



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

“ARAGON”

PROCESAMIENTO DIGITAL DE IMAGENES

T E S I S

Que para obtener el Título de:

INGENIERO EN COMPUTACION

Presenta:

LILIA LENCISO GARCIA

Asesor: Ing. Donaciano Jiménez Vázquez

TESIS CON
FALLA DE ORIGEN

San Juan de Aragón, Edo. de Méx. 1993



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE

INDICE.	I
PROLOGO.	VI
CAPITULO 1.	
ELEMENTOS PRINCIPALES DEL PROCESAMIENTO DIGITAL.	1
1.1 CONCEPTOS BASICOS.	2
1.1.1 SERIE Y TRANSFORMADA DE FOURIER.	2
1.2 IMAGEN DIGITAL.	6
1.3 TIPOS DE IMAGENES (CLASIFICACION DE IMAGENES).	8
1.4 CARACTERISTICAS DE LAS IMAGENES.	9
1.4.1 ELEMENTOS DEL PROCESAMIENTO DIGITAL.	9
1.4.2 ESCALA DE GRIS.	12
1.4.3 LUMINANCIA.	12
1.4.4 COLOR.	14
1.5 RESTRICCIONES.	18
1.5.1 RUIDO EN IMAGENES.	18
1.6 ALMACENAMIENTO DE IMAGENES DIGITALES.	23
1.7 IMAGENES MULTIESPECTRALES.	24
1.8 IMAGENES ESPACIALES (FOTOGRAFIA ESPACIAL).	25
CAPITULO 2.	
FUNDAMENTOS DEL PROCESAMIENTO DIGITAL DE IMAGENES.	30
2.1 MUESTREO.	31
2.2 CUANTIZACION.	38
2.2.1 CUANTIZACION DE UNA IMAGEN.	38
2.2.2 CUANTIZACION ESCALAR.	38
2.2.3 VECTOR DE CUANTIZACION.	40

2.2.4	PROCESO DE CUANTIZACION DE VARIABLES.	42
2.2.5	CUANTIZACION Y ASIGNACION DE PALABRA BINARIA (PCM).	45
2.2.6	CUANTIZACION DE UNA IMAGEN MONOCROMATICA.	47
2.2.7	CUANTIZACION DE UNA IMAGEN EN COLOR.	52
2.3	CODIFICACION.	53
2.3.1	CODIFICACION DE UNA IMAGEN.	53
2.3.2	MODULACION POR CODIFICACION DE PULSOS (PCM).	53
2.3.2.1	CODIFICACION PCM DE IMAGENES.	55
2.3.2.2	CODIFICACION DE PIXEL.	57
2.3.3.3	MODULACION DIFERENCIAL POR CODIFICACION DE PULSOS (DPCM).	58
2.3.4	MODULACION DELTA.	60
2.3.5	TECNICAS DE CODIFICACION DE IMAGEN.	64
C A P I T U L O 3.		
CUANTIZACION DE IMAGEN.		67
3.1	CUANTIZACION DE IMAGEN.	68
3.2	CUANTIZACION VISUAL.	69
3.2.1	CUANTIZACION DE CONTRASTE.	70
3.2.2	RUIDO PSEUDOALEATORIO DE CUANTIZACION.	71
3.2.3	CUANTIZACION DE IMAGENES HALFTONE.	72
3.2.4	CUANTIZACION DEL COLOR.	74
3.3	DISEÑO DE UN COMPARADOR.	74
3.4	CUANTIZACION PREDICTIVA.	76
C A P I T U L O 4.		
CODIFICACION DE IMAGENES.		79
4.1	CLASIFICACION DE LOS METODOS DE CODIFICACION.	80
4.2	TECNICAS DE CODIFICACION ESTADISTICA.	82
4.2.1	CODIFICACION SIMPLE POR PIXEL.	83
4.2.2	CODIFICACION PREVIA DE PIXEL.	84
4.3	CODIFICACION CORRIDA (RUN CODING).	84
4.3.1	CODIFICACION BIT-PLANE.	86
4.4	CODIFICACION PREDICTIVA.	86

4.4.1	MODULACION DELTA (DM).	87
4.4.2	DPCM.	88
4.5	TECNICAS DE CODIFICACION INTERPOLATIVA.	91
4.5.1	INTERPOLACION DE ORIGEN.	92
4.5.2	INTERPOLACION DE DESTINO.	93
4.6	CODIFICACION DE IMAGEN POR TRANSFORMADA.	94
4.6.1	CODIFICACION DE IMAGEN MONOCROMATICA POR TRANSFORMADA.	96
4.6.1.1	CODIFICACION POR ZONA.	98
4.6.1.2	CODIFICACION DE UMBRAL.	99
4.6.2	CODIFICACION ADAPTIVA POR TRANSFORMADA.	100
4.6.3	CODIFICACION DE IMAGENES EN COLOR POR TRANSFORMADA.	102
C A P I T U L O 5.		
TECNICAS PARA EL PROCESAMIENTO DIGITAL DE IMAGENES.		108
5.1	TRANSFORMACION DE IMAGENES.	109
5.1.1	TRANSFORMACION UNITARIA DE DOS DIMENSIONES.	109
5.1.2	TRANSFORMACION UNIDIMENSIONAL.	110
5.2	TRANSFORMADA DE FOURIER.	110
5.2.1	TRANSFORMADA RAPIDA DE FOURIER (FFT).	112
5.3	PROCESAMIENTO DE IMAGENES POR TRANSFORMADA.	115
5.3.1	TRANSFORMADA HAAR.	115
5.3.2	TRANSFORMADA HADAMARD.	116
5.3.3	TRANSFORMADA KARHUNEN-LOEVE.	118
5.3.4	TRANSFORMADA SLANT.	120
5.3.5	TRANSFORMADA COSENO.	122
5.3.5.1	TRANSFORMADA COSENO DE SIMETRIA SEMEJANTE.	123
5.3.5.2	TRANSFORMADA COSENO DE SIMETRIA IMPAR.	123
5.3.6	TRANSFORMADA SENO.	125
5.3.7	TRANSFORMADA SVD (DESCOMPOSICION DE VALOR SINGULAR).	125
5.4	COMPRESION DE IMAGENES.	127

5.4.1 COMPRESION DE DATOS VS. COMPRESION DE ANCHO DE BANDA.	129
C A P I T U L O 6.	
CONCLUSIONES Y EJEMPLOS DEL PROCESAMIENTO DIGITAL DE IMAGENES.	131
6.1 RESTRICCIONES PARA LA CONVERSION A/D.	132
6.1.1 VELOCIDAD DE MUESTREO.	132
6.1.2 IMAGENES MONOCROMATICAS Y.	132
6.2.3 IMAGENES DE COLOR.	135
6.2.3.1 SERIAL COMPUESTA.	137
6.2.3.2 SERIAL POR COMPONENTE.	138
6.2 CUANTIZACION Y CODIFICACION.	139
6.3 ALGUNOS EJEMPLOS DE IMAGENES CUANTIZADAS Y CODIFICADAS A 8 BITS 256 NIVELES.	140
6.3.1 ALGORITMOS.	140
6.4 TECNICAS DE COMPRESION DE IMAGENES.	143
6.4.1 TRANSFORMADA RAPIDA DE FOURIER.	143
6.4.2 TRANSFORMADA DISCRETA COSENO.	146
6.5 APLICACIONES.	151
CONCLUSIONES.	153
APENDICE.	156
BIBLIOGRAFIA.	167

P R O L O G O

La tesis que a continuación se presenta, se refiere al procesamiento digital de imágenes, cuyo objeto principal es el de dar las bases necesarias para poder digitalizar una imagen y una vez obteniendo esto, poder ver diferentes formas de procesamiento de la misma.

El tema del procesamiento digital de imágenes es de un desarrollo relativamente reciente, además de que se requiere de una computadora para su análisis y manipulación. Una de las causas por las cuales se utiliza la computadora, es su gran capacidad de procesamiento de datos, rapidez y almacenamiento de datos, pero debido a que las computadoras trabajan con datos que son más bien numéricos que gráficos, se hace necesario que para poder manipular una imagen, ésta sea convertida a su forma numérica.

En el Capítulo 1, se mencionan los elementos necesarios para realizar el procesamiento digital de una imagen, dichos elementos son los conceptos de lo que es una imagen, características principales, restricciones, clasificación y ejemplos de como son utilizadas después de haber realizado su procesamiento.

En el Capítulo 2, se describen los fundamentos del procesamiento digital, es decir las bases, se comienza describiendo el concepto que es la principal del procesamiento, el muestreo, ya que mediante este proceso se obtienen los puntos representativos de la imagen para posteriormente analizarlos, procesarlos o bien transmitirlos. También se mencionan los conceptos de cuantización y codificación, con la primera se

convierte la señal muestreada a valores procesables por una computadora y con la segunda se recupera la imagen a la vez que se obtiene una reducción de recursos para su posterior transmisión.

En los Capítulos 3 y 4, se describe de una forma más detallada los procesos de cuantización y codificación.

En el Capítulo 5, se presentan las diferentes técnicas de procesamiento digital de imágenes, en las que se incluye la transformación y compresión de las mismas. Con la transformación, se obtiene una mayor flexibilidad en el procesamiento, mejores algoritmos, obteniendo con esto, mejores formas de implementación. Y con la compresión, se almacena y transmite la imagen resultante de una forma más óptima.

En el último capítulo, se presentan algunos ejemplos prácticos de aplicación del procesamiento digital de imágenes, tomando como antecedente lo mencionado en capítulos anteriores. Además se incluyen programas de computación elaborados en lenguaje de alto nivel (lenguaje C) para dar una forma práctica de aplicación.

CAPITULO 1

ELEMENTOS PRINCIPALES DEL PROCESAMIENTO DIGITAL.

- 1.1 CONCEPTOS BASICOS.
 - 1.1.1 SERIE Y TRANSFORMADA DE FOURIER.
- 1.2 IMAGEN DIGITAL.
- 1.3 TIPOS DE IMAGENES (CLASIFICACION DE IMAGENES).
- 1.4 CARACTERISTICAS DE LAS IMAGENES.
 - 1.4.1 ELEMENTOS DEL PROCESAMIENTO DIGITAL.
 - 1.4.2 ESCALA DE GRIS.
 - 1.4.3 LUMINANCIA.
 - 1.4.4 COLOR.
- 1.5 RESTRICCIONES.
 - 1.5.1 RUIDO EN IMAGENES.
- 1.6 ALMACENAMIENTO DE IMAGENES DIGITALES.
- 1.7 IMAGENES MULTIESPECTRALES.
- 1.8 IMAGENES ESPACIALES (FOTOGRAFIA ESPACIAL).

1.1 CONCEPTOS BASICOS.

1.1.1 Serie y transformada de Fourier.

Quando es transmitida una información a través de un sistema de comunicación, la velocidad de transmisión esta relacionada con la rapidez con que las señales cambian en el tiempo. Por lo que cualquier señal que exista en un intervalo de tiempo es posible descomponerla en una serie de Fourier.

Existen dos representaciones de la serie de Fourier: la serie trigonométrica y la serie exponencial compleja, la primera se refiere al empleo de senos y cosenos en su serie o bien, al empleo de senos con ángulos de defasamiento y la segunda al empleo de términos exponenciales complejos e $\exp[\pm jn\omega_0 t]$.

La serie exponencial esta representada en la ecuación 1.1:

$$f(t) = C_0 + \sum_{n=1}^{\infty} \left[C_n \exp[jn\omega_0 t] + C_{-n} \exp[-jn\omega_0 t] \right] \quad (1.1)$$

$$f(t) = \sum_{n=-\infty}^{\infty} C_n \exp[jn\omega_0 t] \quad (1.2)$$

de donde:

$$C_n = \left[\frac{1}{T} \right] \int_{-T/2}^{T/2} f(t) \exp[-jn\omega_0 t] dt, \quad n = 0, \pm 1, \pm 2, \dots \quad (1.3)$$

En las figuras 1.1 y 1.2 se muestran dos ejemplos de funciones discretas en el tiempo (pulso rectangular y pulso triangular); si se obtiene la serie de Fourier para ambas funciones, se representarían como sigue:

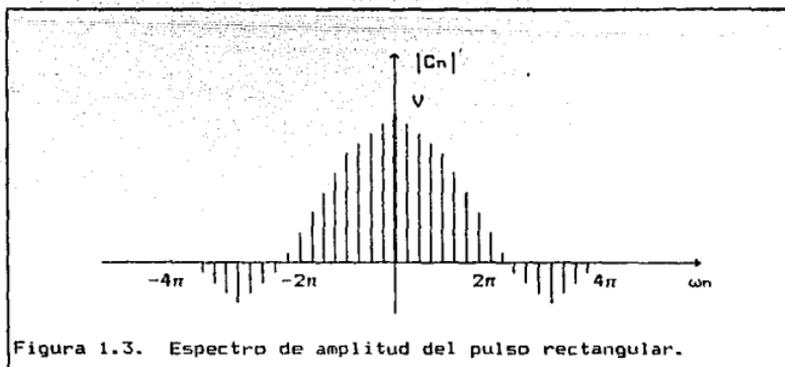
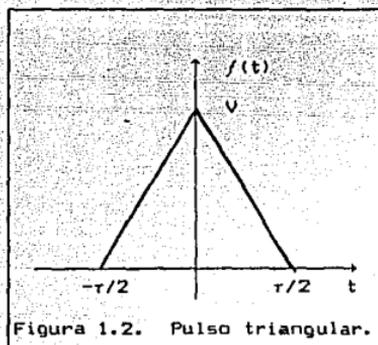
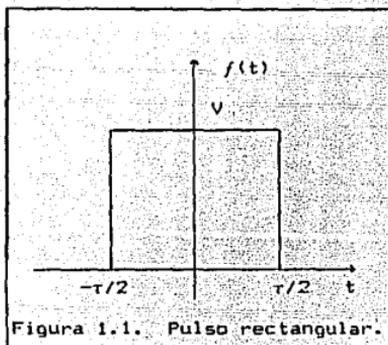
$$C_n = A \text{Sa} \left[\frac{n\pi}{2} \right], \quad (1.4)$$

para la figura 1.1 y

$$C_n = \left(\frac{A}{2}\right) \operatorname{sinc}^2\left(\frac{n}{2}\right), \quad (1.5)$$

para la figura 1.2.

Los espectros de amplitud para ambas funciones estarían representados como se muestra en las en las figuras 1.3 y 1.4.



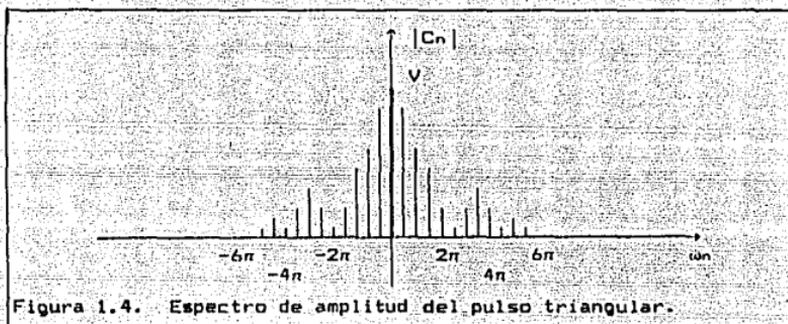


Figura 1.4. Espectro de amplitud del pulso triangular.

Si se observan ambos espectros, se observa que la figura 1.4 tiende más rápido a cero que la figura 1.3.

En la serie de Fourier se analizan funciones periódicas en el tiempo, pero para una aproximación más real es necesario utilizar funciones continuas en el tiempo. Para poder establecer la relación de tiempo-frecuencia de las funciones continuas se introduce el concepto de Transformada de Fourier.

La Transformada de Fourier también es conocida como Integral de Fourier esta definida por la ecuación:

$$F(\omega) = \mathcal{F}[f(t)] = \int_{-\alpha}^{\alpha} f(t) e^{-jn\omega t} dt \quad (1.6)$$

es decir,

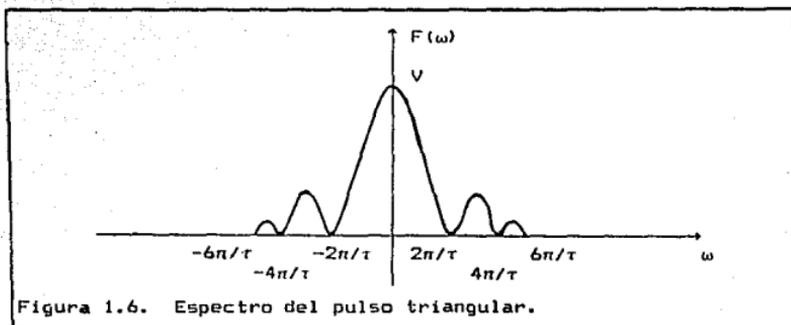
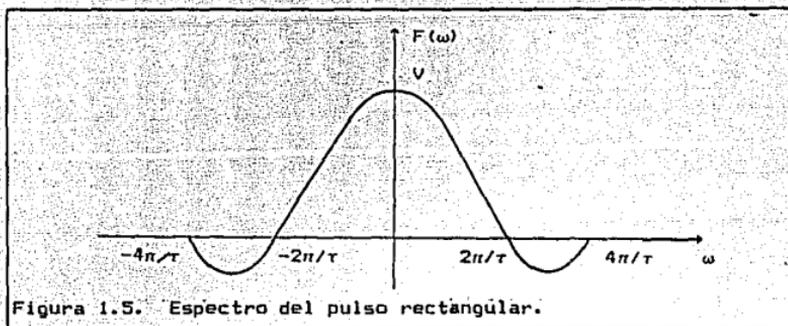
$$F(\omega) = \int_{-\alpha}^{\alpha} f(t) e^{-j\omega t} dt \quad \text{dado que } n\omega \rightarrow \omega \quad (1.7)$$

Si se obtiene la transformada de Fourier de la figura 1.1, resulta un espectro en frecuencia como el que muestra en la figura 1.5, cuyo valor es el siguiente:

$$F(\omega) = A\tau \text{Sa} \left[\frac{\omega\tau}{4} \right] \quad (1.8)$$

y en el caso de la figura 1.2, se tiene una espectro como el mostrado en la figura 1.6, con un valor de:

$$F(\omega) = \left[\frac{A\tau}{2} \right] \text{Sa}^2 \left[\frac{\omega\tau}{4} \right] \quad (1.9)$$



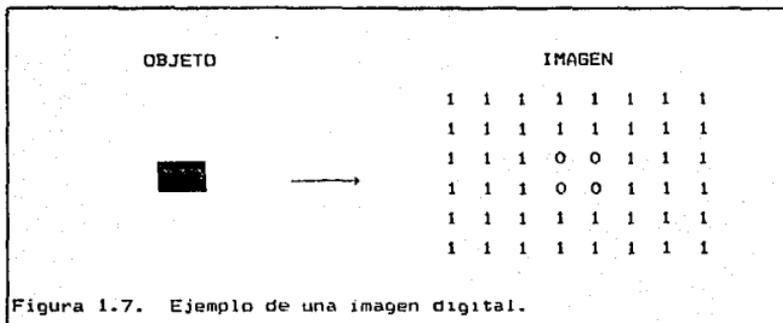
Observese como los espectros de la Serie de Fourier son semejantes

a los de la Transformada de Fourier lo cual implica la relación entre señales continuas y discretas en el tiempo¹.

1.2 IMAGEN DIGITAL.

Una imagen es una representación, semejanza o imitación de un objeto o cosa; una descripción de una gráfica o animación, una presentación de algo para representar otro algo.

Una imagen digital se considera como un arreglo matricial de números. Un ejemplo es mostrado en la figura 1.7, la escena es un cuadro negro sobre un fondo blanco, en la misma figura se presenta una posible representación digital. Cada punto dentro de la representación digital corresponde a una área de un objeto en el espacio; un valor digital es asignado a cada punto dentro de la imagen digital relacionando la intensidad del área del objeto en el espacio.

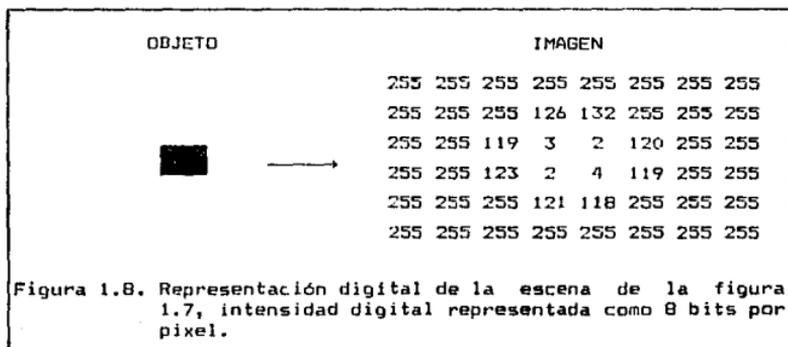


¹ Para más detalles vease Capítulo 2, Mischa Schwartz, Transmisión de información modulación y ruido.

El componente individual de los elementos dentro de la imagen digital son referidos como picture elements. Dos abreviaciones comunes para este término son pixel y pel. El valor digital de cada pixel representa la intensidad digital. Usando esta notación, la intensidad digital del pixel de la línea 3, muestra 4, en la imagen mostrada en la figura 1.7 es cero. En una imagen en blanco y negro, una intensidad de cero representa negro.

La figura 1.8 contiene una representación digital diferente de la misma escena mostrada en la figura 1.7; la misma frecuencia de muestreo (que será analizada posteriormente) ha sido usada para generar esta versión digital, pero el número de bits usados para representar la intensidad digital ha sido incrementada de 1 a 8. Los valores de la intensidad digital pueden ser de un rango de cero a 255, con cero se representa el negro y con 255 se representa el blanco.

Los efectos que ocurren en una representación digital cuando se muestrea una escena continua pueden ser vistos en la figura 1.8. Primero, los valores de intensidad digital representan el cuadro negro. Los cuatro pixeles representan la uniformidad del cuadro negro tienen valores de intensidad digital de 3, 2, 2, y 4.



La mayoría de los sistemas están provistos de representaciones de muestras digitales de una escena, la cual presenta un grado de ruido no deseado dentro de la imagen muestreada. Si no se presenta ruido durante el proceso de muestreo, la intensidad de los valores digitales de los 4 píxeles sería la misma.

Un segundo efecto que puede ser observado es la falta de una transición aguda en la imagen digital muestreada a los bordes del cuadro negro. En la figura 1.7 el cuadro es claramente definido en los límites, pero en la figura 1.8 se presentó lo que realmente ocurre cuando el límite es muestreado para crear una representación digital discreta de la escena.

1.3 TIPOS DE IMÁGENES (CLASIFICACION DE IMÁGENES).

Las imágenes se presentan en varias formas: visibles, no visibles, abstractas, reales, adecuadas para el análisis por computadora y no adecuadas. Las imágenes se clasifican tomando en cuenta la forma o método con el cual son generadas.

Si consideramos como un conjunto a todos los objetos, las imágenes forman un subconjunto (figura 1.9).

Dentro del subconjunto de las imágenes, se encuentran varios subconjuntos, uno de ellos es el de las imágenes visibles (percibidas por el ojo humano), en éste se encuentran otros subconjuntos: fotografías, dibujos, pinturas; también está el subconjunto de las imágenes ópticas (generadas por lentes y hologramas).

Las imágenes físicas no visibles son distribuciones de propiedades físicas medibles, por ejemplo, temperatura, presión, altitud, densidad de población. El otro subconjunto contiene las imágenes abstractas, entre las que se encuentran las funciones

matemáticas, las funciones continuas y discretas o imágenes digitales (sólo las imágenes digitales pueden ser procesadas por la computadora).

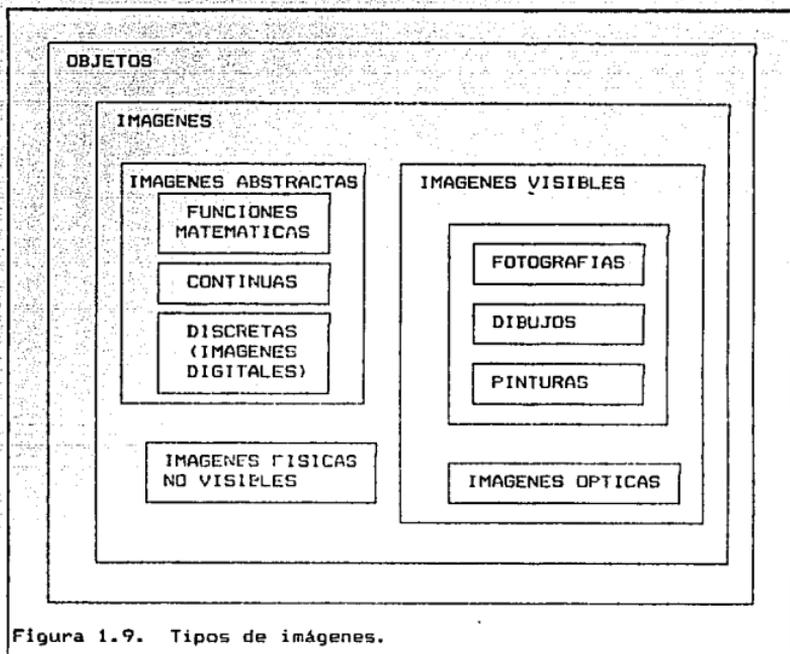
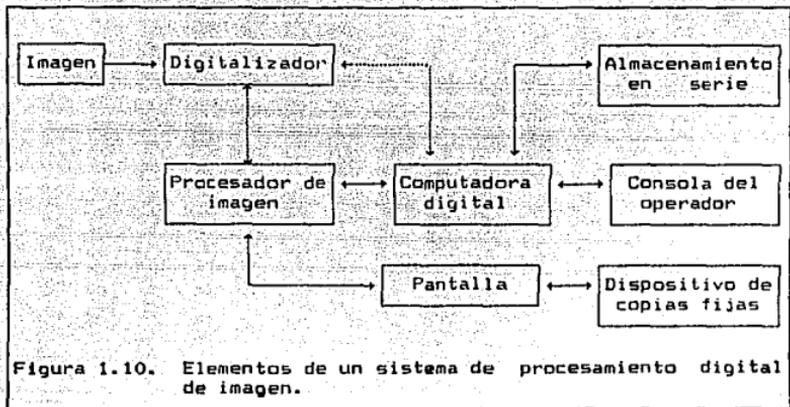


Figura 1.9. Tipos de imágenes.

1.4 CARACTERISTICAS DE LAS IMAGENES.

1.4.1 Elementos del procesamiento digital.

Los elementos básicos de un sistema de procesamiento digital de imagen son mostrados en la Figura 1.10.



A) Procesador de imagen.

El procesador de imagen es el corazón de un sistema de procesamiento digital de imagen. Un procesador de imagen consiste en módulos hardware que ejecutan cuatro funciones básicas: adquisición de la imagen, almacenamiento, procesamiento de bajo nivel y despliegue.

El módulo de adquisición de imagen, también llamado frame grabber, generalmente tiene como entrada una señal de video de TV y convierte esta señal a su forma digital. El módulo de almacenamiento, también llamado memoria de refrescamiento (frame buffer), es una memoria capaz de almacenar una imagen digital entera. El módulo de procesamiento ejecuta funciones de bajo nivel como operaciones aritméticas y lógicas, este módulo es también llamado Unidad Aritmética Lógica (ALU). La función del módulo de despliegue es de leer una imagen de memoria, convertir la información digital a una señal de video analógica y direccionar la señal de salida a un monitor u otro dispositivo.

B) Digitalizador.

Un digitalizador convierte una imagen a una representación numérica propia para una entrada en la computadora digital. Entre los dispositivos más comunmente usados están los microdensitómetros, lentes luminosos de los scanners, analizadores de imagen, camaras vidicon y arrellos fotosensitivos de estado sólido. Los dos primeros requieren de la imagen digitalizada, en forma de transparencia o fotografía; los restantes también las aceptan pero tienen la ventaja adicional de digitalizar imágenes naturales que tengan suficiente intensidad de luz para excitar el detector.

C) Computadora digital.

Los sistemas de computación usados para procesar una imagen, van desde microprocesadores hasta grandes sistemas de computación capaces de ejecutar funciones computacionalmente intensivas dentro de grandes arreglos de imagen. Los parámetros que influyen en la estructura de una computadora depende de la aplicación y los datos que se requieran para llevarla a cabo. Una imagen digital requiere de gran cantidad de memoria durante su procesamiento.

D) Dispositivos de almacenamiento.

Una imagen digital consiste en 512 x 512 píxeles, cada uno de los cuales es cuantizado dentro de ocho bits y requiere de 0.25 megabytes de almacenamiento. Los tres dispositivos de almacenamiento usados para este trabajo son discos magnéticos (700 Mb o más de capacidad) almacenan 2,800 imágenes del tamaño antes mencionado; cintas magnéticas (6,400 bytes por cada 2.5 cm.) almacenan una imagen por cada cm. y discos ópticos (4 Gb de capacidad) almacenan aproximadamente 16, 000 imágenes por disco.

E) Despliegue y dispositivos de grabación.

Los monitores monocromáticos y en color son los principales dispositivos de despliegue usados en los sistemas de procesamiento de imágenes actuales. Los monitores son direccionados por la(s) salidas del módulo de despliegue de imagen dentro del procesador de imagen. Estas señales pueden ser alimento dentro del dispositivo de grabación de imagen cuya función produce una copia fija (pantalla fotográfica, fotografías, y transparencias) de la esencia de la imagen dentro de la pantalla del monitor. Otros medios de despliegue incluye el tubo de rayos catódicos (TRC) y dispositivos de impresión.

1.4.2 Escala de gris

Las imágenes digitalizadas tienen únicamente dos niveles de la escala de gris: blanco y negro. Se pueden tener 8 bits 256 niveles que van desde el blanco hasta el negro, pasando por una escala de gris con una alta resolución. La resolución de una imagen depende en gran parte a los niveles de la escala de gris.

El ojo humano puede diferenciar aproximadamente 40 diferentes matices de gris, en la práctica el número empleado rebasa la capacidad del ojo humano.

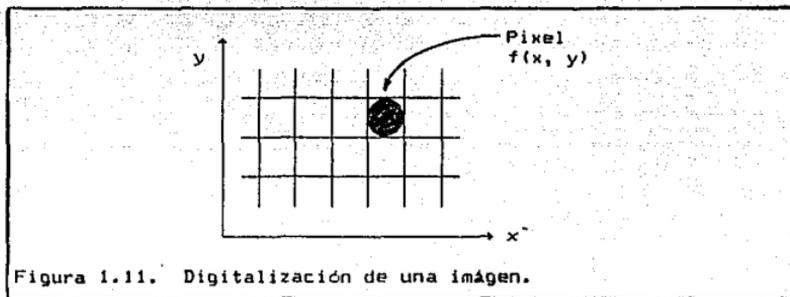
El término imagen se refiere a una función de dos dimensiones luz-intensidad, denotada por $f(x, y)$, en donde el valor o amplitud de f esta dado en coordenadas espaciales. (x, y) da la intensidad (brillo) de la imagen en un punto. (x, y) es proporcional a la brillantez (nivel de gris) de la imagen en un punto (figura 1.11).

1.4.3 Luminancia.

Desde que la luz es una forma de energía, $f(x, y)$ debe ser

finita y no ser cero, esto es,

$$0 < f(x, y) < \alpha \quad (1.10)$$



Las imágenes que percibimos en nuestras actividades diarias consisten de luz reflejada de los objetos. La naturaleza básica de $f(x, y)$ puede ser considerada como su esencia caracterizada por dos componentes. Un componente es la suma del origen de la luz, incidida sobre la escena vista, mientras que el otro es la suma de la luz reflejada por el objeto en la escena. Estos componentes son llamados iluminación y reflejo de los componentes, y son denotados por $i(x, y)$ y $r(x, y)$, respectivamente. Las funciones $i(x, y)$ y $r(x, y)$ combinadas como un producto forman $f(x, y)$:

$$f(x, y) = i(x, y) r(x, y) \quad (1.11)$$

donde

$$0 < i(x, y) < \alpha \quad (1.12)$$

y

$$0 < r(x, y) < 1 \quad (1.13)$$

La ecuación 1.13 indica el factor de reflejancia, el cual es limitado por 0 (total absorción) y por 1 (total reflejancia). La naturaleza de $i(x, y)$ es determinada por el origen de la luz, mientras que $r(x, y)$ es determinada por las características de el

objeto en la escena.

La intensidad de una imagen monocromática f en coordenadas (x, y) es llamada nivel de gris (l) de la imagen en ese punto. Desde la ecuación 1.11 hasta la 1.13, es evidente que l es falso en el rango

$$L_{\min} \leq l \leq L_{\max} \quad (1.12)$$

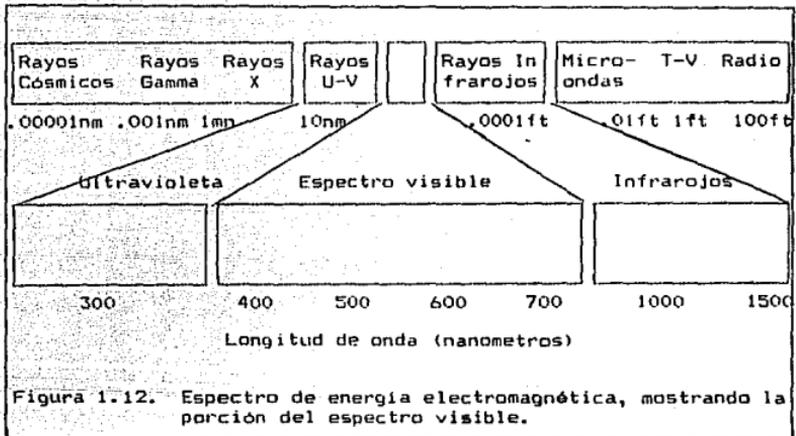
En teoría, se requiere que L_{\min} sea positivo, y L_{\max} sea finito. En la práctica, $L_{\min} = (i_{\min}) (r_{\min})$ y $L_{\max} = (i_{\max}) (r_{\max})$. Usando por encima los valores de iluminación y reflejancia como una norma, uno puede esperar los valores de L_{\min} es de aprox. 0.005 y para L_{\max} es de aprox. 100 dentro de aplicaciones del procesamiento de imágenes.

EL intervalo $[L_{\min}, L_{\max}]$ es llamado escala de gris. En la práctica es común variar estos intervalos numéricos a intervalos $[0, L]$, donde $l = 0$ es considerado como negro y $l = L$ es considerado como blanco en esta escala. Todos los valores intermedios son considerados matices de gris variando continuamente desde el negro al blanco.

1.4.4 Color

La luz es una forma de energía electromagnética esto es, la imagen esta dada por la distribución de ondas largas. No toda la radiación electromagnética es visible para el ojo humano. De hecho, la porción entera visible de la radiación es sólo dentro del límite de la banda de onda larga de 380 a 780 nm. (Figura 1.12). Esta radiación cuando incide sobre el ojo produce una variedad de sensaciones, por ejemplo, la sensación de brillo producida por el conjunto de varias ondas largas. Sin embargo, la luz de varias ondas largas también producen otra importante información visual llamada color.

Las ondas largas de la luz producen un estímulo visual. Como estas ondas varían, los colores percibidos (o, mejor dicho el matiz) de los cambios de luz varían, desde el rojo (corresponde a la onda larga más larga) pasando por el naranja, amarillo, verde y del azul al violeta.



Las mezclas de las ondas largas producen la percepción de matices, algunos de los cuales (por ejemplo, el púrpura) no pueden ser obtenidos desde una sola onda larga. La luz blanca (o gris) también puede ser obtenida por una mezcla de colores. De hecho, cualquier matiz puede ser obtenido por una mezcla de tres colores primarios (rojo, verde y azul) en proporción adecuada (figura 1.13).

