

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA DIVISIÓN DE INGENIERÍA ELÉCTRICA

Transmisión de imágenes utilizando el estándar JPEG 2000

**TESIS PRESENTADA POR:
SANDRA CAROLINA BAUTISTA CORTÉZ**

Dirigida por: Dr. Víctor García Garduño

Jurado:

Presidente: Ing. Juan Fernando Solórzano Palomares

Vocal: Dr. Víctor García Garduño

Secretario: Ing. Jesús Reyes García

1er Spte.: Dr. Miguel Moctezuma Reyes

2do Spte.: Ing. Damián Federico Vargas Sandoval

Ciudad Universitaria, México D.F.

**A mis abuelitos Pepe[†], Pelancho y Eva, mis padres Fer
y Guanda, a mis hermanas Nor y Kika, a Edgar y a
mis amigos.**

**Agradezco:
A la vida por esto y muchas cosas más que me ha dado.**

**A mis profesores, a mi director de tesis y sinodales por su apoyo
y enseñanzas.
A mis compañeros de butacas.**

El primero de... ¿?

Introducción

Actualmente las comunicaciones inalámbricas han tenido un crecimiento acelerado; donde los servicios de multimedia de banda ancha, el correo electrónico, la transmisión de archivos y la distribución de servicios; tales como radio, TV y el abastecimiento de software; han jugado un rol importante dentro del avance de los sistemas inalámbricos. Al mismo tiempo, el procesamiento digital de señales tuvo que evolucionar, por lo que el análisis y el procesamiento de video, imágenes, audio y datos, hoy por hoy tienen altos índices de funcionamiento.

Estos progresos han ayudado al diseño de redes inalámbricas para soportar la mayor cantidad de servicios a diferentes tasas de transferencia, reduciendo las pérdidas durante la transmisión. Por tal motivo es importante para un ingeniero en telecomunicaciones conocer la problemática asociada a una transmisión inalámbrica, así como las técnicas modernas para corregirla, que den como resultado una transmisión óptima.

El objetivo de este tema de tesis es dar una perspectiva técnica de un modelo de transmisión sobre medios inalámbricos sin abarcar un protocolo para su distribución en redes inalámbricas. Debido a la importancia de la compresión de imágenes dentro de las comunicaciones de multimedia se tratará el tema de la transmisión de imágenes fijas usando un estándar reciente de compresión de imágenes, llamado JPEG2000 (Joint Photographic Expert Group 2000 por sus siglas en inglés).

Durante el desarrollo de esta misma se expondrá el estado del arte del procesamiento de imágenes, de la codificación y del modelado de un canal, para posteriormente mostrar algunos ejemplos de aplicaciones. Con el objetivo de facilitar la comprensión de JPEG2000 se mostrarán teóricamente algunos algoritmos utilizados en el estándar con más detalle. También se hará énfasis en la codificación conjunta de fuente canal para la transmisión de imágenes, ya que como es tradicional en los modelos de comunicación el proceso de codificación es dividido en dos partes: en la codificación de fuente y en la codificación de canal y aquí se mostrarán las ventajas de realizar una codificación conjunta de fuente canal.

La presente tesis se divide en siete capítulos, los cuales de una manera general introducen a la transmisión de información de imágenes sobre canales inalámbricos, específicamente, canales de radio móviles, donde los efectos de retraso y multitrayectorias determinan numerosos parámetros para la implementación de sistemas de comunicación móvil.

En el primer capítulo se introduce a los principios y fundamentos de la codificación de imágenes, además de las técnicas más usadas en la actualidad. En el capítulo dos se proporciona una descripción del estándar JPEG2000, asimismo sus alcances, ventajas y funcionalidades. Este estándar pretende ser el sucesor de JPEG en

muchas áreas de aplicación, como en el internet, cámaras digitales, multimedia, etc., por lo cual es importante su conocimiento. Debido a la importancia de realizar una óptima transmisión, en el capítulo tres se trata la codificación de canal, que es el proceso de protección de error. En este capítulo se verán las características de algunos esquemas de codificación de canal más usados, comenzando con la explicación de los códigos de bloque lineales hasta otros más complicados como son los códigos convolucionales.

Ya que es necesario el conocimiento de las características de propagación de un canal para el diseño de un buen sistema de comunicación, en el capítulo cuatro se introduce a los modelos de canal de propagación de radio; como los canales inalámbricos interiores, exteriores y satelitales; además de los modelos de canales de desvanecimiento por multitrayectoria y los modelos de desvanecimiento de corta y larga trayectoria.

En el capítulo cinco se incluyen los fundamentos de la codificación conjunta fuente canal, aquí se exponen algunas operaciones que se realizan en la codificación de fuente y de canal, la asignación de los recursos de bits óptimos de la codificación de fuente y de un sistema de codificación conjunta fuente canal.

En el capítulo seis se muestran algunas aplicaciones del estándar JPEG2000, iniciando con sus funcionalidades y características para posteriormente mostrar algunos ejemplos de aplicaciones que se han realizado con este estándar. La demostración de las características del estándar se realizara con ayuda de las utilerías del software Kakadu, el cual es una implementación de la primera parte del estándar en C++.

Por último, en el capítulo siete se describen otras diligencias viables en una transmisión inalámbrica; como la transmisión progresiva de imágenes y de video, utilizando el esquema de codificación conjunta fuente canal.

Índice

Introducción	iv
1. Codificación de fuente	1
1.1. Codificación espacial de fuente	3
1.1.1. Codificación Run Length	4
1.1.2. Codificación de Huffman	4
1.1.3. Codificación aritmética	5
1.1.4. Transformada Coseno Discreta.	6
1.2. La Transformada Wavelet	9
1.2.1. Definición matemática	9
1.2.2. Transformada Wavelet Discreta	11
1.2.3. Set Partitioning in Hierarchical Trees	14
1.3. Codificación de fuente temporal	14
1.3.1. Diferencia de cuadros.	15
1.3.2. Compensación y Estimación de Movimiento	15
Referencias	16
2. JPEG 2000 (Joint Photographic Experts Group 2000)	18
2.1. Estructura del estándar	19
2.2. Arquitectura básica del estándar	20
2.2.1. Modelo de imagen fuente	21
2.2.2. Cuadrícula de referencia	22
2.2.3. Tiling	24
2.2.4. Cambio a nivel DC	25
2.2.5. Transformación de componentes	26
2.2.6. Transformada Wavelet	27
2.2.7. Cuantización	30
2.2.8. Codificación de entropía	30
2.2.9. Recintos	31
2.2.10. Codificación del bit-plane	32
2.2.11. Paquetes y capas	36
2.2.12. Codificación de la cabecera del paquete	38
2.2.13. Codificación del cuerpo del paquete	38
2.3. Control de índice	39
2.4. Codificación de una región de interés	39
2.5. Code Stream	41
2.6. Formato de archivo	42
2.7. Escalabilidad	44
2.8. Recuperación de error	46
2.9. Características visuales	47
Referencias	48

3. Codificación de canal	50
3.1. Teorema de codificación de canal de Shannon	50
3.2. Códigos de bloque	51
3.2.1. Códigos lineales	53
3.2.1.1. Código de Hamming	53
3.2.2. Códigos cíclicos	54
3.2.2.1. Codificación	54
3.2.2.2. Generador y matriz de chequeo de paridad	55
3.2.2.3. Corrección de errores y el síndrome	55
3.3. Códigos Bose-Chaudhuri-Hocquenghem	56
3.4. Codificación de Reed Solomon	56
3.5. Códigos convolucionales	57
3.5.1. Formas de representación de un código convolucional	58
3.5.2. Decodificación de códigos convolucionales	59
3.5.2.1. Algoritmo de Viterbi	59
3.6. Códigos Convolucionales Incluidos a través de Índice Compatible	60
3.6.1. Definición de los códigos RCPC	61
3.6.2. Criterio de calidad para códigos RCPC	63
Referencias	65
4. Modelos de canal	66
4.1. Mecanismos básicos de propagación	66
4.2. Ambientes de propagación	67
4.3. Modelos de pérdidas de trayectorias a largas distancias (Large scale path loss models)	69
4.3.1. Modelo de propagación en el espacio libre	70
4.3.2. Modelo de reflexión en tierra (2-rayos)	70
4.3.3. Modelo de pérdida de trayectoria a largas distancias	72
4.3.4. Modelos de propagación exterior	72
4.3.4.1. Modelo de Longley-Rice	73
4.3.4.2. Modelo de Durkin	73
4.3.4.3. Modelo de Okumura	73
4.3.4.4. Modelo de Hata	74
4.3.4.5. Extensión para el modelo de Hata	75
4.3.4.6. Modelo Walfisch y Bertoni	75
4.3.5. Modelos de propagación interior	76
4.3.6. Trazado de rayo y modelado específico de sitio	76
4.4. Modelos de desvanecimiento de la señal por multitrayectorias o por pequeñas distancias (Multipath fading or small scale fading models)	77
4.4.1. Parámetros de los canales de multitrayectorias móviles	77
4.4.1.1. Propagación con retraso de tiempo y Ancho de banda coherente	77
4.4.1.2. Propagación Doppler y Tiempo coherente	78
4.4.2. Tipos de desvanecimiento a pequeñas escalas	78
4.4.3. Distribución Rayleigh y Ricean	79
4.4.3.1. Canal de desvanecimiento Rayleigh	79

4.4.3.2. Canal de desvanecimiento Ricean	81
4.4.4. Modelos Estadísticos para canales de atenuación por multi-trayectorias	82
4.5. Canales de ruido gaussiano blanco aditivo (AWGN)	83
4.6. Canales simétricos binarios (BSC)	85
4.7. Modelo de canales de Markov de estado-finito	85
Referencias	88
5. Esquema de Codificación de canal-fuente conjunta	90
5.1. Codificación canal fuente conjunta en imágenes	91
5.1.1. Estructura general de un sistema codificación canal fuente conjunta en imágenes	92
5.2. Codificación canal-fuente conjunta	93
5.2.1. Codificación de transformada	93
5.2.2. Propiedades estadísticas de las subbandas	93
5.2.3. Cuantización y estrategias de codificación	94
5.2.3.1. Definición matemática	95
5.2.3.2. Cuantización escalar	96
5.2.3.3. Cuantización vectorial	104
5.2.4. Uso eficiente de los recursos del bit	112
5.2.4.1. El problema de la asignación de bit	112
5.2.4.2. Codificación de subbanda	115
5.2.5. Efectos de transmisión de error	117
5.2.5.1. Índice de codificación de canal	118
5.2.6. Índice y distorsión global	118
5.2.6.1. Asignación de bit de un canal fuente conjunto	119
Referencias	119
6. Codificación de imágenes	121
6.1. Cualidades de la codificación de JPEG2000	121
6.2. Transmisión de imágenes JPEG2000 sobre canales ruidosos	132
6.2.1. Estructura de la fuente	133
6.2.2. Descripción general del algoritmo de asignación de índice para JSCC	134
6.2.3. Ejemplos de transmisión	135
Referencias	138
7. Otras aplicaciones	139
7.1. Transmisión progresiva de imágenes	139
7.2. Transmisión de video	140
Referencias	144
Conclusiones	146

Capítulo 1

Codificación de fuente.

Los fundamentos teóricos de la teoría de la información acumulados en los trabajos pioneros de Shannon [1], en 1948 fueron los primeros que matemáticamente definieron la cantidad de información presente en una fuente. Gracias a esta teoría se introducen algunos fundamentos dentro de los cuales las comunicaciones pueden tomar lugar.

La teoría de la información trata una variedad de problemas asociados con los límites de funcionamiento de los sistemas de transmisión de información, tal como los descritos en la figura 1.1

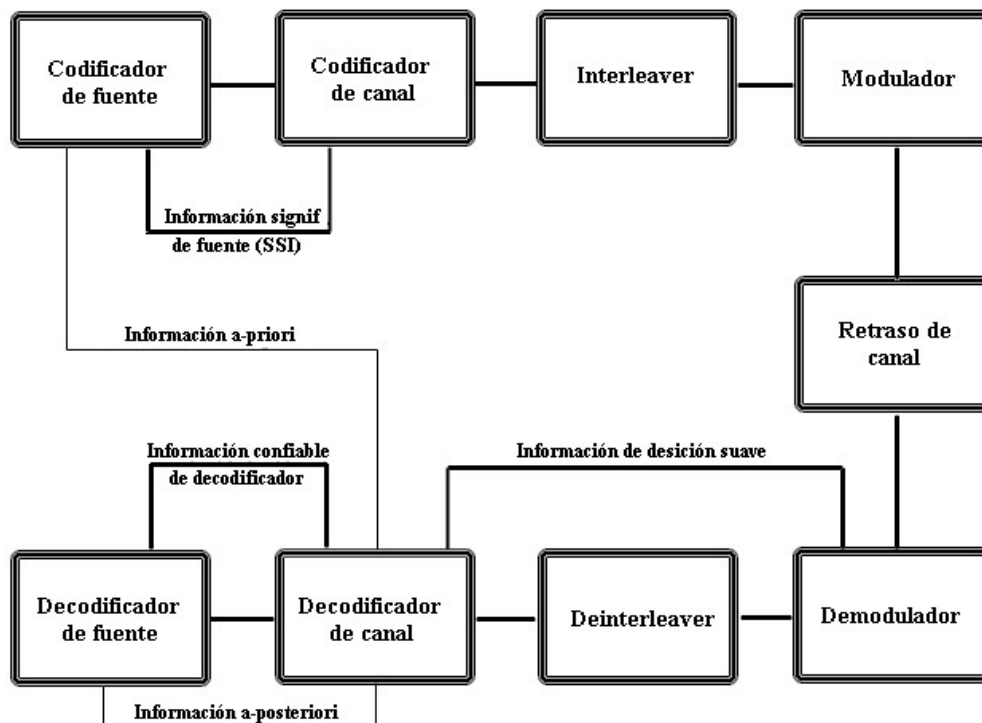


Figura 1.1 Modelo Básico de Transmisión de la Teoría de la información.

Basta decir que el estado del trasmisor en la figura 1.1 incorpora un codificador de fuente, un codificador de canal, un interleaver, y un modulador y sus funciones inversas en el receptor. Estos bloques se trataran en el desarrollo de este capítulo y los subsecuentes.

La codificación de fuente, conocida también como compresión de datos, es el proceso en el cual la información redundante de una fuente es removida teniendo como resultado un ahorro en espacio y ancho de banda. La información puede ser reconstruida tomando

en cuenta el mismo criterio de fidelidad por un proceso inverso, conocido como decodificación de fuente o descompresión de datos.

El método de compresión es conocido como compresión sin pérdidas si la recuperación de la información es perfecta. Ahora una compresión con pérdidas resulta si hay pérdidas considerables entre la fidelidad de la fuente original con la recuperada. Los datos perdidos en este método de compresión nunca pueden recuperarse provocando una calidad de la fuente menor, así, como un menor índice de transmisión de cuadros por segundo; lo cual, puede ser una desventaja de este tipo de métodos. En cambio en un sistema sin pérdidas la cantidad de memoria de almacenamiento es mayor que en un sistema con pérdidas siendo igualmente una desventaja. Para contrarrestar estos problemas existen varios conceptos dentro de la codificación de video, entre los cuales encontramos la codificación de imagen por cuadro y la codificación de video por cuadro en el dominio espacial y temporal.

Una estructura para un codificador de fuente puede ser representada como se ilustra en la figura 1.2. El preprocesador describe un compresor con pérdidas o sin pérdidas. En el caso de un compresor con pérdidas habrá un proceso adicional para la secuencia de video en ejecución. Este procesamiento adicional toma las diferencias entre cuadros, además del cálculo de vectores de movimiento requeridos por el decodificador para realizar la compensación de movimiento. A pesar del uso de un compresor con pérdidas o sin pérdidas la salida puede ser adicionalmente reducida por un código de entropía.

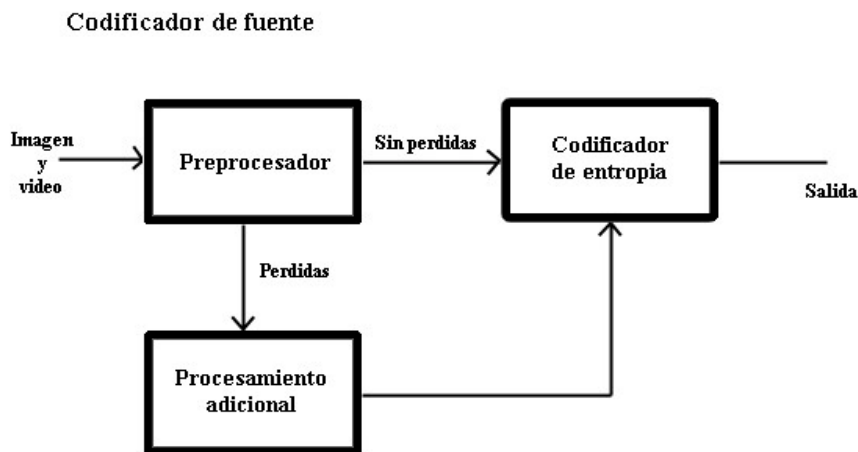


Figura 1.2 Estructura de un codificador de fuente para compresión de imágenes y video

Antes de definir un método de codificación de fuente se mencionaran los sistemas métricos para medir la calidad de una imagen que resulta de un proceso de codificación. El primer sistema es Mean Opinion Score (MOS) el cual se basa en la retroalimentación obtenida de un grupo de personas quienes juzgan la calidad subjetiva de la imagen decodificada contra la imagen original. Este sistema métrico está basado sobre cinco diferentes niveles o marcas usados para determinar la distorsión de la imagen

decodificada; las opciones son: muy irritante, irritante, poco irritante, perceptible pero no irritante e imperceptible.

Otro sistema métrico utilizado es el Mean Squared Error (MSE), el cual mide la distorsión entre la imagen original a y la imagen decodificada b , por medio de la siguiente fórmula:

$$MSE(a, b) = \frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [a(x, y) - b(x, y)]^2 \quad 1.1$$

donde N y M son el número de columnas y filas respectivamente, y (x, y) son las coordenadas actuales del píxel. Sin embargo el uso de Peak Signal to Noise Ratio (PSNR) es más común. El PSNR es similar al MSE excepto por un término adicional: el cuadrado del valor del píxel pico, o 255 para un píxel de 8 bits sin signo. El PSNR es usualmente medido en decibeles (dB) y se define como:

$$PSNR(a, b) = 10 \log_{10} \left[\frac{255^2}{\frac{1}{N \times M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} [a(x, y) - b(x, y)]^2} \right] \quad 1.2$$

Hay que aclarar que el PSNR no es la medida más fiel para definir que tan bien se ve una imagen reconstruida cuando se compara con la original, pues algunas veces se obtiene un buen resultado con el PSNR y uno muy malo en el MOS.

1.1. Codificación de Fuente Espacial.

En la codificación de fuente espacial, la compresión toma lugar dentro del espacio de la imagen removiendo solo la información redundante dentro del mismo, sin tomar en cuenta la redundancia temporal. También conocida como codificación intraframe, puede ser dividida en dos tipos: con pérdidas y sin pérdidas. Algunos ejemplos de esquemas de compresión de imágenes sin pérdidas incluyen los códigos de Huffman, la codificación Run Length (RLC) y el análisis Wavelet. Un ejemplo de una imagen con pérdidas es el basado en el estándar Joint Picture Experts Group (JPEG) [2]. La codificación intraframe remueve la redundancia en el dominio espacial ya que los datos con alta correlación espacial pueden ser bien comprimidos en el dominio de la frecuencia.

Debido a los estudios fisiológicos se ha demostrado que los parámetros del sistema visual humano (HVS por sus siglas en inglés) son más sensibles a los componentes de baja frecuencia que a los componentes de alta frecuencia. Los componentes de baja frecuencia son equivalentes a los valores promedios de luminancia en una imagen y los componentes de alta frecuencia representan las siluetas de las áreas en la imagen. Por lo cual la codificación espacial puede ser llevada a cabo removiendo esos datos que son imperceptibles dentro de un cuadro por el HVS.

1.1.1. Codificación Run Length

La codificación Run Length es un método muy simple sin pérdidas para comprimir datos. Esta técnica es una generalización del método de eliminación de blancos. Con esta técnica se puede reducir cualquier secuencia de caracteres cuando el nivel de ocurrencia es de tres o más caracteres iguales consecutivos. Un ejemplo donde esta técnica es aplicable podría ser la compresión de una imagen en dos tonos (blanco y negro), que solo consisten de unos y ceros.

Cuando se encuentran 3 o más caracteres iguales consecutivos lo que se hace es sustituir esta secuencia por:

- un carácter especial indicador de compresión
- el carácter que se comprime
- número que indica la cantidad de caracteres que se comprimen

Un ejemplo de esta codificación podría ser, si se tiene la siguiente cadena de entrada:

fgjjerØØØØØØØØØØttrrrrrrywqad

una vez realizada la compresión, la cadena resultante será:

fgSch4jjerScØ9Sct3Scr5ywqad

donde Ø representa un carácter blanco y Sc es el carácter especial indicador de compresión.

El proceso de descompresión consiste en recorrer la cadena de datos que llega a través del canal. Cuando encuentra un carácter especial que indique compresión sabrá que en esa posición se ha realizado una compresión y que el siguiente carácter indica el carácter que ha sido comprimido y a continuación aparece el número que indica cuantos caracteres fueron comprimidos y de esta forma podrá reconstruir la cadena original.

1.1.2. Codificación de Huffman

La codificación de Huffman iniciada por D.A. Huffman en 1962 [3], es otra técnica sin pérdidas usada para reducir la entropía de los datos al codificar, de tal forma que el símbolo más probable es representado con el código de palabra más pequeño. La entropía de los datos indica el “desorden” o la variedad que existe en ellos, en el ejemplo de imagen sería una medida de la variedad de luminancia que se presenta en ella.

Esta técnica se sitúa dentro de la familia de códigos con palabras de longitud variable. Esto quiere decir que un mismo símbolo puede ser representado por cadenas de longitud distinta según el conjunto de datos en el que esté. Esta característica de las palabras de código variable ayuda a decrementar la cantidad de redundancia en un conjunto de datos consiguiendo de esta forma la compresión.

De la definición de Huffman la entropía de una fuente se define como sigue:

$$H = \eta = -\sum_i p_i \log_2 p_i \quad [\text{bits/símbolo}], \quad 1.6$$

donde P_i la probabilidad de ocurrencia del símbolo i .

El valor de H es teóricamente más bajo que el número de bits requerido para representar un símbolo de la fuente. Las palabras código de Huffman pueden ser mostradas para satisfacer la siguiente limitación:

$$\eta \leq \bar{L} < \eta + 1,$$

donde \bar{L} es la longitud de la palabra código.

Con referencia a la tabla, para la construcción de un código de Huffman se sigue un algoritmo muy simple y fácil como se muestra a continuación:

Símbolo	Fase 1	Fase 2	Fase 3	Fase 4	Fin
S_0	0.4	0.4	0.4 \searrow	0.6 0 \rightarrow	1.0
S_1	0.3	0.3	0.3 0 \nearrow	0.4 1	
S_2	0.2	0.2 0 \rightarrow	0.3 1		
S_3	0.07 0 \rightarrow	0.1 1			
S_4	0.03 1				

1. Obtener la probabilidad de ocurrencia de cada símbolo fuente s_i .
2. Clasificarlos en orden incremental de probabilidad, con los símbolos de probabilidad más alta al inicio de la tabla y los de más baja probabilidad al final.
3. Asignar un 1 binario al símbolo de probabilidad más baja y un 0 binario al próximo símbolo con probabilidad más alta.
4. Remover los dos símbolos de probabilidad más baja de la tabla y añadirlos juntos para obtener una probabilidad combinada.
5. Si la probabilidad combinada es 1, entonces parar. Si no reinsertar la probabilidad combinada en la tabla y seguir nuevamente el paso 2.
6. Para obtener las palabras código de Huffman para un símbolo particular, trace la secuencia de 0s y 1s binarios de la última fase anterior para ese símbolo (como en la siguiente tabla). Este es el código Huffman para ese símbolo.

Símbolo fuente	Probabilidad	Código A	Código B
S_0	0.4	0	1
S_1	0.3	01	00
S_2	0.2	011	010
S_3	0.07	0111	0110
S_4	0.03	01111	0111

1.1.3. Codificación aritmética

En la codificación aritmética, que también es una técnica de codificación sin pérdidas, un número variable de símbolos de entrada son requeridos para producir cada símbolo de código. Una secuencia de símbolos fuentes es representada por un subintervalo de números reales dentro del intervalo unitario $[0,1]$. Intervalos más pequeños requieren de

más bits para especificarlos; intervalos más grandes requieren menos. Los símbolos sucesivos en los datos de entrada reducen el tamaño del intervalo actual proporcionalmente a sus probabilidades; los símbolos más probables reducen un intervalo por una cantidad más pequeña que los símbolos menos probables y así se añaden menos bits al mensaje. La codificación aritmética es más compleja que la codificación de Huffman, típicamente esta provee una ganancia de compresión mayor, aproximadamente 10% más que la codificación de Huffman en aplicaciones con imágenes.

1.1.4. Transformada Coseno Discreta.

La Transformada Coseno Discreta (DCT) es uno de los métodos más usados para la compresión con pérdidas de imágenes en el dominio de la frecuencia.

Esta transformada cuenta con una buena propiedad de compactación de energía, en este aspecto su comportamiento es muy similar a la KLT (Karhunen-Loève Transform), que produce coeficientes no correlacionados, con la diferencia de que los vectores base de la DCT dependen sólo del orden de la transformada seleccionada, y no de las propiedades estadísticas de los datos de entrada. La decorrelación de los coeficientes es muy importante para la compresión, ya que, el tratamiento posterior de cada coeficiente se puede realizar de forma independiente, sin pérdidas de eficiencia de compresión. Otro aspecto importante de la DCT es la capacidad de cuantificar los coeficientes utilizando valores de cuantificación que se eligen ad-hoc.

La DCT también está relacionada con la Transformada Discreta de Fourier (DFT), con la diferencia de que la DCT es una transformada real, debido a que los vectores base se componen exclusivamente de funciones coseno muestreadas.

Definición matemática

La TCD bidimensional (**) se define como:

$$F(u, v) = \frac{4c(u)c(v)}{N^2} \times \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)\pi u}{2N}\right] \cos\left[\frac{(2y+1)\pi v}{2N}\right] \quad 1.3$$

donde N es la longitud de uno de los lados de la imagen,

$$c(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & \text{caso contrario} \end{cases}, \quad c(v) = \begin{cases} \frac{1}{\sqrt{2}} & v = 0 \\ 1 & \text{caso contrario} \end{cases} \quad 1.4$$

$u=0, \dots, N-1$, y $v=0, \dots, N-1$. Las coordenadas de píxel son x y y , mientras u y v son los coeficientes de las coordenadas. De igual modo, la inversa de la DCT (IDCT) es dada por una ecuación similar a la siguiente:

* Se enfoca a la DCT bidimensional porque es la más utilizada en la compresión de imágenes.

$$F(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \frac{4c(u)c(v)}{N^2} \cos\left[\frac{(2x+1)\pi u}{2N}\right] \cos\left[\frac{(2y+1)\pi v}{2N}\right] \quad 1.5$$

La DCT descompone la señal en componentes de frecuencia espacial llamados coeficientes DCT. Al aplicar la transformada muchos de los coeficientes obtenidos son pequeños, es decir, la mayoría de la energía de los datos se almacena en pocos coeficientes. Los coeficientes de baja frecuencia aparecen en la parte superior izquierda de la matriz de transformación, y los coeficientes de alta frecuencia están en la parte inferior derecha de la matriz. El Sistema Visual Humano (HVS) es más sensible a los errores de los coeficientes de alta frecuencia que a los de los coeficientes en baja frecuencia

El objetivo de la DCT es conseguir que una imagen (dominio espacial) o secuencia de imágenes (dominios espacio-temporal), se traslade a un dominio transformado de tal forma que se reduzca el ancho de banda para la transmisión o los requerimientos para el almacenamiento; de tal forma que la subsiguiente recuperación de la imagen o secuencia de imágenes mediante la transformada inversa, no presente una distorsión perceptible.

En la codificación por transformada de imágenes, una imagen $N \times N$ es dividida generalmente en bloques, cada uno de tamaño $L \times L$. Por simplicidad las filas y las columnas de una imagen, y también del bloque, se supone que son del mismo tamaño (por esta suposición no se pierde generalidad). En general, se han utilizado bloques de tamaño 8×8 y 16×16 en codificación de imagen.

Después de dividir la imagen en bloques de tamaño $L \times L$ en el dominio transformado, el selector descarta algunos de los coeficientes tanto de forma adaptativa como fija. En la figura 1.3 se muestran las imágenes base de la DCT bidimensional. El bloque superior izquierdo es de intensidad uniforme, representando el promedio de un bloque de imagen. La progresión de izquierda a derecha representa un número creciente de bordes verticales. De forma similar, la progresión desde arriba hacia abajo representa un número creciente de bordes horizontales; en el bloque inferior derecho se produce la mezcla máxima de bordes horizontales y verticales (este bloque tiene una forma de tablero de ajedrez).

El no tomar en cuenta los coeficientes de alta frecuencia en el dominio de la DCT bidimensional implica la supresión de las “imágenes bases” correspondientes a la imagen original, por lo tanto podemos considerar la DCT bidimensional como un proceso de descomposición desde el dominio espacial hacia el dominio de las “imágenes bases” de la DCT. Esta descomposición estructural puede utilizarse adaptativamente seleccionando los coeficientes de la transformada bloque por bloque base, para su procesamiento posterior, mediante la asignación a cada bloque de un número finito de clases basada en la distribución de sus coeficientes.

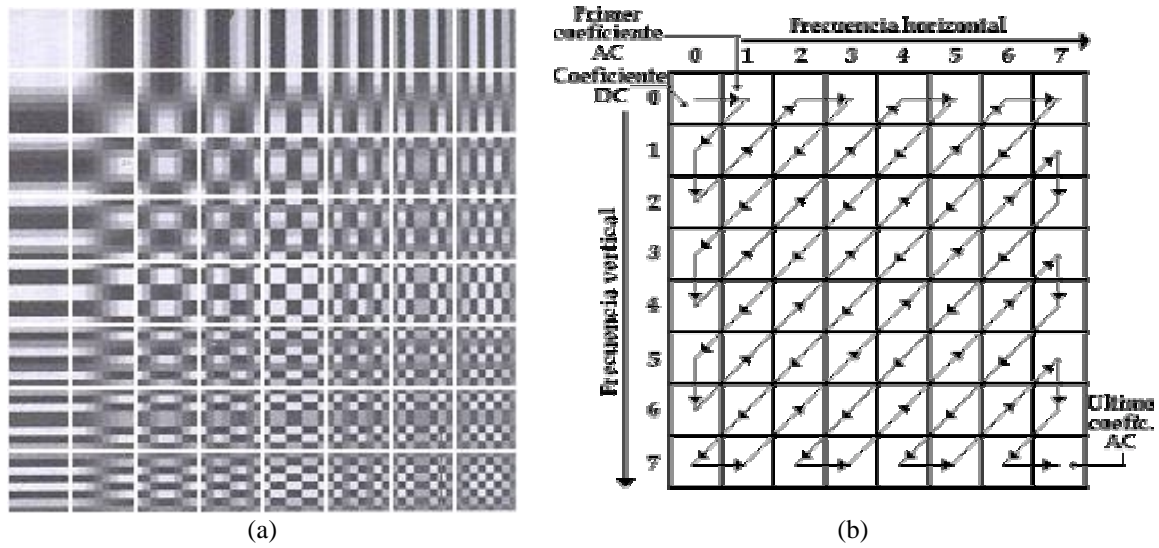


Figura 1.3 (a) Coeficientes de la DCT, (b) barrido o exploración en zigzag.

Las siguientes operaciones incluyen la cuantificación y codificación, antes de transmitir la información de la imagen en un flujo de bits serie a través de un enlace digital (microondas, cable coaxial, fibra óptica o satélite), hacia el receptor que realizará las operaciones inversas. Aunque la figura 1.4 muestra la codificación de una imagen basada en la DCT, ésta es igualmente válida para la codificación de una imagen a través de cualquier transformada ortogonal.

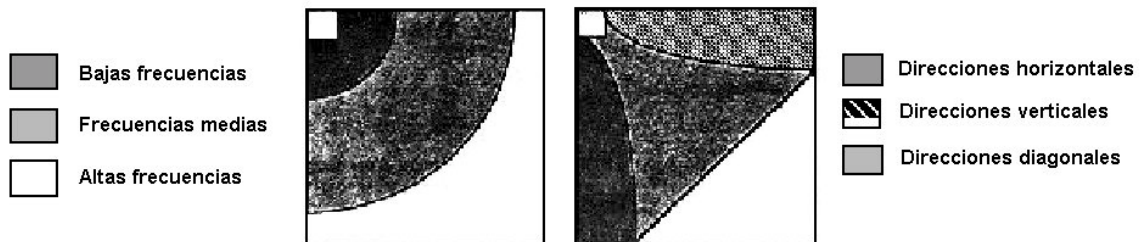


Figura 1.4 Distribución en frecuencia de los coeficientes de la DCT bidimensional y las características de bloque que representan; el coeficiente DC es el cuadradito de la esquina superior izquierda.

Como conclusión de la compresión de imágenes basadas en la DCT, podemos decir que los datos se clasifican de acuerdo con el grado de importancia de su contribución tanto a la información contenida como a la calidad subjetiva de la imagen; una vez que se logra tal clasificación, entonces aquellos elementos de los datos que no son importantes, desde el punto de vista de la escala de grises y la capacidad de resolución espacial del receptor pueden ser omitidos; esto hace posible un mayor grado de compresión de imagen.

1.2. La Transformada Wavelet

La transformada wavelet es una forma de codificación de fuente espacial.

La transformada wavelet es una función que convierte datos del dominio amplitud-tiempo al dominio de escala-tiempo. Ésta pertenece a la familia de decodificadores de fuentes conocidos como decodificadores de sub-banda. La transformada wavelet es similar a la Transformada de Fourier Continua Corta excepto por una diferencia importante: el análisis de ventanas de la transformada wavelet escala la frecuencia durante el análisis. En otras palabras para componentes de frecuencia alta (escala pequeña) en una señal el análisis de ventanas pequeño es requerido. Para frecuencias bajas (escala grande) corresponde un análisis de ventanas grande. Esta escalada permite la ejecución del análisis local, el cual puede ayudar en la detección de picos de alta frecuencia que aparecen relativamente en un breve instante mientras se fija a una señal de baja frecuencia.

La transformada wavelet y los paquetes de wavelets resultan ser adecuadas para una gran variedad de aplicaciones. Las wavelets dan lugar a representaciones multirejilla, puesto que mantienen la ortogonalidad respecto a sus dilataciones y translaciones. Sus características de localidad tiempo-frecuencia resultan útiles en el análisis y procesado de señales no estacionarias [4]. Debido a la propiedad de reconstrucción perfecta y la ausencia de degradación perceptual en los bordes, resultan adecuadas en aplicaciones de codificación de vídeo [5]. Por otro lado, los coeficientes wavelet presentan un decaimiento en su magnitud que es el más rápido de entre todas las bases ortonormales, lo que los hace muy dispersos. Por esta razón las bases wavelet resultan ser óptimas para la compresión de datos [6,7], reducción de ruido, y estimación estadística [8]. También son numerosas las aplicaciones existentes en el área de las telecomunicaciones y en el procesado de la señal, debido a la naturaleza multifrecuencia de la transformada. Citemos como ejemplos, las técnicas de codificación con sub-bandas y los transmultiplexores.

1.2.1. Definición matemática

La transformada wavelet continua (CWT) es una alternativa a la STFT. La CWT combina la aplicabilidad al análisis tiempo-frecuencia, propia de la STFT, con las propiedades matemáticas de la transformada de Fourier. La transformada wavelet proyecta una función del tiempo (la señal) en una función de a y τ (en vez de ω y τ). El parámetro a es denominado escala, mientras que τ es translación. Suponemos que la señal $x(t)$ es tal que $x(t) \in L^2(\mathbb{R})$, y por lo tanto

$$\int x^2(t) \delta t < \infty \quad 1.7$$

La CWT de $x(t)$ es

$$X_{CWT}(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-\tau}{a}\right) \delta t \quad 1.8$$

donde $\psi(t)$ es la 'wavelet madre' y las funciones-base $\frac{\psi\left(\frac{t-\tau}{a}\right)}{\sqrt{a}}$, resultado de las dilataciones y translaciones a que es sometida $\psi(t)$, constituyen la 'familia de wavelets'.

La transformación puede interpretarse también como la implementación de un banco de filtros no uniforme (y continuo en el tiempo) en el se mantiene una relación constante entre la resolución en frecuencia y en el tiempo, dependiendo de a . Valores grandes de a dan lugar a una mayor resolución espectral, mientras que los valores pequeños proporcionan mejor resolución temporal.

Puede demostrarse que la CWT, al contrario que la STFT, permite una reconstrucción perfecta de la señal, sin ninguna ambigüedad, a partir de los coeficientes de la transformada [9]

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{t>0}^{+\infty} X_{CWT}(a, \tau) \frac{1}{\sqrt{a}} \psi\left(\frac{t-\tau}{a}\right) \frac{1}{a^2} \delta a \delta \tau \quad 1.9$$

donde C_ψ es una constante que depende únicamente de $\psi(t)$ y a es positivo. Para garantizar la reconstrucción perfecta, la constante debe cumplir

$$C_\psi = \int_{\mathbb{R}} \frac{|\Psi(\omega)|^2}{\omega} d\omega < \infty \quad 1.10$$

En la figura 1.5 se muestran las celdas tiempo-frecuencia (escala) a que dan lugar la transformada wavelet y la STFT. Estas celdas tiempo-frecuencia nos informan donde están localizadas las funciones base en el tiempo y la frecuencia. En el caso de la transformada wavelet, el escalado por octavas da lugar a que las frecuencias centrales se distribuyan conforme a una escala logarítmica, frente a la escala lineal propia de la STFT.

