# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

## FACULTAD DE INGENIERIA

# CODIFICACION DE IMAGENES POR MEDIO DE TRANSFORMADAS ORTOGONALES TRASLAPADAS Y CODIFICACION VECTORIAL

T E S I S

QUE PARA OBTENER EL TITULO DE:

**INGENIERO EN COMPUTACION**

P R E S E N T A N:

MARIA GUADALUPE RAMIREZ FLORES
JORGE MANUEL PEREZ GODINEZ

D I R E C T O R D E T E S I S:

DR. FRANCISCO J. GARCIA UGALDE

MEXICO D.F. 1996

TESIS

COMPLETA

MARIA GUADALUPE RAMIREZ FLORES

A mis padres.

A mi abuelita.

A mi tía Mari y a mi padrino Ricardo.

A mis hermanos: Nacho, Nelly y Liz.

A Jorge Manuel Pérez Godínez.

A mis maestros y amigos.

A todos ustedes les dedico este trabajo como tributo a todo el cariño, guía, apoyo y ejemplo que me han dado a lo largo de mi vida.

JORGE MANUEL PEREZ GODINEZ

Quiero dedicar este trabajo a todas aquellas personas que han sido tan importantes en mi vida y que, sin su apoyo, jamás hubieramos podido concluirlo.

Primeramente a mi papá cuya vida ejemplar fue una gran motivación para dar mi mayor esfuerzo.

A mi mamá que con gran cariño supo conducirme hasta este momento tan importante en mi vida.

A mis hermanos Fer, Geli, Mari y Lalo por brindarme su apoyo y cariño.

A tí Lupita por tanta paciencia, siempre recordaré estos años como el mayor triunfo de una gran amistad.

A mis profesores y amigos por su apoyo y enseñanzas.

Un agradecimiento especial al Dr. Francisco García Ugalde por su gran apoyo; con toda nuestra admiración y respeto.

# INDICE

Página

Página

Página

Página



**CAPITULO 5**
**RESULTADOS Y CONCLUSIONES**

# CAPITULO 1

## INTRODUCCION

### 1.1 PLANTEAMIENTO DEL PROBLEMA

Una imagen generalmente se representa como una función de intensidad contra espacio de representación en el caso de imágenes fijas y en el caso de video como una función de intensidad contra espacio de representación y tiempo.

Un pixel es la unidad de espacio de representación, es el elemento mínimo de una imagen digitalizada. A cada pixel se le asigna un valor que indica la intensidad de gris, en el caso de imágenes en blanco y negro; para imágenes a color, el valor para cada pixel se compone de la intensidad de tres colores distintos, generalmente rojo, verde y azul (Sistema RGB).

El objetivo fundamental de la compresión de imágenes es producir la representación digital de una imagen a una baja tasa de bits con un mínimo de pérdida en la calidad de la imagen. A la función de compresión frecuentemente se le llama codificación a baja tasa de bits o simplemente codificación.

La capacidad de compresión ha sido básica para la tecnología de comunicación robusta a larga distancia, almacenamiento de imágenes de alta calidad y encriptación de mensajes. La compresión continúa siendo una tecnología clave en comunicaciones porque a pesar de los avances en los medios de transmisión ópticos con ancho de banda relativamente ilimitado, existe una necesidad cada vez mayor de medios de banda limitada tales como enlaces de radio y satelitales, y medios de almacenamiento como HD, CD-ROM y chips de memoria de estado sólido, con capacidad limitada.

De aquí la importancia de obtener imágenes con una mayor calidad a bajas tasas de bits; una de las técnicas de compresión más utilizadas y que ha presentado mejores resultados es la codificación por transformada, sin embargo el tipo de procesamiento que exige, produce ciertos efectos en la imagen que son conocidos como efectos de interbloqueo que degradan su calidad, obligando a utilizar tasas de bits más altas; es en este punto donde se enfocará el desarrollo del presente trabajo; el cual pretende mostrar

los beneficios del uso de la Transformada Ortogonal Traslapada (LOT) que en conjunción con un cuantizador vectorial permitirá reducir la tasa de transmisión conservando la calidad de las imágenes al disminuir los efectos de interbloqueo.

## 1.2 UBICACION DE LA CODIFICACION DENTRO DEL ESQUEMA DE COMUNICACION DIGITAL

En el esquema de comunicación digital, fig. 1.2.1 la compresión de imágenes cae dentro del proceso de codificación de fuente.

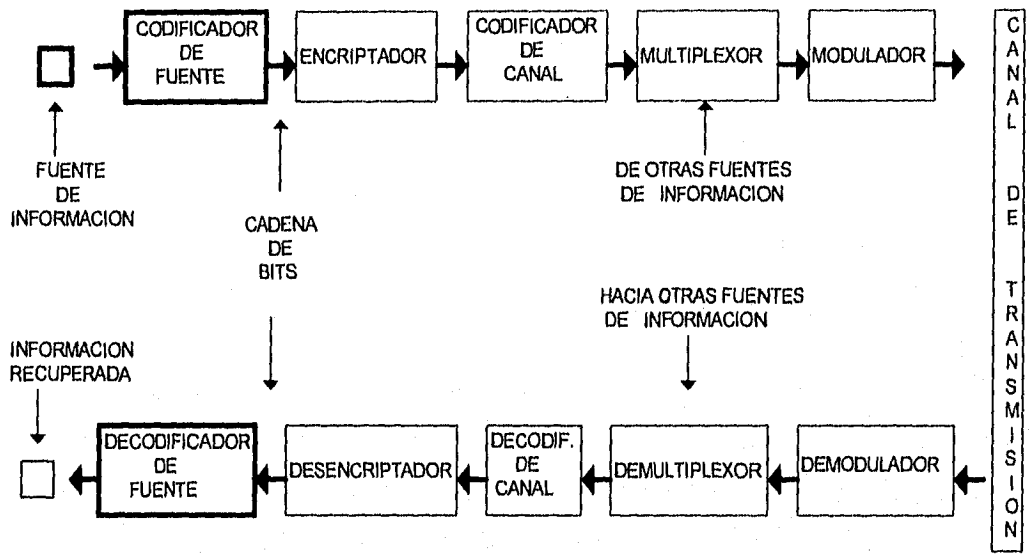



FIG. 1.2.1 Esquema de Comunicación Digital.

De acuerdo con este esquema, la función principal del codificador de fuente es optimizar la representación de la información minimizando la tasa de bits lo más posible sin degradar significativamente la señal de entrada. La tasa de bits se mide en bits por muestra o por segundo (bps).

Una vez codificada, la información pasa a través del codificador de canal, el cual se encarga de hacer un mapeo de los vectores de entrada que se encuentran en un espacio vectorial determinado, a un espacio vectorial de dimensión superior; es decir, el

codificador de canal agrega redundancia para dar protección contra errores de transmisión.

Finalmente, si se van a transmitir otro tipo de señales provenientes de distintas fuentes de información a través del mismo canal de transmisión, se realiza un multiplexaje de las señales, para después adecuar la información al canal de transmisión por medio de la modulación; el modulador maximiza la tasa de bits que puede soportar un canal dado o medio de almacenamiento sin causar un nivel de probabilidad de bit en error inaceptable; en la modulación la tasa de bits se mide en bits por segundo por Hertz (bps/Hz). En los sistemas llamados de modulación codificada, las operaciones de codificación de canal y modulación se combinan para aumentar la eficiencia. Los procesos de codificación de fuente y de canal pueden ser algunas veces integrados para incrementar la eficiencia de las comunicaciones digitales.

## 1.3. FUNDAMENTOS TEORICOS

### 1.3.1. PROPIEDADES DE LAS IMAGENES

Las imágenes tienen propiedades que deberemos tomar en cuenta para lograr una compresión eficiente. Entre las más importantes tenemos:

a) NO ESTACIONARIDAD: Se dice que un proceso aleatorio es estacionario en sentido estricto si conserva todos sus momentos iguales a lo largo del tiempo, y en sentido amplio si conserva los dos primeros (media y varianza). Una imagen no conserva sus momentos muestrales dado que contiene tanto regiones donde la intensidad varia lentamente como zonas de alto contraste como en los bordes de los elementos de la imagen.

Se han diseñado algoritmos adaptivos para cuantización, predicción y asignación de bits para mejorar la eficiencia de los sistemas de procesamiento con entradas no estacionarias.

b) PERIODICIDAD: Existen distintas fuentes de periodicidad en señales visuales como la periodicidad de línea a línea en imágenes fijas o de paquete a paquete en video. Aunque en la realidad las imágenes no son exactamente periódicas una función importante del algoritmo de compresión es remover la redundancia realizando una buena predicción por medio del seguimiento de la señal en forma adaptiva y codificando sólo la variación con respecto a la predicción.

c) DENSIDAD ESPECTRAL DE POTENCIA: En un análisis global o de larga duración las señales visuales tienden a tener un espectro de frecuencia paso bajas, sin embargo al realizar el análisis de corta duración se observan tanto frecuencias altas como bajas.

### 1.3.2 FUENTES DISCRETAS DE INFORMACION

Una fuente de información típicamente produce una salida de amplitud en función de una o más variables. Las dos variables independientes más comunes son el tiempo y la posición. En el caso discreto la fuente puede producir M señales distintas, a lo largo del tiempo o la posición discreta.

#### 1.3.2.1 SEÑALES ESTADISTICAMENTE INDEPENDIENTES.

Si consideramos solamente el caso de dependencia temporal, entonces la salida en amplitud puede escribirse como:

$$mk = fq(tk) \qquad 1.3.1$$

donde la función fq incluye la cuantización en amplitud y tk representa el tiempo discreto. La señal mk puede tomar un valor cualquiera de M niveles de cuantización (q=1,2,3,...,M). Si las señales son estadísticamente independientes, entonces:

$$P(mk,mk-1) = P(mk)P(mk-1) \qquad 1.3.2$$

### 1.3.2.2 SEÑALES ESTADISTICAMENTE DEPENDIENTES

Cuando existe dependencia entre las muestras, el valor de una señal dada depende de los valores previos. Esta dependencia puede modelarse por un proceso de Markov. En general, un proceso de Markov de orden $\nu$ esta definido por la probabilidad condicional

$$P(mk/mk-1,mk-2,mk-3,...,mk-\nu) \qquad 1.3.3$$

El modelo de Markov más común es un modelo de Markov de primer orden ($\nu$=1) el cual se define por la probabilidad condicional:

$$P(mk/mk-1) \qquad 1.3.4$$

Para el modelo de Markov de primer orden, la probabilidad de que se de la señal i en el tiempo k y la señal j en el tiempo k+1 esta definida por:

$$P(i,j) = P(i)P(j/i) \qquad 1.3.5$$

### 1.3.2.3 ENTROPIA EN UNA FUENTE DE INFORMACION

Podemos definir la información contenida en una sola señal para el modelo de Markov como

$$Imk = -\log_2 P(mk/mk-1,mk-2,mk-3,...,mk-\nu) \qquad 1.3.6$$

y para el caso del modelo estadísticamente independiente como:

$$Imk = -\log_2 P(mk) \qquad 1.3.7$$

Así podemos definir la entropía de una fuente como el promedio de información contenida en la misma, y por lo tanto, para un modelo de Markov de primer orden tenemos:

$$H = -\sum_{i=1}^{M}\sum_{j=1}^{M} P(i,j)\log_2 P(j/i) \qquad 1.3.8$$

donde P(i,j) = P(i)P(j/i)

Y para el modelo de señales estadísticamente independientes tenemos:

$$H = -\sum_{i=1}^{M} P(mi)\log_2 P(mi) \qquad 1.3.9$$

Si todas las señales son igualmente probables, entonces tenemos que la información contenida es:

$$H = \log_2 M \qquad 1.3.10$$

### 1.3.2.4 REDUNDANCIA EN UNA FUENTE DE INFORMACION

La redundancia esta definida como sigue:

$$\text{Redundancia} = \log_2 M - H \qquad 1.3.11$$

Donde M es el número de niveles de cuantización y H es la entropía de la fuente, es claro que en una fuente de señales estadísticamente independientes con probabilidad igual para todas las señales el valor de la redundancia es cero.

## 1.4 TECNICAS DE COMPRESION DE IMAGENES

Existen dos grandes clases de técnicas de compresión de imágenes, las REVERSIBLES O DE REDUCCION DE REDUNDANCIA y las IRREVERSIBLES O DE REDUCCION DE ENTROPIA. A continuación describiremos brevemente estas dos clases.

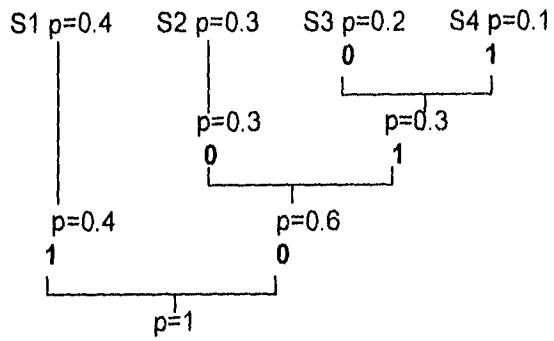
### 1.4.1 TECNICAS REVERSIBLES.

Son aquellas donde se intenta disminuir la redundancia de la señal, sin perder información, también reciben el nombre de técnicas de reducción de redundancia. Estas técnicas remueven aquella porción de los datos que pueden ser reinsertados o reconstruidos al recibir la señal. Entre estas técnicas tenemos:

a) CODIGOS DE HUFFMAN

La idea fundamental es codificar con un menor número de bits aquellas señales más probables, esto se hace obteniendo primero el histograma de la imagen y ordenándolo según un orden de probabilidades decreciente, posteriormente se agrupan las dos señales de menor probabilidad, asignándole previamente un uno a la señal de menor probabilidad y un cero a la de mayor probabilidad al agrupar las señales se obtiene una rama con la suma de las probabilidades agrupadas, esto se continúa haciendo para todas las señales hasta obtener una última rama de probabilidad uno. El código para cada señal se obtiene leyendo los unos y ceros asignados desde la rama final hasta la señal a codificar, como se muestra en el siguiente ejemplo:.

EJEMPLO




CODIGO

S1 1

S2 00

S3 010

S4 011

b) CODIGO RUN-LENGTH

Este código se basa en la eliminación de pixeles adyacentes del mismo valor, transmitiéndose sólo el número de pixeles adyacentes y el valor común.

La compresión que se puede lograr con este sistema depende de las características de la imagen.

c) BIT PLANE

Una imagen con N niveles de gris, puede considerarse como un conjunto de $\log_2 N$ planos de un bit cada uno, cada uno de estos planos puede ser codificado usando el método RUN-LENGTH, lo que aumenta la probabilidad de valores repetidos adyacentes y por lo tanto la capacidad de compresión del sistema, aunque este tipo

de codificación es muy sensible al ruido de canal pues un error en los planos más significativos, puede acarrear un error grande en la imagen reconstruida.

### 1.4.2 TECNICAS IRREVERSIBLES O DE REDUCCION DE ENTROPIA

En estas técnicas puede haber una reducción de redundancia, pero la idea fundamental es eliminar cierta cantidad de información de la señal produciendo la mínima distorsión posible, las medidas de distorsión de la señal pueden tener en cuenta la forma en que el ser humano percibe la señal, en este caso hablamos de técnicas de compresión de señales basadas en modelos de percepción humana, o bien podemos tener medidas de distorsión teóricas como el error cuadrático medio.

Entre los ejemplos principales de estas técnicas tenemos:

a) CODIFICACION PCM

Este tipo de codificación es la que realiza la conversión analógica digital de una señal, se realiza la discretización del tiempo tomando muestras de la señal al menos a la frecuencia de Nyquist, y la discretización en la amplitud asignando a cada muestra el nivel de cuantización más cercano.

b) CODIFICACION POR TRANSFORMADA

En este tipo de codificación se aplica a la imagen una transformación lineal obteniéndose así una representación donde la energía se concentra en un menor número de coeficientes, estos coeficientes se cuantizan, codifican y transmiten, posteriormente en el receptor se decodifican y se aplica la transformación inversa obteniéndose así la imagen reconstruida. Para realizar la compresión de una imagen, sólo se codifican los coeficientes de máxima energía, por lo que se busca que la transformación concentre la mayor cantidad de energía en el menor número de coeficientes.

Dentro de esta técnica de codificación se encuentra el método propuesto en este trabajo, el cual presenta una mejora significativa a las transformadas empleadas tradicionalmente.

c) TECNICAS PREDICTIVAS

En este caso se presupone una estructura dada para la imagen por ejemplo un proceso de Markov de primer orden, así, se codifica un pixel y se realiza una predicción del siguiente, lo único que se codifica entonces es el error entre la predicción y el valor real de la señal. La eficiencia de este tipo de codificación depende de la calidad de la predicción, y de la relación entre la estructura predefinida y la imagen real.

## 1.5 APLICACIONES DE LA COMPRESION DE IMAGENES

Dentro de las principales aplicaciones de la compresión de imágenes podemos señalar:

a) TRANSMISION DE IMAGENES FIJAS. Una imagen a color de 500x500 pixeles con un formato no comprimido de 24 bits por pixel (bpp) requerirá alrededor de 100s de tiempo de transmisión sobre un canal de 64 kbps. Con una codificación de 0.25 bpp, el tiempo de transmisión es de aproximadamente un segundo, un número que debe ser considerado como excelente para mostrar imágenes fijas. La tecnología actual para codificar una imagen de 500x500 es capaz de proveer una buena calidad a 0.25 bpp para una amplia gama de imágenes a color, asumiendo una distancia de visión de aproximadamente 6 veces el tamaño de la imagen. Para muchas imágenes, incrementar la tasa de bits a 1 bpp proporciona una imagen excelente y en algunos casos de calidad perceptualmente perfecta. La transmisión correspondiente a través de un canal de 64 kbps es de 4 s.

La aplicación de videoteléfono el cual asume el uso de una línea telefónica y de un modem a 9.6 kbps involucra imágenes de muy baja resolución espacial y temporal. Una resolución típica es de 100 x 100 pixeles por frame, y alrededor de 3 a 6 frames por segundo. A una resolución temporal menor, el sistema genera una secuencia degradada de servicio de imágenes fijas algunas veces llamado video de imagen congelada.

b) VIDEO DIGITAL. Las videoconferencias requieren resolución CIF (360x288 pixeles por imagen), o al menos 1/4 de la resolución CIF (180x144 pixeles por imagen). La resolución de la entrada temporal es usualmente un submúltiplo de 30 cuadros por segundo, por ejemplo 15 y hasta 10, para tasas de bits de 1.5 Mbps. Con resolución CIF y tasas de bits de 1.5 Mbps, la calidad de comunicaciones del servicio es generalmente alta. Con un cuarto de la resolución CIF o alguna otra resolución baja y con una resolución temporal también baja es posible producir menores tasas de bits como 48 o 112 Kbps. Pero la calidad de video es útil únicamente si se aceptan niveles bajos de nitidez en la imagen de salida y niveles muy bajos de movimiento en la escena de entrada, como la vista de la cabeza y hombros de una sola persona, un ambiente más bien correspondiente a la videotelefonía que a la videoconferencia. Las tasas de bits de 48 y 112 Kbps son apropiadas para sistemas ISDN con tasas 64 y 128 Kbps respectivamente. La tasa de bits de 384 Kbps es un número interesante en el actual estado de la tecnología. A esta tasa es posible obtener un nivel aceptable aunque no alto de calidad en la codificación de una escena de videoconferencia.

El CD-ROM tiene una capacidad de reproducción de 1.5 Mbps y una capacidad de almacenamiento de algunos gigabits. Si el video puede comprimirse a 1 Mbps un dispositivo de CD-ROM podría almacenar y reproducir una hora o más de señal de video, junto con sonido estéreo comprimido. Esta capacidad es central para varias aplicaciones recientes, tales como la reproducción de video en un CD de audio, o video direccionable.

c) ALMACENAMIENTO DE IMAGENES EN BASES DE DATOS. Una característica de las bases de datos actuales es la capacidad de manejar tipos de datos binarios o bien apuntadores a archivos binarios, esta característica da la posibilidad de almacenar información visual o sonora. La necesidad de realizar transacciones rápidas a través de una red en este tipo de sistemas hace indispensable la compresión tanto de imágenes como de sonido.

d) RECONOCIMIENTO DE PATRONES. Las técnicas de compresión de imágenes nos permiten extraer las características más relevantes de una señal, útiles para la identificación de la clase a la que pertenece una señal determinada, además de utilizarse en el preprocesamiento para obtener una representación manejable por los algoritmos de reconocimiento.

# CAPITULO 2

## CODIFICACION POR TRANSFORMADA

### 2.1 CONCEPTOS BASICOS

Una de las herramientas más útiles en el procesamiento de señales es la representación de la señal en diferentes espacios a través de transformadas lineales. Lo anterior se debe a que ciertas características de la señal pueden representarse mejor en un espacio que en otro. Así usando una transformada podemos realizar diferentes operaciones básicas del procesamiento digital de señales tales como el filtrado y la codificación.

Antes de calcular la transformada de una señal debemos dividirla en bloques. Cada bloque x es un conjunto de muestras consecutivas de la señal que podemos representar como:

$$x = [\, x(mM) \;\; x(mM+1) \;\; \ldots\ldots \;\; x(mM+M-1) \,]^{T} \qquad 2.1.1$$

en la ecuación 2.1.1 el operador T significa transpuesta, pues consideraremos los bloques de la señal como vectores columna. m representa el índice del bloque. La notación correcta para x debería de ser x(m) para hacer claro el hecho de que estamos trabajando con una secuencia de bloques. Sin embargo, esto podría causar confusión con otros índices por lo que esta dependencia deberá entenderse en forma implícita. La dimensión de x es M a este valor se le conoce también como largo o tamaño del bloque.

Una imagen debe dividirse en bloques porque la complejidad de los algoritmos de transformación crece exponencialmente con el tamaño de la imagen, así resulta más simple calcular varias transformadas de orden menor que una transformada de orden mayor. Además se aprovechan las características locales de la imagen.

Considere la siguiente transformación lineal sobre x

$$X = A^{T}x \qquad 2.1.2$$

Llamamos a X la transformada de x y a A la matriz de transformación o simplemente transformación. Si los elementos de A son complejos el operador T significa conjugada transpuesta.

Podemos recobrar la señal original a partir de su transformación por medio de la inversa de $A^T$

$$x = [A^T]^{-1} X \qquad 2.1.3$$

La ecuación mencionada es válida para cualquier transformada invertible: Considerando que A es una matriz ortogonal tenemos que

$$A^{-1} = A^T$$

por lo tanto

$$x = A X \qquad 2.1.4$$

Usamos $A^T$ en la transformada directa y A en la inversa de tal forma que las funciones básicas de la transformada queden como las columnas de A, así el késimo elemento de X será la proyección de x sobre la késima función básica, y x se obtendrá a través de una combinación lineal de todas estas proyecciones. Cada proyección se obtiene como el producto interno entre el bloque de entrada x y la késima función básica. Para las transformadas de bloque tradicionales la matriz A es cuadrada de orden M de forma tal que el número de coeficientes de transformación en X es igual al número de muestras en x.

En aplicaciones prácticas el uso de transformaciones ortogonales tiene muchas ventajas tales como:

a) La transformación inversa se obtiene inmediatamente de la transformación directa sin necesidad de inversión.

b) Teniendo el diagrama de flujo de la transformada directa se obtiene el de la transformada inversa siguiéndolo en sentido inverso.

c) Una transformada ortogonal conserva la energía de la señal, esto es:

$$\|x\| = \|X\|$$

donde

$$\|x\| = \sum_{k=1}^{M} \left| [x]_k \right|^2 \qquad 2.1.5$$

donde $[x]_k$ denota el késimo elemento de x.

En el caso de la codificación por transformada buscamos además, que la mayor parte de la energía se concentre en el menor número de coeficientes, esto es:

$$\sum_{n=0}^{M-1} \left| [X]_n \right|^2 \cong \sum_{n=0}^{D-1} \left| [X]_n \right|^2 \qquad 2.1.6$$

Donde D es menor que M.

En la discusión anterior, hemos asumido que el bloque de la señal de entrada es unidimensional. Sin embargo en la codificación de imágenes cada bloque es una matriz de MxM definido por

$$.B = \begin{vmatrix} a_{mM,mM} & a_{mM+1,mM} & \cdots & \cdots & a_{mM+M-1,mM} \\ a_{mM,mM+1} & a_{mM+1,mM+1} & \cdots & \cdots & a_{mM+M-1,mM+1} \\ \cdots\cdots & \cdots\cdots & \cdots & \cdots & \cdots\cdots \\ \cdots\cdots & \cdots\cdots & \cdots & \cdots & \cdots\cdots \\ a_{mM,mM+M-1} & a_{mM+1,mM+M-1} & \cdots & \cdots & a_{mM+M-1,mM+M-1} \end{vmatrix}^T \qquad 2.1.7$$

y la transformada es un tensor tetradimensional, para estas aplicaciones lo más conveniente es usar transformadas separables donde el tensor para señales bidimensionales puede escribirse como:

$$[D]_{lmrs} = [A]_{lr} * [A]_{ms} \qquad 2.1.8$$

donde * denota el producto Kronequer.

La ecuación anterior implica que la transformada para una señal bidimensional puede obtenerse en dos pasos, primero debe calcularse la transformada de todos los renglones del bloque y después debe calcularse la transformada sobre las columnas del bloque obtenido en el paso anterior.

## 2.2 TRANSFORMADA KARHUNEN-LOÈVE (KLT)

Suponga que deseamos transmitir una señal senoidal, podemos tomar muestras y transmitirlas secuencialmente, el número de valores a transmitir dependerá de la frecuencia y duración de la señal y de la precisión con la que deseemos hacer la reconstrucción. Por ejemplo si deseamos transmitir una senoidal a 1 Khz, durante un segundo muestreada a la frecuencia de Nyquist deberemos transmitir o almacenar 2000 valores, sin embargo sabemos que para reconstruir una señal senoidal sólo necesitamos saber su amplitud, frecuencia, fase, duración y el hecho mismo de que es senoidal. Desde el punto de vista de la teoría de la información, las muestras tomadas de la señal senoidal y en general de cualquier señal determinística están altamente correlacionadas y contienen poca información. Por otro lado los cinco parámetros (amplitud, frecuencia, fase, duración y forma) están completamente descorrelacionados y contienen la misma información que el conjunto de todas las muestras de la señal original.

La KLT hace exactamente lo anterior cuando la señal de entrada es un proceso de Markov de primer orden. Es decir, la KLT toma una función aleatoria y obtiene una representación descorrelacionada de esta.