El color es un importante aspecto de energía radiante visible por el cual un observador puede distinguir entre diferentes composiciones espectrales. Por eso, un estímulo de color, es especificado por energía radiante visible de una intensidad dada y

una composición espectral.

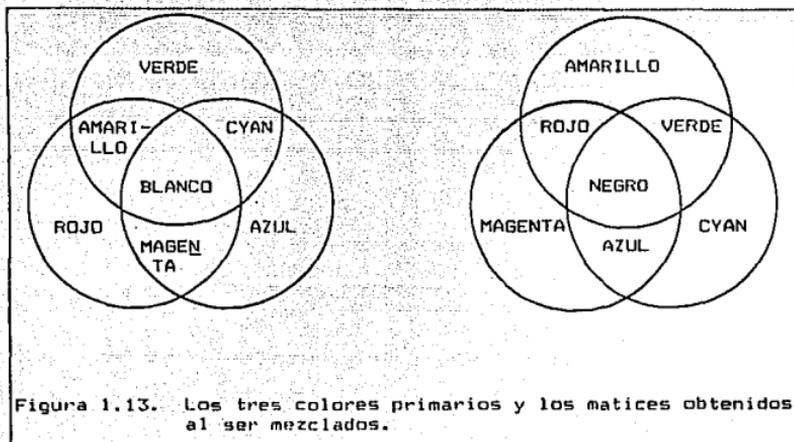


Figura 1.13. Los tres colores primarios y los matices obtenidos al ser mezclados.

El color es generalmente caracterizado por juntar nombres para los diferentes estímulos, por ejemplo, blanco, gris, negro, rojo, verde, azul. Los estímulos de color son generalmente más agradables para el ojo humano que el estímulo "blanco y negro". El colorido de imágenes blanco y negro por transformaciones de intensidad dentro de los colores es llamado pseudo-color. Esta adición de color regularmente se realiza en proporción a la escala de gris, esto es, a cada valor le corresponderá una determinada escala de gris.

La especificación del color consiste de tres partes: (a) semejanza del color, (b) diferencias de color, (c) apariencia del color (o percepción del color).

El estudio del color es importante en diseño y desarrollo de sistemas de visión del color. El uso del color en el despliegue de

Imágenes no sólo es más agradable, sino que facilita la recepción de más información visual. El ojo humano puede distinguir sólo cerca de 32 sombras de gris discreto; por el contrario es menos limitado en la discriminación del color y es capaz de distinguir por arriba de varios cientos de sombras de color discreto. Por esta razón, el color es una importante herramienta en el realce de la imagen.

El color varias veces es usado en la interpretación de imágenes que podrían desplegarse en blanco y negro. Cuando esto es así, el color es adherido como un mecanismo de ayuda al ser humano en la interpretación del análisis de datos. Una imagen de 8 bits contiene intensidades desde cero hasta 255 que pueden ser transformadas dentro de una imagen en color. El color producido en una imagen por translación a una imagen monocromática dentro de la presentación de colores es llamada falso color de imagen. Esto no tiene relación entre el color producido en el falso color de imagen y el verdadero color del objeto que fué imaginado.

El falso color en las imágenes es producido por intensas transformaciones. La intensidad de los rangos de entrada del digitalizador de imagen es transformada dentro de tres rangos de intensidad separados a la salida: uno es para el rojo, otro para el verde y el otro para el azul, estos componentes de un color salen de la composición de la imagen, es decir, de una entrada particular de imagen, se producen tres salidas de imagen.

Los atributos de la percepción son el brillo, el matiz y la saturación. Los brillos representan la luminancia percibida. El matiz de un color se refiere a la "rojura", "verdor" y así sucesivamente. Para el origen de luces monocromáticas, la diferencia en matices son manifestados por la diferencia de ondas largas. La saturación es ese aspecto, es la percepción de más y más luz sumada a una luz monocromática. Estas definiciones son algo imprecisas porque el matiz, la saturación y el brillo cambian

cuando la onda larga, la intensidad, el matiz o la suma de la luz blanca en un color es cambiado.

1.5 RESTRICCIONES.

1.5.1 Ruido en las imágenes

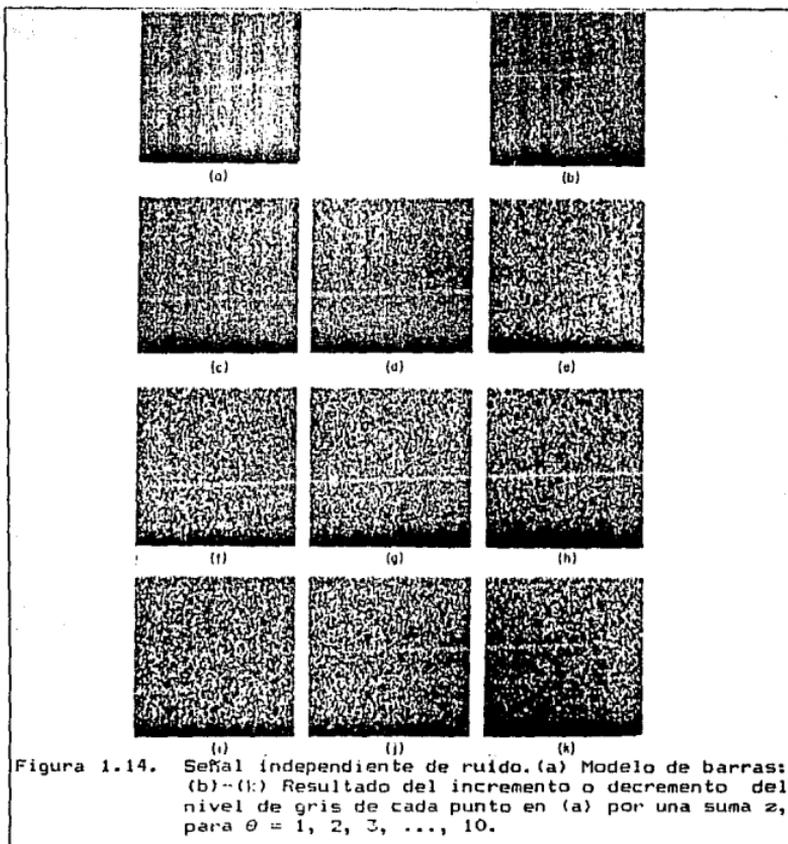
Las imágenes son subjetivas a diferentes tipos de ruido. Algunos son independientes de la señal de la imagen, otras no; algunas son "coherentes", mientras que otras no son correlacionales de punto a punto.

Cuando una imagen es transmitida, el ruido de canal se introduce y su valor es generalmente independiente de la potencia de la señal. Una situación similar se presenta cuando una imagen es rastreada por una cámara vidicon de televisión. En este caso, se puede escribir $g = f + v$, donde el ruido v y la entrada de la imagen f no son correlacionales.

En la figura 1.14, se muestran los efectos del ruido de una imagen independiente. Las barras tienen un nivel de gris constante s , y el espacio tiene nivel de gris r . Cada punto tiene un nivel de gris incrementado o decrementado por una suma z , escogidos aleatoriamente para que caiga en el rango $|z| \leq \theta$ ($s - r$), para varios valores de θ . Se nota la igualación cuando el rango de niveles de ruido excede las diferencias entre las barras y el fondo; las barras son apenas visibles.

Las estimaciones de los niveles de gris de la imagen cuando el ruido está presente, requieren de conocimientos de estática de la imagen y del ruido. Si se conoce la no ruidosidad que la imagen contiene en una gran región del nivel de ruido constante, se puede obtener un discernimiento dentro del ruido estático, por análisis de las fluctuaciones del nivel de gris en la región de la imagen

ruidosa.



En muchos casos el nivel de ruido depende de la señal de la imagen, por ejemplo, cuando una imagen es rastreada por el lente luminoso del scanner. Si el ruido es proporcional a la señal, $g = f + \alpha f$, se tiene $g = f(1 + \alpha) = \beta f$, de modo que se puede

juzgar esta situación como una no correlación de ruido, esto es algo multiplicativo que aditivo. Un ejemplo, es el ruido "coherente" en una imagen de televisión por la presencia del rastreador de líneas de TV; aquí v es un modelo de barra máxima sobre las líneas del rastreador y ceros entre estos. Un ejemplo menos trivial es la fotografía granulada, al fotografiar imágenes se forman brinco en revelado por "granos" en la emulsión.

El ruido proporcional de la señal de la figura 1.14, es análoga al de la figura 1.15 excepto que en la figura 1.14a los rangos de ruido para puntos sobre las barras y sobre los espacios son proporcionales a s y r , respectivamente. Se nota que las barras son algo menos visibles en este caso, porque el rango del ruido para los espacios no es un subintervalo de los rangos para las barras, como en la figura 1.15 (b).

Un tipo importante de ruido en las imágenes digitales, es el ruido de cuantización (o error de cuantización), el cual es la diferencia entre una imagen cuantizada y la original. Para una probabilidad de densidad de los niveles de gris, este error podría ser minimizado por selección debido a los niveles de cuantización.

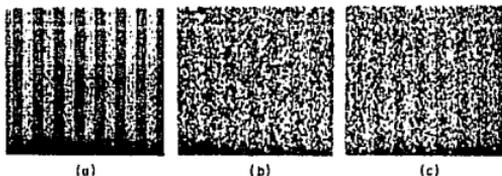
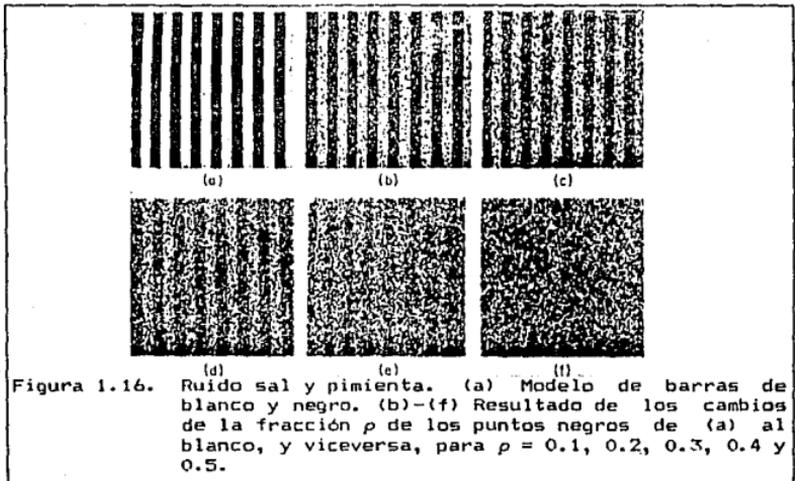


Figura 1.15. Señal dependiente del ruido. (a) El modelo de barras tiene dos niveles de gris, r y s . (b) Resultado de la adición del ruido. (c) Resultado del uso del promedio de los dos rangos de ruido de (b) a cada punto de (a); aquí el ruido no es dependiente de la señal más larga.

Muchas veces se quiere la conversión de una imagen que contiene sombras de gris dentro de una imagen "blanco y negro" por el umbral. El resultado a la salida de la imagen puede ser considerado como parte de un "objeto" negro dentro de un "fondo" blanco, o viceversa. Si el objeto original fuera también ruidoso, la región negra podría contener puntos blancos dispersos, y la región blanca puntos negros dispersos. Esta condición es conocida como ruido sal y pimienta. De manera general, se puede aplicar este término a cualquier situación en la cual los puntos dispersos de una imagen son marcadamente oscuros o luminosos que su inmediato circundante.

La figura 1.16 muestra un ejemplo del ruido sal y pimienta, un modelo de barras blancas y negras en el cual una fracción ρ de los puntos blancos han sido cambiados para negros y viceversa, donde $\rho = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ en las figuras 1.16a - 1.16f, respectivamente.



El ruido puede depender no solamente del nivel de gris de la imagen en un punto dado, también depende de los niveles en puntos cercanos.

Una imagen también se procesa al aplicarse una transformación inversa (por ejemplo, la transformada de Fourier), y la aplicación de la transformación inversa genera la necesidad de procesar la imagen como salida. Los errores (o ruido) que pueden introducirse por las operaciones en el dominio de la transformación, son importantes para analizar los efectos de esos errores en sobre la imagen de salida.

Si el ruido de la imagen independiente es sumado para la transformación de Fourier de una imagen, la reconstrucción de la imagen deseada también contiene ruido independiente, el cual por el Teorema de Parseval² tendrá el mismo efecto. Un resultado similar se mantiene si el ruido es sumado para el espectro de poder de la imagen.

El ruido de cuantización en el dominio de Fourier es un caso especial, dado que el rango de valores al hacer la transformación de una imagen es generalmente mejor que la imagen misma, con más de los poderes esenciales encontrados bajo frecuencias espaciales. Esto es especialmente importante para el uso desigual del espacio de niveles de cuantización cuando se cuantize una transformación de Fourier.

Los errores en la fase de cuantización de la transformación tienden a tener malos efectos en la reconstrucción de imágenes lo que provoca errores en la amplitud de cuantización.

² Véase, Capítulo 1, Taub Herbert, Principios Communication Systems, Mc Graw-Hill.

1.6 ALMACENAMIENTO DE IMÁGENES DIGITALES

El desarrollo de la tecnología computacional ha hecho posible la utilización de técnicas de imágenes digitales económicas en la operación de sistemas. A principios de los 70's, surgen en el mercado los primeros desplegados de imagen volátiles y archivadores de película. Los desplegados volátiles fueron los precursores de los de hoy en día y provocaron la proliferación de terminales de computadora y estaciones de trabajo.

La evolución de los microprocesadores, la integración de los circuitos electrónicos y fabricación de herramientas para circuitos de muy larga escala de integración hizo posible el desarrollo de componentes económicamente viables para los requerimientos de los sistemas de procesamiento digital de imágenes. La evolución de las computadoras personales, y los productos asociados comerciales son ahora el soporte del procesamiento de imágenes en computadoras personales.

La necesidad de almacenamiento compacto para imágenes digitales son el comienzo del desarrollo de sistemas de almacenamiento óptico de bases de disco. Estos sistemas proveen almacenamientos económicos de grandes archivos de datos digitales de imágenes. El uso de discos de video comerciales para el almacenamiento y el despliegue de imágenes de baja resolución propician un barato y compacto mecanismo de almacenamiento para imágenes. El desarrollo de la fibra óptica para redes de área local podrían proporcionar la banda ancha alta requerida para el desarrollo de grandes distribuciones de procesos de imágenes y sistemas de almacenamiento de archivos que podrían soportar muchos usos simultáneamente.

La evolución tecnológica, habilitada por los desarrollos computacionales ha sido acelerada, y el porcentaje con el cual nuevos productos comerciales son integrados en el mercado se

incrementa. Sistemas compactos, altamente capaces y económicos para el procesamiento de imágenes digitales por computadora, será continuo en futuro.

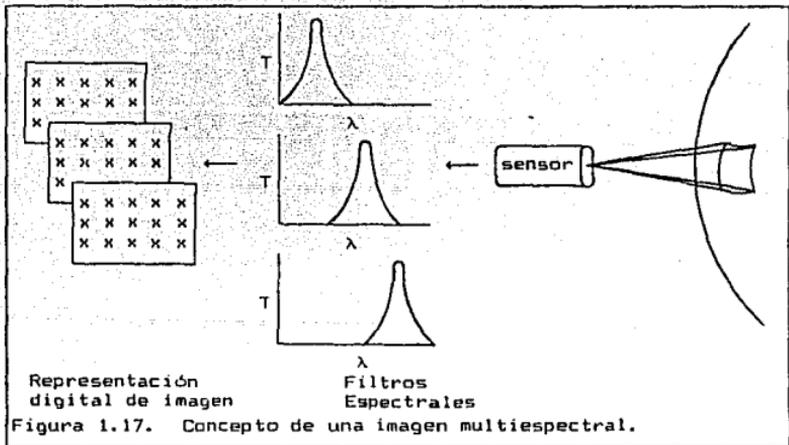
1.7 IMAGENES MULTIESPECTRALES

Las técnicas de procesamiento digital de imágenes no son inherentemente limitadas para las imágenes "blanco y negro" o "color sencillo". La figura 1.17 muestra un concepto de un sensor que adquiere una imagen digital de la misma escena en diferentes partes del espectro electromagnético. La misma entrada de la imagen es pasada a través de tres filtros espectrales, dentro del cual se separa la imagen en tres componentes espectrales. Cada componente es entonces digitalizado, de modo que uno de los tres componentes de la imagen en color de la escena que representa, es producido. En este ejemplo, los tres filtros podrían corresponder a el rojo, verde, y azul en respuesta a las regiones de la película de color convencional, y las tres imágenes digitales podrían ser representaciones de los componentes espectrales rojo, verde y azul de la escena original.

Una variedad de técnicas son usadas para el proceso, manipulación y despliegue de imágenes multiespectrales. El concepto de imagen multiespectral es bastante general, las técnicas de procesamiento multiespectral de imagen pueden involucrar la manipulación de la imagen conteniendo más de tres componentes espectrales.

La figura 1.18 ilustra el concepto más general de una imagen multiespectral, en el cual un gran número de componentes de imagen son representadas. Sobre el lado izquierdo de la figura 1.18, una imagen multiespectral muestra 11 componentes espectrales separados. Varios de los componentes de la imagen multiespectral no son derivaciones originales de los sensores de imagen. Sobre el

lado derecho de la figura 1.18 incluyen tres componentes de imagen (rojo, verde y azul) más otros componentes de imagen que son representados por valores de población, elevación, y calidad del aire. En este caso, el formato de la imagen digital ha sido usada para la representación del dato que no es adquirido originalmente por un sensor de imagen.



El procesamiento digital de imagen provee el establecimiento de técnicas para combinación, unión, análisis, y despliegue múltiple de desiguales tipos de datos.

1.8 IMAGENES ESPACIALES (FOTOGRAFIA ESPACIAL)

Las técnicas de procesamiento de imágenes son usadas en el soporte de varias disciplinas científicas involucradas en la exploración del planeta:

Geología. Involucra el estudio de la estructura de los planetas y sus satélites, incluyendo la composición, dinámica y estructura de su superficie.

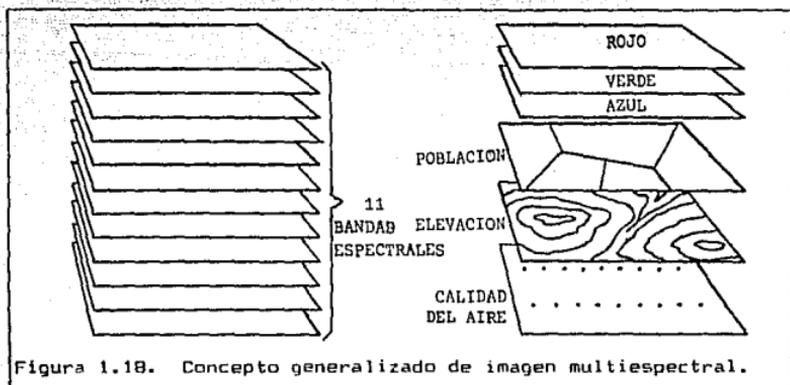
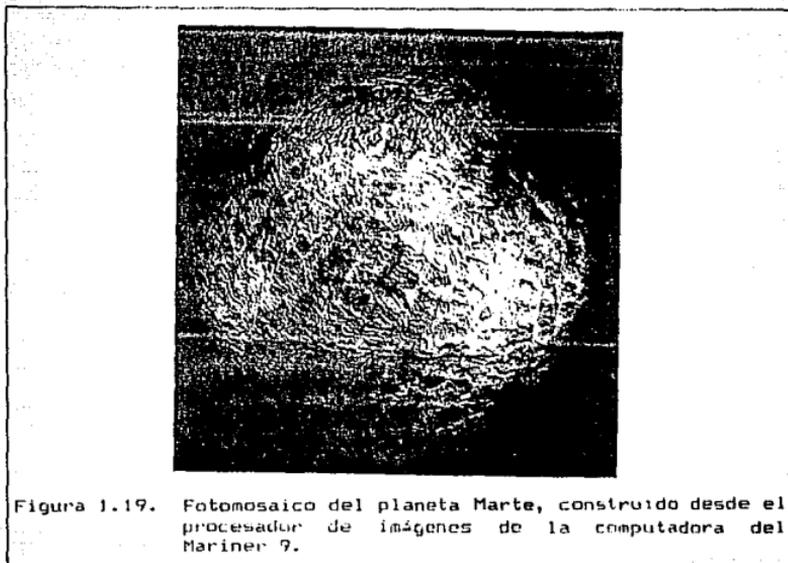


Figura 1.18. Concepto generalizado de imagen multiespectral.

Cartografía. Las técnicas de imágenes digitales han sido usadas como una ayuda en el mapeo de los planetas y sus satélites. Las técnicas incluyen transformaciones geométricas (para proyección de imágenes digitales dentro de estándares cartográficos de proyección), filtrado digital, mosaico digital de imágenes, y la determinación de la elevación desde fotografía estereoscópica de imágenes. La figura 1.19 muestra el primer fotomosaico de un planeta sensed remotamente.

Ciencias atmosféricas. El procesamiento de imágenes es una herramienta usada para determinar la composición atmosférica y las propiedades radiométricas y el análisis dinámico del movimiento atmosférico, usando la secuencia de tiempo de imágenes. La figura 1.20 es una muestra de el uso de la secuencia de tiempo de imágenes en un análisis de la dinámica atmosférica del planeta Urano.



Exploración mineral. Normalmente involucra el uso de imágenes multiespectrales para el análisis de la composición de la superficie esto podría indicar la deposición de minerales y análisis de la estructura de la superficie.

Determinación de la composición de la superficie. La composición de la superficie puede ser estimada desde el conocimiento de las propiedades de reflectancia de varios tipos de minerales. Las imágenes multiespectrales proveen un mecanismo para la determinación de la composición de la superficie.

Monitoreo Océanico. Las imágenes multiespectrales pueden ser usadas para el monitoreo de la calidad del agua del océano, especialmente líneas costeras cercanas. Los dispositivos de

Imágenes de alta resolución pueden ser usados para determinar el modelo de ondeo y estructuras de hielo.

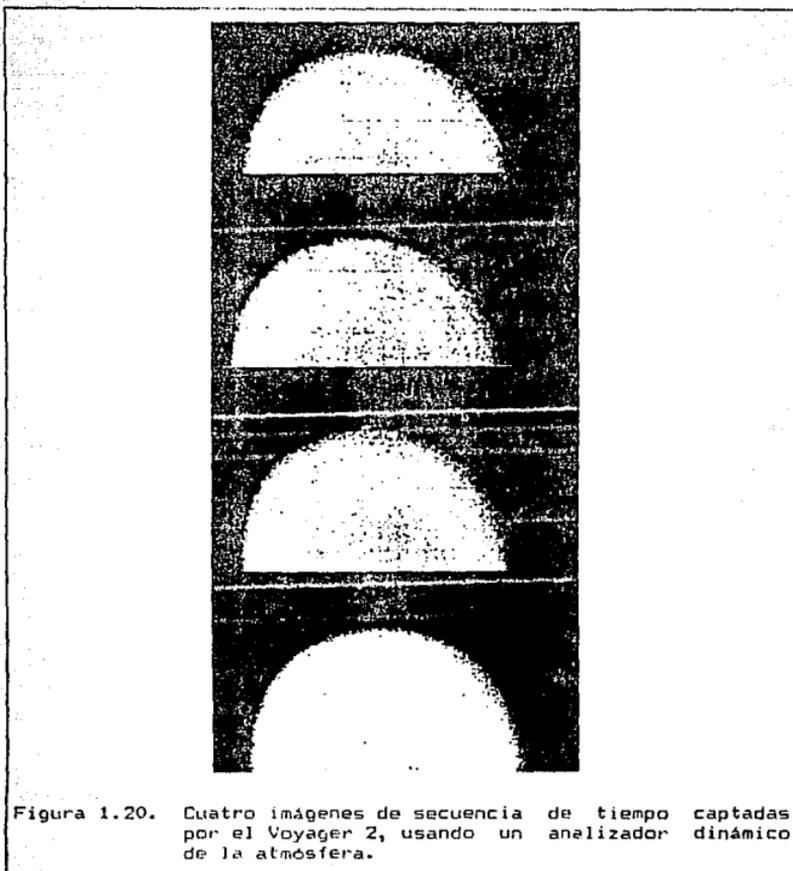


Figura 1.20. Cuatro imágenes de secuencia de tiempo captadas por el Voyager 2, usando un analizador dinámico de la atmósfera.

**NO
EXISTE
PAGINA**

CAPITULO 2

FUNDAMENTOS DEL PROCESAMIENTO DIGITAL DE IMAGENES.

2.1 MUESTREO.

2.2 CUANTIZACION.

- 2.2.1 CUANTIZACION DE UNA IMAGEN.
- 2.2.2 CUANTIZACION ESCALAR.
- 2.2.3 VECTOR DE CUANTIZACION.
- 2.2.4 PROCESO DE CUANTIZACION DE VARIABLES.
- 2.2.5 CUANTIZACION Y ASIGNACION DE PALABRA BINARIA (PCM).
- 2.2.6 CUANTIZACION DE UNA IMAGEN MONOCROMATICA.
- 2.2.7 CUANTIZACION DE UNA IMAGEN EN COLOR.

2.3 CODIFICACION.

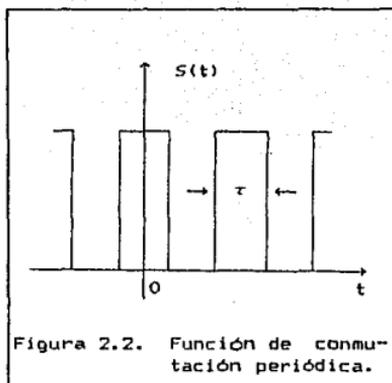
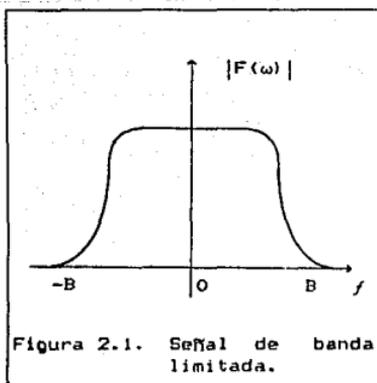
- 2.3.1 CODIFICACION DE UNA IMAGEN.
- 2.3.2 MODULACION POR CODIFICACION DE PULSOS (PCM).
 - 2.3.2.1 CODIFICACION PCM DE IMAGENES.
 - 2.3.2.2 CODIFICACION DE PIXEL.
 - 2.3.2.3 MODULACION DIFERENCIAL POR CODIFICACION DE PULSOS (DPCM).
- 2.3.4 MODULACION DELTA.
- 2.3.5 TECNICAS DE CODIFICACION DE IMAGEN.

2.1 MUESTREO.

En un sistema de procesamiento digital de imágenes, una parte importante es el muestreo de imágenes, el cual consiste en tomar puntos representativos de la imagen, formar un arreglo de números para posteriormente procesarlos, de aquí se obtiene otro arreglo de números los cuales son usados para la reconstrucción de la imagen.

El proceso de muestreo es fundamental no sólo para preparar las señales para ser transmitidas, sino que también es importante para el análisis de datos y en el procesamiento por computadora.

Una señal muestreada $f_s(t)$ contiene toda la información de una señal $f(t)$ y de $f_s(t)$ se puede obtener $f(t)$. Considerando una señal $f(t)$ limitada en banda a B Hertz, con una transformada de Fourier como la que se muestra en la figura 2.1, puede demostrarse que la información no se destruye en el proceso de muestreo. Utilizando una señal de conmutación o de retención $S(t)$ (figura 2.2), la cual tiene un ancho de pulso τ que para fines prácticos es muy pequeño.



El proceso de muestreo consiste en la multiplicación de las dos señales en el tiempo, es decir:

$$f_s(t) = f(t) S(t) \quad (2.1)$$

lo que nos da como resultado la señal mostrada en la figura 2.3. Si se toman muestras muy espaciadas, ocurre el fenómeno de traslape de los espectros, por lo que se deduce que la frecuencia de muestreo es igual a³:

$$f_c(t) \geq 2 \cdot AB \quad (2.2)$$

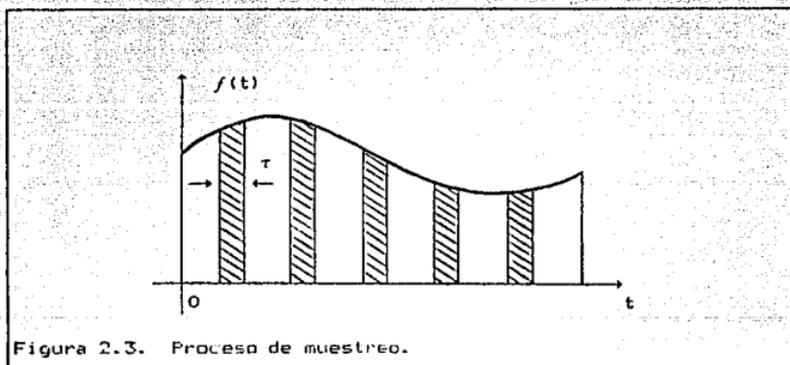


Figura 2.3. Proceso de muestreo.

En el análisis espectral se observa que las componentes armónicas se alejan o se aproximan unas de otras dependiendo de la frecuencia de muestreo f_c (figura 2.4).

³ Para más detalle vease, Capítulo 3, Mischa Schwartz, Transmisión de Información Modulación y Ruido, Mc Graw-Hill.

Con lo anterior se puede comprobar que el proceso de muestreo no destruye la información de $f(t)$ ⁴.

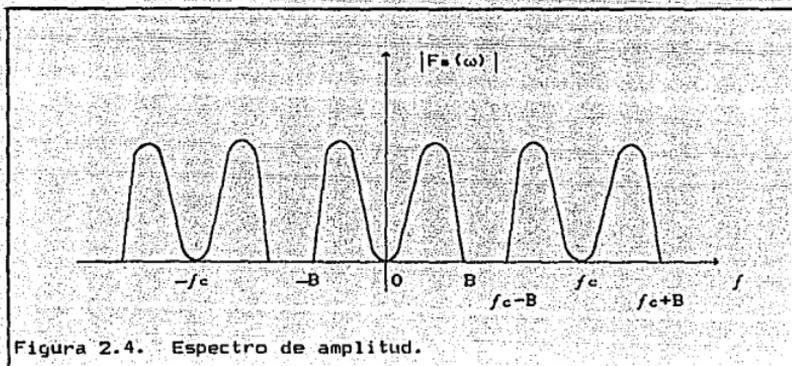


Figura 2.4. Espectro de amplitud.

En el caso de muestreo de imágenes el procedimiento es similar. El término imagen se refiere a una función de dos dimensiones luz-intensidad, denotada por $f(x, y)$ (como se mencionó en el capítulo I), en donde el valor o amplitud de f en coordenadas espaciales (x, y) da la intensidad (brillo) de la imagen en un punto.

Para una forma conveniente en el procesamiento por computadora, una función de imagen $f(x, y)$ debe ser digitalizada en espaciamiento y en amplitud. La digitalización de las coordenadas espaciales (x, y) serán referidas como muestreo de imagen, mientras que la amplitud de digitación será llamada

⁴ Para más detalles del proceso de muestreo vease, Capítulo 3, Mischa Schwartz, Transmisión de Información Modulación y Ruido, Mc Graw-Hill.

cuantización del nivel de gris.

Suponemos a $f(x, y)$ como una imagen continua, dentro de un arreglo de muestras de la forma $N \times N$ como se observa en la ecuación 2.3, donde cada elemento del arreglo es una cuantización discreta:

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,N-1) \end{bmatrix} \quad (2.3)$$

El lado derecho de la ecuación representa lo que comunmente es llamado imagen digital, donde cada elemento del arreglo es referido como pixel.

Los procesos de digitalización requieren de tomar una decisión para determinar un valor de N y también sobre el número de niveles de gris discretos permitidos para cada pixel. Esto es en la práctica común del procesamiento digital de imagen para permitir que estas cantidades sean entero de potencia de base dos es decir,

$$N = 2^n \quad (2.4)$$

y

$$G = 2^m \quad (2.5)$$

donde G denota el número del nivel de gris. Esos niveles discretos son igualmente espaciados entre 0 y L dentro de la escala. Usando la ecuación 2.4 y 2.5 el número, b , de bits requeridos para almacenar una imagen esta dado por

$$b = N \times N \times m \quad (2.6)$$

Por ejemplo, una imagen de 128 x 128 con 64 niveles de gris requiere de 98,304 bits de almacenamiento. La Tabla 2.1 resume los valores de b para rangos típicos de N y m . La Tabla 2.2 da los números correspondientes de 8 bytes. Generalmente, esto no es práctico desde el punto de vista de la programación para llenar un byte completamente, si ésta emplea el traslapamiento de un pixel desde un byte a el siguiente. Así en la Tabla 2.2 representa el número mínimo de bytes necesitados para cada valor de N y m donde no es permitido el encimamiento. Por ejemplo, si $m = 5$, se asume que sólo un pixel es almacenado en un byte, se pierden tres bits que no fueron usados del byte.

$N \backslash m$	1	2	3	4	5	6
32	1,024	2,048	3,072	4,096	5,120	6,144
64	4,096	8,192	12,288	16,384	20,480	24,576
128	16,384	32,768	49,152	65,536	81,920	98,304
256	65,536	131,072	196,608	262,144	327,680	393,216
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864

Tabla 2.1. Número de bit de almacenamiento para varios valores de N y m .

Puesto que la ecuación 2.3 es una aproximación a una imagen continua, una pregunta razonable a responder que este punto es cuántas muestras y niveles de gris son necesarios para una buena aproximación. La resolución (ejemplo, el grado de detalle discernible) de una imagen es dependiente de N y m . Entre más se incrementa el parámetro, el arreglo digitalizador será más próximo a la imagen original. Sin embargo, la ecuación 2.6 claramente señala el hecho infortunado que almacena y consecuentemente, los requerimientos de procesamiento se incrementan rápidamente como

una función de N y m .

$N \backslash m$	1	2	3	4	5	6
32	128	256	512	512	1,024	1,024
64	512	1,024	2,048	2,048	4,096	4,096
128	2,048	4,096	8,192	8,192	16,384	16,384
256	8,192	16,384	32,768	32,768	65,536	65,536
512	32,768	65,536	131,072	131,072	262,144	262,144

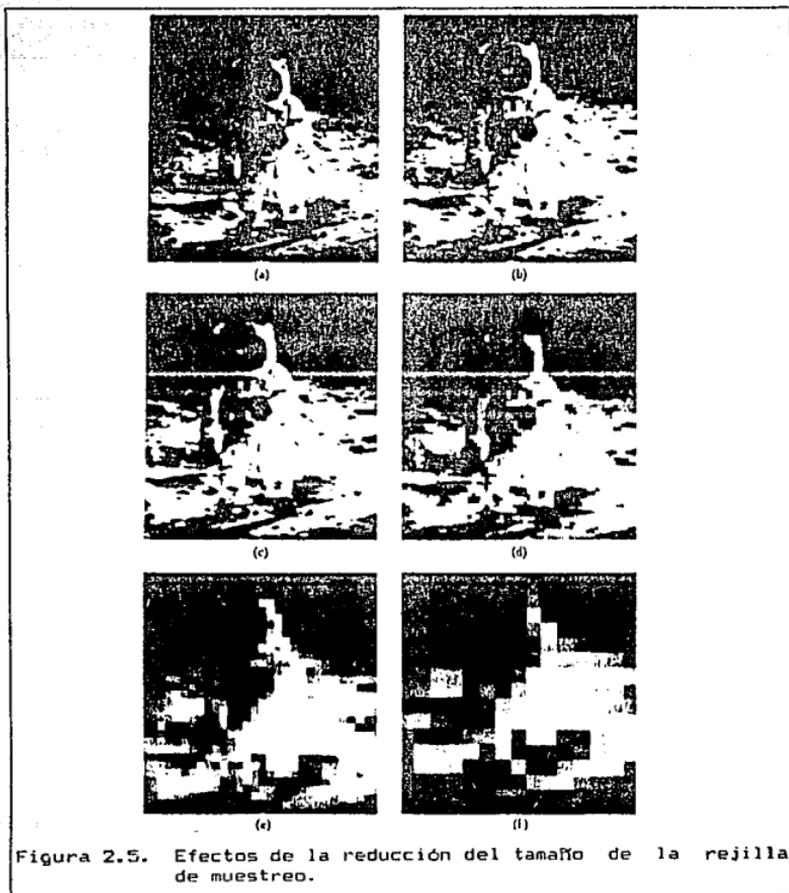
Tabla 2.2. Número de B bytes de almacenamiento para varios valores de N y m .