También se observa que la resolución en frecuencia disminuye al incrementarse la frecuencia central (las celdas son rectángulos) mientras que en el caso de la STFT se mantiene constante (las celdas son cuadrados). Resulta bastante razonable que un sistema de medida posea una resolución que sea un porcentaje constante de la cantidad que se está midiendo. De hecho, algunos sistemas fisiológicos como la córnea o la retina poseen esa característica [10].

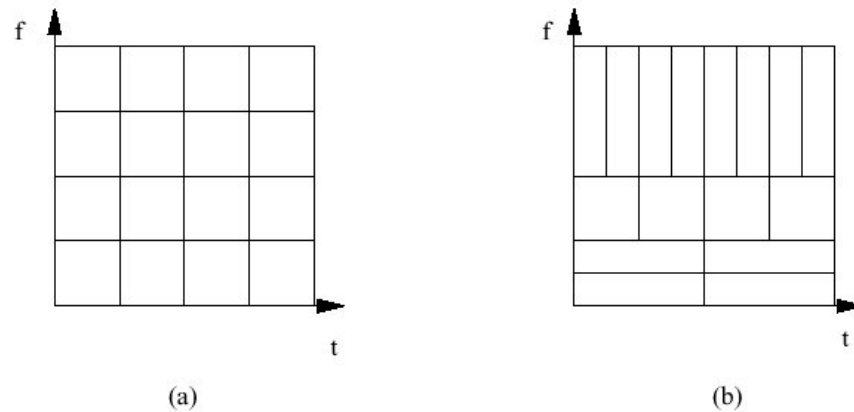


Figura 1.5. (a) La transformada STFT en el dominio del tiempo-frecuencia. (b) la transformada wavelet en el dominio tiempo-escala.

1.2.2. Transformada Wavelet Discreta

El análisis multirresolución no es una aportación de las wavelets, este análisis ya se venía utilizado en muchas áreas. En el campo del procesamiento por computadora, es usual realizar una aproximación sucesiva de las imágenes, con un aumento gradual de resolución. El conjunto formado por las imágenes de diferentes resoluciones forma una pirámide. Muchos problemas de procesamiento, como la detección de patrones, estimación del movimiento o reconocimiento de objetos, pueden resolverse de una forma eficiente con estas pirámides. La técnica de codificación de Burt y Adelson [11] sigue esta línea. En el área de gráficos por computadora, el método de refinamientos sucesivos, que va generando aproximaciones cada vez más finas de las superficies y curvas, también es una técnica multirresolución. También están muy relacionados con esta idea los métodos multigrad para la resolución de ecuaciones en diferencias parciales.

Mallat demostró que una selección inteligente de las wavelets ortogonales puede proporcionar una forma sistemática de realizar este tipo de análisis de la señal. Conforme a esta idea, Mallat definió un análisis multirresolución (MRA) [12].

Una técnica usando los filtros basados en el MRA de Mallat, usando la transformada wavelet es para la realización de codificadores de dos canales de subdanda como se ilustra en la figura 1.6.

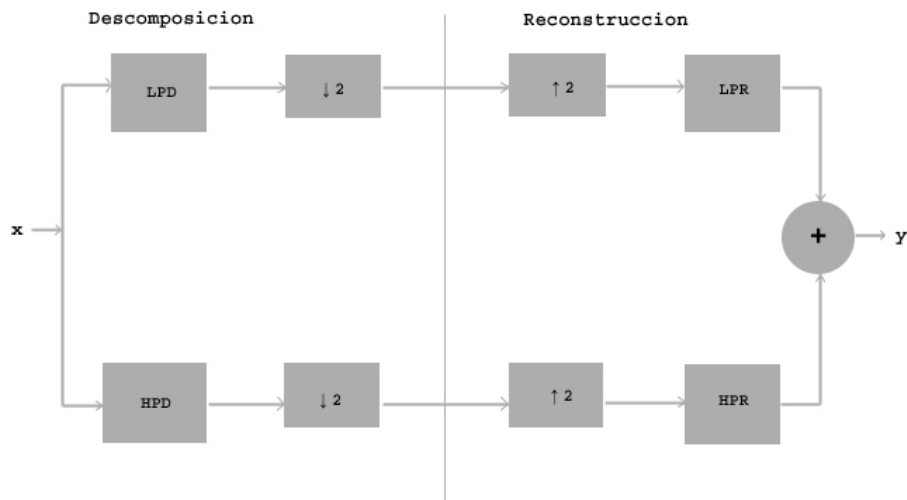


Figura 1.6. Estructura de un banco de filtros básicos para 2 canales de codificación de sub-banda con filtros de descomposición y reconstrucción paso bajas y altas.

En el estado de descomposición los filtros paso altas, ψ , y paso bajas, ϕ , obtienen componentes de alta y baja frecuencia respectivamente. Este resultado en el filtro paso bajas produce una versión aproximada de la señal original. Similarmente el filtro paso altas produce una versión detallada de la señal.

En la reconstrucción o síntesis de la señal original, la señal detallada y aproximada son primero sobremuestreadas por 2 antes de ser convolucionadas con la reconstrucción de los filtros paso bajas y paso altas. Para añadir a las salidas de reconstrucción de los filtros, la obtención de una copia de la señal original.

Transformada Wavelet en imágenes.

La DWT aplicada a imágenes proporciona una matriz de coeficientes, conocidos como coeficientes wavelet. Si a una imagen le aplicamos la DWT obtenemos cuatro tipos de coeficientes: aproximaciones, detalles horizontales, detalles verticales y detalles diagonales. La aproximación contiene la mayor parte de la energía de la imagen, es decir, la información más importante, mientras que los detalles tienen valores próximos a cero.

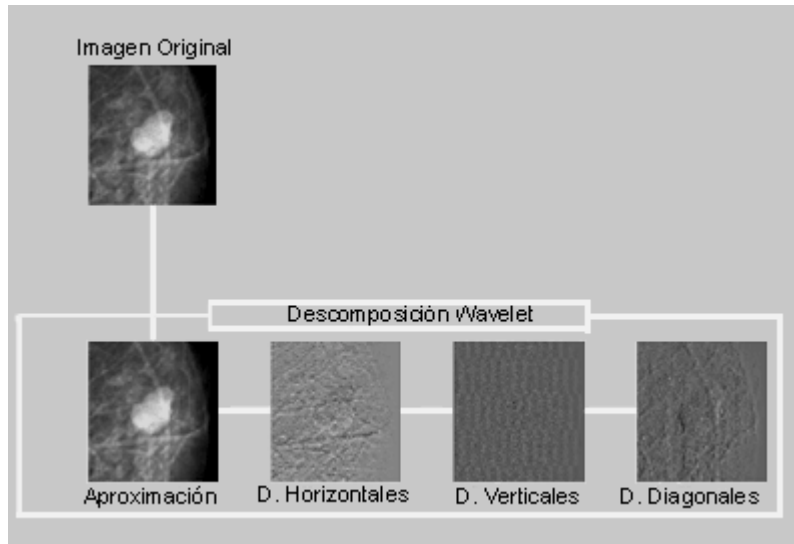


Figura 1.7 Descomposición wavelet de primer nivel.

La elección de las wavelets analizadoras juega un papel muy importante en los resultados finales. Entre las características más importantes a tener en cuenta se encuentran: soporte compacto, simetría, etc. Las wavelets biortogonales son las más eficientes para un posterior proceso de compresión, y en particular, aquellas con pocos coeficientes, ya que el coste de obtención de los coeficientes se incrementa con su número.

Generalmente, la energía de las imágenes se concentra en las frecuencias bajas. Una imagen tiene un espectro que se reduce con el incremento de las frecuencias. Estas propiedades de las imágenes quedan reflejadas en la transformada wavelet discreta de la imagen. Los niveles más bajos de compresión se corresponden con las bandas de alta frecuencia. En particular, el primer nivel representa la banda de más alta frecuencia y el nivel más fino de resolución. A la inversa, el último nivel (n) de descomposición corresponde con la banda de frecuencia más baja y la resolución más tosca. Así, al desplazarse de los niveles más altos a los más bajos, o sea, de baja resolución a alta resolución, se observa una disminución de la energía contenida en las subbandas recorridas.

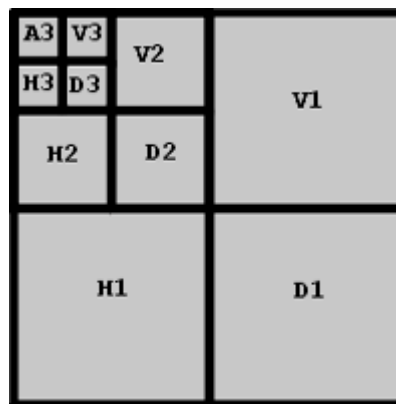


Figura 1.8 Esquema de la organización de los coeficientes wavelet.

Si los coeficientes wavelet obtenidos por medio de la transformada wavelet discreta (DWT) para un nivel concreto poseen pequeñas magnitudes (valores próximos a cero), se espera que esos coeficientes wavelet estén en los primeros niveles de descomposición. El aumento del nivel de descomposición wavelet produce unos coeficientes con mayores magnitudes. Adicionalmente, se puede comprobar como existen similitudes espaciales a través de las subbandas.



Figura 1.7 (a) Imagen original de Bárbara, (b) Organización de los coeficientes Wavelet

En la figura anterior se puede observar los contornos de Bárbara en los distintos niveles y cómo son más bastos en el primer nivel de descomposición, además de cierta similitud entre los distintos niveles.

1.2.3. Set Partitioning in Hierarchical Trees.

El SPIHT representa el comienzo de una nueva generación de codificadores wavelet que emplean un código sofisticado. Este método de compresión ha sido creado por A. Said, W. A. Pearlman, profesores del Instituto Politécnico Rensselaer y miembros del CIPR (Center for Image Processing Research).

El método de compresión *Set Partitioning In Hierarchical Trees* [13] no es una simple ampliación de los métodos tradicionales de compresión, sino que representa un gran avance en este campo. El SPIHT se aprovecha de las ventajas aportadas por la utilización de las wavelets. Al igual que el EZW permite la transmisión progresiva de la información por orden de bits más significativos, y también logra imágenes con una gran calidad y altas tasas de compresión.

1.3. Codificación de fuente temporal.

La codificación temporal de fuente, también conocida como codificación interframe, es usualmente implementada en algoritmos de codificación de videos como H.261, H.263 y MPEG. Ésta permite reducir la cantidad de redundancia temporal que esta presente entre un cuadro y el próximo. Existen diferentes métodos para realizar la codificación temporal, tales como, diferencia de cuadros, la estimación de movimiento y la predicción por compensación de movimiento. Estos métodos se discutirán en los siguientes temas.

1.3.1 Diferencia de cuadros.

La diferencia de cuadros es el simple hecho de sustraer el actual cuadro con el cuadro anterior y codificar la diferencia. Como generalmente hay poca diferencia entre dos cuadros consecutivos, sus diferencias son mínimas. La operación puede ser representada por:

$$d_t = I_t - I_{t-1} \quad 1.11$$

donde I_t e I_{t-1} es el cuadro actual y previo respectivamente, y d_t es la diferencia de cuadro.

La codificación de la diferencia puede ser hecha por la RLE, o por la DCT, seguido el resultado por una codificación de entropía. Esto es entonces transmitido hasta llegar a un decodificador, el cual lleva a cabo el proceso inverso para recobrar la diferencia. La diferencia es añadida nuevamente a la versión reconstruida del frame para obtener el frame corriente reconstruido, esto es:

$$\bar{I}_t = \bar{I}_{t-1} + d_t$$

$$\bar{I}_t = I_t + \bar{I}_{t-1} - I_{t-1},$$

$$\bar{I}_t = I_t + e_t$$

donde \bar{I}_t y \bar{I}_{t-1} son los frames reconstruidos actual y anterior respectivamente. El error entre el frame reconstruido anterior y el frame original anterior, e_t , es cero si \bar{I}_{t-1} y I_{t-1} son iguales.

1.3.2. Compensación y Estimación de Movimiento.

La compensación del movimiento es la técnica de compensación para el movimiento de píxeles, grupos de píxeles, u objetos de un frame al siguiente. Para trabajar con la compensación de movimiento, primero debe ser llevada a cabo la estimación de movimiento. La estimación de movimiento es el proceso para obtener una medida que diga al decodificador la dirección de movimiento de un píxel o de un grupo de píxeles. La medida es usualmente hecha en forma de vectores de movimiento (MV) y es usualmente aplicada a un bloque de píxeles, es decir 16x16, y puede ser expresado como:

$$e_t(x, y) = b_t(x, y) - b_{t-1}(x + u, y + v), \quad 1.12$$

donde $b_t(x, y)$ es el bloque en un tiempo t y $b_{t-1}(x + u, y + v)$ es un bloque en un tiempo $t-1$ adicionado con el vector (u, v) . El error entre ambos es $e_t(x, y)$.

Existen diversos algoritmos que se usan para operan los vectores de movimiento de los bloques (u, v) . Algunos de ellos son los métodos de búsqueda completa, búsqueda algorítmica 2-D, búsqueda jerárquica paralela 1-D, y otros método jerárquicos [14-17].

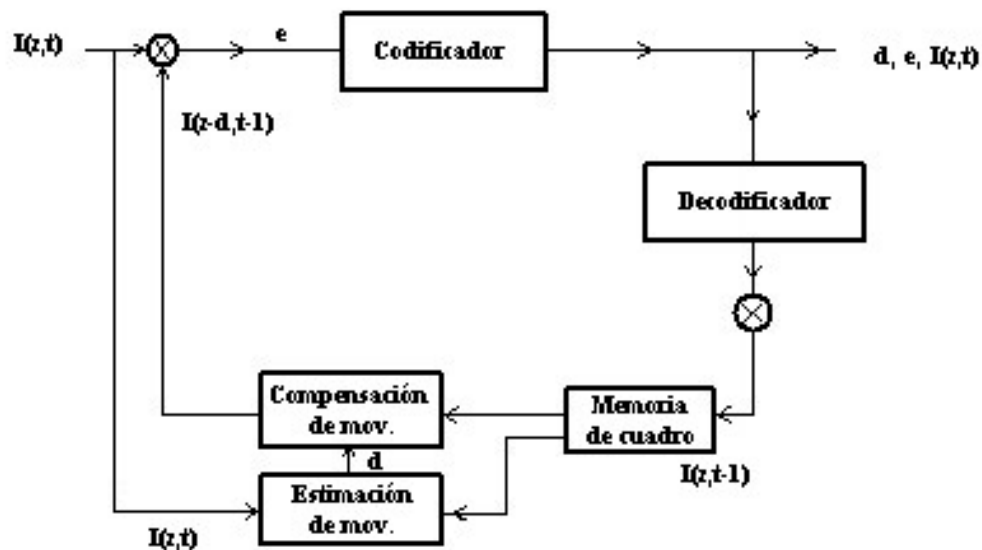


Figura 1.8 Compensación y estimación de movimiento.

Referencias

- [1] C.E. Shannon, "The mathematical theory of communication", Bell Systems Technical Journal, vol. 27, 1948.
- [2] Estándar JPEG.
- [3] D.A. Huffman "A method for the construction of minimum redundancy codes" Proceeding IRE, vol. 40, pp. 1098-1101, 1962.
- [4] M. Akay, editor. "Time Frequency and Wavelets in Biomedical Signal Processing" IEEE Press, 1996.
- [5] P.-Y. Cheng, J. Li, y C.-C. J. Kuo. "Rate control for an embedded wavelet video coder", IEEE Transactions on Circuits and Systems for Video Technology, vol. 7 no. 4, pp. 696_702, Aug. 1997.
- [6] A. S. Lewis and G. Knowles. "Image compression using the 2-D wavelet transform", IEEE Transactions on Image Processing, 1(2), April 1992.
- [7] D. Sinha y A. Tewfik. "Low bit rate transparent audio compression using adapted wavelets", IEEE Trans. Sig. Proc., vol. 41 pp. 3463_3479, 1993.
- [8] N. Saito. "Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion" En E. Foufoula-Georgiou y P. Kumar, editores, Wavelets in Geophysics. Academic Press Inc.
- [9] I. Daubechies. "Ten lectures on wavelets. Philadelphia", SIAM Press, 1992.
- [10] O. Rioul y M. Vetterli. "Wavelets and signal processing", IEEE SP Magazine, pp 14_38, Oct. 1991.
- [11] P. J. Burt y E. H. Adelson. "The laplacian pyramid as a compact image code", IEEE Transactions on Communications, vol. 31, no. 4, April 1983.
- [12] S. G. Mallat. "A theory for multiresolution signal decomposition: The wavelet representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7. pp. 674_693, July 1989.

- [13] A. Said y W. A. Pearlman, “A new fast and efficient image codec based on set partitioning in hierarchical trees”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, June 1996.
- [14] ISO/IEC/JTC1/SC29, “Generic Coding of Moving Picture and Associated Audio Information: Video”, March 1995.
- [15] V. Serferidis y M. Ghanbari, “Generalized block matching motion estimation”, in *Proceedings of SPIE: Visual Communications and Image Processing*, Junio 1992.
- [16] J.R. Jain y A.K. Jain, “Displacement measurement its applications in interframe coding”, *IEEE Transactions on Communications*, vol. 29, no. 2, pp 1799-1808, Diciembre 1981.
- [17] L.G. Chen, W.T. Chen, Y.S. Jehng y T.D. Chiueh, “An efficient parallel motion estimation algorithm for digital image processing”, *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 1, no. 4, pp 378-385, Diciembre 1991.

Capítulo 2

JPEG 2000 (Joint Photographic Experts Group 2000).

Debido a la expansión continua de aplicaciones de multimedia e Internet, las necesidades y requerimientos de las tecnologías usadas crecieron y evolucionaron, por lo que en Marzo de 1997 un nuevo llamado para contribuciones fue emitido para el desarrollo de un nuevo estándar de compresión de imágenes fijas. En Noviembre de 1997, el comité de JPEG 2000 decidió sobre una estructura que usaba la codificación de sub-banda wavelet para un nuevo estándar. La decisión fue basada en los resultados de las pruebas de 24 propuestas mostradas por varias compañías y universidades. Estos resultados mostraron que la codificación wavelet es capaz de mejorar la calidad de una imagen objetivamente y subjetivamente mejor que la Transformada Coseno Discreta (TCD).

Para facilitar el desarrollo del estándar, un modelo de verificación fue establecido, éste es básicamente la especificación de un sistema de codificación compuesto de una colección de utilerías. Este proyecto tuvo la intención de crear un nuevo sistema de codificación de imágenes para diferentes tipos de imágenes fijas (bi-nivel, escalas de grises, color, multi-componentes), con diferentes características (imágenes naturales, científicas, médicas, sensoriales remotas, de prueba, gráficas, etc.) permitiendo diferentes modelos de imagen (cliente-servidor, transmisiones en tiempo real, archivos de colecciones de imágenes, buffer limitado y recursos de ancho de banda, etc.) preferentemente dentro de un sistema unificado. Este sistema de codificación proveería operaciones a bajos índices de bits con rendimientos superiores en índices de distorsión y calidad de imagen subjetiva a comparación de otros estándares ya existentes, sin sacrificar el rendimiento de otros puntos en el espectro del índice de distorsión, incorporando al mismo tiempo muchas características importantes.

El proceso de estandarización, el cual es coordinado por la JTC1/SC29/WG1 de la ISO/IEC produjo el diseño final del estándar internacional [1,2] (FDIS por sus siglas en inglés) y el estándar internacional (IS por sus siglas en inglés) fue registrado para Diciembre del 2000. Solo los cambios editoriales son esperados a este estado y por lo tanto, no habrá más cambios en la Parte I del estándar.

El estándar JPEG 2000 [3] tiene todas las características de los viejos estándares y muchos más en un codestream y formato de archivo coherente. Específicamente éste incluye:

- codificación o decodificación sin pérdidas y con pérdidas (la decodificación sin pérdidas requiere de codificación sin pérdidas),
- progresión para ambas fidelidades por resolución y píxel seleccionables en el tiempo de decodificación,
- codificación y decodificación a tasas fijas y tamaños fijos,
- decodificación de regiones de interés,

- soporte de imágenes de color, escala de grises, gráficos, textos, bi-niveles,
- análisis gramatical del codestream sin decodificar
- acceso aleatorio y procesamiento (operaciones de rotación, traslación, filtrado, etc.) del codestream,
- diferentes estilos de codificación para los múltiples componentes,
- descripción de color,
- registro de imagen (protección de la imagen),
- inclusión de meta-datos (descripción basada en contenido),
- arquitectura abierta,
- robustez en los errores de bits,
- decodificación fuera de orden.

2.1 Estructura del estándar

El estándar JPEG2000 esta comprendido en siete partes, como se lista en la tabla 2.1.

Parte	Título
1	Sistema de codificación principal
2	Extensiones
3	Movimiento JPEG-2000
4	Conformidad
5	Software de referencia
6	Formato de imagen compuesto
7	Reporte técnico

Tabla 2.1. Partes del estándar JPEG 2000.

La parte I [4] es el algoritmo básico (núcleo) y es libre de regalías. Éste especifica la mínima funcionalidad que un codec debe proveer para ser compatible con el estándar. La parte II consiste de tecnologías opcionales no requeridas para todas las implementaciones. Evidentemente las imágenes codificadas con la tecnología de la parte II no podrán ser decodificadas por un decodificador de la parte I. La parte II incluye cuantización de codificación de malla (TCQ por sus siglas en inglés), wavelet definidas por el usuario, esquemas mejorados de capacidad de recuperación de error, escalamiento basado en métodos de codificación ROI, codificación mezclada de longitud fija y variable, etc. La parte III añade movimiento a JPEG 2000 y esta basado en la parte I. Las ventajas adicionales de esta parte son video de alta calidad, capacidad de recuperación de error en el video en situaciones de canal ruidoso tales como wireless e Internet. El formato de archivo de movimiento de JPEG 2000 (MJ2) será interoperable con el formato de archivo JPEG 2000 (JP2) y el formato de archivo de MPEG-4 MP-4. La parte IV del estándar define las pruebas de conformidad y la parte V describe los dos software's de referencia como software de alta calidad (libres). Actualmente dos implementaciones de software de referencia existen: el software JJ2000 (desarrollado por Canon Research France, EPFL y Ericsson), que es una implementación en Java del JPEG2000. El

software JasPer es una implementación en C por Image Power y la Universidad de British Columbia. La parte VI define el formato de imagen compuesto, mientras la parte VII concluye la serie de JPEG2000 con un reporte técnico con la directriz de una función de soporte mínimo de la parte I.

2.2 Arquitectura básica del estándar.

El motor de compresión de JPEG2000 (codificador y decodificador) es ilustrado en el diagrama de bloques de la figura 2.1. En el codificador, la transformada discreta es primero aplicada sobre los datos de la imagen fuente. Los coeficientes transformados son entonces cuantizados y codificados entropicamente antes de formar la salida del codestream (bitstream). El decodificador es el proceso inverso del codificador.

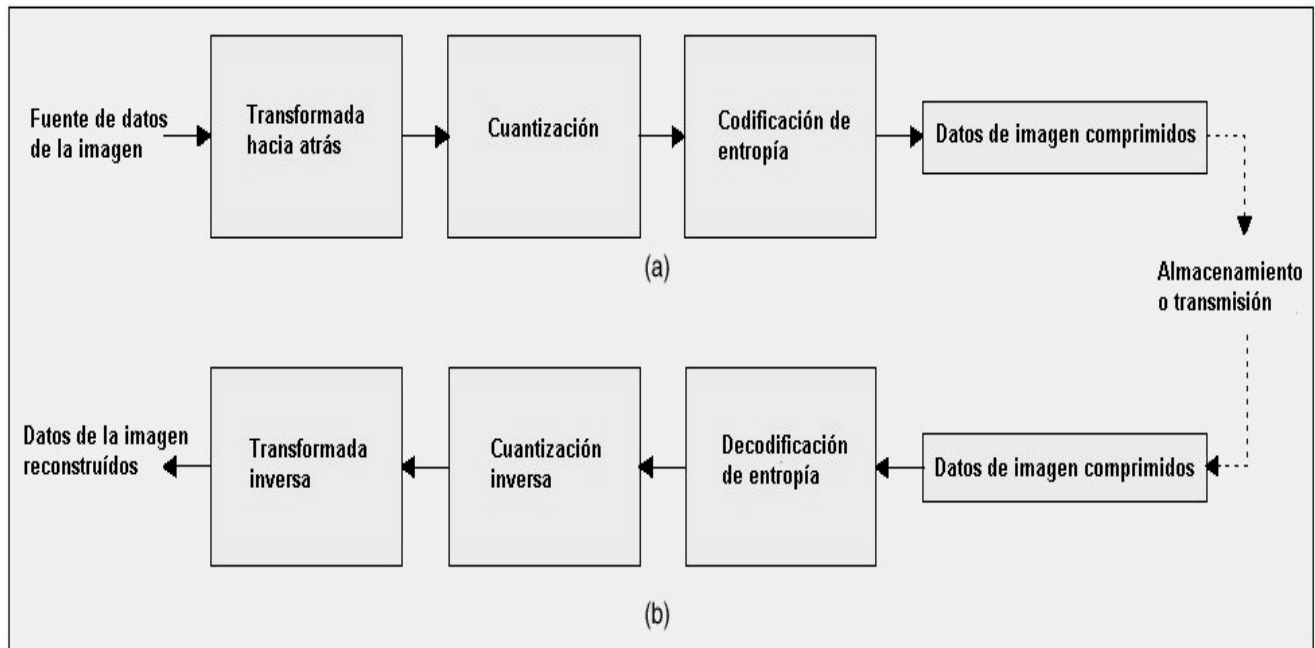


Figura 2.1. Diagrama de bloques general de JPEG2000

Aunque este diagrama de bloques general se parece a un codificador JPEG convencional, hay diferencias radicales en todos los procesos de cada bloque del diagrama. Una rápida visión general del sistema completo es la siguiente:

- La imagen fuente es descompuesta en componentes.
- Los componentes de la imagen son (opcionalmente) descompuestos en tiles rectangulares. La componente-tile es la unidad básica de la imagen original o la imagen reconstruida.
- Una transformada wavelet es aplicada sobre cada tile. El tile es descompuesto en diferentes niveles de resolución.

- Los niveles de descomposición se componen de subbandas de coeficientes que describen las características de frecuencias de las áreas locales de los componentes de los tiles.
- Las subbandas de coeficientes son cuantizadas y colocadas en arreglos rectangulares de “codeblocks”.
- Los planos de bits de coeficientes en un codeblock (por ejemplo, los bits de igual importancia a través de los coeficientes en un codeblock) son codificados entropicamente.
- La codificación puede ser hecha de tal forma que ciertas regiones de interés puedan ser codificadas a calidad mayor que el resto.
- Los marcadores son añadidos al bitstream para permitir la capacidad de recuperación de error (error resilience).
- El codestream tiene un encabezado principal al inicio que describe la imagen original y varios estilos de descomposición y codificación que son usados para localizar, extraer, decodificar y reconstruir la imagen con las resoluciones, fidelidad, regiones de interés u otras características deseadas.
- El formato de archivo opcional describe el significado de la imagen y sus componentes en el contexto de la aplicación.

Habría que hacer notar que el motor de codificación básico de JPEG2000 esta basado en el algoritmo de codificación de bloque incrustado con un truncamiento optimizado de bitstreams embedidos (EBCOT por sus siglas es inglés) [5,6].

2.2.1 Modelo de imagen fuente.

Antes de analizar internamente el codec, es importante entender el modelo de imagen que el codec empleará. Desde el punto de vista de un codec, una imagen esta compuesta de una o más componentes (hasta un límite de 2^{14}) como se muestra en la figura 2.2a. Como se ilustra en la figura 2.2b cada componente consiste de arreglos rectangulares de muestras. Los valores de las muestras de cada componente son de valor entero, y pueden ser ambas con signo o sin signo con una precisión de 1 a 128 bits/muestra. El signo y la precisión de los datos muestra son especificados sobre una base por componente.

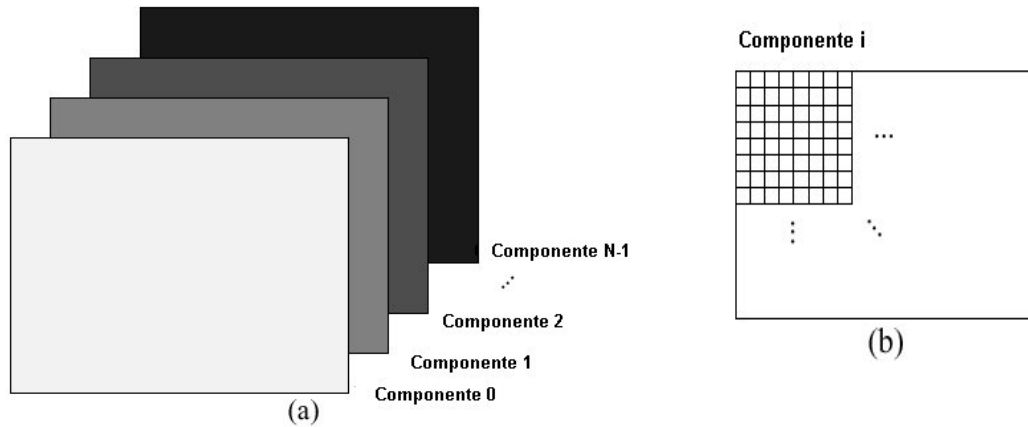


Figura 2.2 Modelo de imagen de fuente a) imagen con N componentes b) componente individual.

Todos los componentes son asociados con la misma dimensión espacial en la imagen fuente, pero representan diferentes espectros o información auxiliar. Por ejemplo, una imagen de color RGB tiene tres componentes, donde una componente representa cada uno de los planos de color rojo, verde y azul. En el caso de una imagen en escala de grises, hay solo una componente, que corresponde al plano de luminancia. Las diversas componentes de una imagen no necesitan ser muestreadas con la misma resolución, por lo cual las componentes por sí mismas pueden tener diferentes tamaños.

2.2.2 Cuadrícula de referencia

Dada una imagen el codec describe la geometría de las diversas componentes en términos de una cuadrícula rectangular llamada cuadrícula de referencia. La cuadrícula de referencia tiene la forma general mostrada en la figura 2.3. La cuadrícula es de tamaño $X_{siz} \times Y_{siz}$ con el origen localizado en su esquina izquierda superior. La región de la esquina izquierda superior ($X_{0siz} \times Y_{0siz}$) a la esquina derecha inferior ($X_{siz-1} \times Y_{siz-1}$) es llamada el área de imagen, y corresponde a los datos de la pintura a representar. El ancho y el largo de la cuadrícula de referencia no puede exceder $2^{23}-1$ unidades, imponiendo un límite superior en el tamaño de una imagen que puede ser tratado por el codec.

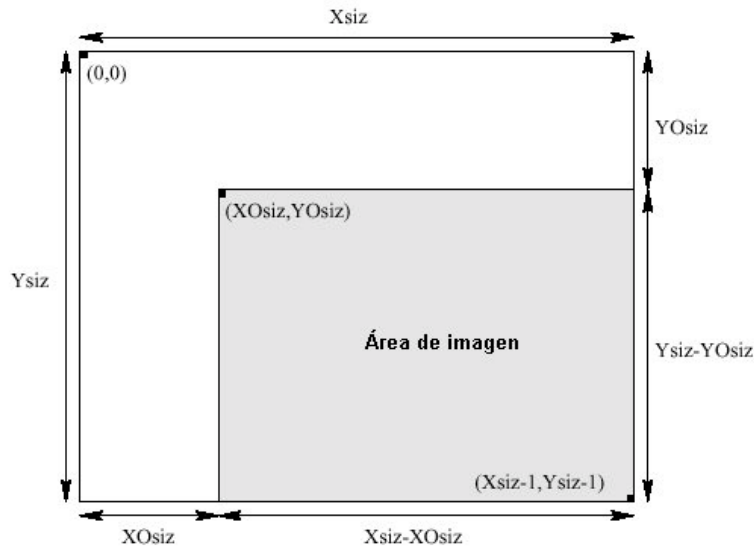


Figura 2.3 Cuadrícula de referencia.

Todos los componentes son representados sobre el área de imagen de la cuadrícula de referencia. Puesto que los componentes no necesitan ser muestreados a la resolución completa de la cuadrícula de referencia, información adicional es requerida para establecer su correspondencia. Para cada componente se indica el período de muestreo horizontal y vertical en unidades de la cuadrícula de referencia, denotadas como XR_{siz} y YR_{siz} , respectivamente. Estos dos parámetros únicamente especifican una cuadrícula de muestreo (rectangular) constando de todos los puntos cuyas posiciones horizontales y verticales son múltiplos enteros de XR_{siz} y YR_{siz} , respectivamente. Todos los puntos que caigan dentro del área de imagen, constituyen muestras de la componente en cuestión. Así, en términos de la del sistema de coordenadas, una componente tendrá el

tamaño $\left(\left\lceil \frac{X_{siz}}{XR_{siz}} \right\rceil - \left\lfloor \frac{X0_{siz}}{XR_{siz}} \right\rfloor \right) \times \left(\left\lceil \frac{Y_{siz}}{YR_{siz}} \right\rceil - \left\lfloor \frac{Y0_{siz}}{YR_{siz}} \right\rfloor \right)$ y su muestra izquierda superior corresponde al punto $\left(\left\lfloor \frac{X0_{siz}}{XR_{siz}} \right\rfloor, \left\lfloor \frac{Y0_{siz}}{YR_{siz}} \right\rfloor \right)$.

Del diagrama, el tamaño del área de imagen es $(X_{siz}-X0_{siz}) \times (Y_{siz}-Y0_{siz})$. Para una imagen dada, muchas combinaciones de los parámetros X_{siz} , Y_{siz} , $X0_{siz}$ y $Y0_{siz}$ pueden ser escogidos para obtener un área de imagen con el mismo tamaño. Los parámetros $X0_{siz}$ y $Y0_{siz}$ no son fijos a cero mientras que los parámetros X_{siz} y Y_{siz} son conjunto para el tamaño de la imagen. Por lo que hay sutiles implicaciones para cambiar los parámetros $X0_{siz}$ y $Y0_{siz}$ (guardando el tamaño del área de imagen constante). Tales cambios afectan las características del codec en diversas formas importantes. Estas características permiten un número de operaciones básicas para ser llevadas a cabo eficientemente en las imágenes codificadas tales como cortado, flipping horizontal/vertical, y rotación por un entero múltiple de 90 grados.

2.2.3 Tiling

El término tiling se refiere a la partición de la imagen fuente original dentro de bloques rectangulares no sobrepuestos (tiles) sobre la cuadrícula de referencia, los cuales son comprimidos independientemente como si fueran imágenes enteramente distintas. Todas las operaciones incluyendo la mezcla de componentes, la transformada wavelet, la cuantización y codificación de entropía son llevadas a cabo independientemente de los tiles imagen como se muestra en la figura 2.4. La componente tile es la unidad básica de la imagen original o reconstruida. El tiling reduce los requerimientos de memoria y puesto que son también reconstruidos independientemente, pueden ser usados para decodificar partes específicas de la imagen en lugar de toda la imagen. Todos los tiles tienen exactamente las mismas dimensiones, excepto quizás los localizados en la frontera de la imagen. Tamaños arbitrarios de tiles son permitidos hasta incluir la imagen entera o a ella misma, es decir un tile puede ser la imagen entera.

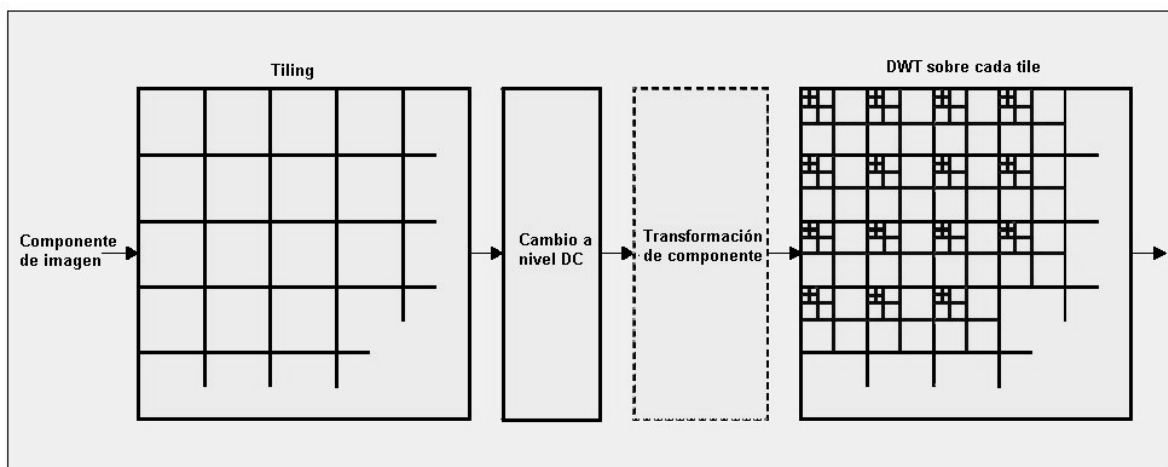


Figura 2.4. Tiling, cambio a nivel DC, transformada de color (opcional) y DWT de cada componente de la imagen.

Los componentes con diferentes factores de submuestreo son tileados con respecto a una cuadrícula de alta resolución, la cual asegura la consistencia espacial sobre los componentes tile resultantes. Como es de esperar el tiling afecta la calidad de la imagen objetivamente y subjetivamente. Tiles más pequeños crean más artefactos tiling comparados con tiles más grandes (los valores de PSNR son el promedio de todos los componentes). En otras palabras, los tiles más grandes funcionan visualmente mejor que los tiles más pequeños. La degradación es más severa en el caso de índices de bits bajos que en el caso de índices de bits altos. Esto se puede observar en la figura 2.5.