Consideremos el vector de entrada x de media cero

$$x = [x(0), x(1), \ldots, x(k), x(M-1)]^T \qquad 2.2.1$$

Si $\{A_k\}$ es el conjunto de vectores linealmente independientes que describen un espacio vectorial N-dimensional entonces podemos representar x como una

combinación lineal de las funciones básicas, esto es:

$$x = \sum_{k=0}^{M-1} X_k A_k \qquad 2.2.2$$

Donde $X_k$ está definido como la proyección de x sobre la k-ésima función básica:

$$X_k = \langle x, A_k \rangle / \langle A_k, A_k \rangle \qquad 2.2.3$$

Aquí <.,.> denota el producto interno.

Supongamos que sólo retenemos los primeros D coeficientes de transformación (D menor que M), entonces el vector x puede representarse como:

$$\hat{x} = \sum_{k=0}^{D-1} X_k A_k \qquad 2.2.4$$

usamos la notación $\hat{x}$ para indicar que es una aproximación de x.

Buscamos que esta aproximación sea la mejor posible en el sentido del error cuadrático medio.

El error entre x y $\hat{x}$ está dado por:

$$\varepsilon = E[x - \hat{x}]^2 \qquad 2.2.5$$

donde E[ . ] denota el operador esperanza definido por:

$$E[x(k)] = \sum_{k=0}^{M-1} x(k) P(x(k)) \qquad 2.2.6$$

entonces siguiendo de 2.2.2,2.2.4 y 2.2.5

$$\varepsilon = E\left\{\left[\sum_{k=0}^{M-1} X_k A_k - \sum_{k=0}^{D-1} X_k A_k\right]^2\right\}$$

$$\varepsilon = E\left\{\left[\sum_{k=0}^{D-1} X_k A_k + \sum_{k=D}^{M-1} X_k A_k - \sum_{k=0}^{D-1} X_k A_k\right]^2\right\}$$

$$\varepsilon = E\left\{\left[\sum_{k=D}^{M-1} X_k A_k\right]^2\right\}$$

Tomando el producto punto tenemos:

$$\varepsilon = E\left\{\left[X_D A_D + X_{D+1} A_{D+1} + \ldots + X_{M-1} A_{M-1}\right]^2\right\}$$

La expresión anterior se desarrolla como la suma de los cuadrados más los dobles productos de las combinaciones, sin embargo, los dobles productos de las combinaciones son cero dado que las funciones $A_k$ son ortogonales. Por lo tanto tenemos:

$$\varepsilon = E\left[\left(X_D A_D\right)^2 + \left(X_{D+1} A_{D+1}\right)^2 + \ldots + \left(X_{M-1} A_{M-1}\right)^2\right]$$

$$\varepsilon = E\left[\left\langle \sum_{k=D}^{M-1} X_k A_k, \sum_{k=D}^{M-1} X_k A_k \right\rangle\right] \qquad 2.2.7$$

Considerando que las funciones básicas son ortonormales

$$\langle A_k, A_i \rangle = \begin{cases} 1 & \text{si} \quad k = i \\ 0 & \text{si} \quad k \neq i \end{cases} \qquad 2.2.8$$

Obtendremos

$$\varepsilon = E\left[\sum_{k=D}^{M-1} |X_k|^2\right] \qquad 2.2.9$$

Note que la expresión anterior representa la energía en los últimos coeficientes, por lo tanto al minimizarla maximizaremos la eficiencia de la compactación de energía y la ganancia de codificación.

De 2.2.3, 2.2.8 y 2.2.9

$$\varepsilon = E\left[\sum_{k=D}^{M-1} |\langle x, A_k \rangle|^2\right] \qquad 2.2.10$$

Tomando en cuenta la definición de producto interno

$$\langle x, y \rangle = y^T x = x^T y$$

Podemos reducir la ecuación anterior a:

$$\varepsilon = E\left[\sum_{k=D}^{M-1} A_k{}^T x x^T A_k\right]$$

$$\varepsilon = \sum_{k=D}^{M-1} A_k{}^T E[x x^T] A_k \qquad 2.2.11$$

Si definimos la matriz de autocovarianza de un vector aleatorio x de media cero como

$$R_{xx} = E[x x^T] \qquad 2.2.12$$

y minimizamos la ecuación 2.2.11 sujeto a la condición de ortonormalidad de 2.2.8 obtendremos el siguiente problema de multiplicadores de Lagrange:

$$\text{grad}\,\varepsilon = \lambda_k \text{grad}\langle A_k, A_k \rangle$$

$$\frac{\partial}{\partial A_k}\left\{\varepsilon - \lambda_k\left\langle A_k, A_k\right\rangle\right\} = 0 \qquad 2.2.13$$

este problema conduce a la solución de un sistema de ecuaciones homogéneo de la forma:

$$\left(R_{xx} - \lambda_k[I_M]\right)A_k = 0 \qquad 2.2.14$$

donde $\lambda_k$ son los multiplicadores de Lagrange e [IM] es la matriz identidad de orden M.

Así la minimización del MSE en una representación truncada de x conduce al problema de valores característicos en 2.2.14.

El conjunto de vectores básicos obtenidos diagonalizará la matriz de autocovarianza en el dominio de la transformada. Lo que indica la minimización de la redundancia.

Sea A la matriz de vectores característicos dada por

$$[A] = |\ A_0 \quad A_1 \quad \ldots\ldots \quad A_{M-1}\ | \qquad 2.2.15$$

Entonces

$$A^T R_{xx} A = R_{xx} = \mathrm{diag}\left\{\lambda_0, \lambda_1, \ldots\ldots, \lambda_{M-1}\right\} \qquad 2.2.16$$

Podemos observar que la matriz A se compone de los vectores característicos de la matriz de autocovarianza en forma de columnas. La KLT es precisamente la transformación A.

Existen consecuencias obvias del problema de diagonalización en 2.2.14, por ejemplo el cálculo de las funciones básicas depende de la matriz de autocovarianza de la señal de entrada y por lo tanto no pueden predeterminarse, además la KLT no tiene una estructura que permita la obtención de un algoritmo rápido.

Existen algunos casos para los cuales hay solución analítica, uno de ellos es cuando la señal se comporta como un proceso de Markov estacionario de primer orden, en este caso la matriz de autocovarianza es definida como:

$$[R_{xx}]_{kn} = \rho^{|k-n|} = \begin{vmatrix} 1 & \rho & \rho^2 & .... & \rho^{M-1} \\ \rho & 1 & \rho & .... & \rho \\ . & . & . & . & . \\ . & . & . & . & . \\ \rho^{M-1} & \rho^{M-2} & \rho^{M-3} & . & 1 \end{vmatrix} \qquad 2.2.17$$

En tal caso la solución fue dada como sigue [V. Grenander and G. Szego, "Toeplitz Forms and their Applications." Springer-Verlag, Berlin and N.Y., 1969]:

$$[A]_{nk} = A_k(n) = \left[2/(M+\lambda_k)\right]^{1/2} \operatorname{sen}\left\{w_k\left[(n+1)-(M+1)/2\right]+(k+1)\pi/2\right\}$$

$$n,k = 0,1,...,M-1 \qquad 2.2.18$$

donde:

$$\lambda_k = \frac{(1-\rho^2)}{1-2\rho\cos(w_k)+\rho^2} \qquad 2.2.19$$

donde la $w_k$ se obtiene de la solución de la ecuación

$$\tan(Mw) = -\frac{(1-\rho^2)\operatorname{sen}(w)}{\left[\cos(w)-2\rho+\rho^2\cos(w)\right]} \qquad 2.2.20$$

Se dice que la KLT es óptima porque cumple con las siguientes propiedades:

a) Descorrelaciona completamente la señal en el dominio de la transformada.

b) Contiene la mayor energía en el mínimo número de coeficientes de transformación.

c) Minimiza el MSE en reducción de ancho de banda o en la compresión de información.

La implementación de la KLT involucra la estimación de la matriz de autocovarianza de la señal, el cálculo de los vectores característicos, la construcción de la matriz de transformación y la transformación misma. Todo esto hace que la KLT sea una herramienta ideal pero impráctica.

## 2.3 TRANSFORMADA COSENO DISCRETA (DCT)

### 2.3.1 DEFINICION

La transformada DCT es una aproximación a la transformada KLT cuando la señal de entrada está altamente correlacionada, es decir la matriz de autocovarianza tiene la forma:

$$[R_{xx}]_{k,n} = \underset{\rho \Rightarrow 1}{\rho}^{(k-n)} = \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ .. & .. & .. & \dots & .. \\ .. & .. & .. & \dots & .. \\ 1 & 1 & 1 & \dots & 1 \end{vmatrix} \qquad 2.3.1$$

Así la ecuación 2.2.20 se reduce a:

$$\tan(Mw) = 0 \qquad 2.3.2$$

lo que nos da la solución para $w_k$ como:

$$w_k = \frac{k\pi}{M} \qquad 2.3.3$$

Los valores característicos de 2.2.19 desaparecen para $w_k$ distintas de cero, sin embargo para $\lambda_k$ cuando $w_k = 0$ el problema se indetermina por lo que debemos tomar en cuenta la siguiente propiedad de la matriz de autocorrelación

$$\sum_{k=0}^{M-1}[R_{xx}]_{kk} = \sum_{k=0}^{M-1}\lambda_k \qquad 2.3.4$$

y dado que los elementos de la diagonal de Rxx son todos unos tenemos que

$$M = \sum_{k=0}^{M-1}\lambda_k$$

pero como

$$\lambda_k = 0 \qquad \text{para} \qquad k \neq 0 \qquad 2.3.5$$

tenemos que

$$\lambda_k = M \qquad \text{para} \qquad k = 0 \qquad 2.3.6$$

Sustituyendo 2.3.3, 2.3.5 y 2.3.6 en 2.2.18

$$A_{no} = \frac{1}{\sqrt{M}} \qquad 2.3.7$$

$$A_{nk} = \sqrt{\frac{2}{M}}\operatorname{sen}\left\{k\left(n+\frac{1}{2}\right)\frac{\pi}{M}+\frac{\pi}{2}\right\}$$

$$A_{nk} = \sqrt{\frac{2}{M}}\cos\left[k\left(n+\frac{1}{2}\right)\frac{\pi}{M}\right] \qquad k \neq 0 \qquad 2.3.8$$

Así podemos definir el kernel de la transformada coseno discreta como:

$$A_{nk} = \sqrt{\frac{2}{M}} c(k) \cos\left[\left(n+\frac{1}{2}\right)\frac{k\pi}{M}\right] \qquad 2.3.9$$

donde

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{para} \quad k=0 \\ 1 & \text{en otro caso} \end{cases} \qquad 2.3.10$$

Una variación de la DCT obtenida en un intento por mejorar la respuesta en frecuencia de la DCT es la DCT-IV definida como:

$$A_{nk} = \sqrt{\frac{2}{M}} \cos\left[\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right] \qquad 2.3.11$$

### 2.3.2. ALGORITMOS RAPIDOS PARA EL CALCULO DE LA DCT

Una transformada rápida de Fourier (FFT) de M puntos como se presentará en la siguiente sección puede utilizarse para calcular una DCT de M puntos, por medio de un simple reordenamiento de los datos a la entrada más M/2-1 multiplicaciones complejas como fue propuesto inicialmente por Narashima.

Considerando la definición de la DCT sin tomar en cuenta (por facilidad) a $c(k)\sqrt{\frac{2}{M}}$ en 2.3.9 tenemos que el vector transformado $X_k$ está dado por:

$$X_k = \sum_{n=0}^{M-1} x(n) \cos\frac{\pi(2n+1)k}{2M} \quad ; \quad k=0,1,..,M-1 \qquad 2.3.12$$

Considerando ahora una M par y definiendo una nueva secuencia de entrada y(n) dada por

$$y(n) = x(2n)$$

$$y(M-1-n) = x(2n+1)$$

$$\text{para } n = 0,1,\ldots,M/2-1 \qquad 2.3.13$$

Haciendo la substitución de 2.3.13 en 2.3.12 se puede calcular X(k) como la suma de dos transformadas de datos pares e impares de orden M/2:

$$X(k) = \sum_{n=0}^{\frac{M}{2}-1} y(n)\cos\left[\frac{\pi(4n+1)k}{2M}\right] + \sum_{n=0}^{\frac{M}{2}-1} y(M-1-n)\cos\left[\frac{\pi(4n+3)k}{2M}\right] \qquad 2.3.14$$

para k= 0,1,...,M-1

sustituyendo n'=M-1-n, en el segundo término de la expresión anterior y dado que como $\cos(2\pi+\alpha) = \cos(\alpha)$, 2.3.14 se puede expresar mediante:

$$X(k) = \sum_{n=0}^{M-1} y(n)\cos\left[\frac{\pi(4n+1)k}{2M}\right] \qquad 2.3.15$$

para k= 0,1,...,M-1

Esta expresión puede ser evaluada como:

$$X(k) = \mathrm{Re}\{H(k)\} \;; \qquad k=0,1,..,M-1 \qquad 2.3.16$$

donde H(k) resulta de transformar 2.3.15 en una forma compleja:

$$H(k) = e^{j\pi k/2M}\sum_{n=0}^{M-1} y(n)e^{j2\pi nk/M}\ ; \qquad k=0,1,..,M-1 \qquad 2.3.17a$$

Podemos observar que esta expresión consta del cálculo de una IFFT de la secuencia y(n) y de M-1 multiplicaciones complejas. Tomando la parte real como lo indica 2.3.16 se puede obtener 2.3.15.

Entonces dado que sólo hay que tomar la parte real de 2.3.17a, esta expresión puede reemplazarse por:

$$(k) = e^{-j\pi k/2M}\sum_{n=0}^{M-1} y(n)e^{-j2\pi n/2M} \qquad 2.3.17b$$

La cual implica el cálculo de una FFT de la secuencia y(n). Es fácil verificar que la multiplicación por $e^{-j\pi k/2M}$ en esta última ecuación tiene la ventaja de que el

$$\cos\left[\frac{\pi k}{2M}\right] = \text{sen}\left[\frac{\pi(M-k)}{2M}\right] \ , \qquad \text{para } k=1,2,..,M/2-1 \qquad 2.3.18$$

de tal manera que X(k) puede ser evaluada a partir de la expresión 2.3.17b como:

$$X(k) = \text{Re}\left[H(k)\right] \ ; \qquad k=0,1,...M/2 \qquad 2.3.19a$$

$$X(M-k) = -\text{Im}\left[H(k)^*\right]; \qquad k=1,2,...M/2-1 \qquad 2.3.19b$$

Así podemos ver que la DCT de un bloque x(n) puede obtenerse a partir del cálculo de una FFT aplicada sobre un vector con los datos de entrada reordenados más M/2-1 multiplicaciones complejas, como lo indican las expresiones 2.3.19a y 2.3.19b.

De manera similar, una IDCT puede evaluarse a partir de la siguiente expresión:

$$P(n) = Re\left\{\sum_{k=0}^{M-1}\left[e(k)X(k)e^{j\pi k/2M}\right]e^{j2\pi kn/M}\right\} \qquad 2.3.20$$

para n= 0,1,..,M-1

donde la IDCT de la secuencia X(k) en x(n) es calculada como:

$$x(2n) = P(2n) \quad ; \quad x(2n+1) = P(M-1-n) \qquad 2.3.21$$

para n = 0,1,..,M/2-1

La expresión 2.3.20 nos indica el cálculo de una IFFT, tomando de esta sólo la parte real en un orden determinado por 2.3.21

ALGORITMO RAPIDO PARA EL CALCULO DE LA DCT-IV.

La DCT-IV de un vector de entrada x puede escribirse como:

$$X_k = \sum_{n=0}^{M-1} x_n \cos\left[\frac{\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\pi}{M}\right] \qquad 2.3.22$$

donde omitimos el factor de normalización $\sqrt{\frac{2}{M}}$ por simplicidad.

Dado el vector de entrada $[x_0 \quad ..... \quad x_{m-1}]$, podemos definir dos nuevos vectores $[u_n]$ y $[v_n]$ como:

$$u_n = x_{2n}$$
$$v_n = x_{M-1-2n}$$

Entonces la ecuación 2.3.22 puede ser escrita como:

$$X_k = \sum_{n=0}^{M/2-1} u_n \cos\left[\frac{\left(2n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\pi}{M}\right] + \sum_{n=0}^{M/2-1} v_n \cos\left[\frac{\left(M-1-2n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\pi}{M}\right]$$

Lo cual es equivalente a:

$$X_{2k} = \mathrm{Re}\left\{W_{2M}{}^{k}\sum(u_n + jv_n)W_{8M}{}^{4n+1}W_{M/2}{}^{kn}\right\}$$

$$X_{M-1-2k} = -\mathrm{Im}\left\{W_{2M}{}^{k}\sum(u_n + jv_n)W_{8M}{}^{4n+1}W_{M/2}{}^{kn}\right\}$$

donde

$$W_M^{nk} = e^{-2\pi nkj/M}$$

En la ecuación anterior, vemos que $[X_k]$ puede ser obtenida a partir de una DFT de longitud M/2 de $(u_n + jv_n)W_{8M}{}^{4n+1}$.

El algoritmo rápido para el cálculo de la DCT-IV se puede resumir en los siguientes pasos:

1.- Comenzar con un vector de entrada de M números reales y rearreglar la secuencia de acuerdo a:

$$x_{2n} \Rightarrow x_n$$
$$x_{M-1-2n} \Rightarrow x_{n+M/2}$$

para n = 0,1,..M/2-1

2.- Calcular

$$x_n \Leftarrow \mathrm{Re}\left\{\left[x_n + jx_{n+M/2}\right]\exp\left[-\frac{j\left(n+\frac{1}{4}\right)\pi}{M}\right]\right\}$$

$$x_{n+M/2} \Leftarrow \mathrm{Im}\left\{\left[x_n + jx_{n+M/2}\right]\exp\left[-\frac{j\left(n+\frac{1}{4}\right)\pi}{M}\right]\right\}$$

Después de este paso, tenemos un vector complejo de M elementos, el cual se divide en dos vectores reales : la parte real está en $\left[x_0 \ldots\ldots x_{M/2-1}\right]$, y la parte imaginaria está almacenada en $\left[x_{M/2} \ldots\ldots x_{M-1}\right]$.

3.- Calcular

$$\left[x_0 \ldots x_{M-1}\right] \Leftarrow \mathrm{DFT}\left\{\left[x_0 \ldots x_{M-1}\right], M/2\right\}$$

En este paso, usamos un cálculo de largo M/2 para calcular la DFT de $\left[x_n\right]$, la primera mitad del vector contiene la parte real y la segunda contiene la parte imaginaria tanto en el vector de entrada como en el de salida.

4.- Calcular

$x_n \Leftarrow x_n \exp(-jn\pi/M)$ , n= 1,2,..,M/4-1,M/4+1,..,M/2-1

$x_{M/4} = \sqrt{2}(1-j)x_{M/4}$

En este paso, $[x_n]$ es aún un vector complejo.

5.- Reordenamiento de datos.

$$x_{2n} \Leftarrow x_n \qquad , n=0,1,..M/2-1$$

$$x_{M-1-2n} \Leftarrow -x_{n+M/2} \qquad ; n=0,1,..M/2-1$$

Después de este paso, el vector de salida se interpreta nuevamente como real y contiene los coeficientes de la DCT-IV.

La DCT-IV se utilizará para el cálculo de la DCT y de las transformadas ortogonales traslapadas, siguiendo el algoritmo anterior.

## 2.4 TRANSFORMADA DISCRETA DE FOURIER (DFT)

### 2.4.1 INTRODUCCION

La DFT es una de las herramientas más importantes en el procesamiento digital de señales, esto se debe tanto a su uso en la estimación de espectro como a su utilidad para calcular otras transformadas, el éxito de la DFT se debe principalmente a la existencia de algoritmos rápidos que permiten su cálculo en tiempo real. En esta sección estudiaremos las propiedades de la DFT así como los algoritmos SPLIT RADIX y SPLIT RADIX VALUADO REAL que son de utilidad en el cálculo tanto de la DCT como de la DCT-IV que a su vez se utilizan en el cálculo de la transformada LOT.

### 2.4.2 PROPIEDADES DE LA DFT

La DFT es una transformada por su propio derecho y posee una inversa. Es análoga a la transformada de Fourier continua (CFT), pero con algunas diferencias importantes. Son estas diferencias las que más frecuentemente conducen a confusiones cuando la DFT comienza a usarse. Así un buen entendimiento de la DFT

y sus propiedades es importante antes de usar los algoritmos rápidos. La cercana relación entre la DFT y la CFT es lo que la hace preferible en muchas ocasiones. Es el descubrimiento de algoritmos rápidos lo que la hace realmente aplicable al procesamiento de señales. Si la DFT de una serie en el tiempo de M muestras usando un método tradicional requiere de $M^2$ operaciones, usando los algoritmos rápidos se requieren $M\log_2 M$ operaciones. Esto muestra el ahorro de tiempo de cómputo sobre todo cuando M es grande.

En esta sección la DFT será relacionada a la CFT. No damos pruebas formales para estas relaciones. La transformada de Fourier para señales continuas está dada por:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}dt \qquad 2.4.1$$

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{j2\pi ft}df \qquad 2.4.2$$

donde X(f) es la función en el dominio de la frecuencia y x(t) es la función en el dominio del tiempo y $j = \sqrt{-1}$.

La transformada de Fourier discreta análoga esta dada por:

$$Y(k) = \sum_{n=0}^{M-1} y(n)e^{-j2\pi kn/M} \quad ; \quad k=0,1,..,M-1 \qquad 2.4.3$$

$$y(n) = \frac{1}{M}\sum_{k=0}^{M-1} Y(k)e^{j2\pi kn/M} \quad ; \quad n=0,1,..,M-1 \qquad 2.4.4$$

donde Y(k) es el k-ésimo coeficiente de la DFT y y(n) es la n-ésima muestra de una serie en el tiempo. y(n) puede ser compleja y Y(k) es invariablemente compleja.

Varias definiciones de la DFT y su inversa se encuentran en la literatura, pero difieren de la definición dada aquí solamente por un factor de escalamiento. Note que

la DFT se aplica solamente sobre un tiempo limitado y produce un número finito de puntos de transformación. Por esta razón es algunas veces llamada transformada finita de Fourier. Es útil extender el rango de la definición de Y(k) sobre todos los enteros (tanto positivos como negativos).

Entonces encontramos que tanto el tiempo como las funciones de frecuencia discreta se repiten con período M. Esto es:

$$Y(k) = Y(k+M) = Y(k+2M) = \ldots . \qquad 2.4.5$$

$$y(n) = y(n+M) = Y(n+2M) = \ldots . \qquad 2.4.6$$

Ahora deseamos ver como la DFT se relaciona a la CFT. Para hacer esto primero consideraremos la función de tiempo x(t) y la magnitud de su CFT dada por $\left[X(f)^2\right]$. Como se muestra en la fig. 2.4.1a. Asumiremos que X(f) no es de banda limitada. Por razones obvias solamente una porción finita de x(t) puede ser analizada. La porción finita puede ser aislada multiplicando x(t) por una ventana rectangular w(t) como se muestra en la figura 2.4.1b viendo x(t) a través de esta ventana se produce la figura mostrada en 2.4.1c.

La ventana rectangular tiene una CFT que tiene la forma de una función sinc(x) la magnitud cuadrada de esta $\left[\text{sinc}(x)\right]^2$ la mostramos en la figura 2.4.1b.

La multiplicación en el dominio del tiempo es equivalente a la convolución en el dominio de la frecuencia tal que la CFT de una porción finita de x(t) se obtiene convolucionando X(f) con W(f). Esto se muestra en la fig. 2.4.1c. Note que esta convolución ha distorsionado el espectro original. Para realizar el análisis de esta señal de largo finito de manera discreta, primero tendrá que ser muestreada. La versión muestreada se representa en la fig. 2.4.1d donde la función continua original ha sido reemplazada por una serie de impulsos uniformemente espaciados, cada uno de peso proporcional a la función continua al tiempo del muestreo.

La CFT de una señal muestreada consiste de la suma de un número infinito de versiones de la CFT de la señal original cada una separada 1/2T Hz: esto se muestra en la fig. 2.4.1d.

El espectro resultante se muestra con una línea continua. La línea punteada muestra la parte del espectro original que se traslapó con el espectro original adyacente que se suman para dar esa parte de espectro resultante. Obviamente si el espectro original es de banda limitada a una frecuencia fo menor a 1/2T entonces no habrá traslape de las versiones adyacentes del espectro original. En este caso el espectro de la señal muestreada será idéntico al espectro de la señal original en un intervalo de -fo <f <fo y por lo tanto la señal original podrá ser recuperada pasándola a través de un filtro ideal paso bajas con frecuencia de corte fo. Esto es inherente al proceso de muestreo, si por el contrario fo es mayor a 1/2T entonces las versiones consecutivas del espectro original se traslapan y al sumarse producen una versión distorsionada del espectro original. A este fenómeno se le conoce como "aliasing".

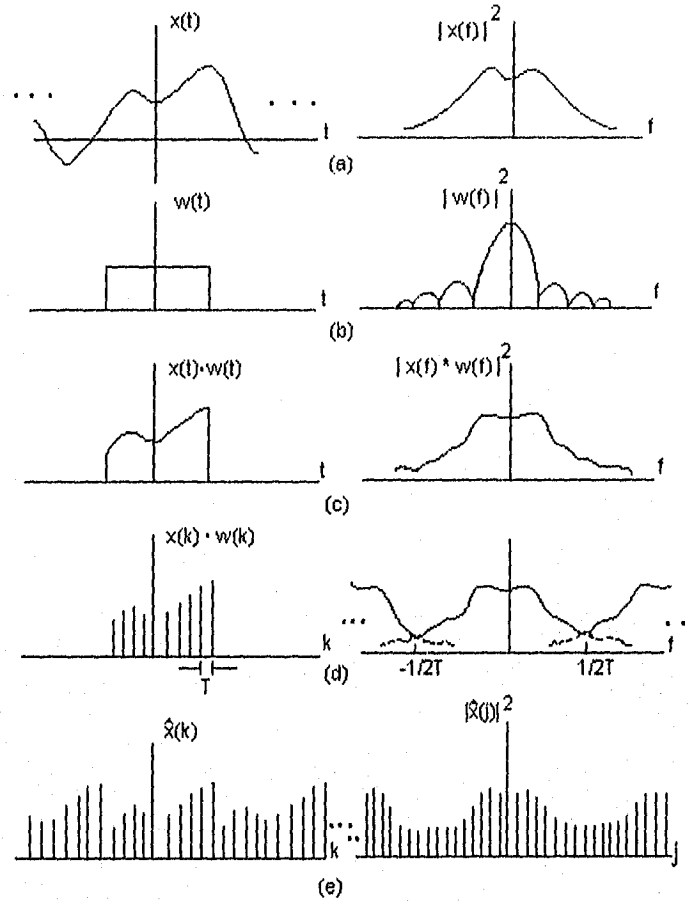



FIG. 2.4.1 **a)** Señal en tiempo y su espectro. **b)** Ventana regular y su espectro **c)** Señal obtenida al multiplicar la señal de a) por la ventana en b). **d)** Versión muestreada de c) **e)** repetición periódica de d)

Ahora consideremos la repetición periódica de $\hat{x}(n)$ de la señal muestreada como se indica en la fig. 2.4.1e. El espectro de esta señal X(k) es discreto, y es de hecho una versión muestreada del espectro de un periodo simple. Esto se muestra en la figura 2.4.1e. Ahora la DFT es simplemente un mapeo reversible de M términos de $\hat{x}(n)$ en M términos, ambos periódicos por lo tanto cualquiera de los dos conjuntos es suficiente para describir el otro.

