y de los parámetros de la estrategia de búsqueda.

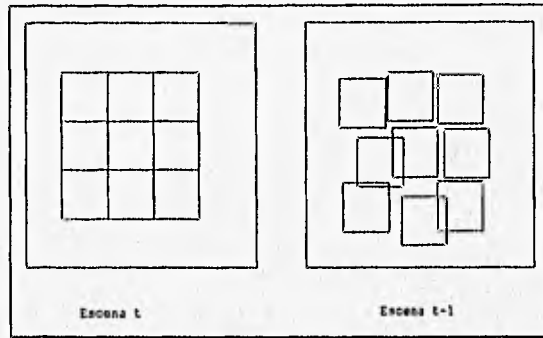


Figura 5.2 Estrategia por apareamiento de bloques.
Se observa que pueden existir traslapes entre los bloques.

5.4.1 Funciones de máxima similitud.

El criterio de apareamiento consiste en la elección de una función de máxima similitud. El apareamiento de los bloques puede ser cuantificado de acuerdo a diversas funciones, incluyendo la de máxima correlación cruzada, mínimo error cuadrático medio (EQM), mínima diferencia absoluta media (MAD) y la máxima cuenta de pel apareado (MPC).

El criterio de mínimo error cuadrático medio, EQM, es calculado de la siguiente manera:

$$EQM(d) = \frac{1}{M \times N} \sum_{p \in \beta} [I(p-d, t-1) - I(p, t)]^2 \quad \dots 5.8$$

donde β denota un bloque de dimensiones $M \times N$ y $d = (dx, dy)$ es un vector de desplazamiento.

El vector d seleccionado será aquel que minimice el EQM.

Por otro lado, la diferencia absoluta media se define como:

$$MAD(d) = \frac{1}{M \times N} \sum_{p \in \beta} |I(p-d, t-1) - I(p, t)| \quad \dots 5.9$$

En este criterio de apareamiento existe un problema y es que este se deteriora conforme el tamaño de la ventana de búsqueda aumenta, debido a la presencia de varios mínimos locales.

Otra alternativa es el criterio de máxima cuenta de pel apareado; en esta aproximación, cada pel dentro del bloque β es clasificado como pel apareado o como pel no apareado, de acuerdo a:

$$A(p, d) = \begin{cases} 1 & \text{si } |I(p-d, t-1) - I(p, t)| \leq T \\ 0 & \text{en otro caso.} \end{cases} \quad \dots 5.10$$

donde T es un umbral predeterminado. Así el número de pels apareados es dado por:

$$MPC(d) = \sum_{p \in \beta} A(p, d) \quad \dots 5.11$$

Esto es, la estimación del vector de movimiento es el valor d' con el que se encuentre el más alto número de pels apareados.

5.4.2 Estrategias de búsqueda.

Una mejor estimación del vector de desplazamiento es obtenida al optimizar los criterios de apareamiento. Estos pueden ser evaluados con la llamada "búsqueda completa" sobre todos los posibles valores que pueda tomar d . Aunque esta estrategia es la mejor, consume demasiados recursos de cómputo, recurriéndose así a estrategias más sencillas.

La primer medida para reducir el tiempo de cómputo usualmente consiste en limitar el área de búsqueda a una ventana de dimensiones $2M+M_j$, $2N+N_j$, llamada ventana de búsqueda, la cual está centrada alrededor del pel para el cual el vector de movimiento será estimado, donde M_j y N_j son enteros predeterminados.

Otra práctica comúnmente empleada para obtener una menor carga computacional es obtener la estimación de vectores de movimiento en una malla de pels con el fin de no comparar en todos los pels del bloque. Por ejemplo cada ocho pels y cada ocho líneas usando un bloque de 16×16 , y después interpolando el campo de movimiento a los pels no usados dentro del bloque.

Con el fin de no realizar una búsqueda completa, se recurre a algoritmos subóptimos entre los cuales se cuenta con el método de la búsqueda de los tres pasos y el de búsqueda cruzada.

5.4.2.1 Búsqueda de los tres pasos.

Cosiste en evaluar el criterio de apareamiento sobre un conjunto discreto de vectores de desplazamiento, eligiendo el mejor y tomándolo como base en una siguiente iteración; el nuevo conjunto discreto de vectores disponibles posee una amplitud menor.

El algoritmo se describe a continuación:

Paso 1.

Se identifican en las direcciones N, S, W, E, NE, NW, SE, SW los puntos a una distancia $l/2$ del punto centrado en el bloque de la escena actual en proceso, y también a una distancia cero.

Paso 2.

Se aplica el criterio de máxima similitud elegido, comparando un bloque de la escena anterior centrado, en cada uno de los puntos identificados anteriormente, con el bloque de la escena actual en proceso.

Paso 3.

Se escoge el punto donde el criterio de máxima similitud obtuvo un mejor resultado, y a partir de ese punto, se repite el paso uno, disminuyendo la distancia l a la mitad, hasta que $l=1$

El vector de desplazamiento asignado a cada uno de los pels del bloque en cuestión se obtiene restando el último punto elegido en el paso tres, del punto donde se centra el bloque actual. El nombre de este algoritmo se debe a que originalmente la distancia inicial para el paso l era de 8, esto es, 2^3 .

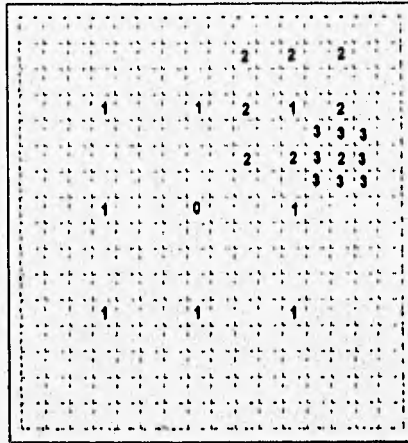


Figura 5.3 Ejemplo de búsqueda con el método de los tres pasos.

5.4.2.2 Búsqueda cruzada.

A diferencia del algoritmo anterior, el algoritmo de búsqueda cruzada no disminuye la distancia de búsqueda a menos que el vector elegido sea nulo. Las variantes en este método dependerán del conjunto de vectores elegidos para la evaluación del criterio de máxima similitud. El algoritmo que se describe a continuación utiliza un conjunto de vectores, de tal forma que los puntos donde se evalúa forman el signo de "+".

Paso 1.

Se identifican en las direcciones N, S, W, E, los puntos a una distancia 1/2 del punto centrado en el bloque de la escena actual en proceso, y también a una distancia cero.

Paso 2.

Se aplica el criterio de máxima similitud elegido, comparando un bloque de la escena anterior, centrado en cada uno de los puntos identificados anteriormente, con el bloque de la escena actual en proceso.

Paso 3.

Se escoge el punto donde el criterio de máxima similitud obtuvo un mejor resultado, y a partir de ese punto, se repite el paso uno. Si el punto elegido es igual al punto donde se centra el bloque actual en proceso, o es igual al obtenido del paso

tres de la iteración anterior, la distancia l se reduce a la mitad, en caso contrario, l no se modifica. El proceso se repite hasta que $l=1$.

El vector de desplazamiento asignado a cada uno de los pels del bloque en cuestión, se obtiene restando el último punto elegido en el paso tres, del punto donde se centra el bloque actual.

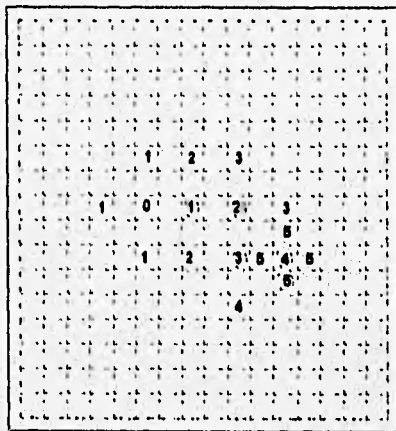


Figura 5.4 Ejemplo de búsqueda cruzada.

6. Estimación de movimiento, casos multiresolución y multirestricciones.

6.1 Introducción

Para la estimación de un campo de movimiento aparente, la sola hipótesis de la invarianza de la luminancia, es insuficiente. Los métodos multirestricciones abordan este problema por medio de la hipótesis de la existencia de varias funciones invariantes al movimiento. En contraparte los métodos piramidales o multiresolución se enfocan a la obtención de un campo inicial de estimación del movimiento aparente entre las imágenes procesadas a una resolución menor.

6.2 Métodos multirestricciones

Como hemos visto en capítulos anteriores, la sola ecuación:

$$\nabla IU + \frac{d}{dt} I = 0 \quad \dots 6.1$$

es insuficiente para la estimación de las dos componentes de $U=(dx/dt, dy/dt)$, de aquí la necesidad de restricciones adicionales que nos permitan dicha estimación. Una alternativa es buscar alguna otra función invariante de movimiento J . Si al instante $t=0$ existen dos invariantes de movimiento I y J , tales que $I(p) = I^*$ y $J(p) = J^*$, al instante t el punto p puede ser localizado en la intersección de las funciones I y J , para cualquier t .

El movimiento de p puede ser identificado de manera única, cuando las dos funciones I y J se intersectan, lo anterior garantiza una solución única para el siguiente sistema lineal:

$$\begin{aligned} \nabla IU + \frac{d}{dt} I &= 0 \\ \nabla JU + \frac{d}{dt} J &= 0 \end{aligned} \quad \dots 6.2$$

De manera general, para cada punto de la imagen, se pueden considerar varias funciones invariantes de movimiento I^i , el conjunto de estas funciones forman un sistema lineal denominado, sistema multirestricciones:

$$\begin{aligned} \nabla I^1 U + \frac{d}{dt} I^1 &= 0 \\ \vdots \\ \nabla I^N U + \frac{d}{dt} I^N &= 0 \end{aligned} \quad \dots 6.3$$

donde $N \geq 2$.

La existencia de una solución única es determinada por la disponibilidad de dos ecuaciones linealmente independientes.

6.2.1 Búsqueda de funciones invariantes de movimiento.

Las funciones invariantes de movimiento pueden ser clasificadas según su origen:

- I^i resulta de la formulación de restricciones sobre la señal o el campo de movimiento, tales como la invarianza de la amplitud del gradiente espacial de la luminancia, con respecto a la rotación, con respecto a la expansión uniforme, y con respecto a un movimiento hiperbólico, por ejemplo.
- I^i resulta de la aplicación de operadores locales a la señal original I , para extraer características locales, tales como momentos de la distribución, de la intensidad luminosa (media, varianza, etc.), o medidas de contraste, entropía o de energía.
- I^i Es una función multispectral proveniente de diferentes bandas frecuenciales, relativas a la señal a analizar I . Las componentes de crominancia rojo, verde y azul, pueden formar un sistema multirestricciones, en el supuesto de la invarianza de la crominancia al movimiento en cada una de sus componentes y de que sean linealmente independientes unas de otras. El mismo tratamiento puede ser aplicado a bandas frecuenciales producto de una transformación multiresolución de la imagen original.

6.2.2 Aproximación multirestricción - multiresolución para la estimación de movimiento.

El método siguiente se funda en la hipótesis de un movimiento aparente idéntico en todas las bandas frecuenciales de la señal a analizar I . De esta forma, el método multirestricciones tendrá una función invariante de movimiento en cada banda frecuencial. Así podemos explotar la EMC para cada nivel de resolución, siendo cada banda frecuencial considerada como una nueva imagen, sobre la que aplicaremos los tratamientos habituales.

6.3 Tratamiento de los métodos multirestricciones

El sistema multirestricciones es:

$$\begin{bmatrix} I_x^1 & I_y^1 \\ I_x^2 & I_y^2 \\ \vdots & \vdots \\ I_x^N & I_y^N \end{bmatrix} \begin{bmatrix} U_x \\ U_y \end{bmatrix} = \begin{bmatrix} I_x^1 \\ I_y^1 \\ \vdots \\ I_x^N \\ I_y^N \end{bmatrix} \quad \dots 6.4$$

Despejando U se tiene:

$$\begin{bmatrix} U_x \\ U_y \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^N (I_x^j)^2 & \sum_{j=1}^N I_x^j I_y^j \\ \sum_{j=1}^N I_x^j I_y^j & \sum_{j=1}^N (I_y^j)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{j=1}^N I_x^j I_x^j \\ \sum_{j=1}^N I_y^j I_y^j \end{bmatrix} \quad \dots 6.5$$

Se puede observar que esta solución se basa esencialmente en el cálculo de derivadas espaciales y temporales, las cuales son muy sensibles al ruido y esto conduce a estimaciones erróneas.

Por esta razón, se recurre a métodos de regularización para eliminar este inconveniente. En este caso, la solución propuesta por [BAZ91] es un método de resolución iterativo y recursivo.

6.3.1 Solución iterativa recursiva.

A partir de la ecuación 4.18, se establece un sistema similar al 6.4:

$$\begin{aligned}
 -I_x^1(p-d^{t-1}, t-1)W_x - I_y^1(p-d^{t-1}, t-1)W_y &= DFD^1(p, d^{t-1}) \\
 -I_x^2(p-d^{t-1}, t-1)W_x - I_y^2(p-d^{t-1}, t-1)W_y &= DFD^2(p, d^{t-1}) \\
 &\vdots \\
 -I_x^N(p-d^{t-1}, t-1)W_x - I_y^N(p-d^{t-1}, t-1)W_y &= DFD^N(p, d^{t-1})
 \end{aligned} \quad \dots 6.6$$

donde:

$$W = \begin{bmatrix} dx - dx^{t-1} \\ dy - dy^{t-1} \end{bmatrix} \quad \dots 6.7$$

Despejando el vector de desplazamiento obtenemos la solución final:

$$\begin{bmatrix} \hat{d}_x^t \\ \hat{d}_y^t \end{bmatrix} = \begin{bmatrix} \hat{d}_x^{t-1} \\ \hat{d}_y^{t-1} \end{bmatrix} - \left[\begin{array}{cc} \sum_{j=1}^N (I_x^j)^2 & \sum_{j=1}^N (I_x^j I_y^j) \\ \sum_{j=1}^N (I_x^j I_y^j) & \sum_{j=1}^N (I_y^j)^2 \end{array} \right]^{-1} \begin{bmatrix} \sum_{j=1}^N (I_x^j DFD^j) \\ \sum_{j=1}^N (I_y^j DFD^j) \end{bmatrix} \quad \dots 6.9$$

Como se puede observar, la única condición para la existencia de la solución, es la no singularidad de la matriz de gradientes involucrada.

6.4 ESTIMACIÓN DE MOVIMIENTO EN MULTIRESOLUCIÓN

Un método para acelerar la convergencia de los métodos de estimación de movimiento es la adopción de estructuras piramidales (multiresolución), que puede ser implantado por medio de arquitecturas paralelas.

La estimación de movimiento multiresolución es una extensión de las técnicas monoresolución, aplicada de manera jerárquica a los diferentes niveles de

la estructura piramidal, propagando los vectores estimados de desplazamiento de un nivel mayor a uno menor, para utilizarlo como condición inicial del nuevo proceso iterativo.