Figura 2.5 Imagen original y con tiling.

2.2.4 Cambio a nivel DC

El codec espera que sus datos de muestra de entrada tengan un rango dinámico nominal que este simétricamente distribuido alrededor del cero. El estado de preprocesamiento del codificador simplemente asegura que esta esperanza sea conocida. Supóngase que una componente particular tiene P bits/muestra. Las muestras pueden ser con signo o sin signo, conducidas a un rango dinámico nominal de $[-2^{P-1}, 2^{P-1}-1]$ y $[0, 2^{P-1}]$, respectivamente. Si los valores de las muestras son sin signo, el rango dinámico nominal claramente es no simétrico alrededor del cero. Por lo que prioritario a la operación de la transformada wavelet discreta (DWT por sus siglas en inglés) sobre cada tile imagen, todas las muestras de la componente tile imagen son cambiadas al nivel DC por la sustracción de la misma cantidad del sesgo 2^{P-1} , donde P es la precisión de componente. El cambio de nivel DC solamente es llevado a cabo sobre las muestras sin signo de las componentes. El cambio de nivel no afecta las variancias. Esto actualmente convierte una representación sin signo a una representación de complemento a dos, o viceversa. Si la transformada de color es usada, el cambio a nivel DC es llevada a cabo antes de la operación de la transformada de componente hacia delante como se muestra en la figura 2.4 y 2.6. En el lado del decodificador el cambio de nivel DC inverso es llevado a cabo sobre las muestras reconstruidas por la adición a ellas del sesgo de 2^{P-1} después de la operación de la transformada de componente inversa.

2.2.5 Transformación de componentes

JPEG 2000 soporta imágenes de múltiples componentes. Por lo que en el codificador, el estado de preprocesamiento es seguido por un estado de transformación de componentes, que opera sobre todos los componentes juntos y sirve para reducir la correlación entre los componentes permitiendo una codificación más eficiente.

El estándar soporta dos diferentes transformaciones de componentes, una transformación de componente irreversible (ICT por sus siglas en inglés) que puede ser usada para codificación con pérdidas y una transformación de componentes reversibles (RCT por sus siglas en inglés) que puede ser utilizada para codificación con pérdidas y sin pérdidas. El diagrama de bloques del codificador de multicomponentes de JPEG2000 es descrito en la figura 2.6.

Las transformadas son definidas para operar sobre las tres primeras componentes de una imagen, con la suposición que los componentes 0, 1 y 2 corresponden a los planos de color rojo, verde y azul. Debido a la naturaleza de estas transformadas, las componentes en las cuales las transformadas operan deben ser muestreadas a la misma resolución. Como consecuencia de los hechos de arriba, la ICT y RCT solamente pueden ser empleados cuando la imagen codificada tiene por lo menos tres componentes, y las primeras tres componentes son muestreadas a la misma resolución. Después del estado de transformación de componentes en el codificador, los datos de cada componente son tratados independientemente.

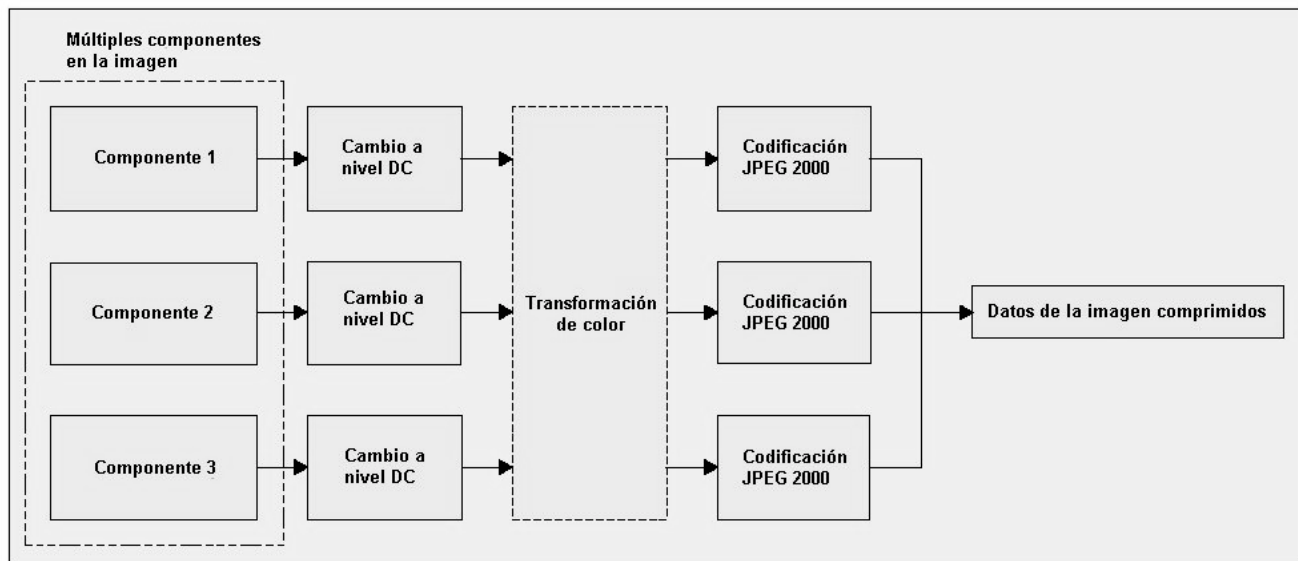


Figura 2.6 Codificador de componente múltiple de JPEG 2000. La transformación de color es opcional. Si es empleada esta puede ser reversible o irreversible.

La ICT es nada más que la transformada de espacio color RGB a YCrCb clásica. Esta transformada solamente puede ser usada con la transformada wavelet irreversible 9/7. La transformada hacia delante es definida como:

$$\begin{pmatrix} \gamma \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

La transformada inversa es mostrada por:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & -0.71414 \\ 1.0 & 1.772 & 0 \end{pmatrix} \cdot \begin{pmatrix} \gamma \\ C_b \\ C_r \end{pmatrix}$$

La RCT es simplemente una aproximación entero a entero reversible, ésta además solo puede ser usada con la transformada wavelet reversible 5/3. La transformada hacia delante es dada por:

$$\begin{pmatrix} \gamma \\ V \\ U \end{pmatrix} = \begin{pmatrix} \left[\frac{R+2G+B}{4} \right] \\ R-G \\ B-G \end{pmatrix}$$

La transformada inversa es dada por:

$$\begin{pmatrix} G \\ R \\ B \end{pmatrix} = \begin{pmatrix} \gamma - \left[\frac{U+V}{4} \right] \\ V+G \\ U+G \end{pmatrix}$$

Si la transformada de multicomponentes fue aplicada durante la codificación, su inversa es aplicada en la decodificación. A menos que la transformada sea irreversible, sin embargo, la inversión solo puede ser aproximada debido a los efectos de la aritmética de precisión finita.

2.2.6 Transformada Wavelet

El estándar JPEG2000 emplea una transformada wavelet discreta (DWT por sus siglas en inglés) para operar sobre los componentes individuales. A través de esta transformada, una componente tile es dividida en numerosas bandas de frecuencias (subbandas). Estas subbandas consisten de coeficientes que describen la frecuencia espacial horizontal y vertical características de la componente tile original. Debido a las propiedades estadísticas de estas señales de subbandas, los datos transformados pueden ser codificados más eficientemente que los datos originales sin transformar.

Ambas transformadas DWT reversible e irreversible pueden ser empleadas en el estándar y este último soporta dos modos de filtrado: el basado en convolución y el basado en levantamiento. En ambos modos para facilitar el filtrado en las fronteras de la señal se emplea una extensión simétrica periódica [7]. El bloque de construcción básica

usando levantamiento para cada transformada es la descomposición de subbanda de una dimensión (1-D) por 2 canales (muestras paso altas y paso bajas) del banco de filtros máximamente decimados uniformemente (UMDFB por sus siglas en inglés) cuya forma general se muestra en la figura 2.7. El lado de análisis del UMDFB descrito en 6a es asociado con la transformada hacia delante y el descrito en la figura 2.6b es asociado con la transformada inversa. En el diagrama $\{A_i(z)\}_{i=0}^{\lambda-1}$, $\{Q_i(z)\}_{i=0}^{\lambda-1}$ y $\{s_i(z)\}_{i=0}^{\lambda-1}$ denotan la función de transferencia del filtro, el operador de cuantización y la ganancia (escalar), respectivamente. Como una imagen es una señal de 2-D, claramente renecesitaría un UMDFB de 2-D, pero en la aplicación de un UMDFB de 1-D de manera separada, se obtiene eficientemente un UMDFB de 2-D. Entonces la transformada wavelet es calculada por la aplicación recursiva de un UMDFB de 2-D para las señales de subbanda paso bajas obtenidas de cada nivel de descomposición. Una muestra paso bajas representa un muestreo bajo, una versión de resolución baja del conjunto original; mientras una muestra paso altas representa un versión residual de muestreo bajo del conjunto original, necesitada para la reconstrucción perfecta del conjunto original del conjunto paso bajas.

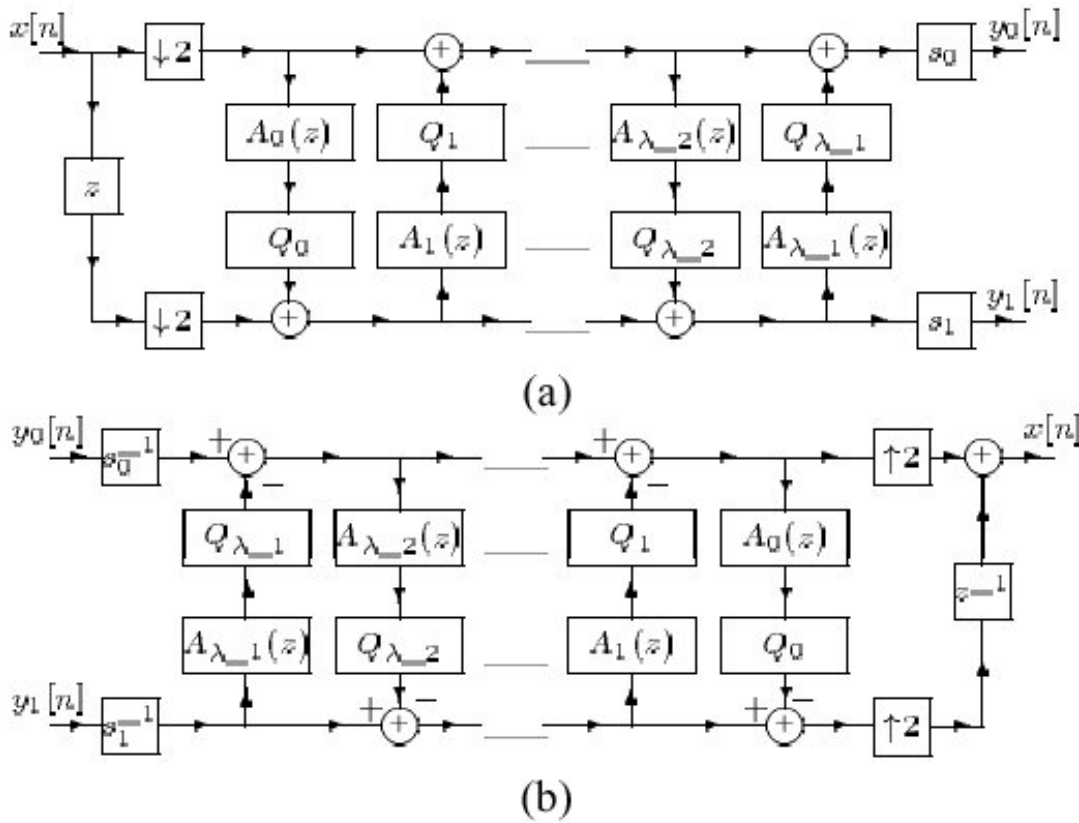


Figura 2.7. Realización de UMDFB de 2 canales de 1-D. a) lado de análisis b) lado de síntesis.

Supóngase que una transformada wavelet de nivel ($R-1$) es aplicada a los datos de una componente tile, los cuales pasan de una manera iterativa a través de los filtros, produciendo señales de subbandas. Cada aplicación del lado de análisis de los filtros produce cuatro subbandas: una paso bajas horizontal y vertical (LL), una paso bajas horizontal y paso altas vertical (LH), una paso altas horizontal y paso bajas vertical (HL)

y una paso altas horizontal y vertical (HH). Una descomposición wavelet de nivel ($R-1$) es asociada con R niveles de resolución, numerados del 0 hasta $R-1$, con 0 y $R-1$, correspondiendo a las resoluciones más gruesas y más finas, respectivamente. Cada subbanda de la descomposición es identificada por su orientación (por ejemplo LL, LH, HL, HH) y su correspondiente nivel de resolución (por ejemplo $0,1,\dots,R-1$). La señal de componente tile de entrada es considerada para ser una banda LL_{R-1} . A cada nivel de resolución (excepto la más baja) de la banda LL es otra vez descompuesta. Por ejemplo, la banda LL_{R-1} es descompuesta para producir las bandas LL_{R-2} , LH_{R-2} , HL_{R-2} y HH_{R-2} . Entonces para el próximo nivel, la banda LL_{R-2} es descompuesta y así por el estilo. Este proceso se repite hasta obtener la banda LL_0 y resulta en una estructura de subbandas como la que se ilustra en la figura 2.8.

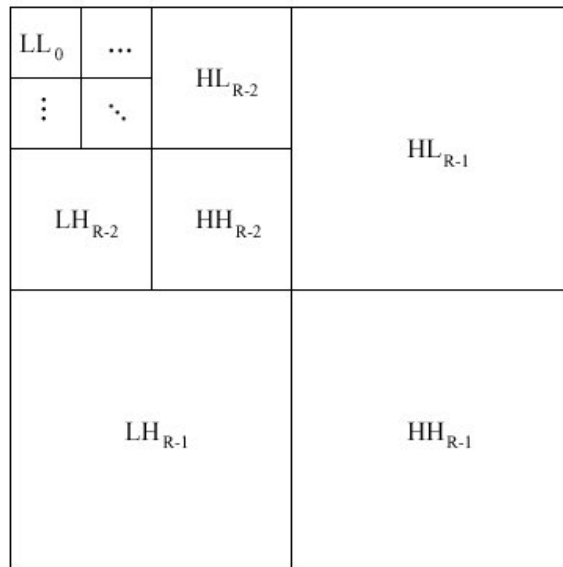


Figura 2.8. Estructura de subbandas.

Teniendo la estructura de transformación general se pueden describir las dos transformadas soportadas por el estándar, la transformada 5/3 y 9/7. La transformada 5/3 es reversible, entero a entero y no lineal. Esta transformada fue propuesta en [8], y es simplemente una aproximación a una transformada wavelet lineal propuesta en [9] por Le Gall. Como la transformada 5/3 es reversible puede ser empleada para ambas codificaciones con pérdidas y sin pérdidas.

La transformada 9/7 es no reversible y real a real. Sin embargo es también lineal en aritmética exacta. Esta transformada propuesta en [10] por Daubechies, es empleada en el estándar de compresión de huellas digitales del FBI. La transformada 9/7, carente de la propiedad reversible, puede ser solamente usada para codificación con pérdidas. El número de niveles de resolución es un parámetro de cada transformada. Un valor típico para este parámetro es seis (asumiendo una imagen lo suficientemente grande).

La transformada inversa wavelet se realiza en el decodificador, pero debido a los efectos de la aritmética de precisión finita, el proceso inverso no es garantizado para ser exacto si la transformada reversible fue empleada.

2.2.7 Cuantización

Después de la transformación, todos los coeficientes son cuantizados. La cuantización permite mayor compresión en la representación de los coeficientes de la transformada con solo la precisión mínima requerida para obtener el nivel deseado de calidad en la imagen. En otras palabras, la cuantización facilita una compresión mayor por el descartamiento de información visualmente insignificante. La cuantización es una de las dos fuentes primarias de pérdidas de información en la ruta de la codificación.

Los coeficientes transformados son cuantizados usando una cuantización escalar con una zona muerta cerca del origen en la parte I del estándar, y una cuantización vectorial de codificación de malla en la parte II del estándar. Un diferente cuantizador es empleado para los coeficientes de cada subbanda, y cada cuantizador tiene solo un parámetro, su tamaño de paso. Matemáticamente en el estándar el proceso de cuantización es definido por:

$$q_b(u, v) = \text{sign}(a_b(u, v)) \cdot \left[\frac{|a_b(u, v)|}{\Delta_b} \right]$$

donde Δ_b es el tamaño del paso de cuantización, $a_b(u, v)$ son los coeficientes de transformación de la subbanda b y $q_b(u, v)$ es la salida del cuantizador de la subbanda.

En el caso de una compresión con pérdidas el tamaño del paso del cuantizador es elegido en conjunción con el control de índice. En el caso sin pérdidas, el tamaño del paso del cuantizador es forzado a ser uno. El tamaño del paso de cuantización es especificado relativo al rango dinámico de subbanda b .

En el decodificador, la decuantización intenta deshacer los efectos de la cuantización. Si los coeficientes de la transformada son enteros y el tamaño del paso de cuantización son todos iguales a uno (como en el caso de codificación sin pérdidas), habrá pérdida de información, ya que el proceso de inversión es solamente aproximado. Los valores de los coeficientes cuantizados son obtenidos de los índices del cuantizador. Matemáticamente, el proceso de decuantización es definido como:

$$a_b(u, v) = (q_b(u, v) + r \text{sign}(q_b(u, v))) \cdot \Delta_b$$

donde r es un parámetro de sesgo. Aunque el valor de r normativamente no es especificado en el estándar, probablemente muchos decodificadores usaran el valor de un medio.

2.2.8 Codificación de entropía

La codificación de entropía es lograda por medio de un sistema de codificación aritmética que comprime símbolos binarios relativos a un modelo de probabilidad adoptiva asociado con cada 18 contextos de codificación diferente. El algoritmo de codificación MQ es usado para llevar a cabo esta tarea y administrar la adaptación de los modelos de probabilidad condicional. Este paso se describe más adelante.

2.2.9 Recintos

Después de la cuantización, cada subbanda es dividida dentro de bloques rectangulares, es decir rectángulos no sobrepuestos. El cuantizador indica para cada subbanda la repartición en bloques de código (codeblocks). Los codeblocks son rectangulares en forma, y su tamaño nominal es un parámetro libre del proceso de codificación, conforme a ciertas limitaciones, más notablemente: la anchura nominal y la altura del codeblock debe ser un número entero con potencia de dos, y el producto de la anchura nominal y altura no puede exceder de 4096.

Supóngase que el tamaño nominal del codeblock es tentativamente elegido para ser $2^{xcb} \times 2^{ycb}$. Los codeblocks se agrupan en lo que se llaman recintos. Puesto que los codeblocks no pueden cruzar los límites del recinto, una reducción en el tamaño nominal del codeblock puede ser requerida si el tamaño del recinto es suficientemente pequeño. Supóngase que el tamaño nominal del codeblock después de cualquier ajuste es $2^{xcb'} \times 2^{ycb'}$ donde $xcb' \leq xcb$ y $ycb' \leq ycb$. La subbanda es repartida en bloques de código (codeblocks) sobreponiendo la subbanda con una cuadrícula rectangular que tiene espaciamientos horizontales y verticales de $2^{xcb'}$ x $2^{ycb'}$ respectivamente, según lo mostrado en figura 2.9. El origen de esta cuadrícula se ancla en (0,0) en el sistema de coordenadas de la subbanda. Una opción típica para el tamaño nominal del codeblock es 64x64 (es decir, $xcb=6$ y $ycb=6$) y no menos de 32x32.

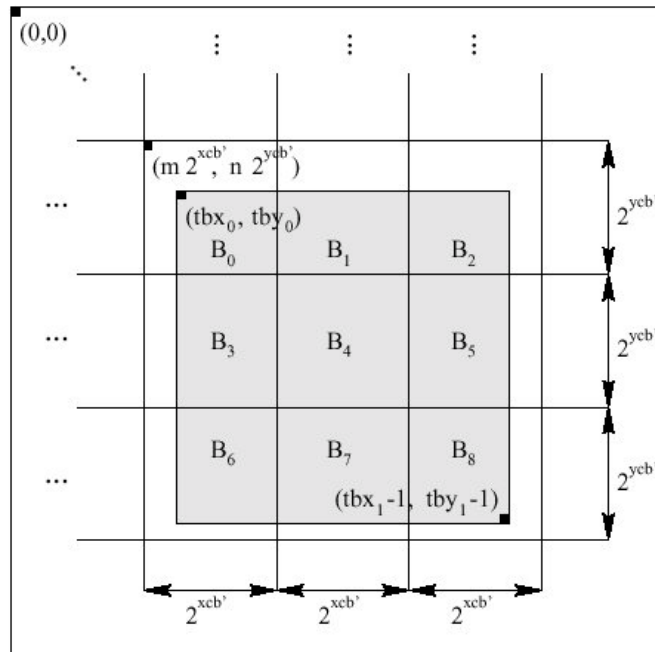


Figura 2.9. Repartición de las subbandas en codeblocks y recintos.

Después de que una subbanda ha sido repartida en codeblocks, cada uno de los codeblocks se codifica independientemente. La codificación se realiza usando un codificador de plano de bit (bit-plane) descrito más adelante. Para cada codeblock, se produce un código embebido (embedded), compuesto de numerosos pasos de codificación.

En el lado del decodificador, los pasos de codificación del bit-plane para varios codeblocks, son decodificados, y los datos que resultan son reunidos en subbandas. De esta manera, se obtienen los índices reconstruidos del cuantizador para cada subbanda. En el caso de una codificación con pérdidas, los índices reconstruidos del cuantizador pueden ser solamente aproximaciones de los índices del cuantizador originalmente disponibles en el codificador. Esto es atribuible al hecho de que el codestream puede incluir solamente un subconjunto de pasos de codificación generados en la codificación. En el caso sin pérdidas, los índices reconstruidos del cuantizador deben ser iguales a los índices originales del codificador, puesto que todos los pasos de codificación deben ser incluidos en la codificación sin pérdidas.

2.2.10 Codificación del bit-plane

Después de que todas las subbandas se han repartido en los bloques del código (codeblocks), cada uno de los codeblocks que resultan se codifican independientemente usando un codificador bit-plane. Aunque la técnica de codificación del bit-plane empleada es similar a las usadas en los codecs de embedded zerotree wavelet (EZW) y en set partitioning in hierarchical trees (SPIHT), hay dos diferencias notables: no se explota ninguna dependencia de interbanda, y hay tres pasos de codificación por bit-plane en vez de dos. La primera diferencia sigue el hecho de que cada codeblock está contenido totalmente dentro de una sola subbanda, y los codeblocks son codificados independientemente uno del otro. No explotando las dependencias de interbanda, mejorando la recuperación de error que puede ser alcanzada. La segunda diferencia es discutiblemente menos fundamental. Usar tres pasos por bit-plane en vez de dos reduce la cantidad de datos asociados a cada paso de codificación, facilitando un control más fino sobre el índice de cambio.

Los tres pasos de codificación por bit-plane en orden son: significación, refinamiento y limpieza. Los tres tipos de pasos de codificación exploran las muestras de un codeblock en el mismo orden fijo mostrado en la figura 2.10. El codeblock se divide en franjas horizontales, cada una tiene una altura nominal de cuatro muestras. Si la altura del codeblock no es un múltiplo de cuatro, la altura de la franja inferior será menor que el valor nominal. Según lo mostrado en el diagrama, las franjas se exploran de arriba hacia abajo. Dentro de una franja, las columnas se exploran de izquierda a derecha. Dentro de una columna, las muestras se exploran de arriba hacia abajo. Es decir, empezando de la esquina izquierda superior, los primeros cuatro bits de la primera columna se exploran. A continuación los primeros cuatro bits de la segunda columna, hasta que el ancho del codeblock se cubre. Entonces los segundos cuatro bits de la primera columna se exploran y así por el estilo. Una exploración vertical similar continúa para cualquier fila sobrante en los codeblocks más bajos de la subbanda [11]. Esta altura de 4 en la franja se ha seleccionado cuidadosamente para facilitar implementaciones eficientes de hardware y software.

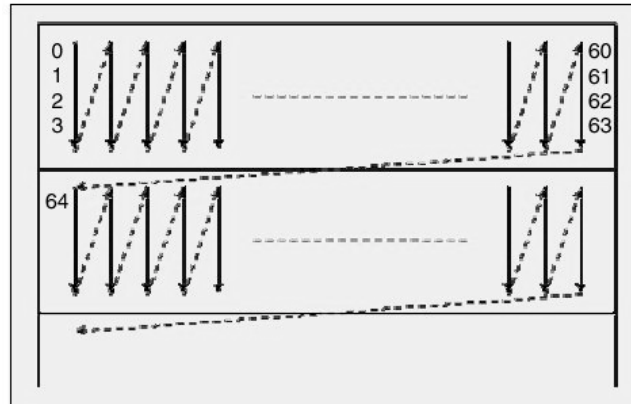


Figura 2.10. Orden de exploración de las muestras en un codeblock.

Cada paso de codificación genera una secuencia de símbolos. Algunos o todos estos símbolos pueden ser codificados entropíicamente. Para los propósitos de la codificación de entropía, un codificador aritmético adaptativo basado en contexto es usado, más específicamente, el codificador MQ del estándar JBIG2 [12]. Para cada paso, todos los símbolos son codificados aritméticamente, o no codificados (es decir, los símbolos binarios se emiten como bits sin procesar). Los pasos de limpieza emplean siempre la codificación aritmética. En el caso de los pasos de significación y refinamiento, existen dos posibilidades, dependiendo de si el modo llamado lazy está permitido. Si el modo lazy está habilitado, sólo los pasos de significación y refinamiento para los cuatro bit-planes más significativos utilizan la codificación aritmética, mientras que los pasos restantes son sin codificar. En otro caso, todos los pasos de significación y refinamiento son codificados aritméticamente. El modo lazy permite complejidad computacional de la codificación del bit-plane para ser significativamente reducido, disminuyendo el número de los símbolos que deben ser codificados aritméticamente. Esto viene, por supuesto, en el costo de codificación eficiente.

Si los múltiples pasos de codificación consecutivos son codificados aritméticamente, pueden formar un solo codeword o cada paso puede formar un codeword separado. Estos casos son determinados por el modo de terminación en efecto. Dos modos de terminación son soportados: la terminación por paso y la terminación por segmento. En el primer caso, solamente el paso pasado de un segmento es terminado. En el segundo caso, se terminan todos los pasos de codificación. Terminar todos los pasos de codificación facilita el mejoramiento de la recuperación de error.

Puesto que se emplea la codificación aritmética basada en contexto, un medio para la selección del contexto es necesario. La selección del contexto es realizada examinando la información de estado de los 4 vecinos conectados o los 8 vecinos conectados de una muestra según lo mostrado en figura 2.11.

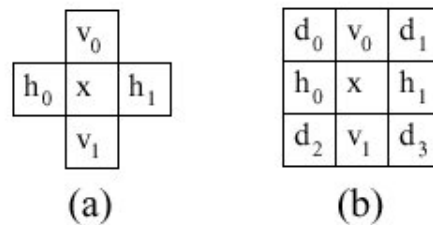


Figura 2.11. Plantilla para la selección de contexto. a) 4 vecinos conectados b) 8 vecinos conectados.

Paso de significación

El primer paso de codificación para cada bit-plane es el paso de la significación. Este paso se utiliza para transportar la trascendencia y la información de signo de las muestras que aun no se han encontrado significativas y se predicen para llegar a ser significativas durante el proceso del actual bit-plane. Las muestras en el codeblocks se exploran en la orden demostrada previamente. Si una muestra todavía no se ha encontrado significativa, y se predice para llegar a ser significativa, la trascendencia de la muestra se codificada con un solo símbolo binario. Si la muestra también pasa a ser significativa, su signo es codificado usando un solo símbolo binario.

Si se está procesando el bit-plane más significativo, todas las muestras se pronostican para continuar siendo insignificantes. Si no, una muestra es pronosticada significativa si cualquier vecino de los 8 conectados ha sido encontrado significativo. Como consecuencia de esta política de predicción, los pasos de significación y refinamiento para el bit-plane más significativo son siempre vacíos (y no necesite ser codificado explícitamente).

Los símbolos generados durante el paso de significación pueden o no pueden ser codificados aritméticamente. Si se emplea la codificación aritmética, el símbolo binario que transporta la información de trascendencia se codifica usando uno de nueve contextos. El contexto particular usado se selecciona basándose en la trascendencia de los 8 vecinos conectados de la muestra y en la orientación de la subbanda con la cual la muestra es asociada (por ejemplo, LL, LH, HL, HH). En el caso que la codificación aritmética es utilizada, el signo de una muestra es codificado como la diferencia entre la signo actual y predicho. Si no, el signo es codificado directamente. La predicción de signo se realiza usando la trascendencia y la información de signo para los cuatro vecinos conectados.

Paso de refinamiento

El segundo paso de codificación para cada bit-plane es el paso de refinamiento. Este paso señala los bits subsecuentes después del bit más significativo de cada muestra. Las muestras del codeblock son exploradas usando el orden mostrado anteriormente. Si una muestra es encontrada significativa en un bit-plane anterior, el bit-plane más significativo siguiente de esa muestra se transmite usando un solo símbolo binario.

Como en el paso de significación, los símbolos del paso de refinamiento pueden ser o no codificados aritméticamente. Si se emplea la codificación aritmética, cada símbolo del refinamiento es codificado usando uno de tres contextos. El contexto particular empleado se selecciona basándose en si se está refinando la segunda posición del MSB y la trascendencia de los 8 vecinos conectados.

Paso de limpieza

El tercer paso de codificación para cada bit-plane es el paso de la limpieza. Este paso se utiliza para transportar la trascendencia e información de signo para esas muestras que todavía no se han encontrado significativas y se pronostican para continuar como insignificantes durante el procesamiento del bit-plane actual.

Conceptualmente, el paso de limpieza no es muy diferente al paso de significación. La diferencia clave es que el paso de limpieza transporta la información sobre las muestras que se predicen para llegar a ser significativas, más bien las que se pronostiquen para seguir siendo insignificantes. Algorítmicamente, sin embargo, hay una diferencia importante entre el paso de limpieza y el de significación. En el caso del paso de limpieza, las muestras se procesan a veces en grupos, más bien que individualmente como en el paso de significación.

Recordando el patrón de exploración de las muestras en un codeblock, el paso de limpieza es mejor explicado como mantenimiento de la exploración vertical. El paso de limpieza procesa simplemente cada una de las exploraciones verticales en orden, cada exploración vertical es procesada como sigue. En el caso en que la exploración vertical contiene cuatro muestras (es decir, exploración completa), la información trascendental es necesaria para todas las muestras, y todas las muestras se pronostican para seguir siendo insignificantes, un modo especial, llamado modo de agregación, es presentado. En este modo, el número de muestras insignificantes al frente en la exploración vertical se codifican. Entonces, las muestras cuya información trascendental es transportada por agregación son saltadas, y el procesamiento continúa con las muestras restantes de la exploración vertical exactamente como se hace en el paso de significación.

Cuando el modo de agregación es introducido, las cuatro muestras de la exploración vertical se examinan. Si las cuatro muestras son insignificantes, un símbolo de agregación todo insignificante es codificado, y el proceso de la exploración vertical es completo. Si no, un símbolo de agregación algo significativo es codificado, y dos símbolos binarios entonces son utilizados para codificar el número de muestras insignificantes al frente en la exploración vertical.

Los símbolos generados durante el paso de limpieza siempre se codifican aritméticamente. En el modo de agregación, se codifica el símbolo de agregación usando un solo contexto, y dos símbolos run length se codifica usando un solo contexto con una distribución de probabilidad uniforme fija. Cuando el modo de agregación no se emplea, la codificación de trascendencia y de signo funciona como en el caso del paso de significación.

2.2.11 Paquetes y capas

La información del paso de codificación se empaqueta en unidades de datos llamadas paquetes, en un proceso designado paquetización. Los paquetes resultantes entonces son la salida del flujo de código (codestream) final.

El proceso del paquetización impone una organización particular sobre los datos del paso de codificación en la salida del codestream. Esta organización facilita muchas de las características deseadas del codec incluyendo el índice de escalabilidad y la recuperación progresiva por fidelidad o resolución. Un paquete no es nada más una colección de datos del paso de codificación. Como se muestra en la figura 2.12, cada paquete incluye dos partes: una cabecera y un cuerpo. La cabecera indica qué pasos de codificación se incluyen en el paquete, mientras que el cuerpo contiene los datos por sí mismos del paso de codificación. Aunque, en el diagrama, la cabecera es seguida inmediatamente por el cuerpo, éste es solamente conceptual. En la codestream, la cabecera y el cuerpo pueden aparecer juntos o por separado, dependiendo de las opciones de codificación en efecto.



Figura 2.12. Estructura del paquete

El índice de escalabilidad se alcanza con capas. Los datos codificados de cada tile se organiza en L capas, numeradas a partir de la 0 a $L-1$, donde $L \geq 1$. Cada paso de codificación es asignado a uno de las L capas o se desecha. Los pasos de codificación que contienen los datos más importantes se incluyen en las capas más bajas, mientras que los pasos de la codificación asociados a detalles más finos se incluyen en capas más altas. En el caso de compresión con pérdidas, algunos pasos de codificación pueden ser desechados (es decir, no incluidos en alguna capa) en los cuales el control de índice de caso debe decidir que pasos incluye en el codestream final. En el caso sin pérdidas, todos los pasos de codificación deben ser incluidos. Si se emplean las capas múltiples (es decir, $L > 1$), el índice de control debe decidir en qué capa debe incluir cada paso de codificación.

Hay que recordar que cada paso de codificación se asocia a una componente, a un nivel de la resolución, a un subbanda, y a un codeblock. En la codificación, un paquete se genera para cada componente, nivel de la resolución, capa, y recinto cuadrutuple (figura 2.13). Un paquete puede ser vacío. Los paquetes vacíos son a veces necesarios puesto que un paquete se debe generar para cada combinación de componente-nivel-capa-recinto incluso si el paquete resultante no transporta ninguna nueva información.

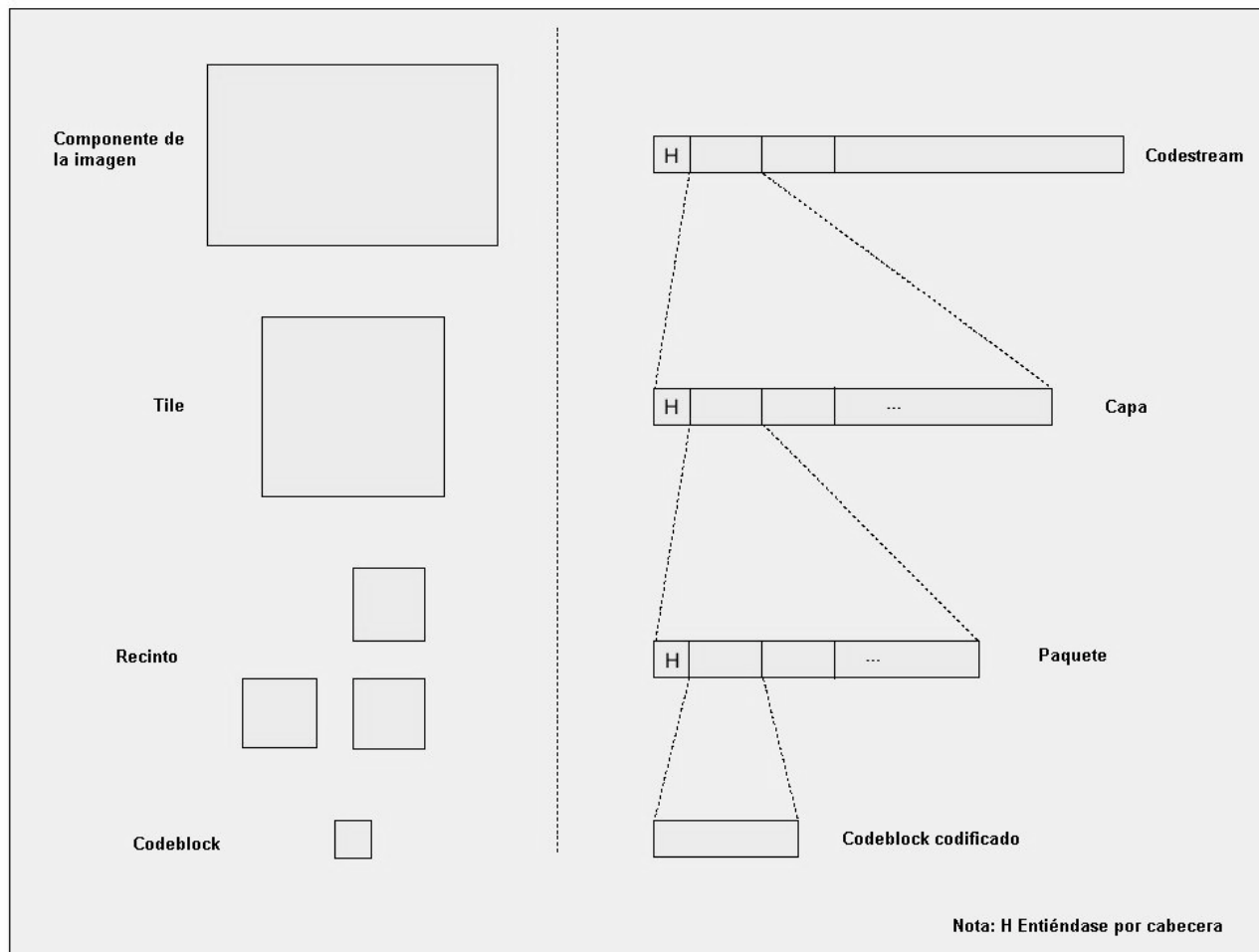


Figura 2.13. Correspondencia conceptual entre el espacio y las representaciones del bitstream.