En vista del comentario anterior, se debe considerar el efecto de las variaciones que se tiene en N y m dentro de la calidad de la imagen. Como es de sospecharse, una "buena" imagen, es difícil de definir porque la calidad requiere de variaciones acordes a la aplicación.

La figura 2.5 muestra el efecto de reducir el tamaño de la rejilla de muestreo en una imagen. La figura 2.5a es de 512×512 , 256 niveles, en la imagen se muestra un astronauta durante el primer aterrizaje en la luna. De la figura 2.5b hasta la 2.5f se muestra la misma imagen, pero con $N = 256, 128, 64, 32$ y 16 . En todos los casos el máximo número de niveles de gris permitidos es de 256.

El área usada para cada imagen será la misma (512×512 puntos desplegados), los píxeles en baja resolución de imagen se duplica en orden para ocupar el campo entero de despliegue. Este produce un efecto de chequeo, el cual es particularmente importante en la baja resolución de imagen. Esto se nota en la

imagen de 256×256 que es más cerrada que la de la figura 2.5a pero la calidad de la imagen se deteriora rápidamente por los otros valores de N .



2.2 CUANTIZACION.

2.2.1 Cuantización de una imagen.

Cualquier cantidad analógica al ser procesada por una computadora digital o sistema digital debe ser representada como un número entero proporcional a esa amplitud. Los procesos de conversión entre muestras analógicas y muestras de valores discretos es llamada cuantización.

2.2.2 Cuantización escalar.

En la figura 2.6 se ilustra un ejemplo típico de cuantización de una señal escalar. En el proceso de cuantización la amplitud de una señal analógica es la comparación de un conjunto de niveles de decisión. Si la amplitud muestreada cae entre dos niveles de decisión, se cuantiza en un nivel de reconstrucción fijo que este dentro de una banda de cuantización. En un sistema digital a cada muestra cuantizada se le asigna un código binario. Un código binario de igual longitud es indicado en el ejemplo.

Para el desarrollo de la señal cuantitativa escalar, las técnicas de cuantización permiten a f y f' representar una amplitud real, el muestreo de la señal escalar y el valor cuantizado, respectivamente. Asumiendo esto, f es una muestra de un proceso no definido con la conocida densidad de probabilidad $p(f)$. Además, f es forzado a caer dentro del rango

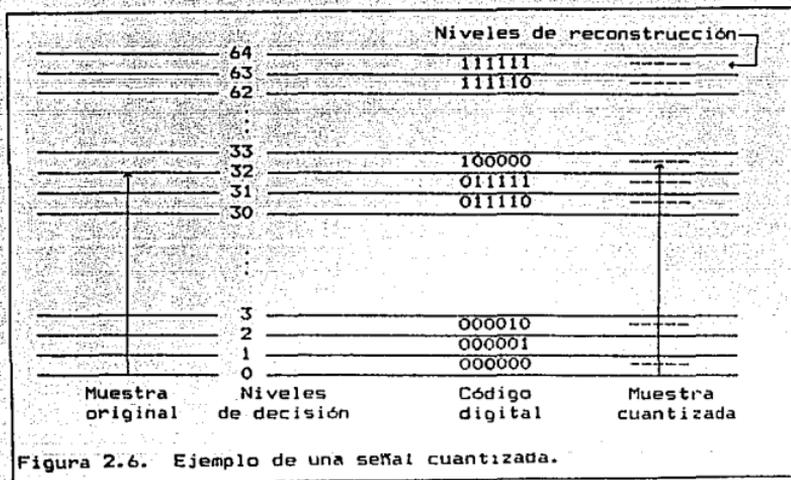
$$a_L \leq f \leq a_U \quad (2.7)$$

donde a_U y a_L representan los límites superiores e inferiores.

El problema de la cuantización se funda en la especificación de un conjunto de niveles de decisión d_j y de un conjunto de niveles de reconstrucción r_j tal que si

$$d_j \leq f \leq d_{j+1} \quad (2.8)$$

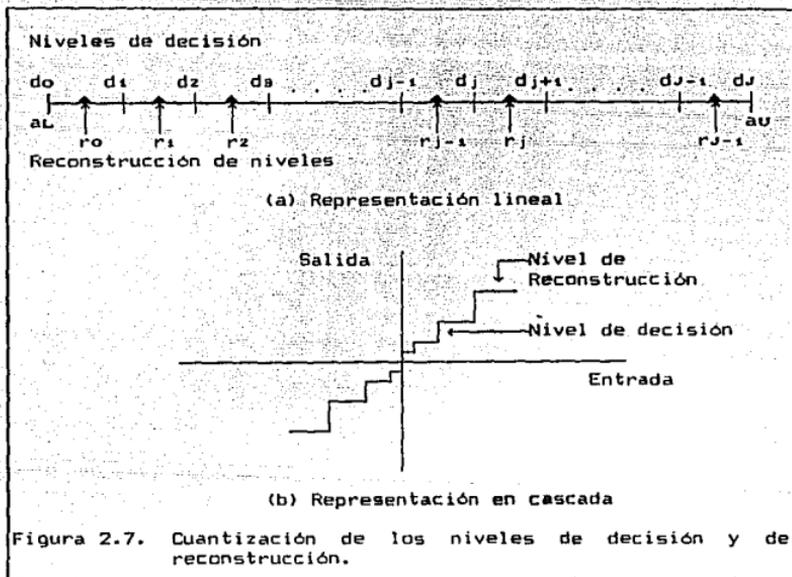
entonces la muestra es cuantizada a un valor de reconstrucción r_j .



La figura 2.7a ilustra la colocación de los niveles de decisión y reconstrucción a lo largo de una línea para J niveles de cuantización. La representación en forma de escalera de la figura 2.7b es otra forma común de representación.

Los niveles de decisión y reconstrucción son elegidos para minimizar algún error de cuantización medido entre f y f' . La cantidad usada en el error de cuantización es el error medio cuadrático porque es una cantidad tratable, y razonablemente correlacional y con un criterio subjetivo. Para J niveles de cuantización, el error medio cuadrático es

$$\epsilon = E\{(f - f')^2\} = \int_{a_L}^{a_U} (f - f')^2 p(f) df = \sum_{j=0}^{J-1} \int_{d_j}^{d_{j+1}} (f - r_j)^2 p(f) df \tag{2.9}$$



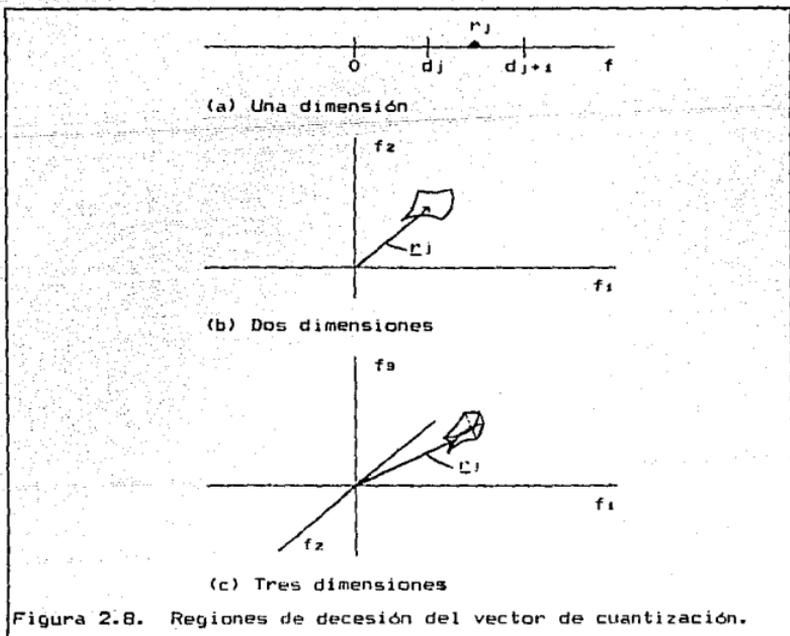
2.2.3 Vector de cuantización.

La cuantización de una secuencia de muestras de amplitud continua es normalmente ejercida sobre una base secuencial. Cada miembro de la secuencia es tratado como una variable escalar y cuantizada separadamente. Esto es, para reducir el error de cuantización por acoplamiento cuantizando y para reconstruir los elementos de la secuencia.

Considerando los $N \times 1$ elementos del vector señal f , el cual asume una muestra de un vector no definido con una densidad de probabilidad de n -ésimo orden

$$p(f) = p(f_1, f_2, \dots, f_N) \quad (2.10)$$

El vector de cuantización de f implica la subdivisión de las N -dimensiones del espacio del vector dentro de las J regiones de decisión cada D_j engloba uno de los J valores de reconstrucción. El vector señal f es cuantizado para la reconstrucción del vector r_j si f cae en la región de decisión D_j . En la figura 2.8 se muestra el vector de cuantización para una, dos, y tres dimensiones. En la formulación general del vector de cuantización, el vector f es mapeado dentro del vector r_j ; los elementos individuales de f no son necesariamente cuantizados individualmente sobre un conjunto de niveles de decisión.



Para el vector de cuantización del error medio cuadrático de cuantización puede ser escrito como

$$\epsilon_c = \sum_{j=0}^{J-1} \int_{D_j} \text{tr}[(f - r_j)(f - r_j)^T] p(f) df \quad (2.11)$$

2.2.4 Proceso de cuantización de variables.

Los números dentro de una computadora digital representan variables de imagen, tal como luminancia o valores de los tres colores primarios normalmente son entrada según los códigos enteros correspondientes a la cuantización de los niveles de reconstrucción de las variables. Por ejemplo, la escala del negro al blanco para luminancia en una imagen monocromática es usualmente una escala lineal como un entero entre 0 y 255. Sin embargo, los códigos enteros no deben tratarse como variables aritméticas; es decir, el código entero debe ser convertido a un número de niveles de reconstrucción decimal real antes de las operaciones aritméticas. De no ser así, se puede dar un error en el procesamiento.

Actualmente, si los códigos enteros donde la no monotonía a lo largo de la escala de gris, no se considera su uso entonces como variables de proceso. La consecuencias del procesamiento aritmético de códigos enteros monótonos, es decir, los números decimal de la reconstrucción de niveles son ahora considerados.

En una computadora digital hay dos formas de representación numérica: entera y real. El rango de números enteros en magnitud es desde 0 hasta algún valor máximo. En una minicomputadora de 16 bits, por ejemplo, el máximo entero positivo es 32,768 (2^{15}). Si una operación aritmética entera resulta en una parte fraccionaria, lo restante es truncado. Así, por ejemplo, la relación 8/3 es representada como el entero 2 quedando fuera lo sobrante al punto

decimal. Con el número real en la computadora la parte fraccionaria de la operación es retenida fuera de la exactitud de la computadora. La relación de los números reales $8./3.$ es representada como $2.66...66.$

La figura 2.9 ofrece una comparación de tres señales estratégicas procesadas. En la figura 2.9a una señal continua escalar f en el rango $al \leq f \leq au$ es sujeta a un punto de transformación $\theta_p(\cdot)$, el cual se somete a la salida de la señal continua variable

$$g = \theta_p(f) \quad (2.12)$$

La figura 2.9b describe un sistema de procesamiento en el cual la variable escalar f es uniformemente cuantizada y codificada antes del procesamiento. El valor entero del código esta dado por

$$j = \left[(J - 1) \left(\frac{f - al}{au - al} \right) \right]_N \quad (2.13)$$

donde $[\cdot]_N$ denota la cercanía del valor entero del argumento. El código entero j es reconstruido a el número real r_j acorde a la relación

$$r_j = \left(\frac{r_{j-1} - r_0}{J - 1} \right) j + r_0 \quad (2.14)$$

Después, el punto de operación de procesamiento es ejecutado sobre la señal de reconstrucción r_j para producir la señal cuantizada de salida

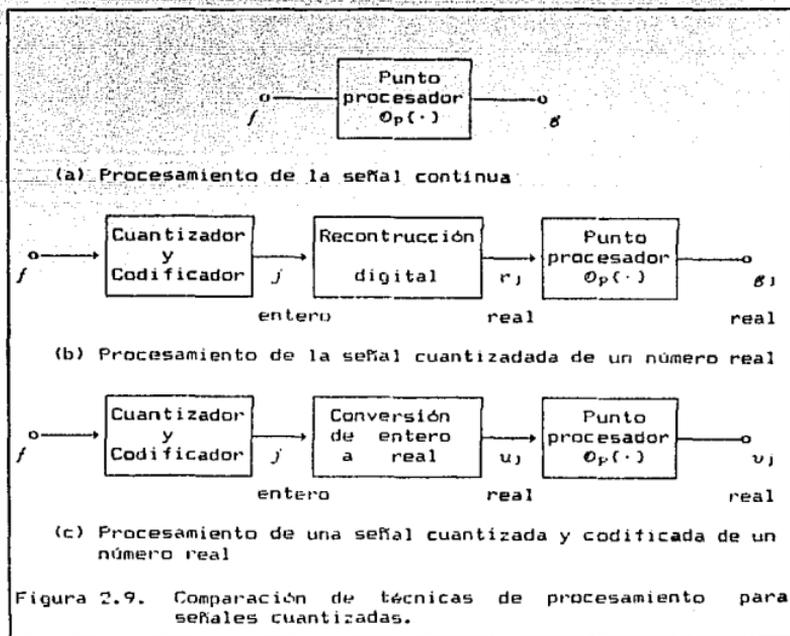
$$g_j = \theta_p(r_j) \quad (2.15)$$

el cual es equivalente a la señal de continua de salida g de la ecuación 2.12 excepto para el error de cuantización introducido por la ecuación 2.13.

La figura 2.9b representa la forma deseada de procesamiento en la computadora de las variables cuantizadas. Por desgracia, con mucha frecuencia, las variables cuantizadas de la señal de procesamiento es erróneamente ejecutada según el procedimiento descrito en la figura 2.7c. El código entero j es convertido a un número real u_j por encima del rango $0., 1., 2., \dots, J - 1.$ Entonces el número real de la señal de salida v_j es computarizado de acuerdo con la relación

$$v = \mathcal{O}_p(u_j) \quad (2.16)$$

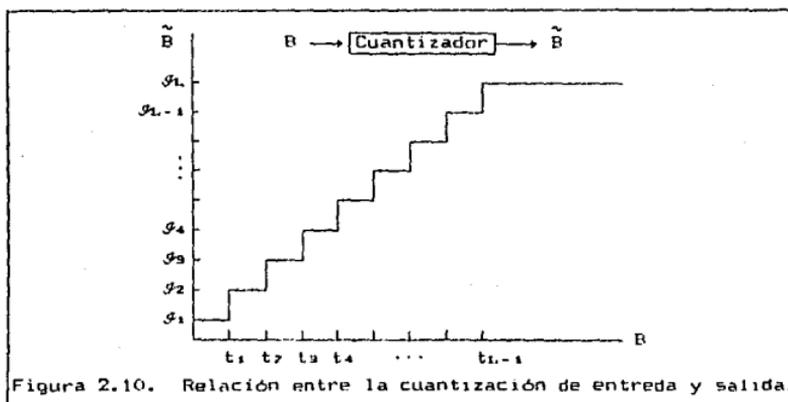
En general, u_j contiene una parte fraccional de la totalidad del número.



2.2.5 Cuantización y asignación de una palabra binaria - PCM.

Uno de los principales pasos para representar una amplitud proporcional finita de información visual es la amplitud del valor de la cuantización; esto es, representar un valor numérico cuyo rango sea una secuencia de números elegidos por una aproximación desde un conjunto de números finitos. Por ejemplo, los voltajes en un rango de 0 a 20 volts sólo pueden ser medidos los cercanos a 0.1 volt.

Se pueden elegir los valores del pixel a cuantizar, o bien se podría escoger el uso de los coeficientes cuantizados de Fourier en la especificación de una imagen. En cualquier caso, la representación resultante, ilustrada en la figura 2.10, se caracteriza por cantidades que pueden tomar solamente un número finito de valores o niveles, estos niveles serían valores continuos en la imagen original. La entrada toma sobre un rango de valores continuos, y la salida tiene solamente un número finito de niveles L , el cual es frecuente una base de dos, por ejemplo, 2^K para K bits.



Cada miembro del conjunto finito de posibles niveles es asignado a una palabra binaria única de una longitud finita, es decir, un código es asignado a cada valor muestra. Este código (como también el conjunto de niveles de aproximación) puede cambiar de tiempo en tiempo dependiendo de ciertas circunstancias. Por ejemplo, esto es ineficiente para cuantizar todos los códigos de los coeficientes de Fourier de una imagen en movimiento.

Teniendo construido un código semejante, el proceso para producir una especificación de un bit proporcional finito de una imagen es simplemente para cuantizar (en el dominio del tiempo o de Fourier) los valores muestreados de la imagen y representar cada nivel cuantizado por un código de palabra asociado. Desde que la muestra ocurre en un porcentaje finito y el código de palabras son de un bit de longitud finita; la representación resultante es de un bit proporcional finito. Codificando las muestras de intensidad de la imagen de esta manera, se denomina modulación por pulsos codificados y es abreviada como PCM.

Las consecuencias de la cuantización es la pérdida de información dado que los valores son específicos con menos exactitud en la imagen original. Estas diferencias de la señal original y la señal cuantizada es llamada error de cuantización.

Reemplazando los píxeles originales por sus aproximaciones cuantizadas, como en la figura 2.10, causa error en la reconstrucción del duplicado de la imagen \tilde{B} , en un buen sistema PCM, parece a la vista como un ruido casual o nieve. En algunas aplicaciones donde se presenta esta la fotografía o el despliegue en el tubo de rayos catódicos (TRC), este ruido puede ser prácticamente invisible con 8 bits cuantizados, por ejemplo, con cuantización de 256 niveles representativos.

Visiblemente el ruido de cuantización PCM es también altamente dependiente sobre el ruido original anterior a la

cuantización de la imagen original. Para algunas aplicaciones sólo se requiere de suficientes bits de cuantización para asegurar un ruido de cuantización pequeño en comparación al ruido de muestreo original. Siete bits cuantizados (o aún seis), son suficientes para algunas aplicaciones donde la imagen original aparece ruidosa.

2.2.6 Cuantización de una imagen monocromática.

En la transmisión básica de imágenes en la modulación por pulsos codificados (PCM), cada imagen muestreada es cuantizada, usualmente por encima de una escala lineal, y asigna un grupo de códigos binarios para la transmisión. Normalmente, los códigos de longitud uniforme son usados para cada nivel de brillantez, y por esto el número de niveles de brillantez L satisface la relación

$$L = 2^b \quad (2.17)$$

donde b representa el número de bits admitidos por imagen muestreada.

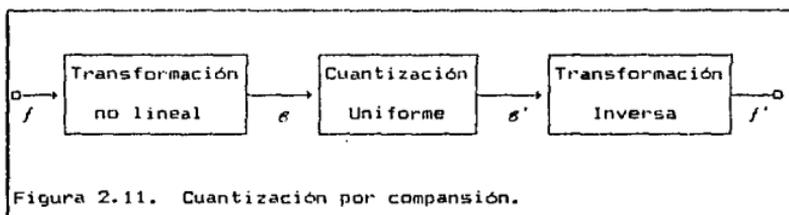
Una relación de compresión binaria puede ser obtenida por PCM. Si la calidad de imagen del sistema de transmisión es juzgado por una relación analítica, entonces b es simplemente tomado como el mínimo valor que satisfaga la medida de la calidad aceptable de imagen cuantizada. Para fijar un valor subjetivo, b fué disminuido hasta que los efectos de cuantización fueron inaceptables.

El ojo humano es capaz de juzgar en brillo absoluto de 10 a 15 sombras de gris, pero es mucho más sensitivo a la diferencia en los brillos de contornos de gris adyacentes. Para una reducción del número de niveles cuantizados, el primer efecto notable es el contorno de la escala de gris causado por un salto en la reconstrucción de la brillantez de la imagen entre los niveles de cuantización en la región donde la imagen original es cambiada

ligeramente en brillo. El número mínimo de bits cuantizados requeridos para la codificación básica PCM hasta evitar el contorno de la escala de gris depende de una variedad de factores incluyendo linealidad de la imagen desplegada y los efectos del ruido antes del sensor de imagen.

Asumiendo que el sensor de imagen produce un pixel de salida muestreado proporcional a la intensidad de brillantez de la imagen, una pregunta interesante sería: ¿Podría la intensidad de la imagen en si misma, o alguna función de la intensidad de la imagen, ser cuantizada? Además, ¿podría la escala cuantizada ser lineal o no lineal? La linealidad o no linealidad de la escala de cuantización pueden ser vista como una materia de implementación.

Una escala de cuantización no lineal dada puede ser realizada por la operación de compansión⁵ (compresión-expansión) de la figura 2.11 en la cual el peso de la amplificación no lineal de la señal continua es cuantizada y ejecutada, seguida por cuantizaciones lineales y por la amplitud inversa del nivel de cuantización. La consideración es limitada aqui hasta la cuantización lineal de compansión de los pixeles muestreados.



⁵ En el Capítulo 3 de esta tesis, se va describir la operación de compansión mediante un comparador.

Se tienen muchos estudios experimentales para determinar el número y empleo de los niveles de cuantización requeridos para minimizar el efecto de contornos en la escala de gris. Goodall realizó algunos de los primeros experimentos sobre televisión digital y concluyó que 6 bits de intensidad de cuantización (64 niveles) son requeridos para una buena calidad y 5 bits (32 niveles) podrían ser suficientes para una proporción de contorno.

Otras investigaciones han alcanzado conclusiones similares. En más estudios, sin embargo, se han tenido algunas preguntas como los lineamientos y la calibración de los sistemas de imagen. Muchas cámaras de televisión y monitores manifiestan una respuesta no lineal a la intensidad de la luz. También, una película fotográfica es usada frecuentemente para registrar los resultados de los experimentos siendo altamente no lineal. Finalmente, cualquier cámara o monitor tiende a disminuir los efectos del ruido en el contorno.

Las figuras 2.12 y 2.13 contienen fotografías de imágenes cuantizadas de una persona con una variación de niveles de cuantización.

En la figura 2.12 la señal de luminancia de la imagen ha sido cuantizada uniformemente desde dos hasta 64 niveles (1 bit y 6 bits). El contorno de la escala de gris en esta imagen es aparente en regiones negras para cinco o menos bits. La figura 2.13 presenta los resultados de una simulación por computadora de cuantización uniforme de la densidad de la imagen, en el cual los 8 bits, 256 niveles de cuantización la imagen fué linealmente reescalada hasta el rango $0 \leq F(j,k) \leq 1.0$ y logarítmicamente modificada de acuerdo a la relación

$$G(j,k) = \frac{\ln [1 + F(j,k)]}{\ln[2]} \quad (2.18)$$

La escala logarítmica de la imagen $G(j,k)$ fué entonces cuantizada

uniformemente con un número variable de niveles hasta producir la

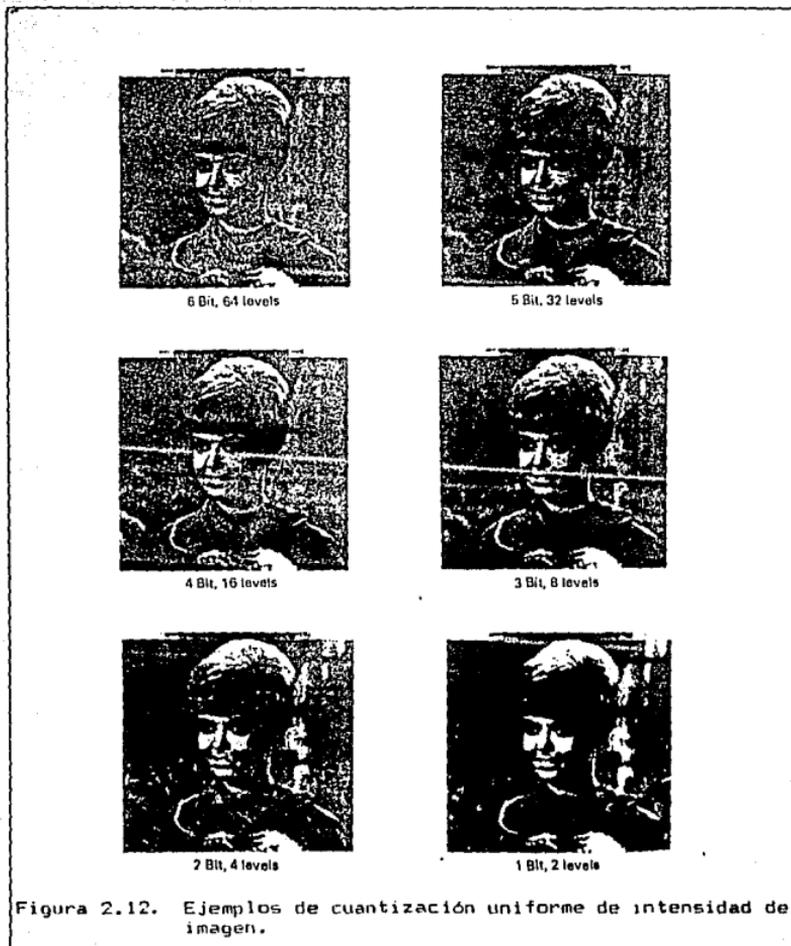
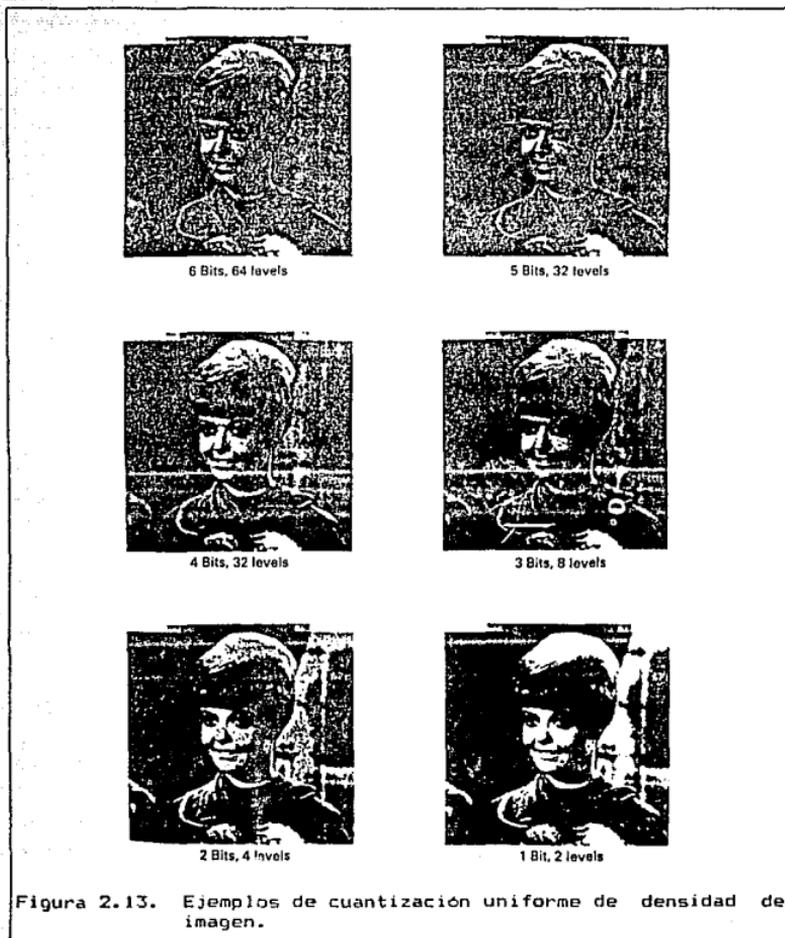


Figura 2.12. Ejemplos de cuantización uniforme de intensidad de imagen.



reconstrucción $G'(j,k)$. Después, la reconstrucción de luminancia

de imagen fué creada por la operación

$$F'(j,k) = \exp(G'(j,k) \ln(2)) - 1 \quad (2.19)$$

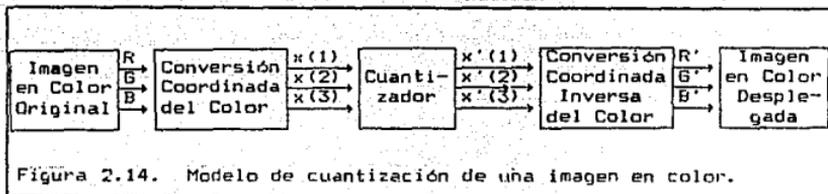
En el despliegue de fotografías de la figura 2.13 para la densidad uniforme de cuantización el contorno de la escala de gris aparece el efecto poco notable para el ejemplo correspondiente de la figura 2.12 para una luminancia uniforme de cuantización.

2.2.7 Cuantización de imágenes en color.

Una imagen en color puede ser representada por los valores de los colores primarios rojo, verde y azul o la función inversa de los valores de los colores primarios, lineal o no lineal. Si los colores primarios rojo, verde y azul son cuantizados individualmente, entonces la selección del número y empleo de los niveles de cuantización siguen las mismas consideraciones generales que para una imagen monocromática. El ojo humano presenta una respuesta no lineal al espectro de luz como lo es la luz blanca, y por esto es subjetivamente preferible en la comparación de valores de los tres colores antes de la cuantización. Sin embargo, como ya se mencionó anteriormente, el ojo humano es más sensitivo a los cambios de brillo; en la región azul del espectro, es moderadamente sensitivo a los cambios en la región espectral verde, y poco sensitivo a los cambios de rojo. Así, es posible la asignación de niveles de cuantización sobre bases más eficientes que usar simplemente un número igual para cada valor de los colores.

En la figura 2.14 se muestra un diagrama a bloques de un sistema de cuantización de imagen en color. Una imagen original descrita por los valores de los colores primarios R, G, B es convertida a tres componentes $x(1)$, $x(2)$, $x(3)$, los cuales son cuantizados. Después, los componentes cuantizados $x'(1)$, $x'(2)$, $x'(3)$ son convertidos nuevamente al color original del sistema

coordinado, produciendo los valores de los colores R' , G' , B' .



2.3 CODIFICACION.

2.3.1 Codificación de una imagen.

El diseño de sistemas de codificación generalmente ha sido la representación de imágenes con una fidelidad aceptable, además de la utilización de un número menor de bits de codificación tanto como sea posible. Si se reduce el número de bits de codificación se permite: (a) la transmisión de imágenes individuales de una forma más rápida; (b) más canales paralelos para ser transmitidos en un enlace de comunicación; (c) una reducción de las necesidades de transmisión; o (d) una mayor capacidad de almacenamiento compacto de imágenes.

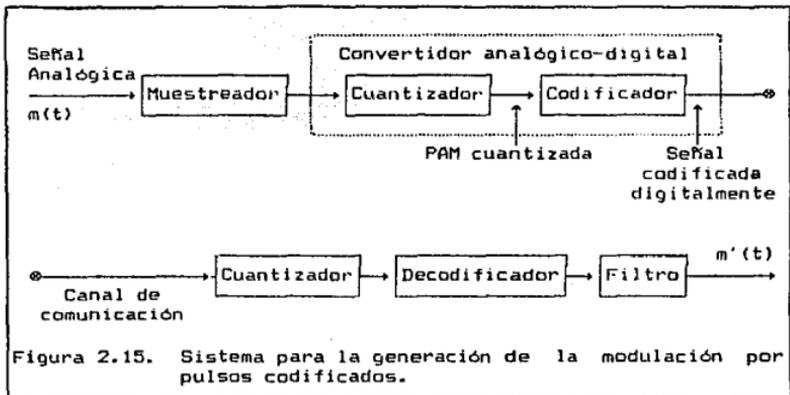
2.3.2 Modulación por codificación de pulsos (PCM).

Para realizar una transmisión de señales analógicas muestreadas, primeramente son digitalizadas y posteriormente codificadas. La combinación de las operaciones de muestreo y cuantización genera una forma de onda PAM cuantizada⁶, que es un

⁶ Para información sobre las ondas PAM vease, Capítulo 5, Taub Herbert, Principles Communication Systems, Mc Graw-Hill.

tren de pulsos cuyas amplitudes están registradas a un número discreto específico, posteriormente el valor de la amplitud es transmitido como pulsos. Los sistemas que trabajan con este tipo de señales son llamados sistemas de modulación por codificación de pulsos (PCM).

En la figura 2.15 se muestra el diagrama con los elementos de la generación de modulación por pulsos codificados PCM. La señal continua primero se filtra en un pasabajos y después se muestrea. Los valores muestra se redondean o cuantifican al valor discreto predeterminado más próximo o nivel cuántico. La señal muestreada y cuantificada resultante es discreta en tiempo (en virtud del muestreo) y en amplitud (en virtud de la cuantificación).



Un codificador convierte las muestras cuantificadas en palabras de código digital, con una palabra de código por cada muestra, y se genera la señal de banda base de modulación por pulsos codificados como una forma de onda digital.

El proceso de cuantización es la conversión de una señal analógica a una digital y consiste en una subdivisión de amplitudes de las señales en un número predeterminado de valores o niveles discretos de amplitud, las señales resultantes de este proceso se denominan cuantizadas. En este proceso existe una pérdida irreparable de información debido a que es imposible reconstruir la señal analógica original a partir de su versión cuantizada.

El proceso de cuantización se muestra en la figura 2.16. En la figura 2.16a el cuantizador tiene la forma característica de escalera, en 2.16b se observa que tiene varios niveles $m_0, m_1, m-1, \text{etc.}$, el tamaño del escalón es S , la señal cuantizada se muestra en 2.16c, $m_q(t)$, $m'(t)$ es la salida del cuantizador, y si el cuantizador es uniforme $v_0 = v_1$; entonces $m'(t) \approx m(t)$. La transición de niveles ocurre en el instante cuando $m'(t)$ cruza el punto medio entre 2 niveles adyacentes, la calidad de la aproximación puede mejorarse reduciendo el tamaño de los escalones S o incrementando el número de niveles permitidos.

Obviamente el proceso de cuantización introduce algunos errores durante la reproducción final de la señal, debido a que la señal demodulada final difiere un poco de la original, el efecto es como si se agregará un ruido adicional al sistema; este ruido es llamado ruido de cuantización.

2.3.2.1 Codificación PCM de imágenes.

La modulación por pulsos codificados (PCM) es la forma más simple de codificación de imagen. En este sistema la señal de la imagen es muestreada, y cada muestra es cuantizada y codificada en forma binaria para su transmisión. Para la transmisión binaria por facsimile (FAX), la señal de la imagen es cuantizada para sólo dos niveles, blanco y negro, y un código con 1 bit por muestra, cero o uno.