Así, el problema se descompone en tres partes fundamentales:

1. La elección de un algoritmo de estimación válido en monoresolución.
2. La elección de una estructura piramidal. La estructura más utilizada es la pirámide paso bajas Gaussiana.
3. La elección de una estrategia de propagación de datos de nivel a nivel. Este punto considera esencialmente la definición del sentido de propagación de las estimaciones en la estructura jerárquica (ascendente, descendente o de manera oscilatoria) y la definición de operadores de interpolación.

La estrategia más utilizada es llamada descendente porque propaga los procesos de estimación de las bajas resoluciones hacia las altas resoluciones. Debido a que las bajas frecuencias espaciales representan las estructuras de la imagen a gran escala y pueden ser utilizadas para la medición inicial de los desplazamientos grandes. Las altas frecuencias espaciales representan los detalles finos de la imagen, y pueden ser utilizadas para refinar los vectores de movimiento. La estrategia descendente explota de manera ventajosa estas propiedades, comenzando con una etapa de estimación burda que será refinada sucesivamente en procesos de estimación sobre resoluciones más altas.

La propagación de los vectores de desplazamiento estimados sobre la estructura piramidal en cada resolución se realiza por medio de interpolación cuando la transferencia es descendente (de bajas a altas resoluciones).

En nuestro estudio, trataremos la estrategia descendente. En esta, el campo de vectores obtenido en una capa de resolución baja es interpolada a la superior inmediata, para luego ser utilizado como estimación inicial en un nuevo proceso de estimación que procederá a su refinamiento. En la estrategia descendente, se presenta el problema de elegir el número de niveles que se van a tratar con la pirámide de datos, este valor es calculado a partir de la magnitud máxima del movimiento a estimar; si este valor es V_{max} entonces el nivel máximo del nivel piramidal será $\text{Log}_2(V_{max}) + 1$. [BAZ91]

6.4.1. Estimación de movimiento sobre pirámides pasobajas.

Es importante precisar que toda estimación en multiresolución es muy dependiente del tipo de estructura jerárquica elegida.

Dada la simplicidad de la pirámide paso bajas, esta es frecuentemente utilizada para la estimación de movimiento, siendo la estrategia descendente usada del siguiente modo:

1. Se generan el conjunto de imágenes $(G_0(t-1), \dots, G_{N-1}(t-1))$ y $(G_0(t), \dots, G_{N-1}(t))$ para formar dos pirámides paso-bajas correspondientes a las escenas $t-1$ y t . El proceso comienza por la estimación de un campo de desplazamiento d_N sobre las cimas de las pirámides, es decir en las imágenes $G_N(t-1)$ y $G_N(t)$.
2. El campo de vectores obtenido d_j es interpolado, su tamaño es multiplicado por 4 para tener un tamaño que concuerde con el cambio de resolución realizado.
3. El campo de vectores así obtenido, es utilizado como campo de desplazamiento inicial d_{j-1}^0 para un nuevo proceso de estimación.

Hemos visto que los métodos pel-recursivos requieren de un valor inicial de desplazamiento para comenzar su proceso de estimación. En el caso monoresolución este valor resultaba de la propagación de una o más estimaciones de vecinos espaciales. Cuando hablamos de una estrategia multiresolución, otro candidato es añadido para los valores iniciales, el cual provendrá de un campo de vectores estimado en el nivel inmediato anterior.

Las etapas 2 y 3 se repiten hasta alcanzar la base de la pirámide G_0 .

7. Compensación de movimiento

7.1 Introducción

La compensación de movimiento es una extensión de las técnicas de estimación de movimiento, a partir de una estructura diferencial calcula y trata de minimizar el error de predicción de movimiento en las imágenes, para transmitir únicamente dicho error con lo que se reduce la cantidad de información enviada.

Esta se descompone en dos partes: la primera es una técnica de estimación usada para predecir el movimiento entre pels de las escenas de la secuencia. En la segunda, el vector de desplazamiento obtenido así, debe ser usado óptimamente para proveer una reconstrucción adecuada en el receptor, el método puede ser expandido a una estructura piramidal.

El método de compensación de movimiento utilizado por los predecesores de Walker y Rao es como sigue: la transmisión de los pels no predichos, así como la de sus direcciones se realiza mediante codificación de longitud variable, *run length*. Si el valor absoluto de la DFD es menor a cierto umbral propuesto no se transmite ningún dato. El método propuesto por Walker y Rao [WAL84] envía la DFD cuando esta es mayor a un umbral, en caso contrario se envía un cero. De esta manera ya no es necesario transmitir las direcciones de los pels no compensados. El flujo de datos se codifica usando longitud variable, de tal manera que si la predicción es buena, la mayor parte de los datos serán ceros, pudiéndose transmitir con una longitud mínima.

Es necesario recalcar para este algoritmo, la preservación de la dualidad al nivel de codificador y decodificador requerida para la reconstrucción de las imágenes en el decodificador.

7.2 Algoritmo del codificador:

1. Usando la \hat{d}^i obtenida del pel anterior, predecir el pel actual, obteniendo el valor del pel correspondiente en la escena anterior desplazado \hat{d}^i .
2. Encontrar la FD y la DFD. Aquí es importante hacer notar que, las imágenes utilizadas para obtenerlas incluyen los efectos del

cuantizador. Esto es: $FD = I(p, t) - I'(p, t-1)$ y $DFD = I(p, t) - I'(p - \hat{d}^{t-1}, t-1)$ donde I' es la imagen reconstruida.

3. Si $|DFD| \leq umbral$, se transmite cero. Si $|DFD| > umbral$ y $|FD| < umbral$, se transmite un *reset* para hacer que $\hat{d}^t = 0$. Si $|DFD| > umbral$ y $|FD| > umbral$, se transmite la DFD. Se reconstruye el valor de $I'(p, t) = I'(p - \hat{d}^t, t-1) + (\text{valor transmitido})$.
4. a) Si $|DFD| \leq umbral$, asignar la \hat{d}^t obtenida del pel anterior al pel actual.
 b) Si $|DFD| > umbral$ y $|FD| < umbral$, asignar $\hat{d}^t = 0$.
 c) Si $|FD| > umbral$ y $|DFD| > umbral$, entonces usar $I'(p, t)$ para encontrar una nueva \hat{d}^t con la que se tratará de minimizar la DFD, de tal forma que $|DFD|$ sea menor al umbral. La \hat{d}^t anterior se usará como estimación inicial para el proceso iterativo, que se realizará hasta que se cumpla que $|DFD| \leq umbral$, o se exceda el límite en el número de iteraciones.
5. Proseguir al siguiente pel y volver al paso 1.

En la figura 7.1 se muestra el diagrama del codificador. Se puede observar la localización del cuantizador dentro del lazo. También se observa que el predictor no usa la imagen I , puesto que esta no está disponible en el decodificador, donde se debe realizar un proceso gemelo. Es importante mencionar que el retardo al que se hace mención en el diagrama corresponde al de una imagen completa.

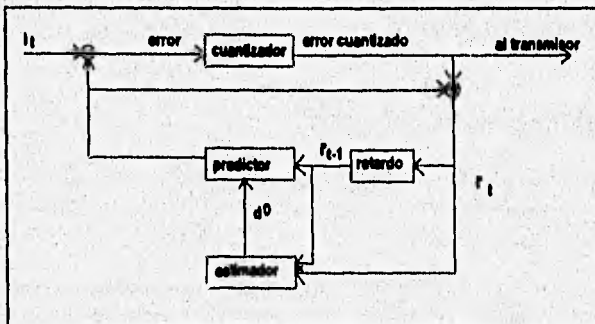


Figura 7.1 Diagrama de bloques del codificador para compensación de movimiento

7.3 Algoritmo del decodificador:

1. Usar \hat{d}^i del pel anterior para predecir el valor del pel actual.
2. Recibir la DFD y añadir esta a la predicción en el pel actual para obtener la reconstrucción en $I'(p, t)$. Si se recibe un *reset*, $I'(p, t) = I(p, t-1)$
3. a) Si se recibe un cero, asignar la \hat{d}^i obtenida del pel anterior al pel actual.
b) Si se recibe un *reset*, asignar $\hat{d}^i = 0$.
c) En cualquier otro caso usar la $I'(p, t)$ obtenida en el paso 2, para actualizar el valor de \hat{d}^i usando el mismo algoritmo usado en el codificador.

La figura 7.2 muestra el diagrama de bloques del decodificador. La señal de entrada es únicamente el error cuantizado proveniente del codificador. Al igual que en el codificador, el retardo en el lazo de realimentación es el de una imagen completa.

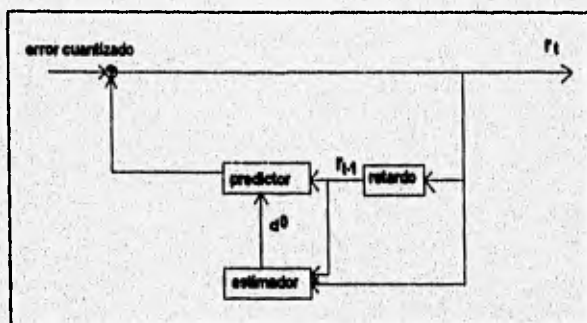


Figura 7.2 Diagrama de bloques del decodificador para compensación de movimiento

Cabe mencionar que la calidad de la imagen reconstruida depende en gran medida del umbral usado para la comparación de la DFD y si la señal de error se cuantifica, este proceso deberá ser realizado antes del proceso de reconstrucción llevado a cabo en el codificador.

8 . Implantación y comparación de los algoritmos.

8.1 *Introducción*

Este capítulo está dedicado a la revisión de los resultados de los algoritmos de estimación y compensación de movimiento implantados para su simulación.

En primer lugar, se describen las estructuras de los algoritmos planteados. Posteriormente se ejemplifican con la simulación de estos sobre secuencias reales de vídeo. Ganancias óptimas son obtenidas para los algoritmos de Netravali y Robbins, ya que el desempeño de estos depende directamente del valor particular de la ganancia para cada secuencia.

Los parámetros para la evaluación de los métodos de estimación de movimiento obtenidos son: la relación señal a ruido, el error cuadrático medio, la entropía de la señal de salida y el porcentaje de pels en los cuales los algoritmos no convergen.

El codificador utiliza algunos de los algoritmos de estimación de movimiento implantados en este trabajo y se evalúa con los mismos parámetros anteriormente citados.

8.1.2 *Criterios de evaluación.*

Los resultados son evaluados de acuerdo a:

- La calidad de la estimación, en términos de píxeles compensados o no compensados
- La calidad de la reconstrucción, calificada visualmente y cuantitativamente a través del error cuadrático medio, EQM, y por la relación señal a ruido, SNR, entre la imagen original y la imagen reconstruida.

El cálculo del error cuadrático medio está definido por:

$$EQM = \frac{1}{MN} \sum_p [I(p,t) - I'(p,t)]^2 \quad \dots 8.1$$

donde la sumatoria se realiza sobre todos los puntos p de la imagen, $I(p,t)$ es la imagen original, $I'(p,t)$ es la imagen reconstruida por el decodificador y (N, M) son el número de líneas y columnas de la imagen tratada.

El otro criterio de calidad usado en el presente trabajo es la relación señal a ruido, SNR, la cual está dada por la siguiente expresión:

$$SNR = 10 \log \left[\frac{[I_{MAX} - I_{MIN}]^2}{EQM} \right] \quad \dots 8.2$$

donde I_{MIN} e I_{MAX} son los niveles mínimo y máximo de cuantización de las imágenes tratadas. Los niveles mínimo y máximo de 0 y 255 son frecuentemente utilizados para imágenes codificadas sobre 8 bits.

La calidad de una imagen se juzga aceptable cuando el valor de su SNR está arriba de los 30 dB.

Los experimentos presentados están aplicados a una secuencia de imágenes reales llamada "Interview" donde se presenta a una mujer que se está levantando y la cámara tiene un movimiento de translación hacia arriba. El tamaño de las imágenes es de 674×536 pels.

Dos imágenes de la secuencia, y la diferencia temporal entre dos consecutivas, son mostradas en las páginas siguientes. El valor del error cuadrático medio es indicado también para recalcar la variación temporal de la luminancia a lo largo de la secuencia.

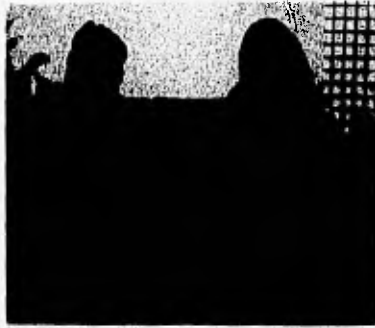


Figura 8.1 Dos escenas de la secuencia Interview.



Figura 8.2 Imagen normalizada resultante de la diferencia temporal entre las escenas tres y cuatro de la secuencia "Interview"; el error cuadrático medio es 152.56

La obtención del gradiente es un punto muy importante en la implantación del problema, sin embargo no existe una solución única, siendo uno de los métodos más utilizados el descrito a continuación. El cálculo del gradiente es muy

susceptible al ruido, por lo tanto a veces se recomienda un filtrado previo de la señal [WAN92].

Los gradientes espaciales de la intensidad (I_x, I_y) son calculados por la convolución de la imagen con cada una de las siguientes mascarar:

$$h_x = \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_y = \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \dots 8.3$$

con lo que:

$$\nabla I(x, y) = [\nabla_x I(x, y), \nabla_y I(x, y)]^T$$

donde

$$\nabla_x I(x, y) = I(x, y+1) + I(x-1, y+1) + I(x+1, y+1) - I(x, y-1) - I(x-1, y-1) - I(x+1, y-1)$$

y

$$\nabla_y I(x, y) = I(x-1, y) + I(x-1, y+1) + I(x-1, y-1) - I(x+1, y) - I(x+1, y-1) - I(x+1, y+1) \quad \dots 8.4$$

8.2 Estructura general de los algoritmos pel-recursivos

La solución iterativa general para los algoritmos estudiados es:

$$d^i = d^{i-1} - TC \quad \dots 8.5$$

donde i es la iteración actual y $TC = (TC_x, TC_y)^T$ es el término de corrección.

A la ecuación anterior están asociadas un conjunto de pruebas de control del proceso iterativo:

- Limitación del término de corrección; si el término de corrección es muy grande, no se asegura la convergencia y un comportamiento oscilatorio puede ocurrir. En el caso contrario, un término de corrección muy pequeño disminuye la rapidez de la convergencia:
- El cálculo iterativo se detiene cuando $|DFD(p, d^i)| \leq T$; donde T es un umbral previamente establecido. En casos de convergencia lenta, el proceso es detenido en un número máximo de iteraciones *IMAX*.

- Elección de un valor inicial. La elección de un valor de desplazamiento entre los obtenidos para los pels vecinos es recomendado para acelerar el proceso de convergencia y mejorar la calidad del vector de desplazamiento. Se tienen varias elecciones entre otras:
 1. Un valor nulo.
 2. Elegir el vector de desplazamiento entre los obtenidos para los pels vecinos, tal que produzca una menor DFD en el pel actual.
 3. Elegir el vector de desplazamiento entre los obtenidos para los pels vecinos de la trama anterior, que produzca la menor DFD para el pel actual, esto es, un vector propagado temporalmente.
 4. Propagar el vector inicial de desplazamiento a partir de una estructura multiresolución
 5. Una combinación lineal de las cuatro anteriores que incluso se puede extender hacia varias tramas anteriores de la secuencia de tal modo que se elige el vector de desplazamiento con la menor DFD.