Un recinto es esencialmente la agrupación de codeblocks dentro de una subbanda. Puesto que los pasos de codificación de los datos de diversos recintos se codifican en paquetes separados, usar recintos más pequeños reduce la cantidad de datos contenidos en cada paquete. Si un paquete contiene menos datos, un probable bit de error resulta en menos pérdida de información. Así, usando tamaños pequeños de recintos se mejora la recuperación de error, mientras la eficiencia de codificación es degradada debido al incremento de número de paquetes.

Más de un ordenamiento de paquetes en el codestream es soportado. Tales ordenamientos se llaman progresiones. Hay cinco progresiones incorporadas definidas: ordenación de capa-resolución-componente-posición, ordenación resolución-capacomponente-posición, ordenación resolución-posición-componente-capacomponente-posición, ordenación posición-componente-resolución-capacomponente-posición, y ordenación componente-posición-resolución-capacomponente-posición. El orden de clase para los paquetes es dado por el nombre de ordenación, donde la posición se refiere al número del recinto, y las claves de clasificación se enumeran de la más significativa al menos significativo. Por ejemplo, en el caso de primera ordenación dada arriba, los paquetes son ordenados primero por capas, en segundo lugar por resolución, tercero por componente, y por último por el recinto. Esto corresponde a una

recuperación progresiva por el panorama de fidelidad. El segundo ordenamiento arriba es asociado con la recuperación progresiva por resolución. Los tres ordenamientos restantes son algo más esotéricos. Es también posible especificar progresiones adicionales definidas por el usuario a expensas del incremento de codificación.

2.2.12 Codificación de la cabecera del paquete

La cabecera del paquete correspondiente a una componente, nivel de resolución, capa, y a un recinto en particular, se codifica como sigue. Primero, un solo símbolo binario se codifica para indicar si algunos datos del paso de codificación se incluyen en el paquete (es decir, si el paquete es no vacío). Si el paquete es vacío, no se requiere la transformación posterior y el algoritmo termina. Si no, se procede a examinar cada subbanda en el nivel de resolución en un orden fijo. Para cada subbanda, se inspeccionan los codeblock que pertenecen al recinto de interés en el orden de exploración de trama según lo mostrado en figura 2.14. Para procesar un solo codeblock, se comienza determinando si algunos nuevos datos del paso de codificación se van a incluir. Si todavía no se ha incluido ningún dato del paso de codificación para este codeblock, la información de inclusión se transporta por medio de un procedimiento de codificación basado en árbol cuadrienal (quadtree). Si no, un bit no procesado se emite para indicar la presencia o la ausencia de nuevos datos del paso de codificación para el codeblock. Si no hay nuevos pasos de codificación incluidos, se procede al proceso del siguiente codeblock en el recinto. Asumiendo que nuevos datos del paso de codificación están para ser incluidos, se continúa con el procesamiento del actual codeblock. Si esta es la primera vez que los datos del paso de codificación se han incluido para codeblock, se codifica el número de bit-planes insignificantes principales para el codeblock usando un algoritmo de codificación basado en árbol cuadrienal. Entonces, el número de nuevos pasos de codificación, y la longitud de los datos asociados a estos pasos se codifica.

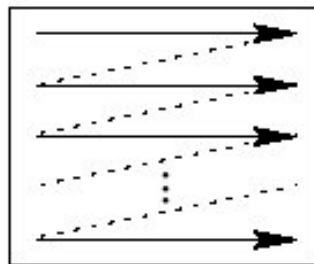


Figura 2.14. Orden de exploración del codeblock en un recinto.

2.2.13 Codificación del cuerpo del paquete.

El algoritmo usado para codificar el cuerpo del paquete es relativamente simple. Los codeblocks se examinan en el mismo orden que en el caso de la cabecera del paquete. Si algunos nuevos pasos fueron especificados en la cabecera correspondiente del paquete, los datos para estos pasos de codificación se concatenan al cuerpo del paquete.

2.3 Control de índice

En el codificador, el control de índice se alcanza a través de dos mecanismos distintos: la elección del tamaño del paso del cuantizador, y la selección del subconjunto de pasos de codificación incluidos en el codestream. El estándar no especifica cómo estos mecanismos deben ser empleados, y es posible utilizar un mecanismo exclusivamente o ambos juntos.

Cuando se emplea el primer mecanismo, los tamaños de paso del cuantizador se ajustan para el control de índice. Como se aumentan los tamaños de paso, el índice disminuye, con el costo de mayor distorsión. Aunque este mecanismo de control de índice es conceptualmente simple, tiene una desventaja potencial. Cada vez que se cambian los tamaños de paso del cuantizador, los índices del cuantizador cambian, y la codificación se debe realizar otra vez. Puesto que la codificación requiere una cantidad considerable de cómputo, esta aproximación para el índice de control puede no ser práctica.

Cuando se utiliza el segundo mecanismo, el codificador puede elegir, desechar pasos de codificación para el índice de control. El codificador sabe la contribución que cada paso de codificación hace para el índice, y también puede calcular la reducción de la distorsión asociada a cada paso de codificación. Usando esta información, el codificador puede entonces incluir los pasos de codificación en orden de la reducción de distorsión que disminuye por índice de unidad hasta que se haya agotado el presupuesto del bit. Esta aproximación es muy flexible en métricas diferentes de distorsión y pueden ser fácilmente acomodadas. Para un tratamiento más detallado del control de índice [3] y [5].

2.4 Codificación de una región de interés

El codec permite que diversas regiones de una imagen sean codificadas con diferente fidelidad. Esta característica se conoce como codificación de la región de interés (ROI). La funcionalidad del ROI es importante donde ciertas partes de la imagen son de mayor importancia que otras. En tal caso, estas regiones necesitan ser codificadas en calidad más alta que el fondo. Durante la transmisión de la imagen, estas regiones necesitan ser transmitidas primero o en una prioridad más alta, como por ejemplo en el caso de la transmisión progresiva.

El esquema de codificación del ROI en la parte I del estándar se basa en el método llamado MAXSHIFT de Christopoulos et al. [13]-[15]. El método de MAXSHIFT es una extensión del método de codificación basado en escalamiento del ROI general [16].

Cuando una imagen se sintetiza de sus coeficientes de transformada, cada coeficiente contribuye solamente a una región específica en la reconstrucción. Así, una forma para codificar una ROI con mayor fidelidad que el resto de la imagen sería identificar los coeficientes que contribuyen a la ROI, y después codificar algunos o todos estos coeficientes con mayor precisión que los otros. Esto es, de hecho, la premisa básica detrás de la técnica de la codificación de la ROI empleada en el codec JPEG-2000.

Cuando una imagen es codificada con una ROI, algunos de los coeficientes de la transformada son identificados más importantes que otros. Los coeficientes de mayor importancia se refieren como coeficientes de la ROI, mientras que los coeficientes restantes se conocen como coeficientes del fondo. Note que hay una correspondencia uno a uno entre los coeficientes de la transformada y los índices del cuantizador, por lo cual se definen índices del cuantizador para la ROI y los coeficientes del fondo, como índices del cuantizador de la ROI y del fondo, respectivamente.

La funcionalidad de la codificación de la ROI afecta el proceso de codificación. En el codificador, antes de que los índices del cuantizador para las diversas subbandas sean codificadas por el bit-plane, los índices del cuantizador de la ROI son escalados hacia arriba por una potencia de dos. Este escalamiento se realiza de tal manera que se asegura de que todos los bits de los índices del cuantizador de la ROI se hallen en planos de bits más significativos que los bits potencialmente distintos a cero de los índices del cuantizador del fondo. Por consiguiente, toda la información sobre los índices del cuantizador de la ROI serán señalizados antes de la información de los índices de la ROI de fondo. De esta manera, la ROI se puede reconstruir a una fidelidad más alta que el fondo.

Antes de que los índices del cuantizador sean codificados en el bit-plane, el codificador examina los índices del cuantizador del fondo para todas las subbandas, buscando el índice con la magnitud más grande. Suponga que este índice tiene su bit más significativo en la posición del bit $N-1$. Todos los índices de la ROI entonces son cambiados por N bits a la izquierda, y la codificación bit-plane procede como en el caso de una no ROI. El valor N del cambio de la ROI se incluye en el codestream.

Puesto que los bits-planes con la información perteneciente a la ROI son completamente separados de éstos que pertenecen al fondo, el número de bits-planes para la ROI y para el fondo puede ser elegido independientemente. Esto da la posibilidad de elegir diversos bits rates para la ROI y para el fondo. Para hacer esto, es suficiente con descartar los bits-planes menos significativos de la ROI y del fondo.

Durante la decodificación, cualquier índice del cuantizador con bits distintos a cero que se hallen en el bit plane N o arriba, se puede deducir que pertenece al sistema de la ROI. Después de que los índices reconstruidos del cuantizador se obtienen del proceso de decodificación del bit-plane, todos los índices en el conjunto de la ROI son reducidos por un cambio a la derecha de N bits. Esto deshace el efecto de escalamiento en el lado del codificador.

El sistema de la ROI se puede elegir para corresponder a los coeficientes de transformación que afectan una región particular en una imagen o un subconjunto de éstos que afectan la región. Esta técnica de codificación de la ROI tiene un número de características deseables. Primero, el ROI puede tener cualquier forma arbitraria y ser disjunta. En segundo lugar, no hay necesidad de señalar explícitamente el sistema de la ROI, puesto que puede ser deducida por el decodificador por el valor del cambio de la

ROI y de la magnitud de los índices del cuantizador. Un ejemplo de codificación ROI se observa en la figura 2.15.

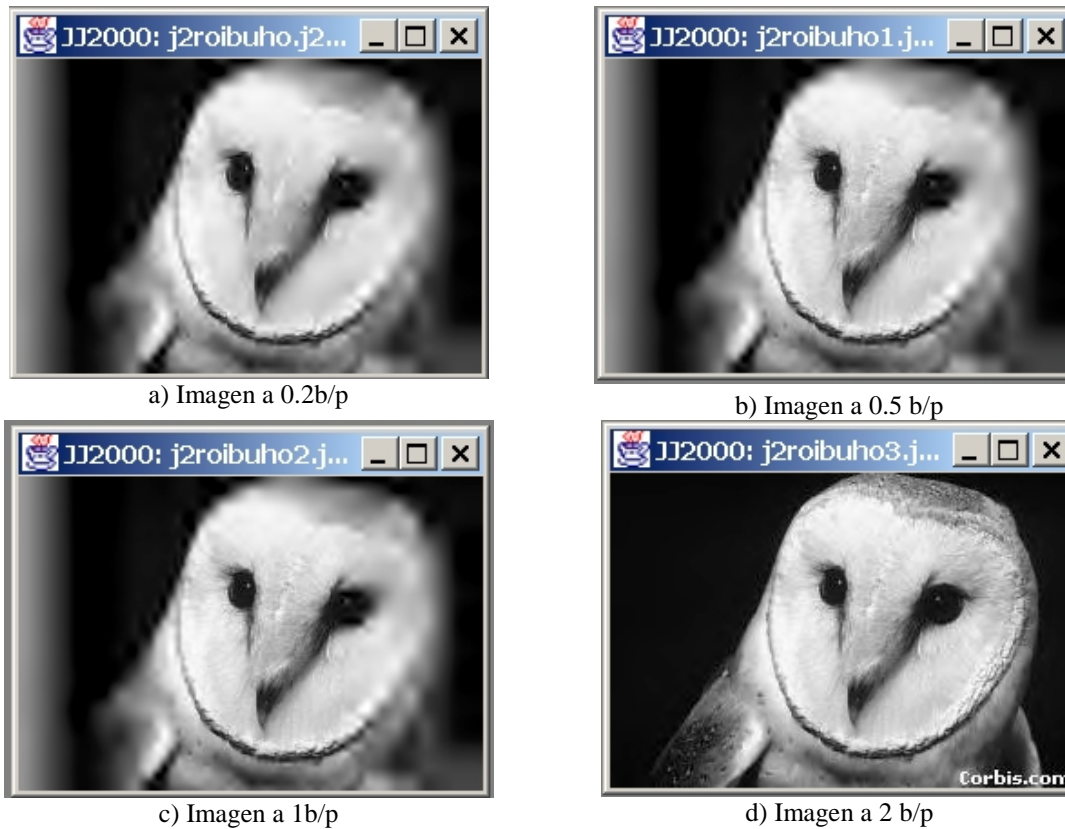


Figura 2.15. Codificación de una región de interés.

2.5 Code Stream

Para especificar la representación codificada de una imagen, dos diferentes niveles de sintaxis son empleados por el codec. El nivel más bajo de sintaxis se asocia con el flujo de código (codestream). El codestream es esencialmente una secuencia de registros etiquetados y sus datos de acompañamiento.

El bloque de construcción básica del codestream es el segmento del marcador. Según lo demostrado en figura 2.16, un segmento del marcador abarca tres campos: el tipo, la longitud, y los campos de los parámetros. El campo (o marcador) de tipo identifica la clase particular de segmento del marcador. El campo de longitud especifica el número de bytes en el segmento del marcador. El campo de parámetros proporciona información adicional específica del tipo de marcador. No todos los tipos de segmentos del marcador tienen campos de longitud y de parámetros. La presencia (o ausencia) de estos campos es determinada por el tipo de segmento del marcador. Cada tipo de segmento de marcador señala su propia clase particular de información.

Tipo	Longitud (si requiere)	Parámetros (si requiere)
16 bits	16 bits	Long. variable

Figura 2.16. Estructura del codestream.

El codestream consiste en una cabecera principal, seguido por la cabecera y el cuerpo del tile, seguidos por un acoplado principal. Una lista de algunos de los segmentos del marcador más importantes es dada en la tabla 2.2. Los parámetros especificados en segmentos del marcador en la cabecera sirven como defectos para el codestream entero. Estos ajustes de defecto, sin embargo, se pueden eliminar para un tile particular especificando nuevos valores en un segmento del marcador en la cabecera del tile.

Tipo	Descripción
Inicio del codestream (SOC)	Señaliza el inicio del codestream. Primer segmento de marcador en el codestream.
Fin del codestream (EOC)	Señaliza el fin del codestream. Último segmento de marcador en el codestream.
Inicio del tile (SOT)	Indica el inicio de la cabecera de un tile. Primer segmento de marcador en la cabecera del tile.
Inicio de datos (SOD)	Señala el final de la cabecera del tile. Último T en la cabecera del tile. El cuerpo del tile sigue inmediatamente después de este segmento de marcador.
Tamaño de imagen y tile (SIZ)	Transmite las características básicas de la imagen y los parámetros de tiling. Segundo segmento del marcador en el codestream.
Estilo de codificación por defecto (COD)	Especifica los parámetros de codificación
Componente de estilo de codificación (COC)	Especifica un conjunto de parámetros de codificación para una sola componente.
Cuantización por defecto (QCD)	Especifica los parámetros de cuantización.
Componente de cuantización (QCC)	Especifica los parámetros de cuantización para una sola componente.
Región de interés	Especifica los parámetros de la codificación de la región de interés.

Tabla 2.2. Tipos de segmentos de marcador.

Todos los segmentos del marcador, cabeceras del paquete, y cuerpos del paquete son múltiplo de 8 bits de longitud. Por consiguiente, se alinean todos los marcadores en bytes, y el codestream por sí mismo es siempre un número entero de bytes.

2.6 Formato de archivo

Un codestream proporciona solamente la información más básica requerida para decodificar una imagen. Mientras que en algunas aplicaciones simples esta información es suficiente, en otras aplicaciones datos adicionales se requiere. Para permitir que datos

adicionales sean especificados, un nivel adicional de sintaxis es empleado por el codec. Este nivel de sintaxis se refiere como formato de archivo. El formato de archivo se utiliza para transportar datos codificados e información auxiliar de la imagen. Aunque este formato de archivo es opcional, éste indudablemente será utilizado extensivamente por muchas aplicaciones, particularmente en aplicaciones de software basadas en computadoras.

El bloque de construcción básico del formato de archivo es referido como una caja. Según lo mostrado en figura 2.17, la caja abarca cuatro campos. El campo LBox especifica la longitud de la caja en bytes. El campo TBox indica el tipo de caja (es decir, la naturaleza de la información contenida en la caja). El campo XLBox es un indicador de longitud extendida que proporciona un mecanismo para especificar la longitud de una caja cuyo tamaño sea demasiado grande para ser codificado solamente en el campo de longitud. El campo DBox contiene datos específicos de un tipo de caja particular.

LBox	TBox	XLBox (si requiere)	DBox
32 bits	32 bits	64 bits	variable

Figura 2.17. Estructura de la caja.

Un archivo es una secuencia de cajas. Puesto que ciertos tipos de cajas se definen para contener otras, hay una estructura jerárquica natural en un archivo. La estructura general de archivo se demuestra en figura 2.18. La caja de firma siempre es la primera, proporcionando una indicación de que el bitstream está ajustado a formato correctamente. La caja de perfil está siempre en segundo lugar, e indicando la versión del formato de archivo con el cual el bitstream se conforma. La caja de la cabecera contiene simplemente un número de otras cajas. La caja de cabecera de la imagen especifica varias características básicas de la imagen (incluyendo el tamaño de la imagen, el número de componentes, etc.). La caja de especificación de color identifica el espacio de color de los datos de la imagen (para los propósitos de exhibición) e indica cual mapa de componentes para el tipo de información espectral (es decir, la correspondencia entre los componentes y los planos de color/opacidad). Cada archivo debe contener por lo menos una caja contigua del codestream. Cada caja contigua de codestream contiene un codestream como datos. En esta manera, los datos codificados de la imagen se encajan en un archivo. Además de los tipos de cajas discutidos, hay también tipos de caja para especificar la resolución de captura y exhibición de una imagen, la información de la gama de colores, la información de propiedad intelectual, y datos específicos de aplicación/vendedor.

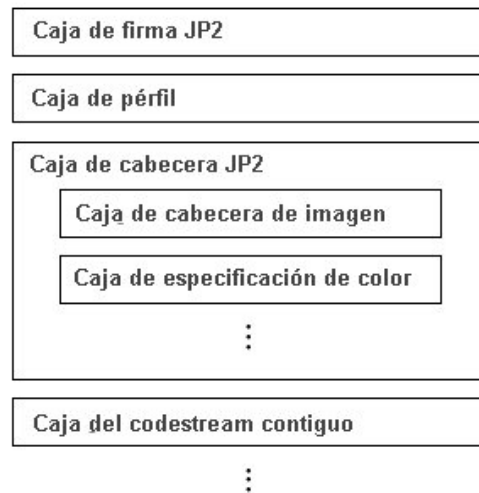


Figura 2.18. Estructura general del formato de archivo.

Aunque alguna información almacenada en el nivel de formato de archivo es redundante, esta redundancia permite la manipulación trivial de archivos sin ningún conocimiento de la sintaxis del codestream. La extensión de nombre de archivo `jp2` es utilizada para identificar los archivos que contienen datos en el formato de archivo JP2 (JPEG2000).

2.7 Escalabilidad

La codificación escalable de imágenes fijas significa la capacidad de alcanzar codificaciones de más de una calidad y/o resolución simultáneamente. La codificación escalable de imágenes implica la generación de una representación codificada (bitstream) en una manera que facilite la derivación de imágenes de más de una calidad y/o de resolución por decodificación escalable. La escalabilidad del bitstream es una propiedad del bitstream que permite decodificar subconjuntos apropiados del bitstream para generar imágenes completas de calidad y/o resolución conmensuradas con la proporción del bitstream decodificado. Los decodificadores de diversas complejidades (de funcionamiento bajo a alto) pueden coexistir para un bitstream escalable. Mientras que los decodificadores de bajo funcionamiento pueden decodificar solamente porciones pequeñas del bitstream generando una calidad básica, los decodificadores del alto rendimiento pueden decodificar mucho más y producir perceptiblemente calidad más alta.

Los tipos más importantes de escalabilidad son la escalabilidad SNR y la escalabilidad espacial o de resolución. El sistema de compresión JPEG2000 soporta escalabilidad. Una ventaja clave de la compresión escalable es que el bit rate deseado o la resolución de reconstrucción no necesita ser conocido en la compresión. Una ventaja adicional de la escalabilidad es su capacidad de proporcionar resistencia a los errores de la transmisión, pues los datos más importantes de la capa más baja se pueden enviar sobre el canal con un mejor funcionamiento de error, mientras que los datos menos críticos de la capa de extensión se pueden enviar sobre el canal con un pobre funcionamiento de

error. Ambos tipos de escalabilidad son muy importantes para aplicaciones de acceso a Internet y bases de datos y el escalamiento del ancho de banda para una distribución robusta.

Escalabilidad SNR

La escalabilidad SNR es propuesta para sistemas con la característica común primaria de que un mínimo de dos capas de calidad de imagen son necesarias. La escalabilidad SNR implica generar por lo menos dos capas de la imagen de la misma resolución espacial, pero a diversas calidades, de una sola fuente de imagen. La capa más baja es codificada por sí misma para proporcionar una calidad básica de imagen y las capas de extensión se codifican para realzar la capa más baja. Una capa de extensión, cuando es agregada de nuevo a la capa más baja, genera una reproducción de más alta calidad de la imagen de entrada. La figura 2.19 ilustra un ejemplo de escalabilidad SNR. La primera imagen es comprimida sin pérdidas y descomprimida a 0,125 b/p, 0,25 b/p, y 0,5 b/p.

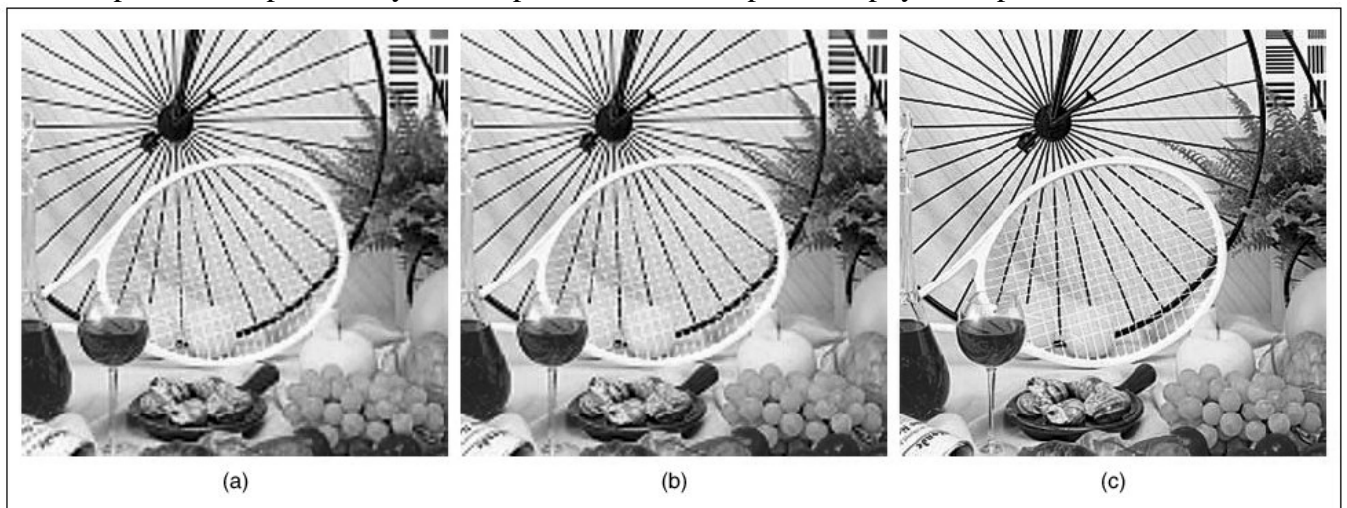


Figura 2.19. Ejemplo de escalabilidad SNR. Imagen comprimida a) 0.125b/p, b) 0.25b/p, c) 0.5b/p

Escalabilidad espacial

La escalabilidad espacial es propuesta para sistemas con la característica común primaria de que un mínimo de dos capas de resolución espacial son necesarias. La escalabilidad espacial implica generar por lo menos dos capas espaciales de resolución de una sola fuente tales que la capa más baja es codificada por sí misma para proporcionar una resolución espacial básica y la capa del extensión emplea la capa más baja interpolada espacialmente y lleva la resolución espacial completa de la fuente de la imagen de entrada. La figura 2.20 muestra un ejemplo de tres niveles de decodificación progresiva por resolución de una imagen. La escalabilidad espacial es útil para el acceso rápido a bases de datos así como la distribución a diversas resoluciones a terminales con diversas capacidades en términos de exhibición y ancho de banda. JPEG2000 soporta también una combinación de escalabilidad espacial y SNR.

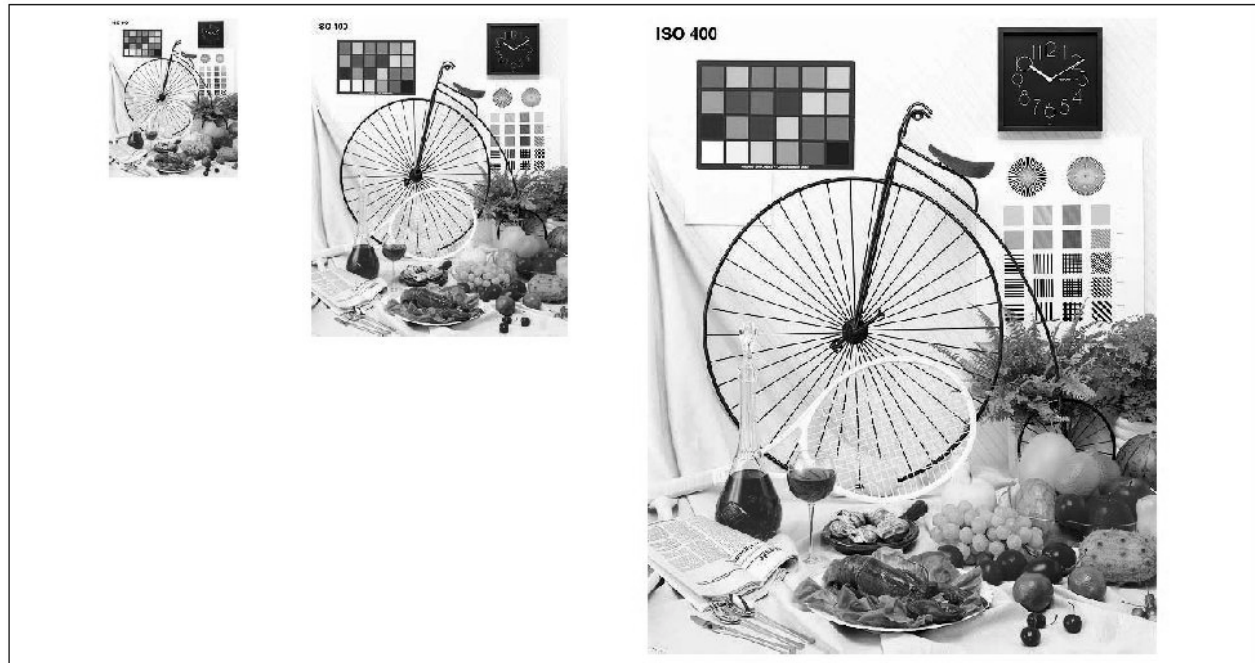


Figura 2.20. Ejemplo de decodificación progresiva por resolución.

El bitstream contiene los marcadores que identifican el tipo de progresión del bitstream. Los datos almacenados en paquetes son idénticos sin importar el tipo de escalabilidad usado. Por lo tanto es trivial cambiar el tipo de la progresión o extraer cualquier dato requerido del bitstream. Para cambiar la progresión de SNR a progresión por resolución, un programa de análisis puede leer los marcadores, cambiar el tipo de progresión en los marcadores, y después escribir los marcadores nuevos en el nuevo orden. De este modo, la transcodificación rápida del bitstream se puede alcanzar por un servidor o gateway, sin requerir el uso de la decodificación y re-codificación de la imagen.

2.8 Recuperación de error (error resilience)

La recuperación de error es una de las características más deseadas en aplicaciones sobre móviles e Internet. JPEG2000 utiliza un codificador de longitud variable (codificador aritmético) para comprimir los coeficientes wavelet cuantizados. Como es sabido, la codificación de longitud variable es propensa a los errores de canal y transmisión. Un bit erróneo resulta en pérdida de sincronización en el decodificador de entropía y la imagen reconstruida puede ser seriamente dañada. Para mejorar el funcionamiento de transmisión de imágenes comprimidas sobre canales propensos a error, se incluyen en el estándar la recuperación de error (error resilience) y utilerías. Las utilerías de recuperación de error tratan con los errores de canal usando las siguientes aproximaciones: partición y resincronización de datos, detección y ocultación de error, y transmisión con calidad de servicio (QoS por sus siglas en inglés) basada en prioridad. La recuperación de error se alcanza en el nivel de codificación de entropía y en el nivel de paquete. La tabla 2.3 resume varias formas de lograr esto.

Tipo de utilería	Nombre
Nivel de codificación de entropía	<ul style="list-style-type: none"> - codeblocks - terminación del codificador aritmético para cada paso - reinicio de contextos después de cada paso de codificación. - selección del puente de codificación aritmética - símbolos de segmentación
Nivel de paquete	<ul style="list-style-type: none"> - formato corto de paquetes - paquetes con marcador de sincronización

Tabla 2.3. Utilerías para la recuperación de error.

La codificación de entropía de los coeficientes cuantizados se realiza dentro de los codeblocks. Ya que la codificación y decodificación de los codeblocks son procesos independientes, los bits erróneos del bitstream de un codeblock serán restringidos dentro de ese codeblock. Para aumentar recuperación de error, la terminación del codificador aritmético se permite después de cada paso de codificación y los contextos se pueden reajustar después de cada paso de codificación. Esto permite que el decodificador aritmético continúe el proceso de decodificación incluso si ha ocurrido un error.

El modo de codificación lazy es también útil para la recuperación de error. Este se relaciona con puente de codificación aritmética opcional en el cual los bits son alimentados como bits sin procesar en el bitstream sin la codificación aritmética. Esto previene los tipos de propagación de error para los cuales la codificación de longitud variable es susceptible.

En el nivel de paquete, un paquete con un marcador de resincronización permite la partición y resincronización espacial. Esto se coloca delante de cada paquete en un tile con un número de serie que comienza en cero y se incrementa con cada paquete.

2.9 Características visuales

El sistema visual humano desempeña un papel importante en la calidad percibida de la representación de imágenes comprimidas. Por lo que se utilizan modelos de sensibilidad de variación del sistema visual para frecuencias espaciales, como medida en la función de sensibilidad de contraste (CSF por sus siglas en inglés). Puesto que el peso del CSF es determinado por la frecuencia visual de los coeficientes de la transformada, habrá un peso del CSF por subbanda en la transformada wavelet. El diseño de los pesos CSF es una edición del codificador y depende de la condición de visión específica bajo la cual la imagen decodificada debe ser vista.

El estándar del JPEG2000 soporta dos tipos de peso (weighting) de frecuencia visual. El peso visual fijo (FVW por sus siglas en inglés) y la codificación progresiva visual o peso progresivo visual (VPW por sus siglas en inglés). En el FVW, solamente un conjunto de pesos CSF se eligen y se aplican de acuerdo con las condiciones de visión. En el VPW, diversos conjuntos de pesos CSF se utilizan en varias etapas de la codificación embebido (embedded). Esto es porque durante un estado de transmisión

progresiva, la imagen es vista a varias distancias. Por ejemplo, en bitrates bajos, la imagen es vista a distancias relativamente grandes, y mientras más y más bits son recibidos se mejora la calidad y la distancia de visión. FVW se puede considerar como caso especial de VPW.

Referencias

- [1] JPEG 2000 Part I: Final Draft International Standard (ISO/IECFDIS15444-1), ISO/IEC JTC1/SC29/WG1 N1855, Aug. 2000.
- [2] JPEG 2000 Requirements and Profiles, ISO/IEC JTC1/SC29/WG1 N1271, Mar. 1999.
- [3] ISO/IEC, ISO/IEC 15444-1, Information technology - JPEG 2000 image coding system, 2000.
- [4] Press Release of the 23rd WG1 Singapore Meeting, ISO/IEC/JTC1/SC29/WG1 N2058, 9 Mar. 2001.
- [5] D. Taubman, "High performance scalable image compression with EBCOT," Proc. of IEEE International Conference on Image Processing, Kobe, Japan, 1999, vol. 3, pp. 344-348.
- [6] D. Taubman, "High performance scalable image compression with EBCOT," IEEE Trans. Image Processing, Vol. 9, No. 7, pp. 1158-1170, July 2000
- [7] B.E. Usevitch, "A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000," IEEE Signal Processing Mag., vol. 18, pp. 22-35, Sept. 2001.
- [8] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," Applied and Computational Harmonic Analysis, vol. 5, no. 3, pp. 332-369, July 1998.
- [9] D. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, New York, NY, USA, Apr. 1988, vol. 2, pp. 761-764.
- [10] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. on Image Processing, vol. 1, no. 2, pp. 205-220, Apr. 1992.
- [11] "Progressive Lossy To Lossless Core Experiment with a Region of Interest: Results with the S, S+P, Two-Ten Integer Wavelets and with the Difference Coding Method", ISO/IEC JTC1/SC29/WG1 N741, Mar. 1998.
- [12] ISO/IEC, ISO/IEC 14492-1, Lossy/lossless coding of bi-level images, 2000.
- [13] C.A. Christopoulos, J. Askelof, and M. Larsson, "Efficient methods for encoding regions of interest in the upcoming JPEG 2000 still image coding standard," IEEE Signal Processing Lett., vol. 7, pp. 247-249, Sept. 2000.
- [14] C.A. Christopoulos, J. Askelof, and M. Larsson, "Efficient encoding and reconstruction of regions of interest in JPEG 2000," in Proc. X European Signal Processing Conf. (EUSIPCO-2000), Tampere, Finland, Sept. 2000, pp. 1133-1136.
- [15] C.A. Christopoulos, J. Askelof, and M. Larsson, "Efficient region of interest encoding techniques in the upcoming JPEG 2000 still image coding standard," in Proc. IEEE Int. Conf. Image Processing (ICIP 2000), vol. II. Vancouver, Canada, Sept. 2000, pp. 41-44.

- [16] E. Atsumi and N. Farvardin, “Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees,” Proc. IEEE Int. Conf. Image Processing, Chicago, IL, Oct. 1998, pp. 87-91.
- [17] A. Skodras, C. Christopoulos, T. Ebrahimi, “The JPEG 2000 still image compression standard”, IEEE Signal Processing Magazine, September 2001.
- [18] M.D. Adams, F. Kossentini, T. Ebrahimi, “JPEG 2000: The Next Generation Still Image Compression Standard”
- [19] K. R. Rao, Y. Huh, “JPEG 2000”, IEEE Region 8 International Symposium on Video/ Image Processing and Multimedia Communications, Juny 2002, Zadar, Croatia.
- [20] K. Osama, Al-Shaykh, I. Moccagatta and H. Chen, “JPEG 2000: A new still image compression standard”, IEEE, 1998.

Codec

El codec usado en las imágenes muestra es JJ2000 desarrollado por Cannon Research, EPFL y Ericsson disponibles en <http://jj2000.epfl.ch>

Capítulo 3

Codificación de canal.

La codificación de canal protege los datos digitales de errores por redundancias introducidas selectivamente en los datos transmitidos. Los códigos de canal que son usados para detectar errores son llamados códigos de detección de error, mientras que los códigos que pueden detectar y corregir errores son llamados códigos de corrección de errores.

La historia de la codificación de canal o la codificación de corrección de errores hacia delante (FEC por sus siglas en inglés) data antes de los trabajos pioneros de Shannon en los cuales el predijo que una comunicación viable arbitraria es factible por una codificación redundante FEC, aunque el se abstuvo de proponer esquemas específicos para implementaciones practicas. Históricamente uno de los primeros códigos prácticos fue el código de Hamming de corrección de errores, el cual es un código de bloque propuesto en 1950. Los códigos FEC convolucionales datan de antes de 1955.

3.1 Teorema de codificación de canal de Shannon

El segundo teorema de Shannon [1], también conocido como teorema de codificación de canal de ruido, demostró que por una codificación adecuada de la información, los errores inducidos por un canal ruidoso pueden ser reducidos hasta algún nivel deseado sin sacrificar el índice de información transferida. Éste esencialmente demuestra que para cada canal con capacidad de canal C , existe un código FEC tal que la información pueda ser transmitida a través del canal a velocidades menores que C con un índice arbitrario de error menor.

La fórmula de capacidad de canal de Shannon es dada por:

$$C = B \log_2 \left(1 + \frac{P}{N_0 B} \right) = B \log_2 \left(1 + \frac{S}{N} \right), \quad 3.1$$

donde C es la capacidad de canal (bits por segundo), B es el ancho de banda de transmisión (Hz), P es la potencia de la señal recibida y N_0 es la densidad de potencia de ruido single-sided.

Ésta ecuación puede ser normalizada por el ancho de banda de transmisión y es dada por:

$$\frac{C}{B} = \log_2 \left(1 + \frac{E_B R_B}{N_0 B} \right) \quad 3.2$$

donde C/B denota la eficiencia de ancho de banda.

El propósito básico de la detección de errores y las técnicas de corrección de errores es el de introducir redundancias en los datos para mejorar el rendimiento de un enlace inalámbrico.

Aunque hay muchos tipos de códigos FEC disponibles para su uso en los sistemas de comunicación, ellos han sido divididos dentro de dos principales categorías: códigos de bloques y códigos convolucionales.

3.2 Códigos de bloque

Los códigos de bloque son códigos FEC que permiten un número limitado de errores para ser detectados y corregidos sin retransmisión. Un código de bloque puede ser un código de bloque lineal o un código de bloque no lineal. Un bloque lineal puede ser un código cíclico u otro.

En los códigos de bloques, los bits de paridad son añadidos en los bloques de bits de mensaje para hacer los codewords o bloques de código. Estos bloques de código secuenciales son formados al tomar palabras de k bits de longitud y generan un codeword de n bits de longitud, con $n > k$. Un total de $n - k$ bits redundantes son añadidos a los k bits de información con el propósito de detectar y corregir errores [2]. El código de bloque es referido como un código (n, k) , y el índice del código es referido como $R_c = k/n$ (bps) y es igual al índice de información dividida por el índice de canal crudo.