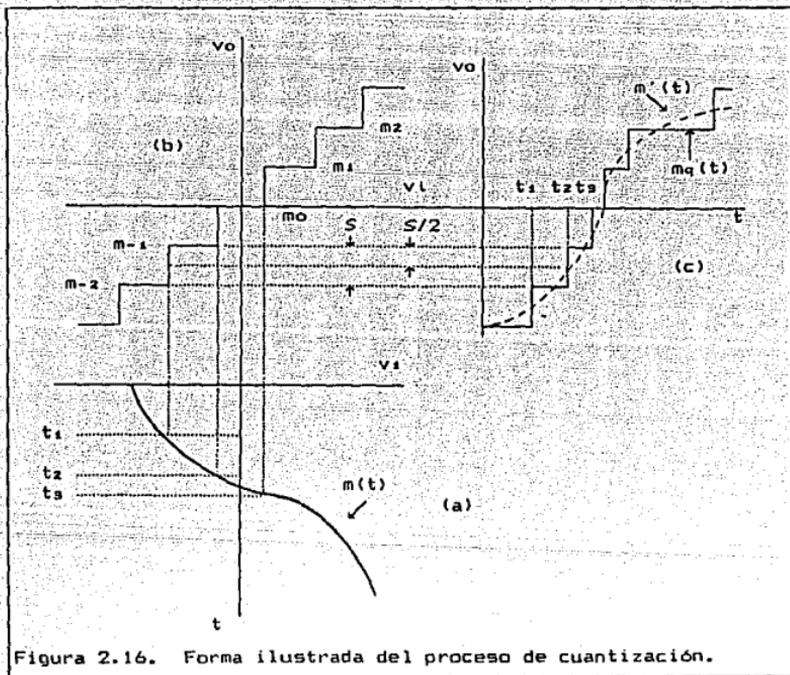


Figura 2.16. Forma ilustrada del proceso de cuantización.

En imágenes monocromáticas usualmente se cuantizan desde 64 a 256 niveles por muestra correspondiendo 6 u 8 bits por muestra de longitud fija para una palabra de código binario. Los sistemas de imágenes de color digital generalmente utilizan de 6 a 8 bits para cada valor de los tres colores rojo, verde y azul.

La codificación PCM se considera como un sistema de codificación de línea baja. La proporción de reducción de la transmisión son generalmente calculadas con respecto a la codificación.

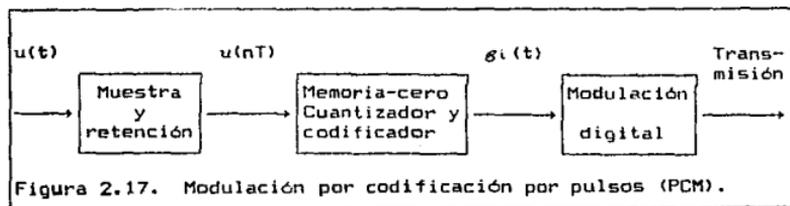
La reducción del número de niveles de cuantización para imágenes monocromáticas o en color codificadas en PCM nos lleva a un efecto llamado contorno, en el cual los saltos discretos entre los niveles de cuantización observados en las regiones de la imagen son lentamente cambiadas en luminancia o valores de los tres colores primarios.

El efecto de contorno puede ser reducido substancialmente por la adición de una pequeña cantidad de ruido aleatorio (pseudorandom) o de agitación determinístico para la señal de la imagen antes de la cuantización.

2.3.2.2 Codificación de pixel.

Para la codificación, cada pixel es procesado independientemente, ignorando las dependencias entre pixel.

En PCM las señales de video que llegan son muestreadas, cuantizadas, y codificadas por un código de palabra adecuado (primero se debe alimentar un modulador digital para la transmisión) (Figura 2.17). La salida del cuantizador es generalmente codificada por un definidor de la longitud de código de palabra que tiene B bits. (B bit es proporcional al promedio del dato original). Comúnmente, 8 bits son suficientes para la calidad de imagen en una transmisión monocromática, mientras que imágenes médicas o señales de video en color pueden requerir de 10 a 12 bits por pixel.



El número de bits cuantizados necesarios para un despliegue de imágenes puede ser reducido de 4 a 8 bits por píxel por el uso de la compansión, contrastando con la cuantización. Las técnicas Halftone⁷ reduce la cuantización de salida a 1 bit por píxel, pero usualmente la muestra de entrada proporcional debe ser incrementado por un factor de 2 a 16. La compresión obtenida por esta técnica es generalmente menor al 2:1.

En términos de una distorsión media cuadrada, la mínima proporción obtenida por PCM esta dada por la fórmula de proporción de distorsión:

$$R_{PCM} = \frac{1}{2} \log_2 \frac{\sigma_u^2}{\sigma_q^2}, \quad \sigma_u^2 < \sigma_q^2 \quad (2.20)$$

donde, σ_u^2 es la varianza⁸ de la entrada del cuantizador y σ_q^2 es la distorsión media cuadrada del cuantizador.

2.3.3.3 Modulación diferencial por codificación de pulsos DPCM

La modulación DPCM es un código predictivo esquemático el cual emplea la correlación entre muestras cercanas de la señal de entrada para reducir la redundancia estadística y la baja transmisión, en PCM la cuantización y la codificación es instantaneamente. En DPCM una estimación de la aproximación de la muestra se basa en la muestra anterior. Esta estimación es abstraída de la muestra actual. La diferencia de estas señales es el error de predicción el cual es cuantizado, codificado y

⁷ En el Capítulo 3 de esta tesis, se va a dar una descripción más amplia de estas técnicas.

⁸ Vease, Capítulo 3, Probabilidad y Estadística para Ingenieros, R. E. Walpole y R. H. Myers, Interamericana.

transmitido al decodificador. El decodificador efectúa la operación inversa, esto es, reconstruye la señal original de los errores predecidos cuantizados.

El diagrama a bloques de este sistema se muestra en la figura 2.18; aquí S_i es la secuencia de los valores de entrada de la muestra, S'_i es la predicción, entonces el error e_i es:

$$e_i = S_i - S'_i \quad (2.21)$$

Este error es cuantizado, codificado y transmitido.

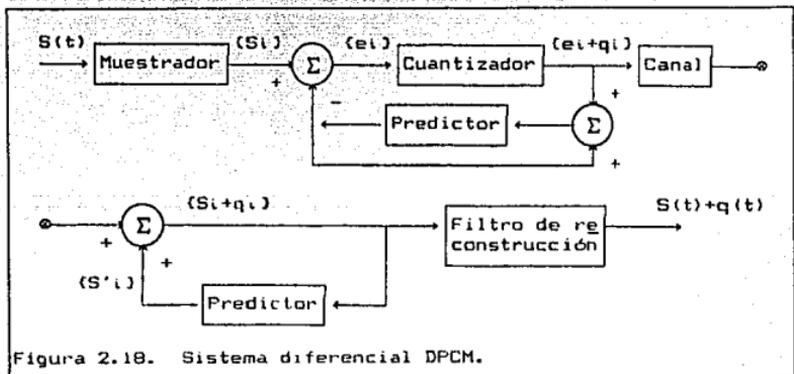


Figura 2.18. Sistema diferencial DPCM.

Cuando el número de niveles de cuantización es grande ($N \geq 8$) y se ocupa predicción lineal, cada muestra S_i se expresa como:

$$S'_i = a_1 S_{i-1} + a_2 S_{i-2} + a_3 S_{i-3} + \dots \quad (2.22)$$

De aquí a_i 's son los coeficientes del predictor.

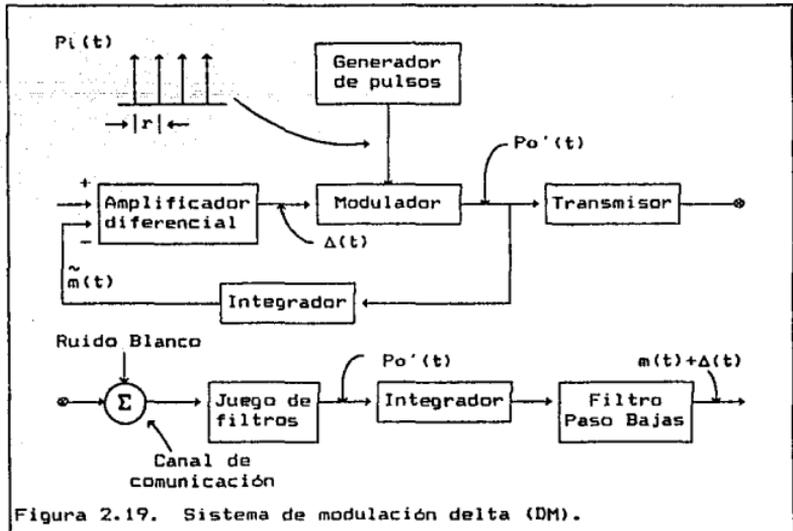
Este sistema es muy empleado en el procesamiento digital de señales de video, debido a que se emplea una menor cantidad de

bits en la codificación, para la compresión del ancho de banda ocupado.

2.3.4 Modulación Delta

La modulación delta es una técnica por la cual una señal analógica puede ser codificada dentro de dígitos binarios, este sistema tiene el mérito que la circuitería electrónica requerida para el transmisor y el modulador es sustancialmente simple, además es casi tan eficiente como el PCM y por lo tanto requiere un ancho de banda similar. La modulación delta es una versión del sistema diferencial PCM (DPCM). (Tiene código de 1 dígito).

Un sistema de modulación delta se muestra en la figura 2.19 el generador de pulso provee regularmente un tren de pulsos.



En la misma figura, el generador de pulso provee un tren de pulsos $P_i(t)$ de amplitud y polaridad fija, estos pulsos son arbitrariamente angostos pero con una área finita (impulsos).

El modulador recibe esos pulsos y también la señal $\Delta(t)$. La salida del modulador $P_o(t)$ es la entrada $P_i(t)$ multiplicado por $+1$ o por -1 , dependiendo de la polaridad de $\Delta(t)$, en este sistema no afecta la amplitud de $P_i(t)$ si $\Delta(t)$ es positiva cuando $P_i(t)$ ocurre, la multiplicación será por $+1$ y si $\Delta(t)$ es negativa entonces será por -1 . La salida $P_o(t)$ es pasada a través de una red de realimentación hacia el circuito integrador y la forma de onda resultante $\tilde{m}(t)$ consistente en una serie de escalones unitarios hacia arriba y hacia abajo. Esta señal $\tilde{m}(t)$ llega hasta el amplificador diferenciador, aquí es donde se compara con la señal original $m(t)$ y decide en base de dicha realimentación si la salida del pulso del modulador positivo o negativo.

La salida del amplificador diferenciador $\Delta(t) = m(t) - \tilde{m}(t)$ decide que polaridad del pulso de salida sería el orden correcto para la diferencia entre los 2 voltajes, el sistema de realimentación tiende a reducir la diferencia.

En la figura 2.20 se muestra las formas características en este sistema, notese que la señal $\tilde{m}(t)$ sigue a la señal $m(t)$ en forma escalonada y dependiendo de esto, está entregando a la salida una señal de pulso $P_o(t)$ que corresponde a un valor determinado por la señal $\tilde{m}(t)$, al comienzo del muestreo por la señal $\tilde{m}(t)$ comienza entregando unicamente pulsos positivos, al encontrarse con la señal $m(t)$ entrega pulsos alternados, es decir un positivo y un negativo, debido a que la señal es constante en este pequeño intervalo, posteriormente al ascender la señal $m(t)$ la señal $\tilde{m}(t)$ la sigue y entrega al ir ascendiendo pulsos de polaridad positivos y después la señal $m(t)$, la salida entrega pulsos negativos.

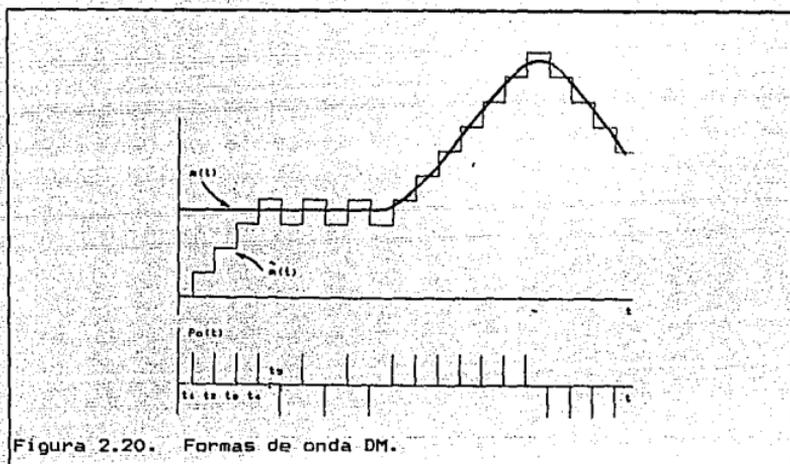


Figura 2.20. Formas de onda DM.

En la práctica los pulsos negativos de $P_o(t)$ pueden ser omitidos en la transmisión sin afectar la relación señal ruido (S/N) en el receptor final. Como en el caso de los sistemas PCM, el ruido también se introduce cuando se emplea la modulación delta. A causa de que la señal $\Delta(t)$ es cuantizada en dos niveles aparece el ruido de cuantización, similar al de PCM, que se presenta en la salida del receptor. Recuerdese que en PCM el ruido podría reducirse usando más niveles de cuantización, en este sistema puede reducirse muestreando con mayor frecuencia.

Ultimamente se encuentran sistemas de modulación delta que emplean velocidades de muestreo muy superiores a la velocidad de muestreo de Nyquist⁹ que se utiliza en PCM, esto significa que la

⁹ Vease, Capítulo 3, Mischa Schwartz, Transmisión de Información Modulación y Ruido, Mc Graw-Hill.

última velocidad de bits es superior a la originalmente esperada con el uso de 2 niveles de cuantización. Además del ruido de cuantización, se presenta otro tipo de ruido, llamado ruido de sobrecarga, este se presenta cuando los niveles de cuantización son muy pequeños para seguir la señal que varía rápidamente. El ruido de cuantización es el resultado del muestreo que en algunos casos la muestra queda arriba o debajo de la curva real y también es proporcional al tamaño del escalón usado S , por lo que al reducir el tamaño del escalón el ruido se reducirá más sin embargo valores pequeños de escalón provocan el ruido de sobrecarga.

En la figura 2.21 se observan dos tipos de ruido. Es obvio que existe un valor óptimo del tamaño del escalón S y este dependerá de las características de la señal de entrada $m(t)$, la velocidad de muestreo y el ruido total que pueda ser tolerado. El ruido de sobrecarga de la figura 2.21 se presenta debido a que el tamaño del escalón S fija un límite máximo a la pendiente de la señal de entrada que el modulador puede seguir.

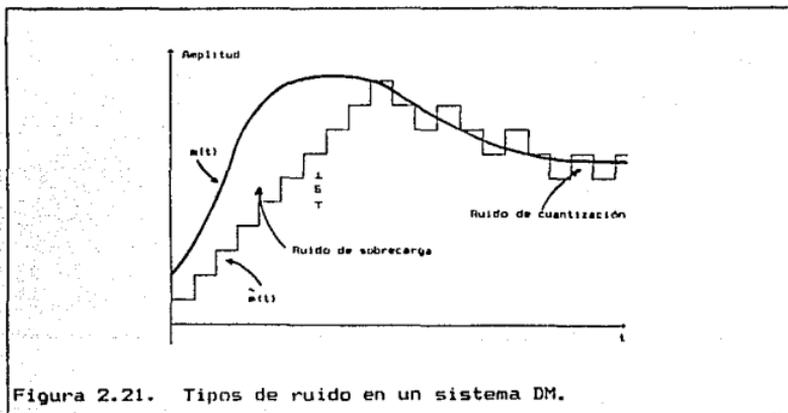


Figura 2.21. Tipos de ruido en un sistema DM.

2.3.3 Técnicas de codificación de imagen.

El objetivo principal, en el diseño de un sistema de codificación es una reducción de la transmisión proporcional del origen de la imagen subjetiva de alta calidad. Hay básicamente sólo dos caminos para cumplir esta meta, la reducción de la estática y la redundancia psicofísica del origen de la imagen.

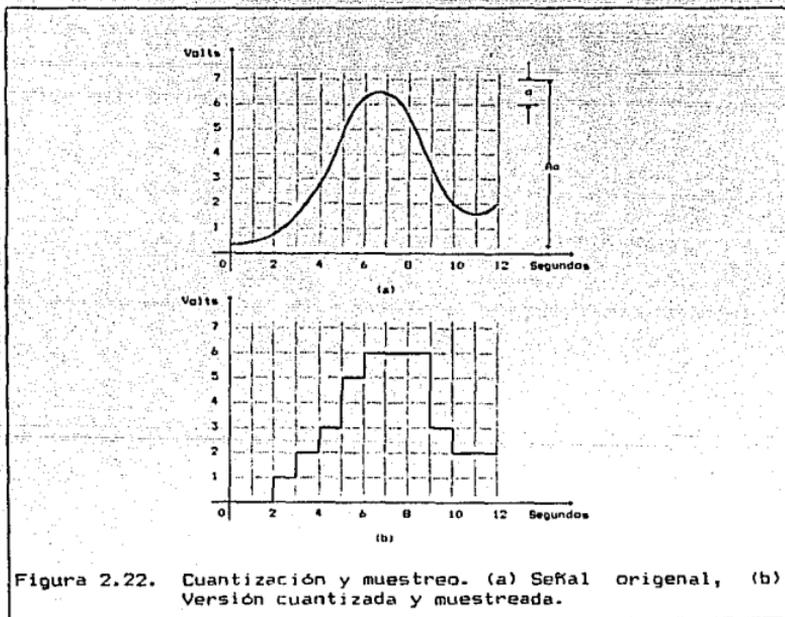
Una imagen original es normalmente altamente correlacional en espacio y temporalidad: esto es una fuerte dependencia entre los valores de píxeles individuales. Esta dependencia puede ser juzgada como una redundancia estática del origen de la imagen. El conocimiento de la redundancia estática del origen en los términos de las amplitudes de códigos de píxeles previos pueden ser reducidos para las necesidades de transmisión. La ventaja de este tipo de codificación es que es una prevención de la información, esto es, libre de error.

Si las imágenes son codificadas en un sistema de transmisión de imagen y son vistas por un observador humano, las limitaciones de percepción de la visión humana pueden ser explotadas para reducir las necesidades de la transmisión. El observador puede ser subjetivo para las limitaciones de agudeza perceptual en amplitud resolución espacial y temporal.

En un sistema específico, los pulsos muestreados deben cuantizarse, o bien, los procesos de cuantización y muestreo se realicen de forma simultánea. Este proceso se observa en la figura 2.22. Por el diseño propio del codificador, es posible descartar información estática de fuera del origen afectando la percepción de todo, o de parte, con una mínima degradación.

Algunos codificadores de imagen, usados para transmisión por facsimile binario, son basados enteramente en conceptos de codificación estática. La mayoría de los codificadores dejan algo

de degradación en orden para ejecutar una reducción en las necesidades de transmisión.



**NO
EXISTE
PAGINA**

C A P I T U L O 3

CUANTIZACION DE IMAGEN.

3.1 CUANTIZACION DE IMAGEN.

3.2 CUANTIZACION VISUAL.

- 3.2.1 CUANTIZACION DE CONTRASTE.
- 3.2.2 RUIDO PSEUDOALEATORIO DE CUANTIZACION.
- 3.2.3 CUANTIZACION DE IMAGENES HALFTONE.
- 3.2.4 CUANTIZACION DEL COLOR.

3.3 DISEÑO DE UN COMPARADOR.

3.4 CUANTIZACION PREDICTIVA.

3.1 CUANTIZACION DE IMAGEN

El siguiente paso del muestreo en la digitalización de una imagen es la cuantización. La mayoría de los algoritmos de compresión de datos de imagen requieren de un cuantizador, éstos se utilizan para revisar algunos de los métodos más comunes de cuantización.

Un cuantizador mapea una variable continua u dentro de una variable discreta u^* , la cual toma el valor de un conjunto finito de números (r_1, \dots, r_L) . Este mapeo es generalmente una función escalar (figura 3.1) y la regla de cuantización es como sigue: Definiendo $(t_k, k = 1, \dots, L + 1)$ como un conjunto de transición de incrementos o niveles de decisión con t_1 y t_{k+1} como el mínimo y máximo valor, respectivamente, de u . Si u consiste en un intervalo $[t_k, t_{k+1})$, entonces es mapeado para $r_k, k \in [1, L]$. La cantidad r_k , llamada reconstrucción de nivel, es el valor cuantizado de u y también esta dentro del intervalo (t_k, t_{k+1}) .

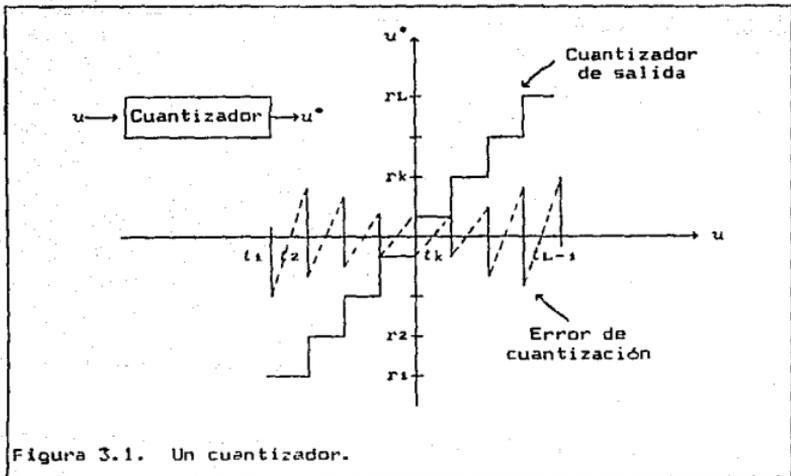
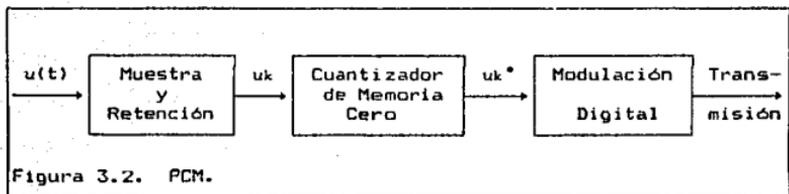


Figura 3.1. Un cuantizador.

El problema para diseñar el cuantizador es cuando se va a determinar la transición óptima y niveles de reconstrucción, dada la probabilidad de densidad y un criterio de optimización. Puesto que el mapeo del cuantizador es irreversible, el cuantizador introduce distorsión la cual con el diseño del cuantizador adecuado se intenta minimizar. Hay varios diseños de cuantizadores útiles que ofrecen varias opciones, entre simplicidad y ejecución.

La cuantización de muestras de imagen para la transmisión es llamada modulación por codificación de pulsos (PCM) (figura 3.2).



Se consideraran sólo cuantizadores de memoria cero, el que opera sobre una muestra de entrada en un tiempo, y la salida del valor depende sólo de esa entrada. Dichos cuantizadores utilizan técnicas de codificación de imágenes tales como la modulación por codificación por pulsos (PCM), DPCM, codificación por transformada.

3.2 CUANTIZACION VISUAL

El número de niveles de cuantización o niveles de gris podrían ser diferentes para suprimir cualquier efecto de contorno. La cuantización uniforme de imágenes típicas, donde los píxeles representan la función de luminancia, requiere de cerca de 64 niveles de gris o seis bits. Los efectos de contorno propician un

efecto visible con 5 bits/píxel. El cuantizador medio cuadrático iguala el histograma de una imagen dada, para necesitar sólo 5 bits/píxel para suprimir el efecto de contorno.

En la cuantización de imágenes se tiene que procesar visualmente (o examinar) por el ojo por ser completamente sensitivo al contorno y errores, el cual afecta la estructura local. Sin embargo el contorno no contribuye mucho al error medio cuadrático. Así, el modelo de una cuantización visual podría probarse para mantener el contorno de cuantización bajando el nivel de visibilidad sobre el rango de luminancia a ser desplegado.

3.2.1 Cuantización de contraste.

El sistema visual humano no percibe igual cambio en la luminancia. Sin embargo, la sensibilidad visual es uniformemente cercana con respecto a la normalización, percepción justa, cambios en la luminancia. Esto es, si L y $L + \Delta L$ son de luminancia diferente, entonces

$$\frac{\Delta L}{L} \approx \text{constante.} \quad (3.1)$$

Esto es llamado regla de Weber. Definida como

$$C = \log L \quad (3.2)$$

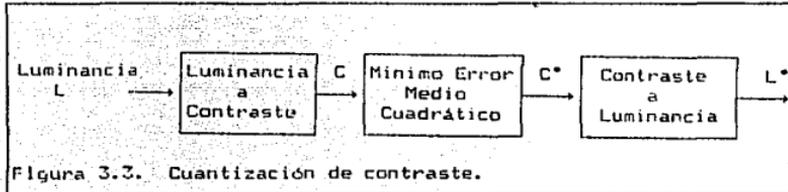
entonces $\Delta C \approx \Delta L/L$ es cercanamente constante. Esto significa que la sensibilidad visual es cercanamente uniforme a los cambios en C , el cual es llamado contraste. Las funciones comunes son

$$C = \begin{cases} \alpha \log (1 + \beta L), & L \in [0, 1) & (3.3) \\ \alpha L^\beta & & (3.4) \end{cases}$$

donde α y β son constantes. Por ejemplo, en la ecuación 3.4 los valores $\alpha = 1$, $\beta = 1/3$, en la ecuación 3.3 los valores

$\alpha = \beta / \log(1 + \beta)$ y α esta entre 6 y 18.

Para la representación de un contraste dado se utiliza el mínimo error medio cuadrático del cuantizador (ver figura 3.3). Para desplegar (o reconstruir) la imagen, la cuantización del contraste es transformada en el interior para el valor de luminancia por la transformación inversa. Estudios experimentales indican que con el cuarto bit de cuantización del contraste, el efecto del contorno puede ser generosamente minimizado.



3.2.2 Ruido pseudoaleatorio de cuantización.

Otro método para suprimir el efecto de contorno es atribuido a L. G. Robert. Primero se adiciona una suma de ruido pseudoaleatorio a las muestras de luminancia antes de la cuantización (ver figura 3.4). Este ruido pseudoaleatorio es también llamado dither. Para desplegar la imagen, la misma secuencia (u otra) pseudoaleatoria es restada desde el cuantizador de salida. La entrada de ruido causa que algún pixel vaya sobre el nivel de decisión original y otros por debajo del nivel de decisión.

Así, el valor promedio del pixel de cuantización es casi el mismo con o sin el ruido aditivo. Durante el despliegue, el ruido tiende a llenar las regiones de contorno en cada camino en que el espacio promedio no es alterado. La suma de la adición dither

podría ser retenida lo suficientemente poco para mantener la resolución espacial, pero lo suficientemente grande para permitir los valores de luminancia varíen aleatoriamente cerca de los niveles de decisión del cuantizador. Usualmente, el ruido podría afectar mínimamente el bit significativo del cuantizador. La calidad de imagen razonable es realizada por el 4 o 5 bit del cuantizador.

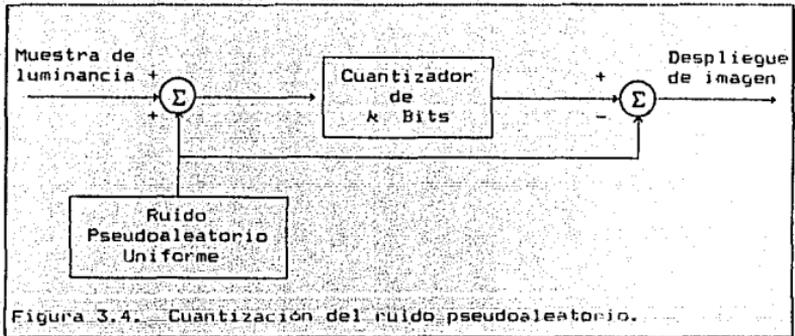
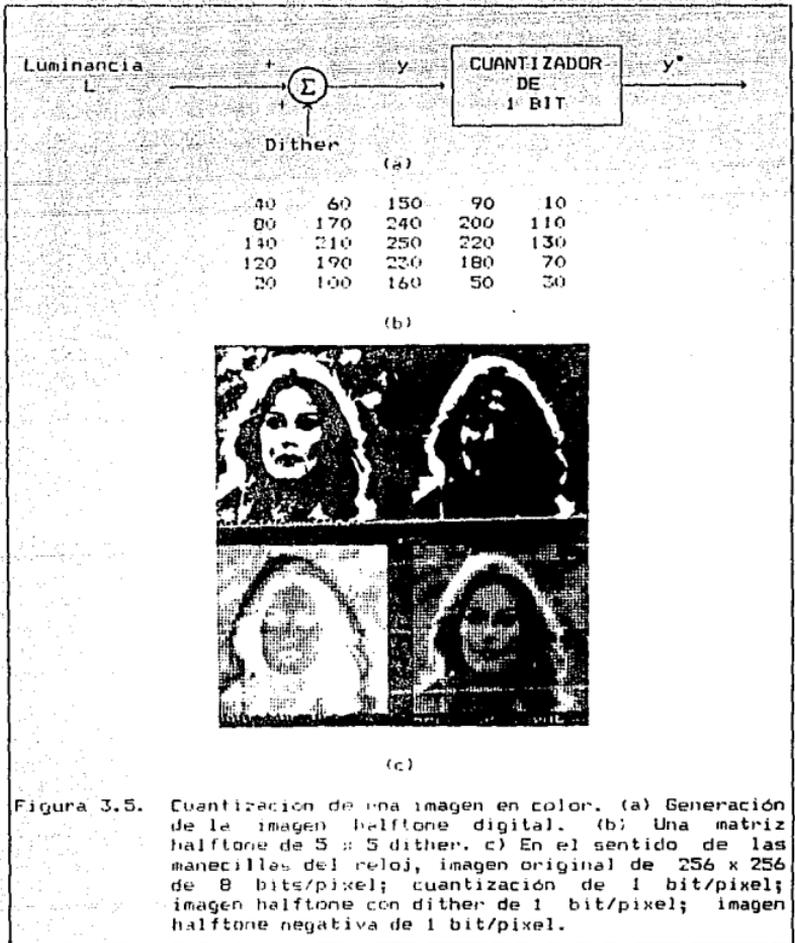


Figura 3.4. Cuantización del ruido pseudoaleatorio.

3.2.3 Cuantización de imágenes halftone.

Las imágenes halftone son imágenes binarias las cuales dan un rendimiento en la escala de gris. Algunos ejemplos son las imágenes impresas comúnmente (por ejemplo, todas las imágenes impresas en papel son halftone). Para cada píxel, la imagen dada (frecuentemente por encima de la muestra, por ejemplo, una imagen de 256 x 256 puede ser impresa sobre una de un grado de 1024 x 1024 de puntos blancos y negros) es sumada al número aleatorio (muestra dither) y la señal resultante es cuantizada por un bit del cuantizador. La salida (0 o 1) representa un punto blanco o negro. En la figura 3.5 se muestra un ejemplo de las imágenes halftone.



En la práctica la señal dither es un modelo pseudoaleatorio bidimensional de umbral (por ejemplo, una matriz dither de 5 x 5)

la cual es representada periódicamente por encima de la imagen (figuras 3.5a y 3.5b). La imagen halftone puede manifestar mureo en los modelos, si la imagen y la matriz dither en períodos similares.

Los mejores algoritmos halftone son diseñados para minimizar el efecto de mureo. La figura 3.5c muestra una imagen halftone de 256×256 de un bit generada digitalmente de una imagen original de 256×256 de 8 bits. Comparando un bit cuantizado de imagen el bit-uno halftone tiene un mejor rendimiento visual. El rendimiento de la escala de gris en halftone es debida a la ejecución promedio del espacio local por el ojo humano.

3.2.4 Cuantización del color

Las consideraciones perceptuales comienzan siempre con la cuantización del color de las señales. Un pixel de una imagen en color podría ser considerada como un vector tridimensional C , cuyos elementos C_1 , C_2 , C_3 representan los tres colores primarios. La gama de color (por ejemplo, la percepción de todos los colores) es un sólido altamente irregular en el espacio tridimensional.

La cuantización de una imagen en color requiere de añadir celdas de cuantización para el color en la gama del color. Además, las celdas de cuantización serán desiguales en el tamaño porque los cambios en el color no coordinan, en general, resulta en cambios iguales en la percepción del color. La figura 3.6 muestra un procedimiento agradable de cuantización del color.

3.3 DISEÑO DE UN COMPARADOR.

Un comparador (compresor-expansor) es un cuantizador uniforme precedido y seguido por una transformación no lineal, como se

muestra en la figura 3.7. La variable aleatoria de entrada u primero es pasado a través de una no memoria de transformación no lineal $f(\cdot)$ para generar otra variable aleatoria w . Esta variable aleatoria es uniformemente cuantizada dado y e (y_i) , el cual es transformado no linealmente por $g(\cdot)$ dando la salida v .

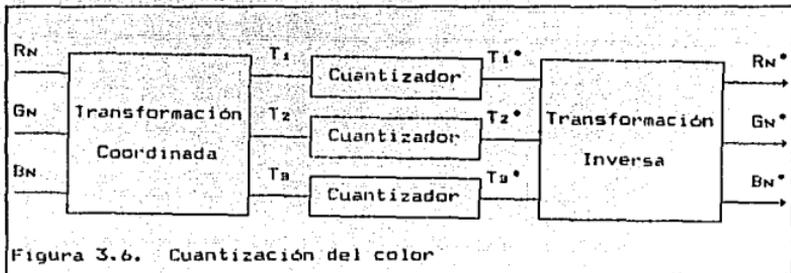


Figura 3.6. Cuantización del color

La transformación se u por u^* es una cuantización no uniforme. Las funciones f y g son determinadas sobre todo por el sistema de aproximaciones del cuantizador de Lloyd-Max. El resultado esta dado por

$$g(x) = f^{-1}(x) \quad (3.5)$$

$$f(x) = 2a \left\{ \frac{\int_{t_1}^x [p_u(u)]^{1/3} du}{\int_{t_1}^{t_{L+1}} [p_u(u)]^{1/3} du} \right\} - a \quad (3.6)$$

donde $[-a, a]$ es el rango de w sobre el cual opera el cuantizador uniforme. En la figura 3.7, se muestra como $f(x)$ y $g(x)$ producen las funciones de compresión y expansión, respectivamente, en su dominio.

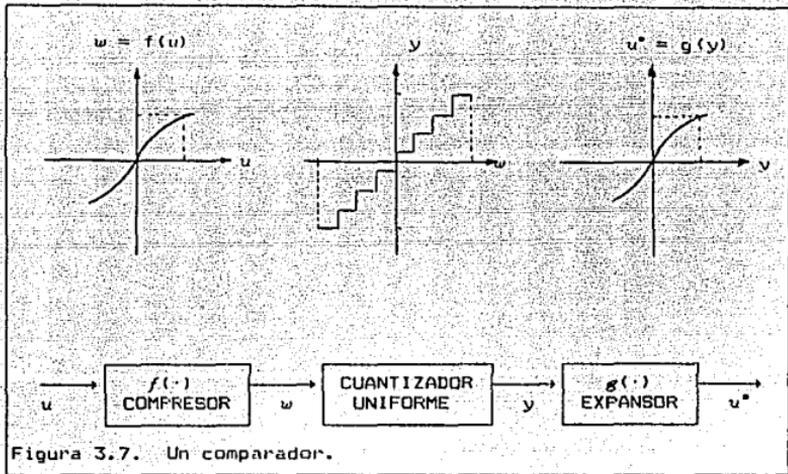


Figura 3.7. Un comparador.

3.4 CUANTIZACION PREDICTIVA.

Si se considera cero la memoria de cuantización de variables aleatorias escalares, por ejemplo, las entradas sucesivas para un cuantizador tratado independientemente. Frecuentemente, la utilidad de una secuencia de datos tiene dependencia estadística o redundancia de una muestra a la siguiente.