Debe ser claro de este ejemplo que la DFT no es simplemente una versión discretizada del espectro de la señal continua de largo infinito original. Existen dos razones para esta desviación de la situación ideal.

(1) El hecho de que sólo una porción finita de la señal continua pueda ser examinada, nos lleva a la necesidad de usar una ventana, la cual causa una distorsión del espectro original.

(2) El aliasing por el muestreo de la señal.

La distorsión del espectro puede reducirse usando una ventana más ancha, esto es, viendo una mayor porción de la señal, ya que una ventana más ancha tiene el lóbulo principal más angosto. Sin embargo, la distorsión total del espectro no sólo depende del ancho del lóbulo principal, sino también de la altura de los lóbulos laterales. Esto puede reducirse usando, una ventana no rectangular.

Los efectos de aliasing se anulan completamente si la frecuencia de muestreo usada es dos veces mayor que la frecuencia máxima de la señal que es muestreada. Esto sólo es posible si la señal es de banda limitada o no contiene componentes de frecuencia muy alta. Podemos en una situación de banda no limitada muestrear a una tasa tan alta como sea posible para mantener los efectos de aliasing al mínimo.

Una de las propiedades más útiles de la DFT es el hecho de que la DFT inversa del producto de dos DFT's es la convolución de las dos series en el tiempo de las DFT's, es decir:

$$\text{IDFT}\{\ \text{DFT}\{x(n)\}\text{DFT}\{y(n)\}\ \} = x(n) * y(n)$$

Esta propiedad permite a la DFT (y de aquí a los algoritmos rápidos) ser usada para calcular la forma de onda de salida de un filtro a cualquier entrada.

Otra importante propiedad de la DFT es que es una transformación lineal tal que por ejemplo la DFT de la suma de dos funciones es la suma de las DFT's de las dos funciones.

Además es una transformación ortogonal de tal forma que la transformada inversa de Fourier discreta es igual a la transpuesta de la matriz de transformación directa.

### 2.4.3 ALGORITMO SPLIT-RADIX PARA EL CALCULO DE LA FFT

El algoritmo split-radix introducido por Duhamel, presenta varias ventajas sobre otros algoritmos, como por ejemplo:

a) No es necesario reordenar los datos dentro del algoritmo.

b) Se puede aplicar a secuencias complejas.

c) Presenta una reducción de redundancia aritmética en secuencias reales.

#### 2.4.3.1 DEFINICION DEL ALGORITMO.

Dado el cálculo de una FFT de una secuencia y(n), dada por Y(k) donde:

$$Y(k) = \sum_{n=0}^{M-1} y(n) W_M^{nk} \qquad 2.4.7$$

y

$$W_M^{nk} = \cos\left(\frac{nk2\pi}{M}\right) - j\,\mathrm{sen}\left(\frac{nk2\pi}{M}\right) \qquad 2.4.8$$

donde

$$W_M^{nk} = e^{2\pi nkj/M}$$

El algoritmo se basa en la descomposición en términos pares y términos impares, la cual se realiza en la siguiente forma:

para los términos pares:

$$Y(2k) = \sum_{n=0}^{M/2-1} \left[ y(n) + y(n+M/2) W_M^{2nk} \right] \qquad 2.4.9$$

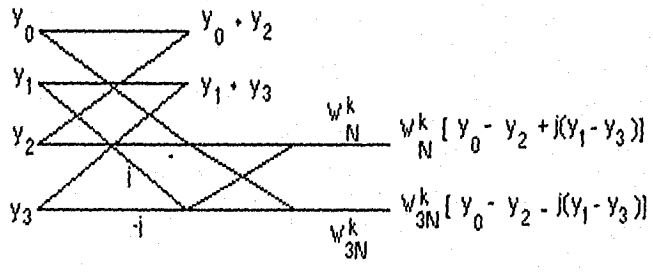
para los términos impares:

$$Y(4k+1) = \sum_{n=0}^{M/4-1} \left\{ \left[ y(n) + y(n+M/2) \right] - j\left[ y(n+M/4) - y(n+3M/4) \right] \right\} W_M^{n} W_M^{4nk} \qquad 2.4.10$$

$$Y(4k+3) = \sum_{n=0}^{M/4-1} \left\{ \left[ y(n) - y(n+M/2) \right] + j\left[ y(n+M/4) - y(n+3M/4) \right] \right\} W_M^{n} W_M^{4nk} \qquad 2.4.11$$

La idea básica del split-radix FFT (SRFFT) consiste en aplicar un mapa indexado radix-2 sobre los términos pares de un mapa indexado radix-4 sobre los términos impares ; se repite el proceso a cada una de las tres etapas resultantes, según las ecuaciones 2.4.9, 2.4.10 y 2.4.11, de forma similar a la decimación en frecuencia radix-2 del algoritmo Cooley-Tukey FFT.

La mariposa básica del SRFFT tiene la siguiente estructura:




El algoritmo avanza estado por estado en su parte superior, para cada una de las N etapas donde $N = \log_2 M$ ; mientras que en la parte inferior por ser radix-4, calcula dos

etapas a un mismo tiempo. El número de operaciones necesarias para el cálculo de una SRFFT para M=64, con una secuencia de entrada compleja, es de 964 sumas y 196 multiplicaciones.

### 2.4.4 SPLIT-RADIX FFT VALUADA REAL (SRFFT-RV)

Si la secuencia de entrada y(n) es real, existe una redundancia de operaciones, ya que Y(k) y Y(M-k) son complejos conjugados. Esto significa que a partir de las ecuaciones 2.4.10 y 2.4.11, resulta inútil calcular ambos coeficientes: Y(4k+1) y Y(4k+3), dado que:

$$Y(4k+3) = Y(M-(4k+1)) = Y^*(4k+1) \qquad 2.4.12$$

Para entender mejor el funcionamiento del algoritmo en cada etapa, se puede considerar la fig. 2.4.2 para una SRFFT y una SRFFT-RV de orden 64.

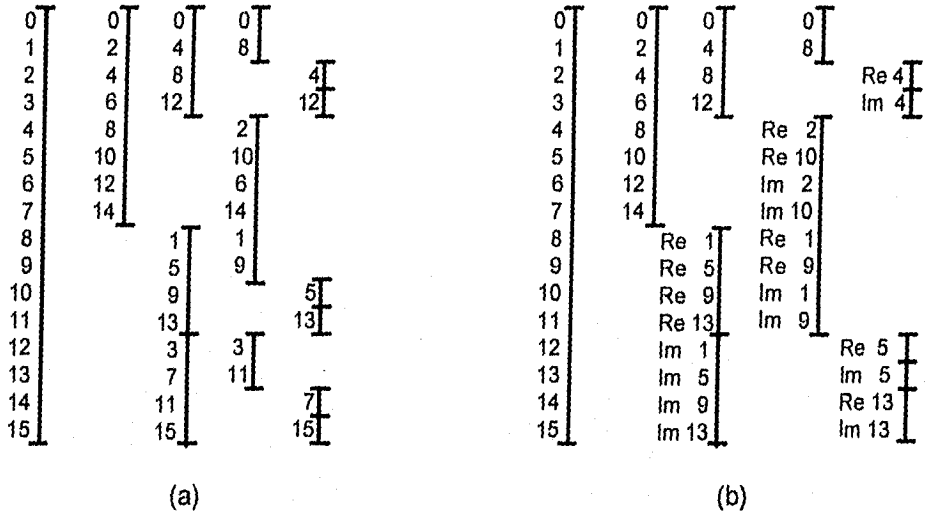



Fig. 2.4.2 Representación esquemática de la progresión del algoritmo SR de 16 puntos: (a) datos complejos, (b) datos reales

En la primera etapa se calcula un bloque de longitud M/2 de términos pares: Y(2k) para k=0,1,..,7; en la segunda etapa se calculan dos bloques de M/4: Y(4k+3) y

Y(4k+1) para k=0,1,..,3. Este proceso se aplica recursivamente a todos los bloques siguientes.

Cuando el algoritmo es utilizado con una secuencia real como en la fig. 2.4.2b, la primera etapa será una secuencia real, pero los dos bloques en la segunda etapa son complejos y como lo indica la ecuación 2.4.12, Y(4k+3) no necesita ser calculado, sin embargo su localización correspondiente en el diagrama puede ser usada para guardar la parte imaginaria del bloque Y(4k+1), este proceso es repetido hasta obtener la transformada correspondiente. Se debe notar que el algoritmo para este caso, debe distinguir entre mariposas reales y complejas. El número de multiplicaciones se reduce a 98 y el de sumas a 420 para un tamaño de M=64.

De acuerdo a la expresión 2.3.17b una DCT implica el cálculo de una FFT de una secuencia real, pero como lo indican las ecuaciones 2.3.19a y 2.3.19b, no hace falta calcular toda la FFT, basta con conocer solo la mitad de esta, para tal caso se utiliza la SRFFT-RV.

En lo que respecta al cálculo de una IDCT, no se puede evitar el hecho de obtenerla a partir de una secuencia compleja, lo cual nos obliga a utilizar el método SRFFT.

## 2.5 TRANSFORMADAS ORTOGONALES TRASLAPADAS

### 2.5.1 INTRODUCCION

Uno de los métodos más eficientes en la codificación de señales es la codificación por transformada, en este tipo de codificación una señal se mapea a un espacio vectorial donde la distribución de energía de la señal original se concentra en el menor número de coeficientes posteriormente estos coeficientes son cuantizados y transmitidos a una tasa de bits menor que la necesaria para transmitir la señal original. Al recibir la señal se decodifica y antitransforma reconstruyendo la señal original.

Como vimos anteriormente, los efectos del error de cuantización se minimizan cuando las funciones básicas de la transformada son el conjunto de vectores característicos de la matriz de autocovarianza de la señal de entrada, estos vectores definen la transformada Karhunen-Loève (KLT). La KLT es la que maximiza la concentración de energía en el mínimo número de coeficientes para cualquier nivel de error deseado y por lo tanto, esto conduce teóricamente a la mínima tasa de bits posible. En la práctica, la Transformada Coseno Discreta (DCT) se prefiere a la KLT, ya que la DCT es independiente de la señal de entrada y es una buena aproximación a la KLT para una gran variedad de señales con espectro paso bajas, y puede ser calculada por medio de algoritmos rápidos.

La codificación por medio de la DCT ha sido un método muy popular en la compresión de imágenes y de voz. Uno de los problemas básicos de la codificación por transformada a bajas tasas de bits y que aún no ha sido resuelto eficientemente es el llamado efecto de interbloqueo. El efecto de interbloqueo es una consecuencia natural del procesamiento independiente de cada bloque, éste se percibe en las imágenes como discontinuidades visibles alrededor de la unión de bloques adyacentes disjuntos.

Para disminuir el efecto de interbloqueo se han propuesto métodos tales como el de filtrado que consiste en aplicar un filtro paso bajas en las regiones colindantes de bloques vecinos, este método tiene la ventaja de que el procesamiento adicional sólo es en el receptor, sin embargo la señal pierde nitidez al filtrarse las frecuencias altas. Otro método es el de traslape el cual consiste en procesar bloques traslapados por medio de una transformación de orden mayor a la del bloque de la señal, transmitiendo así

información redundante para las regiones vecinas; en el receptor se aplica un algoritmo para reconstruir la señal alrededor de los bordes considerando la información de ambas transformaciones. En este procedimiento no hay pérdida de calidad en la señal, sin embargo, tiene la desventaja de incrementar la tasa de bits. Una forma más eficiente de reducir el efecto de interbloqueo la constituyen la familia de transformadas traslapadas ortogonales. Una transformada ortogonal traslapada consiste de una transformación donde el número de funciones básicas es menor al número de coeficientes que describen cada función, siendo así la matriz que transforma la señal una matriz de LxM donde L>M, de esta forma podemos obtener todos los beneficios del método de traslape sin aumentar la tasa de transmisión.

### 2.5.2 ESQUEMA DE CODIFICACION A TRAVES DE TRANSFORMADAS ORTOGONALES TRASLAPADAS

Como hemos visto, en el procesamiento de señales con transformadas de bloque tradicional, dividimos la señal de entrada en bloques de M muestras. Estas muestras son transformadas por medio de una matriz ortogonal de orden M; es decir, tanto el bloque de entrada como el orden de la matriz de transformación tienen el mismo tamaño.

Con una transformada traslapada mapeamos una muestra de entrada de tamaño L a M coeficientes de transformación donde L es la longitud de las funciones básicas de la transformada traslapada (LT) y M es el número de funciones básicas con L>M. Esta idea puede ser visualizada mediante la siguiente figura donde hemos asumido que L=2M.

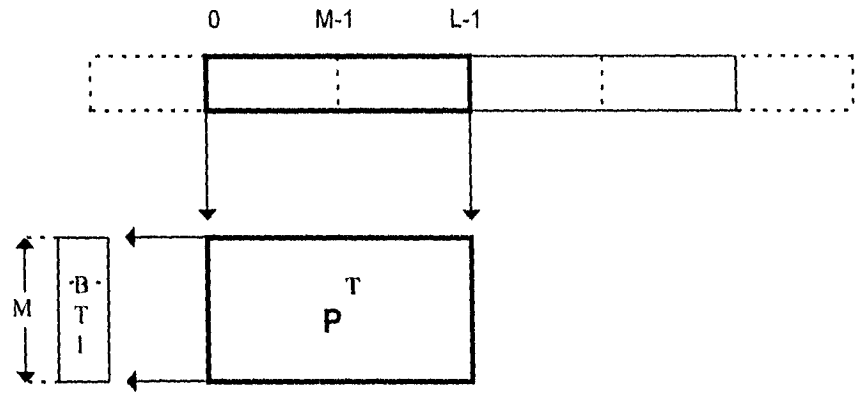



FIG.2.5.2.1a

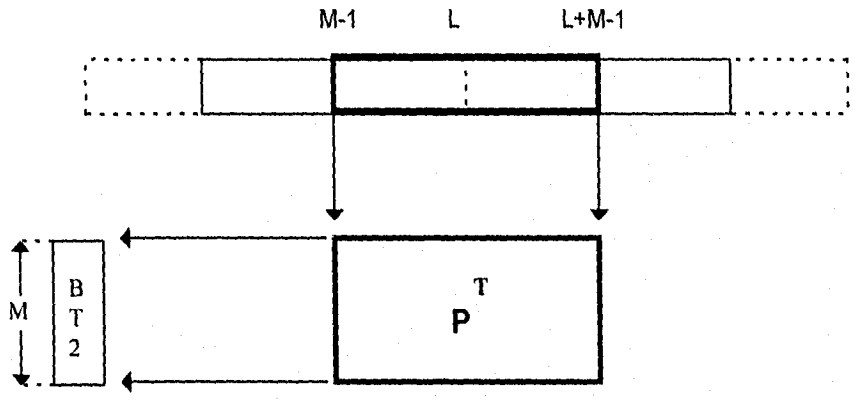



FIG. 2.5.2.1b. Transformada Ortogonal Traslapada.

Como podemos observar en la fig. 2.5.2.1a se procesan L muestras de entrada con la matriz P transpuesta de MxL de cuyos renglones son las funciones básicas de la LT, obteniéndose M coeficientes de transformación en el bloque BT1.

Si introducimos M nuevas muestras de la señal al buffer de entrada obtendremos M nuevos coeficientes transformados en BT2. De esta forma se realiza el traslape de L-M muestras y como podemos observar por cada M nuevas muestras que entran al buffer se

calculan M nuevos coeficientes de transformación por lo que la tasa de transmisión general del sistema no se altera.

La transformada inversa no es trivial como en el caso de las transformadas tradicionales. Al realizar la LT inversa de un bloque transformado de orden M obtendremos un bloque reconstruido de orden L y así para cada bloque, por lo que es necesario realizar un algoritmo de reconstrucción de las regiones traslapadas; como puede verse en la siguiente figura:

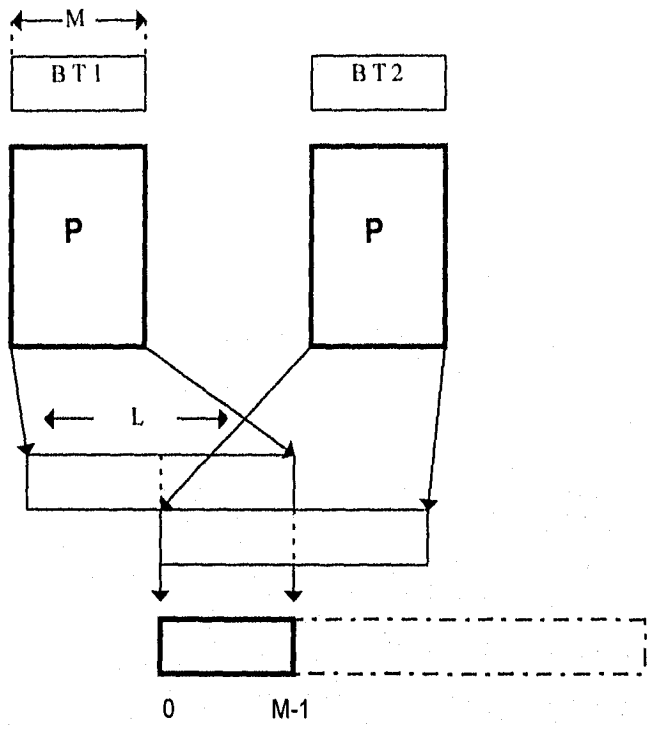



FIG. 2.5.2.2 Transformada Ortogonal Traslapada Inversa.

Podemos observar que no tiene sentido hablar de la LT inversa de un solo bloque ya que únicamente después del traslape de bloques reconstruidos consecutivos podremos recobrar la señal original.

La compresión es posible gracias a que en el dominio de la transformada la distribución de energía de la señal se concentra en los primeros coeficientes, por lo que al cuantizar estos con un mayor número de bits que los últimos podremos obtener una buena representación de la señal a una baja tasa de bits. Existen muchos métodos de

cuantización que buscan obtener la mejor representación de la señal a la menor tasa de bits, estos métodos pueden ser escalares o vectoriales, dependiendo si procesamos cada elemento de la señal independientemente o por vectores; pero de esto hablaremos en el capítulo siguiente.

### 2.5.3. DEFINICION Y PROPIEDADES BASICAS DE LAS TRANSFORMADAS ORTOGONALES TRASLAPADAS (LOT)

El objetivo de esta sección es describir las propiedades que debe cumplir una transformada para que pueda ser considerada una LOT.

Asumiremos por facilidad que las señales procesadas son unidimensionales; la extensión a dos o más dimensiones se puede establecer fácilmente definiendo transformadas separables. En el caso de imágenes la transformación se logra al aplicar inicialmente la transformada a los renglones de la imagen y posteriormente a las columnas.

Podemos asumir que la señal discreta de entrada es un segmento largo de NM muestras, donde M es el tamaño del bloque y N es el número de bloques que componen el segmento. En la codificación tradicional por transformada N bloques de longitud M deben ser transformados independientemente y codificados. En notación matricial si llamamos al vector original de entrada $x_0$ de longitud NM, el vector $y_0$ contiene los coeficientes de transformación de todos los bloques y esta dado por:

$$y_0 = T^T x_0 \qquad 2.5.3.1$$

Donde $T^T$ es la transpuesta de una matriz de bloques diagonal de NMxNM de la forma:

$$T = \begin{vmatrix} D & & & 0 \\ & D & & \\ & & D & \\ & & & \ddots \\ 0 & & & D \end{vmatrix} \qquad 2.5.3.2$$

Donde D es una matriz de orden M, cuyas columnas son las funciones básicas que definen la transformada de cada bloque.

Con la LOT, cada bloque tiene L muestras con L>M por lo tanto los bloques vecinos se traslapan L-M muestras.

La operación básica de la LOT es por lo tanto similar al método de traslape mencionado anteriormente. Una diferencia fundamental es que la LOT mapea las L muestras de cada bloque en M coeficientes de transformación por lo que no se incrementa la tasa de transmisión.

Como vimos en la sección anterior la LOT mapea L muestras de la señal de entrada en M coeficientes de transformación, por lo que la LOT puede ser definida de manera similar a 2.5.3.1 con T definida de la siguiente manera:

$$T = \begin{vmatrix} P_1 & & & 0 \\ & P_0 & & \\ & & P_0 & \\ & & & \ddots \\ 0 & & & P_2 \end{vmatrix} \qquad 2.5.3.3$$

Donde $P_0$ es una matriz de LxM que contiene las funciones básicas de la LOT para cada bloque. Hemos asumido que L=2M. Esta selección se justificará más tarde.

Las matrices $P_1$ y $P_2$ son introducidas debido a que el primero y último bloques de un segmento tienen únicamente un bloque vecino, y por lo tanto la LOT del primero y último bloque deberán definirse de una manera ligeramente diferente, para garantizar que ninguna de las funciones básicas se extienda más allá de los límites de la señal. Por el momento nos concentraremos en $P_0$.

Como podemos ver la LOT de un solo bloque no es invertible, dado que $P_0$ no es cuadrada. Sin embargo, para efectos de reconstrucción del segmento $x_0$, necesitamos garantizar la invertibilidad de T. Otra característica deseable es la ortogonalidad de T, dado que garantiza una buena estabilidad numérica y permite encontrar la transformada inversa a través de una simple transposición de la transformada directa.

Si consideramos $x_0$ como un segmento de la señal cuyo orden tiende a infinito y por lo tanto:

$$T = \begin{vmatrix} . & & & & 0 \\ & P_0 & & & \\ & & P_0 & & \\ & & & P_0 & \\ & & & & . \\ 0 & & & & . \end{vmatrix} \qquad 2.5.3.4$$

La ortogonalidad de T se garantiza si las columnas de $P_0$ son ortogonales, esto es:

$$P_0^T P_0 = I \qquad 2.5.3.5$$

además las funciones de traslape de bloques vecinos deberán también ser ortogonales, esto se define por:

$$P_0^T W P_0 = P_0^T W^T P_0 = 0 \qquad 2.5.3.6$$

donde I es la matriz identidad, y W es el operador de corrimiento definido por:

$$W = \begin{vmatrix} 0 & I \\ 0 & 0 \end{vmatrix} \qquad 2.5.3.7$$

La matriz identidad mencionada es de orden L-M. Veremos que una matriz LOT $P_0$ es factible si satisface las condiciones mencionadas en 2.5.3.5 y 2.5.3.6. Podemos ver que el conjunto de LOTs factibles es un superconjunto de las transformadas no traslapadas.

Junto con las condiciones de ortogonalidad requeridas, podríamos esperar propiedades adicionales para obtener una buena matriz LOT basándonos en nuestro conocimiento de la DCT y la KLT. Si una LOT factible tiene buena concentración de energía sus funciones básicas deben tener propiedades similares a las de la DCT o la KLT. Dos de estas propiedades parecen ser las más relevantes.

Primero, recordemos que la DCT es un buen sustituto de la KLT debido a que las funciones de la DCT aproximan los vectores característicos de la matriz de autocorrelación $R_{xx}$ de un proceso de Markov Gaussiano de primer orden.

$$R_{xx} = \begin{vmatrix} 1 & \rho & \rho^2 & . & . & \rho^L \\ \rho & 1 & \rho & . & . & \rho^{L-1} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \rho^{L-1} & . & . & \rho & 1 & \rho \\ \rho^L & . & . & \rho^2 & \rho & 1 \end{vmatrix}. \qquad 2.5.3.8$$

donde $\rho$ es el coeficiente de correlación intermuestral. Dado que la matriz mencionada es simétrica y Toeplitz, sus vectores característicos (los cuales definen la KLT) son simétricos o antisimétricos, es decir:

$$R_{xx}y = \lambda y \Rightarrow Jy = y \quad \text{ó} \quad Jy = -y \qquad 2.5.3.9$$

Donde J es el operador de identidad contraria definido por:

$$J = \begin{vmatrix} 0 & .. & .. & 0 & 1 \\ 0 & .. & 0 & 1 & 0 \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ 1 & 0 & .. & .. & 0 \end{vmatrix} \qquad 2.5.3.10$$

Dado que la mitad de los vectores característicos de $R_{xx}$ son simétricos, mientras que la otra mitad son antisimétricos; es razonable esperar que la LOT presente también este tipo de simetría, es decir, debe estar formada por M/2 vectores simétricos (pares) y M/2 vectores antisimétricos (nones).

Por otra parte, también podemos asumir que los vectores de menor orden (responsables de la mayor concentración de energía), son secuencias que varían muy lentamente, como por ejemplo, senoidales con bajas frecuencias, como lo son los vectores característicos de $R_{xx}$ para cualquier valor de $\rho$.

Este mismo comportamiento lo presentan las funciones básicas de la DCT.

### 2.5.4 OPTIMIZACION DE LAS TRANSFORMADAS ORTOGONALES TRASLAPADAS

En la sección anterior vimos las dos condiciones que debe cumplir una matriz para que pueda ser considerada LOT factible, también vimos algunas propiedades deseables en la LOT que aprovecharemos ahora para obtener una versión óptima en el sentido de máxima ganancia de codificación.

Una LOT óptima debe minimizar la tasa de bits para cualquier nivel de error de reconstrucción, asumiendo que el modelo de Markov en 2.5.3.8 es aplicable, esto es equivalente a maximizar la compactación de energía (también llamada máxima ganancia de codificación por transformada)

$$G_{TC} = \frac{\frac{1}{M}\sum_{i=1}^{M}\sigma_i^2}{\left(\prod_{i=1}^{M}\sigma_i^2\right)^{\frac{1}{M}}} \qquad 2.5.4.1$$

Donde $\sigma_i^2$ es la i-ésima entrada de la diagonal de la matriz:

$$R_0 = P_0^T R_{xx} P_0 \qquad 2.5.4.2$$

### 2.5.4.1 OPTIMIZACION RECURSIVA

Para maximizar $G_{TC}$ Cassereau [P.Cassereau, "A new class of optial unitary transforms for image processing", M. thesis Dep. Elec. Eng. Comput. Sci. Mass. Inst. Technol. Cambridge MA. May 1983] propuso el método de optimización recursiva, el cual se basa en la obtención iterativa de las columnas de $P_0$. La LOT que se obtiene del algoritmo mencionado depende del espectro de la señal asumida, ya que usa la matriz de covarianza $R_{xx}$; por lo tanto, no existe una solución general. Además este tipo de optimización tiene fuertes no linealidades por lo que la solución podría converger en un mínimo local. Aún si el algoritmo fuera finalmente ajustado para minimizar el problema de convergencia a mínimos locales podría conducir a una matriz LOT sin ninguna estructura especial, por lo que no podríamos obtener un algoritmo rápido limitando así sus aplicaciones prácticas. Por otro parte es un procedimiento altamente sensible a errores numéricos aún usando cálculos de doble precisión.