La habilidad de un código de bloque para corregir errores es una función de la distancia de código. A pesar del índice del código, otros parámetros importantes son la distancia y el peso de un código definidos a continuación.

Distancia del código. La distancia de un codeword es el número de elementos en el cual dos codewords C_i y C_j difieren

$$d(C_i, C_j) = \sum_{l=1}^N C_{i,l} \oplus C_{j,l} \text{ (modulo } - q), \quad 3.3$$

donde d es la distancia del codeword y q es el número de posibles valores de C_i y C_j . Si el código usado es binario la distancia es conocida como la distancia de Hamming. La distancia mínima d_{min} es la distancia más pequeña para un conjunto dado y es dada por:

$$d_{min} = \text{Min}\{d(C_i, C_j)\}. \quad 3.4$$

Peso de código. El peso de un codeword es dado por el número de elementos no cero en el codeword. Para un código binario el peso es básicamente el número de 1's lógicos en el codeword y es dado por:

$$w(C_i) = \sum_{l=1}^N C_{i,l} \quad 3.5$$

Propiedades de los códigos de bloque.

Linealidad. Suponga C_i y C_j son dos palabras código en un código (n,k) , y permita que α_1 y α_2 sean dos elementos seleccionados del alfabeto. Entonces el código es lineal si solo si $\alpha_1 C_i + \alpha_2 C_j$ es también una palabra código. Un código lineal debe contener la palabra código de todo cero.

Sistemático. Un código sistemático es aquel en el cual los bits de paridad son añadidos al final de los bits de información. Para un código (n,k) , los primeros k bits son idénticos a los bits de información, y los $n-k$ bits restantes de cada palabra código son combinaciones lineales de los k bits de información.

Cíclico. Los códigos cíclicos son un subconjunto de la clase de códigos lineales los cuales satisfacen las siguientes propiedades de cambio cíclico. Si $C = [c_{n-1}, c_{n-2}, \dots, c_0]$ es una palabra código de un código cíclico entonces $[c_{n-2}, c_{n-3}, \dots, c_0, c_{n-1}]$, obtenido por un cambio cíclico de los elementos de C , es también una palabra código. Como consecuencia de la propiedad cíclica los códigos poseen una considerable cantidad de estructuras, las cuales pueden ser explotadas en las operaciones de codificación y decodificación.

Las técnicas de codificación y decodificación hacen uso de construcciones matemáticas conocidas como campos finitos. Un campo finito es un sistema algebraico el cual, contiene un conjunto finito de elementos. La adición, sustracción, multiplicación y división de los elementos de un campo finito es lograda sin salir del conjunto. La adición y multiplicación satisfacen las leyes asociativa, conmutativa y distributiva. El elemento inverso aditivo de un elemento del campo a es $-a$ y el inverso multiplicativo de ese elemento es a^{-1} .

Para un número primo p , existe un campo finito el cual contiene p elementos. Este campo primo es denotado como $GF(p)$ porque los campos finitos son también llamados campos de Galois, en honor de su descubridor [2]. Es posible expandir el campo primo $GF(p)$ a un campo de elementos p^m el cual es llamado un campo de extensión de $GF(p)$ y es denotado como $GF(p^m)$, donde m es un entero positivo. Los códigos con símbolos de campo binario $GF(2)$ o sus extensiones de campo $GF(2^m)$ son comúnmente más usados en la transmisión de datos digitales y sistemas de almacenamiento, puesto que la información en estos sistemas es siempre codificada en forma binaria.

En la aritmética binaria, la adición y multiplicación del modulo-2 son usadas. Por lo que la multiplicación para el campo de extensión debe estar definida por lo que los demás elementos del campo pueden ser representados como una secuencia de potencias de α . La operación de multiplicación puede ser usada para producir un conjunto finito de elementos F como se muestra:

$$F = \{0, 1, \alpha, \alpha^2, \dots, \alpha^j, \dots\} = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^j, \dots\}$$

Para obtener un campo finito de elementos de $GF(2^m)$ de F , una condición debe ser impuesta sobre F por lo que ésta podrá contener solo 2^m elementos y es un conjunto cerrado bajo la multiplicación. Esta condición la cual cierra el conjunto de elementos del campo bajo la multiplicación es conocida como polinomio irreducible, el cual toma la siguiente forma:

$$\alpha^{(2^m-1)} + 1 = 0 \text{ o equivalentemente } \alpha^{(2^m-1)} = 1 = \alpha^0$$

Usando el polinomio irreducible, ningún elemento con una potencia más grande que 2^m-2 puede ser reducido para un elemento con una potencia menor que 2^m-2 .

La secuencia de elementos F así llega a ser la siguiente secuencia F^* , cuyos términos no cero son cerrados bajo la multiplicación:

$$F^* = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{2^m-2}, \alpha^{2^m-1}, \alpha^{2^m} \dots\} = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^m-2}, \dots, \alpha^0, \alpha^1, \alpha^2, \dots\}$$

Tomando el primer término de 2^m de F^* y teniendo los elementos del campo finito $GF(2^m)$ en su representación de potencia:

$$GF(2^m) = \{0, 1, \alpha, \alpha^2, \dots, \alpha^{2^m-2}\} = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^m-2}\} \quad 3.6$$

Esto puede ser expuesto para cada elemento 2^m de un campo finito y puede ser representado como un polinomio distinto de grado $m-1$ o menor. Cada elemento no cero de $GF(2^m)$ puede ser denotado como un polinomio $a_i(x)$, donde por lo menos uno de los coeficientes uno de los coeficientes es no cero.

$$\alpha^i = a_i(x) = a_{i0} + a_{i1}x + a_{i2}x^2 + \dots + a_{im}x^{m-1} \quad 3.7$$

La adición de dos elementos de un campo finito es entonces definida como la adición del modulo-2 de cada coeficiente del polinomio como potencias.

$$\alpha^i + \alpha^j = (a_{i0} + a_{j0}) + (a_{i1} + a_{j1})x + \dots + (a_{i,m-1} + a_{j,m-1})x^{m-1} \quad 3.8$$

Así $GF(2^m)$ puede ser construida, y usando las ecuaciones 3.7 y 3.8 la representación del polinomio para los elementos 2^m del campo pueden ser obtenidos.

3.2.1 Códigos lineales

Los códigos lineales de bloque son algunos de los códigos de corrección de errores disponibles actualmente de más fácil implementación y por lo tanto ampliamente usados. El término lineal significa que algún codeword sumado a otro codeword por el modulo-2 resultaría en un tercer codeword. El método generalmente usado para codificar los bits de información es por un proceso de mapeo el cual hace un mapeo de un 1 hacia un 1 de la información de bloque sobre el respectivo codeword.

3.2.1.1 Código de Hamming

Los códigos Hamming [3] son uno de los primeros códigos FEC disponibles. Hay ambos códigos Hamming binarios y no binario. Un código Hamming binario tiene como

propiedad que $n=2^m-1$ y $k=n-m$ donde $m \geq 3$. El índice de este código es

$$R = \frac{k}{n} = \frac{2^m - m - 1}{2^m - 1}, \tag{3.9}$$

el cual se aproxima a 1 cuando $m \rightarrow \infty$.

Los códigos Hamming son definidos por la matriz de parity-check que contiene $n-k$ renglones y n columnas. La tabla 3.1 provee un resumen de los códigos Hamming.

Tabla 3.1 Propiedades de los códigos Hamming

	Código Hamming de corrección de solo un error	Código Hamming de corrección de doble error
Longitud del código (n)	$n=2^m-1$	$n=2^m-1$
Longitud del mensaje (k)	$k=2^m-m-1$	$k=2^m-m-2$
Mensaje	m	$m+1$
Polinomio generador	$g(x)$	$p(x)=p(1+x)g(x)$
Palabra código	$v(x)=m(x)g(x)$	$v(x)=m(x)p(x)$
* m es un entero		

3.2.2 Códigos cíclicos

Los códigos cíclicos son un subconjunto de las clases de códigos lineales los cuales satisfacen la propiedad cíclica. Estos códigos tienen dos ventajas:

- La codificación y el síndrome de cómputo puede ser fácilmente implementados por la retroalimentación lineal rápida de los registros de corrimiento.
- Tienen considerables estructuras algebraicas inherentes por lo tanto los métodos prácticos de decodificación pueden ser encontrados.

Un código cíclico puede ser formado usando un polinomio generador $g(x)$ de grado $(n-k)$. El polinomio generador de un código cíclico $(n-k)$ es un factor de x^n+1 y tiene la forma general:

$$g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \dots + g_1x + 1. \tag{3.10}$$

Un mensaje polinómico $m(x)$ puede ser también definido como:

$$m(x) = m_{k-1}x^{k-1} + \dots + m_1x + m_0, \tag{3.11}$$

donde (x_{k-1}, \dots, x_0) representa los bits de información k .

3.2.2.1 Codificación.

Para combinar el mensaje polinómico y el polinomio generador se llega a un código sistemático, el procedimiento es el siguiente:

1. Premultiplicar el mensaje $m(x)$ por X^{n-k} .
2. Obtener el residuo $b(x)$, el cual contiene los dígitos de chequeo de paridad, por la división de $X^{n-k} m(x)$ por el polinomio generador $g(x)$.

3. Combinar $b(x)$ y $X^{n-k} m(x)$ para obtener el código polinómico $b(x) + X^{n-k} m(x)$.

3.2.2.2 Generador y matriz de chequeo de paridad.

Para obtener la matriz generadora en forma sistemática, los siguientes pasos deben de ser realizados:

1. Dividir X^{n-k+i} por el polinomio generador $g(x)$ para $i=0,1,\dots,k-1$ para obtener: $X^{n-k+i} = a_i(x)g(x) + b_i(x)$, donde el cociente es $a_i(x)$, y el resto es $b_i = b_{i,0} + b_{i,1}X + \dots + b_{i,n-k-1}X^{n-k-1}$.
2. Reacomodar dando: $b_i(x) + X^{n-k-i} = a_i(x)g(x)$, donde es obvio que $b_i(x) + X^{n-k-i}$, para $i=0,1,\dots,k-1$, son múltiplos de $g(x)$. Note que el término X^{n-k-i} representa una matriz de $1 \times k$ que es cero en todas partes excepto en $n-k-i$.
3. Por reacomodo de los polinomios k código dados por $b_i(x) + X^{n-k-i}$, la matriz generadora sistemática de $k \times n$ es obtenida y dada por:

$$G = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & \cdots & b_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ b_{1,0} & b_{1,1} & b_{1,2} & \cdots & b_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ b_{2,0} & b_{2,1} & b_{2,2} & \cdots & b_{2,n-k-1} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & 0 \\ b_{k-1,0} & b_{k-1,1} & b_{k-1,2} & \cdots & b_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \quad 3.12$$

La matriz generadora dada puede ser también expresada como $G = [b|I_k]$, donde I_k es una matriz identidad de $k \times k$.

Una vez que la matriz generadora ha sido encontrada, la matriz de chequeo de paridad de un código cíclico es obtenida de $G = [b|I_k]$ como $H = [I_{n-k}|b^T]$.

3.2.2.3 Corrección de errores y el síndrome.

La palabra recibida es representada por un polinomio de grado $n-1$ o menor, como se muestra por:

$$r(x) = r_0 + r_1X + \cdots + r_{n-1}X^{n-1} = q(x)g(x) + s(x), \quad 3.13$$

donde $q(x)$ es el cociente y $s(x)$ es el síndrome después de que $r(x)$ es dividido por $g(x)$. El polinomio $s(x)$ tiene un grado de $n-k$ o menor, y sus coeficientes forman el síndrome de $(n-k) \times 1$. Examinando las propiedades del síndrome se ve como es utilizado para la corrección de errores.

1. El síndrome del vector recibido es también el síndrome del vector de error. Para observar por qué, ponemos el vector recibido como la suma de un vector de código y el vector de error, esto es:

$$\begin{aligned} r(x) &= c(X) + e(x) \\ e(x) &= r(x) - c(x) \end{aligned}$$

donde finalmente queda que:

$$e(x) = [q(x) + m(x)]g(x) + s(x). \quad 3.14$$

2. Si $s(x)$ es el síndrome de $r(x)$, entonces $s^i(x)$ es el síndrome de $r^i(x)$.
3. El síndrome polinómico $s(x)$ puede ser sumado al vector recibido $r(x)$ para realizar la corrección de errores porque $s(x)$ es idéntico a $e(x)$, y solo si, los errores están dentro de los bits $(n-k)$ de paridad de chequeo de $r(x)$.

3.3 Códigos Bose-Chaudhuri-Hocquenghem

Los códigos Bose-Chaudhuri-Hocquenghem (BCH) [4-6] forman una larga clase de poderosos códigos cíclicos de corrección de errores aleatorios. Estos surgen de la generalización de los códigos Hamming y son usados para corrección de múltiples errores. La longitud de bloque de los códigos es $n=2^m-1$ para $m \geq 3$, y el número de errores que pueden corregir está limitado por $t < (2^m - 1)/2$. Los códigos BCH pueden ser generalizados para crear clases de códigos no binarios los cuales usan m bits por cada código de símbolo. La clase no binaria de códigos BCH más importante y común es la familia de los códigos conocidos como Reed-Solomon. Algunas características de los códigos BCH se muestran en la tabla 3.2.

Tabla 3.2 Propiedades de los códigos BCH.

Longitud de código	$n=2^m-1$
Número de bits de paridad	$n-k \leq mt$
Distancia mínima	$d_{\min} \geq 2t+1$
Capacidad de corrección de errores	$t = \left[d_{\min} - \frac{1}{2} \right]$

3.4 Codificación de Reed Solomon

Los códigos Reed Solomon [7] son una subclase importante de códigos no binarios BCH que son empleados para combatir los errores de ráfaga, razón por la cual son usados en aplicaciones de CD-ROM. La diferencia entre un código binario BCH y un código RS es que los códigos RS protegen mejor los símbolos múltiples de bits que los bits individuales. Excepto por el hecho de que los códigos RS operan sobre $GF(2^m)$, donde m es el número de bits por símbolo, en vez de $GF(2)$, las operaciones usadas para los códigos RS son las mismas para los códigos BCH. Así un codificador RS (n,k) despliega un bloque de k símbolos hacia un bloque de n símbolos por la adición de $n-k$ símbolos redundantes. Los parámetros para la corrección de un t -error del código RS son mostrados en la tabla 3.3.

Los códigos RS proveen un largo rango de codewords que pueden ser usados para diferentes tamaños de mensajes. Debido a la relación entre los códigos RS y BCH, la fácil

decodificación de los códigos BCH, tan bien como la asociación de los algoritmos de decodificación, puede ser extendida a los códigos RS.

Tabla 3.3 Propiedades de los códigos Reed Solomon.

Longitud de código	$n=2^m-1$	<i>símbolos</i>
Tamaño del mensaje	k	<i>símbolos</i>
Dígitos de chequeo de paridad	$n-k=2t$	<i>símbolos</i>
Distancia mínima	$d_{min}=2t+1$	<i>símbolos</i>
Capacidad de corrección de errores	$t = \left\lfloor \frac{n-k}{2} \right\rfloor$	<i>símbolos</i>

3.5 Códigos convolucionales.

Los códigos convolucionales inicialmente introducidos por Elias [8], son fundamentalmente diferentes a los códigos de bloque en que las secuencias de información no están agrupadas dentro de bloques distintos y codificados. En lugar de una secuencia continua de bits de información se mapea dentro de una secuencia continua de los bits de salida del codificador. Este mapeo es altamente estructurado habilitando un método de decodificación considerablemente diferente a los empleados por los códigos de bloques.

Un código convolucional es generado por la circulación de la secuencia de información a través de un registro de corrimiento de estado finito. En general el registro de corrimiento contiene $N(k \text{ bits})$ estados y m generadores de funciones algebraicas lineales basadas sobre polinomios generadores, como se muestra en la figura 3.1. El dato de entrada es cambiado dentro de una cantidad de k registros de corrimiento en un tiempo. El número de bits de salida para cada k bits de entrada de la secuencia de datos es n bits. El índice del código es $R_C = k/n$. El parámetro N es llamado límite de longitud e indica el número de bits de datos de entrada de los cuales la actual salida depende. M es el término de memoria del código y M cambios son requeridos para un bit de entrada que aparece en la salida. Esto determina que tan poderoso y complejo es el código.

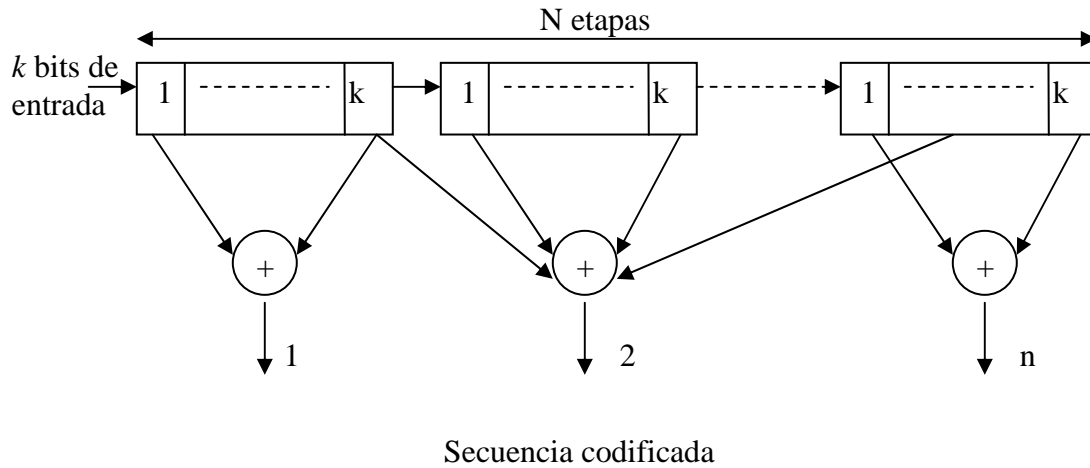


Figura 3.1 Diagrama general de bloque de un codificador convolucional.

3.5.1 Formas de representación de un código convolucional.

Un codificador convolucional puede ser representado por su respuesta a impulso, la cual es el resultado producido por el codificador cuando la entrada consiste de un solo 1 binario. Esta respuesta a impulso es mejor escrita como polinomio generador o equivalentemente como matriz generadora. La i -enésima trayectoria dentro del codificador tendrá como su polinomio generador

$$g^{(i)}(D) = g_0^{(i)} + g_1^{(i)}D + g_2^{(i)}D^2 + \dots + g_M^{(i)}D^M,$$

donde D^M representa el elemento de retraso m -enésimo en el registro de corrimiento.

Una matriz generadora para un código convolucional es semi-infinita ya que la entrada es semi-infinita en longitud. Así, que esta no puede ser una forma conveniente para representar un código convolucional.

Mientras que los polinomios generadores especifican un número n de vectores, uno para cada sumador (adders) usado de modulo-2 n . Cada vector de dimensión $2k$ indica la conexión del codificador hasta el sumador de modulo-2. Un 1 en la i -enésima posición del vector indica que el estado del registro de corrimiento correspondiente está conectado y un cero indica no conexión.

Una tabla lógica puede ser construida mostrando las salidas del codificador convolucional y el estado del codificador para la secuencia de entrada presente en el registro de corrimiento.

Debido a la presencia de los registros de corrimiento y que la trayectoria de cada bit de entrada resulta en un cambio en el estado del registro de corrimiento, el codificador también puede ser representado usando un diagrama de estado. El diagrama de estado es simplemente una gráfica de los posibles estados del codificador y las posibles

transiciones de un estado a otro. Hay diferentes formas del diagrama de estado como el código de árbol y el diagrama trellis.

El diagrama de árbol muestra la estructura del codificador en la forma de un árbol donde las ramas representan los diversos estados y las salidas del decodificador. El código de árbol tiene una estructura dependiente del tiempo donde cada instancia de tiempo t , también es referida como profundidad. El árbol consiste de 2^{kt} nodos, así la complejidad del código de árbol aumenta exponencialmente con t , lo cual lo hace impráctico para su uso.

El código trellis contiene la secuencia de códigos como trayectorias (similar al anterior). Las secuencias de código comparten nodos con estados iguales. Así se obtiene una gráfica con un número constante de nodos 2^N en cada nivel $t \geq M$. Hay un trellis mínimo y es único para cada código lineal. Éste es usado para la decodificación de máxima probabilidad (ML por sus siglas en inglés) de códigos convolucionales usando el algoritmo de Viterbi.

3.5.2 Decodificación de códigos convolucionales.

La función del decodificador es estimar la información de entrada codificada usando una regla o método que resulte en un mínimo posible número de errores. Así, el trabajo de un decodificador convolucional es estimar la trayectoria a través del trellis que fue seguido por el codificador.

Hay un número de técnicas para la decodificación de códigos convolucionales. El más importante de esos métodos es el algoritmo de Viterbi el cual realiza la decodificación de máxima probabilidad de un código convolucional. El algoritmo fue inicialmente descrito por A. J. Viterbi [9,10]. Ambas decodificaciones de decisión suave o dura pueden ser implementadas en los códigos convolucionales.

Otros algoritmos de decodificación para códigos convolucionales son la decodificación secuencial de Fano [11], el algoritmo snack, y la decodificación de retroalimentación.

3.5.2.1 Algoritmo de Viterbi

El algoritmo de Viterbi puede ser descrito de la siguiente forma.

El nodo trellis correspondiente al estado S_j en el tiempo i es denotado $S_{j,i}$. A cada nodo en el trellis se le asigna un valor $V(S_{j,i})$ basado sobre una métrica. Los valores del nodo son operados de la siguiente forma:

1. Poner $V(S_{0,0})=0$ e $i=1$.
2. En un tiempo i , calcular el métrico de la ruta parcial, para todas las trayectorias que entran en cada nodo.
3. Poner $V(S_{j,i})$ igual al métrico de la ruta parcial más pequeña que entra al nodo correspondiente al estado S_j en el tiempo i . El enlace puede ser roto por nodos previos

elegidos por una ruta aleatoria. Las ramas no sobrevivientes será borradas del trellis. De esta forma un grupo de trayectorias mínimas es creado de $S_{0,0}$.

4. Si $i < L + m$, donde L es el número de segmentos de códigos de entrada (k bits para cada segmento) y m es la longitud del registro de corrimiento más largo en el codificador, poner $i = i + 1$ y regresar al paso 2.

Una vez que todos los valores de los nodos han sido calculados, empezar con el estado S_0 , en el tiempo $i = L + m$, y continuar con las ramas sobrevivientes hacia atrás a través del trellis. La ruta así definida es única y corresponde a la salida del decodificador. Cuando una decodificación de decisión dura es realizada, la métrica usada es la distancia de Hamming, mientras que la distancia Euclidiana es usada para una decodificación de decisión suave.

3.6 Códigos Convolucionales Incluidos a través de Índice Compatible.

El diseño de un sistema de codificación con corrección de errores usualmente consiste de la selección de un código fijo con un cierto índice y una capacidad de errores igual a los requerimientos de protección de todos los datos a transmitir y adoptados a las peores condiciones promedio. En muchos casos, sin embargo, uno podría ser más flexible porque los datos a ser transmitidos tienen diferentes necesidades de protección de error y el canal es variante en el tiempo o se tienen insuficientes parámetros conocidos. Como la figura 3.2 lo muestra, la información a ser transmitida podría llevar información importante de la fuente (SSI por sus siglas en inglés) indicando diferentes requerimientos de protección.

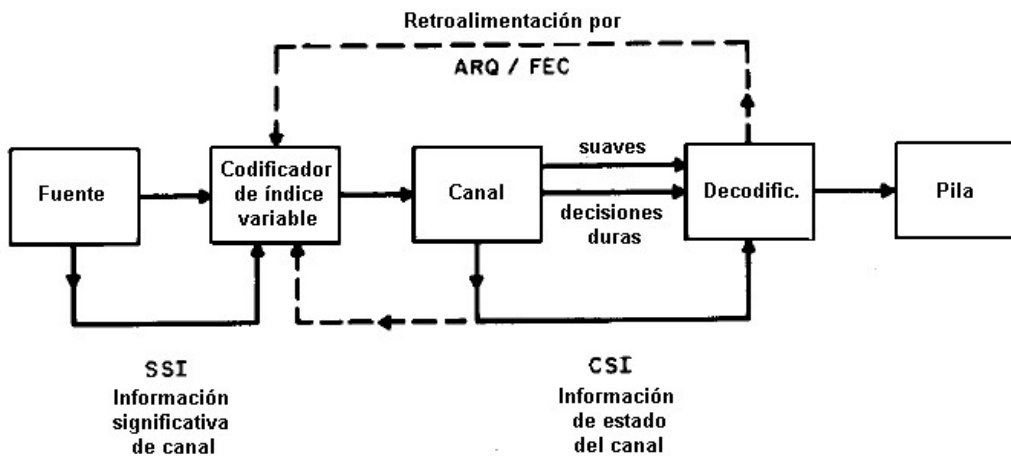


Figura 3.2 Esquema de transmisión codificada con información importante de fuente (SSI) e información de estado del canal (CSI).

Los escenarios mostrados en la figura 3.2 requieren de códigos variables adaptados para las necesidades de la fuente y el canal. Se desea cambiar el índice del código, es decir el número de bits de comprobación, y así la potencia de corrección del código durante la transmisión de un cuadro de información de acuerdo a las necesidades

de la fuente y el canal. Para propósitos prácticos, se desearía tener no justamente un conmutador entre el conjunto de codificadores y decodificadores, pero si un codificador y un decodificador el cual puede ser modificado sin cambiar su estructura básica. Esto puede lograrse por la no transmisión de ciertos bits de código, llamados introducidos a través del código (puncturing). Mandelbaum [12] fue el primero en proponer los códigos punctured para la transmisión de redundancia en pasos incrementales usando códigos Reed Solomon. Para alojar las decisiones suaves y la información de estado del canal (CSI por sus siglas en inglés) en el receptor, un decodificador de máxima probabilidad es requerido. Esto motivo el uso de códigos convolucionales y el algoritmo de Viterbi para la decodificación.

Los códigos convolucionales punctured fueron al principio introducidos por Cain, Clark y Geist [13] con el propósito principal de obtener un decodificador de Viterbi más simple para códigos de índice K/N con dos ramas que llegan a cada nodo en lugar de 2^K ramas. Ellos obtuvieron códigos con índices de $2/3$ y $3/4$ con códigos de índice punctured de $1/2$. Después, Yasuda [14, 15] encontró una familia de códigos de $(N-1)/N$ con códigos punctured de $1/2$ para N hasta 14, y construyó codificadores de índice seleccionable y decodificadores Viterbi usando decisiones suaves.

El concepto de códigos convolucionales es modificado por la generación de familias de códigos por la adición de una restricción de índice de compatibilidad para la regla de punctured. La restricción implica que todos los bits del código de un código de índice punctured alto son usados por los códigos de índice bajo, o en otras palabras, los códigos de índice alto son incrustados dentro de los códigos de índices más bajos de la familia. Si los códigos de índice alto no son lo suficientemente poderosos para decodificar los errores de canal, solamente los bits suplementarios los cuales fueron previamente introducidos (punctured) deben ser transmitidos para mejorar el código. Sin embargo, desde que estos códigos son compatibles la variación del índice dentro de un cuadro de datos es posible para mejorar la protección de error única. A esta clase de códigos se les llama códigos convolucionales incluidos a través de índice compatible [16] (RCPC por sus siglas en inglés).

3.6.1 Definición de los códigos RCPC

Una familia de códigos RCPC es descrita por un código madre de índice $R=1/N$ y memoria M teniendo una matriz tap generadora

$$g = \begin{matrix} \uparrow & \leftarrow M+1 \rightarrow \\ N & (g_{ik}) \\ \downarrow \end{matrix} \quad 3.15$$

con las conexiones tap $g_{ik} \in (0,1)$ donde 1 representa una conexión del estado de registro de cambio k enésima a la salida i enésima. Junto con N , el período puncturing P determina el rango de índices de código

$$R = \frac{P}{P+1} \quad l = 1, \dots, (N-1)P \quad 3.16$$

entre $P/(P+1)$ y $1/N$. Los códigos RCPC son códigos punctured del código madre con matrices puncturing

$$a(l) = \overset{\uparrow}{N} \overset{\leftarrow P \rightarrow}{(a_{ij}(l))} \quad 3.17$$

con $a_{ij}(l) \in (0,1)$ donde 0 implica puncturing.

La restricción de compatibilidad de índice implica la siguiente regla:

si $a_{ij}(l_0)=1$ entonces $a_{ij}(l)=1$ para toda $l \geq l_0 \geq 1$
o equivalente

si $a_{ij}(l_0)=0$ entonces $a_{ij}(l)=0$ para toda $l \geq l_0 \geq (N-1)P-1$

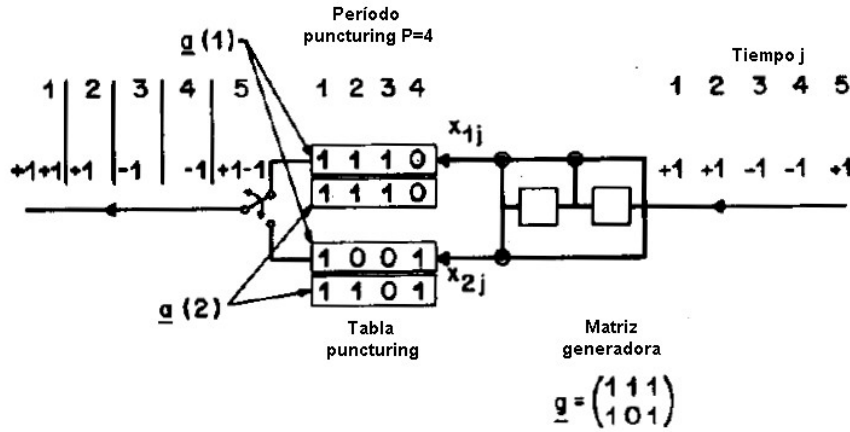


Figura 3.3 Ejemplo de un código convolucional con dos tablas de puncturing de índice compatible.

Para explicar los códigos punctured de índice compatible, se muestra el ejemplo de la figura 3.3 donde un código convolucional con índice de $R=1/2$ con memoria $M=2$ es periódicamente punctured con un período $P=4$. Se describen los símbolos de información binaria y los símbolos de código binario como x_{1j} y x_{2j} por valores ± 1 . Un cero en la tabla de puncturing significa que el símbolo del código no será transmitido. En el ejemplo superior de la figura 3.3, el cuarto bit del brazo superior y el segundo y tercer bit del brazo inferior no son transmitidos. La tabla puncturing puede ser vista como una regla de modulo P para multiplexar los dos flujos de códigos de bits. En lugar de transmitir $2 \cdot P=2 \cdot 4$ solamente $P+l=P+1=5$ bits son transmitidos por cada $P=4$ bits de información. Por lo tanto se ha generado un código de índice $R=4/5$ por el uso solamente de $P+l$ "1"s en lugar de $2P$ "1"s en la tabla de puncturing, la cual es descrita por la matriz $N \times P$

$$a(1) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Suponga el índice del código 4/5 y con la tabla puncturing $a(1)$ no es lo suficientemente poderoso para corregir los errores de canal. Un código más redundante y por lo tanto más poderoso con índices más bajos 4/6, 4/7, o 4/8 sería necesario. En lugar de transmitir todos los bits del código de un código de índice bajo completamente diferente, el código de índice más bajo utilizaría los bits ya transmitidos. Entonces solamente los bits de redundancia incremental adicional deben ser transmitidos. "1"s adicionales en la tabla puncturing de los códigos de índices más bajos pueden por lo tanto

ser colocados donde los ceros aparecen en la matriz puncturing del código previo de índice más alto, por ejemplo,

$$a(2) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}, a(3) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}, a(4) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

Así para $l=1$ hasta 4, se obtiene una familia compatible de códigos derivados del código madre $1/2$ con índices $4/5$, $4/6$, $4/7$, $4/8$ los cuales usan la redundancia incremental. La pregunta, por supuesto, es si todos esos códigos en la familia son buenos y si existen códigos no catastróficos. En el lado del receptor el decodificador utiliza el algoritmo de Viterbi (VA por sus siglas en inglés) que tiene que conocer la regla a puncturing actual. El VA recibe los valores reales y_{ij} cuando una decisión suave es usada. Alternativamente, el decodificador puede usar valores cuantizados de y_{ij} . El VA encuentra la trayectoria con la métrica de probabilidad máxima para y_{ij} estadísticamente independiente por el calculo

$$\max_m \sum_{j=1}^J \lambda_j, \quad 3.18$$

donde el incremento métrico es

$$\lambda_j = \sum_{i=1}^N a_{ij} x_{ij}^{(m)} y_{ij} \quad 3.19$$

y $a_{ij+p} = a_{ij}$ debido a un puncturing periódico. $a_{ij}=0$ significa que ese $x_{ij}^{(m)}$ no ha sido transmitido y y_{ij} no es disponible. El algoritmo de Viterbi opera sobre la malla del código madre $1/N$ con dos ramas llegando a cada estado. La ecuación de incremento métrico simplemente significa que para $a_{ij}=1$ el ROM métrico es accesado usando la dirección $x_{ij}^{(m)} y_{ij}$ para sumar un incremento y para $a_{ij}=0$ este no es accesado.

Para un canal de desvanecimiento, la señal transmitida es multiplicada por un factor de desvanecimiento a_F . Si la interferencia es asumida, a_F es una variable aleatoria real positiva independiente con densidad $p(a_F)$, es decir Rayleigh o Rice. Si este factor puede ser estimado en el receptor y es usado como CSI el métrico-ML para una transmisión coherente es:

$$\lambda_j = \sum_{i=1}^N a_{ij} a_{iF} x_{ij}^{(m)} y_{ij}, \quad 3.20$$

como esperaba, un desvanecimiento profundo ($a_{iF} \rightarrow 0$) tiene el mismo efecto como puncturing ($a_{ij}=0$)

3.6.2 Criterio de calidad para códigos RCPC

Con la decodificación de Viterbi el criterio óptimamente usual es una distancia libre grande d_{free} , un número pequeño de trayectorias a_d , y un peso de error de información pequeño c_d sobre todas las trayectorias con $d \geq d_{free}$. Más específicamente, uno tiene para la frontera superior de Viterbi una probabilidad de evento de error

$$P_E \leq \frac{1}{P} \sum_{d=d_{free}}^{\infty} a_d P_d \quad 3.21$$

y para la probabilidad de error de bit

$$P_b \leq \frac{1}{P} \sum_{d=d_{free}}^{\infty} c_d P_d \quad 3.22$$

P_d es la probabilidad que una trayectoria errónea a una distancia d es seleccionada. El así llamado espectro de distancia $\{a_d\}$ y $\{c_d\}$, el cual será tan pequeño como sea posible, depende solamente del código, a saber, sobre N, M, P, g , y $a(l)$. Debido a la naturaleza variante del tiempo de los códigos RCPC, diferentes puntos de inicio P para trayectorias divergentes tienen que ser consideradas para obtener los números totales de a_d y c_d . Desde que los códigos RCPC son una subclase de códigos punctured, los cuales en cambio son una subclase de los códigos generales de K/N , estos todavía tienen que mostrar que la restricción de los códigos RCPC no es tan severa para prevenir códigos buenos de ser localizados en esa subclase. Un grado de libertad puede ser ganado por el incremento del período P . Los códigos RCPC constituyen una clase específica de códigos variantes en el tiempo con un generador fijo pero con un puncturing variante en el tiempo periódicamente. Se sabe que algunos códigos variantes en el tiempo son mejores que los códigos fijos y algunos teoremas de codificación aleatoria sostienen solo códigos variantes en el tiempo. Por lo tanto, algunas mejoras pueden ser posibles usando un período más grande P . Note que P puede ser lo suficientemente grande si códigos de índices específicos son acontecidos en la familia de códigos RCPC. La complejidad del decodificador VA de un código RCPC es términos de suma, comparación y operaciones seleccionadas por bit decodificado es $2 \cdot 2^M$, mientras que un código general K/N requiere $(2 \cdot 2^M)/K$, en tales operaciones M es la memoria total del decodificador. En el decodificador RCPC, una ambigüedad $N \cdot P$ tiene que ser resuelta en el flujo de datos entrante y la decisión de profundidad de los códigos punctured son generalmente más grandes. Esta última desventaja no es tan grave, cuando, como en la mayoría de las aplicaciones, los flujos de datos son transmitidos con palabras de sincronía y una terminación adecuada de cuadros cortos.

Un método no constructivo es conocido para determinar la matriz generadora g y la matrices punctured $a(l)$ para una familia de códigos RCPC. Por lo tanto una búsqueda por computadora ha sido llevada a cabo bajo algunas restricciones. Para distancias $d \geq d_{free}$, a_d , c_d , y distribuciones de longitud de trayectoria han sido determinadas por un algoritmo de pila.

Los pasos básicos para encontrar una familia de códigos RCPC buenos son:

1. Seleccionar el código madre con índice l/N , memoria M , y generador g . Seleccionar el período puncturing P . Para $l=l_{max}=(N-1)P$ poner $a(l)=1$. Para $l=(N-1)P$ a 2 realizar los pasos 2-4.
2. Entre los elementos $a_{ij}(l)$ los cuales son "1" seleccionar uno y ponerlo a "0" (de acuerdo a la regla de compatibilidad de índice). Observe que las permutaciones cíclicas de la columna dan el mismo funcionamiento de código.
3. Determina la d_{free} y c_d para $d \geq d_{free}$.
4. Repita el paso 2 y 3 hasta encontrar $a(l-1)$ con una máxima d_{free} y un mínimo c_d para $d=d_{free}$, $d_{free}+1$. Si códigos catastróficos solo son encontrados usar el mejor código próximo en el nivel l .

Referencias.