Considerando una secuencia aleatoria $\{u_n\}$ y que la información supuesta sobre la muestra para $n = k - 1$ ha sido transmitida de alguna forma. Dada u_n que denota la reproducción del valor de u_n . Cuando u_k es transmitida, la ganancia es tomada del dato previamente transmitido que elementos podrían contener alguna información de él. Por lo tanto, una cantidad \hat{u}^*_k , una estimación de u_k es predecida desde las muestras previamente transmitidas y la secuencia del error de predicción es definida como

$$e_k \triangleq u_k - \hat{u}_k \quad (3.7)$$

Ahora, es suficiente e_k en lugar de u_k para la transmisión. Si e_k es el valor de cuantización de e_k , u_k^* reproduce el valor de u_k entonces es dada por

$$u_k^* = \hat{u}_k + e_k^* \quad (3.8)$$

Un método común de transmisión de datos utilizando cuantización predictiva es el llamado DPCM. El concepto del sistema es mostrado en la figura 3.8. Los componentes de un sistema DPCM son un predictor y un cuantizador. De las ecuaciones 3.7 y 3.8 es fácil deducir que el error en la reproducción de u_k , está dado por

$$\delta u_k \triangleq u_k - u_k^* = e_k - e_k^* = q_k \quad (3.9)$$

es igual al error en cuantización de e_k . De aquí, para minimizar $\sigma^2(k)$, la varianza del error de predicción, \hat{u}_k podría ser la mejor estimación de la raíz media de u_k y es dada por la media condicional.

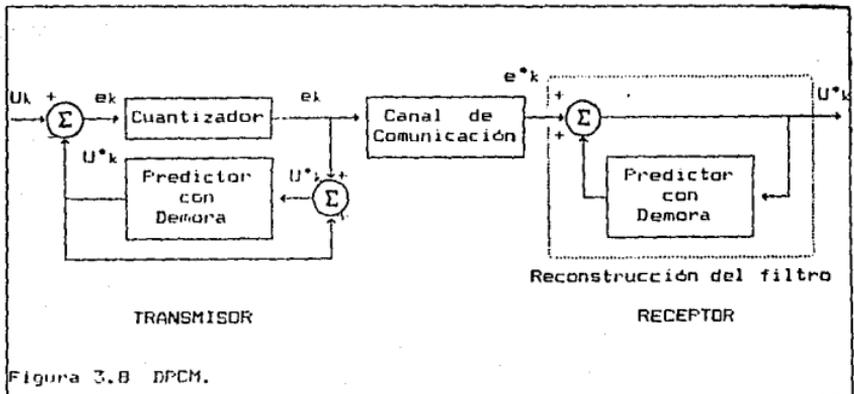


Figura 3.8 DPCM.

**NO
EXISTE
PAGINA**

CAPITULO 4

CODIFICACION DE IMAGENES.

4.1 CLASIFICACION DE LOS METODOS DE CODIFICACION.

4.2 TECNICAS DE CODIFICACION ESTADISTICA.

4.2.1 CODIFICACION SIMPLE POR PIXEL.

4.2.2 CODIFICACION PREVIA DE PIXEL.

4.3 CODIFICACION CORRIDA (RUN CODING).

4.3.1 CODIFICACION BIT-PLANE.

4.4 CODIFICACION PREDICTIVA.

4.4.1 MODULACION DELTA.

4.4.2 DPCM.

4.5 TECNICAS DE CODIFICACION INTERPOLATIVA.

4.5.1 INTERPOLACION DE ORIGEN.

4.5.2 INTERPOLACION DE DESTINO.

4.6 CODIFICACION DE IMAGEN POR TRANSFORMADA.

4.6.1 CODIFICACION DE IMAGEN MONOCROMATICA POR TRANSFORMADA.

4.6.1.1 CODIFICACION POR ZONA.

4.6.1.2 CODIFICACION DE UMBRAL.

4.6.2 CODIFICACION ADAPTIVA POR TRANSFORMADA.

4.6.3 CODIFICACION DE IMAGENES EN COLOR POR TRANSFORMADA.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

4.1 CLASIFICACION DE LOS METODOS DE CODIFICACION.

Existen varios métodos de clasificación de codificación. En general, la codificación puede clasificarse dentro de cuatro categorías: modulación por codificación de pulsos, codificación predictiva, codificación por transformada, codificación interpolativa y extrapolativa, existe además una quinta clase la cual consiste en esquemas miscelaneos que no caen dentro de esas cuatro clases. Cada una de ellas puede ser dividida basandose en los parámetros del codificador que son fijados o cuales de ellos cambian el tipo de datos que se comienzan a codificar.

En la tabla 4.1 se muestra una clasificación de métodos de codificación.

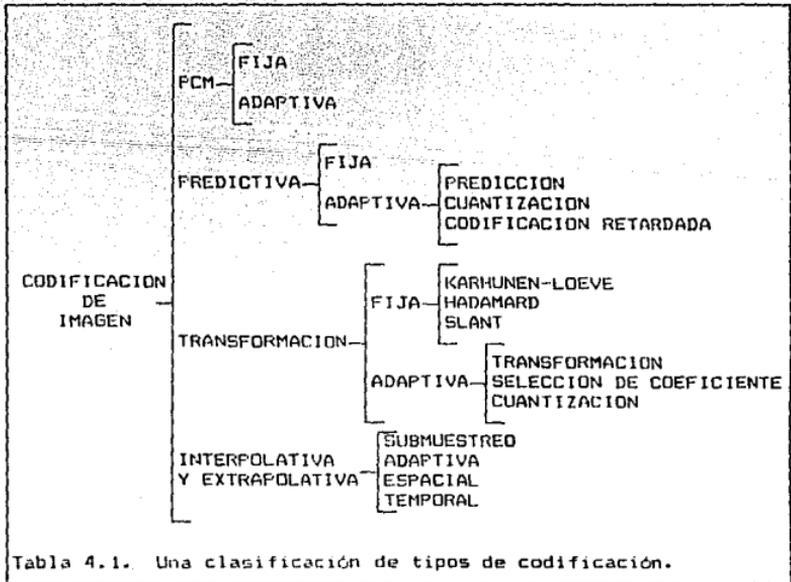


Tabla 4.1. Una clasificación de tipos de codificación.

En PCM, la representación discreta de amplitud de la muestra es provista para usar un suficiente número de niveles de cuantización, así que la degradación debida a la cuantización no es fácilmente visible; el número de niveles usados para cada muestra esta dada por 2^K niveles, cada nivel es representado por una palabra binaria que contiene K bits.

En la codificación predictiva, se obtiene una muestra para predecir el pixel a ser codificado, la predicción se hace usando los valores codificados de los pixeles previamente transmitidos y sólo el error de predicción es cuantizado para su transmisión; tal aproximación puede ser hecha adaptiva por el cambio del predictor base o por la no transmisión del error de predicción siempre que este debajo de cierto umbral.

En la codificación por transformación, se hace primero una representación alternativa de la señal para tomar combinaciones lineales de muestras en un bloque de pixeles (llamados coeficientes) y los coeficientes seleccionados son cuantizados para su transmisión. Varias transformaciones (como la Hadamard simple o la medianamente compleja Karhunen-Loeve) han sido usadas. Los códigos de transformación pueden ser hechos adaptivos por el cambio del tipo de transformación y el criterio para la selección y cuantización de los coeficientes.

La codificación interpolativa y extrapolativa trabajan con un principio diferente a los antes mencionados. La adaptación de estos sistemas consisten en la variación del criterio para la selección de las muestras a ser enviadas, así como también la estrategia para interpolar y extrapolar las muestras restantes.

Algunos sistemas de codificación pueden ser la combinación de las cuatro clases, algunos de ellos son aplicables a un tipo especial de imágenes por ejemplo, la codificación por longitud rápida (Run length coding), algunos otros hacen referencia a

mediciones estadísticas de brillantez en la imagen y son por los tanto codificaciones por técnicas estadísticas.

Existe otro tipo de clasificación: por el tipo de imagen a ser codificada: pictorial y no pictorial. Las imágenes pictoriales son escenas naturales de objetos o despliegues de 2 dimensiones que son normalmente vistas por un observador humano, mientras que las no pictoriales son arreglos de datos de 2 dimensiones, tales como el rango de un radar contra la velocidad de datos que no son normalmente vistos por un observador. Otra clasificación es por el tipo fotométrico: monocromático, color o imágenes multiespectrales.

4.2 TECNICAS DE CODIFICACION ESTADISTICA.

En este tipo de codificación; las mediciones estadísticas de la distribución de brillantez indican que imágenes naturales contienen una suma del número total de bits que son requeridos para codificación directa PCM. Por ejemplo, si una imagen monocromática contiene $N \times N$ pixeles. Si en cada nivel de brillantez del pixel es cuantizado a B niveles entonces un total de

$$T = DN^2 \quad (4.1)$$

diferentes imágenes pueden ser representadas. No todas las T imágenes ocurren normalmente con igual probabilidad.

En un codificador PCM el número de niveles de cuantización por pixel es usualmente cambiado a ser

$$B = 2^b \quad (4.2)$$

donde b es un entero (de 6 a 8). En la codificación PCM se requiere un total de:

$$N_{II} = b N^2 \quad (4.3)$$

bits para cada código de los N^2 píxeles con una longitud constante de código de palabra de b bits, en este tipo de codificación debe tomarse en cuenta la entropía de la señal (H), considerando a P_i como la probabilidad de ocurrencia de la i -ésima posible imagen. La entropía es la medición del número mínimo de bits requeridos para codificar la imagen, y esta dada por

$$H = - \sum_{i=1}^T P_i \log_2 p_i \quad (4.4)$$

Para imágenes naturales, la entropía H es de una magnitud mucho menor que el número de bits de código PCM (N_b), e indica la ineficiencia de codificación de imagen PCM.

Por lo tanto, existe una discordancia entre el número de bits empleado, uno por la codificación PCM y el otro limitado por la entropía H en la cual la entropía lo limita a pocos bits, pero esto es en sentido teórico, en la práctica no es factible que se presente este caso y la estrategia a seguir es asignar códigos de una longitud relativamente corta para imágenes con alta probabilidad de ocurrencia y asignar códigos de una longitud larga para imágenes con poca probabilidad.

4.2.1 Codificación simple de pixel.

Es el más simple de los códigos estadísticos, es uno en el cual a cada pixel se le asigna un grupo de código basado en la amplitud cuantizada.

Para codificación eficiente el código asignado sería tal que el valor del pixel con alta probabilidad de ocurrencia sería asignado un grupo de códigos con un pequeño número de bits. Recíprocamente, al pixel de rara ocurrencia le sería asignado un grupo de código largo; si este proceso es efectuado eficientemente, la longitud promedio de el código sería igual a la simple entropía del pixel de la imagen.

Existen códigos con alta eficiencia que pueden ser empleados para codificar imágenes por pixel, por ejemplo: el código Shannon-Fano y el código Huffman¹⁰, este último es el más eficiente de los códigos posibles. En general, estos códigos son de longitud variable y esto es un problema asociado con la codificación estadística.

4.2.2 Codificación previa del pixel.

Una variante de la codificación previa del pixel es para codificar las diferencias en el valor del pixel después de codificar el primer pixel a lo largo de una línea. Si cada pixel tiene L niveles de cuantización entonces la diferencia puede asumir $2L - 1$ valores.

La estrategia de codificación es como sigue, las pequeñas diferencias reciben códigos de palabra individuales; si la diferencia excede algún nivel específico, el valor actual efectivo del pixel es codificado y anexado a un código prefijo que distingue el código total de palabra de la diferencia codificada. La tabla 4.2 ilustra un código Shannon-Fano para codificar diferencias de pixeles, este código llega a tener eficiencias altas, como del 90%, y además son factibles de implementar.

4.3 CODIFICACION CORRIDA (RUN CODING).

Este tipo de codificación es relativamente simple en el cual la amplitud de pixeles adyacentes a lo largo de una línea son comparados. Si ocurre un cambio significativo en detalle, la corrida es dada a existir. Cualquier función de amplitud del

¹⁰ Véase, Apéndice 3, Digital Image Processing, William K. Pratt, Wiley-Interscience Publication

brillo del pixel en el fin de la corrida o una amplitud de señal diferencia, es transmitida a lo largo con una indicación de la ubicación del fin de la corrida.

Pixel Diferencia D	Palabra código
0	1
+ 1	0 100
- 1	0 101
+ 2	0 110
- 2	0 111
+ 3	00 100
- 3	00 101
+ 4	00 110
- 4	00 111
$ D \geq 5$	000 + 6 bits/pixeles

Tabla 4.2. Ejemplo de codificación por diferencia de pixel sin error.

Si la ubicación del fin de la corrida es determinada por el conteo del número de elementos del comienzo de la línea para la ocurrencia del fin de la corrida, el sistema es llamado codificación corrida-fin (run end coding). La localización del fin de una corrida puede ser especificado en términos de la distancia relativa a partir del fin previo. Este sistema es conocido como codificación de corrida-longitud (run length coding).

Para sistemas de transmisión digital la amplitud y posición de una corrida son transmitidos como un grupo de código de bits.

La codificación corrida-fin tiene la desventaja de requerir un gran número fijo de bits para describir cada posición de la corrida como un punto a lo largo de una línea compuestas de varios cientos de pixeles. El sistema corrida-longitud requiere grupos cortos de bits para especificar la posición de una corrida en promedio. Sin embargo, la longitud variable de la posición del

código presenta algunos problemas en la codificación. Una variación del método de codificación de posición es el límite de la longitud de la posición del código describiendo la longitud corrida para una longitud fija uniforme.

La longitud corrida-longitud es más práctica para imágenes que requieren pocas escalas de gris. Existen modelos probabilísticos que analizan la codificación corrida-longitud y designan a la letra z para representar el número de píxeles en una corrida y en base a esto se analiza mediante distribuciones de probabilidad la corrida entre los píxeles.

4.3.1 Codificación Bit-Plane

En la codificación PCM de una imagen, el código puede ser organizado conceptualmente dentro de planos correspondientes a su posición del píxel con los bits más significativos ocupando el plano inferior. Se ha encontrado que en imágenes naturales los bits del plano inferior rara vez cambian, mientras los bits del plano superior fluctúan casi arbitrariamente.

Una reducción del ancho de banda es posible, en principio, por el estado de transición del bit de la codificación corrida-longitud en cada plano del bit. Sin embargo, en experimentos de simulación indican que la obtención de la reducción del ancho de banda es mucho menor que 2 a 1.

4.4 CODIFICACION PREDICTIVA.

Un sistema de codificación predictiva se muestra en la figura 4.1, en este sistema el valor de cada píxel explorado es predicho basándose en alguna historia previa del elemento explorado; la predicción estimada es restada del valor actual del píxel, y la señal diferencia es entonces cuantizada, codificada y

transmitida.

En el receptor la señal diferencia cuantizada es usada para formar una reconstrucción de la señal de imagen. Una reducción del ancho de banda es posible por cuantización rudimentaria de la diferencia de la señal para ser transmitida.

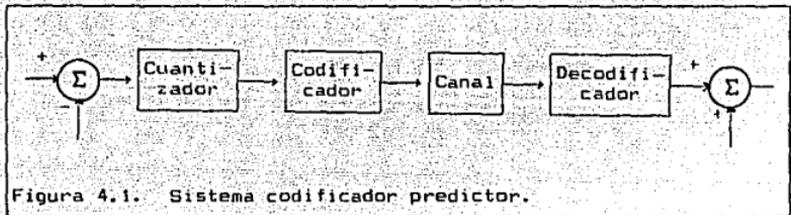


Figura 4.1. Sistema codificador predictor.

4.4.1 Modulación delta (DM).

Aunque este sistema ya se analizó en el capítulo II, ahora es aplicado únicamente a señales analógicas de video.

La modulación delta es la forma más simple de un codificador predictivo de datos. Un sistema de modulación delta es mostrado en la figura 4.2. En este sistema una señal analógica limitada en banda de video alimenta a un dispositivo diferenciador. Si la señal diferencia es positiva, un generador de pulso produce un pulso positivo del tamaño del escalón dado en el instante muestreado, de otro modo se produce un pulso negativo. Estos pulsos son representaciones binarias que son transmitidas, en el receptor estos pulsos son reconstruidos (integrados) para formar la señal de video. En esta versión de la modulación delta la predicción se basa sólo en el pixel previo a lo largo de una línea, y el pixel de la señal diferencia es cuantizado para sólo 2 niveles.

En el uso de este sistema se presentan inconvenientes, los cuales se mencionaron con anterioridad, uno de ellos es el ruido de sobrecarga y uno de los medios de reducirlo es asumir más de 2 niveles, con lo que se aumentará la complejidad del sistema.

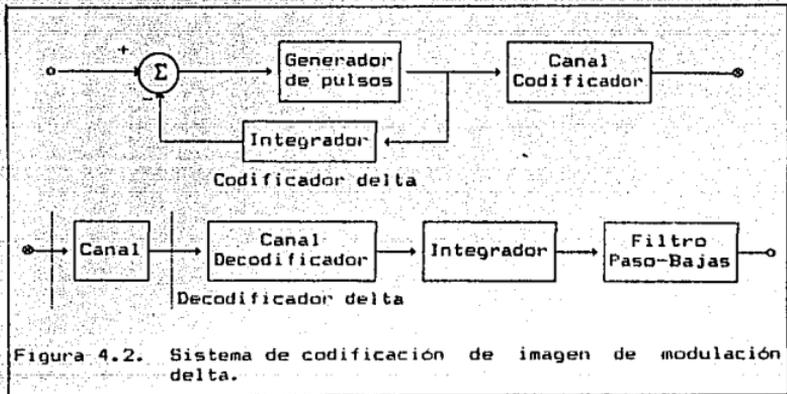


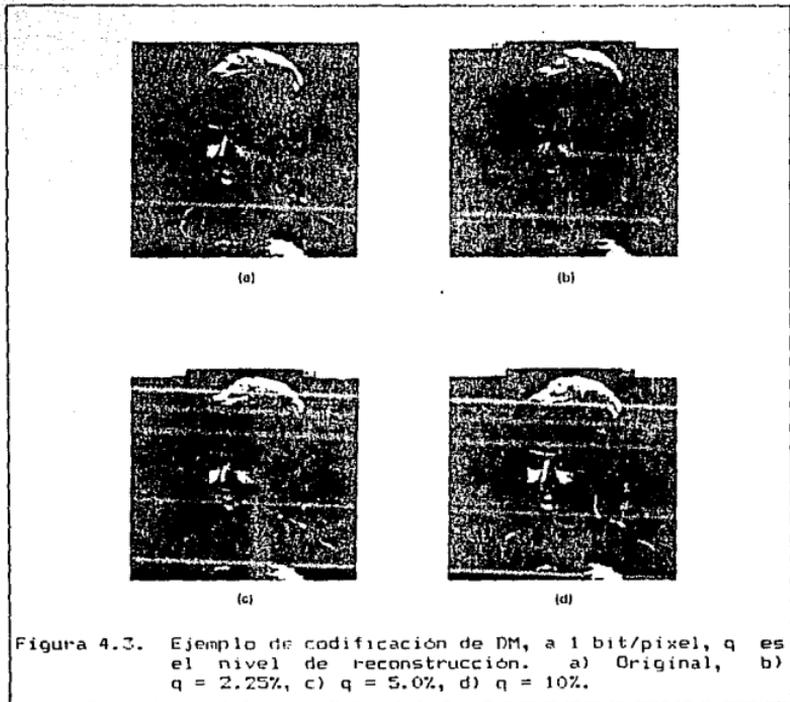
Figura 4.2. Sistema de codificación de imagen de modulación delta.

En la figura 4.3 se muestran fotografías de una simulación por computadora utilizando codificación delta. Cuando el nivel de reconstrucción es puesto al 2.5% del pico de la amplitud de la imagen, la imagen codificada presenta un considerable ruido de sobrecarga; en un nivel de reconstrucción de 10% resulta en un alto grado de ruido de cuantización (error granular), en una reconstrucción de nivel del 5% propicia una relación entre el ruido de sobrecarga y ruido de cuantización.

4.4.2 DPCM

Dependiendo del número de niveles del cuantizador, se distingue entre modulación delta y modulación diferencial por codificación de pulsos, en éste el número de niveles es mayor que

2. Este sistema se basa en el valor de la muestra previa a lo largo de la línea, y la diferencia entre la muestra presente, su estimación es cuantizada y codificada para su transmisión.



La figura 4.4 contiene un diagrama a bloques del sistema codificador de imagen DPCM. En este sistema la imagen continua es espaciadamente muestreada y la diferencia entre el pixel actual y la estimación cuantizada y codificada. Usualmente la señal diferencia es cuantizada a 8 niveles y codificada con 3 o 4 bits.

De este modo la reducción de ancho de banda es de 6 a 8 bits por pixel del convencional PCM a 3 bits por pixel para DPCM.

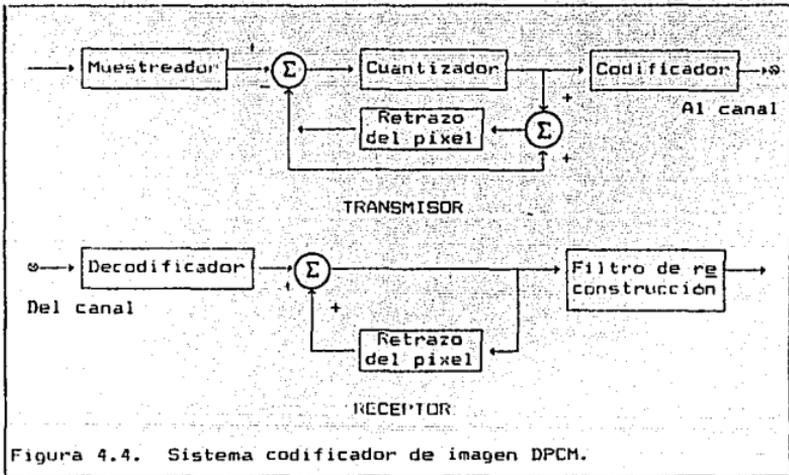


Figura 4.4. Sistema codificador de imagen DPCM.

En un codificador básico DPCM la predicción se basa en la señal diferencia cuantizada del pixel previamente barrido a lo largo de la línea de una imagen. En el receptor la señal diferencia decodificada es reconstruida y combinada desde un predictor idéntico hasta uno en el transmisor para proveer una reconstrucción de la imagen original.

Se propone usar una escala no lineal en el cuantizador para DPCM; ya que con un cuantizador no lineal la calidad de la imagen reconstruida es mejorada substancialmente, sin embargo en otras aplicaciones se ha encontrado que cuando menos 8 niveles de cuantización son siempre requeridos.

Subjetivamente las imágenes codificadas con 3 bits, 8 niveles

en el sistema DPCM, manifiestan en tener la misma calidad que con imágenes con 5 o 6 bits en el sistema PCM, excepto por algunos errores en la proximidad de los bordes de la imagen.

En un sistema PCM de 8 niveles es posibles emplear un código estadístico de longitud variable como el código Huffman que un código de longitud constante de 3 bits y así ejecutar una compresión en la codificación. Estudios de simulación de imágenes indican que un promedio en la proporción de codificación de cerca 2.5 bits por pixel puede ser llevada a cabo con el código Huffman de longitud variable. Un código típico de longitud variable se muestra en la tabla 4.3.

Nivel No.	Código palabra longitud	Código
1	12	100101010101
2	10	1001010100
3	8	10010100
4	6	100100
5	4	1000
6	4	1111
7	3	110
8	2	01
9	2	00
10	3	101
11	4	1110
12	5	10011
13	7	1001011
14	9	100101011
15	11	10010101011
16	12	100101010100

Tabla 4.3. Código típico de longitud variable para codificación DPCM con 16 niveles de cuantización.

4.5 TECNICAS DE CODIFICACION INTERPOLATIVA.

Los sistemas de codificación interpolativa están basados en

representaciones numéricas o técnicas de aproximación por medio de una secuencia o valores de pixel que son acomodados por funciones continuas. Hay dos procesos de interpolación básicos que son aplicados en la codificación de imagen: la interpolación de origen y la interpolación de destino.

4.5.1 Interpolación de origen.

En este sistema los valores de luminancia de una imagen son aproximados por funciones continuas dentro de alguna banda de errores permisibles. En la figura 4.5 se ilustra un interpolador de orden cero.

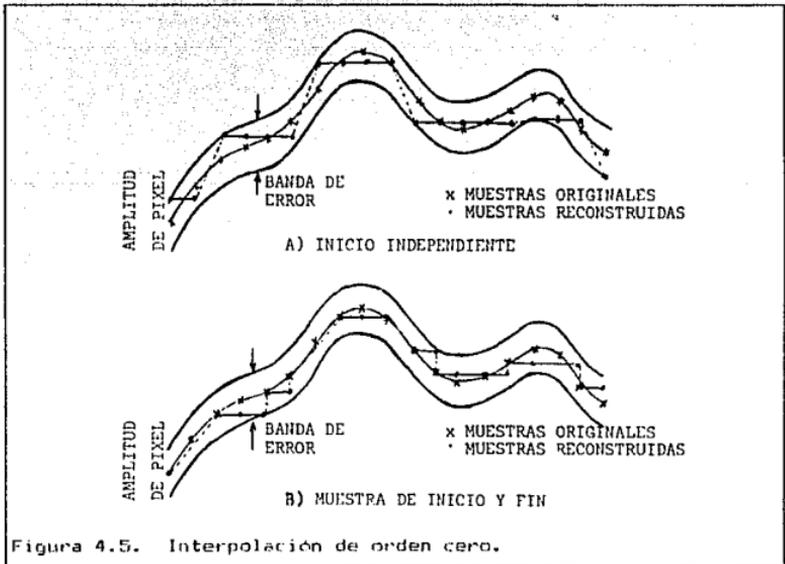


Figura 4.5. Interpolación de orden cero.

En el ejemplo, una banda de error de tolerancia se establece

cerca de cada valor de pixel y una línea de segmentos es acomodada dentro de la banda de error; cada pixel es espaciado por un segmento de línea horizontal. La coordenada vertical y la muestra de cada línea horizontal es entonces transmitida. En el receptor los valores de pixel son reconstruidos para la amplitud del segmento de línea horizontal.

Este tipo de interpolación permite una gran libertad en el acomodo de los segmentos de línea para los datos de imagen y así proveer una representación más eficiente en términos de minimización de las longitudes de línea horizontal. La figura 4.5b, describe un interpolador simplificado de orden cero en el cual el segmento de línea horizontal es restringido al comienzo de la muestra del valor del pixel y al final del tiempo de la muestra.

La forma simplificada de un interpolador de primer orden es equivalente a un codificador de corrida-longitud que codifica los valores de luminancia entre bordes.

Funciones polinomiales de alto orden, pueden ser usadas para codificación interpolativa, pero al computarizarse se involucra en el proceso de interpolación un rápido aumento del grado del polinomio.

4.5.2 Interpolación de destino.

Una reducción del ancho de banda de 2 a 1 puede ser obtenida simplemente al transmitir sólo líneas impares de un cuadro de imagen y entonces generar las líneas pares artificialmente por funciones de interpolación en el receptor. Alternativamente, la imagen original puede ser muestreada en un tablero patrón y los pixeles faltantes pueden ser interpolados en el receptor por una función de interpolación de 2 dimensiones, la reducción de ancho de banda de alto orden puede ejecutarse por submuestreo espacial

de una imagen o interpolación de estas submuestras en el receptor.

El submuestreo puede ser ejecutado simplemente por descartación de muestras de imagen, de acuerdo con algún patrón específico, esto es, sólo se transmite el pixel de la esquina superior izquierda de cada grupo de 4×4 . Sin embargo, esta estrategia lleva a un error de aliasing¹¹ si la imagen original es muestreada a la frecuencia de muestreo de Nyquist o a una menor. Una mejor aproximación es la limitación en banda de la imagen original por una aproximación análoga o discreta.

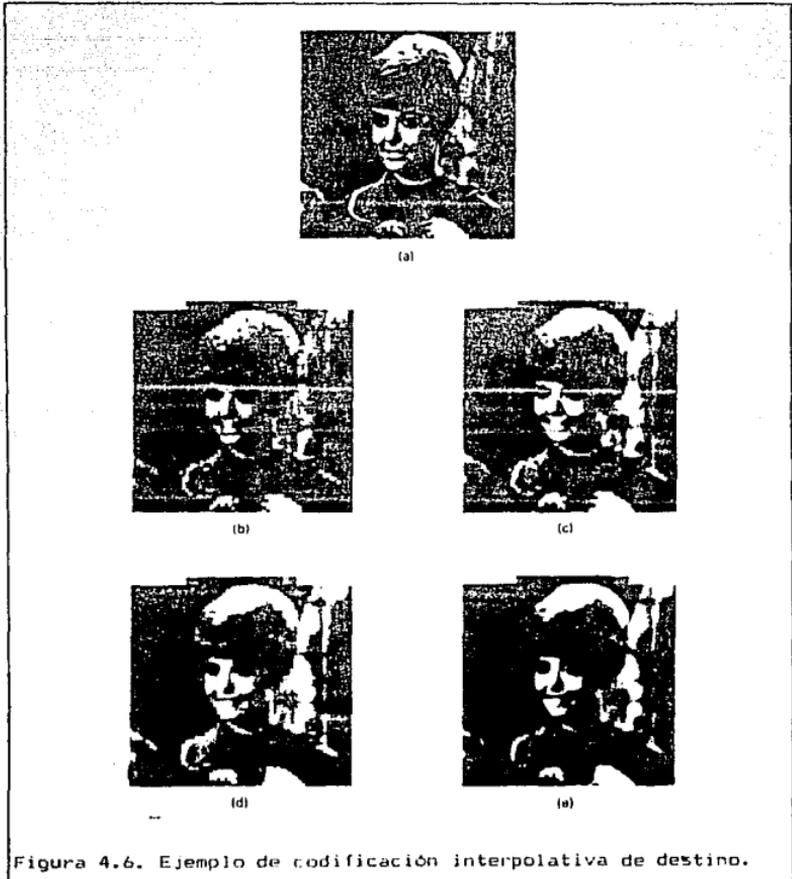
La figura 4.6 contiene varios ejemplos de simulación por computadora de codificación interpolativa de destino en las figuras 4.6b y 4.6c la imagen original de 256×256 pixeles ha sido submuestreada por supresión de muestra para producir un arreglo de 64×64 de muestras las cuales han sido interpoladas por una interpolación de orden cero y bilineal. En las figuras 4.6d y 4.6e son las mismas, excepto que el arreglo de pixeles de 64×64 submuestras ha sido obtenido por un promedio de los pixeles de la imagen original en bloques de 4×4 .

4.6 CODIFICACION DE IMAGEN POR TRANSFORMADA.

Esta codificación representa una desviación de las formas clásicas de codificación de imagen tales como PCM, predictiva e interpolativa en la cual la señal de imagen es directamente codificada. La codificación de imagen por transformada es un proceso indirecto. Una transformación matemática unitaria es ejecutada en los datos de la imagen para producir un número de

¹¹ El error de aliasing es un efecto óptico, conocido también como efecto de máre o modelo de máre. Para más detalle, vease sección 4.4.2, Digital Image Processing, William K. Pratt, Wiley Interscience.

coeficientes transformados, los cuales son cuantizados y codificados para su transmisión. Este sistema ha demostrado ser efectivo y práctico para medios de codificación de imágenes monocromáticas, de color y multiespectrales.



4.6.1 Codificación de imagen monocromática por transformada.

El concepto básico del proceso de codificación por transformada de Fourier es que para imágenes naturales los coeficientes transformados son de una magnitud relativamente baja. Esos coeficientes transformados a menudo pueden ser descartados enteramente, o codificados con pequeños números de código con sólo una ligera distorsión en la imagen.

En varias investigaciones se encontró que la transformada Hadamard podía utilizarse en lugar de la transformada de Fourier obteniéndose un decremento considerable en los requerimientos computacionales para muchas aplicaciones. Esto provocó que se investigara con la transformada discreta Karhunen-Loeve también conocida como transformada Hoetelling y la transformada Haar para la codificación de imágenes¹².

La transformada Karhunen-Loeve provee un mínimo error medio cuadrático al ejecutar la codificación, pero infortunadamente requiere de conocimientos estadísticos de la imagen original y no posee un rápido algoritmo computacional. A la transformada Haar se le atribuye una extremada eficiencia en el algoritmo computacional, pero usualmente resulta en un error relativo de codificación. En el capítulo 5, se mostrará esta característica.

La figura 4.7 contiene un diagrama a bloques de un sistema codificador por transformada para imágenes monocromáticas. En una transformada de 2 dimensiones se toman los píxeles de la imagen sobre la imagen total, o repetidamente sobre subsecciones de imagen llamados bloques. Se denota a $F(j, k)$ como un bloque de píxeles. Para la transformada unitaria de 2 dimensiones que es

¹² Las transformadas Hadamard, Fourier, Karhunen-Loeve y Haar serán desarrolladas en el Capítulo 5 de esta tesis.

ortogonalmente separable, los coeficientes transformados son dados por:

$$F'(u, v) = \sum_{j=1}^N \sum_{k=1}^N F(j, k) A_c(j, u) A_r(k, v) \quad (4.5)$$

donde $A_r(k, v)$ y $A_c(j, u)$ representan el renglón y la columna, respectivamente. Alternativamente, el bloque de pixeles puede ser descrito por la matriz F . Entonces la forma del vector espacio en la matriz de coeficientes transformados esta dado por:

$$F' = A_c F A_r \quad (4.6)$$

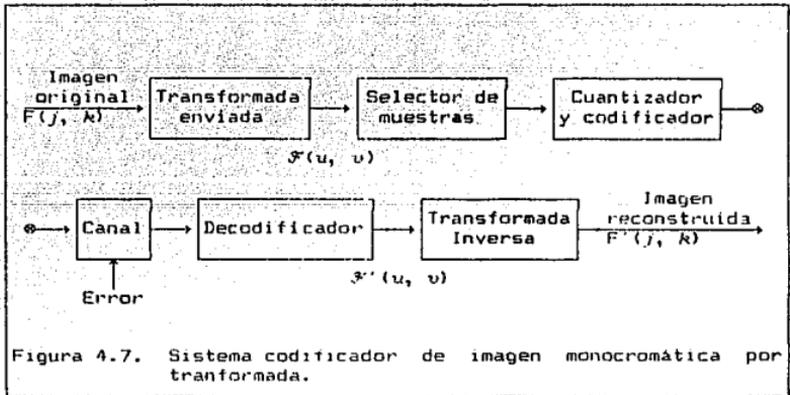


Figura 4.7. Sistema codificador de imagen monocromática por transformada.

Existen 2 estrategias básicas de selección de muestreo: muestreo por zona y muestreo de umbral. En el muestreo por zona la reconstrucción es hecha con una subpuesta de muestras transformadas en ciertas zonas geométricas preespecificadas, usualmente son los coeficientes de baja frecuencia. Para transmisión analógica la amplitud de cada componente en la zona es transmitido, mientras que para la transmisión digital cada componente en una zona es cuantizado y se le asigna un código

binario. El número de niveles de cuantización es usualmente proporcional a la varianza estimada del componente y el número de bits de código es hecho proporcional a la probabilidad esperada de ocurrencia.

Con el muestreo de umbral la reconstrucción de imagen se hace con una subpuesta de las muestras que son tan grandes como el umbral especificado, este muestreo es usualmente empleado en enlaces digitales.

4.6.1.1 Codificación por zona

En el sistema de codificación de transformación por zona la determinación de zona es establecida en cada bloque de transformada. Las muestras transformadas en cada zona son entonces cuantizadas con el mismo número de niveles de cuantización proporcional a la varianza esperada de los coeficientes transformados.