### 2.5.4.2 LOTs CUASIOPTIMAS

Para hacer un diseño de LOT más robusto sería interesante poder evitar los pasos de optimización no lineal. Un camino para aproximarnos a este objetivo es trabajar dentro de un subespacio de todas las matrices LOT posibles. Una LOT óptima dentro de tal subespacio no necesariamente es globalmente óptima pero puede ser fácil de diseñar y tiene un algoritmo rápido.

Presentaremos una aproximación directa para la derivación de una LOT óptima cuando L=2M esta aproximación es virtualmente insensible a errores numéricos y permite una mejor comprensión de la LOT, ya que podemos derivar un algoritmo rápido. La base de esta aproximación es comenzar con una matriz LOT P factible que no necesariamente debe ser óptima. Entonces la matriz

$$P_0 = PZ \qquad 2.5.4.3$$

es también una LOT factible para cualquier Z ortogonal dado que:

$$P_0^T P_0 = Z^T P^T P Z = Z^T Z = I \qquad 2.5.4.4$$

$$P_0^T W P_0 = Z^T P^T W P Z = 0 \qquad 2.5.4.5$$

Podemos definir una LOT factible a partir de la DCT, por

$$P = \frac{1}{2}\begin{vmatrix} D_e - D_o & D_e - D_o \\ J(D_e - D_o) & -J(D_e - D_o) \end{vmatrix} \qquad 2.5.4.6$$

Donde De y Do son las matrices de MxM/2 compuestas de las funciones básicas de la DCT pares y nones respectivamente. Es fácil verificar la factibilidad de P sustituyéndola en las ecuaciones 2.5.3.5 y 2.5.3.6.

Con P definida como se muestra en 2.5.4.6, para obtener una LOT óptima es necesario encontrar la matriz Z a la que se hace referencia en 2.5.4.3. Así tenemos que sustituyendo 2.5.4.3 en 2.5.4.2, obtenemos:

$$R_0 = Z^T P^T R_{xx} P Z \qquad 2.5.4.7$$

Con P y $R_{xx}$ fijas es claro que $G_{TC}$ se maximiza cuando $R_0$ es diagonal, es decir, cuando las columnas de Z son los vectores característicos de $P^T R_{xx} P$ con una Z tal, la matriz LOT $P_0$ es óptima.

Es importante señalar que el método de optimización conduce a una LOT óptima ligada a la selección de la matriz inicial P. Dado que cada columna de P tiene L elementos, con L>M generan un subespacio M dimensional de $R^L$. Para cualquier Z la matriz PZ siempre caerá dentro de tal subespacio y dentro de este subespacio encontraremos la LOT óptima. Sin embargo puede existir una matriz LOT factible que no esté dentro del subespacio generado por las columnas de P es decir no puede ser generada por la función 2.5.4.3. Por lo tanto una LOT óptima derivada del procedimiento anterior podría no ser globalmente óptima en el sentido de maximizar la compactación de energía. Sin embargo, como veremos después, nuestra elección de P en 2.5.4.6 es suficientemente buena considerando que se obtiene la misma compactación de energía

que las funciones de Cassereau las cuales fueron diseñadas para ser globalmente óptimas.

Las funciones obtenidas con este método no son muy sensitivas a variaciones de $\rho$ tal que los resultados para $\rho$=0.8 son virtualmente iguales que para $\rho$=0.95. Las restricciones de ortogonalidad para las funciones a lo largo de los bloques vecinos conducen a funciones básicas que decaen a cero en sus bordes. Así, la discontinuidad a cero en los bordes es mucho menor que en las funciones DCT, esta es una de las principales razones por la que se reducen los efectos de interbloqueo.

Existen dos propiedades básicas de la LOT que son una consecuencia directa de la selección de L=2M. Primero, si las funciones básicas de menor orden para un grupo de bloques consecutivos se sobrepone, la secuencia resultante tiene un valor de DC. Esta es una característica deseable e importante ya que implica que un campo plano puede ser reproducido con un solo coeficiente de transformación por bloque. Si L fuera menor que 2M, la reconstrucción perfecta de una señal de DC con un solo coeficiente por bloque no sería posible. Segundo el hecho de que los bordes de la derecha de las funciones básicas para un bloque r sean inmediatamente adyacentes a los bordes de la izquierda de las funciones para un bloque r+2 evita discontinuidades entre los bloques producidas por los límites de las funciones básicas. Por lo tanto la elección de L=2M es una buena selección.

El factor Z de la matriz LOT óptima Po = PZ puede no ser factorizable en M log(M) estados de mariposas esto es exactamente la misma deficiencia de la KLT óptima para codificación por bloques sin traslape. En la siguiente sección veremos una aproximación a la LOT óptima que puede ser implementada a través de algoritmos rápidos tal como la DCT es la aproximación rápidamente calculable de la KLT.

### 2.5.5 ALGORITMOS RAPIDOS

La base de los algoritmos rápidos para la LOT es la aproximación de la matriz Z por un producto de algunos factores simples. Esta es la principal razón por la que se escogieron las funciones básicas de la DCT en la definición de P dada en 2.5.4.6. Tal definición nos permite obtener una expresión útil para la matriz $P^T R_{xx} P$. Para simplificar la notación nos

referiremos a la matriz de autocorrelación de un proceso de Markov de primer orden dada en 2.5.3.8 como R(2M, $\rho$), donde el primer parámetro representa el orden de la matriz. Podemos relacionar R(2M, $\rho$) a R(M, $\rho$) por medio de la siguiente ecuación:

$$R(2M,\rho) = \begin{vmatrix} R(M,\rho) & B \\ B^T & R(M,\rho) \end{vmatrix} \qquad 2.5.5.1$$

donde $B = \rho Jrr'$ y $r = [1, \rho, \rho^2, \ldots, \rho^{M-1}]$.

Combinando 2.5.4.6 y 2.5.5.1, obtenemos después de algunas manipulaciones

$$R_0 = \begin{vmatrix} R_1 & 0 \\ 0 & R_2 \end{vmatrix} \qquad 2.5.5.2$$

donde los bloques en la diagonal $R_1$ y $R_2$ están dados por:

$$R_1 = D_e'R(M,\rho)D_e + D_o'R(M,\rho)D_o + \rho D_e'rr'D_e + \rho D_o'rr'D_o \qquad 2.5.5.3$$

y

$$R_2 = D_e'R(M,\rho)D_e + D_o'R(M,\rho)D_o - \rho D_e'rr'D_e - \rho D_o'rr'D_o \qquad 2.5.5.4$$

Si aproximamos el coeficiente de autocorrelación $\rho$ a la unidad, las matrices $D_e$ y $D_o$ contendrán los vectores característicos asintóticos pares y nones de $R(M,\rho)$ respectivamente, dado que la DCT es el límite de la KLT cuando $\rho$ tiende a 1. Por lo tanto el término $D_e'R(M,\rho)D_e$ y $D_o'R(M,\rho)D_o$ son asintóticamente diagonales, con entradas positivas. También como $\rho \to 1$, el vector r tendrá todas sus entradas iguales a 1, es decir, será un vector par. Por lo tanto, el término $D_o'rr'D_o$ tenderá a cero. Además el factor $D_e'rr'D_e$.

$$D_e'rr'D_e \Rightarrow \begin{vmatrix} M & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 \end{vmatrix} \qquad 2.5.5.5$$

Así, es claro que $R_1$ será asintóticamente una matriz diagonal con las entradas de la diagonal positivas. Sin embargo el factor $R_2$ puede no tener una diagonal dominante porque el tercer término en 2.5.5.4 se resta de los otros. Al menos podemos esperar la siguiente aproximación:

$$Z = \begin{vmatrix} I & 0 \\ 0 & \hat{Z} \end{vmatrix} \qquad 2.5.5.6$$

Donde $\hat{Z}$ es de orden M/2. Aunque $R_2$ puede no tener una diagonal fuertemente dominante podemos esperar alguna diagonal dominante, tal que $\hat{Z}$ no debe diferir demasiado de la matriz identidad. Se ha demostrado que $\hat{Z}$ puede ser aproximada por medio de una cascada de M/2 -1 planos de rotación, de la forma:

$$\hat{Z} = T_1 T_2 \dots\dots T_{\frac{M}{2}-1} \qquad 2.5.5.7$$

Donde cada plano de rotación se define como:

$$T_i = \begin{vmatrix} I & 0 & 0 \\ 0 & Y(\theta_1) & 0 \\ 0 & 0 & I \end{vmatrix} \qquad 2.5.5.8$$

La matriz $Y(\theta_1)$ es una mariposa de 2x2

$$Y(\theta_i) = \begin{vmatrix} \cos\theta_i & \text{sen}\theta_i \\ -\text{sen}\theta_i & \cos\theta_i \end{vmatrix} \qquad 2.5.5.9$$

Donde $\theta_i$ es el ángulo de rotación, y el factor de identidad en la esquina superior izquierda de la expresión 2.5.5.8 es de orden i-1. Si aplicamos la traspuesta de cada $T_i$ a $\hat{Z}$ en el orden inverso de 2.5.5.8, debemos obtener una aproximación muy cercana a la matriz identidad. Para N = 16 y $\rho$ = 0.95 un conjunto apropiado de ángulos es: $(\theta_1\theta_2.....\theta_7)$ = [0.42 0.53 0.5 0.44 0.35 0.23 0.11]. Con estos ángulos la compactación de energía es $G_{TC}$ = 9.32, el cual es muy cercano al valor de $G_{TC}$ = 9.49 correspondiente a la solución exacta, la pérdida en ganancia de codificación usando la aproximación dada en 2.5.5.7 es únicamente de 0.08 dB. La compactación de energía para una DCT de tamaño 16 es $G_{TC}$ = 8.82 por lo tanto la LOT nos da una mejora de 0.32 dB en el error de reconstrucción rms.

Es importante notar que la aproximación de $\hat{Z}$ por una cascada de M-1 mariposas es satisfactoria para M pequeñas. Cuando M es mayor o igual a 32 la aproximación puede introducir pequeñas discontinuidades en las funciones básicas de menor orden, las cuales pueden notarse como pequeñas imperfecciones en la señal reconstruida. La aproximación de $\hat{Z}$ para M > 16 se analizará en la siguiente sección.

La LOT rápida esta definida por $P_0$ en 2.5.4.3 con P dada por 2.5.4.6 y Z por 2.5.5.6 a 2.5.5.9. La $P_0$ resultante puede también ser escrita como:

$$P = \frac{1}{2}\begin{vmatrix} D_e & D_o & 0 & 0 \\ 0 & 0 & D_e & D_o \end{vmatrix} \begin{vmatrix} I & I & & 0 \\ I & -I & & \\ & & I & I \\ 0 & & I & -I \end{vmatrix} \begin{vmatrix} 0 & 0 \\ I & I \\ I & -I \\ 0 & 0 \end{vmatrix} \begin{vmatrix} I & 0 \\ 0 & \hat{Z} \end{vmatrix} \qquad 2.5.5.10$$

El diagrama de flujo correspondiente a la matriz anterior, para M = 8 se muestra en la figura 2.5.5.1

Podemos notar que el diagrama de flujo puede ser usado tanto para la LOT directa como para la inversa, por medio de la transposición como se hace con todas las transformadas ortogonales. Aunque el diagrama de flujo de la figura 2.5.5.1 parece indicar que necesitamos calcular 2 DCTs de tamaño M para obtener M coeficientes LOT esto no es necesario como veremos más adelante.

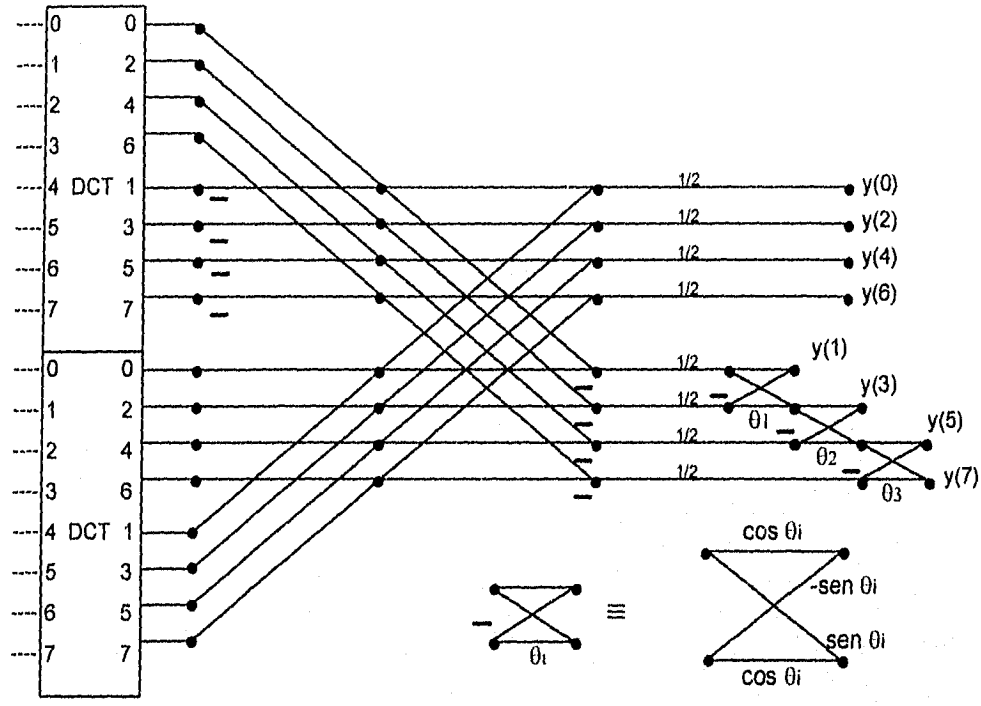



FIG. 2.5.5.1 Diagrama de Flujo para una LOT rápida con M=8

Ahora regresamos al punto en el que necesitamos calcular la LOT del segmento completo $x_0$ y también las relaciones entre $P_0$, $P_1$ y $P_2$ en 2.5.3.3. A partir de la figura anterior es claro que la DCT usada en el bloque r puede también ser usada en parte por los bloques r-1 y r+1, como se muestra en la figura 2.5.5.2

La LOT del primero y último bloques $P_1$ y $P_2$ son obtenidas reflejando los datos de manera par en los bordes del segmento. Esto es equivalente a usar el bloque etiquetado como $H_e$ en la figura 2.5.5.2 donde $H_e$ es la matriz que contiene la mitad de las muestras de las funciones DCT pares, esto es:

$$D_e = \begin{vmatrix} H_e \\ JH_e \end{vmatrix} \qquad 2.5.5.11$$

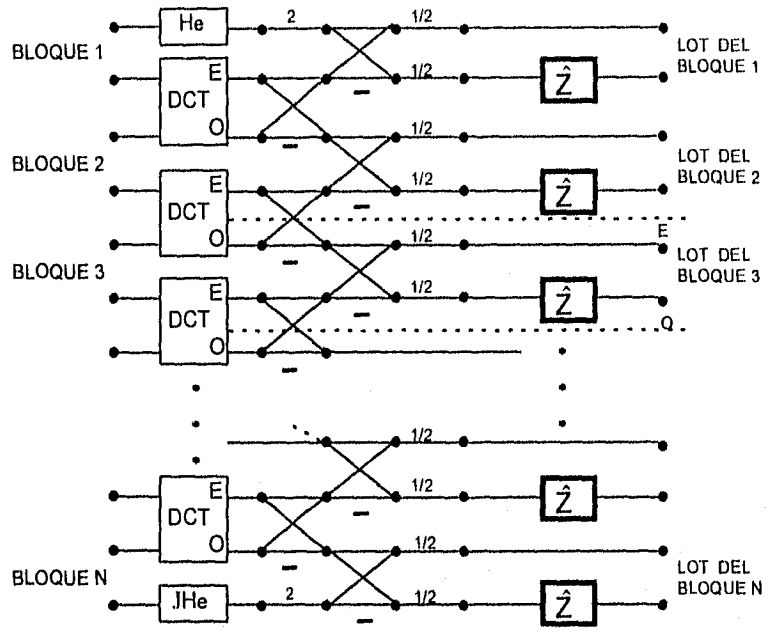



Fig. 2.5.5.2. LOT rápida de N bloques.

Notamos que la LOT de un segmento de datos de N bloques de tamaño M puede ser calculado obteniendo la DCT de todos los bloques como en una codificación por transformada tradicional, y entonces aplicamos las mariposas +1 -1 y $\hat{Z}$ de las figuras 2.5.5.1 y 2.5.5.2.

### 2.5.6 LOT RAPIDA PARA M > 16

Una LOT más general que la descrita anteriormente puede ser obtenida por medio de una factorización diferente a la que vimos anteriormente de la matriz $\hat{Z}$ la cual es:

$$\hat{Z} = D_1' D_2 \qquad 2.5.5.12$$

Donde $D_1$ es la matriz cuyas columnas son las funciones básicas de la DCT de longitud M/2 y $D_2$ es matriz de la transformada seno discreta tipo IV. Note que la matriz $D_1$ aparece transpuesta, y esto corresponde a una DCT inversa en el cálculo de la LOT directa y a una DCT directa en el cálculo de una LOT inversa. El cálculo de la LOT se basa en los algoritmos rápidos de la DCT y de la DCT - IV.

El diagrama de flujo de esta LOT rápida a la que nos referiremos como LOT tipo II se muestra en la figura 2.5.5.3 para la transformada directa. El diagrama de flujo para la transformada inversa correspondiente puede obtenerse fácilmente transponiendo el diagrama de flujo y moviendo las unidades de retardo.

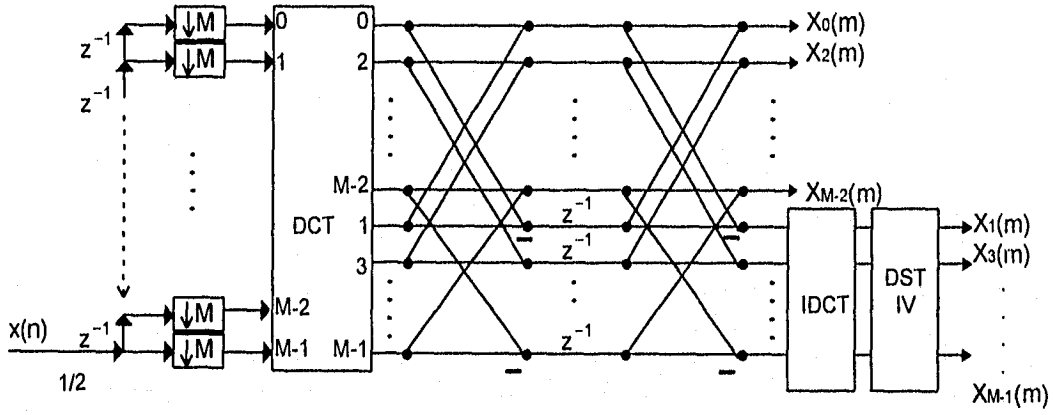



FIG. 2.5.5.3 Transformada Ortogonal Traslapada tipo II.

# CAPITULO 3

## CUANTIZACION VECTORIAL

### 3.1. INTRODUCCION.

Como mencionamos en capítulos anteriores, la técnica de compresión de imágenes por medio de transformadas; nos dice que después de haber aplicado la transformada a la imagen obtenemos datos que necesitan ser cuantizados en amplitud para poder ser transmitidos o almacenados en forma óptima, esta codificación se lleva a cabo por medio de un sistema de cuantización (cuantizador) que puede ser escalar o vectorial.

Un cuantizador vectorial (VQ) es un sistema para mapear una secuencia de vectores continuos o discretos a una secuencia digital adecuada para ser transmitida, o almacenada sobre un canal digital. El objetivo de este sistema es minimizar los requerimientos de ancho de banda del canal de transmisión o de memoria para el almacenamiento digital, manteniendo una adecuada fidelidad de la información.

El mapeo para cada vector podría tener o no tener memoria en el sentido de depender o no de las acciones pasadas del codificador; durante el desarrollo del presente capítulo nos enfocaremos al estudio de los cuantizadores vectoriales sin memoria.

### 3.2 CUANTIZACION

El proceso de cuantización vectorial consiste en dos mapeos: el que hace el codificador $\gamma$, el cual asigna a cada vector de entrada $x = (x_0, x_1, \ldots, x_{k-1})$ un símbolo de canal $\gamma(x)$ en algún conjunto de símbolos de canal M, y el que hace el decodificador $\beta$ asignando a cada símbolo de canal $\nu$ en M un valor en el alfabeto

de reproducción A. Por conveniencia el conjunto de símbolos de canal regularmente es un espacio de vectores binarios.

El alfabeto de reproducción podría ser o no el mismo espacio vectorial que a la entrada; en particular podría consistir de vectores reales de una dimensión diferente.

Si M tiene m elementos, entonces la cantidad $R = \log_2(m)$ es conocida como la tasa del cuantizador en bits por vector y r=R/k es la tasa de bits por símbolo, donde k es la dimensión de los vectores de entrada; de acuerdo con esto, podemos observar que los cuantizadores vectoriales permiten tasas fraccionarias en bits por símbolo, ya que si representamos el símbolo de canal por medio de un solo bit tendríamos una tasa de 1/k bpp para vectores k dimensionales.

La aplicación de un cuantizador para compresión de datos se muestra en forma general en la fig. 3.2.1. Los datos o fuente de información $\{X_n; n = 0,1,..\}$ son una secuencia de vectores consecutivos, que son procesados por medio de un codificador. El codificador produce una secuencia de símbolos de canal $\{U_n; n = 0,1,...\}$; posteriormente la secuencia $\{\hat{U}_n; n = 0,1,...\}$ llega al receptor a través de un canal digital. Finalmente el decodificador mapea esta secuencia a la secuencia reconstruida $\{\hat{X}_n; n = 0,1,..\}$, considerando un canal sin ruido, ya que el teorema de unión entre la codificación de fuente y de canal de la teoría de la información implica que un buen sistema de compresión de datos diseñado para un canal sin ruido puede ser combinado con un buen sistema de corrección de errores para un canal ruidoso y producirá un buen sistema completo, lo que nos permite hacer la consideración ideal de un canal sin ruido para enfocarnos al problema del diseño del sistema de compresión de datos.

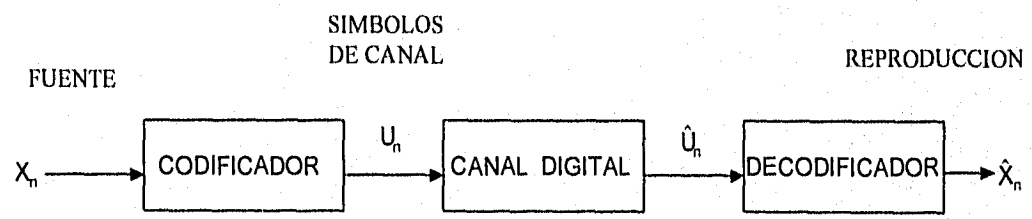



FIG. 3.2.1 Sistema de Compresión de Datos

## 3.3 MEDIDAS DE DISTORSION

El objetivo de un sistema de cuantización es reproducir lo mejor posible una secuencia de datos (que en nuestro caso serán imágenes) a una tasa dada; por lo tanto, para poder definir el desempeño de un cuantizador, debemos establecer el concepto de medida de distorsión.

Una medida de distorsión d es la asignación de un costo $d(x,\hat{x})$ por reproducir cualquier vector x como un vector de reproducción $\hat{x}$. Dada tal medida de distorsión podemos cuantificar el desempeño del cuantizador por la distorsión promedio $E(d(X,\hat{X}))$ entre la entrada y la reproducción final; de tal forma que, un sistema será bueno si produce una distorsión promedio pequeña.

Idealmente una medida de distorsión debe ser: tratable para permitir su análisis, computable, de tal forma que pueda ser evaluada en tiempo real y subjetivamente significativa de manera que una mayor o menor medida cuantitativa de distorsión implique un sistema con mejor o peor calidad, hablando en forma subjetiva.

Una de las medidas de distorsión más comúnmente utilizada y más simple en cuanto a cálculo es la medida de distorsión del error cuadrático; en donde la distorsión se calcula por medio de la distancia entre los vectores de entrada y de reproducción.

$$d\left(X,\hat{X}\right)=\left\|X-\hat{X}\right\|^2=\sum_{i=0}^{k-1}\left(X_i-\hat{X}_i\right)^2 \qquad 3.3.1$$

Como mencionamos anteriormente, nuestro elemento para medir el desempeño de un sistema es la distorsión promedio. En la práctica, la distorsión promedio es el promedio de muestras en un periodo largo de tiempo:

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}d\left(X_i,\hat{X}_i\right) \qquad 3.3.2$$

Si el proceso es estacionario y ergódico, entonces este promedio que restringe el tiempo de valuación es el mismo que la esperanza matemática $E\left(d\left(X,\hat{X}\right)\right)$, sin

embargo, generalmente las distribuciones de probabilidad no son conocidas, por lo que no es posible calcular la esperanza matemática en forma práctica. De aquí tenemos que una aproximación válida para el diseño de sistemas sea tomar una secuencia de información de entrenamiento muy larga , y estimar la distorsión mediante el cálculo del promedio de muestras para después intentar diseñar un código que minimice la distorsión promedio de las muestras para la secuencia de entrenamiento.

Entonces se usa el código para secuencias de examinación producidas por la misma fuente y que son diferentes de la secuencia de entrenamiento. Si el desempeño para las secuencias de entrenamiento y examinación es significativamente diferente, entonces la secuencia de entrenamiento no fue lo suficientemente larga; por lo que habrá que escoger otra. En lo sucesivo, se usará la notación de esperanza matemática como una abreviación para promedios de muestras en tiempo promedio.

## 3.4 PROPIEDADES DE LOS CUANTIZADORES OPTIMOS

Un cuantizador vectorial es óptimo si minimiza la distorsión promedio; para esto es necesario que cumpla con ciertas propiedades. Antes de mencionar estas propiedades es preciso que establezcamos algunos conceptos: La colección de vectores de reproducción posibles $C = \{y: y = \beta(\nu)\}$, para algún $\nu$ en M , es llamado "codebook" de reproducción o simplemente "codebook" del cuantizador y sus miembros son llamados "codewords" (o "templates" o patrones). El codificador conoce la estructura del decodificador y de aquí todos los patrones de salida finales.