- [1] C.E. Shannon, "The mathematical theory of communication", Bell Systems Technical Journal, vol 27, 1948.
- [2] S. Lin y D. J. Jr. Castello "Error Control Coding: Fundamentals and application", Prentice-Hall, Englewood, Cliffs, NJ. 1983.
- [3] R. W. Hamming "Error detecting and error correcting codes", Bell Systems Technical Journal, vol. 29, 1950.
- [4] R. C. Bose and D.K. Ray Chaudhuri, "On a class of error correcting binary group codes", Information and Control, vol. 3, March 1960.
- [5] R. C. Bose and D.K. Ray Chaudhuri, "Further results on error correcting binary group codes", Information and Control, vol. 3, September 1960.
- [6] A. Hocquenghem, "Codes correcteurs d'erreurs", Chiffres, vol. 2, 1959.
- [7] I.S. Reed and G. Solomon, "Polynomial codes over certain finite fields" SIAM Journal on Applied Mathematics, vol. 8.
- [8] P. Elias, "Coding for noise channels", IRE Convention Record, 1955.
- [9] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", IEEE Transactions on Information Theory, Vol. IT-13, April 1967.
- [10] A. J. Viterbi, "Convolutional codes and their performance in communications systems", IEEE Transactions on Information Theory, Vol. COM-19, October 1971.
- [11] R. M. Fano "A heuristic discussion of probabilistic coding" IEEE Transaction on Information Theory, Vol. IT-9, pp. 64-74, April 1963.
- [12] D. M. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy", IEEE Trans. Inform. Theory, Vol IT-20, pp. 388-389, May 1974.
- [13] J. B. Cain, G. C. Clark y J. M. Geist "Punctured convolutional codes of rate $(n-1)/n$ and simplified maximum likelihood decoding", IEEE Trans. Inform. Theory, Vol IT-25, pp. 97-100, Jan 1979.
- [14] Y. Yasuda, "Development of variable-rate Viterbi decoder and its performance characteristics" en Proc. 6th. Int. Conf. Digit. Satellite Commun., Phoenix, AZ, Sept. 1983, pp. XII-24-XII-31.
- [15] Y. Yasuda, K. Kashiki, and Y Hirata "High rate punctured convolutional codes for soft decision Viterbi decoding", IEEE Trans. Commun., vol. COM-32, pp. 315-319, Mar 1984.
- [16] J. Hagenauer, "Rate-compatible Punctured Convolutional Codes (RCPC codes) and their application", IEEE Trans. Commun., vol 36, No. 4, pp 389-400, April 1988.
- [17] Theodore S. Rappaport, "Wireless Communications Principles and Practice", Communication engineers and emerging technologies series, Edit. Prentice Hall, USA 1996.
- [18] Vijay K Garg, Joseph E. Wilkes, "Wireless Personal & Communications Systems", Edit. Prentice Hall, USA, 1996.
- [19] Bossert Martin, "Channel Coding for telecommunications", Edit Jhon Wiley & Sons, LTD, edition in English, England 1999.
- [20] NGan, Yap y Tan, "Video Coding for Wireless Communication Systems", Marcel Dekker, Inc., E.U.A 2001.

Capítulo 4

Modelos de canal.

Los canales de radio móviles ponen limitaciones fundamentales en el funcionamiento de los sistemas de comunicación móvil. La trayectoria de transmisión entre el transmisor y el receptor puede variar desde una simple de línea de vista hasta una que es severamente obstruida por edificios, montañas o follaje. Por tal motivo los canales de radio son extremadamente aleatorios y no ofrecen un análisis fácil. Modelar un canal de radio ha sido históricamente una de las partes más difíciles del diseño de un sistema de radio móvil, y es típicamente hecho de un modo estadístico basado en medidas hechas específicamente para un sistema de comunicaciones deliberado o para una asignación de espectro.

Los modelos de propagación han sido enfocados para predecir el promedio de la fuerza de la señal recibida a una distancia dada del transmisor, tan bien como la variabilidad de la fuerza de la señal en proximidades espaciales cercanas a una ubicación particular. Los modelos de propagación que predicen la fuerza promedio de la señal para una distancia arbitraria entre el transmisor y el receptor son útiles en la estimación del radio del área de cobertura de un transmisor y son llamados modelos de propagación de larga escala (large-scale en inglés) cuando se considera la propagación de la onda a largas distancias. Por otra parte, los modelos de propagación que caracterizan las rápidas fluctuaciones de la fuerza de la señal recibida sobre distancias recorridas muy cortas o en duraciones de tiempo cortos son llamados modelos de desvanecimiento (fading en inglés) o de escala pequeña.

4.1 Mecanismos básicos de propagación

La reflexión, la difracción y la dispersión son los tres mecanismos básicos de propagación en un sistema de comunicación móvil.

La reflexión ocurre cuando una onda electromagnética en propagación golpea sobre un objeto el cual tiene una dimensión más larga comparada con la longitud de onda de la onda propagada. La reflexión ocurre sobre la superficie terrestre, edificios y paredes. Cuando la onda electromagnética incide sobre una superficie entre dos medios transparentes se observa que una parte de la onda electromagnética es reflejada en su camino usual mientras que el resto es transmitido a través del segundo medio. Este doblamiento de la onda electromagnética es llamado refracción.

La difracción ocurre cuando la trayectoria de transmisión entre el transmisor y el receptor es obstruida por una superficie que tiene formas irregulares (bordes). Las ondas secundarias resultantes por la superficie de obstrucción se hacen presentes por todo el espacio e incluso detrás del obstáculo, aumentando la inclinación de las ondas alrededor del obstáculo, por lo que la línea de vista entre el transmisor y el receptor no existe.

La dispersión resulta cuando el medio a través del cual la onda viaja consiste de objetos con dimensiones menores comparadas con su longitud de onda, y donde el número de obstáculos por unidad de volumen es grande. En la práctica el follaje, las señales de tránsito, y los postes de luz inducen dispersión en un sistema de comunicación móvil.

4.2 Ambientes de propagación.

Los ambientes de propagación para un sistema de comunicación caen dentro de tres categorías: canales inalámbricos interiores, exteriores/urbanos y satelitales. El canal puede ser modelado usualmente como un sistema de banda estrecha o de banda ancha dependiendo del ancho de banda de transmisión y del ancho de banda coherente del canal junto con el período de transmisión y el período coherente del sistema.

Cuando una señal de comunicación es transmitida sobre un canal inalámbrico interior, la señal transmitida se descompone en múltiples copias de la señal original correspondiendo a reflexiones de trayectoria múltiple alrededor del medio ambiente como se muestra en la figura 4.1. Esto se debe a factores como la reflexión, refracción, difracción y dispersión dentro del medio interior, por lo que el sistema de comunicación se descompone en trayectorias múltiples de la onda electromagnética, donde cada trayectoria experimenta atenuación, un retraso de fase y de tiempo. En el receptor la señal consiste de la superposición de esas trayectorias. Debido a la naturaleza aleatoria de esos efectos del canal multitrayectoria las trayectorias pueden ser sumadas constructivamente o destructivamente.

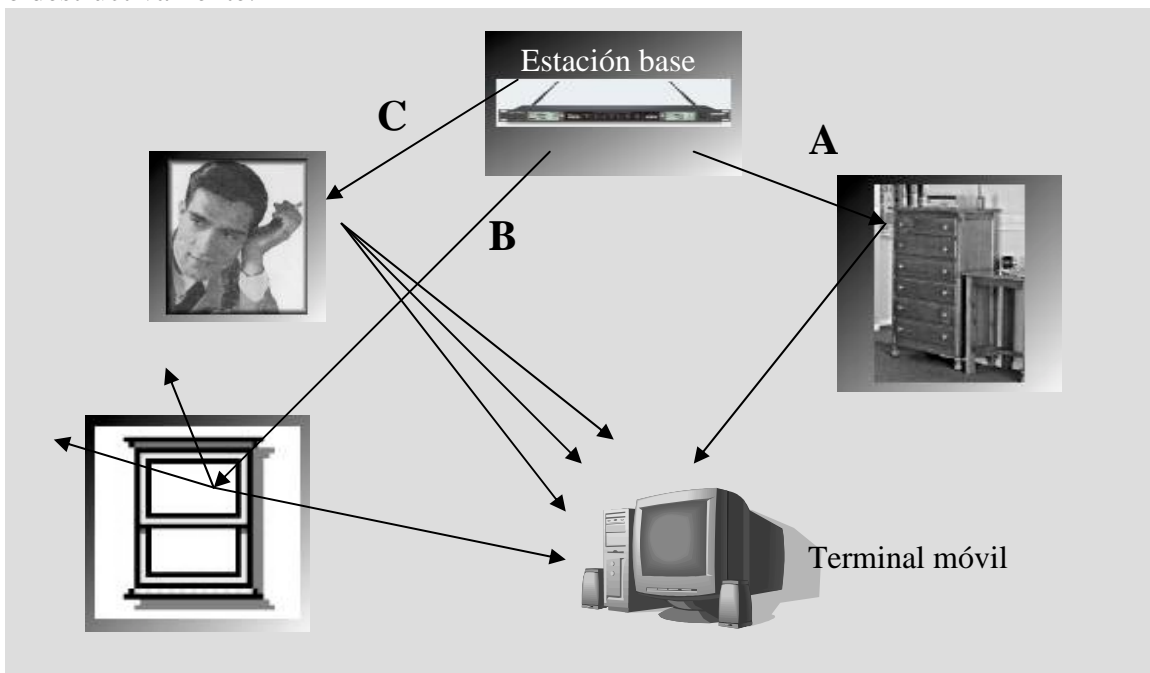


Figura 4.1 Canal inalámbrico interior. a) Reflexión por muebles. b) Refracción a través de las ventanas c) dispersión y difracción por el movimiento de objetos.

Los tres principales efectos de propagación afectan también las transmisiones de radio urbano o exteriores. Estos efectos de propagación en un ambiente urbano son mostrados en la figura 4.2. En una escala muy larga, el canal inalámbrico exterior/urbano puede ser categorizado de acuerdo a sus modos de propagación sobre la atmósfera de la tierra, dependiendo de la frecuencia de transmisión y de otros factores. Los efectos de la ionosfera tienden a ser menos significativos cuando las frecuencias de radio sobrepasan el umbral de 3GHz, con algunos casos excepcionales como son la atenuación por gases e hidrométricos. Hay cuatro diferentes tipos de modo de propagación de la onda en los canales de comunicación inalámbrica exterior como se muestra en la figura 4.3. La propagación de onda de tierra se compone de una onda directa, de una onda reflejada en la tierra y una onda de superficie propagada a lo largo de la superficie terrestre. El segundo tipo de onda de propagación es la onda de cielo o ionosférica, esta onda es menor a los 300MHz y la onda de radio se propaga reflejándose en ambas superficies de la tierra y la ionosfera. Para las ondas de radio entre los 30MHz a 3GHz la comunicación a largas distancias sobre el horizonte puede ser establecida por la dispersión de energía del índice refractivo de irregularidades de la troposfera de la tierra, estas ondas de propagación son llamadas troposferitas o de dispersión hacia atrás. Finalmente en las frecuencias arriba de los 3GHz las propagaciones de línea de vista (LOS por sus siglas en inglés) directa son las predominantes y son principalmente usadas en las comunicaciones satelitales. Los principales problemas para este modo de propagación vienen de la atenuación dentro de ciertas bandas de absorción de la atmósfera terrestre.

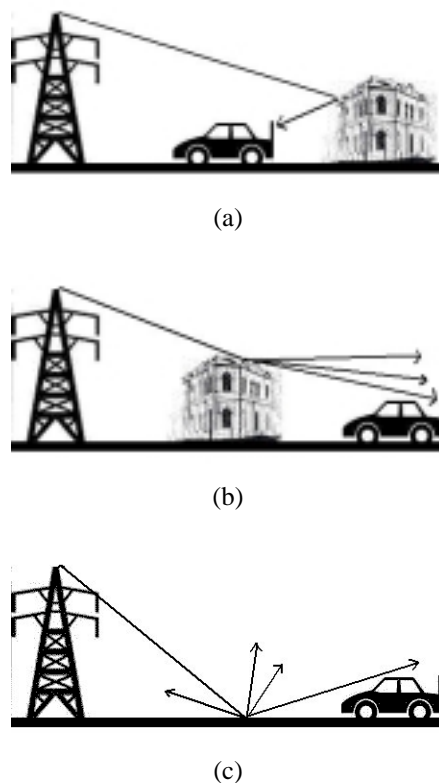


Figura 4.2 Canal inalámbrico exterior. a) Reflexión por edificios. b) Difracción por bordes de edificios c) dispersión por superficies irregulares.

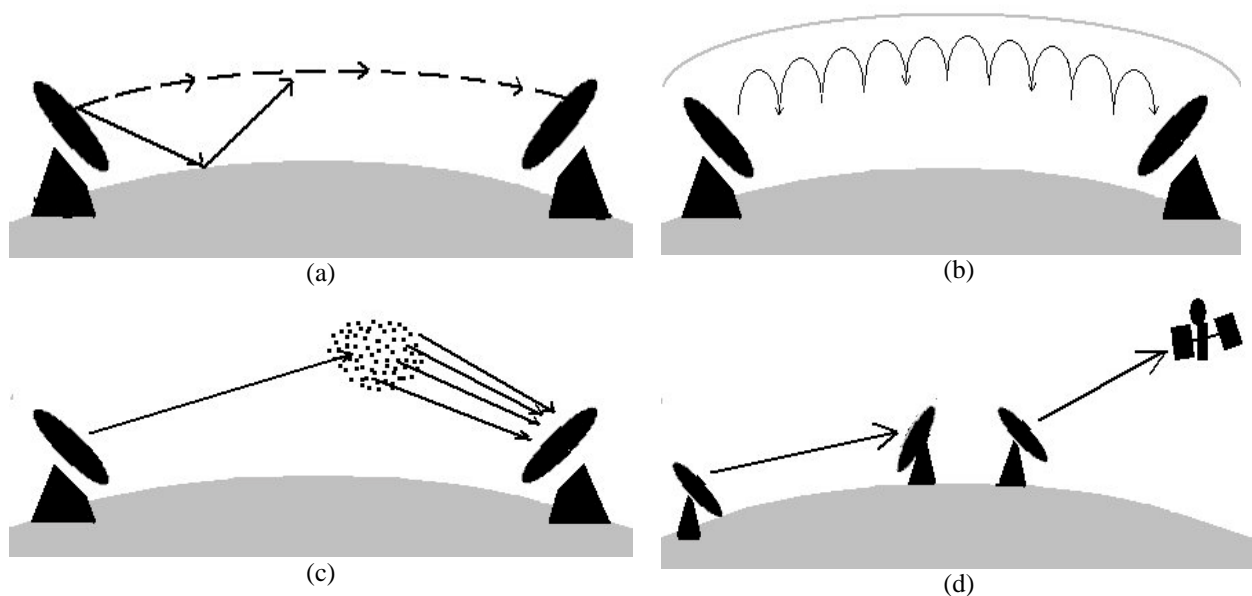


Figura 4.3 Diferentes modos de propagación de las ondas de radio en un canal inalámbrico exterior.

Los canales inalámbricos satelitales también presentan muchos de los fenómenos de los canales de radio mencionados anteriormente. No obstante las atenuaciones atmosféricas son los obstáculos más grandes por afrontar. Las atenuaciones por gases son un factor en la reducción de la amplitud de la señal para cualquier transmisión satelital. Este proceso de absorción es debido a la presencia de oxígeno y vapor de agua en la trayectoria de propagación de la onda de radio. La atenuación por gases incrementa con la frecuencia, y es dependiente de la temperatura, presión y humedad. Una reducción de la amplitud y de dispersión también puede ser provocada por los elementos hidrométricos (lluvia, nubes, nieve, niebla, hielo) en la trayectoria de propagación. Las atenuaciones por lluvias causan grandes deterioros en la señal, especialmente en las bandas de frecuencias arriba de los 10GHz. Las atenuaciones por nubes y niebla son menos severas que las atenuaciones por lluvia, sin embargo son consideradas especialmente para las bandas de frecuencias arriba de los 15GHz. Los efectos por nieve y hielo son mínimos y generalmente desapercibidos para los canales de 30GHz o menores.

Otros factores que afectan las transmisiones satelitales son el ruido, la despolarización, el centelleo ionosférico, el retraso de grupo, el desvanecimiento de multitrayectoria, la refracción y la dispersión.

4.3 Modelos de pérdidas de trayectorias a largas distancias (Large scale path loss models)

En esta parte se introducirá a algunos modelos de pérdidas de trayectorias a largas distancias para canales inalámbricos exteriores o urbanos y satelitales.

4.3.1 Modelo de propagación en el espacio libre

El modelo de propagación en el espacio libre es usado para predecir la intensidad de la señal recibida cuando el transmisor y el receptor tienen un claro, es decir una línea de vista no obstruida entre ellos. Los sistemas de comunicación satelital y las microondas en sus enlaces de líneas de vista utilizan este modelo de propagación. El modelo de espacio libre predice el decaimiento de la potencia recibida como función de la distancia de separación entre T-R. La potencia recibida en el espacio libre por una antena receptora, la cual está separada de una antena transmisora radiante por una distancia d , es dada por la ecuación de espacio libre de Friis.

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L},$$

donde P_t es la potencia del transmisor, $P_r(d)$ es la potencia recibida como una función de la separación T-R, G_t es la ganancia de la antena transmisora, G_r es la ganancia de la antena receptora, d es la distancia de separación en metros, L es el factor de pérdidas no vinculado a la propagación ($L \geq 1$), y λ es la longitud de onda en metros.

Los valores para P_t y P_r deben ser expresados en con las mismas unidades, y G_t y G_r son cantidades adimensionales. Las diversas pérdidas L son usualmente debidas a la atenuación en la línea de transmisión, las pérdidas de los filtros, y las pérdidas en las antenas del sistema de comunicación. Un factor de $L=1$ indica que no hay pérdidas en el sistema de hardware.

La ecuación de espacio libre de Friis muestra como la potencia recibida cae con el cuadrado de la distancia de separación entre T-R. Esto implica que la potencia recibida decae con la distancia a un índice de 20dB/década.

Las pérdidas de trayectoria, las cuales representan la atenuación de la señal como una cantidad positiva de medida en dB, es definida como la diferencia (en dB) entre la potencia transmitida eficaz y la potencia recibida, y puede o no incluir el efecto de la ganancia de las antenas. Las pérdidas por trayectoria (PL) para el modelo de espacio libre es dado por:

$$PL(dB) = 10 \log \frac{P_t}{P_r} = -10 \log \left[\frac{G_t G_r \lambda^2}{(4\pi)^2 d^2} \right],$$

si las ganancias de las antenas no excluidas se asumen que tienen ganancia unitaria.

4.3.2 Modelo de reflexión en tierra (2-rayos)

En un canal de radio exterior generalmente cuenta con más trayectorias que la trayectoria de línea de vista que se asume en el modelo de espacio libre. En algunos casos el modelo de espacio libre es inexacto cuando se usa solo, ya que no todas las rutas de propagación experimentan un comportamiento de espacio libre. El modelo de reflexión en tierra de 2-rayos mostrado en la figura 4.4 es un modelo de propagación útil que está basado en la óptica geométrica, y considera ambas rutas de propagación la directa y la reflejada en tierra entre el transmisor y el receptor. Este modelo ha sido encontrado por ser

razonablemente exacto para predecir la intensidad de señal a larga escala sobre distancias de muchos kilómetros para sistemas de radio móviles que usan torres altas (alturas mayores de 50 metros), tan bien como para canales de línea de vista de microceldas en ambientes urbanos.

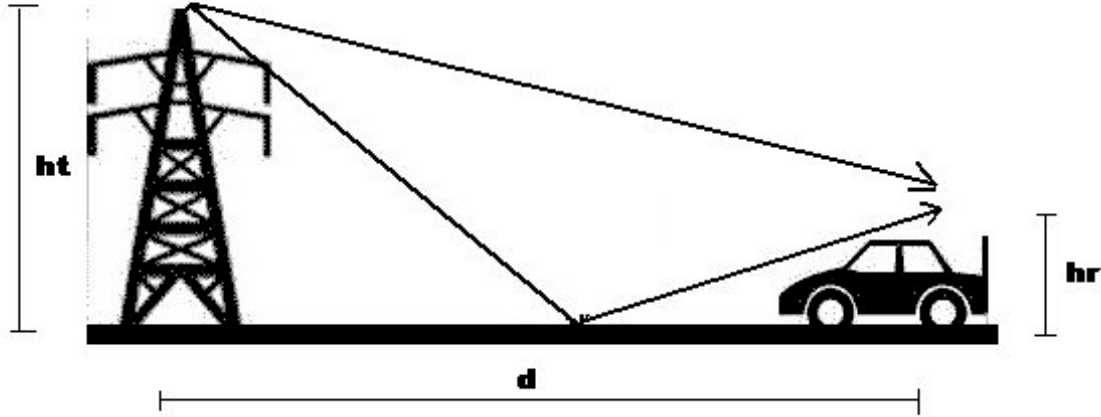


Figura 4.4 Modelo de reflexión en tierra de 2-rayos.

En este modelo la máxima distancia de separación entre T-R es como máximo algunas decenas de kilómetros, y la tierra se asume plana u horizontal. La potencia de la señal recibida es dada por:

$$P_r = 10 \log_{10} \left(\frac{P_t G_t G_r \lambda^2}{(2\pi)^2 d^2} \operatorname{sen}^2 \left(\frac{2\pi h_t h_r}{\lambda d} \right) \right),$$

donde h_t es la altura de la antena transmisora desde la tierra en metros y h_r es la altura de la antena receptora desde tierra en metros. Usando la aproximación de $\operatorname{sen} x \approx x$ para x pequeñas cuando $d \gg h_t, h_r$, y la ecuación anterior queda reducida a:

$$P_r = 10 \log_{10} \left(\frac{P_t G_t G_r \lambda^2 h_t^2 h_r^2}{d^4} \right).$$

Este modelo difiere con el modelo de espacio libre en dos formas. Primero la atenuación en la intensidad de la señal recibida es independiente de la frecuencia de la portadora usada y en segundo término, la potencia de la señal recibida decae la cuarta parte de la potencia sobre la distancia. Las pérdidas por trayectoria para el modelo de reflexión en tierra de 2-rayos son:

$$PL(dB) = 10 \log_{10} \left(\frac{P_t G_t G_r}{P_r} \right) = -10 \log_{10} \left(\frac{\lambda^2}{4\pi^2 d^2} \operatorname{sen}^2 \left(\frac{2\pi h_t h_r}{\lambda d} \right) \right).$$

Existen otros cálculos complementarios que deben de tomarse en cuenta para obtener un mejor resultado al modelar un canal de radio en el espacio libre que miden las pérdidas por obstrucciones en la línea de vista, difracción y dispersión.

4.3.3 Modelo de pérdida de trayectoria a largas distancias.

Los principales modelos de propagación de radio son derivados usando una combinación de métodos analíticos y empíricos. Estas aproximaciones empíricas están basadas sobre curvas apropiadas o expresiones analíticas que recrean un conjunto de datos medidos. Ambos modelos de propagación los teóricos y los basados en medidas indican que promedio de la intensidad de la señal recibida decrece logarítmicamente con la distancia. El promedio de pérdidas de trayectoria a larga escala para una separación arbitraria de T-R es expresado como una función de la distancia con el uso de un exponente de pérdidas de trayectoria, n .

$$\overline{PL}(d) \propto \left(\frac{d}{d_0}\right)^n \quad \text{o} \quad \overline{PL}(dB) = \overline{PL}(d_0) + 10n \log\left(\frac{d}{d_0}\right)$$

donde n es el exponente por pérdidas de trayectoria el cual indica el cambio en que las pérdidas de trayectoria aumentan con la distancia, d_0 es la distancia de referencia cercana, la cual es determinada de las mediciones cerca del transmisor, y d es la distancia de separación entre el T-R. El valor de n depende de un ambiente de propagación específico. Por ejemplo, en el espacio libre, n es igual a 2, y cuando hay obstrucciones, n tendrá un valor más grande.

El modelo de la ecuación anterior no considera el hecho de que un desorden en el ambiente circundante pueda ser bastante diferente en dos diferentes lugares teniendo la misma separación T-R. Por lo que la distribución logarítmica normal describe los efectos de oscurecimiento los cuales ocurren en un gran número de lugares los cuales tienen la misma separación de T-R, pero tienen diferentes niveles de desorden sobre la trayectoria de propagación. Este fenómeno es referido como ensombrecimiento log-normal.

Es claro que debido a los efectos aleatorios de ensombrecimiento, algunos lugares dentro del área de cobertura estarán bajo un umbral particular deseado de la señal recibida. Esto ocasionalmente es útil para computar los límites de cobertura relacionados con el porcentaje de área cubierta dentro de los límites.

4.3.4 Modelos de propagación exterior.

La transmisión radiofónica en un sistema de comunicaciones móviles frecuentemente toma lugar sobre terrenos irregulares. Un número de modelos de propagación son disponibles para predecir las pérdidas por trayectoria sobre terreno irregular. Mientras todos esos modelos ayudan a predecir la intensidad de la señal sobre un punto receptor particular o en un área local específica (llamada sector), los métodos varían ampliamente en su aproximación, complejidad y precisión. La mayoría de estos modelos están basados en una interpretación sistemática de los datos de medición obtenidos en el área de servicio.

4.3.4.1 Modelo de Longley-Rice

El modelo de Longley-Rice [1] es aplicable a sistemas de comunicación punto a punto en el rango de frecuencias de 40MHz a 100GHz, sobre diferentes tipos de terreno. La media de las pérdidas de transmisión es predicha usando la geometría de la trayectoria del perfil del terreno y la refractividad de la troposfera. El modelo de predicción de propagación de Longley-Rice es también referido como el modelo de terreno irregular ITS (por sus siglas en inglés).

El método de Longley-Rice opera en dos modos. Cuando un perfil de trayectoria de terreno detallado es disponible, los parámetros de trayectoria específicos pueden ser fácilmente determinados y la predicción es llamada modo de predicción punto a punto. Sobre la otra forma, si el perfil de trayectoria del terreno no es disponible, el método de Longley-Rice provee técnicas para estimar los parámetros de trayectoria disponibles, y tal predicción es llamada predicción de modo de área.

4.3.4.2 Modelo de Durkin

Una aproximación de predicción de propagación clásica similar a la usada por Longley-Rice es discutida por Edwards y Durkin [2], tan bien como Dadson [3]. Estos documentos describen un simulador por computadora, para predecir el contorno de la intensidad de campo sobre terreno irregular. Aunque este simulador solo predice fenómenos a larga escala, provee una interesante perspectiva de la naturaleza de propagación sobre terreno irregular y las pérdidas causadas por los obstáculos en la trayectoria radiofónica.

La ejecución del simulador de pérdidas de trayectoria Durkin consiste de dos partes. La primera parte accesa a la base de datos topográficos de un área de servicio propuesta y reconstruye la información del perfil de tierra a lo largo de la parte radial que ensambla el transmisor al receptor. La segunda parte del algoritmo de simulación calcula las pérdidas de trayectoria esperadas a lo largo del área radial. Hecho esto, la ubicación del receptor simulado puede ser interactivamente movida a diferentes ubicaciones en el área de servicio para deducir el contorno de la intensidad de la señal.

Este método es muy atractivo porque puede leer un mapa digital de elevación y llevar a cabo una operación de propagación en un sitio específico sobre los datos de elevación. Además éste produce un buen contorno de intensidad de señal en pocos dB. La desventaja de este método es que este no puede predecir adecuadamente los efectos de propagación debidos al follaje, edificios u otras estructuras hechas por el hombre.

4.3.4.3 Modelo de Okumura

El modelo de Okumura es uno de los modelos más ampliamente usados para la predicción de señales en zonas urbanas. Este modelo es aplicable para frecuencias en el rango de 150MHz a 1920MHz (aunque es típicamente extrapolado hasta 3000MHz) y distancias de 1km a 100km. Este puede ser usada para alturas de antenas de una estación base que van desde los 30 m hasta 1000m.

Okumura desarrollo un conjunto de curvas dadas por la atenuación media en el espacio libre (A_{mu}), en un área urbana sobre un terreno cuasi-liso con una altura efectiva de la antena de la estación base (h_{te}) de 200m y una altura de antena móvil (h_{re}) de 3 m. Esas curvas fueron desarrolladas de extensivas mediciones usando antenas verticales omni-direccionales en ambas la base y la móvil, y fueron graficadas como una función de la frecuencia en el rango de 100MHz a 1920MHz y como una función de la distancia de la estación base en el rango de 1km a 100km. Para determinar las perdidas de trayectoria usando el modelo de Okumura, las perdidas en el espacio libre entre los puntos de interés es primero determinada, y entonces el valor de $A_{mu}(f,d)$ es añadido a éste por medio de los factores de corrección para explicar el tipo de terreno.

Otras correcciones también pueden ser aplicadas al modelo de Okumura. Algunos de los parámetros importantes relacionados a los terrenos son la altura de ondulación del terreno, la altura de una cresta aislada, el promedio de las pendientes del terreno y los parámetros mezclados de arena-mar. Una vez que los parámetros relacionados son calculados, los factores de corrección necesarios son sumados o restados. Estos factores de corrección también son disponibles es las curvas de Okumura [4].

El modelo de Okumura esta basado totalmente sobre datos medidos y no provee ninguna explicación analítica. Para muchas situaciones, la extrapolación de las curvas derivadas puede ser hecha para obtener valores fuera de los rangos medidos, aunque la validación de tales extrapolaciones depende de las circunstancias y uniformidad de la curva en cuestión.

El modelo de Okumura es considerado entre los mejores y más simples modelos en términos de precisión en la predicción de perdidas por trayectoria para sistemas de radiodifusión móvil. Su mayor desventaja es su lenta respuesta a rápidos cambios en el terreno, por lo que el modelo es bastante bueno en áreas urbanas y suburbanas, pero no es bueno en áreas rurales.

4.3.4.4 Modelo de Hata

El modelo de Hata [5] es una formulación empírica de las gráficas de perdidas por trayectoria provistas por Okumura, y es valido de 150MHz a 1500MHz. Hata presenta las perdidas de propagación en áreas urbanas como una fórmula estándar y suple las ecuaciones de corrección por aplicaciones para otras situaciones. La fórmula estándar para las pérdidas por trayectoria media en áreas urbanas es dada por:

$$L_{50}(\text{urbano})(dB) = 69.55 + 26.16 \log f_c - 13.82 \log h_{te} - a(h_{re}) + (44.9 + 6.55 \log h_{te}) \log d$$

donde f_c es la frecuencia (en MHz) de 150MHz a 1500MHz, h_{te} es la altura efectiva (en metros) de la antena transmisora en el rango de 30m a 200m, h_{re} es la altura efectiva (en metros) de la antena receptora en el rango de 1m a 10m, d es la distancia de separación T-R (en km), y $a(h_{re})$ es el factor de corrección para la altura efectiva de la antena receptora la cual es una función del tamaño del área de cobertura. Para una ciudad pequeña o mediana, el factor de corrección de la antena receptora es dado por:

$$a(h_{re}) = (1.1 \log f_c - 0.7)h_{re} - (1.56 \log f_c - 0.8)dB,$$

para una gran ciudad es dado por:

$$a(h_{re}) = 8.29(\log 1.54h_{re})^2 - 1.1 \text{ dB} \quad \text{para } f_c \leq 300\text{MHz}$$

$$a(h_{re}) = 3.2(\log 11.75h_{re})^2 - 4.97 \text{ dB} \quad \text{para } f_c \geq 300\text{MHz}.$$

Para obtener las pérdidas por trayectoria en un área suburbana la fórmula estándar de Hata es modificada de la siguiente forma:

$$L_{50}(\text{dB}) = L_{50}(\text{urban}) - 2 \left[\log \left(\frac{f_c}{28} \right) \right]^2 - 5.4$$

y para las pérdidas por trayectoria en áreas rurales abiertas, la fórmula es modificada como:

$$L_{50}(\text{dB}) = L_{50}(\text{urban}) - 4.78(\log f_c)^2 - 18.33 \log f_c - 40.98.$$

Aunque el modelo de Hata no tiene ninguna corrección de trayectoria específica, las expresiones de arriba tienen valores prácticos significantes.

4.3.4.5 Extensión para el modelo de Hata

La Cooperativa Europea para la Investigación Científica y Tecnológica (EURO-COST por sus siglas en inglés) formó el comité de trabajo COST-231 para desarrollar una versión extendida del modelo de Hata. COST-231 propuso la siguiente fórmula para extender el modelo de Hata a 2GHz. [6]

$$L_{50}(\text{urbano}) = 46.3 + 33.9 \log f_c - 13.82 \log h_{te} - a(h_{re}) + (44.9 + 6.55 \log h_{te}) \log_{10} d + C_M$$

donde $a(h_{re})$ se definió anteriormente y

$$C_M = \begin{cases} 0\text{dB} & \text{para ciudades medianas y áreas suburbanas} \\ 3\text{dB} & \text{para metropolis} \end{cases}.$$

La extensión COST-231 del modelo de Hata es restringido por el siguiente rango de parámetros:

$$f : 1500\text{MHz} \text{ a } 2000\text{MHz}$$

$$h_{te} : 30\text{m} \text{ a } 200\text{m}$$

$$h_{re} : 1\text{m} \text{ a } 10\text{m}$$

$$d : 1\text{km} \text{ a } 20\text{km}$$

4.3.4.6 Modelo Walfisch y Bertoni

Un modelo desarrollado por Walfisch y Bertoni [7] considera el impacto de la altura de los tejados y edificios por medio de la difracción para predecir el promedio de intensidad de señal a nivel de las calles. El modelo considera las pérdidas por trayectoria, S , para ser un producto de tres factores.

$$S = P_0 Q^2 P_1,$$

donde P_0 representa las pérdidas por trayectoria en el espacio libre por antenas isotropicas

$$\text{dado por: } P_0 = \left(\frac{\lambda}{4\pi R} \right)^2.$$

El factor Q^2 da la reducción de la señal en los techos debido a las filas de edificios los cuales inmediatamente sombrean el receptor a nivel de las calles. El término P_l está basado sobre la difracción y determina la pérdida de la señal desde los techos hasta las calles.

4.3.5 Modelos de propagación interior

Los canales de radio interiores difieren de los tradicionales canales de radio móvil en dos aspectos – la distancia cubierta es mucho más pequeña y la variabilidad del ambiente es mucho más grande para un rango más pequeño de la distancia de separación T-R. Esto se ha observado ya que la propagación dentro de edificios es fuertemente influenciada por rasgos específicos tales como el diseño del edificio, los materiales de construcción y el tipo de construcción.

La propagación de radio interna es dominada por los mismos mecanismos que la exterior: reflexión, difracción y dispersión. Sin embargo las condiciones son mucho más variables. En general los canales interiores pueden ser clasificados de ambas formas de línea de vista (LOS por sus siglas en inglés) o obstruido (OBS por sus siglas en inglés), con grados variables de desorden.

El campo de la propagación radiofónica interior es relativamente nuevo, la primera oleada de desarrollos ocurrió en los inicios de los años 80. Cox de los laboratorios AT&T Bell y Alexander de British Telecom fueron los primeros en estudiar cuidadosamente las pérdidas de trayectoria internas alrededor de un gran número de casas y edificios de oficinas.

4.3.6 Trazado de rayo y modelado específico de sitio.

En los recientes años, las capacidades de cálculo y de visualización de las computadoras se han acelerado rápidamente. Por lo que nuevos métodos para la predicción de la cobertura de las señales de radio envuelven el uso de modelos de propagación de sitio específico (SISP por sus siglas en inglés) y bases de datos de sistemas de información gráfica (GIS por sus siglas en inglés) [8]. El modelo SISP soporta el trazado de rayo como un medio de modelado determinístico en cualquier medio de propagación interno o externo.

Para la predicción de propagación exterior, las técnicas de trazado de rayos son usadas en conjunción con las fotografías aéreas por lo que las representaciones en tres dimensiones de los edificios pueden ser integradas con el software que realiza los modelos de reflexión, difracción y dispersión. En los ambientes internos, los dibujos arquitectónicos proveen un sitio de representación específico para modelos de propagación.

4.4 Modelos de desvanecimiento de la señal por multitrayectorias o por pequeñas distancias (Multipath fading or small scale fading models)

El desvanecimiento de la señal por multitrayectorias o atenuación a pequeñas distancias es usado para describir las rápidas fluctuaciones de la amplitud de la señal de radio sobre un período corto de tiempo o una distancia corta recorrida. El desvanecimiento de la señal es causado por la interferencia entre dos o más versiones de una señal transmitida, la cual llega al receptor ligeramente a tiempos diferentes. Esas ondas, llamadas ondas de multitrayectoria, se combinan en la antena receptora para obtener una señal resultante la cual puede variar mucho en amplitud o fase, dependiendo de la distribución de la intensidad y el tiempo relativo de propagación de las ondas y del ancho de banda de la señal transmitida.

Las multitrayectorias en un canal de radio crean efectos de desvanecimiento a pequeñas distancias. Los efectos más importantes son:

- Cambios rápidos en la intensidad de la señal en distancias cortas recorridas o intervalos de tiempo.
- Modulación de frecuencias aleatorias debido a los diversos cambios Doppler sobre diferentes señales de multitrayectoria.
- Tiempos de dispersión (ecos) causados por los retrasos de propagación multitrayectoria.

4.4.1 Parámetros de los canales de multitrayectorias móviles.

Para entender las características de los canales de propagación inalámbricos es esencial conocer algunos parámetros pertinentes del canal de radio.

4.4.1.1 Propagación con retraso de tiempo y Ancho de banda coherente.

La propagación con retraso de tiempo, T_{delay} , es una medida de la longitud de la respuesta a impulso para canales inalámbricos de multitrayectorias. Usualmente la respuesta a impulso de un canal de multitrayectorias es caracterizado por su raíz cuadrada media (r.m.s por sus siglas en inglés) de la propagación con retraso de tiempo, $T_{delay\ rms}$, el cual es definido como la desviación estándar del perfil de retraso de la potencia del canal. Típicamente los valores rms de la propagación con retraso de tiempo son del orden de microsegundos para canales de radio exteriores y del orden de nanosegundos en canales de radio interiores.