El segundo método es el más usado por su simplicidad y eficacia para acelerar la convergencia. En este estudio, el vector de desplazamiento inicial es elegido del valor de desplazamiento de algún pel de entre 4 vecinos al actual, el correspondiente a la trama anterior y un vector nulo, como se muestra a continuación:

XXX	X: Pel vecino.
XO	O: Pel actual.

Recursividad: El proceso de estimación utiliza de manera conjunta el proceso de iteración y la propagación de estimaciones precedentes. Este va a efectuarse de pel a pel en la imagen, siguiendo el barrido convencional de una línea de televisión. El algoritmo es el siguiente:

Paso 1.

Seleccionar el valor inicial v_i para un pel p : $d^0 = v_i$

El valor del vector inicial v_i es el que cumple con la menor DFD de entre los valores del vector de desplazamiento asociado a los pels vecinos ya calculados, un vector nulo, valor inicial propagado temporalmente y en el caso de algoritmos multiresolución, a través de interpolación de una estructura piramidal.

Paso 2.

Si $|DFD(p, v_i)| \leq T$ entonces $d = v_i$ y se procede con el siguiente pel. En caso contrario se pasa al calculo iterativo.

Paso 3.

Se calcula el término de corrección TC y se restringe a un cierto rango, con la finalidad de evitar la divergencia del algoritmo.

Se actualiza el vector de desplazamiento $d_i = d_{i-1} - TC$ hasta que el valor de $|DFD(p, d_i)| \leq T$ o bien i alcance el valor máximo permitido de iteraciones.

Paso 4.

Si el valor del vector de desplazamiento obtenido cumple con $|DFD(p, d_i)| \leq T$, entonces $d = d_i$, en caso contrario, si se alcanzó el límite en el número permitido de iteraciones, d se elige entre v_i , d_i y un vector nulo, tal que la DFD sea mínima.

Si el vector de desplazamiento es mayor que un desplazamiento máximo determinado a priori, entonces el vector se nullifica debido a que es muy probable que el método haya divergido para ese pel en particular.

Paso 5.

Se procede con el siguiente pel.

8.2.1 Cálculo del Término de corrección para los diferentes algoritmos

8.2.1.1 Algoritmo de Netravall y Robbins, ecuación 4.7

$$d^i = d^{i-1} - \epsilon DFD(p, d^{i-1}) \nabla I(p - d^{i-1}, t - 1)$$

8.2.1.2 Algoritmo de Netravall y Robbins, Ecuación 4.9

$$d^i = d^{i-1} - \epsilon \operatorname{sgn}[DFD(p, d^{i-1})] \operatorname{sgn}[\nabla I(p - d^{i-1}, t - 1)]$$

8.2.1.3 Algoritmo de Netravali y Robbins, Ecuación 4.14

$$d^t = d^{t-1} - \varepsilon \left\{ \sum_{j=0}^k W_j DFD(p_{x-j}, d^{t-1}) \nabla [I(p_{x-j} - d^{t-1}, t-1)] \right\}$$

8.2.1.4 Algoritmo de Netravali y Robbins, Ecuación 4.19

$$d^t = d^{t-1} - \varepsilon \left[\sum_{j=0}^k \nabla I(p_{x-j} - d^{t-1}, t-1) \nabla I^T(p_{x-j} - d^{t-1}, t-1) \right]^{-1} \\ * \left[\sum_{j=0}^k DFD(p_{x-j}, d^{t-1}) \nabla I(p_{x-j} - d^{t-1}, t-1) \right]$$

8.2.1.5 Algoritmo de Netravali y Robbins, Ecuación 4.25

$$\rho_2^t = \rho_2^{t-1} + \varepsilon_1 DFD(p, \rho_2^{t-1}, d^{t-1}) I(p - d^{t-1}, t-1) \\ d^t = d^{t-1} - \varepsilon_2 \rho_2^{t-1} DFD(p, \rho_2^{t-1}, d^{t-1}) \nabla I(p - d^{t-1}, t-1)$$

8.2.1.6 Algoritmo de Walker y Rao, Ecuación 4.28

$$d^t = d^{t-1} - \frac{DFD(p, d^{t-1}) \nabla I(p, d^{t-1}, t-1)}{\|\nabla I(p - d^{t-1}, t-1)\|^2}$$

8.2.1.7 Algoritmo de Cafforio y Rocca, Ecuación 4.29

$$d^t = d^{t-1} - \frac{DFD(p, d^{t-1}) \nabla I(p, d^{t-1}, t-1)}{\sigma^2 + \|\nabla I(p - d^{t-1}, t-1)\|^2}$$

8.2.1.8 Algoritmo plano, sin término de actualización.

$$d^t = d^0$$

8.2.2 Resultados de las simulaciones

Se llevaron a cabo simulaciones utilizando la secuencia Interview. Primeramente se hizo un análisis con los algoritmos de Netravali y Robbins, para encontrar el valor de ganancia óptimo de cada algoritmo para dicha secuencia. Posteriormente haciendo uso de las ganancias citadas anteriormente, se hizo la estimación de movimiento para calcular el campo de movimiento aparente entre escenas, obteniéndose con este y la escena anterior, una reconstrucción de la escena actual, como

$$I'(p,t) = I(p - \hat{d}, t - 1) \quad \dots 8.6$$

La señal de error se calcula restando $I(p,t) - I'(p,t)$

Para los métodos de estructura piramidal, el campo vectorial de desplazamiento inicial es filtrado y submuestreado, hasta alcanzar las dimensiones del nivel más alto de la estructura piramidal. La estimación se lleva a cabo de nivel a nivel de manera descendente, utilizando como campo de vectores de desplazamiento inicial el obtenido en el nivel anterior.

Del mismo modo, se hizo un análisis para los algoritmos de Walker y Rao, de Cafforio y Rocca, y el método plano propuesto, para tener una base en comparaciones siguientes.

En estas y las simulaciones posteriores se utilizaron los siguientes parámetros: el umbral para la DFD es $T=2$, el término de corrección máximo es dos, y el mínimo es $1/16$, el límite en el número de iteraciones para la actualización del vector d es 5 para los métodos de Walker y Rao y el de Cafforio y Rocca, y 15 para los de Netravali y Robbins. Las imágenes de la secuencia a las que hacen referencia las tablas del presente capítulo y capítulos subsiguientes son las escenas 6 y 7. El desplazamiento máximo permitido es 10.

La tabla 8.1 muestra el desempeño de los algoritmos de Netravali y Robbins con ganancias óptimas. Se observa que el método de la ecuación 4.19 tiene un mejor desempeño, en términos de SNR de la imagen reconstruida, a un costo computacional muy alto. Sin embargo los demás métodos, con excepción del basado en la ecuación 4.14, se desempeñan de manera aceptable, con solo un dB de diferencia y la mitad de costo computacional.

Algoritmos de Netravali y Robbins, con ganancia óptima					
Parámetros	T=2, Iteraciones máximas=15, usando las tramas 6 y 7				
Algoritmo	Ecuación 4.7	Ecuación 4.9	Ecuación 4.14	Ecuación 4.19	Ecuación 4.25
Ganancia óptima	0.00972	0.35	0.000108	0.3	0.000972
% de pels no compensados	28.1	28.2	27.9	16.91	36.7
EQM	15.03	16.32	19.06	12.95	14.6
SNR	36.4	36	35.8	37	36.5
Entropía de la señal de error	3.58	3.61	3.59	3.24	3.56
Tiempo de Cómputo	12.8	15.07	28.4	30.2	21.2

Tabla 8.1.

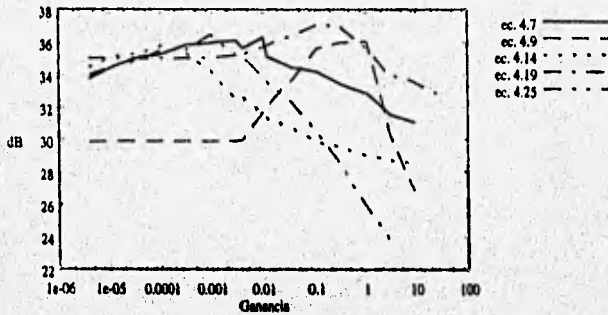


Figura 8.1a. Métodos de Netravali y Robbins, relación señal a ruido

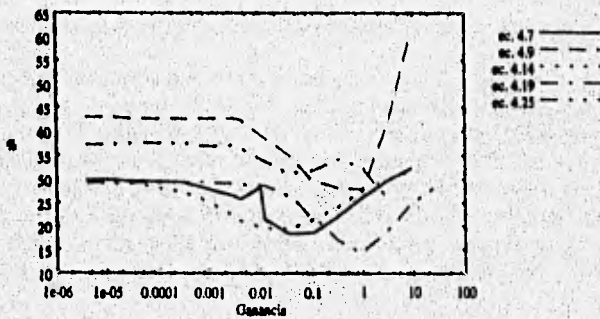


Figura 8.1b. Métodos de Netravali y Robbins, porcentaje de no convergencia

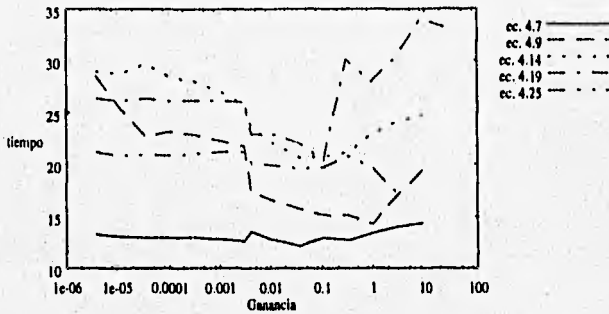


Figura 8.1c. Métodos de Netravali y Robbins, tiempo de cómputo

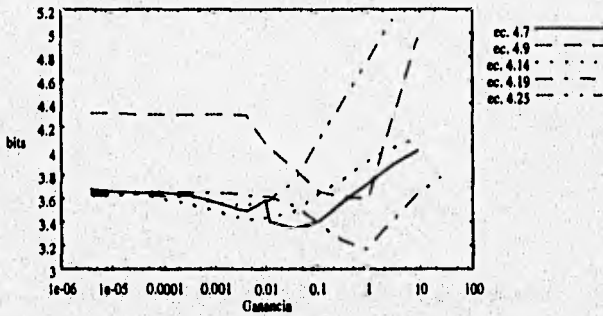


Figura 8.1d. Métodos de Netravali y Robbins, entropía de la señal de salida

La tabla 8.2 muestra el desempeño de los algoritmos de Netravali y Robbins, con ganancias óptimas, en las que se toma un campo de desplazamientos iniciales propagado temporalmente. Se observa que la relación señal a ruido de los 5 métodos ha mejorado con respecto al método correspondiente sin propagación temporal, en ocasiones con un sustancial incremento en la calidad de la imagen reconstruida de hasta casi 3 dB y una disminución del tiempo de cómputo. Se sigue notando un gran consumo de tiempo de cómputo para los algoritmos 5.10 y 5.12 en relación con el resto de ellos para una calidad similar en la imagen reconstruida.

Algoritmos de Netravali y Robbins, con vector inicial propagado temporalmente					
Parámetros	T=2, Iteraciones máximas=15, usando las tramas 6 y 7				
Algoritmo	Ecuación 4.7	Ecuación 4.9	Ecuación 4.14	Ecuación 4.19	Ecuación 4.25
Ganancia óptima	0.00972	0.35	0.000108	0.3	0.000972
% de pels no compensados	20.8	20.6	21.3	13.34	30.7
EQM	10.04	12.64	12.36	8.98	11.67
SNR	38.1	37.1	37.21	38.6	37.5
Entropía de la señal de error	3.22	3.25	3.25	2.98	3.24
Tiempo de Cómputo	12.7	18.21	35.3	26.5	18.03

Tabla 8.2

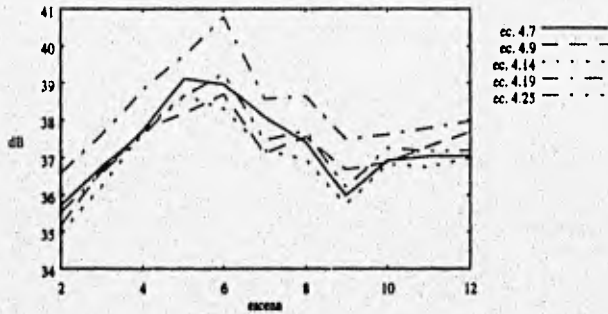


Figura 8.2a. Métodos de Netravali y Robbins con propagación temporal, relación señal a ruido.

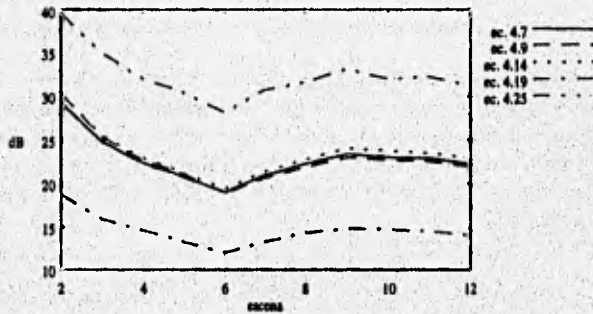


Figura 8.2b. Métodos de Netravali y Robbins, propagación temporal, porcentaje de no convergencia

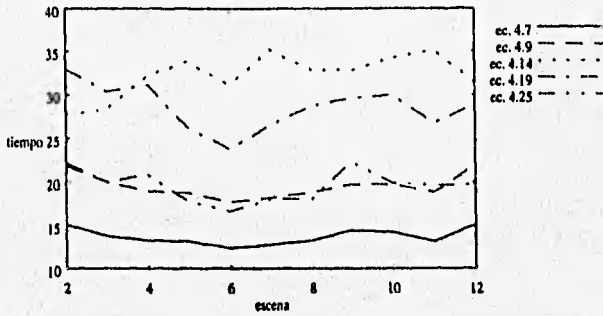


Figura 8.2c. Métodos de Netravalli y Robbins con propagación temporal, tiempo de cómputo

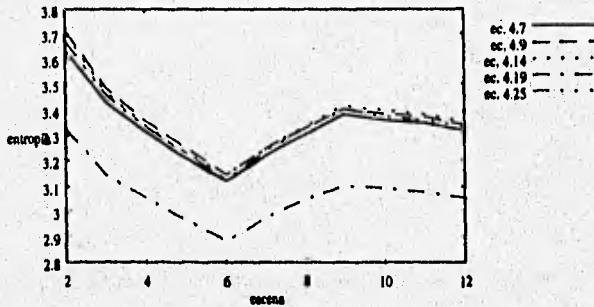


Figura 8.2d. Métodos de Netravalli y Robbins con propagación temporal, entropía de salida

En la tabla 8.3 se aprecia el comportamiento con los algoritmos de Walker y Rao y los de Cafforio y Rocca con $\sigma^2=100$. Se observa que el desempeño de los tres métodos es muy similar, al mismo tiempo se observa una pequeña ganancia en la relación señal a ruido del método de Walker y Rao con estructura piramidal de tres niveles, con un consumo de recursos de cómputo de 2.5 veces más que el método convencional.