Propiedad 1: Teniendo como objetivo minimizar la distorsión promedio, y dado un decodificador específico $\beta$, el mejor codificador $\gamma$ es aquel que selecciona el patrón $\nu$ en M que produce la mínima distorsión posible a la salida. Esto es:

$$\gamma(x) = \min_{\nu \in M} d(x, \beta(\nu)) \qquad 3.4.1$$

Podemos pensar en un codificador $\gamma$ como la partición del espacio de entrada en celdas, donde todos los vectores de entrada que producen una reproducción común son

agrupados juntos. Cuando esta partición sigue la regla de mínima distorsión se llama partición de Voronoi o de Dirichlet.

Propiedad 2: Dado un codificador $\gamma$, entonces, no puede haber mejor decodificador que aquel que asigna a cada símbolo de canal $\nu$ el centroide generalizado (o baricentro o centro de gravedad) de todos los vectores fuente codificados en $\nu$.

El centroide en el caso de una distribución muestral y una distorsión de error cuadrático medio es el centroide Euclidiano o vector suma de todos los vectores codificados a un símbolo de canal dado, esto es, dada la distribución muestral definida por la secuencia de entrenamiento $\{x_i; i = 0..L-1\}$, entonces

$$\text{cent}(\nu) = \left(\frac{1}{i(\nu)}\right) \sum_{x_i;\gamma(x_i)=\nu} x_i \qquad 3.4.2$$

donde $i(\nu)$ es el número de índices i para el cual $\gamma(x_i) = \nu$.

Los nuevos "codewords" representan mejor los vectores de entrenamiento mapeados por los antiguos "codewords", pero producen una diferente partición de distorsión mínima del alfabeto de entrada. Esta es precisamente la base para el algoritmo de generación del "codebook"; y que trataremos en la siguiente sección.

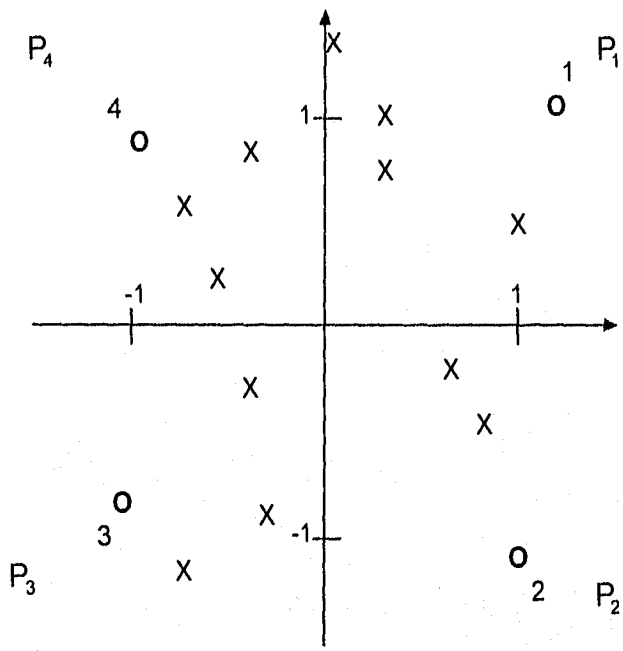
## 3.5 ALGORITMO DE LLOYD-MAX VECTORIAL

De acuerdo con lo visto en la sección anterior, podemos observar que el "codebook" se puede optimizar iterativamente, tomando en cuenta el codificador anterior y usando el codificador de distorsión mínima para el nuevo "codebook". El hecho de que el decodificador pueda ser optimizado considerando al codificador y viceversa forma la base del algoritmo de diseño para PCM óptimo general de Lloyd para una variable escalar con función de densidad de probabilidad conocida y distorsión de error cuadrático medio. Los algoritmos de diseño generalizados para cuantizadores vectoriales considerados aquí se basan en la observación de que el desarrollo básico de Lloyd es

válido para vectores, para distribuciones muestrales y para una variedad de medidas de distorsión. El único requerimiento es que se pueda calcular el centroide. El algoritmo básico es el siguiente:

Paso 1: Dar una secuencia de entrenamiento y un decodificador inicial.

Paso 2: Codificar la secuencia de entrenamiento en una secuencia de símbolos de canal usando el decodificador dado y la regla de mínima distorsión, como se muestra en la fig. 3.5.1. Si la distorsión promedio es pequeña, salir.

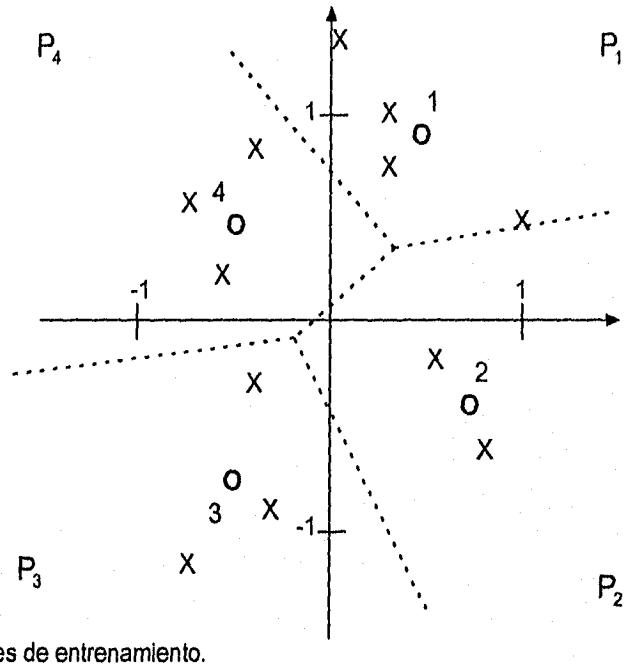



X : Vectores de entrenamiento.
O : "Codewords"
$P_i$ : Región codificada con el "codeword" i

FIG. 3.5.1 Partición bidimensional por mínima distorsión. Cada vector de entrada se mapea dentro del "codeword" más cercano; es decir, el círculo en el mismo cuadrante.

Paso 3: Reemplazar el antiguo "codeword" de reproducción del decodificador para cada símbolo de canal v por el centroide de todos los vectores de entrenamiento que fueron mapeados en v en el paso 2, como lo indica gráficamente la fig. 3.5.2 . Ir al paso 2.

Cada ciclo del algoritmo deberá reducir la distorsión promedio o la deberá mantener igual. Regularmente el algoritmo se detiene cuando la distorsión se decrementa hasta estar por debajo de un umbral pequeño. Debido a los trabajos que desarrollaron Linde, Buzo y Gray aplicando el algoritmo a la cuantización vectorial, también se le ha denominado algoritmo LBG.




X : Vectores de entrenamiento.
O : "Codewords"
$P_i$ : Región codificada con el "codeword" i

FIG. 3.5.2 Centroides de la fig. 3.5.1. El cálculo de los centroides, permite mover los "codewords", de tal forma que representen mejor a los vectores de entrada. La línea punteada muestra la nueva partición para los nuevos "codewords".

## 3.6 CODEBOOKS INICIALES.

De acuerdo con el procedimiento del algoritmo de diseño básico de la sección anterior, observamos que se trata de un algoritmo de mejora iterativa a partir de un "codebook" inicial. Para establecer este "codebook" inicial, podemos iniciar con un "codebook" del tamaño requerido, o bien empezar con un "codebook" pequeño para ir construyendo uno mayor en forma recursiva. A continuación se darán algunas técnicas para poder encontrar el "codebook" inicial.

### 3.6.1. CODIGOS ALEATORIOS.

Consiste en utilizar los primeros $2^R$ vectores de la secuencia de entrenamiento como "codebook" inicial; o bien seleccionar vectores más ampliamente espaciados para tomar un " codebook" más característico.

### 3.6.2 CODIGOS PRODUCTO.

Aquí se utiliza un código escalar como si fuera un cuantizador uniforme k veces en sucesión y se corta el "codebook" vectorial resultante al tamaño correcto. El modelo matemático para esta aproximación es un código producto, el cuál se define de la siguiente forma: decimos que tenemos una colección de "codebooks" $C_i$, i=0,1,..,m-1, cada uno consistente de $M_i$ vectores de dimensión $k_i$, con una tasa de $R_i = \log_2(M_i)$ bits por vector. Entonces el "codebook" producto está definido como la colección de todas las $M = \prod_i M_i$ posibles concatenaciones de m palabras puestas sucesivamente de los m "codebooks" $C_i$. La dimensión k del "codebook" producto es la suma de las dimensiones de los componentes de los "codebooks". El código producto se denota matemáticamente como el producto cartesiano:

$$C = \bigtimes_{i=0}^{m-1} C_i = \{\text{todos los vectoresde la forma } (\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{m-1}); \quad \hat{x}_i \in C_i;\ i=0,1,..,m-1\} \qquad 3.6.1$$

De esta forma tenemos que, usando un cuantizador escalar con tasa R/k, k veces en sucesión, obtendremos un cuantizador vectorial producto k dimensional con una tasa de R bits por vector. Este código producto puede utilizarse como código inicial para el algoritmo de diseño.

### 3.6.3 CODIGOS POR ROMPIMIENTO.

Para encontrar el "codebook" inicial por medio de la técnica de rompimiento, empezamos calculando el centroide de la secuencia de entrenamiento total, que corresponde al "codebook" óptimo a una tasa de 0 bits por vector, representado en la fig. 3.6.1a.

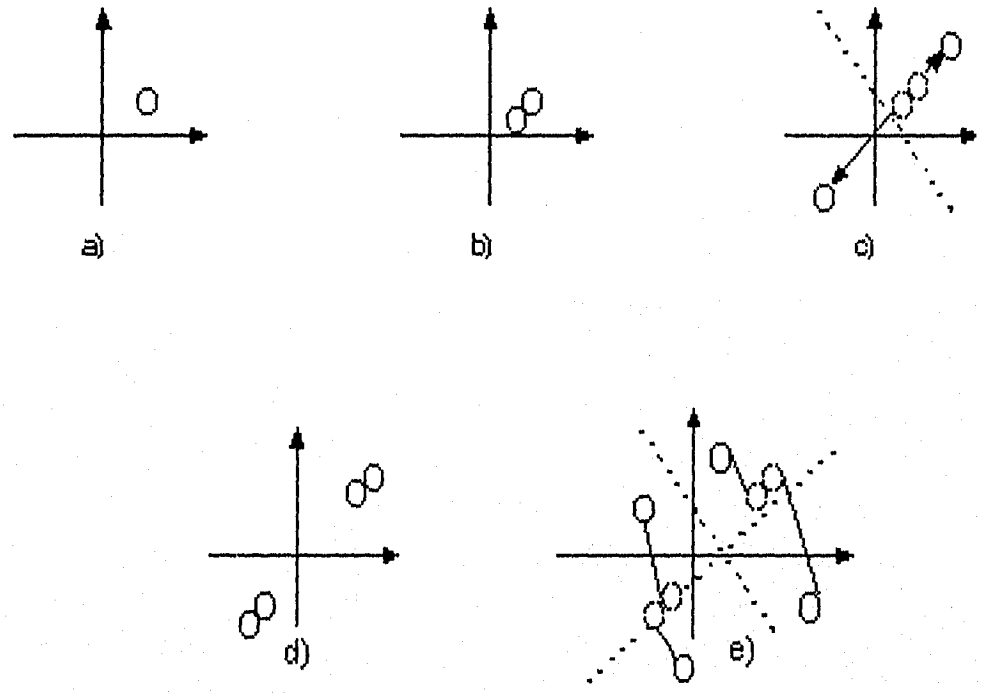



Fig. 3.6.1 Proceso de formación del "codebook" por medio de la técnica de rompimiento.

Este vector patrón inicial, se divide para formar dos patrones (fig. 3.6.1b); esto se puede lograr perturbando ligeramente la energía del primer vector para encontrar el segundo, o bien calcular un vector que se encuentre a cierta distancia del primero. Es conveniente mantener el patrón inicial como miembro del nuevo "codebook" para asegurar que la distorsión no se incrementará. Posteriormente se aplica el algoritmo de Lloyd para obtener el código óptimo a una tasa de un bit por vector (fig. 3.6.1c). El proceso continúa de la misma forma, haciendo que el código final de un estado se rompa para formar el código inicial del siguiente estado (figs. 3.6.1d y 3.6.1e).

## 3.7 ESTRUCTURAS PARA VQ SIN MEMORIA.

Con el fin de minimizar los requerimientos de memoria y de cómputo de los cuantizadores vectoriales sin memoria explorados totalmente, se han propuesto diversas estructuras de VQ's que analizaremos a continuación.

### 3.7.1 VQ EXPLORADO POR ARBOL.

Los cuantizadores vectoriales explorados por árbol, son una derivación de la técnica de rompimiento para encontrar los "codebooks" iniciales. Por ejemplo, considerando que tenemos un buen código de tasa 1 y formamos un nuevo "codebook" a tasa 2 por rompimiento de los dos patrones o "templates" anteriores; entonces, en lugar de aplicar el diseño de un cuantizador vectorial totalmente explorado sobre el "codebook" de 4 patrones; dividimos la secuencia de entrenamiento en dos partes, colocando juntos todos los vectores que fueron codificados en una palabra común en el "codebook" de 1 bit, que corresponden a todos los vectores de la secuencia de entrenamiento que se encuentran en una misma celda de la partición de Voronoi. De tal forma que considerando esta división de la secuencia de entrenamiento, aplicamos el algoritmo de Lloyd y obtenemos un buen "codebook" de 1 bit para cada subsecuencia de vectores de entrenamiento. El "codebook" final se encuentra formado por los cuatro patrones que se encontraron en los dos "codebooks"

de 1 bit. Un codificador explorado por árbol selecciona uno de los "codewords"; utilizando el primer "codebook" de 1 bit que se obtuvo sobre la secuencia de entrenamiento total, de aquí se selecciona un segundo "codebook", para de él obtener el mejor "codeword". El cuantizador vectorial por árbol ayuda a hacer más eficiente su implementación, ya que reduce significativamente el número de búsquedas.

### 3.7.2 VQ MULTIESTADO.

Un cuantizador vectorial multiestado es una variación del cuantizador vectorial explorado por árbol. Se obtiene un "codebook" inicial de la misma forma que en el caso del VQ explorado por árbol. Este "codebook" es utilizado para codificar la secuencia de entrenamiento, de este paso obtenemos una secuencia de entrenamiento de error o de vectores residuales, que se forman a partir de la diferencia entre los vectores de entrada y sus "codewords".

Posteriormente, se aplica el algoritmo de Lloyd para diseñar un cuantizador vectorial para codificar esta secuencia de errores. El símbolo de canal correspondiente se forma combinando los dos "codewords", como se muestra en la fig.3.7.1 .

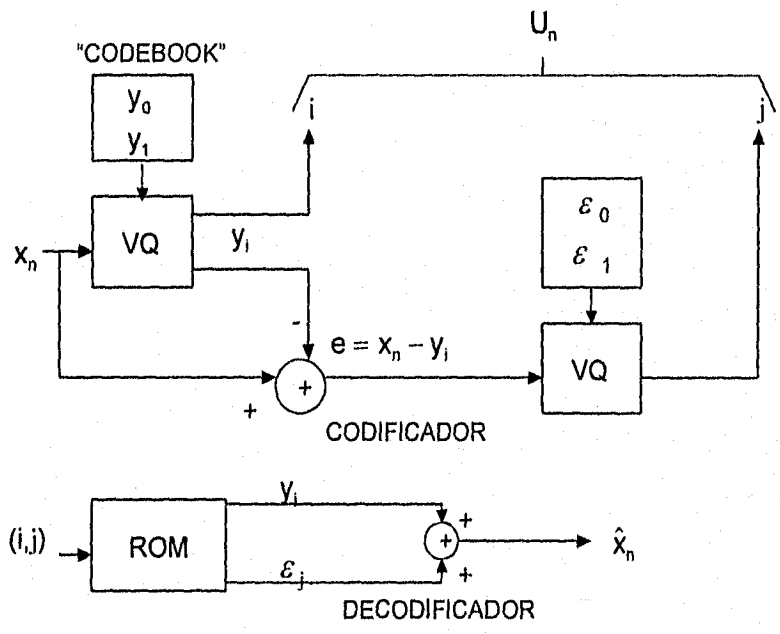



FIG. 3.7.1 VQ Multiestado de dos estados.

### 3.7.3 CODIGOS PRODUCTO

Los códigos producto forman otra estructura para los cuantizadores vectoriales; la técnica básica para implementar un código producto es por medio del uso múltiple de cuantizadores escalares que son fáciles de implementar. Esta técnica es útil cuando existen diferentes aspectos del vector de entrada que se desean codificar en forma separada para producir diferentes efectos; como lo veremos en los siguientes tipos de cuantizadores vectoriales.

#### 3.7.3.1 VQ DE GANANCIA-FORMA

En el VQ de ganancia-forma se utilizan "codebooks" separados, pero interdependientes entre sí, de tal forma que uno de ellos se utiliza para codificar la forma haciendo el producto interno entre un vector de forma con energía unitaria y el vector de entrada ; con lo cual obtendremos el vector de forma; y empleamos otro codificador para la ganancia de la señal; donde la forma está definida como el vector de entrada normalizado, y la ganancia es la energía de la señal. De tal forma que nuestro símbolo de canal final está formado por dos partes, una que representa las características de la forma y otra las de la ganancia de la señal de entrada. Podemos considerar que esta forma de realizar la codificación en dos pasos es una codificación óptima para el "codebook" producto dado; sin embargo, el "codebook" en sí mismo es subóptimo debido a que la estructura de código producto es limitada.

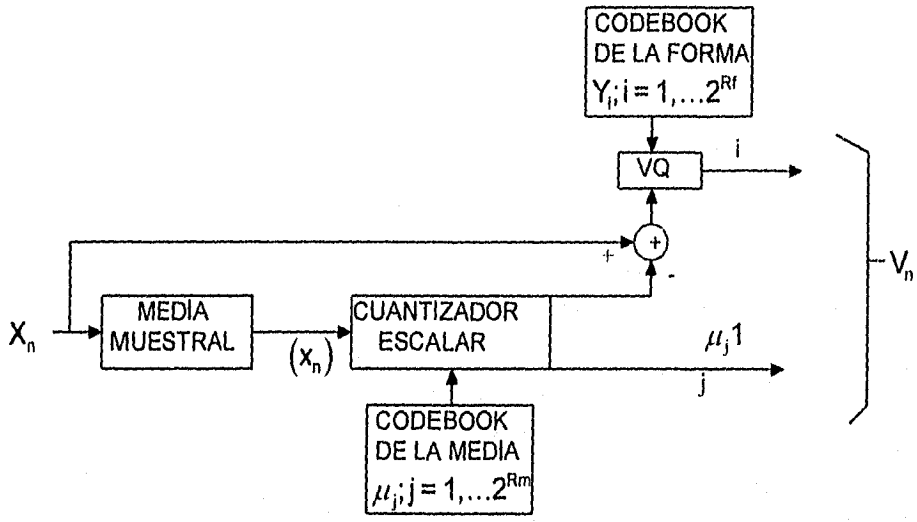
Algunas variaciones del algoritmo básico de Lloyd pueden ser usadas para mejorar iterativamente un código ganancia-forma, optimizando alternativamente la forma para la ganancia y viceversa.

### 3.7.3.2 VQ DE SEPARACION DE LA MEDIA

El cuantizador vectorial por separación de la media es una variación más de los códigos producto, donde ahora en lugar de remover la energía de la señal, lo que hacemos es separar la media muestral; donde la media muestral de un vector k-dimensional está dada por:

$$k^{-1}\sum_{i=0}^{k-1} x_i \qquad 3.7.1$$

En un cuantizador vectorial de este tipo primero se utiliza un cuantizador escalar para codificar la media muestral de un vector; posteriormente, se quita la media muestral de todos los componentes del vector de entrada para formar un nuevo vector con media muestral aproximadamente igual a cero; una vez obtenido este vector, se cuantiza, como se muestra en la fig. 3.7.2.




CODIFICADOR

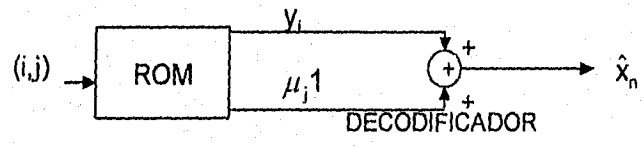



FIG. 3.7.2 VQ de separación de la media.

Para diseñar un VQ por separación de la media, primero usamos el algoritmo para diseñar un cuantizador escalar para la secuencia de la media muestral $x_j$, con j=0,1,..,L-1 . Si q(x) representa la reproducción para x usando el cuantizador; entonces usamos la secuencia de entrenamiento $x_j - q(x_j)1$, donde 1=(1,1,..,1), para diseñar un VQ para la diferencia.

# CAPITULO 4

## ESQUEMA DE CODIFICACION LOT-QV

### 4.1. INTRODUCCION

Después de haber analizado los procesos de codificación por transformada y cuantización vectorial por separado intentaremos ahora integrar estos procesos en el esquema de cuantización LOT-VQ (fig. 4.1.1)

Definiremos los criterios de selección de los parámetros que deben establecerse para aplicar tanto la transformada como el VQ para el caso de imágenes; así como los criterios de evaluación de la compresión y reducción del efecto de interbloqueo.

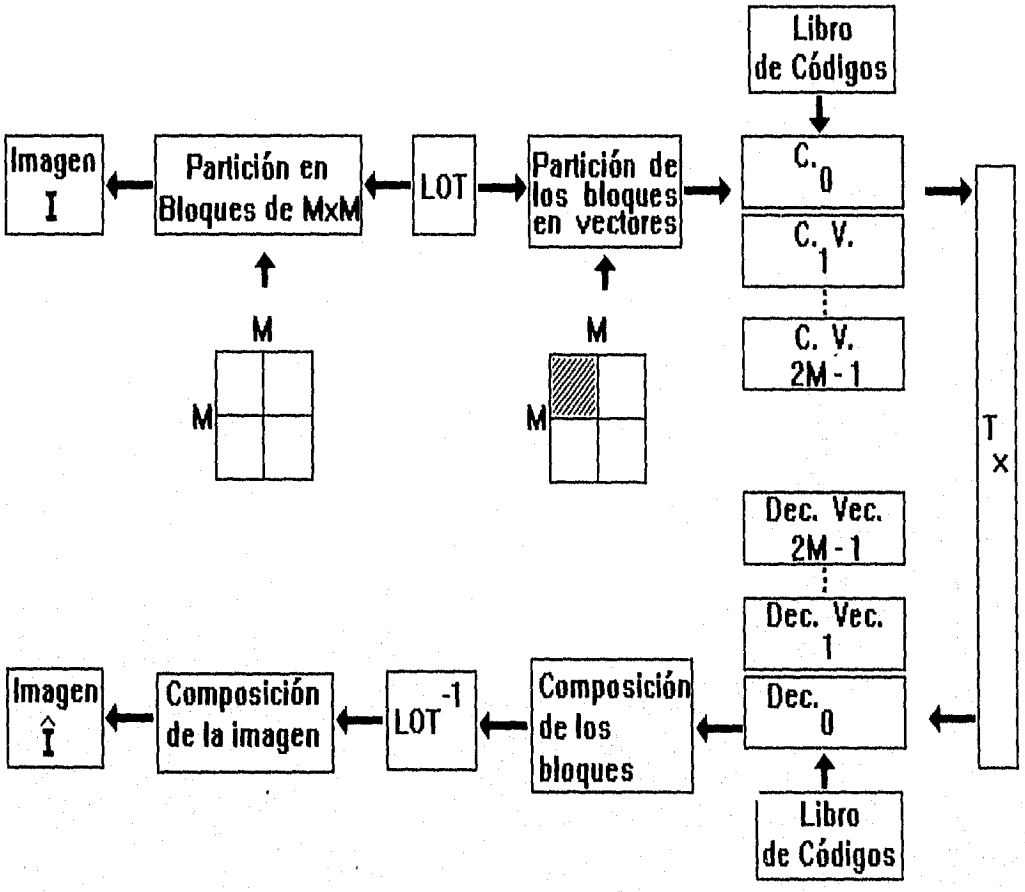



FIG. 4.1.1 Esquema de cuantización LOT-VQ

## 4.2 PARTICION DE LA IMAGEN EN BLOQUES

De acuerdo con el esquema de codificación LOT-VQ de la figura 4.1.1, tenemos que el primer paso consiste en dividir la imagen de entrada en bloques de un tamaño determinado; una imagen debe dividirse en bloques porque la complejidad de los algoritmos de transformación crece exponencialmente con el tamaño de la imagen, así resulta más simple calcular varias transformadas de orden menor que una transformada de orden mayor, además, se aprovechan las características locales de la imagen, sin embargo, a medida que disminuimos el tamaño de cada bloque el MSE a una tasa dada aumenta (fig. 4.2.1) debido a la mayor distorsión del espectro de potencia y a los efectos de interbloqueo. Esto nos lleva a un compromiso entre velocidad y calidad de codificación. En el caso de la LOT debemos considerar además dos circunstancias que hacen más elegibles valores menores de M:

- La LOT tiende a disminuir los efectos de interbloqueo.
- El algoritmo rápido de la LOT para M>=32 es significativamente más complejo ya que requiere del cálculo de una IDCT en lugar de una cascada de mariposas.

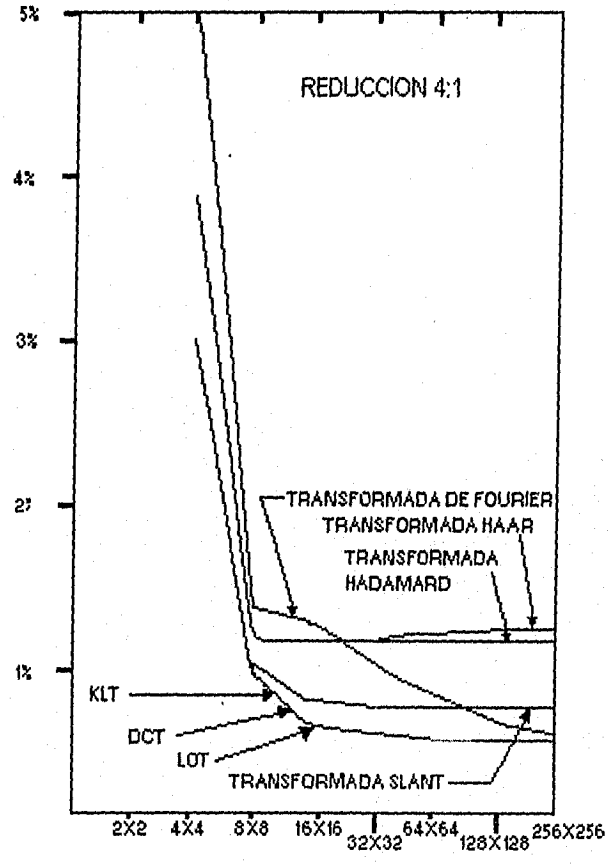



FIG. 4.2.1 Error cuadrático medio de transformadas de imágenes en función del tamaño del bloque. Las estadísticas de la imagen a través de los renglones y columnas son consideradas para un proceso de Markov de primer orden con un coeficiente de correlación de $\rho$=0.95. Sin cuantización ni codificación

Conforme a lo anterior los mejores valores para el tamaño del bloque (M) son 8 y 16 en el caso de codificación por medio de LOT por lo que será con estos valores con los que realizaremos experimentos para evaluar el desempeño de la LOT.