Otra forma de definir la respuesta a impulso de un canal inalámbrico de multitrayectorias es por su máxima propagación de retraso de tiempo, $T_{delay\ max}$, el cual es determinado por el rango de retrasos sobre el cual los picos de retrasos del perfil de retraso de la potencia del canal no es menor a 30 dB desde el pico del primer pulso recibido.

Como una medida de la correlación de desvanecimiento de la señal entre frecuencias, el ancho de banda coherente es directamente relacionado con la propagación con retraso de tiempo. La propagación con retraso de tiempo rms y el ancho de banda coherente B_c son inversamente proporcionales el uno con el otro y es dado por:

$B_c = \frac{1}{2\pi T_{delay}}$, aunque su relación exacta es una función de la estructura de multitrayectorias exactas.

El ancho de banda coherente es una medida estadística del rango de frecuencias sobre el cual el canal puede ser considerado plano (por ejemplo, un canal, el cual pasa por todas las componentes espectrales aproximadamente con una ganancia igual y fase lineal). En otras palabras, el ancho de banda coherente es el rango de frecuencias sobre el cual dos componentes de frecuencia tienen un fuerte potencial para la correlación de amplitud.

4.4.1.2 Propagación Doppler y Tiempo coherente.

La propagación Doppler y el ancho de banda coherente son parámetros los cuales describen la naturaleza variante en el tiempo del canal en un región a corta distancia.

La propagación Doppler puede ser interpretada como una medida de la variación del cambio de la frecuencia portadora, de otra forma más intuitiva, como una medida del índice en el cual el canal cambia. Una pequeña propagación Doppler implica un tiempo de coherencia largo, T_c , o un cambio lento del canal.

El tiempo coherente es el dominio del tiempo doble de la propagación Doppler y es utilizado para caracterizar la naturaleza de la variación del tiempo de la dispersión de la frecuencia de un canal en el dominio del tiempo.

El tiempo de coherencia es actualmente una medida estadística del tiempo de duración sobre el cual la respuesta a impulso del canal es esencialmente invariante, y cuantifica la semejanza de la respuesta del canal a diferentes tiempos. En otras palabras, el tiempo de coherencia es el tiempo de duración sobre el cual dos señales recibidas tienen un potencial intenso por una correlación de amplitud.

4.4.2. Tipos de desvanecimiento a pequeñas escalas

Dependiendo de la relación entre los parámetros de la señal (tales como el ancho de banda, período de símbolo, etc.) y los parámetros del canal (tales como propagación de retraso rms y propagación Doppler), diferentes señales transmitidas se someten a diferentes tipos de desvanecimiento. Los mecanismos de tiempo y frecuencia de dispersión en un canal de radio móvil conducen a cuatro posibles efectos distintos, los cuales se manifiestan dependiendo de la naturaleza de la señal transmitida, el canal y la velocidad. Mientras la propagación con retraso por multitrayectorias conduce al tiempo de dispersión y la frecuencia de desvanecimiento selectiva, la propagación Doppler conduce a la dispersión de frecuencia y tiempo de desvanecimiento selectivo. Los dos

mecanismos de propagación son independientes el uno del otro. La figura 4.5 muestra un árbol de los cuatro tipos diferentes de desvanecimiento.

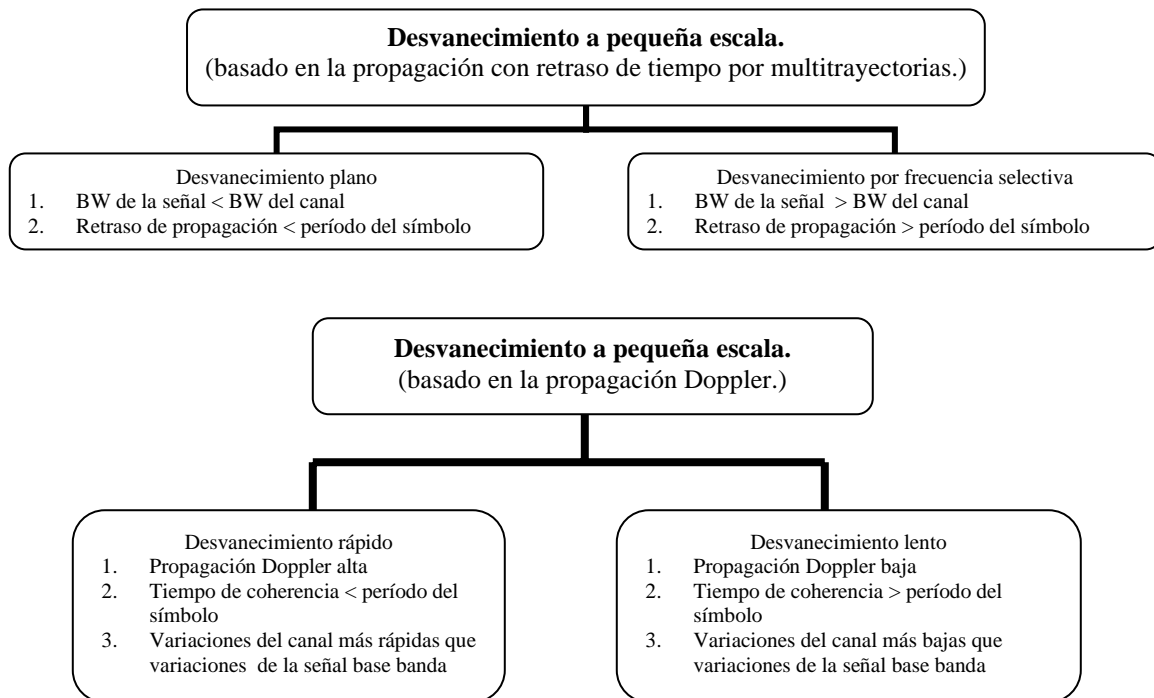


Figura 4.5 Tipos de desvanecimiento a pequeña escala.

4.4.3 Distribución Rayleigh y Ricean

4.4.3.1 Canal de desvanecimiento Rayleigh

En los canales de radio móvil, la distribución Rayleigh es comúnmente usada para describir la naturaleza variable del tiempo estadístico de la envolvente recibida de una señal desvanecida plana, o la envolvente de una componente de multitrayectoria individual. Es bien conocido que la envolvente es la suma de dos señales de ruido gaussiano de cuadratura que obedecen a una distribución de Rayleigh.

Si la señal recibida compuesta consiste de numerosas R_{path} , y cuando ese número de trayectorias de la señal es grande (en teoría $R_{path}=\infty$, pero en la practica $R_{path}>6$), de acuerdo con el teorema del Límite Central (CLT), la señal paso bajas compleja recibida es:

$$r(t) = r_I(t) + jr_Q(t)$$

puede ser modelada como procesos Gaussianos aleatorios complejos. Cuando no hay una línea de vista (NLOS por sus siglas es inglés) las variables aleatorias r_I y r_Q tendrían una media cero y la envolvente compleja recibida, $z(t)=|r(t)|$ tendría una distribución

Rayleigh en cualquier tiempo t , con la función de densidad de probabilidad (pdf por sus siglas en inglés):

$$p_R(r) = \begin{cases} \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} & (0 \leq r \leq \infty), \\ 0 & (r < 0) \end{cases},$$

la cual es graficada en el figura 4.6 (a). σ es el valor rms del voltaje de la señal recibida antes de la detección de envolvente, y σ^2 es la potencia del tiempo promedio de la señal recibida antes de la detección de envolvente. La probabilidad de que la envolvente de la señal recibida no exceda el valor específico R es dado por la correspondiente función de distribución acumulativa (CDF):

$$P_R(r) = P_R(r \leq R) = \int_0^R p_R(r) dr = 1 - e^{-\frac{R^2}{2\sigma^2}}, \quad r \geq 0$$

y esta es graficada en la figura 4.6 (b).

El valor medio r_{mean} de la distribución Rayleigh es dado por:

$$r_{mean} = E[r] = \int_0^{\infty} r p_R(r) dr = \sigma \sqrt{\frac{\pi}{2}} = 1.2533\sigma$$

y la varianza de la distribución de Rayleigh es dada por σ^2 , la cual representa la potencia ac en la señal envolvente:

$$\sigma_R^2 = E[r^2] - E^2[r] = \int_0^{\infty} r^2 p_R(r) dr - \frac{\sigma^2 \pi}{2} = \sigma^2 \left(2 - \frac{\pi}{2} \right) = 0.4292\sigma^2.$$

Como con el canal de desvanecimiento mayor el índice de errores de bits (BER por sus siglas en inglés) derivados de la estimación, se basan en los análisis de la evaluación de la integral:

$$p_a = \int_0^{\infty} p_{a_{AWGN}}(\gamma_b) P_R(\gamma_b) d\gamma_b,$$

donde $\gamma_b = \frac{E_b}{N_0}$, es el índice de señal-ruido, y $P_R(\gamma_b)$ es la función de densidad de probabilidad de γ_b cuando la amplitud de la señal es aleatoria.

Así de [9,10] puede ser resumido el BER para un canal Rayleigh por desvanecimiento lentamente como:

$$p_{a_{Rayleigh}} = \frac{1}{2} \left(1 - \sqrt{\frac{\frac{E_b}{N_0}}{1 + \frac{E_b}{N_0}}} \right).$$

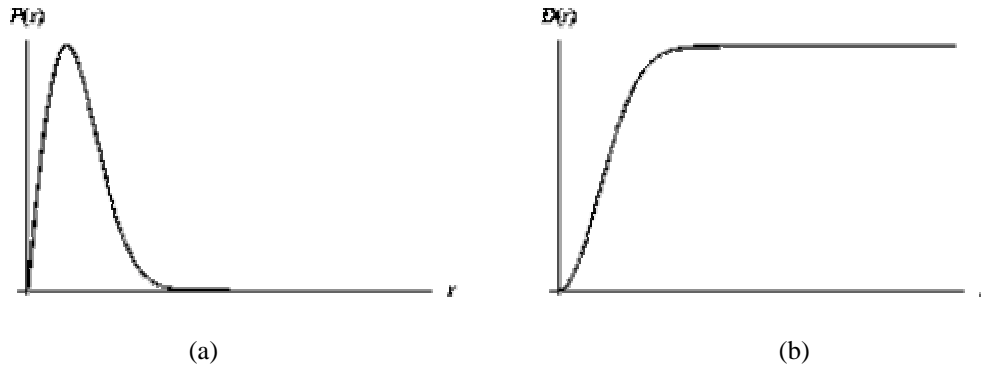


Figura 4.6 Distribución estadística de desvanecimiento Rayleigh

4.4.3.2 Canal de desvanecimiento Rician

Cuando hay una componente de una señal (no desvanecida) estacionaria dominante presente, tal como una trayectoria de propagación de línea directa, el desvanecimiento de pequeña escala sobre una distribución es Rician. En tal situación, las componentes de multitrayectoria aleatorias que llegan a diferentes ángulos son superpuestas sobre una señal dominante estacionaria. A la salida del detector de envolvente, esto tiene el efecto de añadidura de una componente dc a la trayectoria aleatoria.

Justo como para el caso de detección de una onda senoidal de ruido termal, el efecto de una señal dominante que llega con muchas señales de multitrayectoria muy débiles revela un aumento a la distribución Rician. Como la señal dominante llega más débil, la señal compuesta se asemeja a una señal de ruido, la cual tiene una envolvente que es Rayleigh. Así la distribución Rician degenera a una distribución Rayleigh cuando la componente dominante se desvanece.

La distribución Rician es dada por:

$$p(r) = \begin{cases} \frac{r}{\sigma^2} e^{-\frac{(r^2+A^2)}{2\sigma^2}} I_0\left(\frac{Ar}{\sigma^2}\right) & \text{para } (A \geq 0, r \geq 0) \\ 0 & \text{para } (r < 0) \end{cases}$$

El parámetro A denota la amplitud pico de la señal dominante e $I_0(\cdot)$ es la función de Bessel modificada de primera clase y orden cero. La distribución de Rician es frecuentemente descrita en términos de un parámetro K el cual es definido como la proporción entre la potencia de la señal determinística y la varianza de la multitrayectoria. Esta es dada por $K = A^2 / 2\sigma^2$ o en términos de dB

$$K(dB) = 10 \log \frac{A^2}{2\sigma^2} dB .$$

El parámetro K es conocido como el factor de Rician y completamente especifica la distribución Rician. Como $A \rightarrow 0$, $K \rightarrow -\infty$ dB, y como la trayectoria dominante decrece en amplitud, la distribución Rician degenera a una distribución Rayleigh. Cuando el factor K es ∞ , la distribución resultante es una Gaussiana, esto es un canal de ruido gaussiano blanco aditivo (AWGN por sus siglas en inglés).

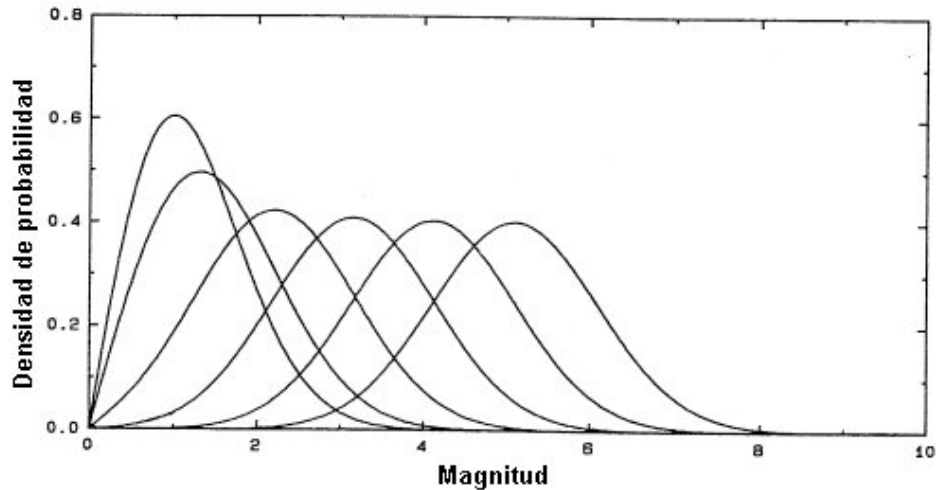


Figura 4.7 Distribución estadística de desvanecimiento Ricean

4.4.4 Modelos Estadísticos para canales de atenuación por multi-trayectorias

Diversos modelos multitrayectorias han sido sugeridos para explicar la naturaleza estadística de un canal móvil. Uno de ellos es el modelo de Clarke [11] el cual está basado sobre la dispersión y es ampliamente usado. Clarke desarrollo un modelo donde las características estadísticas de los campos electromagnéticos de la señal recibida en el móvil son deducidas de la dispersión.

Rice cálculo estadísticas conjuntas para un problema matemático el cual es similar al modelo de desvanecimiento de Clarke, y por eso provee extensiones simples para operar el número promedio de niveles de intersección y la duración de los desvanecimientos. El índice de nivel de intersección (LCR por sus siglas en inglés) y el promedio de la duración del desvanecimiento de una señal desvanecida Rayleigh son dos estadísticas importantes las cuales son importantes para el desarrollo de códigos de control de errores y diversos esquemas que pueden ser usados en sistemas de comunicación móvil.

Como el modelo de Clarke y las estadísticas para desvanecimiento Rayleigh son para condiciones de desvanecimiento planas y no consideran el tiempo de retraso por multitrayectorias. En los sistemas modernos de comunicaciones móviles con altos índices de datos, es necesario modelar los efectos de la propagación con retraso por multitrayectorias tan bien como el desvanecimiento. Un modelo de multitrayectorias comúnmente usado es un modelo de dos rayos por desvanecimiento de Rayleigh independiente.

Otros modelos de multitrayectoria para canales interiores fueron desarrollados por Saleh y Valenzuela [12] y están basados en resultados de mediciones. Este modelo asume que las componentes de multitrayectoria llegan en grupos. La amplitud de las componentes recibida es independiente de las variables aleatorias Rayleigh con

variaciones que decaen exponencialmente con los grupos de retraso tan bien como el exceso de retraso dentro de un grupo.

4.5 Canales de ruido gaussiano blanco aditivo (AWGN)

Consideremos un canal de comunicaciones, el cual existe entre el transmisor y el receptor. Inicialmente se consideran solo enlaces de un canal de comunicación fijo por eso ambas terminales son estacionarias, aunque los canales de comunicación móvil cambian significativamente con el tiempo.

Los canales de comunicación fija frecuentemente pueden ser modelados por una función de transferencia lineal, la cual describe la distorsión del canal. El ruido gaussiano blanco aditivo ubicuo (AWGN por sus siglas en inglés) es un factor limitante fundamental en las comunicaciones por canales lineales de tiempo invariante (LTI por sus siglas en inglés). Aunque las características del canal pueden cambiar debido a factores como cambios de temperatura y cambio de canal, esas variaciones no serán obvias más allá del curso de una sesión de comunicación típica.

Un canal ideal de comunicaciones libre de distorsión tendría una respuesta en frecuencia plana y una respuesta en fase lineal sobre el rango de frecuencias de $-\infty$ a $+\infty$. Pero en los canales prácticos siempre tienen algunas distorsiones lineales debido a su banda limitada, su respuesta en frecuencia no plana y respuesta de fase no lineal. En adición, la respuesta del grupo de retraso del canal es frecuentemente dada, la cual es derivada de la respuesta de fase.

A pesar del canal de comunicación usado, el ruido aleatorio está siempre presente. El ruido puede ser ampliamente clasificado como natural o hecho por el hombre. Ejemplos de ruido producido por el hombre son aquellos debido a aplicaciones eléctricas y luz fluorescente y los efectos de esas fuentes pueden ser mitigadas dentro de la misma fuente. Las fuentes naturales de ruido afectan las transmisiones de radio e incluye las radiaciones estelares y el ruido atmosférico. Existen ventanas de frecuencia de bajo ruido en el rango de 1-10GHz, donde los efectos de esas fuentes pueden ser minimizadas.

El ruido termal natural es ubicuo. Este es debido al movimiento de los electrones, y este se puede reducir por la disminución de la temperatura. Puesto que el ruido termal contiene prácticamente todas las componentes de frecuencia arriba de los 10^3 HZ con igual potencia. Esta es frecuentemente referida como ruido blanco (WN) en una analogía con la luz blanca contenida en todos los colores con igual intensidad. Este proceso WN puede ser caracterizado por su densidad espectral de potencia uniforme (PSD por sus siglas en inglés) $N(\omega) = N_0/2$ mostrada junto con su función de autocorrelación (ACF por sus siglas en inglés) en la figura 4.8.

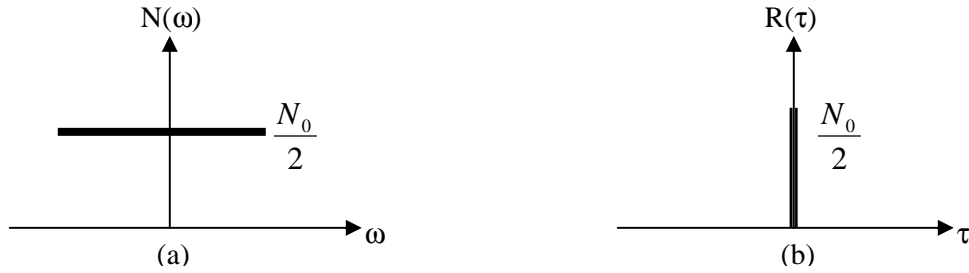


Figura 4.8 Densidad espectral de potencia y autocorrelacion de WN

En un sistema de comunicaciones de banda limitada se limita no solo la señal sino también el ruido, y este filtrado limita la proporción de cambio de la señal de ruido en el dominio del tiempo, introduciendo algo de correlación sobre el intervalo de $\pm 1/2B$. Esta PSD estilizada y ACF de ruido blanco de banda limitada es mostrado en la figura 4.9.

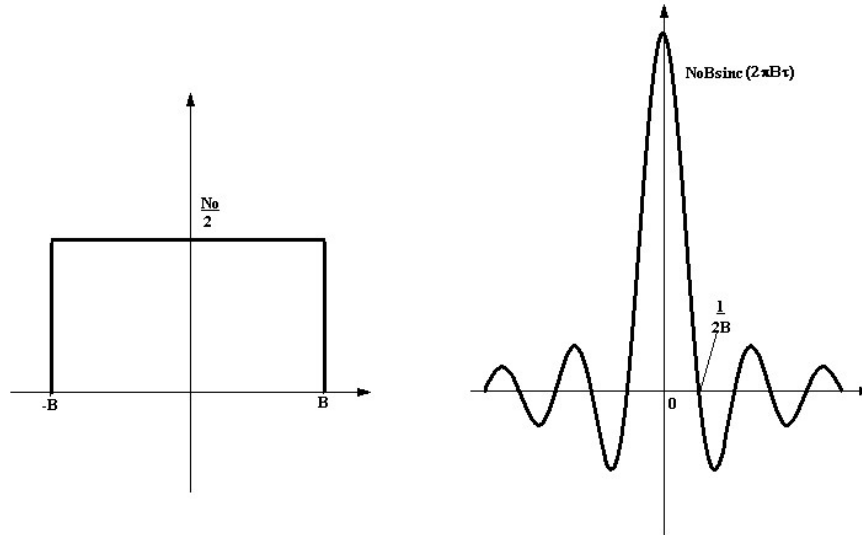


Figura 4.9 Densidad espectral de potencia y autocorrelacion de banda limitada de WN

En el dominio del tiempo, la distribución de tiempo del ruido termal blanco tiene una distribución normal o gaussiana, y desde que esta es inevitablemente añadida a la señal recibida, esta es usualmente referida como ruido gaussiano blando aditivo (AWGN). Note que el AWGN es por lo tanto el ruido generado en el receptor. La función de densidad de probabilidad es bien conocida como la curva con forma de campana de la distribución gaussiana, dada por:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-/(x-m)/2\sigma^2},$$

donde m es la media y σ^2 la variancia. Los efectos del AGWN pueden ser mitigados por el incremento de la potencia de la señal transmitida y por esto reducir los efectos relativos el ruido. La proporción de señal ruido (SNR por sus siglas en inglés) en la entrada del receptor proveen una buena medida de la calidad de la señal recibida.

4.6 Canales simétricos binarios (BSC)

Un canal simétrico binario (BSC por sus siglas en inglés) se propone dando una idealización matemática del problema de transmitir datos en un medio ruidoso dentro de las referencias clásicas de Shannon, donde se asume que ambos la fuente y el canal son discretos.

Un canal simétrico binario satisface las siguientes condiciones:

1. Cada vez que se transmite un bit, un bit llega el receptor; pero puede llegar corrompido.
2. Existe la probabilidad p de que un 0 transmitido sea recibido como 1, y la misma probabilidad p que un 1 transmitido sea recibido como 0.
3. Los errores ocurren independientemente, en el sentido de independencia estadística.

La palabra binario en el nombre indica que solamente dos símbolos (0 y 1) se utilizan. La palabra simétrico se refiere al hecho de que la probabilidad de que un 0 transmitido sea recibido como un 1 es igual que la probabilidad de que un 1 transmitido sea recibido como 0. La suposición de independencia estadística significa que la probabilidad de dos bits sucesivos sean corrompidos es p^2 , la probabilidad de tres bits sucesivos sean corrompidos es p^3 , etcétera.

La función de la entropía para un canal simétrico binario con probabilidad de error p por bit transmitido se define como:

$$H(p) = -p \log_2(p) - (1-p) \log_2(1-p)$$

donde $H(p)$ es definida solo para $0 < p < 1$.

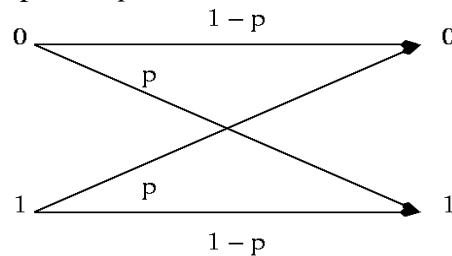


Figura 4.10 Canal simétrico binario (BSC)

4.7 Modelo de canales de Markov de estado-finito

Un canal de Markov de estado finito (FSMC por sus siglas en inglés) es un canal variante en el tiempo discreto cuya variación es determinada por un proceso de Markov de estado finito.

El estudio de los canales de Markov de estado finito emerge de un trabajo temprano de Gilbert [13] y Elliott [14]. Ellos estudiaron un canal de Markov de dos estados conocido como el canal de Elliott-Gilbert. En su modelo de canal cada estado corresponde a una calidad de canal específico el cual puede ser ambos sin ruido o totalmente ruidoso. En general, un canal simétrico binario (BSC) con una probabilidad de

paso dada puede ser asociado con cada estado por lo que la calidad del canal para cada estado puede ser identificada. En algunos casos, modelar un canal de comunicaciones de radio como un canal Elliott-Gilbert de dos estados no es adecuado cuando la calidad del canal varía dramáticamente. Una solución simple es formar un modelo de canal con más de dos estados.

El conjunto $S = \{s_0, s_1, \dots, s_{K-1}\}$ denota un conjunto finito de estados y $\{S_n\}, n = 0, 1, 2, \dots$, es un proceso de Markov constante. Ya que el proceso de Markov constante tiene la propiedad de transiciones estacionarias, la probabilidad de transición es independiente de la señal de tiempo n y puede ser escrita como:

$$t_{j,k} = P_r(S_{n+1} = s_k | S_n = s_j),$$

para todas $n=0, 1, 2, \dots$ y $k, j \in \{0, 1, 2, \dots, K-1\}$. Con esta definición, nosotros podemos definir una matriz de probabilidad de transición de estado \mathbf{T} de $K \times K$ con sus elementos $t_{j,k}$. Note que una matriz de probabilidad de transición de estado tiene la propiedad de que la suma de los elementos de cada fila es igual a 1.

Por otra parte, con la propiedad de transición estacionaria, la probabilidad de estado k en cualquier señal de tiempo permisible n sin cualquier estado de información a otros señales de tiempo puede ser también definida como:

$$p_k = P_r(S_n = s_k), k \in \{0, 1, 2, \dots, K-1\}.$$

Un vector de probabilidad de estado uniforme \mathbf{p} de $K \times 1$ puede ser definido con sus elementos p_k . En muchos casos este vector puede servir como el conjunto de probabilidad de estado inicial.

$$\mathbf{p}^t \mathbf{T} = \mathbf{p}^t.$$

Una completa descripción de un canal de Markov de estado finito requiere de información adicional sobre la calidad del canal para cada estado. Define un vector de probabilidad de paso \mathbf{e} de $K \times 1$ con sus elementos $e_k, k \in \{0, 1, 2, \dots, K-1\}$, siendo la probabilidad del canal simétrico binario asociado con el estado k . Un FSMC es entonces definido únicamente por \mathbf{T} , \mathbf{p} y \mathbf{e} . La probabilidad de error promedio global \mathbf{e} del FSMC es entonces:

$$\mathbf{e} = \mathbf{p}^t \mathbf{e} = \sum_{k=0}^{K-1} p_k e_k.$$

La elección de \mathbf{T} , \mathbf{p} y \mathbf{e} podría no ser arbitraria. Pero hay cuatro limitaciones obvias para \mathbf{p} y \mathbf{e} :

1. $0 < p_k \leq 1, \forall k \in \{0, 1, 2, \dots, K-1\}$
2. $\sum_{k=0}^{K-1} p_k = 1$
3. $0 \leq e_k \leq 0.5, \forall k \in \{0, 1, 2, \dots, K-1\}$
4. $e_i \neq e_j, \text{ si } i \neq j \forall i, j \in \{0, 1, 2, \dots, K-1\}$

La mayoría de los estudios sobre el funcionamiento de los canales Elliott-Gilbert están basados sobre las variaciones de \mathbf{T} , \mathbf{p} y \mathbf{e} bajo las limitaciones mencionadas arriba.

La capacidad de un canal de Markov de estado finito puede ser tratada desde dos puntos de vista: la capacidad del canal y la distorsión. Cuando la información de estado del canal (CSI por sus siglas en inglés) esta disponible, la capacidad C^{CSI} es simplemente la capacidad promedio sobre todos los estados.

$$C^{CSI} = \sum_{k=0}^{K-1} p_k [1 - h(e_k)]$$

donde $h(\cdot)$ es la función de entropía binaria definida como:

$$h(e) = e \log \frac{1}{e} + (1 - e) \log \frac{1}{1 - e}.$$

Desde otro punto de vista es completamente difícil calcular la capacidad de un canal de estado finito de Markov si la CSI no esta disponible. En [13] Gilbert calcula la capacidad de un canal Elliott-Gilbert, el cual es un canal de Markov de dos estados con un estado bueno y un estado malo. Los resultados muestran la probabilidad de paso de esos dos estados igual a 0 y 0.5 respectivamente.

Para un canal FSMC las condiciones óptimas necesarias son derivadas en [15] con la suposición que el receptor tiene el conocimiento de la información de estado del canal actual. Este [15] muestra que cuando la información de estado del canal es disponible para el receptor la distorsión de error cuadrático medio es dada por:

$$D = \sum_{i=1}^N \sum_{j=1}^N \sum_{k=0}^{K-1} p(m(j)|m(i), k) p_k \times \int_{V(i)} p_X(x) (x - y_{j,k})^2 dx$$

donde el conjunto $p_k, k=0,1,\dots,K-1$ es la probabilidad de estado uniforme para que el canal de estado K este en el estado k y $p(m(j)|m(i), k)$ es la probabilidad condicional de que la señal $m(j)$ es recibida dado que la señal $m(i)$ fue transmitida y el canal esta en un estado k . Y $N=2^R$ y $V(i)$ es la región de cuantización para la señal $m(i)$.

Además en [15] también se muestran las condiciones óptimas necesarias, también conocidas como la condición vecina más cercana y la condición de centroide dadas por:

$$V(i) = \left\{ x : \sum_{j=1}^N \sum_{k=0}^{K-1} p(m(j)|m(i), k) p_k (x - y_{j,k})^2 \leq \sum_{j=1}^N \sum_{k=0}^{K-1} p(m(j)|m(l), k) p_k (x - y_{j,k})^2; \forall l \neq i \right\}$$

y

$$y_{j,k} = \frac{\sum_{i=1}^N p(m(j)|m(i), k) p_k \times \int_{V(i)} x p_X(x) dx}{\sum_{i=1}^N p(m(j)|m(i), k) p_k \times \int_{V(i)} p_X(x) dx}.$$

Estas dos ecuaciones formulan las condiciones necesarias para un cuantizador óptimo cuando la información del canal es disponible solo para el decodificador. En ausencia de la información del canal, el canal es simplemente equivalente a un canal simétrico binario. Por otra parte si la información de canal es disponible para ambos el codificador

y el decodificador, entonces el cuantizador óptimo para un canal de estado K es equivalente a, por ejemplo, canales simétricos binarios K .

Ahora dando una probabilidad de error promedio fija e , para un FSMC, un esfuerzo es hecho para encontrar el modelo que optimice el funcionamiento del sistema en el sentido de maximizar la capacidad del canal y minimizar distorsión en la codificación. Por lo cual se derivan los dos teoremas siguientes [16]:

Teorema 1. Supóngase la probabilidad de error promedio e para un canal de Markov de estado K es fijo. Suponga C sea la capacidad del sistema definida como anteriormente se describió. Los resultados de maximizar C con respecto a \mathbf{p} , \mathbf{e} y el número de estados K sujeto a la probabilidad de error promedio global y a las limitaciones de ésta es un canal de Markov de dos estados con:

$$e_0 = 0.5, \quad p_0 = 2e$$

$$e_1 = 0, \quad p_1 = 1 - 2e$$

Teorema 2. Supóngase la probabilidad de error promedio e para un canal de Markov de estado K es fijo. Suponga D sea la distorsión promedio definida como anteriormente se describió. Los resultados de minimizar D con respecto a \mathbf{p} , \mathbf{e} y el número de estados K sujeto a la probabilidad de error promedio global y a las limitaciones de ésta es un canal de Markov de dos estados con:

$$e_0 = 0.5, \quad p_0 = 2e$$

$$e_1 = 0, \quad p_1 = 1 - 2e$$

Dados estos teoremas se puede declarar lo siguiente: Dado un canal de Markov de estado K , se puede optimizar el modelo de canal, en el sentido de incrementar la capacidad del canal o decrementar la distorsión en la codificación, por división de cada uno de esos estados satisfaciendo $0 < e_k < 0.5$ dentro de dos subestados con uno sin ruido y otro enteramente con ruido. Entonces el resultado del canal de Markov de estado finito contendría solo dos tipos de canales simétricos binarios, donde la probabilidad de paso es igual a 0 y 0.5 respectivamente. Note que el FSMC es optimizado con respecto a \mathbf{p} , \mathbf{e} , y K excluyendo la matriz de transición \mathbf{T} .

En [16] se describe un canal de comunicaciones de radio típico llamado canal de desvanecimiento Rayleigh, por medio de un canal de Markov de estado finito, por la división del rango del SNR recibido dentro de un número finito de intervalos. Otro estudio similar al anterior se presenta en [17].

Referencias

- [1] P. L. Rice, A. G. Longley, K. A. Norton y A. D. Barsis, "Transmission Loss Predictions For Tropospheric Communication Circuits" NBS Tech Note 101; 2 volumes, issued May 1, 1966; revised January 1967.

-
- A. G. Longley, P. L. Rice; "Prediction of Tropospheric Radio Transmission Loss over Irregular Terrain, a Computer Method"; ESSA Technical Report; ERL 79 - HS 67, 1968.
- [2] R. Edwards, J. Durkin; "Computer Prediction of Service Area for VHF Mobile Radio Networks" Proceedings of the IEE, Vol. 116, No. 9 pp 1493-1500, 1969.
- [3] C. E. Dadson, J. Durkin, E. Martin; "Computer Prediction of Field Strength in the Planning of Radio Systems"; IEEE Transaction on Vehicular Technology, Vol. VT-24, No. 1, pp 1-7, February 1975.
- [4] T. Okumara, E. Ohmort, K. Fukuda; "Field Strength and its Variability in VHF and UHF Land Movil Service" Review Electrical Communication Laboratory, Vol. 16, No. 9-10, pp. 825-873, September-October 1968.
- [5] Hata, Masaharu, "Empirical Formula for Propagation Loss in Land Mobile Radio Service"; IEEE transaction on vehicular technology, Vol. VT-29 Mo. 3 pp. 317-325, August 1986.
- [6] European Cooperation in the Field of Scientific and Technical Research EURO-COST 231, "Urban Transmission Loss Models for Mobile Radio in the 900 and 1800 MHz Bands" Revision 2, the Hague, September 1991.
- [7] J. Walfisch, H. L. "A Theoretical Model of UHI Propagation in Urban Environments"; IEEE Transaction on Antennas and Propagation; Vol. AP-36, pp. 1788-1796, October 1988.
- [8] T. A. Russel, C. W. Bostian, T. S. Rappaport; "A deterministic, Approach to Predicting Microwave Diffraction by Buildings for Microcellular Systems"; IEEE transactions on antennas and propagation, Vol. 41, No. 12, pp 1640-1649, December 1993.
- [9] J. G. Proakis, "Digital Communication", McGraw Hill, New York, 3rd edition, 1995.
- [10] T. S. Rappaport, "Wireless Communication: Principles and Practice" IEEE Press, Piscataway, NJ, 1996.
- [11] R.H. Clarke, "A statistical theory of mobile-radio Reception", Bell Systems Tech. J. 47, 957-1000 (1968).
- [12] A.M. Saleh, and R.A. Valenzuela, "A statistical model for indoor multipath propagation", IEEE J. Selected Areas Comm., SAC-5, 128-137, (1987).
- [13] E. N. Gilbert, "Capacity of a burst-noise channel", Bell Syst. Tech. J., vol. 39, pp. 1253-1265, Sept. 1960.
- [14] E. O. Elliott, "Estimates or error rates for codes on Bursa-noise channels", Bell Syst. Tech. J., vol. 42, pp 1977-1997, Sept. 1963.
- [15] H. S.Wang, N. Moayeri, "Modeling, capacity, and joint source/channel coding for Rayleigh fading channel", in Proc. 43rd IEEE Vehicular Technology Conf., May 1993.
- [16] H. S. Wang, N. Moayeri, "Finite-State Markov Channel –A useful model for radio communication channels", IEEE Transactions on Vehicular Technology, Vol. 44, No. 1, February 1995.
- [17] Q. Zhang, S. A. Kassam, "Finite-State Markov Model for Rayleigh Fading Channels", IEEE Transactions on Communications, Vol. 47, No. 11, November 1999.
-

Capítulo 5

Esquema de codificación conjunta de fuente canal.

Shannon dijo que la codificación de una fuente de amplitud continua se puede describir por su función de índice de distorsión, que da la tasa mínima necesitada para codificar la fuente con cierta distorsión; y un canal puede ser descrito por su capacidad, que es la tasa más grande en el cual el canal puede transferir información confiablemente. Una señal de fuente se puede transmitir en un canal con cierta distorsión si la capacidad de canal está arbitrariamente cerca de la función de índice de distorsión de la fuente en una distorsión considerada. Además, esto se puede obtener con un código de fuente que cifre la información de fuente a una tasa suficientemente cerca de la función de índice de distorsión seguida por un código de canal que dé una probabilidad suficientemente pequeña de error en una tasa suficientemente cerca de la capacidad de canal. El resultado posterior a menudo se llama teorema de separación de canal-fuente.

Existen canales para los cuales no se sostiene este teorema, porque la capacidad de cierta manera depende del método de codificación. Un ejemplo de tal canal es un canal de acceso múltiple, donde la capacidad depende de la correlación entre las señales de entrada (Cover y Thomas, 1991), otro ejemplo es un canal de desvanecimiento de bloque, donde el canal es descrito en términos de la longitud de los codewords del canal.