Walker y Rao, Cafforio y Rocca			
T=2, Iteraciones máximas = 5, $\sigma^2=100$			
Parámetros	Ec. 4.28	Ec. 4.28 piramidal	Ec. 4.29
% de píels no compensados	22.0	19.6	28.2
EQM	12.44	10.18	13.412
SNR	37.2	38	36.9
Entropía de error	3.3	3.18	3.5
tiempo de cómputo	9.94	24.5	9.28

Tabla 8.3

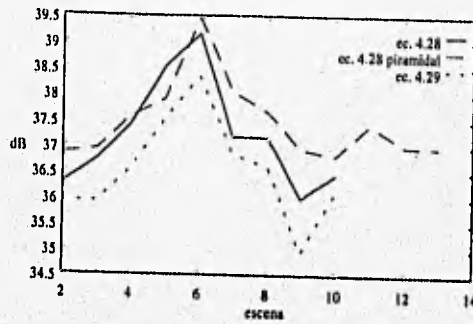


Figura 8-3a. Métodos de Walker y Rao y Cafforio y Rocca, relación señal a ruido

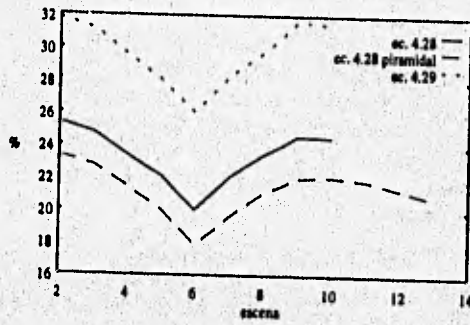


Figura 8-3b. Métodos de Walker y Rao y Cafforio y Rocca, porcentaje de píxels que no convergen

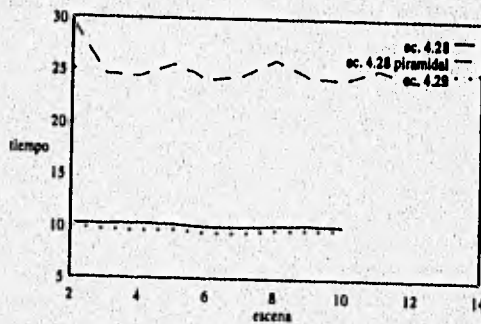


Figura 8-3c. Métodos de Walker Rao y Cafforio y Rocca, tiempo de cómputo

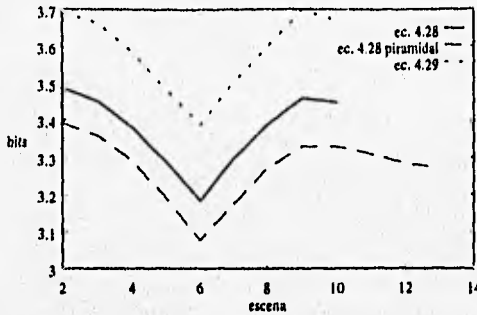


Figura 8-3d. Métodos de Walker y Rao y Cafforio y Rocca, entropía de salida

En la tabla 8.4. se muestran los análisis resultantes de los métodos de Walker y Rao con propagación temporal, con y sin estructura piramidal. Cabe hacer notar que la segunda columna corresponde a la propagación ponderada temporal de dos tramas, la primera, directamente pasada a la actual con un peso de 2/3 y la segunda, dos tramas anteriores a la actual con un peso de 1/3. En este caso no se observa ninguna mejora con respecto a la estructura piramidal y a la propagación temporal extendida. Sin embargo, si hay mejora con respecto a los casos en los que no se usa propagación temporal. En cualquier caso, los métodos de Walker y Rao tienen mejor rendimiento que los de Netravali y Robbins en cualquiera de sus variantes.

Walker y Rao con propagación temporal			
Parámetros	T=2, iteraciones maximas=5		
	Ec. 4.28	Ec. 4.28, 2 escenas	Ec. 4.28 piramidal
% de pels no compensados	18.59	18.97	20.8
EQM	7.55	9.7	10.6
SNR	39.4	38.3	37.9
Entropía de la señal de error	3.12	3.13	3.26
Tiempo de cómputo	7.04	9.94	31.3

Tabla 8-4

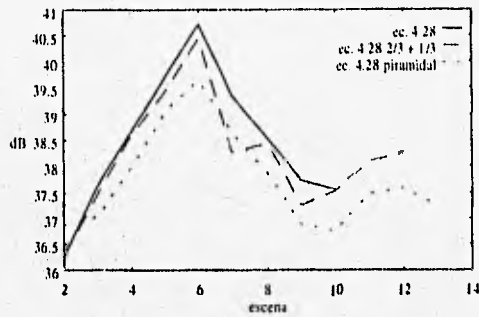


Figura 8-4a. Métodos de Walker y Rao con propagación temporal, relación señal a ruido.

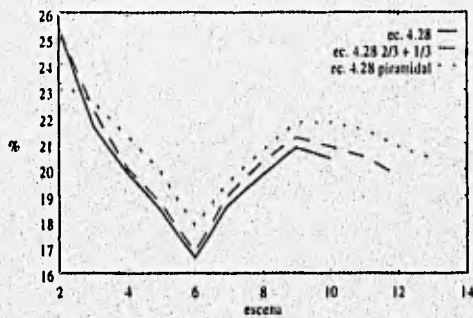


Figura 8-4b. Métodos de Walker y Rao con propagación temporal, porcentaje de pels que no convergen.

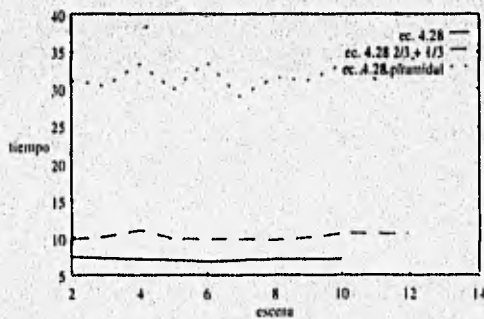


Figura 8-4c. Métodos de Walker y Rao con propagación temporal, tiempo de cómputo

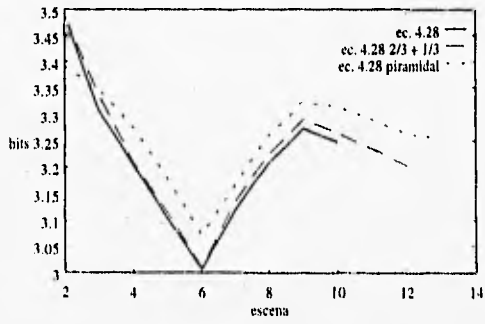


Figura 8-4d. Métodos de Walker y Rao con propugación temporal, entropía de la señal de salida.



Imagen 8.1. Campo de vectores de desplazamiento con el método de Walker y Rao, sin propugación temporal.

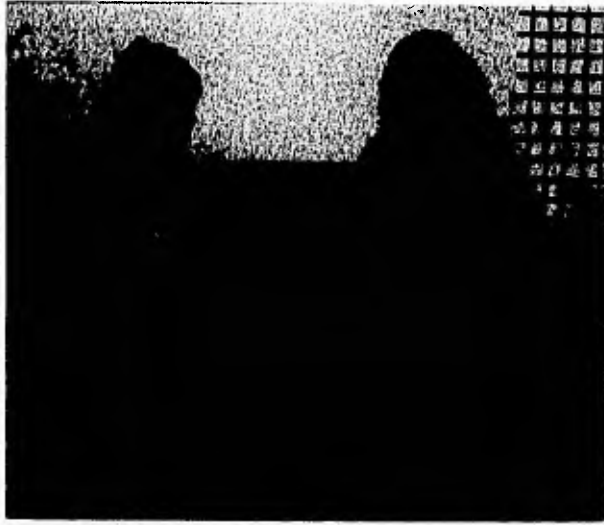


Imagen 8.2. Imagen reconstruida con el método de Walker y Rao, sin propagación temporal.

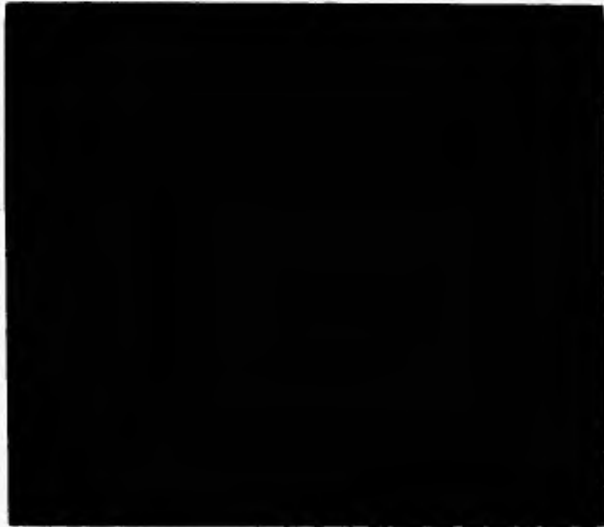


Imagen 8.3. Diferencia normalizada de la imagen original y la reconstruida con el método de Walker y Rao, sin propagación temporal.

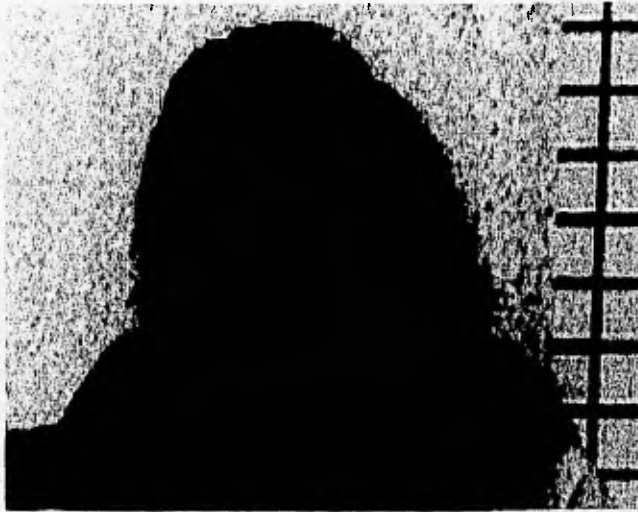


Imagen 8.4. Acercamiento de la imagen reconstruida en una zona con mucho movimiento.

8.3 Estructura general de los algoritmos de apareamiento de bloques

La simplicidad de los algoritmos de apareamiento de bloques es una de sus grandes ventajas, siendo implantados en este trabajo dos de los más representativos, el método de los tres pasos y el método de búsqueda cruzada.

8.3.1 Criterios de máxima similitud

Los criterios implantados para proporcionar una medida de similitud entre bloques de imágenes diferentes son el promedio de la diferencia absoluta, MAD, el error cuadrático medio, EQM, y el criterio de máxima cuenta de pel apareado, MPC, definidos ya en el capítulo cuatro.

8.3.2 Método de los tres pasos

Aunque originalmente el número de pasos es igual a tres, este algoritmo brinda la posibilidad de modificar este parámetro y así juzgar su desempeño con

diferentes valores. El valor de la distancia máxima de búsqueda l , se obtiene como 2^n , donde n es el número de pasos.

El algoritmo es el siguiente, para cada bloque en que se subdivide la escena actual:

Paso 1.

Se identifican en las direcciones N, S, W, E, NE, NW, SE, SW los puntos a una distancia $l/2$ del punto centrado en el bloque de la escena actual en proceso y también a una distancia cero.

Paso 2.

Se aplica el criterio de máxima similitud elegido, comparando un bloque de la escena anterior centrado en cada uno de los puntos identificados anteriormente, con el bloque de la escena actual en proceso.

Paso 3.

Se escoge el punto donde el criterio de máxima similitud obtuvo un mejor resultado, y a partir de ese punto, se repite el paso uno, disminuyendo la distancia l a la mitad, hasta que $l=1$.

Paso 4.

El vector de desplazamiento asignado a cada uno de los pels del bloque en cuestión, se obtiene restando el último punto elegido en el paso tres, del punto donde se centra el bloque actual.

8.3.3 Método de búsqueda cruzada

Como se vió en capítulos anteriores, este método tiene variantes en cuanto a los puntos elegidos para la búsqueda del bloque de máxima similitud. El presente trabajo implanta la variante de búsqueda en forma del signo "+", es decir toma los puntos horizontales y verticales al punto donde se centra el bloque en proceso.

El algoritmo es el siguiente, para cada bloque en que se subdivide la escena actual:

Paso 1.

Se identifican en las direcciones N, S, W, E, los puntos a una distancia $l/2$ del punto centrado en el bloque de la escena actual en proceso y también a una distancia cero.

Paso 2.

Se aplica el criterio de máxima similitud elegido, comparando un bloque de la escena anterior centrado en cada uno de los puntos identificados anteriormente, con el bloque de la escena actual en proceso.

Paso 3.

Se escoge el punto donde el criterio de máxima similitud obtuvo un mejor resultado y a partir de ese punto, se repite el paso uno. Si el punto elegido es igual al punto donde se centra el bloque actual en proceso, o es igual al obtenido del paso tres de la iteración anterior, la distancia l se reduce a la mitad, en caso contrario, l no se modifica. El proceso se repite hasta que $l=1$.

Paso 4.

El vector de desplazamiento asignado a cada uno de los pels del bloque en cuestión, se obtiene restando el último punto elegido en el paso tres, del punto donde se centra el bloque actual.

Debe aclararse que para este método, se ha definido una ventana de búsqueda fuera de la cual no es posible buscar un desplazamiento para el bloque en proceso, en caso de que el algoritmo llegara, o incluso intentara sobrepasar dicha ventana, se asigna al punto más cercano a la frontera, decrementando l a la mitad para la siguiente iteración y buscando solamente en los puntos que queden dentro de la ventana y/o sobre la frontera de esta, repitiéndose el algoritmo hasta que $l = 1$.

8.3.4 Resultados de las simulaciones

Se llevaron a cabo simulaciones utilizando la secuencia "Interview" con el algoritmo de los tres pasos, variando el tamaño del bloque y el número de pasos para observar su desempeño. De igual forma, el análisis del desempeño del método de búsqueda cruzada se realizó variando la distancia de búsqueda, el ancho del bloque y la amplitud de la ventana de búsqueda. Las imágenes a las que hacen referencia las tablas siguientes son las escenas 6 y 7 de la secuencia en cuestión.

La tabla 8.5 muestra el comportamiento de este algoritmo al variar el número de pasos, manteniendo constante el ancho de bloque. Se observa que el mejor desempeño se obtiene cuando el número de pasos es igual a dos.