Para un código de longitud constante, $N(u, v)$ bits son asignados a cada coeficiente resultando

$$L(u, v) = 2 \exp [N(u, v)] \quad (4.7)$$

niveles de cuantización; un total de

$$N_B = \sum_u \sum_v N_B(u, v) \quad (4.8)$$

bits son requeridos para codificar la imagen.

La figura 4.8 ilustra una asignación típica de bits para la codificación en un bloque de 16×16 pixeles.

En la figura 4.9 se muestra el resultado de la simulación de la codificación por zona de imágenes en bloques de 16×16 para

varias transformadas.

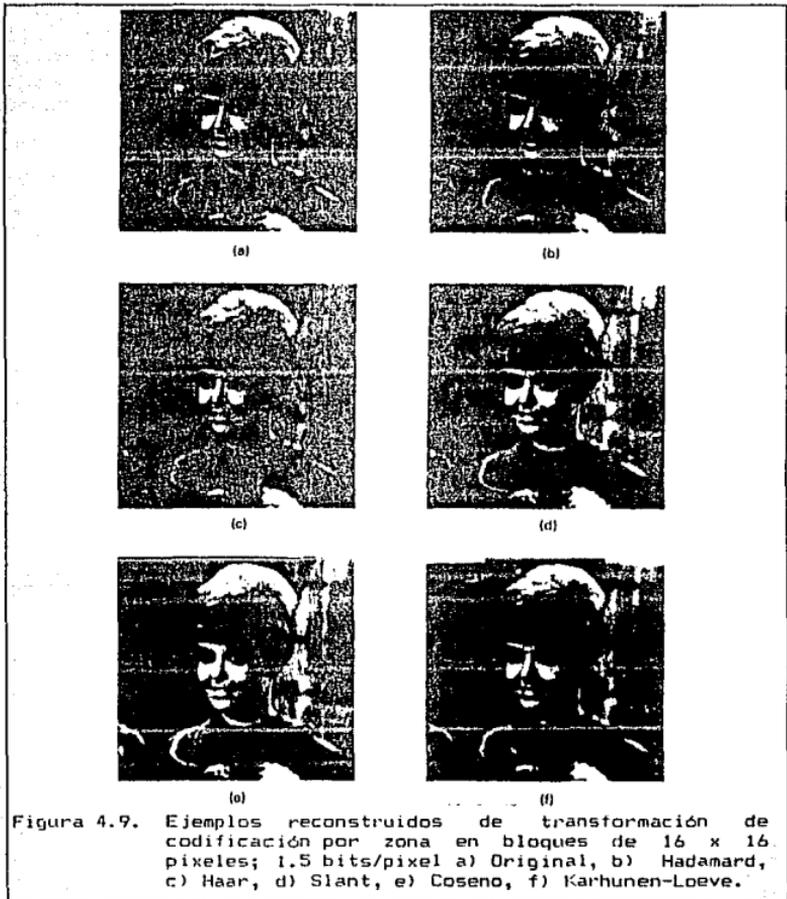
8	8	8	7	7	7	5	5	4	4	4	4	4	4	4	4
8	8	7	6	5	5	3	3	3	3	2	2	2	2	2	2
8	7	6	4	4	4	3	3	2	2	2	2	2	2	2	2
7	6	4	3	2	2	2	2	1	1	1	1	0	0	0	0
7	5	4	2	2	2	2	1	1	1	0	0	0	0	0	0
7	5	4	2	2	2	1	1	0	0	0	0	0	0	0	0
5	3	3	2	2	1	1	0	0	0	0	0	0	0	0	0
5	3	3	2	1	1	0	0	0	0	0	0	0	0	0	0
4	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0
4	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0
4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 4.8. Asignación típica de bits para codificación zonal en un bloque de 16×16 píxeles en una proporción de 1.5 bits por píxel.

4.6.1.2 Codificación de umbral

En un sistema de codificación de umbral cada muestra cuya magnitud es tan grande que un nivel de umbral es cuantizado con un número fijo de niveles y la amplitud es codificada; es necesaria la posición de código de cada muestra significativa en el plano de la transformación.

En la figura 4.10 se muestra los resultados de una simulación por codificación de umbral; debido a que el proceso de codificación es adaptivo, es mejor ejecutado que el proceso de codificación por zona. Con la codificación estandar de umbral el número de coeficientes y el número de bits codificados es dependiente de la imagen para su transmisión.



4.6.2 Codificación adaptiva por transformada.

Los codificadores de tipo adaptivo son en general bastante

complejos de implementar. El funcionamiento de los codificadores en general pueden ser mejorados substancialmente por el monitoreo de la actividad de la escala de grises o los cambios estadísticos dentro de la imagen.

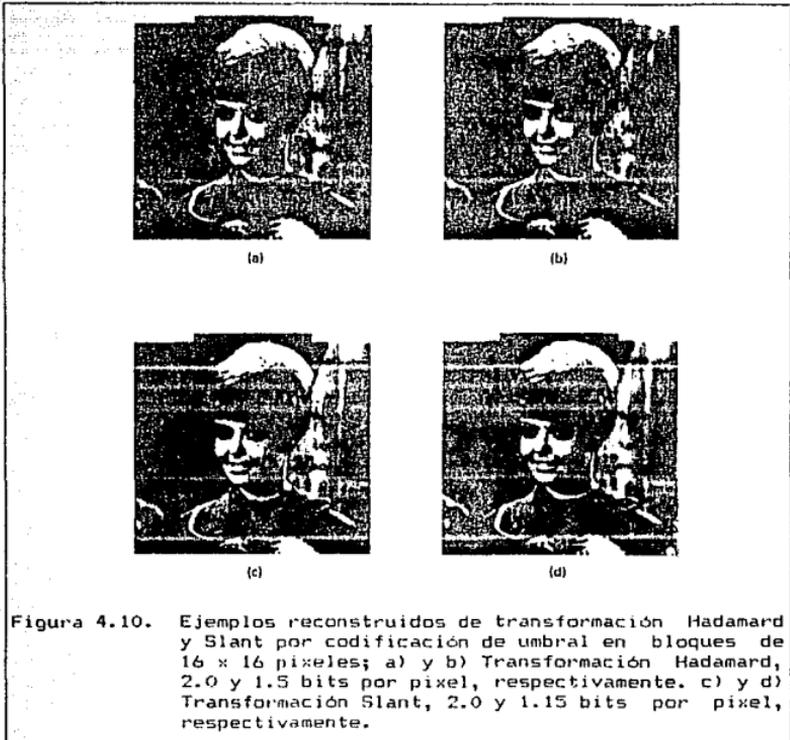


Figura 4.10. Ejemplos reconstruidos de transformación Hadamard y Slant por codificación de umbral en bloques de 16×16 píxeles; a) y b) Transformación Hadamard, 2.0 y 1.5 bits por píxel, respectivamente. c) y d) Transformación Slant, 2.0 y 1.15 bits por píxel, respectivamente.

Una de las propiedades deseadas de un transformador de imagen para los propósitos de codificación es la habilidad para producir una secuencia de coeficientes cercanos no correlacionados para un

campo de imagen altamente correlacionada.

Una técnica de cuantización adaptiva es la transformada de Fourier, en ella la magnitud y la fase de cada coeficiente son codificados adaptivamente. La densidad de probabilidad de la fase $\phi(u, v)$ es modelada como una densidad uniforme y la densidad de la magnitud $\mu(u, v)$ es modelada como una densidad de Rayleigh. El número de niveles de cuantización para los componentes de magnitud y fase son proporcionales a un factor de varianza con la restricción que el número de niveles de cuantización.

Esta técnica también se extiende a la codificación de coeficientes por transformada Hadamard; algunos autores han estudiado un sistema de codificación adaptiva por transformada en la cual los bloques de imagen se clasifican en tres categorías de acuerdo a la actividad de luminancia, estas son: a) Bloque con una suma significativa de detalles; b) Bloques luminosos en promedio con pequeños detalles; c) Bloques oscuros en promedio con pequeños detalles. Estos métodos adaptivos muestran un factor de compresión de 2:1 y se obtienen con un buen diseño de un codificador no adaptivo.

4.6.3 Codificación de imagen de color por transformada.

Para el sistema de codificación de color por transformada los tres colores primarios $R(j, k)$, $B(j, k)$, $G(j, k)$, rojo, azul, verde respectivamente son convertidos a tres señales $Y(j, k)$, $I(j, k)$, $Q(j, k)$ que especifican la información de luminancia y la crominancia respectivamente. La conversión esta definida por:

$$\begin{bmatrix} Y(j, k) \\ I(j, k) \\ Q(j, k) \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R(j, k) \\ G(j, k) \\ B(j, k) \end{bmatrix} \quad (4.9)$$

Esta conversión se realiza para el correcto diseño de los cuantizadores. Estas señales sufren una transformación de 2 dimensiones, resultando 3 transformadas, las cuales son:

$$Y = A \cdot Y \cdot A^T \quad (4.10a)$$

$$Z = A \cdot I \cdot A^T \quad (4.10b)$$

$$Q = A \cdot Q \cdot A^T \quad (4.10c)$$

donde A es la matriz transformada.

La figura 4.11 muestra el proceso de transformación de una imagen de color; después de la transformación de las muestras estas son cuantizadas con el número apropiado de niveles proporcionales a la variancia esperada de cada pixel y con el nivel de cuantización espaciado permitido para minimizar el error mínimo cuadrático. Las muestras cuantizadas $Y'(u, v)$, $Z'(u, v)$, $Q'(u, v)$ son entonces codificadas y transmitidas. En el receptor son decodificadas y transformadas inversamente para obtener las señales:

$$Y' = A^T Y' A \quad (4.11a)$$

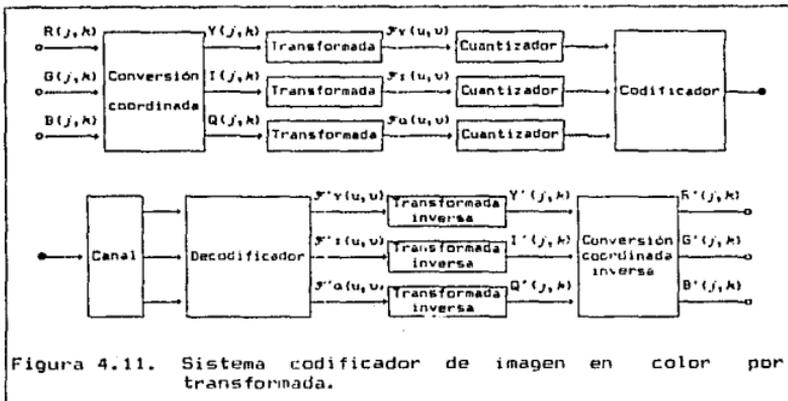


Figura 4.11. Sistema codificador de imagen en color por transformada.

$$I' = A^T \mathcal{J}' A \quad (4.11b)$$

$$Q' = A^T Q A \quad (4.11c)$$

Finalmente se reconstruyen las señales originales.

$$\begin{bmatrix} R'(j, k) \\ G'(j, k) \\ B'(j, k) \end{bmatrix} = \begin{bmatrix} 1.00 & 0.956 & 0.621 \\ 1.00 & -0.272 & -0.647 \\ 1.00 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y'(j, k) \\ I'(j, k) \\ Q'(j, k) \end{bmatrix} \quad (4.12)$$

Para la optimización del diseño de un decodificador de imagen de color es necesario especificar algunas mediciones analíticas de la fidelidad de la imagen en color, infortunadamente, no existen mediciones de este tipo; una alternativa es utilizar un procesamiento de cuantización para minimizar el error medio cuadrático entre las señales Y, I, Q y los planos de color Y', I', Q' quedando definido como:

$$\zeta = \frac{1}{3N^2} \sum_j \sum_k \left\{ [Y(j, k) - Y'(j, k)]^2 + [I(j, k) - I'(j, k)]^2 + [Q(j, k) - Q'(j, k)]^2 \right\}$$

La figura 4.12 muestra el resultado de una simulación por computadora de un sistema de codificación de imagen por transformación Slant utilizando una proporción de 2.0 bits/pixel. Las fotografías monocromáticas de los colores primarios R, G, B , y los componentes de color de imagen Y, I, Q son mostrados en las figuras 4.12a y 4.12b. En las figuras 4.12c y 4.12d muestran degradaciones debidas al proceso de codificación, pero el efecto visual en la reconstrucción es mucho menos visible en la imagen de color por las limitaciones del ojo humano.

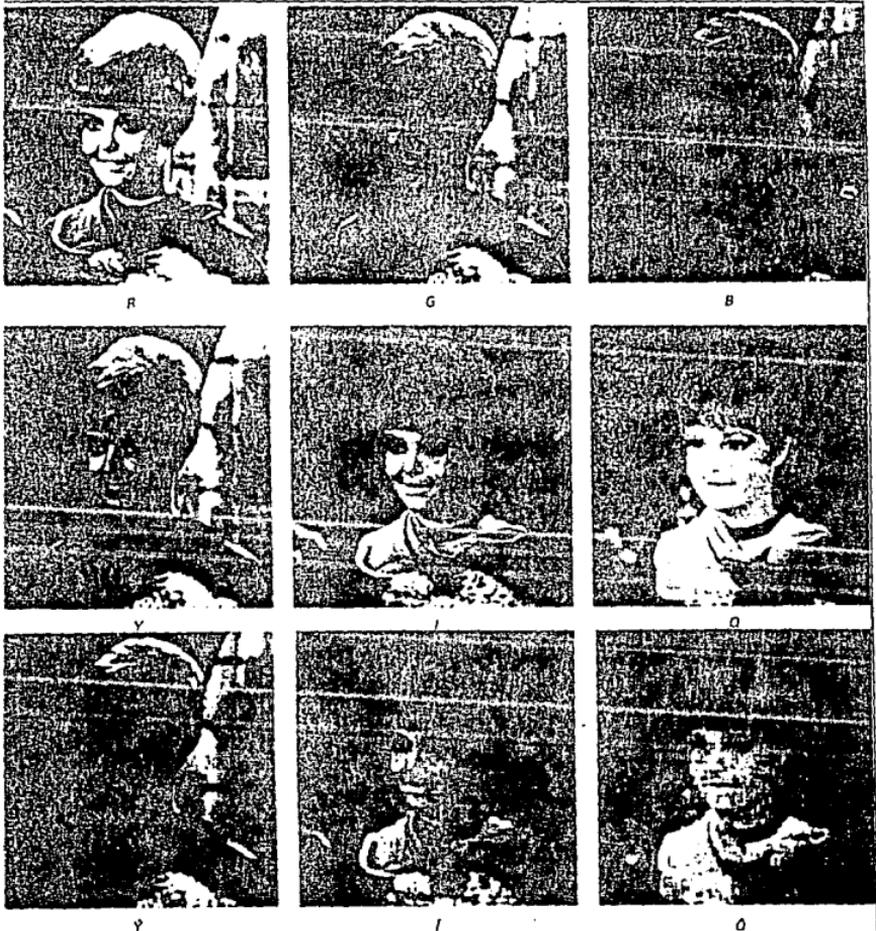


Figura 4.12. (a) Codificación de imágenes en color reconstruidas por transformación slant, 2.0 bits por píxel. (b) Codificación de imágenes en color reconstruidas por transformación Slant, 2.0 bits por píxel. (c) Muestra reconstruida posterior a su transformación.



Figura 4.12. Cont. (d) Muestra ^A reconstruida posterior ^B a su transformación.

**NO
EXISTE
PAGINA**

CAPITULO 5

TECNICAS PARA EL PROCESAMIENTO DIGITAL DE IMAGENES.

5.1 TRANSFORMACION DE IMAGENES.

5.1.1 TRANSFORMACION UNITARIA DE DOS DIMENSIONES.

5.1.2 TRANSFORMACION UNIDIMENSIONAL.

5.2 TRANSFORMADA DE FOURIER.

5.2.1 TRANSFORMADA RAPIDA DE FOURIER (FFT).

5.3 PROCESAMIENTO DE IMAGENES POR TRANSFORMADA.

5.3.1 TRANSFORMADA HAAR.

5.3.2 TRANSFORMADA HADAMARD.

5.3.3 TRANSFORMADA KARHUNEN-LOEVE.

5.3.4 TRANSFORMADA SLANT.

5.3.5 TRANSFORMADA COSENO.

5.3.5.1 TRANSFORMADA COSENO DE SIMETRIA
SEMEJANTE.5.3.5.2 TRANSFORMADA COSENO DE SIMETRIA
IMPAR.

5.3.6 TRANSFORMADA SENO.

5.3.7 TRANSFORMADA SVD (DESCOMPOSICION DE VALOR
SINGULAR).

5.4 COMPRESION DE IMAGENES.

5.4.1 COMPRESION DE DATOS VS. COMPRESION DE ANCHO DE
BANDA.

5.1 TRANSFORMACION DE IMAGENES.

La teoría de la transformación ha jugado un papel importante en el procesamiento digital de imágenes por varios años, y continúa siendo un tópico interesante tanto en teoría como en aplicaciones en este campo. La transformación de imágenes se ha desarrollado importantemente debido a los adelantos tanto en software con la tecnología de circuitos integrados de estado sólido, como en hardware con los algoritmos de convolución rápida.

Los sistemas digitales van adquiriendo mayor flexibilidad en el procesamiento de algoritmos, procesos iterativos, procesos de evaluación de requerimientos y en la toma de decisiones.

La transformada de dos dimensiones es usada para el mejoramiento, restauración, codificación y descripción de imágenes.

5.1.1 Transformación unitaria de dos dimensiones.

La transformación unitaria de dos dimensiones ha encontrado tres aplicaciones principales en el procesamiento de imágenes. Primeramente se ha utilizado para extraer la características principales de las imágenes.

Por ejemplo, en la transformada de Fourier el valor promedio es proporcional al promedio del brillo de la imagen y en términos de alta frecuencia da una indicación de la amplitud y una orientación del contorno dentro de una imagen. Otra aplicación es la codificación de imagen por transformada en la cual una reducción de ancho de banda es realizada por descartación o una cuantización en conjunto de baja magnitud de coeficientes transformados. La tercera aplicación es la reducción dimensional en computación.

5.1.2 Transformación unidimensional.

Cuando una señal unidimensional es representada por una serie ortogonal de funciones base, la imagen puede ampliarse en términos de un juego discreto de arreglos base llamados imágenes base. Estas imágenes base pueden ser generadas por matrices unitarias. Alternativamente, una matriz de imagen de $N \times N$ puede ser vista como un vector de $N^2 \times 1$. Una transformación de imagen provee un juego coordinado o vectores base para el espacio del vector.

Para una función continua, la serie ortogonal de expansión provee series de coeficientes los cuales pueden ser usados para cualquier proceso futuro de las funciones. Para una secuencia unidimensional $\{u(n), 0 \leq n \leq N-1\}$, representada como un vector u de tamaño N ; una transformación unitarias es descrita como

$$v = A u \quad \rightarrow \quad v(k) = \sum_{n=0}^{N-1} a(k, n) u(n), \quad 0 \leq k \leq N-1 \quad (5.1)$$

donde $A^{-1} = A^T$ (unitaria). Esta dado

$$u = A^T v \quad \rightarrow \quad u(n) = \sum_{k=0}^{N-1} v(k) a^*(k, n), \quad 0 \leq n \leq N-1 \quad (5.2)$$

Cabe aclarar que ambas ecuaciones son para el caso discreto. La ecuación 5.2 puede ser vista como una representación en serie de la secuencia $u(n)$. Las columnas de A^T , es decir, los vectores $ak^* = (a^*(k, n), 0 \leq n \leq N-1)^T$ son llamados los vectores base de A . La figura 5.1 muestra ejemplos de vectores base de varias transformadas ortogonales encontradas en el procesamiento de imágenes. La serie de coeficientes $v(k)$ da una representación de la secuencia original $u(n)$ y son beneficiosos en el filtrado, compresión de datos, y otros análisis.

5.2 TRANSFORMADA DE FOURIER.

La transformada discreta de Fourier de dos dimensiones de una

arreglo de imagen se define por una serie de la siguiente forma:

$$F(u, v) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=1}^{N-1} F(j, k) \exp \left[-\frac{2\pi i}{N} (uj + vk) \right] \quad (5.3)$$

donde $i = \sqrt{-1}$. La transformada inversa esta dada por:

$$F(j, k) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=1}^{N-1} F(u, v) \exp \left[\frac{2\pi i}{N} (uj + vk) \right] \quad (5.4)$$

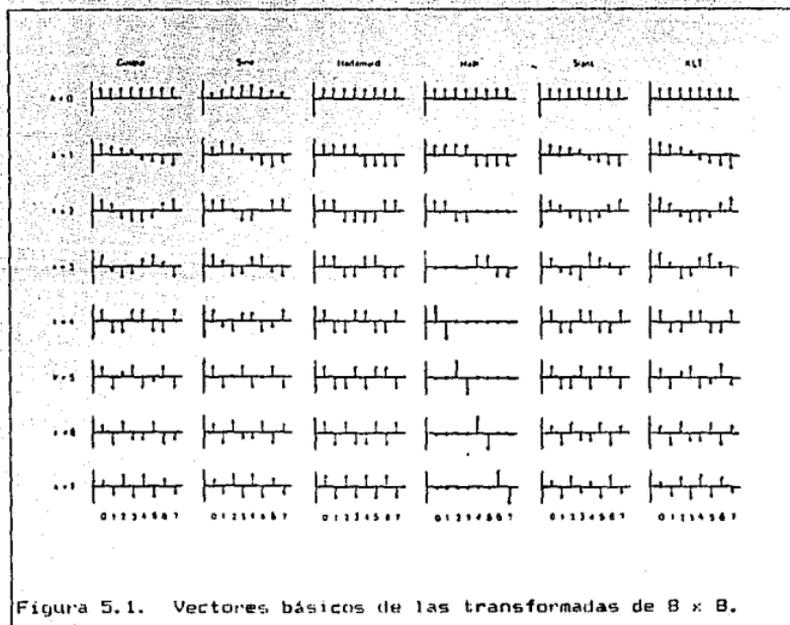


Figura 5.1. Vectores básicos de las transformadas de 8 x 8.

Los índices (u, v) son llamados, frecuencias espaciales de la transformación, en analogía con la transformada continua de Fourier. La transformada de dos dimensiones puede ser calculada como una secuencia de renglones y columnas de una

transformada unidimensional. Las funciones base de la transformada son de exponenciales complejos que pueden ser descompuestos en componentes seno y coseno. Esto es,

$$\begin{aligned} A(j, k; u, v) &= \exp \left[\frac{-2\pi i}{N} (uj + vk) \right] \\ &= \cos \left[\frac{2\pi i}{N} (uj + vk) \right] - i \sin \left[\frac{2\pi i}{N} (uj + vk) \right] \quad (5.5) \end{aligned}$$

$$\begin{aligned} B(j, k; u, v) &= \exp \left[\frac{2\pi i}{N} (uj + vk) \right] \\ &= \cos \left[\frac{2\pi i}{N} (uj + vk) \right] + i \sin \left[\frac{2\pi i}{N} (uj + vk) \right] \quad (5.6) \end{aligned}$$

La transformada de Fourier de dos dimensiones de una imagen es esencialmente una representación de la serie de Fourier de un área de dos dimensiones. Para que la representación de la serie de Fourier sea válida, el área debe ser periódica.

Si el arreglo de imagen representa una área de luminancia, $F(j, k)$ sería una función positiva real. Sin embargo, la transformada de Fourier es, en forma general, compleja. La figura 5.2 contiene fotografías de una imagen y varias versiones de la transformada de Fourier.

5.2.1 Transformada rápida de Fourier (FFT)

Considerando un arreglo bidimensional de valores de imagen $\beta_{m,n}$ obtenidas por muestreo, donde $m = 0, 1, 2, \dots, M-1$ y $n = 0, 1, 2, 3, \dots, N-1$. Cada $\beta_{m,n}$ es proporcional al brillo de la imagen muestreada, su posición esta dada por m y n .

La transformada discreta de Fourier de la función discreta $\beta_{m,n}$ esta dada por:

$$F_{k,l} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \beta_{m,n} \exp \left[-j2\pi \left(\frac{km}{M} + \frac{ln}{N} \right) \right] \quad (5.7)$$

donde $k = 0, 1, \dots, M-1$ e $i = 0, 1, \dots, N-1$.

La inversa es:

$$\beta_{m,n} = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{i=0}^{N-1} F_{k,i} \exp \left[j2\pi \left(\frac{km}{M} + \frac{in}{N} \right) \right] \quad (5.8)$$

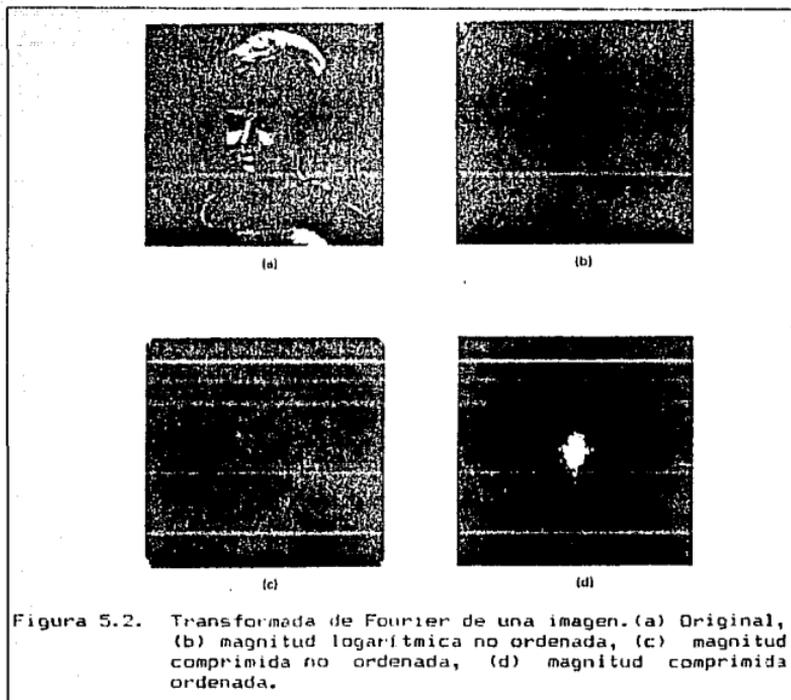


Figura 5.2. Transformada de Fourier de una imagen. (a) Original, (b) magnitud logarítmica no ordenada, (c) magnitud comprimida no ordenada, (d) magnitud comprimida ordenada.

A fin de representar una matriz de píxeles, con una subposición de tamaño $L \leq MN$, de exponenciales complejos de la transformada discreta de Fourier, se escogen L exponenciales con

lo que se alarga la magnitud de los coeficientes, minimizándose el mínimo error cuadrático.

El método usado es dividir la imagen dentro de sub-bloques rectangulares. Cada sub-bloque se expande en una transformada discreta de Fourier y los coeficientes son subejecutados para un umbral adaptivo.

Una cantidad L es determinada para cada sub-bloque tal que sea proporcional a la parte entera de la desviación estandar σ de los sub-bloques de los píxeles;

$$\sigma = \left[\frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=1}^{N-1} (f_{m,n} - a)^2 \right]^{1/2} \quad (5.9)$$

Las frecuencias y amplitudes complejas de los L coeficientes de Fourier de mayor valor son transmitidos; la cuantización lineal adaptiva de los coeficientes de Fourier son usados como precedente para su transmisión.

El máximo coeficiente de Fourier al lado de $F_{0,0}$, determina el valor del nivel de cuantización máximo dentro de cada sub-bloque. Las fases de los coeficientes complejos de Fourier son cuantizados en una escala de 0 a 2π ; el número de bits usados para cuantizarlos se determinan por α .

También se usan 2, 3, 4 o 5 bits para cuantizar todas las fases y amplitudes muestreadas dentro de cada sub-bloque. $F_{0,0}$ es finalmente cuantizado previo a su transmisión, independientemente del valor de α . Los coeficientes no transmitidos son puestos cero en el decodificador del receptor.

Son varias las cosas para considerar el tamaño del sub-bloque, entre ellas el valor de σ que se incrementa con el área del sub-bloque, porque la correlación entre los puntos de imagen generalmente decrece conforme la distancia entre los

pixeles aumenta. En un sub-bloque de mayor tamaño es muy probable que las áreas de pequeña variación en el brillo sean incluidas con áreas de gran variación. En áreas de poca variación es probable que se den pérdidas, por la presencia del mismo sub-bloque de las áreas de gran variación.

El tamaño del sub-bloque escogido por la implementación hardware es de 16×16 . En la implementación de este sistema las magnitudes y las muestras de fase dentro de un sub-bloque son linealmente cuantizadas para 2, 3, 4 o 5 bits dependiendo del valor de α .

5.3 PROCESAMIENTO DE IMAGENES POR TRANSFORMADA.

Entre las transformadas utilizadas para el procesamiento digital de imágenes se encuentra principalmente la transformada de Fourier y sus derivaciones, que tienen un amplio rango de aplicación en problemas de procesamiento de imágenes; de las cuales ya se habló en el punto anterior. Con la utilización de otras formas de transformación se han obtenido mejoras al ser aplicadas en el procesamiento de imágenes.

Con la transformada Haar, Slant, Hadamard, Karhunen-Loeve, Coseno, Seno, SVD (Descomposición de Valor Singular), se han obtenido adelantos en su aplicación en casos específicos, porque se han creado algoritmos de rápidos resultados, reducido el número de cálculos y consecuentemente los métodos de implementación.

5.3.1 Transformada Haar

Haar propuso una transformada simple compuesta de unos, menos unos y ceros. La matriz Haar de 4×4 esta dada por:

$$H_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.10)$$

La de matriz de $B \times B$ queda como sigue:

$$H_8 = \frac{1}{\sqrt{B}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix} \quad (5.11)$$

Esta transformada, ha sido implementada para procesos de muestreo en el cual los renglones de la matriz transformada muestra la secuencia de datos de entrada con una fina resolución incrementándose en potencia de 2. Sin embargo, la calidad de la imagen resultante es mala respecto a otras transformadas por lo que no es muy usada. La figura 5.3 muestra la Transformada Haar de una imagen.

5.3.2 Transformada Hadamard

La transformada Hadamard se basa en la matriz Hadamard la cual es un arreglo de mas y menos unos, cuyos renglones y columnas son ortogonales; la matriz Hadamard satisface la relación

$$HH^T = I \quad (5.12)$$

La matriz Hadamard de 2×2 está dada por:

$$H_2 = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} \quad (5.13)$$



Esta matriz no se restringe a 2 y es extendible a cualquier valor de $N = 2^n$ donde n es un entero, en este caso si H_N es una matriz de tamaño N la matriz es:

$$H_2^N = \frac{1}{\sqrt{2}} \begin{vmatrix} H_N & H_N \\ \dots & \dots \\ H_N & -H_N \end{vmatrix} \quad (5.14)$$

Esta es una matriz de orden $2N$; H. Harmuth en su artículo "transmission of information by orthogonal functions" sugirió que es posible construir una matriz Hadamard de orden $N = 2^n$ en la cual el número de signos por renglón se incrementará de 0 hasta $N - 1$; los renglones pueden ser considerados muestras de ondas rectangulares con subperíodo de $1/N$ unidades; estas son llamadas funciones Walsh. Es por esto, que esta transformada es también llamada Walsh o Walsh-Hadamard.

Para matrices simétricas de orden $N = 2^n$ una transformada Hadamard de 2 dimensiones se puede escribir en series de la siguiente forma:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x, y) (-1)^{p(x, y, u, v)} \quad (5.15)$$

donde

$$p(x, y, u, v) = \sum_{i=0}^{N-1} (x_i y_i + u_i v_i) \quad (5.16)$$

Los términos u_i , x_i , v_i y y_i son representaciones binarias de u , x , v y y , respectivamente. La figura 5.4 presenta la transformada Hadamard de una imagen.

5.3.3 Transformada Karhunen-Loeve.

Las técnicas para la transformación de señales continuas, fué originalmente desarrollada por H. Karhunen y M. Loeve; a H. Hotelling se le acreditan los procesos de conversión de transformación de las señales discretas. Sin embargo, a Karhunen-Loeve se les reconoce ambas transformaciones.

La Transformada Karhunen-Loeve es de forma general:

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) A(x, y; u, v) \quad (5.17)$$

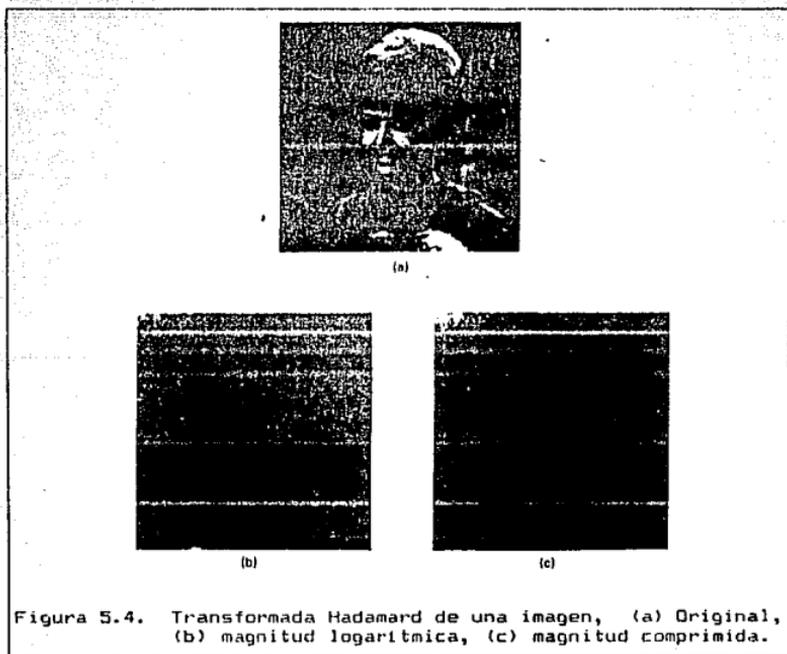


Figura 5.4. Transformada Hadamard de una imagen, (a) Original, (b) magnitud logarítmica, (c) magnitud comprimida.

para la cual $A(x, y; u, v)$ satisface la ecuación:

$$\lambda(u, v) A(x, y; u, v) = \sum_{x'=0}^{N-1} \sum_{y'=0}^{N-1} Kf(x, y; x', y') A(x, y; u, v) \quad (5.18)$$

donde $Kf(x, y; x', y')$ puede separarse en renglones y columnas

resultando:

$$Kf = (x_1, y_1; x', y') = Kc (x_1, x') K_R (y_1, y') \quad (5.19)$$

de (5.18) resulta:

$$A(x_1, y_1; u, v) = A_c(x_1, u) A_R(y_1, v) \quad (5.20)$$

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) A_c(x, u) A_R(y, v) \quad (5.21)$$

que es la transformada Karhunen-Loeve en forma de serie.

5.3.4 Transformada Slant.

La transformada Slant fué diseñada específicamente para codificación de imagen; utiliza un algoritmo computacional rápido, lo que provoca que se obtenga una mayor reproducción de calidad de imagen. Por sus características de funcionamiento es un poco menos óptimo que la transformada Karhunen-Loeve, pero más compleja que la transformada rápida de Fourier. Para un vector de longitud $N = 2$ la transformada Slant es idéntica a la transformada Hadamard de orden 2, es decir:

$$S_2 = \frac{1}{\sqrt{2}} \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} \quad (5.22)$$

La Transformada Slant de orden 4 es obtenida por la operación:

$$S_4 = \frac{1}{2^{1/2}} \begin{vmatrix} 1 & 0 & 1 & 0 \\ a_4 & b_4 & -a_4 & -b_4 \\ 0 & 1 & 0 & -1 \\ -b_4 & a_4 & b_4 & a_4 \end{vmatrix} \begin{vmatrix} S_2 & 0 \\ 0 & S_2 \end{vmatrix} \quad (5.23)$$

ó bien,

$$S_4 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ a_4 + b_4 & a_4 - b_4 & -a_4 + b_4 & -a_4 - b_4 \\ 1 & -1 & 1 & 1 \\ a_4 - b_4 & -a_4 - b_4 & a_4 + b_4 & -a_4 + b_4 \end{pmatrix} \quad (5.23a)$$

donde a_4 y b_4 son constantes de escala real cuyos valores se determinan de acuerdo con las condiciones que hacen a S_4 ortogonal, se ha encontrado que este valor es $a_4 = 2b_4$ y la condición de ortogonalidad dice que $S_4 S_4^t = I$ lleva a encontrar $b_4 = 1 / \sqrt{5}$.