## 4.3 CALCULO DE LA LOT DE DOS DIMENSIONES.

En el capítulo 2 nos enfocamos al desarrollo de la LOT de una señal unidimensional, sin embargo, una imagen es bidimensional, por lo que sin perder de vista el proceso básico de la transformada tendremos que tomar en cuenta ciertas consideraciones para aplicar una transformada ortogonal traslapada a una imagen.

En primer lugar la LOT al igual que otras transformadas presenta la propiedad de separabilidad, es decir, para obtener la transformada de una señal bidimensional es posible aplicar la transformada primero a cada renglón de la imagen como si se tratara de una señal unidimensional y después a cada columna de la matriz obtenida en el paso anterior. Sin embargo, a diferencia de las transformadas tradicionales no es posible procesar independientemente cada bloque pues la LOT de cada uno de ellos depende del bloque anterior. Por lo que en el caso de la LOT necesitamos modificar los cálculos de transformación en el primero y último bloque, de otra forma las funciones básicas de la transformada podrían extenderse fuera de la región que soporta la señal.

Una forma para salvar este problema es asumir que cada renglón y columna se refleja de manera par en sus bordes.

Una vez que se han reflejado los bloques de los extremos y tomando en cuenta la propiedad de separabilidad se procede a transformar cada uno de los renglones como se muestra en la fig. 4.3.1 por lo que obtenemos una matriz de orden NMxNM+M donde N es el número de bloques y M el tamaño de cada bloque como lo indica la figura 4.3.2a. Posteriormente aplicamos simetría par a las columnas de esta matriz transformada y procesamos cada una de las columnas como se muestra en la fig. 4.3.1 para obtener finalmente una matriz de NM+M x NM+M con la energía de la imagen concentrada en los primeros coeficientes de cada bloque.

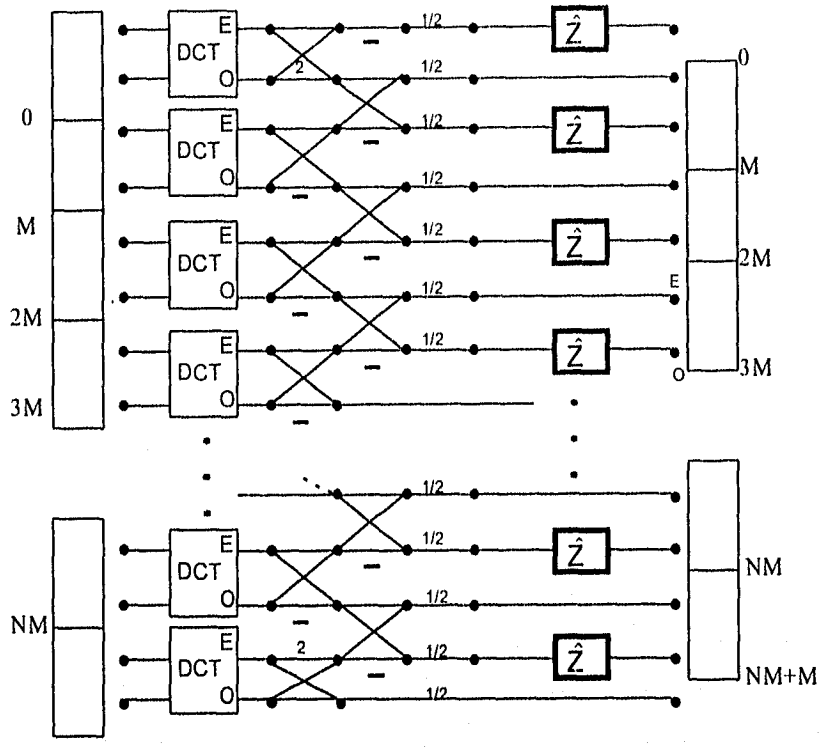



FIG. 4.3.1 Cálculo rápido de la LOT por renglones o columnas.

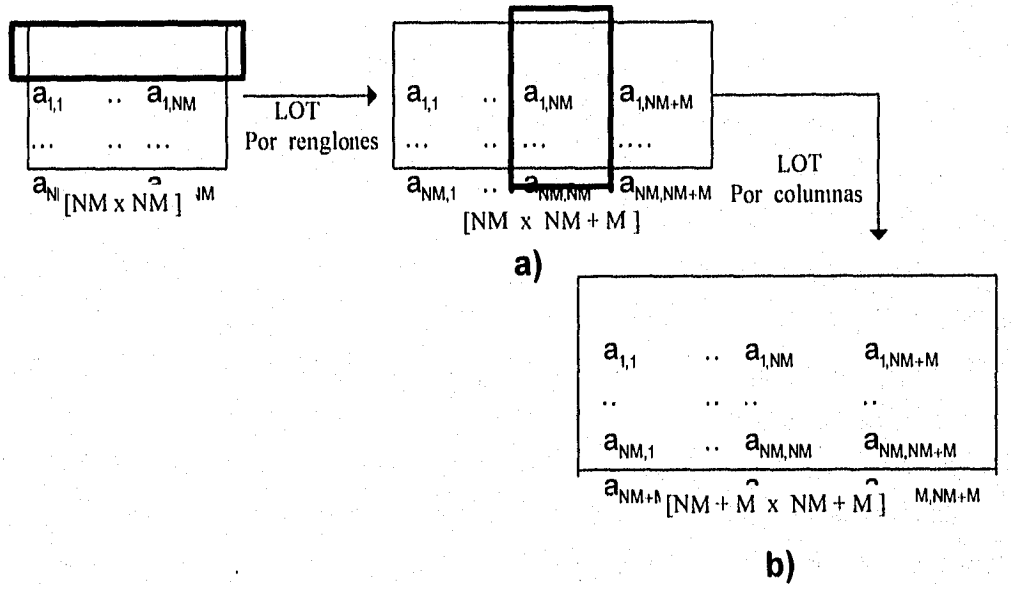



FIG. 4.3.2 Imágenes obtenidas al aplicar la LOT-2D.

En el receptor para obtener la imagen original debemos aplicar la LOT inversa después de decodificar los coeficientes transformados y codificados en el transmisor.

El procedimiento para la obtención de la LOT inversa es muy similar al que sigue la LOT directa tomando en cuenta que ahora necesitaremos el bloque posterior al actual para lograr la reconstrucción de la señal, por lo que debemos almacenar todos los coeficientes transformados para procesar primero las columnas y luego los renglones empezando siempre desde el bloque N+1 hasta el primero como se muestra en la figura 4.3.1. siguiendo el diagrama en sentido inverso; esto se debe a que el proceso es no causal.

## 4.4. PARTICION DE LOS BLOQUES EN VECTORES

Debido a la propiedad de compactación de energía de la LOT la mayor varianza de los coeficientes se localiza en la zona de bajas frecuencias es decir en el extremo superior izquierdo de cada bloque transformado como se muestra en la figura 4.4.1 a) y b).

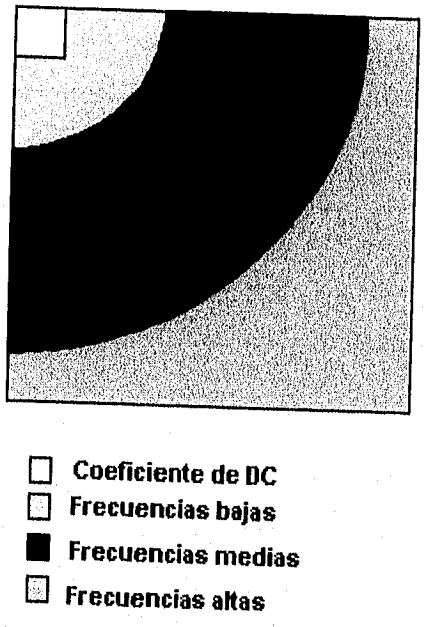



FIG. 4.4.1 a) Distribución de energía de los coeficientes de la LOT y las características de bloque que representan.

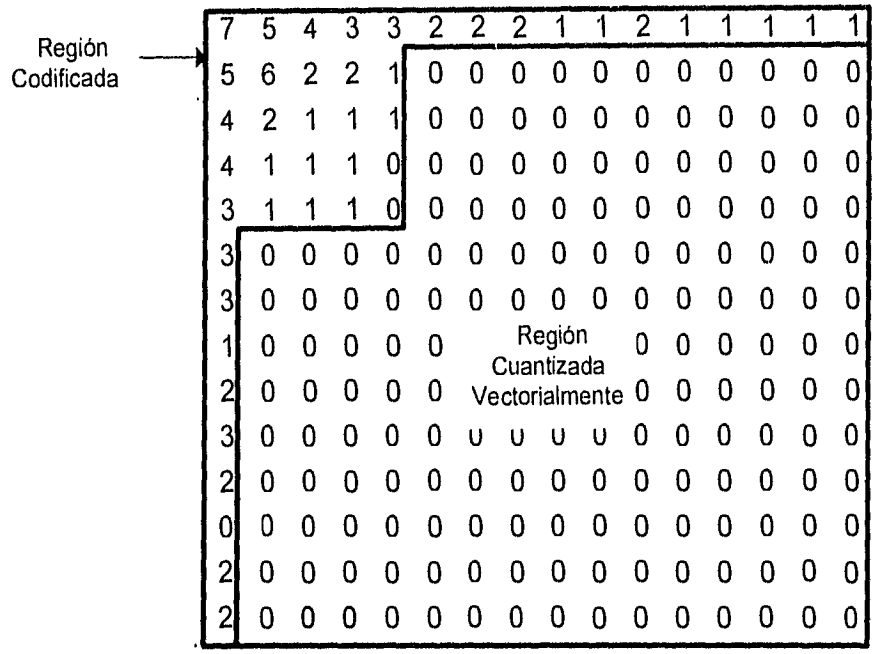



FIG. 4.4.1 b) Matriz de asignación de bits adaptiva. En la zona codificada, los números indican el número de bits asignados a los coeficientes transformados correspondientes. En el caso vectorial, todos los coeficientes son cuantizados vectorialmente.

Otra característica importante que tiene que tomarse en cuenta para realizar la partición de cada bloque es la distribución de energía respecto a las frecuencias horizontales y verticales en el bloque transformado de la LOT la cual es como se muestra en la fig. 4.4.2.

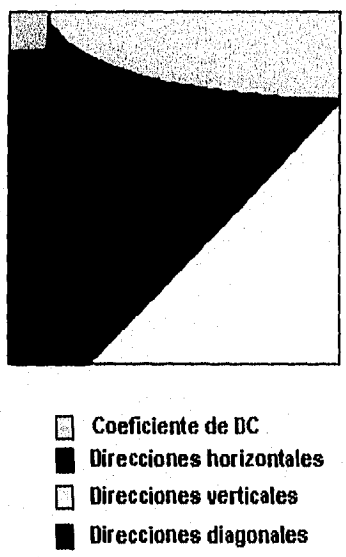



FIG. 4.4.2 Distribución de frecuencia de los coeficientes de la LOT y las características de bloque que representan.

De lo anterior podemos definir dos particiones de los bloques que nos ayuden a agrupar zonas con un contenido de información tal que podamos tratarlas como vectores independientes.

La primera partición es como se muestra en la fig. 4.4.3 el vector v0 corresponde a la componente de DC la cual debe ser cuantizada escalarmente, los vectores v1,v2,..., v2M-1 representan las componentes de AC que deben ser cuantizadas por medio de cuantizadores vectoriales independientes a tasas variables.

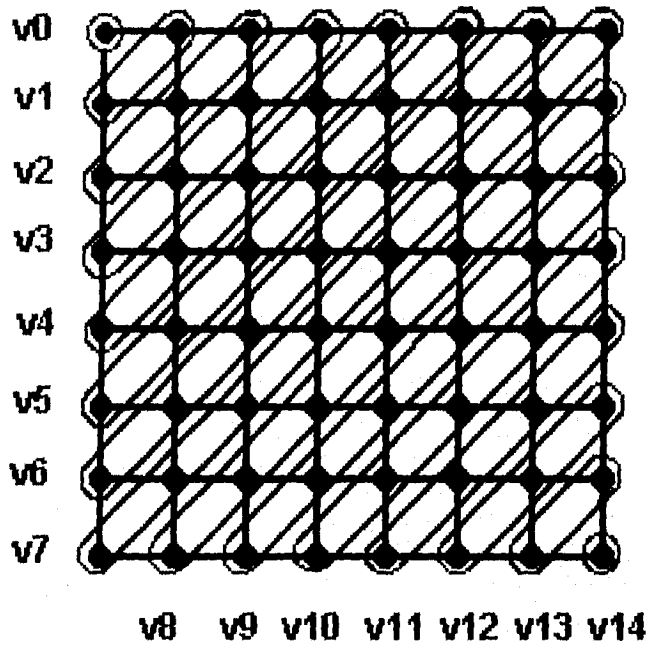



FIG. 4.4.3 Cada bloque de 8 x 8 de la LOT se particiona en un término de dc v0 y 14 vectores v1,v2,...,v14.

Este tipo de partición aprovecha la distribución Laplaciana de la energía en el bloque transformado, sin embargo tiene la desventaja de considerar que todos los bloques tienen la misma distribución de energía. Esta desventaja puede salvarse si se hace una clasificación de los bloques transformados de acuerdo con la distribución de energía de AC como se muestra en la fig. 4.4.4.

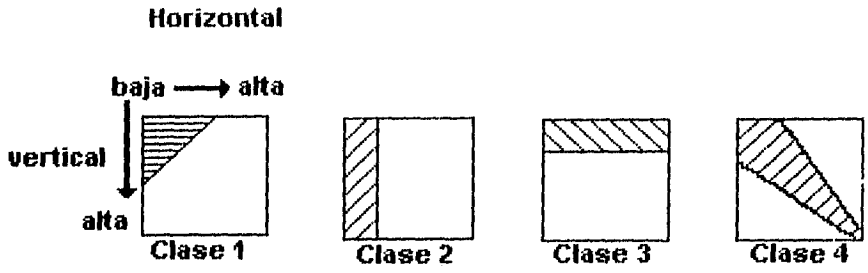



FIG. 4.4.4 Patrones para clasificación.

En los bloques de la clase uno la energía está concentrada en las bajas frecuencias, en los de clase dos en las frecuencias horizontales, en los de clase tres en las frecuencias verticales y en los de clase cuatro en las frecuencias diagonales, de acuerdo a las características de compresión de la LOT podemos asignar un número de bits creciente a cada categoría de bloques en el siguiente orden:

| | Número de bits asignados |
|---|---|
| Clase 1 | 2 |
| Clase 2 | 10 |
| Clase 3 | 24 |
| Clase 4 | 50 |

Un tipo de asignación por vector y clase para una tasa de .4995 bits/pixel se muestra en la siguiente tabla:

Tabla 4.4.1

| | b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | ..... b14 |
|---|---|---|---|---|---|---|---|---|---|
| clase 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ..... 0 |
| clase 2 | 5 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | ..... 0 |
| clase 3 | 7 | 7 | 5 | 3 | 2 | 0 | 0 | 0 | ..... 0 |
| clase 4 | 9 | 10 | 10 | 9 | 7 | 4 | 1 | 0 | ..... 0 |

Otra partición es como se muestra en la fig. 4.4.6

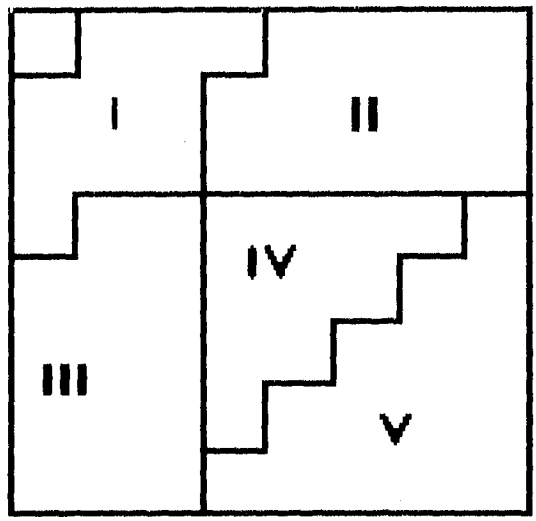



FIG. 4.4.6. Descomposición para los coeficientes de ac.

En esta partición los subbloques I,II,III,IV y V se tratan como vectores separados y se hace una asignación de bits de acuerdo a la varianza de cada vector. Frecuentemente es suficiente codificar el vector I. Los vectores I y II representan las estructuras verticales y los vectores I y III corresponden a las estructuras horizontales.

Una vez obtenidos los vectores debemos establecer el número de bits con que se representará cada clase de vector de acuerdo a la varianza en una secuencia de entrenamiento. El número de bits determinará el tamaño del "codebook" para cada clase de vector de acuerdo con la siguiente fórmula:

$$\text{Tamaño del "codebook"} = 2^{\text{número de bits asignados}} \qquad 4.4.1$$

### 4.4.1 ASIGNACION DE BITS.

En la discusión anterior, cada bloque transformado fue dividido en vectores con el objeto de obtener una partición que se ajuste a la distribución de energía de los bloques de la imagen. Ahora debemos cuantizar cada vector de acuerdo a su energía. Es decir, si una clase de vectores contiene más información que otra debemos representarla con una

mayor cantidad de bits. Para esto analizaremos inicialmente la asignación de bits para el caso escalar.

La teoría de la información dice que si una variable aleatoria con distribución gaussiana tiene varianza $\sigma^2$ y un error cuadrático medio D es aceptable, entonces necesitamos representar la variable aleatoria con solo .5 $\log_2(\sigma^2/D)$ bits. Por lo tanto el contenido de información de un pixel es proporcional a $\log_2(\sigma^2/D)$. Esto nos conduce a que el número de bits asignados a un vector es indicativo de su contribución en información a la información total contenida en la imagen, así el número de bits asignados debe ser proporcional al logaritmo de la varianza.

Mostraremos a continuación que la conclusión anterior nació independientemente del problema de asignación de bits por medio de la minimización del MSE total (o su promedio).

Sea la función f(m,n), m,n = 0,1,2, ..., M-1 el conjunto de pixeles de un bloque transformado de MxM de la imagen y fq(m,n) los pixeles reconstruidos. El error cuadrático medio entre el bloque transformado original y el reconstruido está dado por:

$$D = \frac{1}{M^2} E \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} \left[ f(m,n) - f_q(m,n) \right]^2 \qquad 4.4.2$$

Donde M es la dimensión de cada bloque de la imagen.

Hemos asumido que los pixeles de la LOT son reales.

Sea $\sigma^2(u)$ la varianza del u-ésimo pixel dada por:

$$\sigma^2(m,n) = \frac{1}{N^2} \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} \left( \bar{f}(m,n) - f_v(m,n) \right)^2 \qquad 4.4.3$$

Donde

$\bar{f}(m,n) = \frac{1}{N^2} \sum_{v=0}^{v=N^2-1} f_v(m,n)$ Representa la media de los pixeles.

$f_v(u)$ Representa el u-ésimo pixel del bloque v

y $N^2$ Representa el número total de bloques de la imagen transformada.

entonces podemos observar claramente que $f(m,n)/\sigma^2(m,n)$ es una variable aleatoria con varianza unitaria. Supongamos que el cuantizador para el mn-ésimo pixel fue diseñado en base a la varianza unitaria de entrada. Sea $d_{m,n}(b_{m,n})$ el MSE de cuantización de este cuantizador cuando $b_{m,n}$ son los bits necesarios para representar sus niveles de salida. El error cuadrático medio de cuantización para f(m,n) consigo misma será igual a $\sigma^{2(m,n)}$ $d_{m,n}(b_{m,n})$. Dado que por definición el lado derecho de la ecuación 4.4.2 representa el error cuadrático medio de cuantización para el pixel m,n tenemos que:

$$E\left[f(m,n) - f_q(m,n)\right]^2 = \sigma^2(m,n)d_{m,n}(b_{m,n}) \qquad 4.4.4$$

El promedio del error de cuantización total puede escribirse como:

$$D = \frac{1}{M^2}\sum_{m=0}^{M-1}\sum_{n=0}^{M-1}\sigma^2(m,n)d_{m,n}(b_{m,n}) \qquad 4.4.5$$

La asignación de bits(el valor dado a $b_{m,n}$ ) deberá obtenerse de tal forma que minimice D sujeto a la siguiente restricción:

$$\sum_{m=0}^{M-1}\sum_{n=0}^{M-1}b_{m,n} = M^2 b_{aver} \qquad 4.4.6$$

donde $b_{aver}$ es el número promedio de bits por pixel que deseamos usar para el bloque. En compresión no adaptiva $b_{aver}$ es el mismo para todos los bloques, en compresión adaptiva podríamos formar tres o cuatro categorías de bloques de acuerdo a su nivel de actividad y asignarle una $b_{aver}$ distinta a cada categoría.

De acuerdo a la definición que se haga de $d_{m,n}(b_{m,n})$ el problema de minimización en 4.4.5 sujeto a 4.4.6 tiene las siguientes soluciones:

$$b_{m,n} = b_{aver} + \frac{1}{2}\log_2\sigma^2(m,n) - \frac{1}{2M^2}\sum_{m=0}^{M-1}\sum_{n=0}^{M-1}\log_2\sigma^2(m,n) \qquad 4.4.7$$

$$b_{m,n} = b_{aver} + \frac{2}{\ln 10}\left[\ln\sigma^2(m,n) - \frac{1}{M^2}\sum_{m=0}^{M-1}\sum_{n=0}^{M-1}\ln\sigma^2(m,n)\right] \qquad 4.4.8$$

Tanto en 4.4.7 como en 4.4.8 los valores para $b_{m,n}$ podrían resultar no enteros en cuyo caso deberá redondearse al entero más cercano. Después de tal redondeo es posible que la condición en 4.4.6 no se cumpla por lo que deberá de modificarse la asignación de bits a los pixeles hasta lograr la tasa requerida.

Para el caso vectorial es necesario calcular la asignación de bits escalar. Una vez hecho esto a cada vector se le asignan tantos bits como la suma de los bits asignados a cada uno de sus componentes.

## 4.5 ESQUEMA DE CODIFICACION Y TRANSMISION

Después de haber realizado la partición de cada uno de los bloques transformados en vectores, de acuerdo con alguna de las particiones descritas en la sección anterior, procederemos a cuantizarlos.

El proceso de cuantización de cada bloque se inicia tomando el vector 0 que representa el valor de DC; este vector es de dimensión uno por lo que se cuantiza escalarmente; posteriormente tomamos el vector 1 y buscamos el "codeword" del "codebook" 1 que más se aproxime al vector del bloque transformado, de acuerdo a una medida de distancia dada. Una vez que se encuentra el "codeword" óptimo, únicamente transmitimos el índice que tiene este "codeword" dentro del "codebook". En el receptor se recupera el vector del "codebook" 1 a partir del índice recibido, y se coloca en la misma posición del vector original. Este proceso se repite para todos los vectores del bloque, usando en cada caso un "codebook" diferente (por lo que tendremos tantos "codebooks" como vectores tenga un bloque como se muestra en la fig. 4.1.1); así mismo se repite también para todos los bloques de la imagen transformada.

En el caso de haber hecho una clasificación previa debemos tomar en cuenta la categoría a la que pertenece cada bloque para seleccionar el conjunto adecuado de "codebooks" tanto en el transmisor como en el receptor, por lo que en este caso, además de transmitir la secuencia de índices, también se transmite la clase a la que pertenece cada vector.

La eficiencia del proceso anterior depende de ciertos factores preponderantes tales como: la medida de distancia, el algoritmo de búsqueda, la estructura del

cuantizador vectorial y la obtención del "codebook"; los cuales serán detallados en las secciones siguientes.

### 4.5.1 MEDIDA DE DISTANCIA.

De acuerdo con el proceso de codificación mencionado en la sección anterior, tenemos que para cuantizar cada uno de los vectores de los bloques de la imagen transformada, es necesario buscar el elemento ("codeword") del "codebook" correspondiente más cercano al vector transformado (vector de entrada del cuantizador) , el cuál es el que mejor representa a dicho vector. El "codeword" seleccionado, debe presentar la mínima distorsión o el máximo parecido con el vector de entrada para lograr una buena reconstrucción en el receptor, por lo que es muy importante definir adecuadamente la medida de distancia que usaremos para evaluar el parecido entre ambos vectores.

Algunas de las medidas de distancia que podemos elegir son:

MSE

$$d_1(x, y_i) = \frac{1}{K}\sum_{m=1}^{K}\left(x_m - y_{im}\right)^2 \qquad 4.5.1$$

MAE

$$d_2(x, y_i) = \frac{1}{K}\sum_{m=1}^{K}\left|x_m - y_{im}\right| \qquad 4.5.2$$

In

$$d_3(x, y_i) = \left[\sum_{m=1}^{K}\left|x_m - y_{im}\right|^n\right]^{\frac{1}{n}} = \left\|x - y_i\right\|_n \qquad 4.5.3$$

n-ésima ley de distorsión

$$d_4(x, y_i) = \sum_{m=1}^{K}\left|x_m - y_{im}\right|^n = \left(\left\|x - y_i\right\|_n\right)^n \qquad 4.5.4$$

Distorsión cuadrática pesada

$$d_5(x, y_i) = \sum_{m=1}^{K} w_m (x_m - y_{im})^2 \qquad 4.5.5$$

Distorsión cuadrática general

$$d_6(x, y_i) = (x - y_i)[W](x - y_i)^T \qquad 4.5.6$$

En estas ecuaciones w1,w2,...,wk son los pesos, [W] es una matriz simétrica positiva definida de KxK cuya correcta selección permite incorporar propiedades perceptuales en el VQ, y yi={yi1, yi2,..,yik} es el i-ésimo elemento del "codebook".

Las principales características que debemos tomar en cuenta al seleccionar una medida de distancia determinada son:

- La distancia debe ser no negativa.

- La distancia entre un vector y sí mismo debe ser cero.

- Debe estar correlacionada con la calidad cualitativa de la imagen cuando se aplica globalmente, de tal forma que un aumento o disminución de la distancia se refleje en una degradación o mejora de la calidad perceptual de la imagen.

- Debemos considerar también la complejidad de cálculo que implica, para poder implementarla y saber si es conveniente utilizarla dentro de determinado cuantizador.

### 4.5.2 ALGORITMO DE BUSQUEDA.

El algoritmo de búsqueda se refiere a la forma en la que el cuantizador vectorial evalúa los "codewords" de un "codebook" determinado en busca del óptimo para cada vector de todos los bloques de la imagen transformada. Es decir, se refiere a la forma en la que el VQ realiza la búsqueda del vector de mínima distancia al vector transformado.