Apareamiento de bloques, método de los tres pasos, tamaño del bloque=8				
Parámetros	pasos=4	pasos=3	pasos=2	pasos=1
EQM	45.7	35.1	27.6	41.8
SNR	31.5	32.7	33.7	31.9
Entropía de la señal de error	4.37	4.26	4.17	4.29
Tiempo de Cómputo	14.05	10.64	7.27	3.97

Tabla 8.5

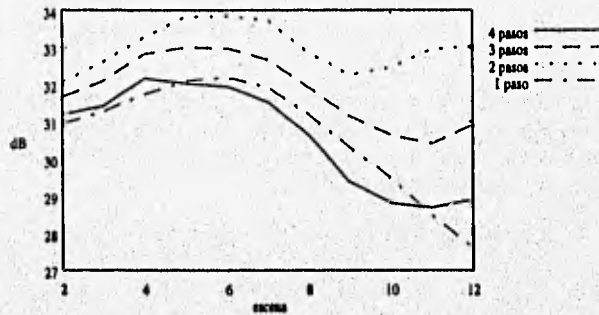


Figura 8.5a. Métodos de apareamiento de bloques, algoritmo de búsqueda de los tres pasos, relación señal a ruido

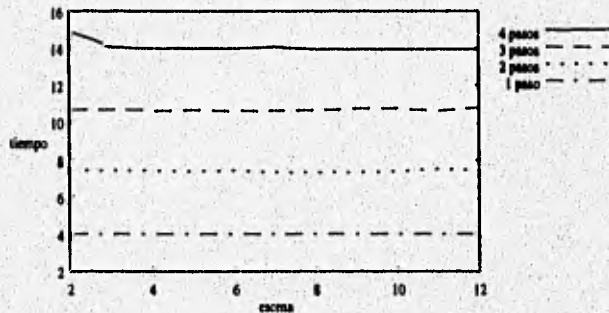


Figura 8.5b. Métodos de apareamiento de bloques, algoritmo de búsqueda de los tres pasos, tiempo de cómputo

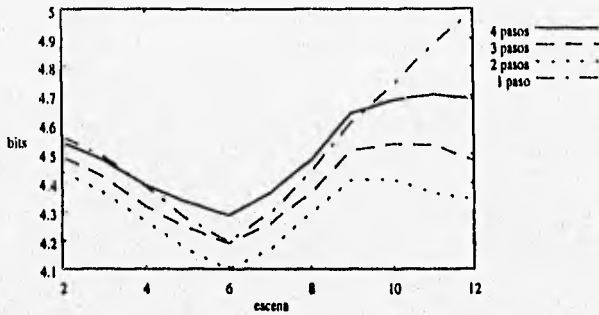


Figura 8.5c. Métodos de apareamiento de bloques, algoritmo de búsqueda de los tres pasos, entropía de la señal de salida

En la tabla 8.6 se ha variado el tamaño del bloque para el método de los tres pasos, se observa que entre más pequeño sea el tamaño del bloque, se obtienen mejores resultados en cuanto a la relación señal a ruido de la imagen reconstruida, a cambio de un mayor tiempo de cómputo.

Apareamiento de bloques, método de los tres pasos, pasos=3				
Parámetros	bloque=16	bloque=8	bloque=4	bloque=2
EOM	37.8	35.1	37	24.9
SNR	32.4	32.7	32.4	34.2
Entropía de la señal de error	4.27	4.26	4.23	3.77
Tiempo de Cómputo	10.77	10.63	11.03	12.07

Tabla 8.6

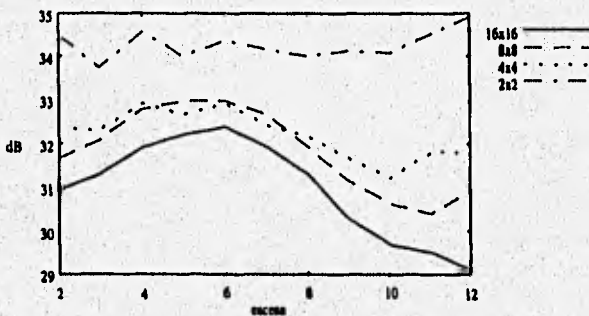


Figura 8.6a. Métodos de apareamiento de bloques, algoritmo de búsqueda de los tres pasos, relación señal a ruido

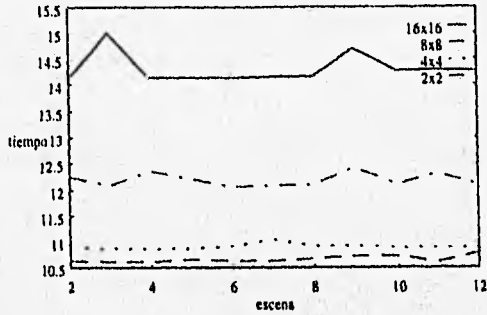


Figura 8.6b. Métodos de apareamiento de bloques, algoritmo de búsqueda de los tres pasos, tiempo de cómputo

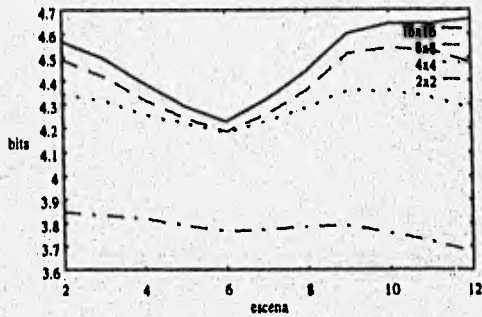


Figura 8.6c. Métodos de apareamiento de bloques, algoritmo de búsqueda de los tres pasos, entropía de la señal de salida.

En la tabla 8.7 se muestran los resultados para diferentes valores de distancia; usados en el algoritmo de búsqueda cruzada sobre la secuencia "Interview", imágenes 6 a 7. Se observa que los parámetros se mantienen constantes desde una distancia 2 hasta 8, lo único que varía es el tiempo de cómputo. En la tabla 8.8 se ha variado el tamaño del bloque con una distancia fija, igual a cuatro. Los mejores resultados en cuanto a SNR y EQM se observan cuando el bloque es más pequeño.

Apareamiento de bloques, método de búsqueda cruzada, tamaño del bloque=8				
Parámetros	distancia=8	distancia=4	distancia=2	distancia=1
EQM	55.4	55.4	55.4	203
SNR	30.7	30.7	30.7	25.1
Entropía de la señal de error	4.4	4.4	4.4	5.36
Tiempo de Cómputo	3.49	3.83	3.31	4.33

Tabla 8.7 Evaluación de distancia con el método de búsqueda cruzada

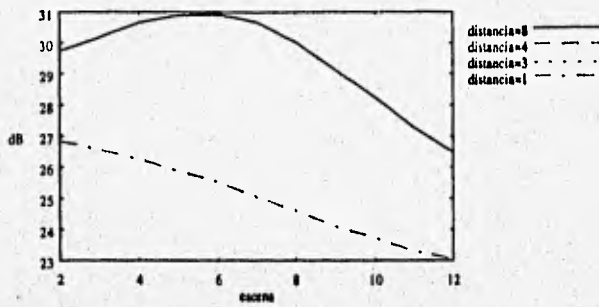


Figura 8.7a. Métodos de apareamiento de bloques, algoritmo de búsqueda cruzada, relación señal a ruido.

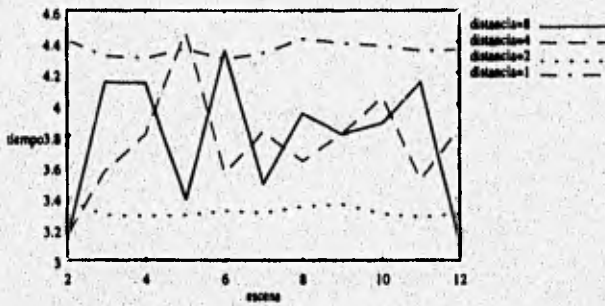


Figura 8.7b. Métodos de apareamiento de bloques, algoritmo de búsqueda cruzada, tiempo de cómputo

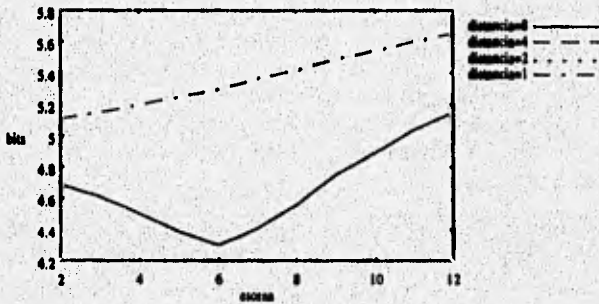


Figura 8.7c. Métodos de apareamiento de bloques, algoritmo de búsqueda cruzada, entropía de la señal de salida.

Apareamiento de bloques, método de búsqueda cruzada, distancia=4			
Parámetros	tamaño=2	tamaño=4	tamaño=8
EQM	45.2	51.7	55.4
SNR	31.6	31	30.7
Entropía de la señal de error	4.14	4.33	4.4
Tiempo de Cómputo	2.54	4.5	3.31

Tabla 8.8 Evaluación del tamaño del bloque con el método de búsqueda cruzada

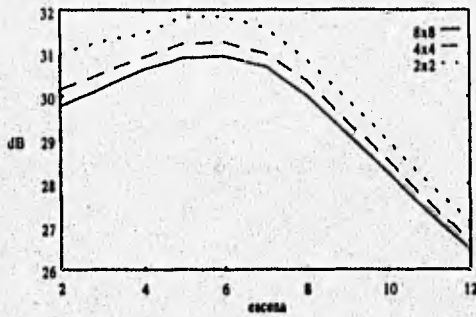


Figura 8.8a. Métodos de apareamiento de bloques, algoritmo de búsqueda cruzada, relación señal a ruido.

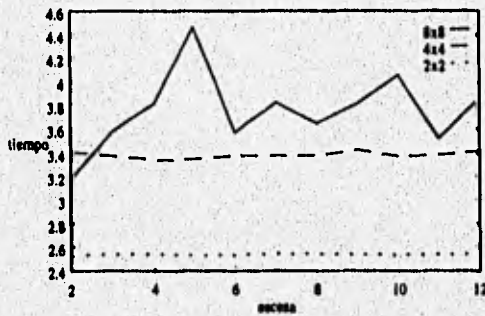


Figura 8.8b. Métodos de apareamiento de bloques, algoritmo de búsqueda cruzada, tiempo de cómputo.

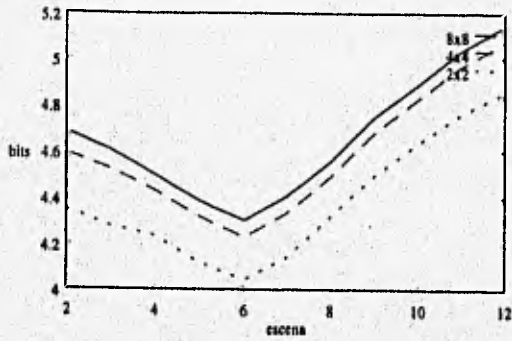


Figura 8.8c. Métodos de apareamiento de bloques, algoritmo de búsqueda cruzada, entropía de la señal de salida.



Imagen 8.5. Campo de vectores de desplazamiento con el método de apareamiento de bloques de los tres pasos, para un tamaño de bloque de 8×8 , con propagación temporal.



Imagen 8.6. Imagen reconstruida con el método de apareamiento de bloques de los tres pasos, para un tamaño de bloque de 8×8 , con propagación temporal.

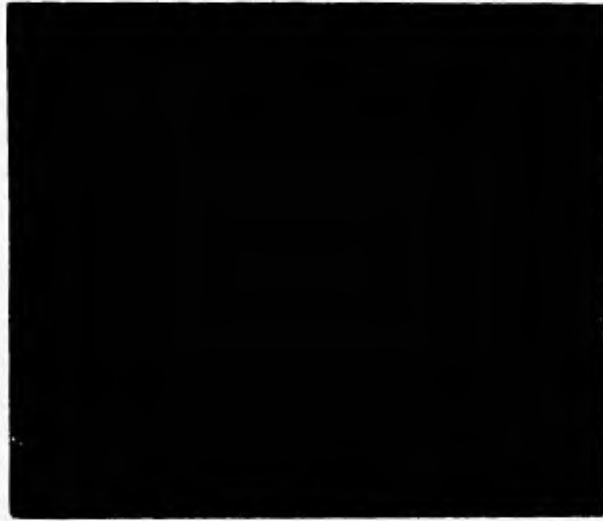


Figura 8.7. Diferencia normalizada de la imagen original y la reconstruida con el método de apareamiento de bloques de los tres pasos, para un tamaño de bloque de 8×8 , con propagación temporal.



Figura 8.8. Acercamiento de la imagen reconstruida en una zona con mucho movimiento con el método de apareamiento de bloques de los tres pasos, para un tamaño de bloque de 8×8 , con propagación temporal.

8.4 Métodos propuesto por los autores

8.4.1 Método mixto

Combinando las características de los dos grandes grupos de estimación de movimiento, se implantó un algoritmo que aprovecha las ventajas que cada uno de ellos ofrece. Este algoritmo al que denominamos mixto, se basa en obtener un campo inicial de vectores de desplazamiento utilizando un método de apareamiento de bloques, dado que en general estos métodos no brindan una muy buena estimación, pero por la simplicidad de sus algoritmos hacen que su tiempo de proceso sea muy breve. Con el campo vectorial de inicio así obtenido, se hace un ajuste con algún método pel recursivo, el cual como podrá elegir un buen vector inicial, hará menos iteraciones y obtendrá un mejor vector de desplazamiento final.

El algoritmo que utiliza la búsqueda de los tres pasos y la estimación de Walker y Rao es el siguiente:

Paso 1

Se identifican en las direcciones N, S, W, E, NE, NW, SE, SW los puntos a una distancia $l/2$ del punto centrado en el bloque de la escena actual en proceso y también a una distancia cero.

Paso 2.

Se aplica el criterio de máxima similitud elegido, comparando un bloque de la escena anterior centrado en cada uno de los puntos identificados anteriormente, con el bloque de la escena actual en proceso.

Paso 3.

Se escoge el punto donde el criterio de máxima similitud obtuvo un mejor resultado y a partir de ese punto, se repite el paso uno, disminuyendo la distancia l a la mitad, hasta que $l=1$.

Paso 4.

El vector de desplazamiento asignado a cada uno de los pels del bloque en cuestión, se obtiene restando el último punto elegido en el paso tres, del punto donde se centra el bloque actual.

Paso 5.

Se repite el paso uno, con el siguiente bloque de la escena, hasta finalizar la imagen.

Paso 6.

Seleccionar el valor inicial v_i para un pel p , $d_0 = v_i$. El valor del vector inicial v_i es el que cumple con la menor DFD de entre: los valores del vector de desplazamiento asociado a los pels vecinos ya calculados, el vector nulo y el vector de desplazamiento correspondiente al pel p , calculado en el paso 4.

Paso 7.

Si $|DFD(p, v_i)| \leq T$ entonces $d = v_i$ y se procede con el siguiente pel. En caso contrario se pasa al cálculo iterativo.

Paso 8.

Se calcula el factor de corrección TC obtenido con el método de Walker y Rao de la ecuación 4.28 y se restringe a un cierto rango, con la finalidad de evitar la divergencia del algoritmo.

Se actualiza el vector de desplazamiento $d_i = d_{i-1} - TC$ hasta que el valor de $|DFD(p, d_i)| \leq T$ o bien i alcance el valor máximo permitido de iteraciones.

Paso 9.

Si el valor del vector de desplazamiento obtenido cumple con $|DFD(p, d_i)| \leq T$, entonces $d = d_i$. En caso contrario, si se alcanza el límite en el número permitido de iteraciones, d se elige entre v_i , d_i y un vector nulo, tal que la DFD sea mínima.

Si el vector de desplazamiento es mayor que un desplazamiento máximo determinado a priori, entonces el vector se nulifica, debido a que es muy probable que el método haya divergido para ese pel en particular.