La matriz general para N términos, esta dada por:

$$S_N = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ a_N & b_N & & -a_N & b_N & \\ 0 & 1 & I_{(N/2)-2} & 0 & I_{(N-2)-2} & \\ 0 & 1 & 0 & 0 & -1 & 0 \\ -b_N & a_N & & b_N & a_N & \\ 0 & 0 & I_{(N/2)-2} & 0 & -I_{(N-2)-2} & \end{pmatrix} \begin{pmatrix} S_{N/2} & 0 \\ 0 & S_{N/2} \end{pmatrix} \quad (5.24)$$

donde $I_{(N/2)-2}$ es la matriz identidad de dimensión $(N/2)-2$. Las constantes a_N y b_N pueden ser encontradas por las siguientes ecuaciones:

$$a_2 = 1 \quad (5.25)$$

$$b_N = [1 + 4(a_{N/2})^2]^{1/2} \quad (5.26)$$

$$a_N = 2b_N a_{N/2} \quad (5.27)$$

$$a_{2N} = [(3N^2) / (4N^2 - 1)]^{1/2} \quad (5.28)$$

$$b_{2N} = [(N^2 - 1) / (4N^2 - 1)]^{1/2} \quad (5.29)$$

La transformada Slant requiere de un total de $N \log_2 N + (N/2)-2$ adiciones y subtracciones más $2N - 4$ multiplicaciones para un vector con una dimensión de datos N .

Fotografías de la transformada Slant de una imagen se muestran en la figura 5.5.

5.3.5 Transformada Coseno

Es conocido que en las series de Fourier para cualquier función real continua y simétrica, contienen únicamente coeficientes reales que corresponden a los términos coseno de dichas series.

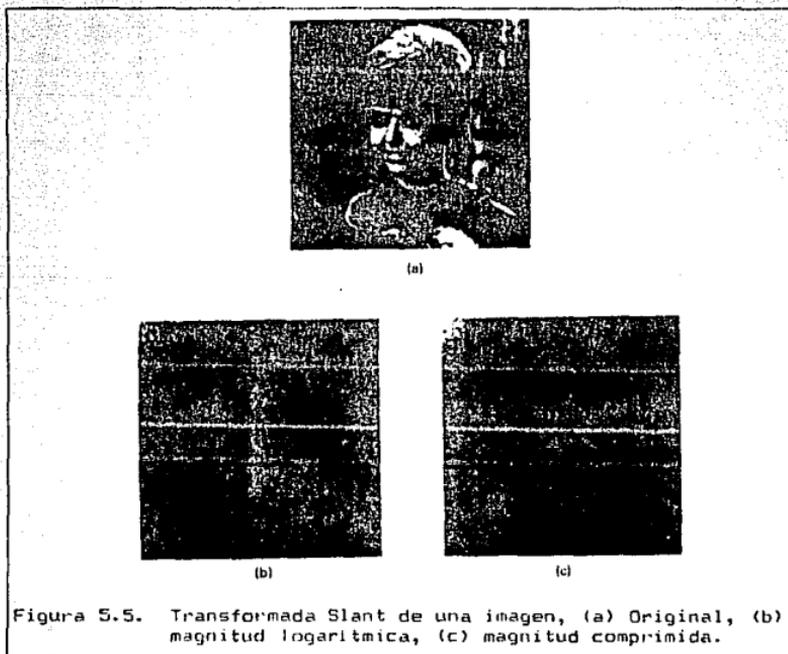
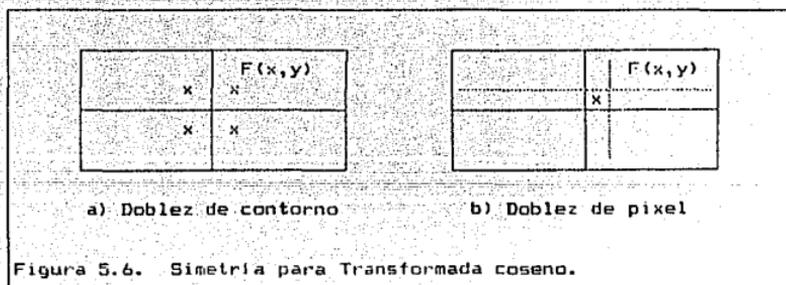


Figura 5.5. Transformada Slant de una imagen, (a) Original, (b) magnitud logaritmica, (c) magnitud comprimida.

Esto puede entenderse a la transformada de Fourier bajo su

propia interpretación. Existen dos caminos en los cuales una imagen puede hacerse simétrica, como se muestra en la figura 5.6. Por la primer técnica las imágenes son dobladas hacia un contorno y en el segundo método las imágenes son dobladas y sobrepuestas por un pixel:

De este modo, para una imagen de $N \times N$ pixeles, el primer método es llamado Transformada Coseno de Simetría Semejante, resultando un arreglo de $2N \times 2N$ pixeles; el segundo método llamado de Transformada Coseno de Simetría Impar produce un arreglo de $2N - 1 \times 2N - 1$ pixeles.



5.3.5.1 Transformada coseno de simetría semejante.

Si se tiene un arreglo de imagen simétrica será formada por reflexión del arreglo de imagen hacia los contornos de acuerdo a la relación:

$$F_s(x,y) = \begin{cases} F(x, y) & x \geq 0, y \geq 0 \\ F(-x, y) & x < 0, y \geq 0 \\ F(x, -y) & x \geq 0, y < 0 \\ F(-x, -y) & x < 0, y < 0 \end{cases} \quad (5.30)$$

Para esta construcción $F_s(x,y)$ es simétrica hacia el punto

$x = -1/2$, $y = -1/2$ y tomando la transformada de Fourier hacia el punto de simetría resulta:

$$\mathcal{F}_s(u, v) = \frac{1}{2N} \sum_{x=-N}^{N-1} \sum_{y=-N}^{N-1} F_s(x, y) \exp \left\{ \frac{-2\pi i}{2N} \left[u \left(x + \frac{1}{2} \right) + v \left(y + \frac{1}{2} \right) \right] \right\} \quad (5.31)$$

para $u, v = -N, \dots, -1, 0, 1, \dots, N-1$; debido a que $F_s(x, y)$ es real y simétrica, la ecuación (5.31) se reduce a:

$$\mathcal{F}_s(u, v) = \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x, y) \cos \left[\frac{\pi}{N} u \left(x + \frac{1}{2} \right) \right] \cos \left[\frac{\pi}{N} v \left(y + \frac{1}{2} \right) \right] \quad (5.32)$$

De manera alternativa, los términos de las ecuaciones (5.31) y (5.32) pueden ser obtenidas por una transformada de Fourier de $F(x, y)$ sobre $2N$ puntos obteniéndose:

$$\mathcal{F}_s(u, v) = \frac{2}{N} \operatorname{Re} \left\{ \exp \left[\frac{i\pi u}{2N} \right] \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x, y) \exp \left[\frac{-2\pi i}{2N} (ux + vy) \right] \right\} \quad (5.33)$$

5.3.5.2 Transformada coseno de simetría impar

Para la transformada coseno de simetría impar, el arreglo de imagen queda definida de la siguiente forma:

$$F_s(x, y) = \begin{cases} F(x, y) & x, y \geq 0 \\ F(-x, y) & x < 0, y \geq 0 \\ F(x, -y) & x \geq 0, y < 0 \\ F(-x, -y) & x, y < 0 \end{cases} \quad (5.34)$$

Tomando en cuenta la transformada de Fourier se genera el arreglo:

$$F_u(u, v) = \frac{1}{2N-1} \sum_{x=-N+1}^{N-1} \sum_{y=-N+1}^{N-1} F(x, y) \exp \left[-\frac{2\pi i}{2N-1} (x u, y v) \right] \quad (5.35)$$

para $u, v = N+1, \dots, -1, 0, 1, \dots, N-1$.

5.3.6. Transformada seno

La transformada seno, fué introducida como un algoritmo rápido para substituir a la transformada Karhuen-Loeve.

Esta transformada es definida como:

$$F(u, v) = \frac{1}{N+1} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x, y) \sin \left[\frac{(x+1)(u+1)}{N+1} \pi \right] \sin \left[\frac{(y+1)(v+1)}{N+1} \pi \right] \quad (5.36)$$

su inverso es de forma similar. La transformada seno puede también ser calculada con un algoritmo de la transformada de Fourier. La figura 5.7 muestra las fotografías de la transformada seno de una imagen.

5.3.7 Transformada SVD (Descomposición de Valor Singular)

La transformada SVD, esta basada en la técnica de descomposición de valores simples o singulares. Está elaborada para uso de técnicas de análisis numérico; si se utiliza en combinación con técnicas pseudo-inversas representa un intento de ampliar resultados desde el área de análisis numérico para modelos de sistemas de imagen lineal. Se encuentra definida por:

$$F = A_c F A_r^T \quad (5.37)$$

y la inversa es

$$F = A_c^T F A_r \quad (5.38)$$



El renglón de la matriz de transformación ejecuta la operación de diagonalización resultando,

$$A_R [F^T F] A_R^T = \Delta \quad (5.39)$$

donde Δ es una matriz diagonal cuyos términos $\lambda(u)$ son los valores de $F^T F$. De forma similar se obtiene la columna:

$$A_C [F F^T] A_C^T = \Delta \quad (5.40)$$

Substituyendo (5.38) en (5.39) y (5.40) indican que

$$F = \Delta^{1/2} \quad (5.41)$$

donde $\Delta^{1/2}$ es una matriz diagonal consistente en términos $\lambda^{1/2}(u)$ llamados valores singulares de F.

La matriz imagen F puede ser expresada en términos compactos de una serie como:

$$F = \sum_{u=1}^N F(u, u) bc(u) br^T(u) \quad (5.42)$$

donde $bc(u)$ y $br(u)$ representan las u-ésimas columnas y renglones de las matrices Bc^T y Br^T .

Con la transformada SVD, la imagen F de $N \times N$ pixeles es representada por sólo N componentes. La figura 5.8 muestra ejemplos de la transformada SVD de una imagen con los productos de las matrices FFT y F^TF .

5.4 COMPRESION DE IMAGENES.

Con el avance tecnológico en las comunicaciones, la demanda para el almacenamiento y transmisión de imágenes se ha ido incrementando. La tecnología computacional ha elaborado procesos de almacenamiento y digitación que han preparado el camino para la implementación de técnicas de compresión de datos para mejorar la eficiencia del almacenamiento y transmisión de imágenes.

La compresión de datos de la imagen es considerada como la minimización de la información de unidades acarreadas usadas para representar una imagen. Por la simplicidad y dinámica de la compresión de datos, es el muestreo de banda limitada de imágenes en donde un número infinito de puntos por área de muestreo unitario es reducido a una muestra sencilla de imagen fuera de cualquier pérdida de información (asumiendo un filtro paso bajas ideal). Consecuentemente el número de muestras por área unitaria es infinitamente reducido.

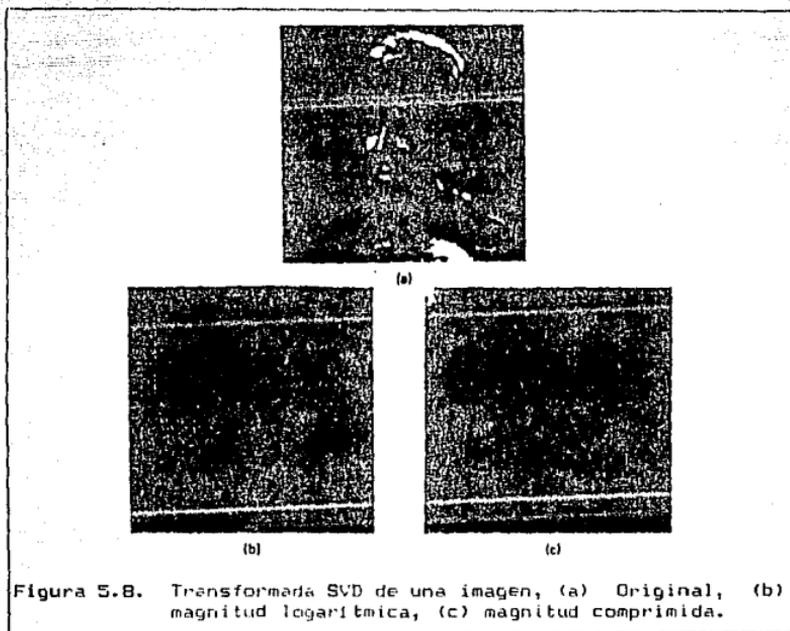


Figura 5.8. Transformada SVD de una imagen, (a) Original, (b) magnitud logarítmica, (c) magnitud comprimida.

La compresión de datos ofrece un almacenamiento y transmisión de imagen, en donde el objetivo es la minimización de la memoria para el almacenamiento y ancho de banda para la transmisión. Típicamente, una imagen al ser comprimida y decodificada para reconstruir la imagen original es acompañada por alguna distorsión.

La eficiencia de un algoritmo de compresión depende de la habilitación de la compresión de datos, del resultado de la distorsión y también por la complejidad de su implementación. La complejidad del algoritmo es importante considerando el hardware que es utilizado.

5.4.1 Compresión de datos vs. Compresión de ancho de banda.

El sólo proceso de conversión de una señal analógica de video en una señal digital resulta en un incremento de los requerimientos de ancho de banda para la transmisión. Así, el poco tiempo de digitalización de la información ha tenido ventajas con la forma analógica, en términos de la flexibilidad del procesamiento, facilidad de acceso en el almacenamiento, el alto grado de ruido, posibilidad de errores en la transmisión, etc.

Las técnicas de compresión de datos buscan la minimización en el costo de la digitalización de la información y a la vez por la flexibilidad del procesamiento reducir el ancho de banda de la señal digital y análogamente los requerimientos en el ancho de banda.

Los métodos de compresión de datos de la imagen son clasificados en dos categorías; la primera se refiere a la explotación de la redundancia en el dato, esta redundancia esta en relación con la predicción, casualidad, facilidad, etc. En la segunda, la compresión es obtenida por la transformación preservativa de energía de la imagen dentro de otro arreglo, tal que la información es empaquetada en un mínimo número de muestras.

**NO
EXISTE
PAGINA**

C A P I T U L O 6

CONCLUSIONES Y EJEMPLOS DEL PROCESAMIENTO DIGITAL DE IMAGENES.

- 6.1 RESTRICCIONES PARA LA CONVERSION A/D.
 - 6.1.1 VELOCIDAD DE MUESTREO.
 - 6.1.2 IMAGENES MONOCROMATICAS Y.
 - 6.2.3 IMAGENES DE COLOR.
 - 6.2.3.1 SENAL COMPUESTA.
 - 6.2.3.2 SENAL POR COMPONENTE.
- 6.2 CUANTIZACION Y CODIFICACION.
- 6.3 ALGUNOS EJEMPLOS DE IMAGENES CUANTIZADAS Y CODIFICADAS A 8 BITS 256 NIVELES.
 - 6.3.1 ALGORITMOS.
- 6.4 TECNICAS DE COMPRESION DE IMAGENES.
 - 6.4.1 TRANSFORMADA RAPIDA DE FOURIER (FFT).
 - 6.4.2 TRANSFORMADA COSENO.
- 6.5 APLICACIONES.

6.1 RESTRICCIONES PARA LA CONVERSION A/D.

6.1.1 Velocidad de muestreo.

En el capítulo 2, se mencionaron las características del muestreo, se demostró que la frecuencia de muestreo es $f_c \geq 2 AB$. Para el caso de señales analógicas de video las cuales contienen un rango de frecuencia de hasta 6 MHz para la señal compuesta, al muestrear una señal de este tipo se tendrían frecuencias de muestreo superiores a 122 MHz lo cual implica un proceso más complejo, por lo que es necesario que las señales compuestas de video sean menores a 6 MHz. Para lograr esto, es necesario un proceso de filtrado paso bajas para evitar frecuencias muy elevadas, al limitar las frecuencias altas, se elimina contraste a cualquier tipo de imagen.

Las características del muestreador deben ser como se indicó en el capítulo 2. Aún cuando la frecuencia de muestreo es elevada, puede ser manipulada por una simple PC con velocidad de procesamiento alto, sin embargo esto proporcionará una imagen de calidad media, puesto que las componentes de altas frecuencias son eliminadas.

6.1.2 Imágenes monocromáticas.

Las imágenes monocromáticas contienen escalas de gris equivalentes a una imagen de color, estas escalas de gris dependen de la señal de Luminancia (Y) cuya proporción se rige por la siguiente ecuación de porcentaje:

$$Y = 0.30 R + 0.59 V + 0.11 A \quad (6.1)$$

Por ejemplo, para una señal de barras de color (patrón) los porcentajes son los siguientes:

$$\text{BLANCO} \quad Y = 0.30 R + 0.59 V + 0.11 A = 1.00 \quad (6.2)$$

es decir, el color blanco contiene los tres colores primarios.

Para el caso de la señal compuesta de video, este se ubica en la parte inferior del pulso de sincronía horizontal como se muestra en la figura 6.1).

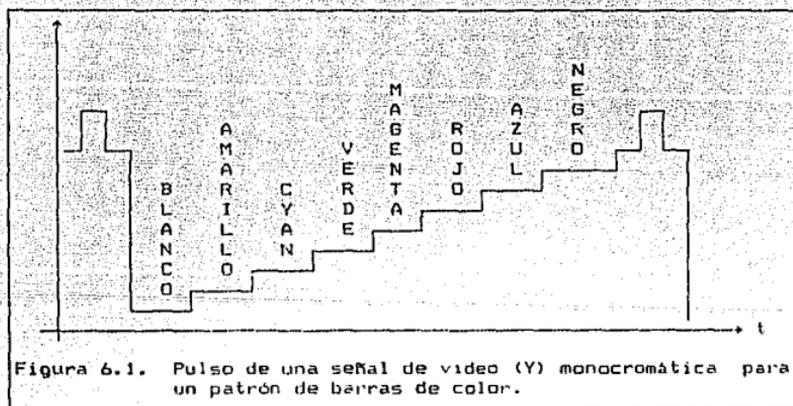


Figura 6.1. Pulso de una señal de video (Y) monocromática para un patrón de barras de color.

Unicamente que para el caso de la señal referida la banda blanca se encuentra en el nivel más bajo de voltaje, esto es porque las señales de video tienen la característica de transmisión negativa, es decir, que el color negro se ubica en el voltaje más alto y el opuesto que es el color blanco en el bajo, resultando una señal como la que se muestra en la figura 6.1. En la misma figura se muestran las 8 bandas de color para la señal Y en porcentajes de la escala de gris. Los porcentajes para las bandas de color resultan:

$$\text{BLANCO} \quad Y = 1.00 \quad (6.3)$$

AMARILLO	$Y = 0.89$	(6.4)
CYAN	$Y = 0.70$	(6.5)
VERDE	$Y = 0.59$	(6.6)
MAGENTA	$Y = 0.41$	(6.7)
ROJO	$Y = 0.30$	(6.8)
AZUL	$Y = 0.11$	(6.9)
NEGRO	$Y = 0.00$	(6.10)

Si se hace referencia a la escala de gris, en el caso de una señal patrón se tendrían 8 niveles de gris y sólo con 3 bits sería suficiente para representar dicha escala. Una imagen no contiene solamente 8 niveles, sino que contiene más niveles como se mencionó en el capítulo 1, de los cuales el ojo humano puede distinguir 40, esto implica que de toda la señal (figura 6.1) se dividía en el nivel máximo (100%).

Por esta situación, las imágenes monocromáticas tienen un espectro en frecuencia menos denso, es decir, se tienen un ancho de banda menor a los 64 MHz, esto implica una frecuencia de muestreo más baja. El inconveniente de esto, es en la resolución de la imagen, puesto que sólo contendrá la escala de gris desde el blanco hasta el negro, omitiéndose los detalles de color.

Este tipo de codificación es llamado codificación por componente, el cual muestra solamente la señal de luminancia y procesa sólo esa señal sin tomar en cuenta los detalles de color. Este tipo de codificación ahorra velocidad en el muestreador, también ahorra el número de bits por muestra puesto que en estas señales sólo se codifican la escala de gris por el pixel, aumentando con esto el contenido del archivo de datos de imagen considerablemente de la señal Y.

Un ejemplo de aplicación de digitalización de imágenes monocromáticas es el video-telefono en donde se codifican las imágenes del transmisor y receptor a través de estas

características. Otras aplicaciones son las video-conferencias, la transmisión de fotografías de periódicos y revistas impresas, si se compara una señal de voz con una digitalizada se notaría una gran diferencia, debido a que en la primera se requiere de una velocidad de aproximadamente 64 Kb/seg. y con la otra señal se requiere de aproximadamente 100 Mb/seg.

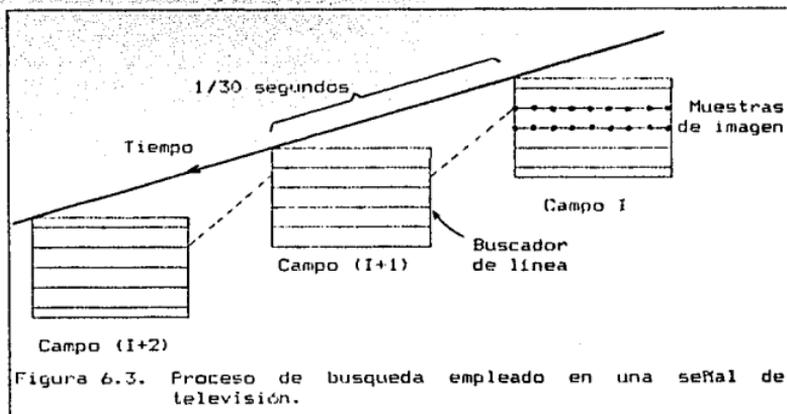
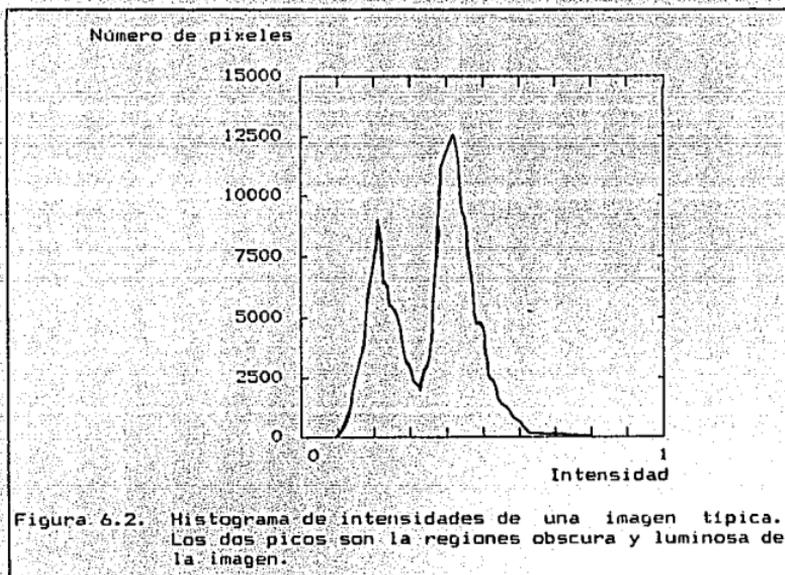
Una codificación eficiente para reducción del número de bits de una imagen es a través de una representación apropiada de la señal, por ejemplo en una transformación de coeficientes la cual deberá ser reversible. También se logra a través de reducción estadística, la cual involucra procesos de correlación entre muestras de cada pixel o correlación entre pixeles adyacentes como se mencionó en el capítulo 4.

En la codificación de las señales monocromáticas se tendrá una gráfica de intensidad contra número de pixeles para el caso de escalas de gris en intensidad luminosa (figura 6.2)

Para el muestrear una señal de video se emplea un barrido convencional como el mostrado en la figura 6.3, el cual convierte una imagen de video monocromática a una secuencia de muestras. En esta figura se observa que existen 2 campos entrelazados (campos impares y campos pares). Cuando se codifica este tipo de señal puede obtenerse una correlación entre líneas adyacentes de un campo hacia otro, además de una correlación entre pixel y pixel adyacente, esto es debido a que estos se encuentran más próximos; con esto se tiene un ahorro en el almacenamiento de los datos empleando una técnica de codificación adecuada.

6.1.3 Imágenes en color.

La señal compuesta de video de imágenes en color contiene tres señales básicas: (a) Señal de luminancia, (b) Señal de crominancia y (c) Señal de sonido.



En el punto anterior se analizó la señal de luminancia, en donde se determinó que los porcentajes de la señal de color representaban la escala de gris de dicha señal. Para el caso de la señal de color, la información está contenida dentro de la señal de crominancia, en esta señal se encuentran dos señales básicas: Señal Q y Señal I. Ambas señales se encuentran defasadas entre sí 90°, este defasamiento obedece a que podrán modularse en cuadratura para su posterior transmisión. Estas señales cuentan con los porcentajes respectivos para altas y bajas frecuencias de color, los porcentajes en base a los tres colores primarios son los siguientes:

$$I = 0.60 R - 0.29 V - 0.32 A \quad (6.11)$$

$$Q = 0.21 R - 0.52 V + 0.31 A \quad (6.12)$$

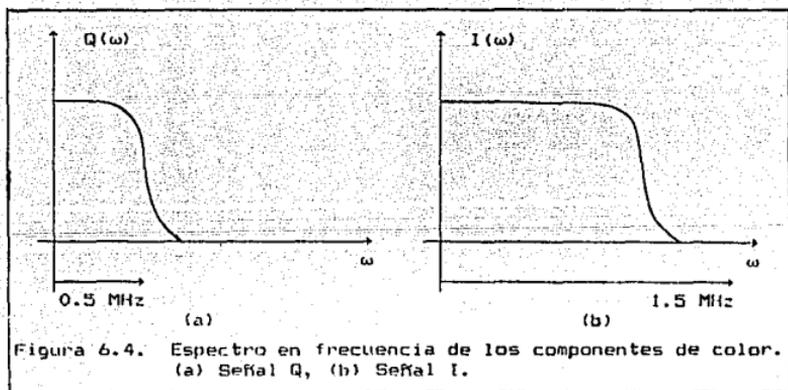
En la figura 6.4, se muestra el espectro en frecuencia para la señal de color; la señal Q representa los detalles gruesos de color, tales como líneas, barras y fondos de color, para el caso de I representa los detalles finos de color, estos detalles son frecuentemente líneas delgadas, bordes de imagen, etc. En la figura 6.4a se muestra el ancho de banda de la señal Q el cual se restringe hasta 0.5 MHz puesto que esta señal únicamente es para frecuencias bajas de la señal de color; la señal I abarca las frecuencias más altas de la señal de color, esto se muestra en la figura 6.4b.

Las señales de color se manejan de manera conjunta para la transmisión de imagen cromática analógica. Para la digitalización de imágenes de color, se tienen dos formas de codificar: (a) Codificación de imagen por señal compuesta y (b) Codificación de imagen por componente.

6.1.3.1 Señal compuesta.

En la señal compuesta se encuentra la señal de luminancia y

La señal de crominancia (ésta contiene las señales Q e I) se localiza a 3.58 MHz de la portadora Y. Cuando se muestrea la señal compuesta, los rangos de frecuencia abarcan cerca de los 6 MHz, y basándose en el muestreo de Nyquist se tendrá una frecuencia superior a 12 MHz. Al muestrear por señal compuesta se tiene el inconveniente en la adquisición y el muestreo, debido a que una computadora personal convencional no puede adquirir una frecuencia de muestreo tan alta, por esta razón, difícilmente es conveniente codificar una señal de imagen de color por señal compuesta.



6.1.3.2 Señal por componente.

En la señal de imagen de color, pueden muestrearse los componentes de la misma, estos a su vez tienen frecuencia de muestreo muchos más bajas que toda la señal de video. Para la señal I, su ancho de banda es de aproximadamente 1.5 MHz y su frecuencia de muestreo sería mayor a 3 MHz, dando con esto una frecuencia más baja para la adquisición de la misma, para la señal Q el ancho de banda resulta de 0.5 MHz y por lo tanto tiene una

frecuencia de muestreo de 1 MHz. Si se muestrean ambas señales, estas pueden ser mayores de 3 MHz en los dos casos, y por lo mismo su procesamiento resulta menos complejo que al utilizar todos los componentes de imagen.

Hay que hacer notar que una imagen de color, es una imagen monocromática adisionada de color en las zonas respectivas, es decir, que se puede procesar por separado la señal de color y la señal de luminancia para posteriormente sumarse con una escena normal.

6.2 CUANTIZACION Y CODIFICACION.

Para que una imagen sea procesada digitalmente, la señal de entrada debe ser muestreada y una vez que se ha muestreado se codifica y se cuantiza. Por ejemplo, para transmitir un documento por FAX la señal es cuantizada en pixeles blanco y negro, y posteriormente se codifica con un bit por muestra (0 o 1).

La selección del número de pixeles depende de la aplicación y de la calidad que se desee en la imagen. La reducción del número de pixeles origina el efecto de falso contorno, este efecto es perjudicial en aplicaciones tales como la transmisión de imágenes de televisión, tomografía y percepción remota; porque se presentan pequeñas variaciones en la región de luminancia.

En señales monocromáticas, por ejemplo la televisión en blanco y negro, la señal es codificada utilizando de 64 a 256 niveles por muestra lo que corresponde a 6 u 8 bits por muestra, respectivamente. De igual forma, para imágenes en color, cada componente (rojo, verde y azul) es muestreado y codificado a 6 u 8 bits (64 o 256 niveles).

6.3 ALGUNOS EJEMPLOS DE IMAGENES CUANTIZADAS Y CODIFICADAS A 8 BITS 256 NIVELES.

6.3.1 Algoritmos.

A continuación se presenta un programa por medio del cual se lee una imagen, y se despliega en pantalla.

```
#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <alloc.h>
#define PANTALLA_X 320
#define PANTALLA_Y 200
typedef unsigned char BYTE;
typedef BYTE huge * AP_IMG;
        AP_IMG buffer;

void inicializa(BYTE modo);
void lee_archivo(void);
void limpia(void);
void paleta_64();

unsigned char far *video=MK_FP(0xA000,0x0000);

/*****/
/*
/* " PROCESAMIENTO DIGITAL DE IMAGENES "
/*
/* Programa:      LECTURA DE UNA IMAGEN DE 256x256
/*
/* NOTAS:
/*      El presente programa consta de dos modulos
/*      1).- LEE_archivo
/*           pregunta por el nombre del archivo a leer,
/*           inicializa a modo grafico 320x200 y
/*           se despliega la imagen centrada de 256x200.
/*
/*      2).- Inicializa a modo texto de 80x25
/*
/*
/*****/
main ()
{
    lee_archivo(); /* lectura y despliegue de archivo */
    getch();      /* pausa */
    inicializa(0x03); /* se inicializa a modo texto de 80x25 */
    /* limpia(); */
}

```

```

void lee_archivo()
{
    FILE *fp; /* manejador de archivo*/
    char nom[20],aux[5];
    unsigned long dim_x=0,dim_y=0,i=0,j=0;
    int ofset=0;
    textcolor(YELLOW );
    textbackground(BLUE);
    clrscr();
    gotoxy(5,10);
    printf("Prog. que lee una imagen de Disco y la Despliega en pantalla");
    gotoxy(20,13);
    printf("Unicamente lee imagenes de 256 x 256 ");
    gotoxy(20,17);
    printf("Dame nombre del Archivo: ");
    gets(nom);

    /* gotoxy(20,17);
    printf("Dame la dimension en x: ");
    gets(aux);
    dim_x= atoi(aux);
    gotoxy(20,19);
    printf("Dame la dimension en y: ");
    gets(aux);
    dim_y= atoi(aux);
    */ dim_x=255;
    dim_y=255;
    /* reservacion de memoria */
    buffer=(AP_IMG) farmalloc ((dim_x*dim_y*sizeof(BYTE) ));
    if(buffer)
        ( /* si hay suficiente memoria disponible */

    fp=fopen(nom,"rb");
    if(fp)
        ( gotoxy(30,23);
          printf(" LECTURA DE ARCHIVO EN PROCESO");
          fread(buffer,dim_x*dim_y,1,fp);
        /*
          for(i=0; i< dim_y; i++)
              for(j=0; j< dim_x;j++)
                  buffer[i*dim_x + j]=fgetc(fp);
        */
          gotoxy(30,24);
          printf(" FIN DE LECTURA DE ARCHIVO ");
          /* despliegue de imagen */
          inicializa(0x13); /* se inicializa a mode grafico 320x200 */
          paleta_64(); /* inicializamos la paleta a 64 colores */
          ofset=32;
          for(i=0; i< PANTALLA_Y; i++)
              (
                  for(j=0; j< 256;j++)
                      video[i*PANTALLA_X+j+ofset]= buffer[ i*256 + j] / 4;
              )
    )
}

```

```

    getch();
    for(i=0; i< PANTALLA_Y; i++)
    {
        for(j=0; j< 256; j++)
            video[i*PANTALLA_X+j+offset]= 63 - (buffer [i*256+j]/4);
    }
    else
    {
        textcolor(RED);
        gotoxy(30,24);
        printf(" ARCHIVO NO SE ENCUENTRA EN DISCO");
    }
}
else
{
    textcolor(RED);
    gotoxy(30,24);
    printf(" NO HAY SUFICIENTE MEMORIA RAM");
}
}
}
}
}

void limpia()
{
    int long i;
    for( i=0 ; i <= (PANTALLA_X*PANTALLA_Y) ; i++)
        video[i]= 0;
}

void inicializa( BYTE modo)
{
    unsigned long indice;
    union REGS ent,sal;

    /*INICIALIZACION A MODEO 13 */

    ent.h.al= modo; /* PARAMETRO DE INICIALIZACION A MODO X */
    ent.h.ah= 0;
    int86(0x10,&ent,&sal);
}

void paleta_64()
{
    unsigned long indice;
    union REGS ent,sal;

    ent.h.al= 0x10; /* REORGANIZACION DE PALETA A NIVELES DE GRISES
    ent.h.ah= 0x10; /* 64 NIVELES */
    for(indice=0 ; indice < 0x40 ; indice++)
    {
        ent.x.bx= indice;
        ent.h.dh= indice;
        ent.h.cl= indice;
        ent.h.ch= indice;
    }
}

```

```
    int86(0x10,&ent,&sal);  
  )  
}
```

6.4 TÉCNICAS DE COMPRESIÓN DE IMAGENES.

En el capítulo 5 se mencionó en lo que consiste la compresión de imágenes, ahora se mencionarán algunos ejemplos.

La compresión de imágenes esta presente en áreas en donde es necesario el almacenamiento o transmisión de las mismas. El almacenamiento de imágenes se requiere en áreas como la educación, la publicidad (multimedia), la medicina (tomografía computarizada y resonancia magnética), la percepción remota (imágenes multiespectrales y de radar), la edición por computadora, etc. La transmisión de imágenes se aplica en la difusión de imágenes por televisión, el videoteléfono, videoconferencias, la percepción remota, comunicación entre computadoras, FAX, etc.