Dentro de los algoritmos de búsqueda que han dado mejores resultados al encontrar los "codewords" que proporcionan la mínima distorsión posible de acuerdo con el "codebook" utilizado; se encuentra el algoritmo de búsqueda total o exhaustiva, que como su nombre lo indica consiste en comparar cada uno de los "codewords" con el vector de entrada, guardando el índice del que presente la distancia mínima este algoritmo se puede describir por medio de los siguientes pasos:

Paso 1.- Inicializamos el valor de la distancia mínima con cero.

$$d_{min} = 0$$

Paso 2 .- Dado un vector de entrada x y un "codebook" C={yi, i=1,2,...,W} (donde W es el número de "codewords" que tiene el "codebook"), calculamos la distancia d entre el vector x y el primer "codeword" y1.

$$d = d(x,yi)$$

Paso 3 .- Si la distancia obtenida en el paso anterior es menor a la distancia mínima obtenida hasta este momento entonces asignamos el valor de d a $d_{min}$

Paso 4 .- Regresar al Paso 2 tomando ahora el siguiente "codeword". Se repite el algoritmo hasta que se haya comparado el vector x con todos los "codewords" del "codebook" C.

Como mencionamos anteriormente, este algoritmo es el que nos ofrece los mejores resultados en cuanto a minimizar la distancia entre la imagen de entrada y la imagen reconstruida, además de ser muy fácil de implementar, sin embargo tiene la desventaja de ser muy lento, ya que compara cada uno de los vectores de todos los bloques de la imagen con cada uno de los "codewords" de los "codebooks" correspondientes, por lo que el número de operaciones y comparaciones que tiene que realizar se eleva en gran medida dependiendo de la dimensión de los vectores de entrada y del tamaño de los "codebooks". Es por esto que se han realizado ciertas variaciones al algoritmo de búsqueda total con el fin de reducir el tiempo de cómputo.

Un algoritmo que tiende a reducir el número de operaciones y por lo tanto el tiempo de búsqueda es el algoritmo de búsqueda por árbol. Este algoritmo es utilizado en los cuantizadores vectoriales explorados por árbol, que fueron mencionados en el capítulo anterior en la sección 3.7.1; y consiste básicamente en buscar ligas a través de varios "codebooks" hasta encontrar el "codeword" deseado; es decir, dado un vector de entrada x y un "codebook" inicial C, buscamos en él el "codeword" que presente la mínima distancia al vector de entrada, este "codeword", nos sirve para elegir un segundo "codebook" (o nivel del árbol ), en donde se lleva a cabo el mismo procedimiento, y así sucesivamente hasta recorrer todo el árbol, el "codeword" que se seleccione al final, será nuestro vector de reconstrucción definitivo. La ventaja de este algoritmo radica en que los "codebooks" de cada nivel son de una dimensión menor que en el "codebook" completo que utilizamos en el algoritmo de búsqueda total, por lo tanto se reduce el número de comparaciones que se tienen que hacer, es decir si queremos cuantizar a una tasa de 10 bits por vector, en lugar de hacer $2^{10} = 1028$ cálculos y comparaciones de distancia por vector de la imagen transformada sobre un "codebook" total , podemos dividir nuestra búsqueda en "codebooks" con tasas de 4,4 y 2 bits por vector con lo que realizaríamos $2^4+2^4+2^2=36$ comparaciones; como se observa la reducción de cálculo es bastante significativa, sin embargo, la desventaja que presenta este tipo de búsqueda es que no se garantiza que los "codewords" seleccionados sean los que proporcionen la mínima distorsión posible, por lo que deberemos sacrificar calidad de la imagen por rapidez.

La elección del algoritmo de búsqueda adecuado dependerá de las características del equipo con el que se cuente para implementarlo y de las prioridades que se tengan en cuanto a rapidez o calidad.

### 4.5.3 ESTRUCTURA DEL VQ.

Como hemos visto, el algoritmo de búsqueda influye directamente en la estructura del VQ, la cual puede modificar el procedimiento mencionado al inicio de este capítulo, como en el caso del VQ multiestado, el de ganacia y forma o el de separación de la media , vistos en el capítulo anterior, en donde hay que llevar a cabo un proceso más elaborado para realizar la cuantización, ya que se tienen que transmitir datos adicionales a los

índices de los "codewords", como la media cuantizada de cada uno de los vectores de entrada, el error o características específicas de la imagen.

### 4.5.4 OBTENCION DEL "CODEBOOK".

El "codebook" es una parte fundamental para el proceso de codificación, de su correcta obtención depende en gran medida que logremos una reconstrucción aceptable, por lo tanto es muy importante establecer el algoritmo de obtención del "codebook" óptimo, basado en el algoritmo de Lloyd-Max, descrito en el capítulo anterior.

El algoritmo de obtención del "codebook" puede resumirse en los siguientes pasos:

Paso 1.- Obtener un "codebook" inicial, que puede ser de la misma dimensión que el "codebook" final (como en el caso de los "codebooks" aleatorios) o bien de una dimensión menor, a partir del cuál se irá formando iterativamente el "codebook" final (como en los "codebooks" por rompimiento.); en este caso el vector inicial puede ser el centroide de los vectores de la secuencia de entrenamiento.

Paso 2 .- Codificar la secuencia de entrenamiento utilizando el "codebook" obtenido; asignando a cada vector de entrada el "codeword" más cercano de acuerdo con alguna medida de distancia dada. Si la distorsión promedio es baja y hemos llegado a la tasa que queremos transmitir, salir.

Paso 3.- Calcular los centroides de todos los vectores de entrenamiento que fueron asignados a cada uno de los "codewords" y reemplazar el "codebook" por dichos centroides.

Paso 4.- En el caso de haber iniciado con un "codebook" de una dimensión menor que el "codebook" final, aplicar la técnica de rompimiento para incrementar la dimensión del "codebook" y regresar al paso 2.

Como podemos observar, la obtención del "codebook", se encuentra íntimamente ligada a la medida de distancia utilizada, al algoritmo de búsqueda y a la elección del "codebook" inicial, por lo que deberemos tomar en cuenta todas las consideraciones mencionadas anteriormente en cuanto a la distancia y al algoritmo de búsqueda, para obtener un buen "codebook" final, por lo que respecta a la elección del "codebook" inicial, tenemos que tiene cierta influencia en cuanto a rapidez por ejemplo, el "codebook" inicial aleatorio, puede permitir una convergencia más rápida hacia el "codebook" óptimo, dependiendo de las muestras de la secuencia de entrenamiento que elijamos para formar el "codebook" inicial.

# CAPITULO 5

## RESULTADOS Y CONCLUSIONES

### 5.1 INTRODUCCION

Después de establecer el esquema de codificación LOT-VQ, es necesario contar con ciertas formas de medir el desempeño del mismo, por lo que en este capítulo nos enfocaremos a establecer los criterios de evaluación que nos ayudarán a determinar si este esquema de codificación realmente proporciona mejoras en cuanto a la calidad de las imágenes, tanto cuantitativamente como subjetivamente, sobre todo a bajas tasa de bits y prestando especial atención a la reducción de los efectos de interbloqueo que marcarán la presencia de la transformada ortogonal traslapada.

El capítulo se desarrollará presentando inicialmente un marco teórico de los criterios de evaluación tanto cuantitativos como subjetivos, para después presentar y analizar las pruebas y resultados obtenidos de la experimentación hecha con el esquema de codificación LOT-VQ, basándonos en el marco teórico inicial.

### 5.2. CRITERIOS DE EVALUACION

#### 5.2.1 EVALUACION DE LA CODIFICACION

En general los diversos criterios cuantitativos no son verdaderos evaluadores del desempeño de la codificación de video. Para obtener un juicio justo acerca del sistema de codificación la calificación subjetiva (por medio de la inspección visual de las imágenes procesadas) es esencial. En la literatura sin embargo medidas cuantitativas como el error cuadrático medio y la relación señal a ruido frecuentemente son usadas para comparar diferentes codificadores. En este contexto, podemos definir las siguientes medidas de desempeño:

### 5.2.1.1 ERROR CUADRATICO MEDIO (MSE)

$$E\left\{\left[x(m,n)-\hat{x}(m,n)\right]^2\right\} \approx \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left[x(m,n)-\hat{x}(m,n)\right]^2 \qquad 5.2.1$$

donde x(m,n) y $\hat{x}(m,n)$ son las intensidades de las imágenes originales y reconstruidas en el renglón m y la columna n, respectivamente. El tamaño de la imagen es MxN. La diferencia entre las imágenes original y reconstruida es el error de reconstrucción. La aproximación descrita por el lado derecho de 5.2.1 mejora cuando el MSE es calculado sobre un gran número de imágenes.

### 5.2.1.2 ERROR CUADRATICO MEDIO NORMALIZADO

$$NMSE_a = \frac{\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left[x(m,n)-\hat{x}(m,n)\right]^2}{\frac{1}{MN}\sum_{m=1}^{M-1}\sum_{n=1}^{N-1}\left[x(m,n)\right]^2} \qquad 5.2.2$$

$$NMSE_b = \frac{\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left[x(m,n)-\hat{x}(m,n)\right]^2}{x_{pp}^2} \qquad 5.2.3$$

En 5.2.2 el error cuadrático medio es normalizado por la energía de la imagen y en 5.2.3 por la intensidad pico a pico $x_{pp}$. Para un PCM de 8 bits, $x_{pp}$ es 255.

### 5.2.1.3 ERROR MEDIO ABSOLUTO (MAE)

$$MAE = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|x(m,n)-\hat{x}(m,n)\right| \qquad 5.2.4$$

### 5.2.1.4 ERROR MEDIO ABSOLUTO NORMALIZADO (NMAE)

$$NMAE = \frac{MAE}{\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|x(m,n)|} \qquad 5.2.5$$

La correspondiente razón señal a ruido (SNR) en decibeles se define por medio de $-10\log_{10}(NMSE)_a$ y $-10\log_{10}(NMSE)_b$. Para una secuencia o secuencias de imágenes, los errores son calculados sobre el dominio temporal también por medio de la expresión $-10\log_{10}(NMSE)$,$-10\log_{10}(NMSE)_a$ y $-10\log_{10}(NMSE)_b$. Por ejemplo el error cuadrático medio calculado en 5.2.1 cambia sobre el dominio temporal a:

$$\frac{1}{MNP}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{p=0}^{P-1}\left[x(m,n,p)-\hat{x}(m,n,p)\right]^2 \qquad 5.2.6$$

donde el tercer índice p se refiere al dominio temporal.

### 5.2.1.5 EVALUACION SUBJETIVA

Para la evaluación subjetiva, un grupo de observadores (suponiendo que son expertos en sistemas de codificación de imágenes) observan tanto la imagen original como la procesada bajo condiciones aleatorias de luz y distancia, cada uno da una calificación y se obtiene una calificación media de opinión basándose sobre una escala de evaluación como la tabla 5.2.1

Tabla 5.2.1
Tabla de evaluación subjetiva

| Opinión | Puntos |
|---|---|
| No perceptible | 7 |
| Apenas perceptible | 6 |
| Definitivamente perceptible pero sólo deteriora ligeramente la imagen | 5 |
| Deteriora la imagen pero no desagradablemente | 4 |
| Algo desagradable | 3 |
| Definitivamente desagradable | 2 |
| Extremadamente desagradable | 1 |

Aunque los criterios cuantitativos y subjetivos son los puntos de prueba para evaluar un algoritmo, igualmente importantes son otros factores como la complejidad en la implementación (realización en hardware) y varias características opcionales pertinentes a una aplicación específica. Por ejemplo en la mayoría de las implementaciones se ha usado la DCT en lugar de la transformada óptima KLT pues los factores de complejidad en la implementación han pesado más que las pequeñas diferencias en la evaluación objetiva o subjetiva.

### 5.2.2 EVALUACION DE LA REDUCCION DE EFECTOS DE INTERBLOQUEO.

El desempeño de la LOT en cuanto a la reducción de efectos de interbloqueo puede ser evaluada tanto subjetiva como objetivamente.

Subjetivamente deberá asignarse una calificación a los efectos de interbloqueo de acuerdo a su perceptibilidad de la siguiente forma:

| | |
|---|---|
| 5 | Imperceptible |
| 4 | Perceptible pero no molesto. |
| 3 | Ligeramente molesto |
| 2 | Molesto |
| 1 | Muy molesto |

Tabla 5.2.2. Evaluación subjetiva de los efectos de interbloqueo.

Para evaluar el desempeño de LOT con respecto a otros métodos de codificación y reducción de efectos de interbloqueo deberá elegirse un límite de evaluación y codificar a tasas cada vez menores hasta rebasar el límite elegido, un método será mejor a otro cuando se logre una menor tasa de transmisión al rebasar el límite. Por ejemplo podemos disminuir la tasa de transmisión hasta que los efectos de interbloqueo entren dentro del nivel 5 (imperceptibe) en ese momento registraremos la tasa que logramos y la compararemos con la obtenida con otros métodos de compresión y reducción de efectos de interbloqueo en el mismo proceso.

La medida objetiva usada puede ser el error cuadrático medio normalizado (NMSE), aunque el NMSE no es una medida exacta de la calidad de la imagen, generalmente hemos encontrado que la imagen decodificada es muy buena cuando el NMSE es menor a 0.25%, y es usualmente inaceptable cuando el NMSE se encuentra alrededor de .36%. Sin embargo el NMSE no es una medida muy sensible al aumento o disminución de los efectos de interbloqueo, ya que este efecto se presenta no por un mayor error entre la imagen original y la reconstruida, sino por una desincronización de este error entre los diferentes bloques de la imagen por lo que para evaluar la reducción de efectos de interbloqueo es preferible utilizar la evaluación subjetiva.

## 5.3. PRUEBAS Y RESULTADOS DEL ESQUEMA DE CODIFICACION LOT-VQ

La técnica que emplearemos para la experimentación consiste en utilizar las transformadas LOT y DCT dentro del mismo esquema de codificación, con el fin de establecer un punto de comparación que nos permita evaluar el desempeño del esquema de cuantización LOT-VQ, tanto en forma subjetiva como cuantitativa. El uso de la DCT

como punto de comparación se justifica, ya que es una transformada que ha dado muy buenos resultados dentro de la compresión de imágenes

A continuación presentamos las condiciones y resultados de las pruebas aplicadas a dos imágenes, utilizando un cuantizador vectorial totalmente explorado.

Para todas las pruebas realizadas, utilizamos imágenes de 256 x 256 pixeles, que fueron particionadas en bloques de 8 x 8, que a su vez fueron particionados como se mostró en la fig. 4.4.3 estos vectores tuvieron la asignación de bits que se muestra en la siguiente tabla, de acuerdo con la tasa de compresión requerida:

| TASA | v0 | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 | v10 | v11 | v12 | v13 | v14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0.1** | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0.2** | 8 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0.3** | 8 | 6 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0.4** | 8 | 7 | 5 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0.5** | 8 | 7 | 7 | 5 | 3 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0.6** | 8 | 9 | 7 | 6 | 4 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0.7** | 8 | 9 | 8 | 8 | 6 | 4 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0.8** | 8 | 9 | 10 | 9 | 8 | 5 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **0.9** | 8 | 9 | 10 | 10 | 9 | 7 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1.0** | 8 | 10 | 10 | 10 | 9 | 7 | 5 | 4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1.1** | 8 | 10 | 10 | 10 | 10 | 9 | 7 | 5 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1.2** | 8 | 10 | 10 | 10 | 10 | 9 | 9 | 7 | 3 | 1 | 0 | 0 | 0 | 0 | 0 |
| **1.3** | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 6 | 1 | 0 | 0 | 0 | 0 | 0 |
| **1.4** | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 7 | 4 | 2 | 0 | 0 | 0 | 0 |
| **1.5** | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 6 | 2 | 1 | 0 | 0 | 0 |
| **1.6** | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 7 | 4 | 3 | 1 | 0 | 0 |
| **1.7** | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 7 | 3 | 1 | 0 | 0 |
| **1.8** | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 6 | 2 | 1 | 0 |
| **1.9** | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 4 | 1 | 0 |
| **2.0** | 8 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 9 | 7 | 3 | 1 |

Tabla 5.3.1. Asignación de bits para tasas de 0.1 a 2 bits por pixel.

### 5.3.1. PRUEBAS CUANTITATIVAS.

Para la evaluación cuantitativa utilizaremos el cálculo del error cuadrático medio (MSE), error cuadrático medio normalizado (NMSE) y la relación señal a ruido (SNR) para

establecer la calidad de la imagen.

**ERROR CUADRATICO MEDIO (MSE)**

A continuación mostramos los resultados obtenidos al calcular el error cuadrático medio como lo indica la ecuación 5.2.1, para las imágenes Lena e "interview" [1], aplicando la LOT y la DCT para diferentes tasas de compresión. Lena e "Interview" son imágenes originalmente codificadas a 8 bpp en escala de gris. Lena tiene un tamaño de 256x256 pixeles e "Interview es un segmento de 256x256 pixeles de la imagen original.

Imagen: **Lena**

Medida: **Error Cuadrático Medio (MSE)**

| TASA (bpp) | LOT | DCT | Δ |
|---|---|---|---|
| **0.1** | 563.100235 | 649.293396 | 86.193161 |
| **0.2** | 337.866379 | 389.225342 | 51.358963 |
| **0.3** | 256.838882 | 289.965805 | 33.126923 |
| **0.4** | 204.402695 | 229.389999 | 24.987304 |
| **0.5** | 169.148407 | 185.186096 | 16.037689 |
| **0.6** | 141.793945 | 154.722336 | 12.928391 |
| **0.7** | 113.217407 | 126.367599 | 13.150192 |
| **0.8** | 93.995071 | 105.129913 | 11.134842 |
| **0.9** | 81.862183 | 87.644119 | 5.781936 |
| **1.0** | 74.772018 | 78.553757 | 3.781739 |
| **1.1** | 59.937195 | 62.452820 | 2.515625 |
| **1.2** | 48.774078 | 50.193100 | 1.419022 |
| **1.3** | 37.699005 | 40.005386 | 2.306381 |
| **1.4** | 33.506241 | 35.266739 | 1.760498 |
| **1.5** | 26.469391 | 27.764496 | 1.295105 |
| **1.6** | 23.025146 | 23.551727 | 0.526581 |
| **1.7** | 17.585770 | 18.276718 | 0.690948 |
| **1.8** | 14.234894 | 14.857544 | 0.62265 |
| **1.9** | 12.088654 | 12.615784 | 0.52713 |
| **2.0** | 9.605011 | 10.165176 | 0.560165 |

Tabla 5.3.2. Error Cuadrático Medio de la imagen Lena a diferentes tasas de compresión

[1] "Interview" fue donada por el "Centre Commun d'Etudes en Télévision et Télécommunications de France

Imagen: **"Interview"**

Medida: **Error Cuadrático Medio (MSE)**

| TASA (bpp) | LOT | DCT | Δ |
|---|---|---|---|
| **0.1** | 335.905380 | 365.973648 | 30.068268 |
| **0.2** | 237.510528 | 273.133072 | 35.622544 |
| **0.3** | 153.538391 | 199.614975 | 46.076584 |
| **0.4** | 117.099411 | 145.584534 | 28.485123 |
| **0.5** | 84.318832 | 103.486816 | 19.167984 |
| **0.6** | 63.416367 | 79.725311 | 16.308944 |
| **0.7** | 40.172958 | 49.000565 | 8.827607 |
| **0.8** | 28.170898 | 34.834732 | 6.663834 |
| **0.9** | 20.961899 | 25.660095 | 4.698196 |
| **1.0** | 17.144226 | 21.269196 | 4.12497 |
| **1.1** | 11.566391 | 13.029312 | 1.462921 |
| **1.2** | 8.499359 | 9.518845 | 1.019486 |
| **1.3** | 5.781845 | 6.457153 | 0.675308 |
| **1.4** | 5.242004 | 5.647156 | 0.405152 |
| **1.5** | 4.456177 | 4.715881 | 0.259704 |
| **1.6** | 4.214966 | 4.336899 | 0.121933 |
| **1.7** | 3.950317 | 3.997833 | 0.047516 |
| **1.8** | 3.876282 | 3.895889 | 0.019607 |
| **1.9** | 3.841562 | 3.844849 | 0.003287 |
| **2.0** | 3.826842 | 3.838501 | 0.011659 |

Tabla 5.3.3. Error Cuadrático Medio de la imagen "interview" a diferentes tasas de compresión

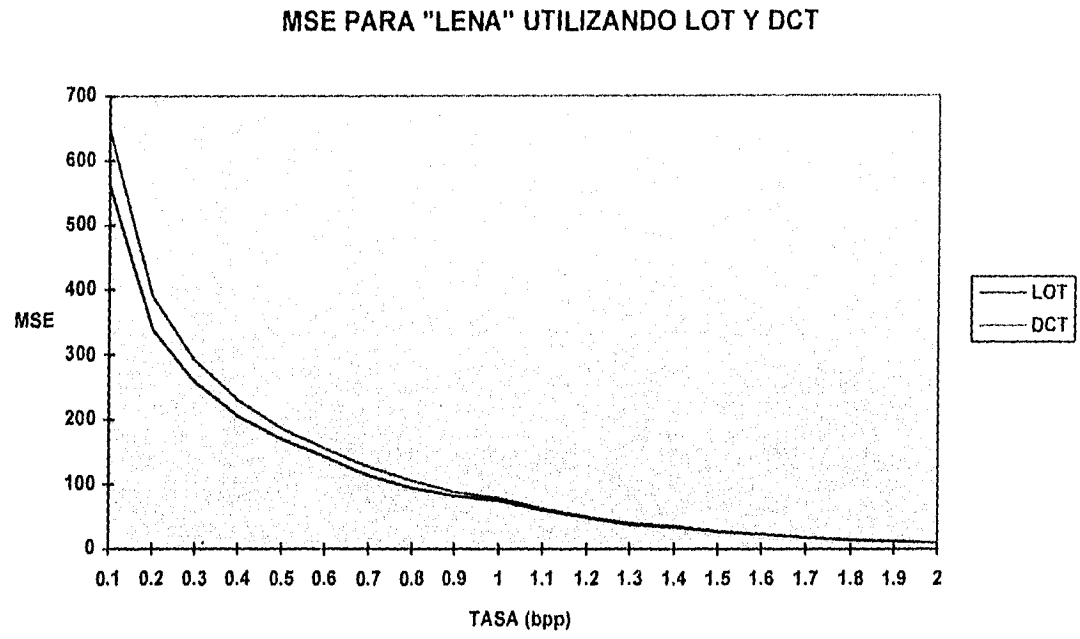



FIG. 5.3.2. Error Cuadrático Medio de la imagen Lena a diferentes tasas de compresión

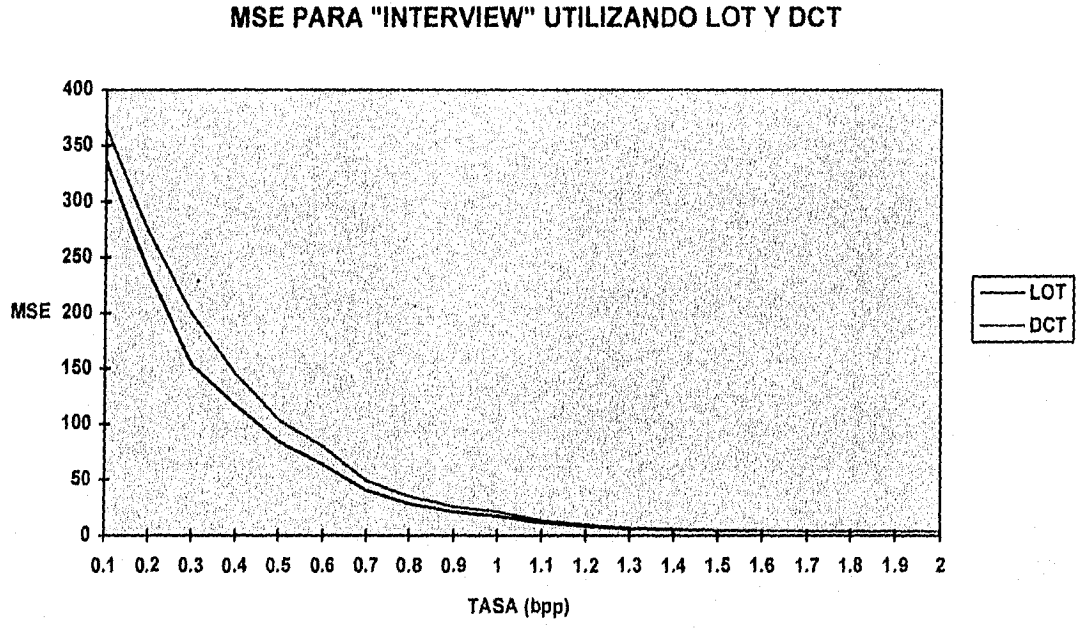



FIG.5.3.3. Error Cuadrático Medio de la imagen "interview" a diferentes tasas de compresión

## ERROR CUADRATICO MEDIO NORMALIZADO (NMSE)

El error cuadrático medio normalizado se calculó por medio de la ecuación 5.2.2, a continuación se muestran los resultados obtenidos para dos imágenes codificadas a diferentes tasas de compresión.