Paso 10.

Se procede con el siguiente pel de la imagen hasta haber procesado todos los pels de la misma.

8.4.2 Método plano

Como una forma de evaluar el desempeño real de los algoritmos estudiados en este trabajo, se elaboró un método sencillo que no realiza estimación, únicamente escoge el vector de desplazamiento entre los pels vecinos

Algoritmo:

Paso 1:

Para el pel actual p se elige el vector de desplazamiento inicial seleccionado entre un vector propagado temporalmente, un vector nulo y un vector cuyas componentes tienen magnitud "0" o "1".

Paso 2:

Se continúa con el siguiente pel, siguiendo un barrido estándar.

8.4.3 Resultados de las simulaciones

Se hicieron simulaciones sobre la secuencia "Interview" con el método mixto y el plano, optándose por implantar el método de los tres pasos combinado con el de Walker y Rao. Se fijó el tamaño del bloque igual a dos, el número de pasos igual a tres y el número máximo de iteraciones igual a cinco.

La tabla 8.9 nos ofrece la oportunidad de comparar los resultados de los métodos de mejor desempeño de Netravali y Robbins y el de Walker y Rao, el método mixto y el plano. Es posible observar que la calidad de la señal reconstruida en el método mixto es superior a la de cualquiera de los otros métodos y el tiempo de procesamiento no es mayor al del método de Netravali y Robbins al que supera por casi 2.3 dB en cuanto a la relación señal a ruido.

Algoritmo	Comparación de los métodos			
	ec. 4.28	ec. 4.19	mixto	plano
	T=2, lmax=5	T=2, lmax=15	T=2, lmax=5,	T=2
% de pels no compensados	18.59	13.34	12.65	---
EQM	7.55	8.98	5.23	41.6
SNR	39.4	38.6	40.9	31.9
Entropía de error	3.12	2.98	2.83	3.5
Tiempo de cómputo	7.04	26.5	25.6	3.17

Tabla 8-9

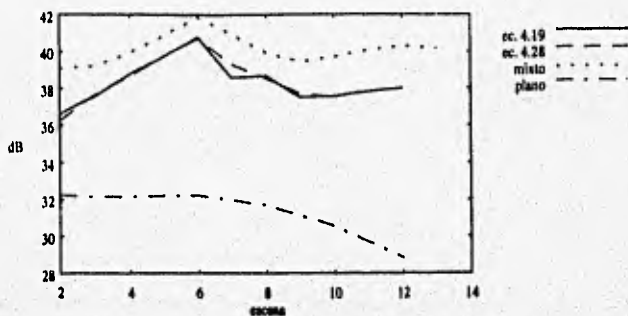


Figura 8.9a. Método mixto, relación señal a ruido, comparación con los otros métodos.

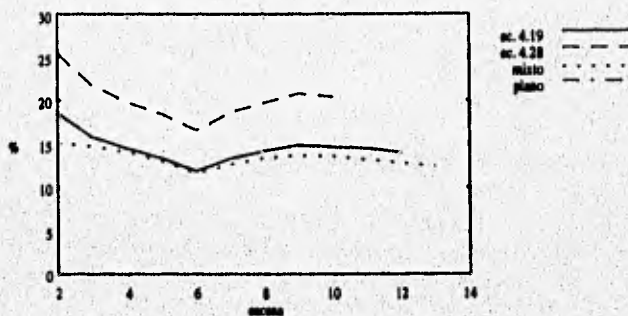


Figura 8.9b. Método mixto, porcentaje de píxeles que no convergen, comparación con los otros métodos.

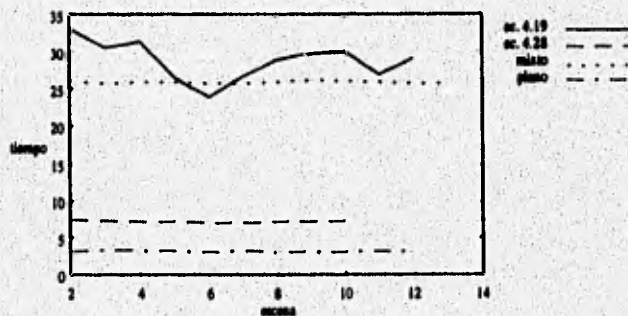


Figura 8.9c. Método mixto, tiempo de cómputo, comparación con los otros métodos.

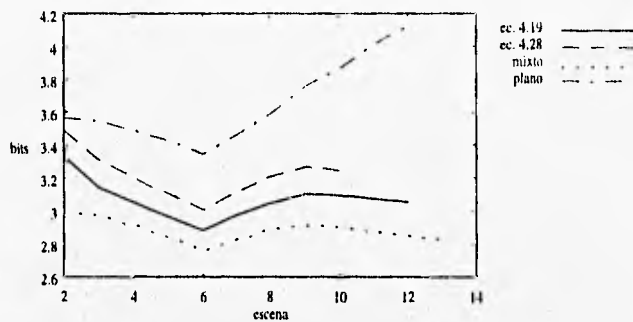


Figura 8.9d. Método mixto, entropía de la señal de salida, comparación con los otros métodos.



Imagen 8.9. Campo de vectores de desplazamiento con el método "mixto".



Imagen 8.10. Imagen reconstruida con el método "mixto".



Imagen 8.11. Diferencia normalizada de la imagen original y la reconstruida con el método "mixto".



Imagen 8.12. Acercamiento de la imagen reconstruida en una zona con mucho movimiento con el método "mixto".

8.5 Estructura general para los algoritmos de compensación de movimiento

La simulación en la codificación de imágenes sobre la secuencia "Interview", se hace primero con la implantación de un codificador - decodificador como el que especifican Walker y Rao, basado en un lazo DPCM y el cual sólo utiliza una predicción espacial para el valor de desplazamiento. Este es modificado en la parte de la estimación para introducir los algoritmos de Netravali y Robbins y así observar su comportamiento con otros estimadores. Posteriormente es propuesto un codificador con la misma filosofía, pero que tiene dos tipos de predicciones, una temporal y otra espacial, donde se introduce además el algoritmo propuesto por los autores en la parte de predicción temporal.

8.5.1 Codificador de Walker y Rao para compensación de movimiento

En este algoritmo la DFD utiliza la imagen actual a codificar y la imagen anterior reconstruida, excepto al inicio de la secuencia donde utiliza las dos primeras imágenes originales.

Algoritmo para el codificador:

Paso 1.

Para el punto actual, se establece un vector de desplazamiento igual al asignado al pel vecino anteriormente procesado.

Paso2.

Si el valor absoluto de la DFD valuada con el vector de desplazamiento así obtenido es menor o igual a un umbral preestablecido, el valor transmitido del error se hace igual a cero y se pasa al paso 4. En caso contrario se valúa la DFD con un vector de desplazamiento nulo, si el nuevo valor de la DFD está dentro del rango del umbral preestablecido se transmite RESET y se asigna un vector nulo pasando al paso 4.

Paso 3.

Se transmite el valor de la DFD valuada con el vector obtenido en el paso 1.

Paso 4.

Se reconstruye la imagen actual, utilizando el vector de desplazamiento obtenido en el paso 1 cuando la transmisión es diferente de RESET, sumando el valor de error transmitido. Si se transmite RESET se utiliza un vector nulo para la reconstrucción. Si el valor del error transmitido es igual a RESET o cero, se regresa al paso 1 para procesar el siguiente pel.

Paso 5.

Se procede a la actualización del vector obtenido en el paso 1, usando algoritmos pel recursivos y se regresa al paso 1.

Algoritmo para el decodificador

Paso 1.

Para el punto actual, se establece un vector de desplazamiento igual al asignado al pel vecino anteriormente procesado.

Paso2.

Si el valor del error recibido es igual a RESET, el vector de desplazamiento actual se iguala a cero y el pel para la imagen actual se hace igual al de la imagen anterior regresando al paso 1. En caso contrario la imagen actual es reconstruida

con el vector obtenido en el paso 1, sumándole la señal de error recibida. Si dicho error es cero, se procede al paso 1.

Paso 3.

Se procede a la actualización del vector obtenido en el paso 1, usando algoritmos pel recursivos y se regresa al paso 1.

8.5.2 Resultados

A continuación se muestran los resultados para el codificador de Walker y Rao sobre la secuencia "Interview" usando como método de estimación la ecuación 4.28. En la tabla 8.10 se observa como se comporta el codificador a diferentes umbrales, la entropía de la señal de error disminuye a medida que el umbral aumenta, a expensas de la relación señal a ruido. A diferencia de los métodos de estimación, la reconstrucción en este caso mantiene unos niveles elevados de relación señal a ruido a umbrales grandes. La ventaja que se observa sobre el método plano es una sustancial reducción en la entropía de la señal de error, contra un aumento del tiempo de cómputo.

Codificador de Walker y Rao						
Parámetros	T=1	T=3	T=5	T=7	T=9	Plano T=9
% de bits no compensados	38.1	14.98	7.55	4.26	2.74	---
SNR	34.5	44.6	40.2	37.7	36.0	36.0
EQM	0.231	2.27	6.22	11.15	16.58	16.16
Entropía de error	4.22	2.98	2.07	1.477	1.118	1.47
Tiempo de cómputo	31.1	21.5	18.6	14.22	12.98	7.36

Tabla 8.10

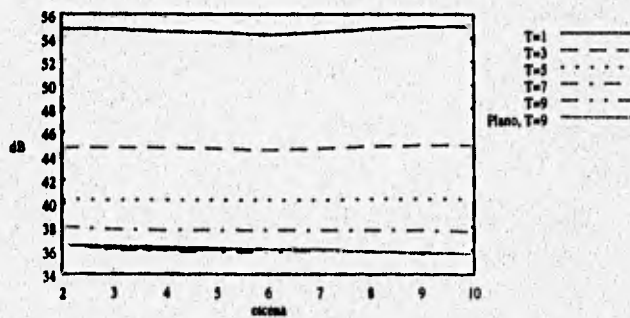


Figura 8.10a. Codificador de Walker y Rao, método de Walker y Rao y método plano, relación señal a ruido.

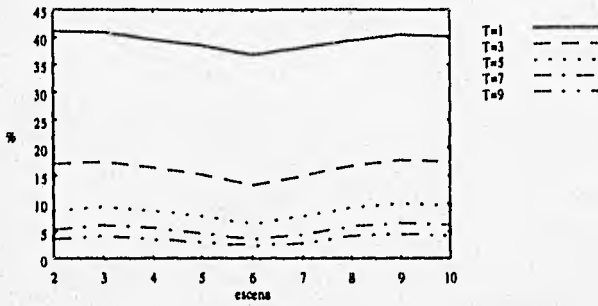


Figura 8.10b. Codificador de Walker y Rao, método de Walker y Rao y método plano, porcentaje de pels que no convergen.

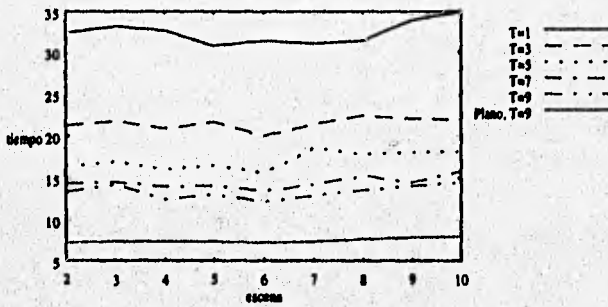


Figura 8.10c. Codificador de Walker y Rao, método de Walker y Rao y método plano, tiempo de cómputo.

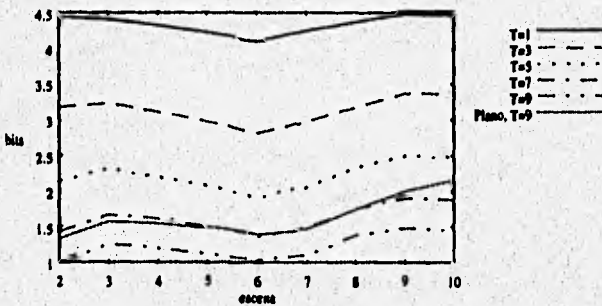


Figura 8.10d. Codificador de Walker y Rao, método de Walker y Rao y método plano, entropía de la señal transmitida.

En la tabla 8.11 se muestra el comportamiento del mismo codificador con propagación temporal del vector de desplazamiento inicial. Este se usa únicamente en los casos de actualización del vector de desplazamiento para el pel en proceso. Se observa alguna disminución en la entropía de la señal de error y en el tiempo de cómputo; manteniéndose la relación señal a ruido. Cabe hacer notar que la diferencia en tiempo del método *plano* con el de Walker y Rao, ya no es tan significativa.

Codificador de Walker y Rao, propagación temporal						
Parámetros	T=1	T=3	T=5	T=7	T=9	Plano T=9
% de pels no compensados	34.4	12.93	6.29	3.52	2.19	...
SNR	54.4	44.5	40.2	37.6	36.0	34.0
EQM	0.237	2.32	6.24	11.19	16.44	16.6
Entropía de error	4.14	2.86	1.928	1.353	1.008	1.455
Tiempo de cómputo	20.2	13.11	10.23	8.85	8.07	7.31

Tabla 8.11

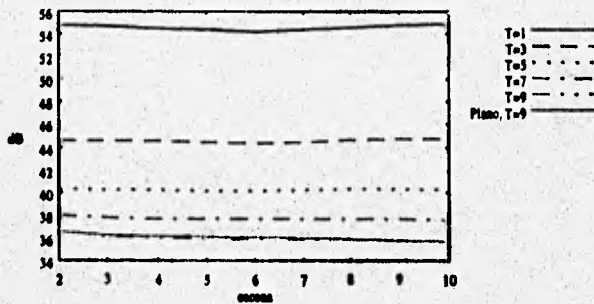


Figura 8.11a. Codificador de Walker y Rao, método de Walker y Rao y método plano con propagación temporal, relación señal a ruido.

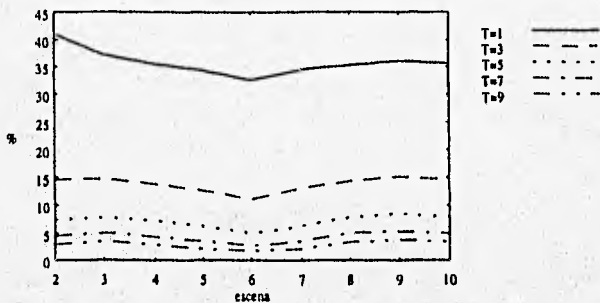


Figura 8.11b. Codificador de Walker y Rao, método de Walker y Rao y método plano con propagación temporal, porcentaje de pels que no convergen.

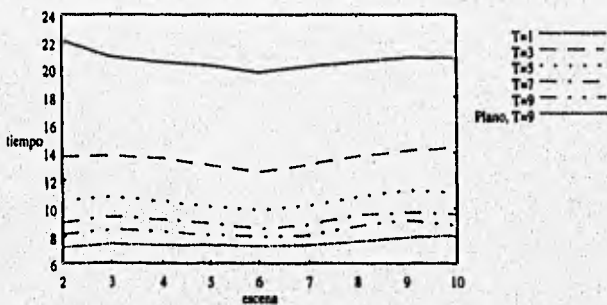


Figura 8.11c. Codificador de Walker y Rao, método de Walker y Rao y método plano con propagación temporal, tiempo de cómputo.