El volumen de información que representan las imágenes digitales son, por ejemplo, para la televisión digital de 216 Mbits/s, para la televisión de alta definición de 1.6 Gbits/s, para el videoteléfono es de 64 Kbits/s, para videoconferencia es de 384 Kbits/s a 2 Mbits/s, para tomografía computarizada de 1 Mbyte/s por corte.

El estudio de métodos y algoritmos de compresión de imágenes se desarrolla rápidamente conforme van pasando los años, esto debido a la necesidad que se tiene de desarrollar métodos que faciliten el manejo de imágenes tanto fijas como en movimiento, así como el desarrollo de circuitos específicos.

6.4.1 Transformada rápida de Fourier (FFT).

En la transformación discreta de Fourier (DFT), tanto la

entrada como la salida consisten de una secuencia de números definidos uniformemente en puntos espaciados en el tiempo y en la frecuencia, respectivamente. En efecto, la DFT proporciona directamente una evaluación numérica en la computadora. Además, los cálculos pueden ser implementados más eficientemente usando el clásico algoritmo denominado algoritmo de la transformada rápida de Fourier (FFT). Este algoritmo ha jugado un papel importante en las aplicaciones del procesamiento digital de señales.

La FFT refiere a una clase de eficientes algoritmos para el cálculo de la DFT. Básicamente un algoritmo de la FFT logra un cálculo eficiente al seguir la estrategia de "dividir y vencer", por lo cual el cálculo de la DFT original es dividida sucesivamente en pequeños cálculos de la DFT.

El algoritmo de la transformada rápida de Fourier (FFT) de Cooley-Tukey ha encontrado un gran uso en los códigos de transformación de imágenes. La FFT es usada para obtener la compresión del ancho de banda en la transmisión de una imagen. Un modelo práctico utilizando la FFT ha sido presentado por Anderson y Huang en 1971, el cual trata subsecciones de imágenes extendidas en series de Fourier y descartando los coeficientes de Fourier relativamente bajos de energía para derivar conjuntos de números los cuales son más pequeños en tamaño para una calidad de reconstrucción de imagen buena.

A continuación se presenta una parte de un programa por medio del cual se lee una imagen, y se despliega en pantalla, así como su transformación utilizando la transformada rápida de Fourier. Si se desea una información completa de este programa, ver el Apéndice.

```
/* inicio de fft */
  le = n;
  windex = 1;
  for (l = 0 ; l < m ; l++) {
```

```

le = le/2;

/* primera iteracion no tiene multiplicaciones */
for(i = 0 ; i < n ; i = i + 2*le) {
    xi = x + i;
    xip = xi + le;
    temp.real = xi->real + xip->real;
    temp.imag = xi->imag + xip->imag;
    xip->real = xi->real - xip->real;
    xip->imag = xi->imag - xip->imag;
    *xi = temp;
}

/* restantes iteraciones almacenadas en w */

wptr = w + windex - 1;
for (j = 1 ; j < le ; j++) {
    u = *wptr;
    for (i = j ; i < n ; i = i + 2*le) {
        xi = x + i;
        xip = xi + le;
        temp.real = xi->real + xip->real;
        temp.imag = xi->imag + xip->imag;
        tm.real = xi->real - xip->real;
        tm.imag = xi->imag - xip->imag;
        xip->real = tm.real*u.real - tm.imag*u.imag;
        xip->imag = tm.real*u.imag + tm.imag*u.real;
        *xi = temp;
    }
    wptr = wptr + windex;
}
windex = 2*windex;
}

/* re-arreglamiento de bit-inverso */

j = 0;
for (i = 1 ; i < (n-1) ; i++) {
    k = n/2;
    while(k <= j) {
        j = j - k;
        k = k/2;
    }
    j = j + k;
    if (i < j) {
        xi = x + i;
        xj = x + j;
        temp = *xj;
        *xj = *xi;
        *xi = temp;
    }
}

/* escalamos todos los valores por 1 entre la raiz de (n) */

```

```

scale = (float)(1.0/sqrt(n));
for(i = 0 ; i < n ; i++) {
    x->real = scale*x->real;
    x->imag = scale*x->imag;
    x++;
}
)
)

```

6.4.2 Transformada Discreta Coseno.

La transformada discreta coseno puede ser implementada con un algoritmo computacional, y tener una ejecución limitada por la transformación de Karhunen-Loeve. Sin embargo, la transformación de Karhunen-Loeve es considerada para ser la transformación óptima, este algoritmo no es general para habilitaciones computacionales rápidas.

A continuación se presenta una parte de un programa por medio del cual se obtiene la transformación de una imagen utilizando la transformada discreta coseno. Si se desea una información completa de este programa, veer el Apéndice.

```

/* calculo de la transformada casenoidal directa */
buffer=transforma_cos(datos,dim_x,dim_y,imagen_real,buffer,escala,1);
pinta_imagen(buffer,dim_x,dim_y,1);

/* calculamos la transformada inversa del coseno */
buffer=transforma_cos(datos,dim_x,dim_y,imagen_real,buffer,escala,-1);
pinta_imagen(buffer,dim_x,dim_y,1);

/* liberamos memoria */
farfree( (void *) imagen_real);
farfree( (void *) datos);
) /* fin del if */
else
{
    printf("No hay suficiente memoria");
    printf("no se proceso nada");
    getch();
}
)

AF_IMG transforma_cos(COMPLEX *datos,int dim_x, int dim_y,

```

```

        int *imagen_real, AP_IMG buffer, float escala, int direc)
(
    unsigned long int i, j, row, col;
    int m=0;
    m=base_2(dim_x+1);

    /* PROCESAMIENTO DE RENGLONES */

    for(row=0; row<=(dim_y); row++)
    {
        for(i=0; i<=(dim_x); i++)
            if(direc==1)
            {
                datos[i].real=(float) buffer[row*(dim_x+1)+i];
                datos[i].imag= 0.0;
            }
            else /* cargamos coeficientes de transformada inversa */
            {
                datos[i].real=(float) imagen_real[row*(dim_x+1)+i]/escala;
                datos[i].imag= 0.0;
            }

        if(direc==1) /* transformada cosenoidal directa*/
        {
            arregla_datos(datos,m); /* agrupa pares y despues impares */
            fft(datos,m); /* llamamos la fft */
            /* multiplicamos por W de 2N y ademas por raiz de 2 */
            multiplica_w2n(datos,m,direc);
        }
        else
        {
            coseno_inversa(datos,m);

            /* guardamos parte real */
            for(i=0; i<=(dim_x); i++)
            {
                imagen_real[row*(dim_x+1)+i]= floor( datos[i].real*escala)
            }
        }
    } /* fin del for de "row"*/

    /* PROCESAMIENTO DE COLUMNAS */

    for(col=0; col<=(dim_x); col++)
    {
        j=col;
        for(i=0; i<=(dim_x+1); i++)
        {
            /* Copiamos real e imaginario a cero */
            datos[i].real=(float) imagen_real[j]/escala;
            j+=(dim_x+1);
            datos[i].imag=0.0;
        }
    }
}

```

```

if(direc==1) /* transformada cosenoidal directa*/
{
    arregla_datos(datos,m); /* agrupa pares y despues impares */
    fft(datos,m);
    /* multiplicamos por W de 2N y ademas por raiz de 2 */
    multiplica_w2n(datos,m,direc);
}
else
{
    coseno_inversa(datos,m);
}

j=col;
for(i=0;i<=(dim_x);i++) /*guardamos valor de la transformacion*/
{
    imagen_real[j]= floor ( datos[i].real*escala );
    j+=(dim_x+1);
}
} /* fin del for de "col"*/

/* quitamos el factor de escala */
for(i=0;i<=(dim_x+1)*(dim_y+1);i++)
buffer[i]=floor(sqrt((imagen_real[i]/escala)*(imagen_real[i]/escala)));

return(buffer);
}

void limpia()
{ int long i; /* limpiamos pantalla mandando zeros */
  for( i=0 ; i <= (PANTALLA_X*PANTALLA_Y) ; i++)
    video[i]= 0;
}

int maximo(int x, int y)
{
  if (x > y) return (x);
  else return (y);
}

void pinta_imagen (unsigned char huge *imagena, int dim_x, int dim_y,
int flag)
{ unsigned long i=0,j=0;
  int max;

  /* encontramos el maximo */
  max=imagena[0];
  for ( i = 0 ; i <= (dim_x+1)*(dim_y+1); i++)
    max=maximo(max,imagena[i]);

  if(dim_y >= 200)
    dim_y=199;

  if(dim_x >= 320)

```

```

        dim_y=319;
        if(flag==0)
        (
            for(i=0; i<= dim_y; i++)
                for(j=0; j<= dim_x; j++)
                    video[i*PANTALLA_X+j+128]=imagena[i*(dim_x)+j];
        )
        else
        (
            for(i=0; i<= dim_y; i++)
                for(j=0; j<= dim_x; j++)
                    video[i*PANTALLA_X+j]=imagena[i*(dim_x)+j]*(255/max)/4;
        )
    )

/* arregla los datos de entrada en pares e impares espejados */
void arregla_datos(COMPLEX *datos, int m)
( int n=1, i;
  float par[256], impar[256]; /*variables auxiliares de intercambio */

  n = 1 << m; /* longitud de la imagen */

  for(i=0; i<=n/2-1; i++)          /* agrupamos pares a impares */
  { par[i] = datos[2*i].real;
    impar[i]= datos[2*i+1].real;
  }

  for(i=0; i<=n/2-1; i++)          /* guardamos primera mitad pares
  { datos[i].real=par[i];          /* y segunda mitad impares espejados */
    datos[n/2+i].real=impar[n/2-1-i];
  }
)

void desarregla_datos(COMPLEX *datos, int m)
( int n=1, i;
  float par[256], impar[256]; /*variables auxiliares de intercambio */

  n = 1 << m; /* longitud de la imagen */

  for(i=0; i<=n/2-1; i++)          /* agrupamos pares a impares */
  { par[i] = datos[i].real;
    impar[i]= datos[n/2+i].real;
  }

  for(i=0; i<=n/2-1; i++)          /* agrupamos pares a impares */
  { datos[2*i].real=par[i];
    datos[2*i+1].real=impar[n/2-1-i];
  }
)

void multiplica_w2n(COMPLEX *datos, int m, int direc)
( int n, i;
  float s, c;
  n = 1 << m;

```

```

for(i=0; i<n;i++)
(
    c= cos(PI*i/(2*n));
    s= direc*sin(PI*i/(2*n));
    datos[i].real= ( datos[i].real*c + s*datos[i].imag);
)
/* multiplicacion por el factor alfa[k]; a[k]=1 k=0 y /*
/* a[k]=sqrt(2) a[k]>0 /*
for(i=1; i<n;i++)
    datos[i].real= datos[i].real*sqrt(2.0);
)

void inicializa( BYTE modo)
(
    unsigned long indice;
    union REGS ent,sal;

    ent.h.al= modo; /* PARAMETRO DE INICIALIZACION A MODO X */
    ent.h.ah= 0;
    int86(0x10,&ent,&sal);
)

void paleta_64()
(
    unsigned long indice;
    union REGS ent,sal;

    ent.h.al= 0x10; /* REORGANIZACION DE PALETA A NIVELES DE GRISES */
    ent.h.ah= 0x10; /* 64 NIVELES */
    for(indice=0 ; indice < 0x40 ; indice++)
    (
        ent.x.bx= indice;
        ent.h.dh= indice;
        ent.h.cl= indice;
        ent.h.ch= indice;
        int86(0x10,&ent,&sal);
    )
)

int base_2( int num)
(
    float resul=0;
    if (num>1)
        resul= log(num)/log(2);
    else
        resul=0;
    return( floor(resul) );
)

void main ()
(
    buffer=lee_archivo(&dim_x, &dim_y); /*lectura y despliegue de archivo*/
    if ( buffer)

```

```

(
  inicializa(0x13); /* se inicializa a modo grafico 320x200 */
  paleta_64(); /* inicializamos la paleta a 64 colores */
  pinta_imagen(buffer,dim_x, dim_y,0);
  aplica_transfor_cos(buffer, dim_x, dim_y);
  getch();
  free((void *)buffer);
)
inicializa(0x03); /* se inicializa a modo texto de 80x25 */
)

```

6.5 APLICACIONES.

La digitalización o conversión de señales, imágenes y demás datos a números, provenientes de un experimento o medición, permite el análisis matemático para extraer, clasificar y cuantificar la información. La información visual da origen al procesamiento digital de imágenes, a continuación se mencionará una aplicación a problemas de Biomedicina.

Una parte importante de la información de un problema biomédico puede estar contenida en forma de imágenes o señales bidimensionales interpretables visualmente, por ejemplo, la tinción de diferentes elementos o sustancias, etc. Estas imágenes son denominadas imágenes biomédicas. Para clasificar un objeto biológico existen dos formas: IN VIVO (radiografías, ecografías, tomografías, técnicas de medicina nuclear, etc.) o bien IN VITRO (análisis de muestras aisladas como tejidos o células, microscopía óptica y electrónica y el análisis macroscópico).

La figura 6.5 muestra una Microestación de Procesamiento Digital de Imágenes, la cual se divide en unidad de adquisición y unidad de procesamiento. La primera esta encargada de transformar una escena física o real en un modelo informático en dos dimensiones denominado imagen digital, y la segunda encargada de realizar las operaciones sobre las imágenes (procesamiento, análisis, restitución) y de manejar los periféricos informáticos y específicos.

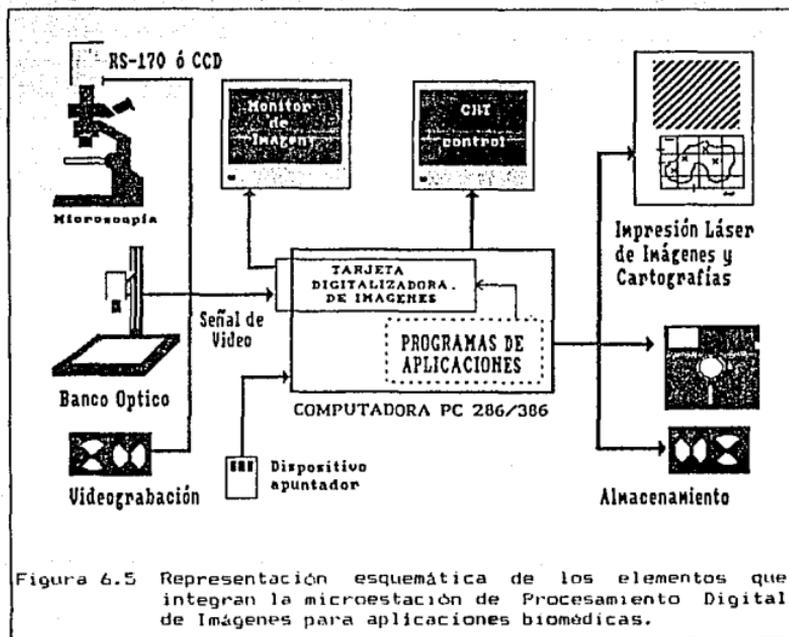


Figura 6.5 Representación esquemática de los elementos que integran la microestación de Procesamiento Digital de Imágenes para aplicaciones biomédicas.

Los elementos de la unidad de adquisición comprende el objeto a procesar, el sistema de formación de imagen, el captor de la imagen adaptado a la escena que se va a modelar y el dispositivo de digitalización y visualización de la imagen adquirida; y los de la unidad de procesamiento son: procesamiento sobre la imagen de video en tiempo real o procesamiento de la imagen fuera de línea, visualización y edición en el monitor de imagen, impresora laser. La microestación esta basada en computadoras PC-AT compatibles.

CONCLUSIONES

En esta tesis se han expuesto los conceptos básicos asociados con la formación de una imagen y una introducción a los procesos básicos del procesamiento digital; así como las técnicas utilizadas durante estos procesos.

Se mencionaron los componentes básicos de un sistema de procesamiento digital de imagen como son: la adquisición de la imagen, la conversión de los datos de la imagen a datos manejables por una computadora, el procesamiento de los datos y los dispositivos para el despliegue de las imágenes.

El contenido de esta tesis estuvo enfocado principalmente a el procesamiento de los datos, es decir se trataron los temas correspondientes al análisis matemático, muestreo, codificación, cuantización y métodos de transformación de imágenes.

El procesamiento digital de imágenes o manipulación de imágenes utilizando la computadora es una técnica de trabajo que se caracteriza por tener aplicaciones en diversos problemas, lo cual lo convierte en una técnica dinámica que va evolucionando conforme pasa el tiempo. Debido a esta evolución se desarrollan nuevos algoritmos que hacen posible que se agilice el procesamiento de imágenes.

Existen otros factores que intervienen en el procesamiento digital de imágenes y son la tendencia que tienen los equipos de cómputo a bajar de costo y desarrollo de unidades de proceso y almacenamiento masivo (que tienden a ofrecer más capacidad en un menor espacio por el desarrollo de nuevas tecnologías en semiconductores y materiales).

Además de estas ventajas, se une el desarrollo de nuevas arquitecturas en computadoras que emplean arreglos de microprocesadores de bajo costo como sensores, hacen posible el crecimiento de esta técnica, es el trabajo en nuevas aplicaciones como diagnóstico médico, percepción remota, automatización industrial, arte, soporte de métodos de incursión militar, etc.

Con este trabajo se pretende obtener un interés especial, por las técnicas que sirven como base del procesamiento digital de imágenes, para que se incrementen los sistemas que tienen como base esta área y se puedan ampliar las aplicaciones haciéndolas más prácticas.

**NO
EXISTE
PAGINA**

A P E N D I C E

A continuación se presenta el programa completo por medio del cual se lee una imagen de un archivo, para posteriormente ser desplegada en pantalla; utilizando la transformada rápida de Fourier y la transformada discreta coseno.

```

#include <dos.h>
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <alloc.h>
#include <conio.h>
#include <math.h>
#define PANTALLA_X 320
#define PANTALLA_Y 200
#define PI 3.1415926
typedef unsigned char BYTE;
typedef struct {float real,imag;} COMPLEX;
typedef BYTE huge * AP_IMG;
AP_IMG buffer;
/* DECLARACION DE FUNCIONES */
void multiplica_w2n(COMPLEX *datos,int m,int direc);
void arregla_datos(COMPLEX *datos,int m);
void desarregla_datos(COMPLEX *datos,int m);
int base_2( int num);
void paleta_64();
void inicializa( BYTE modo);
void printa_imagen(unsigned char huge *imagen, int dim_x, int dim_y,
int flag);
void aplica_transfor_cos(unsigned char huge*buffer,int dim_x,int dim_y)
AP_IMG transforma_cos(COMPLEX *datos,int dim_x, int dim_y,
int *imagen_real,AP_IMG buffer,float escala,int direc);
int dim_x,dim_y;
void coseno_inversa(COMPLEX *datos,int l);
unsigned char huge *video=MK_FP(0xA000,0x0000);
/* NOTAS: */
/* El presente program consta de dos modulos */
/* 1).- LEE_archivo */
/* pregunta por el nombre del archivo a leer, */
/* inicializa a modo grafico 320x200 y */
/* se despliega la imagen centrada de 256x200 */
/* */
/* 2).- inicializa a modo texto de 80x25 */
/* */

```

```

AP_IMG lee_archivo( int *dim_x, int *dim_y)
{
    FILE *fp; /* manejador de archivo*/
    AP_IMG tempo;
    char nom[20],aux[5];
    int j;

    textcolor(YELLOW );
    textbackground(BLUE);
    clrscr();
    gotoxy(5,10);
    printf("Prog. que hace la transformacion discreta del coseno (TDC)");
    gotoxy(20,13);
    printf("unicamente para imagenes multiples de 2 a la n");
    gotoxy(20,15);
    printf("Dame nombre del Archivo: ");
    gets(nom);

    gotoxy(20,17);
    printf("Dame la dimension en x: ");
    gets(aux);
    *dim_x= atoi(aux);
    gotoxy(20,19);
    printf("Dame la dimension en y: ");
    gets(aux);
    *dim_y= atoi(aux);
    /* reservacion de memoria */
    buffer=(AP_IMG) farmalloc (( (*dim_x)*(*dim_y) * sizeof(BYTE) ) );
    if(buffer)
    { /* si hay suficiente memoria disponible */

fp=fopen(nom,"rb");
if(fp)
    { gotoxy(30,23);
      printf(" LECTURA DE ARCHIVO EN PROCESO");
      tempo=buffer;
      for(j=0;j<=(*dim_y);j++)
      { fread( (void *)tempo, sizeof(BYTE),(*dim_x+1),fp);
        tempo+=(*dim_x+1);
      }

      gotoxy(30,24);
      printf(" FIN DE LECTURA DE ARCHIVO ");
      return(buffer);
    }
    else
    { textcolor(RED );
      gotoxy(30,24);
      printf(" ARCHIVO NO SE ENCUENTRA EN DISCO");
    }
}
}

```

```

else
    ( textcolor(RED );
      gotoxy(30,24);
      printf(" NO HAY SUFICIENTE MEMORIA RAM");
      getch();
      exit(0);
    )

    free(fp);
    return(NULL);
}

/*****
/* al modulo FFT se le pasan dos parametros */
/* entradas: */
/* - un apuntador al arreglo de elementos a transformar */
/* - el numero de elementos del arreglo */
/* */
/* salidas: */
/* el valor de salida esta en el mismo */
/* arreglo que se paso de entrada */
*****/

void fft(x,m)
    COMPLEX *x;
    int m;
{
    static COMPLEX *w; /*variable donde se almacena el complejo w*/
    static int mstore = 0; /*almacenamiento para futuras referencias */
    static int n = 1; /*longitud de la fft */

    COMPLEX u,temp,tm;
    COMPLEX *xi,*xip,*xj,*wptr;

    int i,j,l,le,windex;
    float scale;

    double arg,w_real,w_imag,wrecur_real,wrecur_imag,wtemp_real;

    if(m != mstore) {
        if(mstore != 0) free(w);
        mstore = m;
        if(m == 0) return; /* si m=0 entonces regresa */
    }
    /* n = 2**m = longitud de la fft multiplo de 2 */

    n = 1 << m;
    le = n/2;

    /* localidad de almacenamiento de w */

    w = (COMPLEX *) calloc(le-1,sizeof(COMPLEX));

```

```

if(!w) {
    printf("\ no hay suficiente memoria ");
    exit(1);
}

/* calculo del valor de w recursivamente */

arg = 4.0*atan(1.0)/le;          /* calculo de PI/le */
wrecur_real = w_real = cos(arg);
wrecur_imag = w_imag = -sin(arg);
xj = w;
for (j = 1; j < le; j++) {
    xj->real = (float)wrecur_real;
    xj->imag = (float)wrecur_imag;
    xj++;
    wtemp_real = wrecur_real*w_real - wrecur_imag*w_imag;
    wrecur_imag = wrecur_real*w_imag + wrecur_imag*w_real;
    wrecur_real = wtemp_real;
}

/* inicio de fft */

le = n;
windex = 1;
for (l = 0; l < m; l++) {
    le = le/2;

/* primera iteracion no tiene multiplicaciones */

    for(i = 0; i < n; i = i + 2*le) {
        xi = x + i;
        xip = xi + le;
        temp.real = xi->real + xip->real;
        temp.imag = xi->imag + xip->imag;
        xip->real = xi->real - xip->real;
        xip->imag = xi->imag - xip->imag;
        *xi = temp;
    }

/* restantes iteraciones almacenadas en w */

    wptr = w + windex - 1;
    for (j = 1; j < le; j++) {
        u = *wptr;
        for (i = j; i < n; i = i + 2*le) {
            xi = x + i;
            xip = xi + le;
            temp.real = xi->real + xip->real;
            temp.imag = xi->imag + xip->imag;
            tm.real = xi->real - xip->real;
            tm.imag = xi->imag - xip->imag;
            xip->real = tm.real*u.real - tm.imag*u.imag;

```

```

        xip->imag = tm.real*u.imag + tm.imag*u.real;
        *xi = temp;
    )
    wptr = wptr + windex;
}
windex = 2*windex;
}

/* re-arreglamiento de bit-inverso */

j = 0;
for (i = 1 ; i < (n-1) ; i++) {
    k = n/2;
    while(k <= j) {
        j = j - k;
        k = k/2;
    }
    j = j + k;
    if (i < j) {
        xi = x + i;
        xj = x + j;
        temp = *xj;
        *xj = *xi;
        *xi = temp;
    }
}

/* escalamos todos los valores por 1 entre la raiz de (n) */
scale = (float)(1.0/sqrt(n));
for(i = 0 ; i < n ; i++) {
    x->real = scale*x->real;
    x->imag = scale*x->imag;
    x++;
}

}

void coseno_inversa(COMPLEX *datos,int l)
{int n,m,k;
float aux_datos[256],c0;
n= l <<1;

c0=datos[0].real*sqrt(1.0/n);

for(m=0;m<n;m++)
{
for(k=1;k<n;k++)
{
if(k==1)
aux_datos[m]=datos[k].real;
else
aux_datos[m]+=datos[k].real *cos( ((PI *k *m) /n) +
( PI*k) / (2*n)) *sqrt(2.0/n);
}
}
}

```

```

        aux_datos[m]+=c0;
    }
    /* copiamos los valores a datos[k] */
    for(k=0;k<n;k++)
        datos[k].real=aux_datos[k];
}

void aplica_transfor_cos(unsigned char huge *buffer,int dim_x,int dim_y
( COMPLEX *datos;
int *imagen_real;
float escala;
    imagen_real=(int *) farmalloc (((dim_x+1)*(dim_y+1) * sizeof(int)));
    datos = (COMPLEX *) farmalloc( (dim_x+1)*sizeof(COMPLEX) );

if ( imagen_real && datos)
{
    escala= sqrt(dim_x);
    /* calculo de la transformada casenoidal directa */
    buffer=transforma_cos(datos,dim_x,dim_y,imagen_real,buffer,escala,1);
    pinta_imagen(buffer,dim_x,dim_y,1);

    /* calculamos la transformada inversa del coseno */
    buffer=transforma_cos(datos,dim_x,dim_y,imagen_real,buffer,escala,-1);
    pinta_imagen(buffer,dim_x,dim_y,1);

    /* liberamos memoria */
    farfree( (void *) imagen_real);
    farfree( (void *) datos);
} /* fin del if */
else
{
    printf("No hay suficiente memoria");
    printf("no se proceso nada");
    getch();
}
}

AP_IMG transforma_cos(COMPLEX *datos,int dim_x, int dim_y,
    int *imagen_real,AP_IMG buffer,float escala,int direc)
{
    unsigned long int i,j,row,col;
    int m=0;
    m=base_2(dim_x+1);

    /* PROCESAMIENTO DE RENGLONES */

    for(row=0;row<=(dim_y);row++)
    {
        for(i=0;i<=dim_x;i++)
            if(direc==1)
            {

```

```

        datos[i].real=(float) buffer[row*(dim_x+1)+i];
        datos[i].imag= 0.0;
    }
    else /* cargamos coeficientes de transformada inversa */
    {
        datos[i].real=(float) imagen_real[row*(dim_x+1)+i]/escala;
        datos[i].imag= 0.0;
    }

if(direc==1) /* transformada cosenoidal directa*/
{
    arregla_datos(datos,m); /* agrupa pares y despues impares */
    fft(datos,m); /* llamamos la fft */
    /* multiplicamos por W de 2N y ademas por raiz de 2 */
    multiplica_w2n(datos,m,direc);
}
else
{
    coseno_inversa(datos,m);

    /* guardamos parte real */
    for(i=0;i<=dim_x;i++)
    {
        imagen_real[row*(dim_x+1)+i]= floor(datos[i].real*escala);
    }
} /* fin del for de "row"*/

/* PROCESAMIENTO DE COLUMNAS */

for(col=0;col<=dim_x;col++)
{
    j=col;
    for(i=0;i<(dim_x+1);i++)
    {
        /* copiamos real e imaginario a cero */
        datos[i].real=(float) imagen_real[j]/escala;
        j+=(dim_x+1);
        datos[i].imag=0.0;
    }

if(direc==1) /* transformada cosenoidal directa*/
{
    arregla_datos(datos,m); /*agrupa pares y despues impares*/
    fft(datos,m);
    /* multiplicamos por W de 2N y ademas por raiz de 2 */
    multiplica_w2n(datos,m,direc);
}
else
{
    coseno_inversa(datos,m);
}

j=col;

```

```

    for(i=0;i<=(dim_x);i++) /*guardamos valor de la transformacion*/
    {
        imagen_real[j]= floor( ( datos[i].real*escala );
        j+=(dim_x+1);
    }
} /* fin del for de "col"*/

/* quitamos el factor de escala */
for(i=0;i<=(dim_x+1)*(dim_y+1);i++)
buffer[i]=floor(sqrt((imagen_real[i]/escala)*(imagen_real[i]/escala)));
return(buffer);
}

void limpia()
{ int long i; /* limpiamos pantalla mandando ceros */
  for( i=0 ; i <= (PANTALLA_X*PANTALLA_Y) ; i++)
    video[i]= 0;
}

int maximo(int x, int y)
{
  if (x > y) return (x);
  else return (y);
}

void pinta_imagen(unsigned char huge *imagena, int dim_x, int dim_y,
                  int flag)
{ unsigned long i=0,j=0;
  int max;

  /* encontramos el maximo */
  max=imagena[0];
  for( i = 0 ; i <= (dim_x+1)*(dim_y+1) ; i++)
    max=maximo(max,imagena[i]);

  if(dim_y >= 200)
    dim_y=199;

  if(dim_x >= 320)
    dim_x=319;

  if(flag==0)
  {
    for(i=0; i<= dim_y; i++)
      for(j=0; j<= dim_x;j++)
        video[i*PANTALLA_X+j+128]=imagena[i*(dim_x)+j];
  }
  else
  {
    for(i=0; i<= dim_y; i++)
      for(j=0; j<= dim_x;j++)
        video[i*PANTALLA_X+j]=imagena[i*(dim_x)+j]*(255/max)/4;
  }
}

```

```

    )
}

/* arregla los datos de entrada en pares e impares espejados */
void arregla_datos(COMPLEX *datos,int m)
{ int n=1,i;
  float par[256],impar[256]; /*variables auxiliares de intercambio */
  n = 1 << m; /* longitud de la imagen */

  for(i=0; i<=n/2-1;i++) /* agrupamos pares a impares */
    { par[i] = datos[2*i].real;
      impar[i]= datos[2*i+1].real;
    }
  for(i=0; i<=n/2-1;i++) /* guardamos primera mitad pares
    { datos[i].real=par[i]; /* y segunda mitad impares espejados */
      datos[n/2+i].real=impar[n/2-1-i];
    }
}

void desarregla_datos(COMPLEX *datos,int m)
{ int n=1,i;
  float par[256],impar[256]; /*variables auxiliares de intercambio */
  n = 1 << m; /* longitud de la imagen */

  for(i=0; i<=n/2-1;i++) /* agrupamos pares a impares */
    { par[i] = datos[i].real;
      impar[i]= datos[n/2+i].real;
    }
  for(i=0; i<=n/2-1;i++) /* agrupamos pares a impares */
    { datos[2*i].real=par[i];
      datos[2*i+1].real=impar[n/2-1-i];
    }
}

void multiplica_w2n(COMPLEX *datos,int m,int direc)
{ int n,i;
  float s,c;
  n = 1 << m;

  for(i=0; i<n;i++)
    {
      c= cos(Pi*i/(2*n));
      s= direc*sin(Pi*i/(2*n));
      datos[i].real= ( datos[i].real*c + s*datos[i].imag);
    }
  /* multiplicacion por el factor alfa[k]; a[k]=1 k=0 y */
  /* a[k]=sqrt(2) a[k]>0 */
  for(i=1; i<n;i++)
    datos[i].real= datos[i].real*sqrt(2.0);
}

```

```
void inicializa( BYTE modo)
{
    unsigned long indice;
    union REGS ent, sal;

    ent.h.al= modo; /* PARAMETRO DE INICIALIZACION A MODO X */
    ent.h.ah= 0;
    int86(0x10, &ent, &sal);
}

void paleta_64()
{
    unsigned long indice;
    union REGS ent, sal;

    ent.h.al= 0x10; /* REORGANIZACION DE PALETA A NIVELES DE GRISES */
    ent.h.ah= 0x10; /* 64 NIVELES */
    for(indice=0 ; indice < 0x40 ; indice++)
    {
        ent.x.bx= indice;
        ent.h.dh= indice;
        ent.h.cl= indice;
        ent.h.ch= indice;
        int86(0x10, &ent, &sal);
    }
}

int base_2( int num)
{
    float resul=0;
    if (num>1)
        resul= log(num)/log(2);
    else
        resul=0;
    return( floor(resul) );
}

void main ()
{
    buffer=lee_archivo(&dim_x, &dim_y); /*lectura y despligue de archivo*/
    if ( buffer)
    {
        inicializa(0x13); /* se inicializa a mode grafico 320x200 */
        paleta_64(); /* inicializamos la paleta a 64 colores */
        pinta_imagen(buffer, dim_x, dim_y, 0);
        aplica_transfor_cos(buffer, dim_x, dim_y);
        getch();
        farfree((void *)buffer);
    }
    inicializa(0x03); /* se inicializa a modo texto de 80x25 */
}
```

**NO
EXISTE
PAGINA**

B I B L I O G R A F I A

1. CASTLEMAN, Kenneth R.
Digital Image Processing.
Primera Edición.
U.S.A., Ed. Prentice Hall, 1979.
2. GONZALEZ, Rafael.
Digital Image Processing.
Segunda Edición.
U.S.A., Ed. Addison Wesley, 1987.
3. GRAY, Robert H.
Vector Quantization.
I.E.E.E. ASSP Mag., Vol. 1, pp. 4-29, Abr. 1984.
4. GREEN, William B.
Digital Image Processing.
Segunda Edición.
U.S.A., Ed. Van Nostrand Reinhold, 1989.
5. JAIN, Anil K.
Fundamentals of Digital Image Processing.
Primera Edición.
U.S.A., Ed. Prentice Hall, 1989.
6. JAIN, Anil K.
Image Data Compression Review.
I.E.E.E. Vol. 69, No. 3, pp. 349-389, Mar. 1981.

7. NETRAVALI, Arun N.
Maskel, Barry G.
Digital Pictures: Representation and Compression.
Primera Edición.
U.S.A., Ed. Plenum Press, 1988.
8. NETRAVALI, Arun N.
Limb, John D.
Picture Coding.
I.E.E.E., Vol. 68, No. 3, pp. 366-406, Mar. 1980.
9. OLIVER, B. M.
Pierce, R. J.
Shanon, C. E.
The Philosophy of PCM.
I.R.E., Vol. 36, pp. 1324-1331, Nov. 1948.
10. OPPENHEIM, Alan V.
Application of Digital Signal Processing.
Primera Edición.
U.S.A., Ed. Prentice Hall, 1978.
11. PANTER, Philip F.
Modulation Noise and Spectral Analysis.
Primera Edición.
U.S.A., Ed. Mc Graw-Hill, 1965.
12. PRATT, William K.
Digital Image Processing.
Primera Edición.
U.S.A., Ed. Wiley Interscience, 1978.

13. ROSENFELD, Azriel.
Digital Picture Processing.
Segunda Edición.
Vol. 1.
U.S.A., Ed. Academic Press, 1982.
14. ROSENFELD, Azriel.
Digital Picture Processing.
Segunda Edición.
Vol. 2.
U.S.A., Ed. Academic Press, 1982.
15. ECHUARTZ, Mischa.
Transmisión de información Modulación y Ruido
Tercera Edición (Primera en español).
Mexico, Ed. Mc Graw-Hill, 1983.
16. TAUB, H.
Shilling, D. L.
Principles of Communication Systems.
Segunda Edición.
U.S.A., Ed. Mc Graw-Hill, 1980.
17. Image Transmission Techniques.
Primera Edición.
U.S.A., Ed. Academic Press, 1979