Imagen: **Lena**

Medida: **Error Cuadrático Medio Normalizado (NMSE)**

Norma: 12594.558502

| TASA (bpp) | LOT (%) | DCT (%) | Δ |
|---|---|---|---|
| **0.1** | 4.470980 | 5.155349 | 0.684369 |
| **0.2** | 2.682638 | 3.090425 | 0.407787 |
| **0.3** | 2.039285 | 2.302310 | 0.263025 |
| **0.4** | 1.622945 | 1.821342 | 0.198397 |
| **0.5** | 1.343028 | 1.470366 | 0.127338 |
| **0.6** | 1.125835 | 1.228486 | 0.102651 |
| **0.7** | 0.898939 | 1.003351 | 0.104412 |
| **0.8** | 0.746315 | 0.834725 | 0.08841 |
| **0.9** | 0.649981 | 0.695889 | 0.045908 |
| **1.0** | 0.593685 | 0.623712 | 0.030027 |
| **1.1** | 0.475898 | 0.495871 | 0.019973 |
| **1.2** | 0.387263 | 0.398530 | 0.011267 |
| **1.3** | 0.299328 | 0.317640 | 0.018312 |
| **1.4** | 0.266037 | 0.280016 | 0.013979 |
| **1.5** | 0.210165 | 0.220448 | 0.010283 |
| **1.6** | 0.182818 | 0.186999 | 0.004181 |
| **1.7** | 0.139630 | 0.145116 | 0.005486 |
| **1.8** | 0.113024 | 0.117968 | 0.004944 |
| **1.9** | 0.095983 | 0.100169 | 0.004186 |
| **2.0** | 0.076263 | 0.080711 | 0.004448 |

Tabla 5.3.4. Error Cuadrático Medio Normalizado de la imagen Lena a diferentes tasas de compresión

Imagen: **"Interview"**

Medida: **Error Cuadrático Medio Normalizado (NMSE)**

Norma: 22687.856171

| **TASA** (bpp) | **LOT** (%) | **DCT** (%) | Δ |
|---|---|---|---|
| **0.1** | 1.480551 | 1.613082 | 0.132531 |
| **0.2** | 1.046862 | 1.203873 | 0.157011 |
| **0.3** | 0.676743 | 0.879832 | 0.203089 |
| **0.4** | 0.516133 | 0.641685 | 0.125552 |
| **0.5** | 0.371647 | 0.456133 | 0.084486 |
| **0.6** | 0.279517 | 0.351401 | 0.071884 |
| **0.7** | 0.177068 | 0.215977 | 0.038909 |
| **0.8** | 0.124167 | 0.153539 | 0.029372 |
| **0.9** | 0.092393 | 0.113101 | 0.020708 |
| **1.0** | 0.075566 | 0.093747 | 0.018181 |
| **1.1** | 0.050981 | 0.057429 | 0.006448 |
| **1.2** | 0.037462 | 0.041956 | 0.004494 |
| **1.3** | 0.025484 | 0.028461 | 0.002977 |
| **1.4** | 0.023105 | 0.024891 | 0.001786 |
| **1.5** | 0.019641 | 0.020786 | 0.001145 |
| **1.6** | 0.018578 | 0.019116 | 0.000538 |
| **1.7** | 0.017412 | 0.017621 | 0.000209 |
| **1.8** | 0.017085 | 0.017172 | 0.000087 |
| **1.9** | 0.016932 | 0.016947 | 0.000015 |
| **2.0** | 0.016867 | 0.016919 | 0.000052 |

Tabla 5.3.5. Error Cuadrático Medio Normalizado de la imagen "interview" a diferentes tasas de compresión

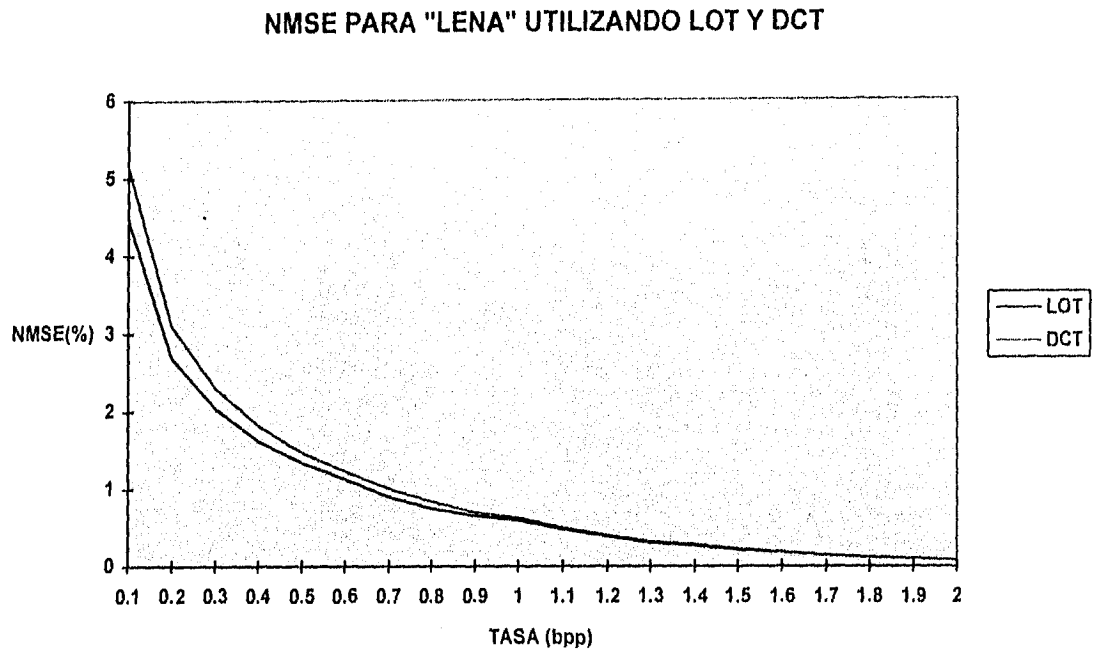



FIG. 5.3.4. NMSE de la imagen Lena a diferentes tasas de compresión

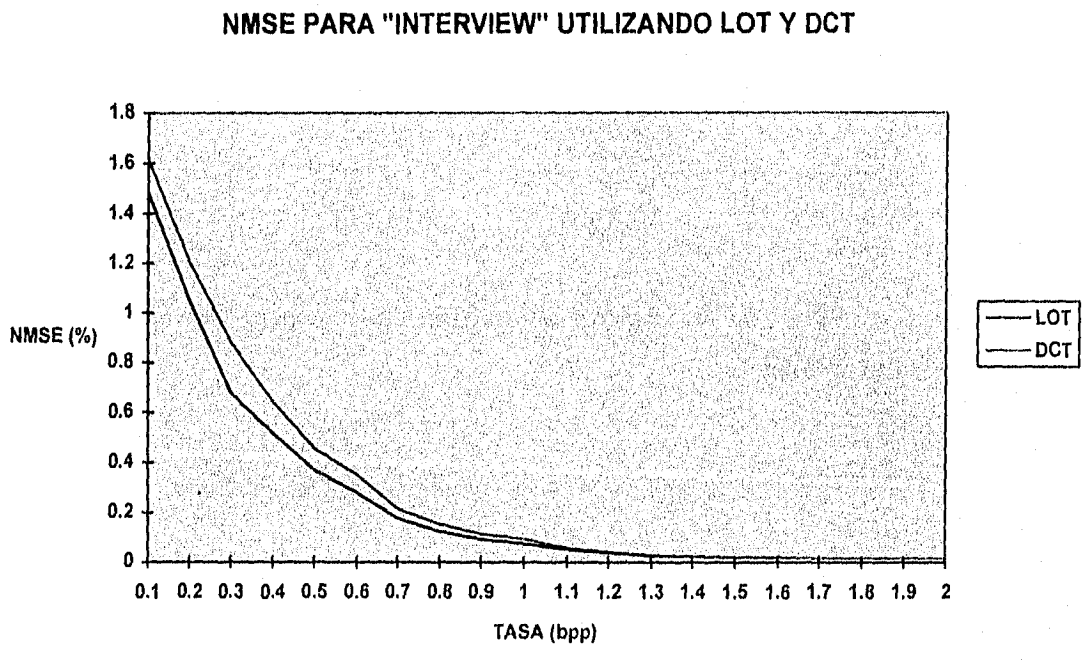



FIG. 5.3.5. NMSE de la imagen "interview" a diferentes tasas de compresión

**RELACION SEÑAL A RUIDO (SNR).**

La relación señal a ruido se calculó a partir del error cuadrático medio normalizado, por medio de la siguiente ecuación: $-10\log_{10}(NMSE)_a$. Los resultados obtenidos se muestran a continuación:

Imagen: **Lena**

Medida: **Relación Señal a Ruido (SNR)**

| TASA (bpp) | LOT (dB) | DCT (dB) | Δ |
|---|---|---|---|
| **0.1** | 13.495973 | 12.877419 | 0.618554 |
| **0.2** | 15.714379 | 15.099818 | 0.614561 |
| **0.3** | 16.905221 | 16.378362 | 0.526859 |
| **0.4** | 17.896962 | 17.396085 | 0.500877 |
| **0.5** | 18.719149 | 18.325745 | 0.393404 |
| **0.6** | 19.485253 | 19.106298 | 0.378955 |
| **0.7** | 20.462698 | 19.985471 | 0.477227 |
| **0.8** | 21.270778 | 20.784566 | 0.486212 |
| **0.9** | 21.870993 | 21.574600 | 0.296393 |
| **1.0** | 22.264439 | 22.050159 | 0.21428 |
| **1.1** | 23.224861 | 23.046313 | 0.178548 |
| **1.2** | 24.119940 | 23.995390 | 0.12455 |
| **1.3** | 25.238527 | 24.980648 | 0.257879 |
| **1.4** | 25.750580 | 25.528172 | 0.222408 |
| **1.5** | 26.774396 | 26.566938 | 0.207458 |
| **1.6** | 27.379810 | 27.281607 | 0.098203 |
| **1.7** | 28.550213 | 28.382847 | 0.167366 |
| **1.8** | 29.468293 | 29.282358 | 0.185935 |
| **1.9** | 30.178057 | 29.992667 | 0.18539 |
| **2.0** | 31.176861 | 30.930673 | 0.246188 |
| **Promedio** | 22.997369 | 22.678307 | 0.319062 |

Tabla 5.3.6. Relación Señal a Ruido de la imagen Lena a diferentes tasas de compresión

Imagen: **"Interview"**

Medida: **Relación Señal a Ruido (SNR)**

| TASA (bpp) | LOT (dB) | DCT (dB) | Δ |
|---|---|---|---|
| **0.1** | 18.295765 | 17.923436 | 0.372329 |
| **0.2** | 19.801106 | 19.194192 | 0.606914 |
| **0.3** | 21.695765 | 20.556003 | 1.139762 |
| **0.4** | 22.872387 | 21.926782 | 0.945605 |
| **0.5** | 24.298689 | 23.409084 | 0.889605 |
| **0.6** | 25.535921 | 24.541972 | 0.993949 |
| **0.7** | 27.518596 | 26.655924 | 0.862672 |
| **0.8** | 29.059928 | 28.137810 | 0.922118 |
| **0.9** | 30.343628 | 29.465352 | 0.878276 |
| **1.0** | 31.216756 | 30.280424 | 0.936332 |
| **1.1** | 32.925956 | 32.408720 | 0.517236 |
| **1.2** | 34.264073 | 33.772092 | 0.491981 |
| **1.3** | 35.937270 | 35.457524 | 0.479746 |
| **1.4** | 36.362961 | 36.039637 | 0.323324 |
| **1.5** | 37.068311 | 36.822306 | 0.246005 |
| **1.6** | 37.309994 | 37.186142 | 0.123852 |
| **1.7** | 37.591615 | 37.539688 | 0.051927 |
| **1.8** | 37.673781 | 37.651869 | 0.021912 |
| **1.9** | 37.712866 | 37.709142 | 0.003724 |
| **2.0** | 37.729529 | 37.716318 | 0.013211 |
| **Promedio** | 30.760745 | 30.219721 | 0.541024 |

Tabla 5.3.7. Relación Señal a Ruido de la imagen "interview" a diferentes tasas de compresión

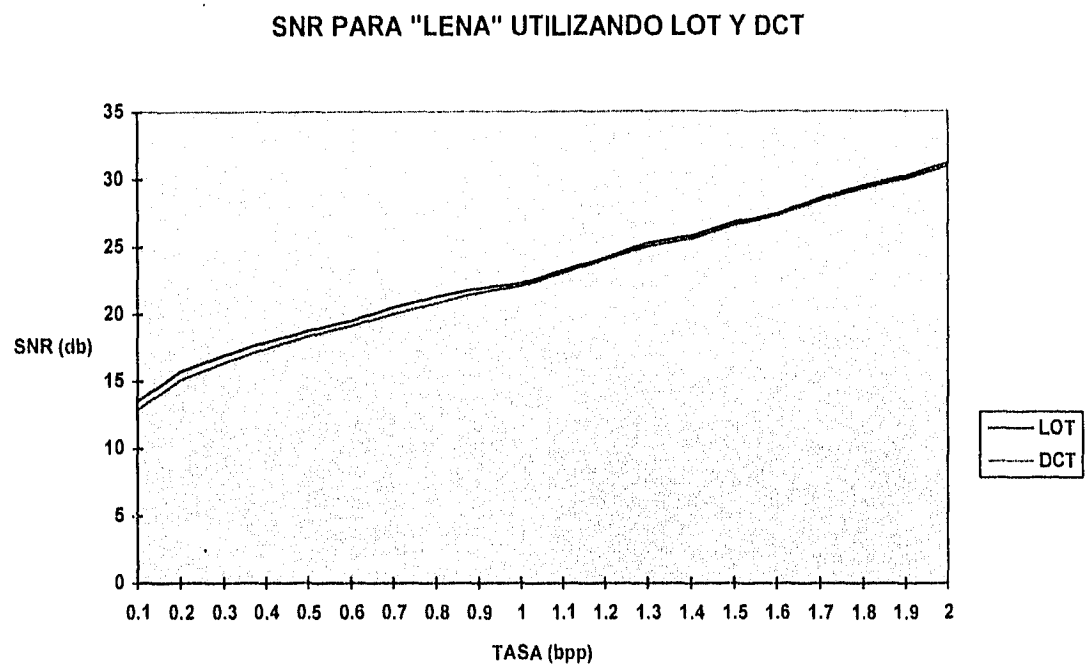



FIG. 5.3.6. Relación Señal a Ruido de la imagen Lena a diferentes tasas de compresión

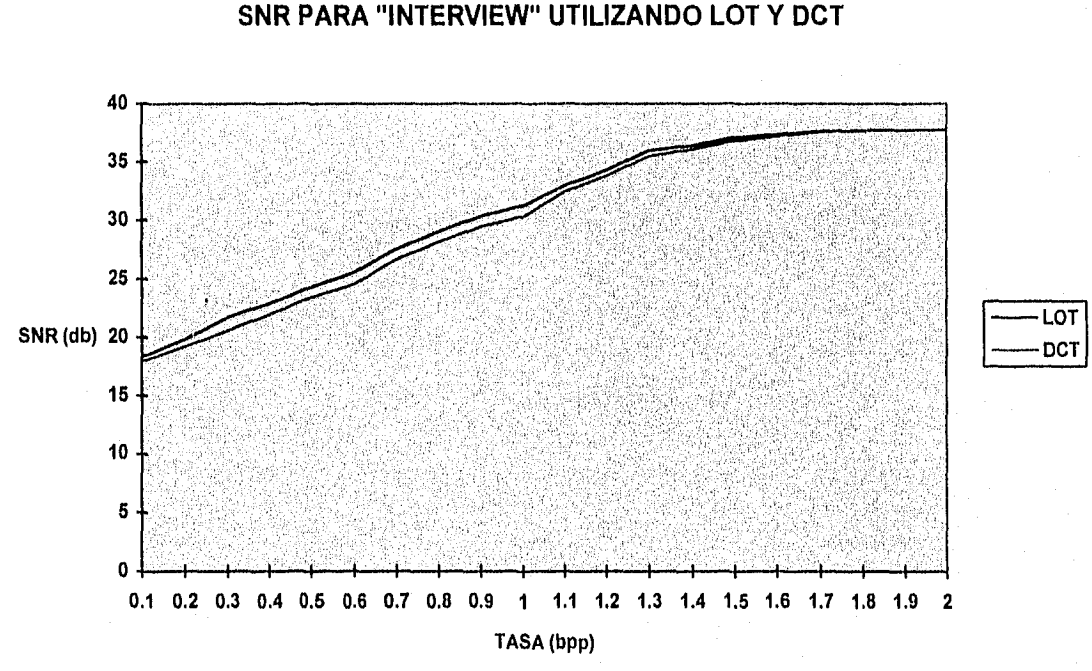



FIG. 5.3.7. Relación Señal a Ruido de la imagen "interview" a diferentes tasas de compresión

De acuerdo con los resultados mostrados anteriormente podemos decir que en ambos casos el desempeño de la LOT fue mejor que el de la DCT, en términos de relación señal a ruido la mejora fue de 0.32dB en promedio para Lena mientras que para "interview" la mejora fue de 0.54dB, de igual forma mejoraron el NMSE y el MSE.

### 5.3.2. PRUEBAS SUBJETIVAS.

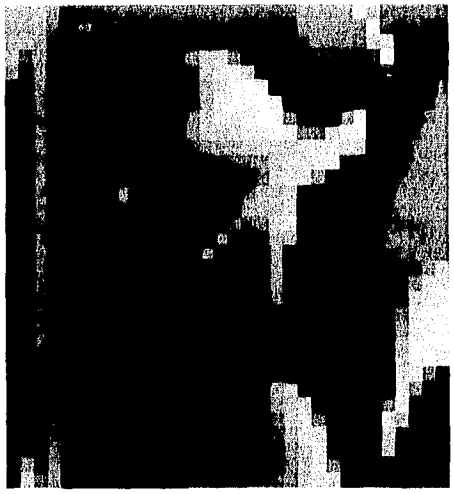
La evaluación subjetiva, se basa principalmente en la comparación de las imágenes obtenidas al aplicar la LOT y la DCT sobre el esquema de codificación a una misma tasa de compresión, principalmente para valorar la reducción de los efectos de interbloqueo, así como la calidad de las imágenes a diferentes tasas de compresión.

A continuación mostraremos las imágenes obtenidas en cada caso, proporcionando el nombre de la imagen, la tasa de compresión, la transformada (Tr) que fue utilizada dentro del esquema de codificación, la evaluación subjetiva (E.S.) y la evaluación de la reducción de los efectos de interbloqueo (R.I.), estos dos últimos aspectos, fueron evaluados de acuerdo con las tablas 5.2.1 y 5.2.2.

**Imagen: Lena**

Imagen original.

Tasa: 0.1 bpp
Tr: LOT
E.S.: 1
R.I.: 1



Tasa: 0.1 bpp
Tr: DCT
E.S.: 1
R.I.: 1

Tasa: 0.2 bpp
Tr: LOT
E.S.: 1
R.I.: 1

Tasa: 0.2 bpp
Tr: DCT
E.S.: 1
R.I.: 1

Tasa: 0.3 bpp
Tr: LOT
E.S.: 1
R.I.: 1

Tasa: 0.3 bpp
Tr: DCT
E.S.: 1
R.I.: 1

Tasa: 0.4 bpp
Tr: LOT
E.S.: 2
R.I.: 2

Tasa: 0.4 bpp
Tr: DCT
E.S.: 1
R.I.: 1

Tasa: 0.5 bpp

Tr: LOT

E.S.: 3

R.I.: 2

Tasa: 0.5 bpp

Tr: DCT

E.S.: 2

R.I.: 2

Tasa: 0.6 bpp

Tr: LOT

E.S.: 3

R.I.: 3

Tasa: 0.6 bpp

Tr: DCT

E.S.: 2

R.I.: 2

Tasa: 0.7 bpp

Tr: LOT

E.S.: 4

R.I.: 4

Tasa: 0.7 bpp

Tr: DCT

E.S.: 3

R.I.: 2

Tasa: 0.8 bpp

Tr: LOT

E.S.: 5

R.I.: 5

Tasa: 0.8 bpp

Tr: DCT

E.S.: 4

R.I.: 3

Tasa: 0.9 bpp
Tr: LOT
E.S.: 6
R.I.: 5

Tasa: 0.9 bpp
Tr: DCT
E.S.: 5
R.I.: 4

Tasa: 1.0 bpp
Tr: LOT
E.S.: 7
R.I.: 5

Tasa: 1.0 bpp
Tr: DCT
E.S.: 6
R.I.: 4

Tasa: 1.1 bpp
Tr: LOT
E.S.: 7
R.I.: 5

Tasa: 1.1 bpp
Tr: DCT
E.S.: 7
R.I.: 4

Tasa: 1.2 bpp
Tr: LOT
E.S.: 7
R.I.: 5

Tasa: 1.2 bpp
Tr: DCT
E.S.: 7
R.I.: 4

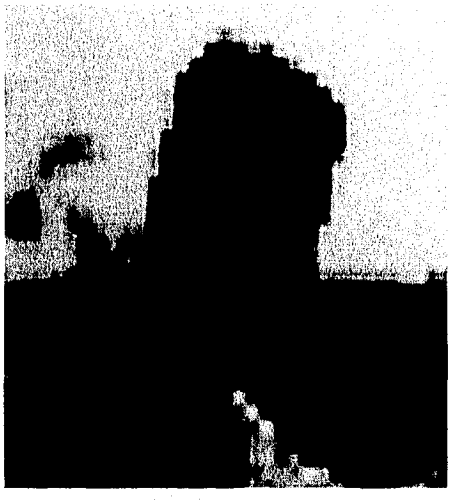
Tasa: 1.3 bpp
Tr: LOT
E.S.: 7
R.I.: 5

Tasa: 1.3 bpp
Tr: DCT
E.S.: 7
R.I.: 5

Como podemos observar en la secuencia de imágenes que hemos presentado, se distingue claramente el efecto que produce la LOT sobre las imágenes al reducir el efecto de interbloqueo, esto se aprecia mejor en las tasas más bajas como 0.1,0.2,0.3 y 0.4 bpp; sin embargo, a tasas más altas, también se aprecia una mejora en la calidad de la imagen, por ejemplo, a 0.9 bpp, el efecto de interbloqueo se presenta en el hombro de Lena cuando se cuantiza por medio de la DCT, mientras que para la LOT ya no se observa. De acuerdo con los resultados obtenidos de la evaluación subjetiva se obtiene una imagen de calidad perceptualmente perfecta a 1.0 bpp para la LOT, mientras que para la DCT se obtiene a una tasa de 1.3 bpp (a tasas mayores de 1.3 bpp la calidad de la imagen ya no mejora perceptualmente.) ; por lo que obtenemos una ganancia de 0.3 bpp. Cabe mencionar que los resultados obtenidos tras la evaluación subjetiva dieron mayores ventajas a la LOT que los obtenidos por medio de la evaluación cuantitativa debido a que las medidas de error utilizadas no reflejan fielmente los efectos de interbloqueo.

**Imagen: Interview**

Imagen original

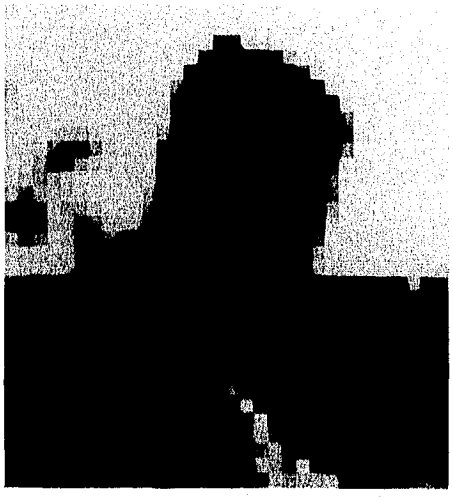


Tasa: 0.1 bpp

Tr: LOT

E.S.: 1

R.I.: 1



Tasa: 0.1 bpp

Tr: DCT

E.S.: 1

R.I.: 1

Tasa: 0.2 bpp

Tr: LOT

E.S.: 1

R.I.: 1

Tasa: 0.2 bpp

Tr: DCT

E.S.: 1

R.I.: 1

Tasa: 0.3 bpp

Tr: LOT

E.S.: 2

R.I.: 2

Tasa: 0.3 bpp

Tr: DCT

E.S.: 2

R.I.: 1

Tasa: 0.4 bpp

Tr: LOT

E.S.: 3

R.I.: 2

Tasa: 0.4 bpp

Tr: DCT

E.S.: 2

R.I.: 2

Tasa: 0.5 bpp

Tr: LOT

E.S.: 3

R.I.: 2

Tasa: 0.5 bpp

Tr: DCT

E.S.: 2

R.I.: 2

Tasa: 0.6 bpp

Tr: LOT

E.S.: 4

R.I.: 3

Tasa: 0.6 bpp

Tr: DCT

E.S.: 3

R.I.: 2

Tasa: 0.7 bpp

Tr: LOT

E.S.: 4

R.I.: 3

Tasa: 0.7 bpp

Tr: DCT

E.S.: 3

R.I.: 2

Tasa: 0.8 bpp
Tr: LOT
E.S.: 5
R.I.: 4

Tasa: 0.8 bpp
Tr: DCT
E.S.: 4
R.I.: 3

Tasa: 0.9 bpp
Tr: LOT
E.S.: 6
R.I.: 4

Tasa: 0.9 bpp
Tr: DCT
E.S.: 5
R.I.: 3

Tasa: 1.0 bpp
Tr: LOT
E.S.: 7
R.I.: 5

Tasa: 1.0 bpp
Tr: DCT
E.S.: 6
R.I.: 4

Tasa: 1.1 bpp
Tr: LOT
E.S.: 7
R.I.: 5

Tasa: 1.1 bpp
Tr: DCT
E.S.: 7
R.I.: 5

"Interview" se aproxima más rápido a la imagen original ya que es una imagen que tiene poco detalle por lo que los efectos de interbloqueo tienden a desaparecer más rápidamente; sin embargo siempre es notoria la diferencia entre las imágenes obtenidas por medio de la LOT y la DCT. En este caso obtuvimos la reconstrucción perceptualmente perfecta a 1.0 bpp al codificar con la LOT y a 1.1 bpp con la DCT, lo cual nos da una ganancia de 0.1bpp.

## 5.4 CONCLUSIONES

De acuerdo a los resultados cuantitativos y subjetivos presentados en la sección anterior podemos decir que el desempeño de la LOT es significativamente mejor que el de la DCT ya que se logró tanto una mayor tasa de compresión como una reducción en los efectos de interbloqueo lo que permite el diseño de sistemas que logren una excelente calidad de imágenes a tasas más bajas todo esto sacrificando ligeramente el tiempo de codificación ya que la complejidad de la LOT es superior a la de la DCT, sin que este incremento sea significativo tomando en cuenta el estado de la tecnología actual de procesadores digitales de señales. Otro factor que promueve la implementación de sistemas que utilicen la LOT es que es posible adaptar los sistemas actuales basados en la DCT, ya que la LOT se calcula por medio de la DCT y un conjunto de mariposas complejas.

# BIBLIOGRAFIA


Gersho Allen, Gray Robert M., Vector Quantization And Signal Compression, Kluwer Academic Publishers, E.U.A., 1992

Malvar Henrique S., Signal Processing with Lapped Transforms, Artech House, E.U.A., 1991, 357pp.

Rao K.R., P.YIP, Discrete Cosine Transform, Academic Press Inc., E.U.A., 1990, 490pp.

Rosenfeld Azriel, Digital Picture Processing, Academic Press Inc., E.U.A.,1982, Vol1

Lynch Thomas J., Ph. D., Data Compression Techniques and Applications, VNR (Van Nostrand Reinhold Company), E.U., 1985, 345pp.

Malvar Hernique S., Staelin David H., "The LOT: Transform Coding Without Blocking Effects", IEEE Transaction on Acoustics, Speech and Signal Processing (ASSP), Vol.37, No. 4, U.S.A., Abril 1989, pp. 553-559.

Gray Robert M., "Vector Quantization", IEEE ASSP, U.S.A., Abril, 1984, pp. 4-29.

Jayant Nikil, Johnston James, Jafranek Robert, "Signal Compression Based on Models of Human Perception", Proceedings of the IEEE, Vol. 81, No. 10, U.S.A., Octubre 1993, pp. 1385-1422.

Fagan A.D., "An introduction to the fast Fourier Transform", The marconi review, first quarter 1979, U.S.A., pp. 38-47.