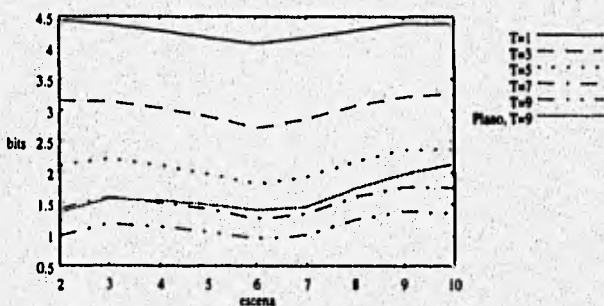


Figura 8.11d. Codificador de Walker y Rao, método de Walker y Rao y método plano con propagación temporal, entropía de la señal transmitida.

6.5.3 Codificador de Walker y Rao modificado para métodos de apareamiento de bloques

En este algoritmo de manera parecida al anterior, el criterio de máxima similitud utiliza la imagen actual a codificar y la imagen anterior reconstruida, excepto al inicio de la secuencia donde utiliza las dos primeras imágenes originales. El umbral con que se compara la función de máxima similitud, es obtenido multiplicando el número de pels del bloque por el umbral preestablecido para la equivalencia de luminancias entre pels, generalmente es aceptado un valor de 2 o 3.

Algoritmo para el codificador:

Paso 1.

Para el bloque actual, se establece un vector de desplazamiento igual al asignado al bloque vecino anteriormente procesado.

Paso 2.

Si el valor regresado por la función de máxima similitud valuada con el vector de desplazamiento así obtenido es menor o igual a un umbral preestablecido, el valor transmitido del error se hace igual a cero y se procede al paso 4. En caso contrario se valida la función de máxima similitud con un vector de desplazamiento nulo, si el nuevo valor de la función de máxima similitud está dentro del rango del umbral preestablecido se transmite RESET y se asigna un vector nulo pasando al paso 4.

Paso 3.

Se transmite el valor de la DFD para cada pel del bloque, valuada con el vector obtenido en el paso 1.

Paso 4.

Se reconstruye la imagen actual, utilizando el vector de desplazamiento obtenido en el paso 1 cuando la transmisión es diferente de RESET, sumando el valor de error transmitido a cada pel del bloque. Si se transmite RESET se utiliza un vector nulo para la reconstrucción. Si el valor del error transmitido es igual a RESET o cero, se regresa al paso 1 para procesar el siguiente bloque.

Paso 5.

Se procede a la actualización del vector obtenido en el paso 1, usando algoritmos de apareamiento de bloques y se regresa al paso 1.

Algoritmo para el decodificador

Paso 1.

Para el bloque actual, se establece un vector de desplazamiento igual al asignado al bloque vecino anteriormente procesado.

Paso 2.

Si el valor del error recibido es igual a RESET, el vector de desplazamiento actual se iguala a cero y el bloque para la imagen actual se hace igual al de la imagen anterior regresando al paso 1. En caso contrario la imagen actual es reconstruida con el vector obtenido en el paso 1, sumándole la señal de error recibida a cada pel del bloque. Si dicho error es cero, se procede al paso 1.

Paso 3.

Se procede a la actualización del vector obtenido en el paso 1, usando algoritmos de apareamiento de bloques y se regresa al paso 1.

8.5.4 Resultados

En la tabla 8.12, usando el método de los tres pasos y siendo la función de máxima similitud la de mínima diferencia absoluta, se observa que aunque la relación señal a ruido es alta con umbrales pequeños, la entropía de la señal de error resultante es muy grande en comparación con los codificadores que usan técnicas pel recursivas. No se observa ninguna mejora con respecto al uso de propagación temporal, cuyos resultados se muestran en la tabla 8.13.

Codificador de Walker y Rao modificado para apareamiento de bloques						
Parámetros	T=1	T=3	T=5	T=7	T=9	Plano T=9
SNR	60	43.1	37.9	35.2	33.4	36.0
EQM	0.0649	3.16	10.50	19.6	29.6	16.16
Entropía de error	5.33	4.48	3.29	2.41	1.887	1.47
Tiempo de cómputo	31.2	19.24	12.0	8.23	6.28	7.36

Tabla 8.12

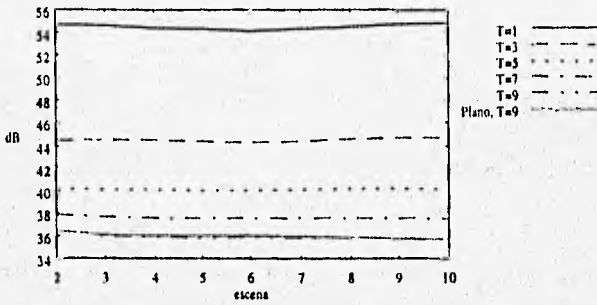


Figura 8.12a. Codificador de Walker y Rao, método de los tres pasos y método plano sin propagación temporal, relación señal a ruido.

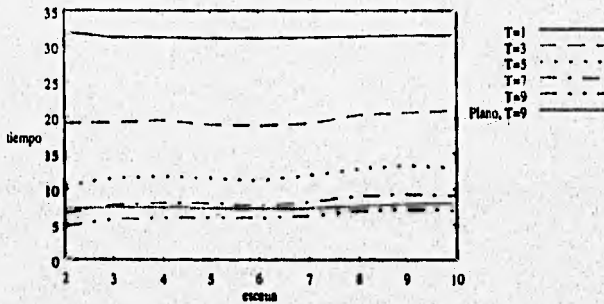


Figura 8.12b. Codificador de Walker y Rao, método de los tres pasos y método plano sin propagación temporal, tiempo de cómputo.

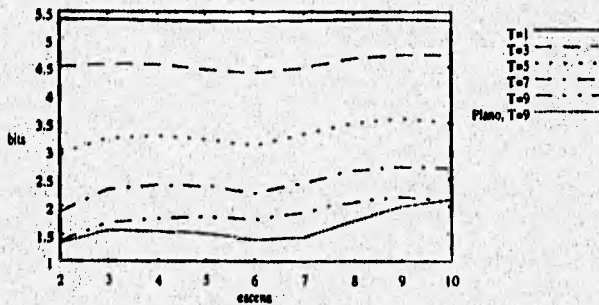


Figura 8.12c. Codificador de Walker y Rao, método de los tres pasos y método plano sin propagación temporal, entropía de la señal transmitida.

Codificador de Walker y Rao modificado para apareamiento de bloques, propagación temporal						
Parámetros	T=1	T=3	T=5	T=7	T=9	Plano T=9
SNR	60	43.1	37.9	35.2	33.4	34.0
EQM	0.0649	3.16	10.5	19.6	29.6	16.6
Entropía de error	5.33	4.48	3.29	2.41	1.887	1.455
Tiempo de cómputo	31.2	19.20	12	8.20	6.26	7.31

Tabla 8.13

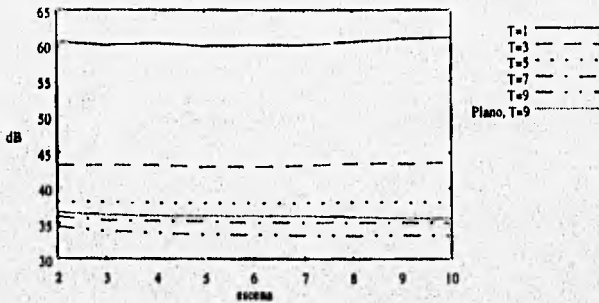


Figura 8.13a. Codificador de Walker y Rao, método de los tres pasos y método plano con propagación temporal, relación señal a ruido.

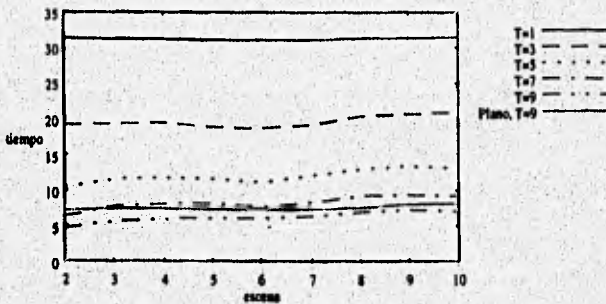


Figura 8.13b. Codificador de Walker y Rao, método de los tres pasos y método plano con propagación temporal, tiempo de cómputo.

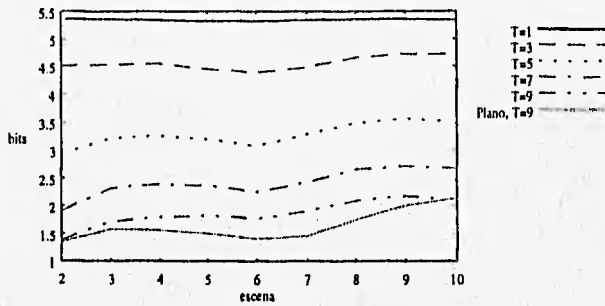


Figura 8.13c. Codificador de Walker y Rao, método de los tres pasos y método plano con propagación temporal, entropía de la señal transmitida.

8.5.5 Resultados para el codificador del método mixto

Dado que la naturaleza del método propuesto no permite incluirlo directamente en el codificador de Walker y Rao o su versión modificada, se ha implantado una nueva modificación donde el vector de desplazamiento inicial es obtenido a partir de las dos escenas anteriores a la procesada, aplicándoseles el *método mixto*. Así se eligió también como predictor el método de Walker y Rao y en la parte de apareamiento de bloques, el de los tres pasos. Los resultados se muestran en la tabla 8.14

Codificador de Walker y Rao modificado para método mixto						
Parámetros	T=1	T=3	T=5	T=7	T=9	Plano T=9
% pels no apareados	33.5	13.08	6.99	4.35	2.94	...
SNR	54.7	44.6	40.26	37.7	36.0	34.0
EQM	0.221	2.23	6.13	11.17	16.7	16.6
Entropía de error	4.36	3.17	2.27	1.692	1.31	1.455
Tiempo de cómputo	2.99	11.64	15.09	16.55	17.38	7.31

Tabla 8.14

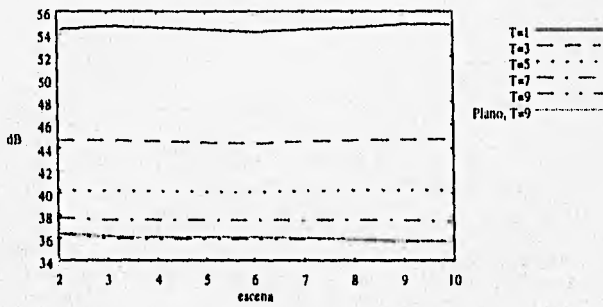


Figura 8.14a. Codificador de Walker y Rao, método mixto y método plano, relación señal a ruido.

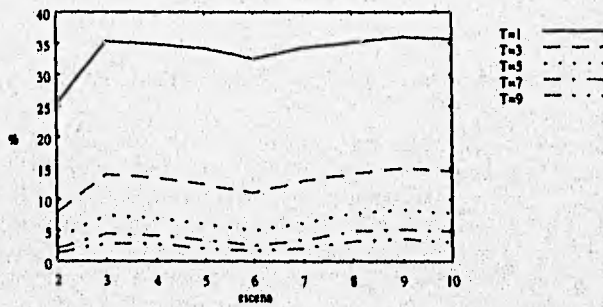


Figura 8.14b. Codificador de Walker y Rao, método mixto y método plano, porcentaje de pels que no convergen.

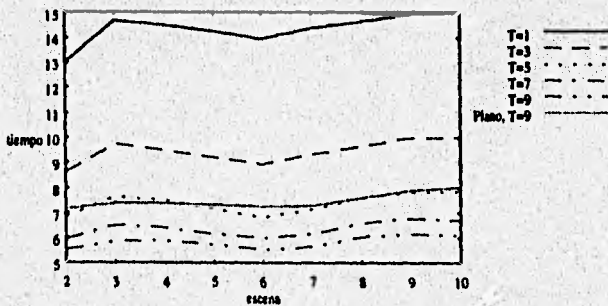


Figura 8.14c. Codificador de Walker y Rao, método mixto y método plano, tiempo de cómputo.

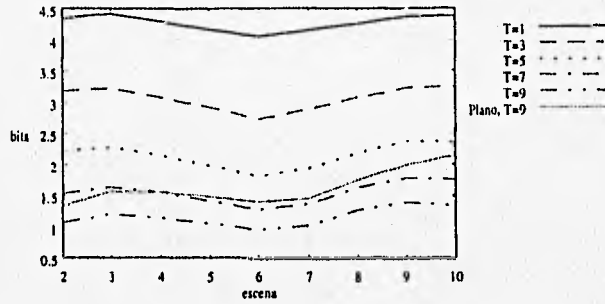


Figura 8.14d. Codificador de Walker y Rao, método mixto y método plano, entropía de la señal transmitida.

En la tabla 8.15 se muestra una comparación entre los métodos de Walker y Rao, por apareamiento de bloques, plano y mixto, para un umbral $T=9$.

Codificador de Walker y Rao, $T=9$					
Método	Walker y Rao	W y R propagación	BM	Plano	Mixto
% pefes no apareados	2.19	2.74	---	---	2.94
SNR	36.0	36.0	33.4	34.0	36.0
EQM	16.44	16.58	29.6	16.6	16.7
Entropía de error	1.008	1.118	1.887	1.455	1.31
Tiempo de cómputo	8.07	12.98	6.26	7.31	17.38

Tabla 8.15

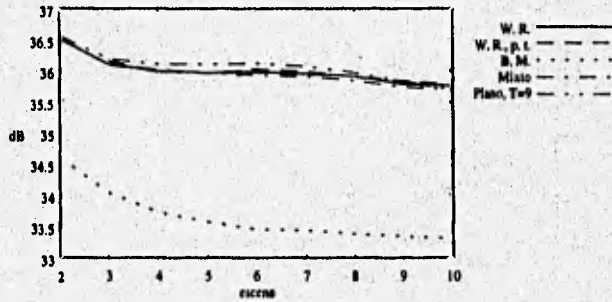


Figura 8.15a. Codificador de Walker y Rao, relación señal a ruido, $T=9$.

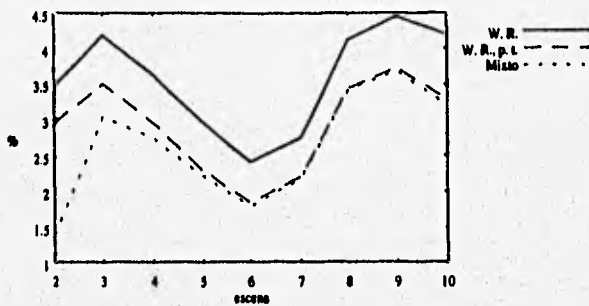


Figura 8.15b. Codificador de Walker y Rao, porcentaje de bits que no convergen, $T=9$.

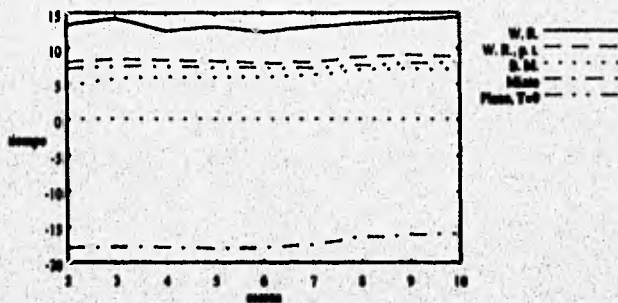


Figura 8.15c. Codificador de Walker y Rao, tiempo de cómputo, $T=9$.

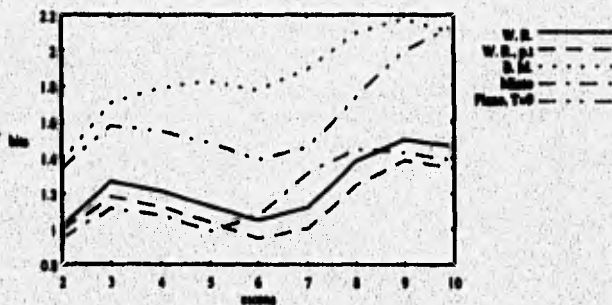


Figura 8.16d. Codificador de Walker y Rao, entropía de la señal transmitida, $T=9$.

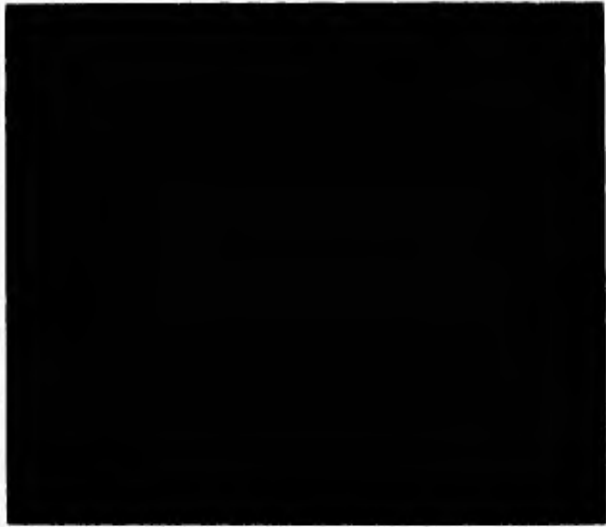


Imagen 8.13. Campo de vectores de desplazamiento con el método de Walker y Rao usando un codificador de Walker y Rao con umbral $T=9$.



Figura 8.14. Imagen reconstruida con el método de Walker y Rao usando un codificador de Walker y Rao con umbral $T=9$.

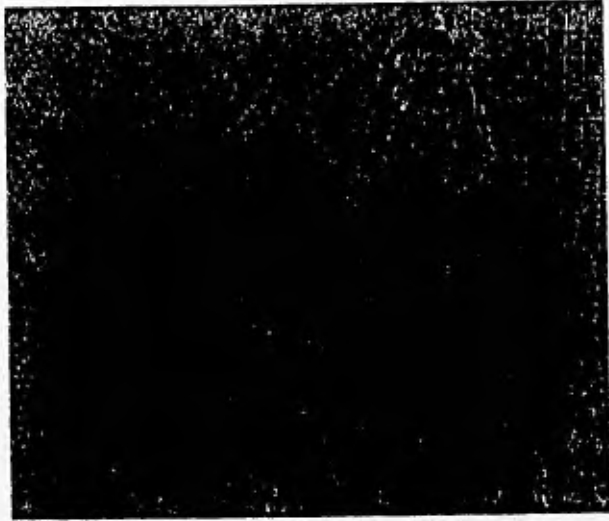


Imagen 8.15. Diferencia normalizada de la imagen original y la reconstruida con el método de Walker y Rao usando un codificador de Walker y Rao con umbral $T=9$.



Imagen 8.16. Diferencia normalizada de la imagen original y la reconstruida con el método de Walker y Rao usando un codificador de Walker y Rao con umbral $T=9$. Ampliación de una zona de gran movimiento.



Imagen 8.17 Señal de error a la salida del codificador, usando el método de Walker y Rao, con $T=9$.

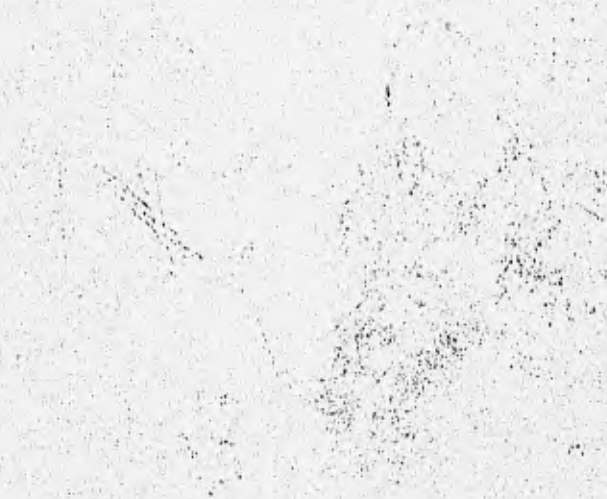


Imagen 8.18 Señal de error a la salida del codificador, usando el método de Walker y Rao, con $T=9$. En esta imagen se han removido los símbolos de "reset".

9. Conclusiones

La compresión de información en secuencias de imágenes en movimiento obtenido mediante el procesamiento digital, requiere la implantación de algoritmos que puedan eliminar la redundancia de información temporal y espacial de una manera eficiente.

Estos métodos debido a las características inherentes al tipo de señales estudiadas requieren realizar grandes cantidades de cálculos, por lo que es un requisito su simplicidad para reducir el tiempo en que lo hacen.

En este trabajo se revisaron los conceptos de estimación y compensación de movimiento con el fin de obtener una compresión de la información mediante un codificador que transmita únicamente la información esencial, no redundante y un decodificador que a partir de esta reconstruya la señal de la manera más exacta posible.

Se implantaron algoritmos de estimación de movimiento con el fin de obtener el mejor predictor posible para el método de compensación de movimiento. Las características que estos algoritmos deben reunir son: una alta relación señal a ruido y un tiempo de cómputo no muy grande. Se estudiaron dos casos, los métodos por recursivos y los de apareamiento de bloques, para los cuales se midió su desempeño bajo diferentes condiciones, tales como la propagación temporal y piramidal del campo de vectores inicial, diferentes umbrales de error, tamaños de bloque y áreas de búsqueda.

Para los métodos por recursivos se implantaron varios algoritmos propuestos por Netravali y Robbins [NET79], Walker y Rao [WAL84] y Cafforio y Rocca [CAF83], encontrándose para el conjunto de imágenes con los que se evaluaron, un mejor desempeño para el algoritmo de Walker y Rao con propagación temporal, entérmino de la relación señal a ruido y entropía de la señal de error. Se observó que para todos los algoritmos el uso de un campo inicial de vectores adecuado conlleva a un mejor desempeño de los algoritmos y una mayor rapidez en su convergencia. En el presente trabajo para dicho campo de vectores, se obtuvieron los mejores resultados mediante la propagación temporal de los vectores de desplazamiento de la escena anterior a la procesada, aunque también se experimentó con propagación del campo de vectores a través de dos escenas, así como en una estructura multiresolución y algunas combinaciones de las anteriores.

Aunque los métodos propuestos por Netravali y Robbins convergen en un tiempo menor a los propuestos por Walker y Rao y Cafforio y Rocca, su desempeño se ve directamente afectado por la necesidad de obtener un valor de ganancia óptimo, que es dependiente de la señal procesada y que el mismo algoritmo es incapaz de obtener, esto no ocurre para los algoritmos propuestos por Walker y Rao

y Cafforio y Rocca, que poseen una ganancia adaptable a la señal procesada, por lo que convergen con menor número de iteraciones con una calidad de estimación superior.

La estimación de movimiento usando técnicas de apareamiento de bloques resultó mucho más rápida que haciendo uso de técnicas pel recursivas, con un detrimento en la calidad de la estimación. El desempeño de estos métodos se ve afectado por el tamaño del bloque, observándose que entre más pequeño sea, mejor será la estimación obtenida, a un costo de tiempo de procesamiento mayor. La elección de funciones de máxima similitud no influye en la calidad de los resultados, excepto en el tiempo de procesamiento. En cuanto a técnicas de búsqueda, el método de los tres pasos se mostró superior al de búsqueda cruzada, probablemente debido a la restricción en el área de búsqueda que automáticamente este último define.

Aprovechando las ventajas, por un lado de la rapidez de los métodos por apareamiento de bloques y por el otro de la calidad de estimación de los métodos pel recursivos, se implantó un método que combina a ambos, consistiendo en la obtención de un campo burdo de vectores a partir de un método de apareamiento de bloques, que será refinado usando técnicas pel recursivas. Los resultados de este método se reflejaron en una calidad de estimación mayor que las obtenidas usando las dos técnicas separadamente, sin un aumento de tiempo de proceso significativo.

Una vez estudiados los métodos de estimación de movimiento se procedió al estudio de la compensación de movimiento, mediante la implantación de un codificador/decodificador de tipo DPCM. El punto clave para lograr la compresión en la compensación de movimiento, está en la elección de un buen predictor con lo cual la señal de error generada tendrá una entropía menor. La calidad de la imagen reconstruida en el decodificador, depende únicamente del umbral elegido para la decisión de transmitir la señal de error, o la DFD y no del método de estimación. Sin embargo, la entropía de la señal de error depende de ambos.

Comparando los métodos de apareamiento de bloques, con los pel recursivos usados como predictor en el compensador de movimiento, la entropía de la señal transmitida es mayor en el primer caso, por lo cual los autores se inclinan por el uso de métodos pel recursivos.

Dado que las características del método implantado por los autores para la estimación de movimiento, son diferentes a las de cualquiera de los dos mencionados anteriormente, su implantación directa en la compensación de movimiento de Walker y Rao no fue posible; en su lugar se implantó una variante donde el vector de condiciones iniciales es obtenido a través de este método y usado en cada actualización cuando el predictor falla. Aunque el método de estimación se mostró superior, su implantación en el codificador mostró unos resultados pobres, en comparación con el codificador visto anteriormente. Cabe esperar que un algoritmo de codificación en el que ocurra una mejoría en la adaptación del método

propuesto y un mejor uso de los valores iniciales de predicción mejorará los resultados obtenidos.

Haciendo un análisis comparativo entre los métodos de estimación y de compensación de movimiento, se observó que para la misma calidad de reconstrucción en términos de la relación señal a ruido, se puede usar un umbral de error mucho mayor en la compensación, obteniéndose además una entropía menor.

Por último, es necesario señalar que a partir de las bases estudiadas en este trabajo hay un gran campo de acción que poco a poco será ocupado conforme la tecnología avance en cuanto a aumento de velocidad de procesamiento y hacerlo así más accesible para aplicaciones reales de televisión digital y videoconferencias.

BIBLIOGRAFIA

1. WU92 WU-SHENG L., ANDREAS A., y MARCEL D. Two-Dimensional Digital Filters, 1992.
2. WAL84 D. R. WALKER y K. R. RAO "Improved Pel-Recursive Motion Compensation". IEEE Transactions on communications, vol. Com 32, No 10, Octubre de 1984. Páginas 1128-1134.
3. WAN92 Q. WANG y R. J. CLARKE. "Motion estimation and compensation for image sequence coding". Signal Processing: Image Communication 4 (1992). Páginas 161-174.
4. BAA91 BAAZIZ NADIA Approches d'estimation et de compensation de mouvement multirésolutions pour le codage de séquences d'images, Octubre de 1991. Páginas 99-137, 173-197.
5. MUR95 A. MURAT TEKALP Digital video processing, 1995. Páginas 72-129.
6. NET79 NETRAVALI A. N. y ROBBINS J. B. "Motion compensated television coding, part 1". Bell system technical Journal, 58(3):629-668, 1979
7. BUR83 P. J. BURT y E. H. ADELSON, "The Laplacian Pyramid as a Compact Image Code", IEEE transactions on communications, vol. com-31, no. 4, abril 1983.
8. ARB86 J. H. ARBEITER y R.F. BESSELER, "A Two Dimensional Real-Time Video Pyramid Processor" ,RCA Review, Marzo 1986, Vol. 47, No. 1
9. HAY83 S. HAYKIN, Communication Systems, John Wiley & Sons, Syngapore, 1983
10. FLE94 P. J. FLEMMING, "Introduction to Parallel Computing", Memoria del primer taller internacional de procesamiento paralelo TIPP '94, IIMAS UNAM, México 1994
11. BUR85 R. L. BURDEN, J.D. FAIRES, Análisis numérico, Grupo Editorial Iberoamérica, México, 1985

12. ROB83 J. D. ROBBINNS, A. N. NETRAVALI, "Recursive motion compensation: a review", NATO ASI Series, Vol F2, Berlín, 1983
13. SKL SKLAR, "Digital Communication Systems"
14. ALV93 A. ALVARADO, M. L. CASTILLO, Compresión de imágenes utilizando una estructura piramidal multiresolución. Tesis profesional, UNAM México, 1993
15. JAY93 N. JAYANT, J JOHNSTON y R. SAFRANEK, "Signal Compression Based on Models of Human Perception", Proceedings of the IEEE Vol. 81 No. 10, Octubre 1993
17. CAF83 C. CAFFORIO, F. ROCCA, "The differential method for motion estimation", Image Sequence Processing and Dynamic Scheme Analysis, NATO ASI Series, vol. 12, New York, 1983
18. PAN88 Y. M LE PANNERER, "La transmission numérique des images", La Recherche, No. 176, Vol. 19, Febrero 1988
19. GAL91 D. LE GALL, "MPEG: A Video Compression Standard for Multimedia Applications", Communications of the ACM, Vol. 34, No. 4, Abril 1991
20. KAP85 S. KAPPAGANTULA y D. R. RAO, "Motion Compensated Interframe Image Prediction", IEEE Transactions on Communications, Vol. COM-33, No. 9, septiembre de 1985

12. ROB83 J. D. ROBBINNS, A. N. NETRAVALI, "Recursive motion compensation: a review", NATO ASI Series, Vol F2, Berlín, 1983
13. SKL SKLAR, "Digital Communication Systems"
14. ALV93 A. ALVARADO, M. L. CASTILLO, Compresión de imágenes utilizando una estructura piramidal multiresolución. Tesis profesional, UNAM México, 1993
15. JAY93 N. JAYANT, J JOHNSTON y R. SAFRANEK, "Signal Compression Based on Models of Human Perception", Proceedings of the IEEE Vol. 81 No. 10, Octubre 1993
17. CAF83 C. CAFFORIO, F. ROCCA, "The differential method for motion estimation", Image Sequence Processing and Dynamic Scheme Analysis, NATO ASI Series, vol. 12, New York, 1983
18. PAN88 Y. M LE PANNERER, "La transmission numérique des images", La Recherche, No. 176, Vol. 19, Febrero 1988
19. GAL91 D. LE GALL, "MPEG: A Video Compression Standard for Multimedia Applications", Communications of the ACM, Vol. 34, No. 4, Abril 1991
20. KAP85 S. KAPPAGANTULA y D. R. RAO, "Motion Compensated Interframe Image Prediction", IEEE Transactions on Communications, Vol. COM-33, No. 9, septiembre de 1985