Otro problema con el teorema de separación es que los códigos deben tener longitudes ilimitadas para encontrarse arbitrariamente cerca de los límites. Esto significa que para cierto retraso en la codificación, la separación de la codificación de la fuente y la codificación del canal no puede ser óptima. Quizás el problema más importante es que aunque la codificación separada de fuente y canal puede ser óptima para cierto canal, la separación es altamente débil (no robusta). Si los parámetros del canal cambian de modo que la capacidad sea más baja que la esperada, el código del canal no puede dar una baja probabilidad de error, y el código de fuente entonces no puede dar una baja distorsión si los códigos se diseñan para estar muy cerca de los límites. Además, la separación significa que si la capacidad es más alta que la esperada, la distorsión no puede bajar.

Los problemas mencionados arriba son razones para mirar hacia la codificación conjunta del canal con la fuente, a pesar del teorema de separación. El término conjunto (o combinado) de la codificación del canal con la fuente será utilizado en todos los métodos donde por lo menos los bloques de codificación de fuente y de codificación de canal estén unidos en un bloque común o se diseñen conjuntamente de una cierta manera.

Un código de canal fuente conjunto puede incluir más bloques que la codificación de fuente y la codificación del canal. Si el código se diseña para una fuente con memoria, significa que los bloques de descomposición y reconstrucción son incluidos. Si se diseña

para utilizar la naturaleza de amplitud continua de la fuente, por ejemplo usando la información suave en el decodificador, los bloques de correspondencia de símbolo y de detección son parte del codificador de canal fuente conjunto y del decodificador, respectivamente.

De las posibles deficiencias mencionadas de la separación del codificador, puede ser deducido que el uso de un código de canal fuente conjunto puede ser justificado:

- si los códigos de la fuente y del canal separados no conocidos pueden dar el funcionamiento del código de canal fuente conjunto,
- si el código de canal fuente conjunto puede conseguir el funcionamiento deseado con una complejidad más baja que los códigos separados,
- si debajo de los límites de retraso, el código conjunto de canal fuente funciona mejor que el código separado,
- si el código conjunto de canal fuente da buenos resultados para un amplio rango de calidades del canal, siendo así más robusto contra las calidades desconocidas del canal.

El uso de la codificación conjunta de fuente canal tiene algunas desventajas. Los sistemas de comunicación modernos son a menudo absolutamente complejos, donde los datos se pueden transmitir a través de una cadena de canales de comunicación que tienen características muy diferentes. La separación de la codificación de fuente y del canal significa que los usuarios finales solo necesitan enfocarse en la codificación de fuente y no tienen que saber mucho sobre la naturaleza de los segmentos del canal, mientras que los nodos del sistema de comunicación necesitan solamente centrarse en la codificación apropiada del canal para los segmentos del canal que ellos sirven, y no en la fuente que se está transmitiendo. Si la codificación conjunta de fuente canal va a ser aplicada a tal sistema de comunicación, la fuente puede tener que ser recodificada en los nodos para concordar con diferentes clases de canales. Esto significa que los nodos deben saber sobre la codificación de fuente para diferentes tipos de fuentes. Así, la codificación conjunta de fuente canal se adapta lo mejor posible para las situaciones donde el canal puede ser conocido por los usuarios finales.

5.1 Codificación conjunta de fuente canal en imágenes

Debido a las aplicaciones que los sistemas modernos de comunicación ofrecen, la compresión de una imagen es un tema importante, por lo cual se necesitan métodos que proporcionen la suficiente protección de error para aplicaciones como almacenamiento y transmisión de una imagen. Por lo que, como se menciono anteriormente, la codificación conjunta de fuente canal puede dar ventajas tales como robustez contra calidades desconocidas del canal y baja complejidad de cómputo para obtener la misma distorsión, características muy importantes en la transmisión de una imagen en medios inalámbricos.

Uno de los primeros trabajos de codificación conjunta de fuente canal en imágenes fue hecho por Modestino y Daut (1979), donde se negociaron los índices del codificador de fuente y de canal utilizándose una protección de error desigual para los

bits de codificación por la modulación de código por pulso diferencial (DPCM por sus siglas en inglés). Éste fue ampliado más adelante por una codificación de fuente por transformada (Modestino, Daut y Vickers, 1981).

Trabajos más recientes sobre la protección de error desigual en la codificación conjunta de fuente canal de una imagen han sido hechos por Fazel y Lhullier (1990), Tanabe y Farvardin (1992), quienes consideraron una codificación de fuente de longitud variable, y por Chande y Farvardin (2000), quienes combinaron la protección de error desigual con una transmisión progresiva de imágenes. Ruf y Modestino (1999) utilizaron técnicas operacionales de la función de distorsión para negociar los índices de codificación de la fuente y del canal en un esquema de protección de error desigual y encontraron los límites del funcionamiento teórico de información, y Cai y Chen (2000) también incluyeron un filtro all-pass para mejorar el funcionamiento. Kozintsev y Ramchandran (1998) y Zheng y Liu (1999) combinaron la protección de error desigual con una modulación de multiresolución, donde el sistema de modulación es no uniforme, permitiendo diferentes probabilidades de error para diferentes bits. La combinación de codificación de fuente y de modulación de multiresolución en estos documentos da una correspondencia de canal fuente directa.

Otra aproximación de codificación conjunta de fuente canal de una imagen es el uso de una cuantización robusta, como fue hecho por Vaishampayan y Farvardin (1990), que utilizaron la asignación de bit y cuantizadores óptimos para un canal con errores. Skoglund (1995) utilizó una estructura basada en Hadamard con una decodificación suave. Chen y Fischer (1998) utilizaron una cuantización robusta junto con un distorsionador de frecuencia para reducir el impacto perceptivo del ruido. El codificador propuesto por Lervik y Ramstad (1996) tiene un cuantizador que se diseñó conjuntamente con un sistema de modulación no optimizado. Este sistema fue mejorado en (Lervik, 1996).

5.1.1 Estructura general de un sistema de codificación conjunta fuente canal en imágenes

La estructura de un sistema de codificación conjunta de fuente canal es mostrado en la figura 5.1, donde una imagen de entrada es decorrelacionada por el uso de una codificación por transformada teniendo unidades de entrada de datos representadas dentro de una señal i por un cuantizador de m bits. Entonces códigos de canal diferentes son aplicados a diferentes bits de cada señal de acuerdo a su respectiva importancia en los datos reconstruidos. Después de que los datos codificados pasan a través de un canal ruidoso. El receptor es simplemente un proceso inverso. Los bits disponibles pueden ser apropiadamente asignados entre el cuantizador y los codificadores de canal para equilibrar las distorsiones causadas por el ruido de cuantización y el ruido del canal por lo que la distorsión de un punto al otro puede ser minimizada.

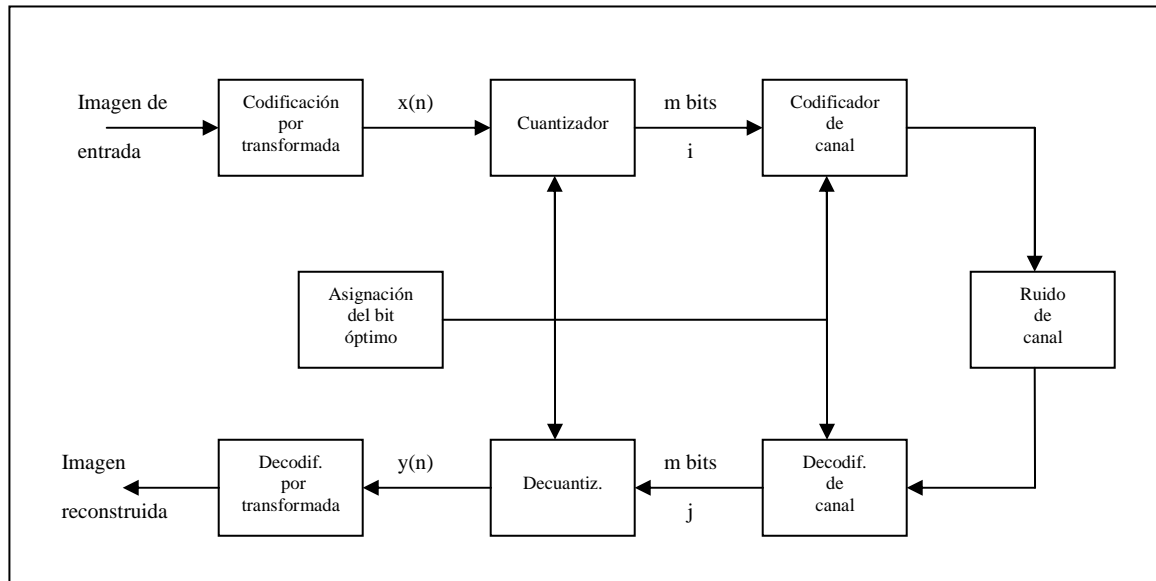


Figura 5.1 Estructura de un sistema de codificación de canal fuente conjunta.

5.2 Codificación canal-fuente conjunta

5.2.1 Codificación de transformada

Como en otros esquemas de codificación, la imagen primero es sometida a una transformación antes de la cuantización, para decorrelacionar la señal fuente y tener los datos más disponibles para la compresión. Esto podría ser por medio de una Transformada Coseno Discreta (DCT por sus siglas en inglés) o una Transformada por subbanda (SBT por sus siglas en inglés). Como las representaciones multiresolución, o Transformadas Wavelet Discretas (DWT por sus siglas en inglés) han sido aplicadas muy exitosamente en la codificación de imágenes y han demostrado un funcionamiento superior subjetivamente y objetivamente en ambos grados de compresión altos y bajos, generalmente en los sistemas de codificación conjunta de fuente canal se utiliza este tipo de transformada. La idea básica de este tipo de transformadas SBT es analizar la imagen original en todas sus bandas dentro de un número de imágenes de banda angosta que son codificadas y transmitidas separadamente; las subbandas decodificadas serán entonces usadas para sintetizar una replica de la imagen original.

5.2.2 Propiedades estadísticas de las subbandas

Para modelar las subbandas dentro de un sistema de codificación es importante conocer sus propiedades estadísticas como el valor de la media, variancia y los coeficientes de correlación de cada una de las subbandas de la imagen original. Además también es importante conocer la forma de la distribución de las subbandas para el diseño de los cuantizadores de las diferentes subbandas.

Se ha encontrado en [1] que la distribución de las subbandas puede ser comparada con una Distribución Gaussiana Generalizada (GGD por sus siglas en inglés). La función

de densidad de probabilidad (pdf por sus siglas en inglés) asociada con la GGD es dada por:

$$p(x) = \left[\frac{\alpha \eta(\alpha, \beta)}{2\Gamma(1/\alpha)} \right] \exp\left\{ -[\eta(\alpha, \beta)|x|]^\alpha \right\}, \quad 5.1$$

$$\text{con } \eta(\alpha, \beta) = \beta^{-1} \left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)} \right]^{\frac{1}{2}}, \quad 5.2$$

donde $\alpha > 0$ es el parámetro de forma que describe el índice de decaimiento exponencial, y β es la desviación estándar de la distribución y $\Gamma(\cdot)$ es la función gama. La GGD con $\alpha = 2.0$ coincide con una distribución Gaussiana; mientras que para $\alpha = 1.0$ esta llega a ser una distribución Laplaciana. La GGD con α en el rango de $0.1 < \alpha < 1.0$ provee un modelo útil para densidades broad-tailed. Note que para valores grandes de α la distribución tiende a ser una distribución uniforme.

5.2.3 Cuantización y estrategias de codificación

La cuantización significa proveer una aproximación a las señales y parámetros de la señal por medio de un número finito de niveles de representación. Este proceso es no reversible por lo cual siempre introduce ruido. Los niveles de representación constituyen un alfabeto finito el cual es usualmente representado por símbolos binarios, o bits. La correspondencia de símbolos en un alfabeto finito a bits es no única.

Es así como un cuantizador es un dispositivo que opera sobre una señal para producir un número finito de niveles de amplitud o niveles de cuantización. El número de niveles N en la mayoría de los cuantizadores es invariablemente una potencia de 2. Si $N = 2^R$ cada nivel N es codificado a un número binario, y cada valor de la señal es representado en forma binaria como una palabra de R bits correspondiendo a su valor de cuantización.

El tipo de cuantizador ha variado en algunos trabajos de codificación conjunta de fuente canal de acuerdo al tipo de fuente (con o sin memoria, con o sin pérdidas) y sus propiedades estadísticas (pdf y σ^2). Por ejemplo en [1] utilizan un cuantizador de umbral uniforme (UTQ por sus siglas en inglés) para cuantizar todas las subbandas a excepción de la LFS la cual usa UTQ junto con los códigos de Huffman para su codificación, en [2] Farvardin utiliza un cuantizador de memoria cero; en [3] Modestino usa UTQ para la banda de más baja frecuencia y un cuantizador gaussiano generalizado óptimo (SOGGQ por sus siglas en inglés) no uniforme para las demás bandas. En [4] se utiliza un cuantizador SOGGQ y un cuantizador uniforme optimizado de fuente (SOUQ por sus siglas en inglés). Algunas técnicas importantes usadas para la cuantización y codificación de imágenes son mostradas a continuación.

5.2.3.1 Definición matemática

Un cuantizador Q , es matemáticamente definido como una correspondencia $Q : \mathbb{R}^p \rightarrow C$ [5]. Esto significa que vectores dimensionales p en el espacio vectorial \mathbb{R}^p son correspondidos dentro de una colección finita C de vectores que están también en \mathbb{R}^p . Esta colección C es también llamada libro de códigos (codebook en inglés) y el número de vectores en el codebook o puntos de reproducción, N , es conocido como el tamaño del codebook. Las entradas del codebook son conocidas como codewords o codevectors. Si $p = 1$ se tiene un cuantizador escalar (SQ por sus siglas en inglés). Si $p > 1$ se tiene un cuantizador vectorial (VQ por sus siglas en inglés). La resolución, el índice de código o simplemente la tasa de un cuantizador Q es $r = \frac{\log_2 N}{p}$, el cual mide el número de bits

por componente vectorial usadas para representar los vectores de entrada y da una indicación del promedio o precisión que es alcanzable con un cuantizador si el codebook esta bien diseñado.

Suponga que C tiene N entradas y_1, y_2, \dots, y_N . Para cada codevector, y_i existe una región R_i , tal que cualquier vector de entrada $x \in R_i$ sea mapeada o cuantizada a y_i . La región R_i es llamada una región de Voronoi [5,6] y es definida para ser el conjunto de todas las $x \in \mathbb{R}^p$ que son cuantizados para y_i .

Medida de distorsión

Una medida de distorsión o distancia entre dos vectores \hat{x} y \hat{y} que pertenecen al mismo espacio vectorial simbólicamente es dado por $d(x,y)$. La distancia es definida por:

$$d(x, y) = \sum_{i=1}^p |x_i - y_i|^r. \quad 5.3$$

Cuando $r=2$, la distancia euclidiana cuadrada emerge y ésta es frecuentemente usada en el diseño de los cuantizadores. También se le llama la varianza de error de cuantización.

Esta medida de distorsión cumple con tres propiedades dadas por:

1. Positividad: $d(x,y)$ es un número real más grande que o igual a cero con igualdad si y solo si $x=y$.
2. Simetría: $d(x,y)=d(y,x)$ y
3. Desigualdad triangular $d(x,z) \leq d(x,y) + d(y,z)$.

Criterio óptimo

Hay dos condiciones necesarias para que un cuantizador sea óptimo. La primera condición conocida como la reglas del vecino más próximo que declara que un cuantizador con un vector de entrada x mapea a un codevector lo más cercano a éste. Es

decir, \mathbf{x} corresponde a \mathbf{y}_i si y solo si $d(x, y_i) \leq d(x, y_j) \forall j \neq i$. Esto permite definir la región de Voronoi como:

$$R_i = \{x \in R^P : d(x, y_i) \leq d(x, y_j) \forall j \neq i\}.$$

La segunda condición especifica el cálculo del codevector \mathbf{y}_i dada una región de Voronoi R_i . El codevector \mathbf{y}_i es calculado para minimizar la distorsión promedio en R_i el cual es denotado por D_i donde:

$$D_i = E[d(x, y_i) | x \in R_i].$$

Diseño de Algoritmos

El diseño de los algoritmos es formulado para encontrar los codewords y las regiones de Voronoi a fin de minimizar la distorsión promedio global D dada por:

$$D = E[d(x, y)]. \quad 5.4$$

Si la densidad de probabilidad $p(x)$ de los datos \mathbf{x} es conocida, la distorsión promedio es:

$$D = \int d(x, y_i) p(x) dx = \sum_{i=1}^N \int_{R_i} d(x, y_i) p(x) dx. \quad 5.5$$

Si $r=2$ la varianza de error de cuantización o la distorsión de error cuadrático medio es dada por:

$$\sigma_q^2 = \sum_{i=1}^N \int_{R_i} (x, y_i)^2 p(x) dx.$$

5.2.3.2 Cuantización escalar

El cuantizador escalar (SQ por sus siglas en inglés) es el más simple de todos los esquemas de compresión con pérdidas. Como anteriormente se mencionó un cuantizador escalar es definido como una correspondencia en una dimensión ($p=1$).

Cuantizador escalar Lloyd-Max

El método de Lloyd-Max es usado para diseñar cuantizadores escalares y asume que la densidad de probabilidad de los datos escalares $p(x)$ es conocida [7,8]. Si los codewords (o niveles de reconstrucción) son denotados por y_1, y_2, \dots, y_N . Para cada codeword y_i , la región de Voronoi es un intervalo continuo $R_i = [v_i, v_{i+1})$. Donde v_i son los niveles de decisión y $v_i = -\infty, v_{i+1} = \infty$. La distorsión promedio es:

$$D = \sum_{i=1}^N \int_{v_i}^{v_{i+1}} d(x, y_i) p(x) dx. \quad 5.6$$

Obteniendo las derivadas parciales de D con respecto a v_i y y_i se obtienen las regiones óptimas de Voronoi y los codewords.

En el caso particular cuando $d(x, y) = (x_i - y_i)^2$, se muestra en [7] que una solución óptima para minimizar el error cuadrático medio es:

$$v_i = \frac{y_i + y_{i+1}}{2} \text{ para } 2 \leq i \leq N \quad 5.7$$

$$y_i = \frac{\int_{v_i}^{v_{i+1}} xp(x)dx}{\int_{v_i}^{v_{i+1}} p(x)dx} \text{ para } 1 \leq i \leq N. \quad 5.8$$

Estas dos ecuaciones no lineales tienen que ser resueltas simultáneamente dando los valores frontera v_1 y v_{N+1} . En la práctica, estas ecuaciones pueden ser resueltas por un esquema iterativo tal como el método de Newton.

Cuando el número de niveles de cuantización es grande, una solución aproximada puede ser obtenida por el modelado de la densidad de probabilidad $p(x)$ como una función constante de piecewise (figura 5.2) como:

$$p(x) \cong p(\hat{v}_k), \quad \hat{v}_k \stackrel{\Delta}{=} \frac{1}{2}(v_k + v_{k+1}), \quad v_k \leq x < v_{k+1}. \quad 5.9$$

Usando la aproximación de distorsión en el caso de $r=2$ y llevando a cabo las minimizaciones requeridas, una solución aproximada para los niveles de decisión es obtenida por:

$$y_i = \frac{A \int_{v_i}^{v_{i+1}} [p(x)]^{1/3} dx}{\int_{v_i}^{v_{N+1}} [p(x)]^{1/3} dx} + v_1, \quad 5.10$$

donde $A = v_{L+1} - v_1$ y $z_k = (K/L)A$, $k = 1, \dots, N$. Este método requiere v_1 y v_{N+1} , también llamados puntos de sobrecarga, sean finitos. Estos valores los cuales determinan el rango finito de A del cuantizador, tienen que ser supuestos antes de la colocación de los niveles de decisión y reconstrucción. Una vez que los niveles de transición $\{v_i\}$ han sido determinados, los niveles de reconstrucción $\{y_i\}$ pueden ser fácilmente determinados por el promedio de v_i y v_{i+1} . La distorsión cuadrática media del cuantizador es obtenido por:

$$D = \frac{1}{12N^2} \left\{ \int_{v_1}^{v_{N+1}} [p(x)]^{1/3} dx \right\}^3. \quad 5.11$$

Ésta es una fórmula muy útil porque da una estimación del error del cuantizador directamente en términos de la densidad de probabilidad y del número de niveles de cuantización. Este resultado es exacto para densidades de probabilidad constantes en piecewise.

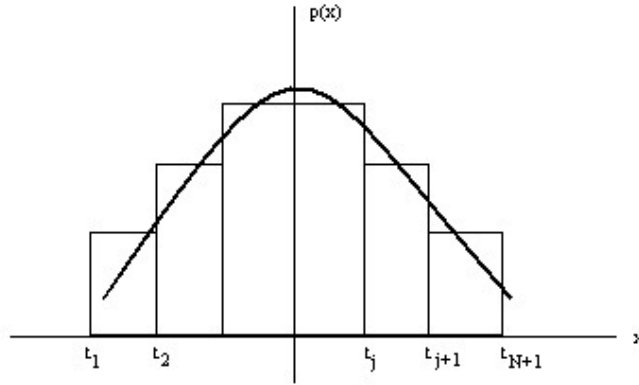


Figura 5.2 Aproximación constante piecewise de una p(x).

Para tasas altas de compresión R , se puede mostrar que el MSE de una cuantización de Lloyd-Max se comporta como:

$$d(R) \cong \varepsilon_*^2 \sigma^2 2^{-2R}, \tag{5.12}$$

donde ε_*^2 es función de una pdf particular. Para pdf simétricas, con media cero y planas se tiene:

$$\varepsilon_*^2 \sigma^2 = \frac{2}{3} \left[\int_0^\infty \sqrt[3]{p(x)} dx \right]^3. \tag{5.13}$$

Los valores de ε_*^2 para fuentes uniformes, laplacianas y gaussianas son dadas en la tabla 5.1.

	D(R)	Lloyd-Max	Codificación de entropía
Uniforme	0.703	1	1
Laplaciana	0.865	9/2	1.232
Gausiana	1	2.721	1.423

Tabla 5.1 Valores de ε_*^2 para varias pdf's.

Cuantizador Óptimo Uniforme.

Para distribuciones uniformes, las ecuaciones del cuantizador Lloyd Max llegan a ser lineales, dando intervalos iguales entre los niveles de transición y los niveles de reconstrucción. Éste es también llamado cuantizador lineal. Suponga:

$$p(x) = \begin{cases} \frac{1}{v_{N+1} - v_1}, & v_1 \leq x \leq v_{N+1}, \\ 0, & \text{en otro caso} \end{cases}, \tag{5.14}$$

de 5.8 se obtiene que los niveles de reconstrucción son:

$$y_i = \frac{(v_{i+1}^2 - v_i^2)}{2(v_{i+1} - v_i)} = \frac{v_{i+1} + v_i}{2} \tag{5.15}$$

y de 5.7 y 5.15 se obtiene que los niveles de decisión son:

$$v_i = \frac{v_{i+1} - v_{i-1}}{2},$$

lo cual da $v_i - v_{i-1} = v_{i+1} - v_i = \text{constante} = \Delta$.

Finalmente se obtiene:

$$\Delta = \frac{v_{N+1} - v_1}{N}, \quad v_i = v_{i-1} + \Delta, \quad y_i = v_i + \frac{\Delta}{2}. \quad 5.16$$

Así todas las transiciones tan bien como los niveles de reconstrucción son igualmente espaciados. El error de cuantización $e = x - x_q$ es uniformemente distribuido sobre el intervalo $(-\Delta/2, \Delta/2)$. Así el error cuadrático medio es dado por:

$$D = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} x^2 dx = \frac{\Delta^2}{12}. \quad 5.17$$

Para un cuantizador uniforme que tiene R bits, la variancia σ^2 de la variable aleatoria uniforme cuyo rango es A es $A^2/12$, por lo que se tiene $\Delta = A/2^R$. Y ésto da:

$$\frac{D}{\sigma^2} = 2^{-2R} \Rightarrow SNR = 10 \log_{10} 2^{2R} = 6B \text{ dB}. \quad 5.18$$

Así el índice de ruido-síñal alcanzado por el cuantizador cuadrático medio óptimo para una distribución uniforme es 6 dB por bit.

Si se considera el ejemplo de una entrada x con amplitudes en el rango $x \in (-x_{\max}, x_{\max})$ para un cuantizador uniforme se tiene:

$$\Delta = 2x_{\max} 2^{-2R}$$

por lo tanto

$$D = \sigma_q^2 = \frac{1}{3} x_{\max}^2 2^{-2R}. \quad 5.19$$

La función característica de un cuantizador uniforme se puede representar gráficamente como se muestra en la figura 5.3. En la figura se pueden identificar los siguientes conceptos:

- Tamaño del escalón de cuantización Δ , define la distancia entre dos niveles de cuantización.
- Zona Granular, o zona de la característica en la que el error de cuantización se mantiene entre los valores $\Delta/2$ y $-\Delta/2$.
- Zona de Sobrecarga, o zona en la que todas las entradas se cuantizan al máximo (o mínimo) nivel de cuantización, creciendo el error de cuantización de forma lineal.

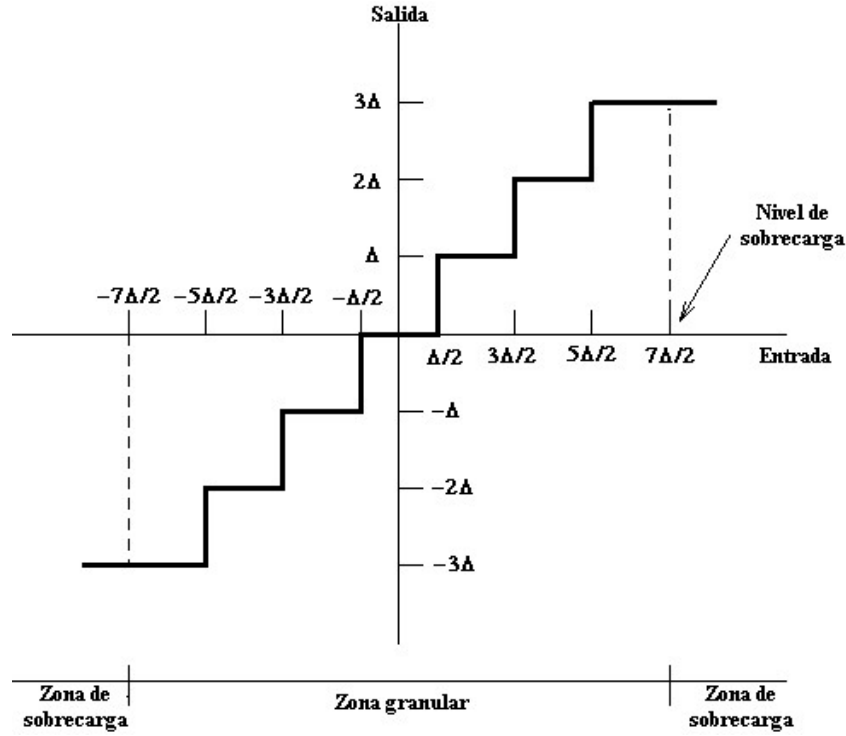


Figura 5.3 Característica de un cuantizador uniforme

Cuantización escalar para una codificación de entropía

El cuantizador Lloyd-Max minimiza el MSE sujeto a un límite sobre el tamaño del código, N . La presunción es que $R = \log_2 N$ bits será usada para señalar el codeword escogido por el cuantizador. Si $\log_2 N$ es un entero, esto es fácil. Si $\log_2 N$ no es entero, entonces $R = \lceil \log_2 N \rceil < \log_2 N + 1$ bits que pueden ser usados para señalar una señal a un tiempo. Alternativamente se pueden bloquear L señales q_0, q_1, \dots, q_{L-1} juntas para formar una “súper-senal” en $\{0, 1, \dots, N^L - 1\}$. Esta súper-senal puede ser entonces señalizada usando:

$$R = \frac{1}{L} \lceil \log_2 N^L \rceil < \log_2 N^L + \frac{1}{L} \text{ bits/señal.}$$

Así, se puede ver que el cuantizador Lloyd-Max puede ser fielmente aproximado con una codificación de longitud fija, aun cuando N no es una potencia de 2.

Considerando una codificación sin perdidas donde la aplicación de códigos de longitud variable como los de Huffman o los aritméticos para las señales producidas por la cuantización escalar no hay efecto sobre $\overline{Q^{-1}(q)} = \overline{Q^{-1}(Q(x))}$. Por lo tanto el MSE no cambia. Sin embargo la tasa de cuantización de la codificación de entropía puede aproximarse a la entropía de las señales, la cual es igual a $H(X)$. Por facilidad se asume que una codificación de entropía eficiente es utilizada por lo que:

$$R = H(X) \leq \log_2 N, \tag{5.20}$$

con una desigualdad alcanzando una igualdad si y solo si todas las salidas del cuantizador son igualmente probables.

En el caso que $H(X) \leq \log_2 N$ el cuantizador Lloyd-Max no es un cuantizador óptimo para una codificación de entropía. El cuantizador óptimo minimiza el MSE sujeto a una limitante en la entropía. De la ecuación 5.20 se puede observar que $N > 2^R$. N puede ser elegido arbitrariamente sin concernir al funcionamiento. De hecho para variables aleatorias sin límites (por ejemplo laplacianas, gaussianas) $N=\infty$ generalmente es la opción óptima.

Para una tasa R deseada, se busca minimizar

$$D = \sum_{i=1}^N \int_{v_i}^{v_{i+1}} (x, y_i)^2 p(x) dx \quad 5.21$$

sujeto a la limitante

$$H(X) = -\sum_{i=1}^N p_i \log_2 p_i \leq R \quad 5.22$$

donde

$$p_i = \int_{v_i}^{v_{i+1}} p(x) dx .$$

Usado la técnica de los multiplicadores de Lagrange se busca minimizar:

$$J(\lambda) = D + \lambda H(X) . \quad 5.23$$

Si existe una $\lambda \geq 0$ tal que la solución a la minimización ilimitada de $J(\lambda)$ produzca un $H(X)=R$, la misma solución satisfecerá el problema limitado de las ecuaciones 5.21 y 5.22. Obteniendo $\frac{\partial}{\partial x_i} J(\lambda) = 0$, se revela que y_i es la media condicional de la región de

Voronoi. Similarmente, diferenciando con respecto a v_i se obtiene:

$$(v_i - y_{i-1})^2 - (v_i - y_i)^2 - \lambda(\log_2 p_{i-1} - \log_2 p_i) = 0 . \quad 5.24$$

Para una λ , estas ecuaciones forman la base para un algoritmo iterativo el cual es la generalización del algoritmo de Lloyd-Max para un cuantizador escalar.

Cuando $H(X)$ es muy grande, el cuantizador escalar óptimo en el caso de una codificación de entropía es uniforme para PDF's planas. Los valores de salida del cuantizador pueden ser escritos de la forma $i\Delta, i = 0, \pm 1, \pm 2, \dots$ y los umbrales son los puntos medios entre los valores de salida. Así los intervalos de cuantización son dado por:

$$R_i = \left[i\Delta - \frac{\Delta}{2}, i\Delta + \frac{\Delta}{2} \right), \quad 5.25$$

y de la ecuación 5.6 tenemos

$$D = \sum_{i=1}^N \int_{i\Delta - \frac{\Delta}{2}}^{i\Delta + \frac{\Delta}{2}} (x, y_i)^2 p(x) dx = \frac{\Delta^2}{12}, \quad 5.26$$

similarmente

$$H(X) = - \sum_{i=-\infty}^{\infty} p_i \log_2 p_i = h(X) - \log_2 \Delta. \quad 5.27$$

Se asume una codificación de entropía eficiente por lo que $R \cong H(X)$. Entonces de la ecuación 5.27 $\Delta \cong 2^{h(X)-R}$. Substituyendo en la ecuación 5.26 se tiene:

$$d(R) \cong \frac{1}{2} 2^{2h(X)} 2^{-2R}. \quad 5.28$$

A tasas bajas, el cuantizador escalar óptimo de codificación de entropía no es lo suficientemente uniforme. Sin embargo, un cuantizador con intervalos uniformes, pero con codewords centroides, son casi óptimos. Frecuentemente, para PDF's con media cero, una pequeña mejora en las características de $d(R)$ puede ser obtenida por la ampliación del intervalo cerca de 0. Este intervalo R_0 , es algunas veces llamado el "zero-bin". Los cuantizadores de este tipo son usualmente llamados "cuantizadores escalares uniformes de zona muerta". La ampliación de R_0 incrementa algo la distorsión, pero frecuentemente disminuye $H(X)$ lo suficientemente para compensar este efecto. Los intervalos de un cuantizador de zona muerta son de la forma:

$$R_i = \begin{cases} [-(1-\xi)\Delta, (1-\xi)\Delta) & i = 0 \\ [(i-\xi)\Delta, (i+1-\xi)\Delta) & i > 0, \\ [(i-1+\xi)\Delta, (i+\xi)\Delta) & i < 0 \end{cases} \quad 5.29$$

donde $\xi < 1$ determina el ancho de R_0 .

Este cuantizador puede ser implementado como:

$$i = Q(x) = \begin{cases} \text{sign}(x) \left\lfloor \frac{|x|}{\Delta} + \xi \right\rfloor & \frac{|x|}{\Delta} + \xi > 0 \\ 0 & \text{otro caso} \end{cases}. \quad 5.30$$

Casos especiales interesantes ocurren cuando $\xi=0$ y $\xi=1/2$. Cuando $\xi=1/2$, R_0 tiene un ancho de Δ . En cambio con $\xi=1$ resulta un ancho del zero-bin de 2Δ .

Como anteriormente se dijo los valores de reconstrucción óptimos son centroides. Sin embargo por simplicidad, algunos valores fijos dentro de R_0 son frecuentemente usados. En este caso:

$$y_i = \begin{cases} 0 & i = 0 \\ \text{sign}(i)(|i| - \xi + \delta)\Delta & i \neq 0 \end{cases},$$

donde $0 \leq \delta < 1$ especifica el empleo de y_i dentro de R_0 . El caso con $\delta=1/2$, produce y_i en el centro de R_i . El cuantizador que resulta de $\xi=0$ y $\delta=1/2$ es descrito en la figura 5.4.

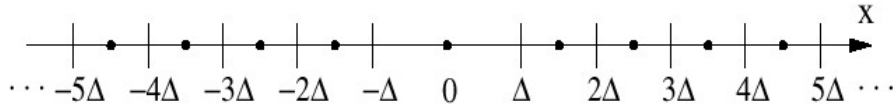


Figura 5.4 Cuantizador escalar uniforme con zona muerta.

Cuantización escalar embedded

Una característica deseable de los sistemas de compresión es la habilidad de refinar sucesivamente los datos reconstruidos del bitstream al ser decodificados. En esta situación el decodificador reconstruye una aproximación de los datos reconstruidos después de la decodificación de una porción del bitstream comprimido. Como la mayor parte del bitstream comprimido es decodificada, la calidad de reconstrucción puede ser mejorada, hasta que una calidad de reconstrucción completa es lograda sobre la decodificación del bitstream entero. Los sistemas de compresión que poseen esta propiedad son facilitados por la cuantización embedded.

En la cuantización embedded, los intervalos de los cuantizadores de tasa alta son incrustados dentro de los intervalos de los cuantizadores de tasa baja. Equivalentemente los intervalos de los cuantizadores de tasas más bajas son divididos para producir los intervalos de los cuantizadores de tasas altas. Considere una secuencia de K cuantizadores escalares embedded $Q_0, Q_1, Q_2, \dots, Q_{K-1}$. Los intervalos de Q_{K-1} son entonces incrustados dentro de los intervalos de Q_{K-2} , los cuales en cambio son incrustados dentro de los Q_{K-3} y así sucesivamente. Equivalentemente los intervalos de Q_0 son divididos para conseguir los intervalos de Q_1 , los cuales a su vez son divididos para obtener los intervalos de Q_2 y así sucesivamente.

Específicamente cada intervalo de Q_0 ($R_0, q_0, = 1, 2, \dots, M_0 - 1$) es dividido entre M_1 intervalos de $R_{q_0, q_1}, q_1, = 1, 2, \dots, M_1 - 1$. El número total de intervalos de Q_1 es entonces de $M_0 M_1$. Similarmente los intervalos de Q_1 son divididos para obtener los intervalos de Q_2 como $R_{q_0, q_1, q_2}, q_2, = 1, 2, \dots, M_2 - 1$. En general entonces, Q_k ($k = 0, 1, 2, \dots, K - 1$) tiene

$$\prod_{j=0}^k M_j \text{ intervalos dados por } R_{q_0, q_1, \dots, q_k}.$$

Un notable ejemplo donde todos los cuantizadores embedded pueden ser óptimos es en el caso uniforme. Un ejemplo en particular es el cuantizador de zona muerta uniforme. Para el caso cuando $\xi = 0$, se tiene:

$$q = Q(x) = \text{sign}(x) \left\lfloor \frac{|x|}{\Delta} \right\rfloor \quad 5.31$$

$$y \quad \overline{Q}^{-1}(q) = \begin{cases} 0 & q = 0 \\ \text{sign}(q)(|q| + \delta)\Delta & q \neq 0 \end{cases} \quad 5.32$$

Este cuantizador tiene incrustado dentro de él, todos los cuantizadores de zona muerta con tamaños de paso $2^p\Delta$ para enteros $p \geq 0$.

Asumiendo que la magnitud de q puede ser representada por K bits, entonces q puede ser escrita en la forma de magnitud de señal como:

$$q = Q_{K-1}(x) = s, q_0, q_1, \dots, q_{K-1}, \tag{5.33}$$

donde s es la señal, q_0 es el bit más significativo (MBS) y q_{K-1} es el bit menos significativo (LSB). Ahora, se supone

$$q^{(p)} = Q_{K-1-p}(x) = s, q_0, q_1, \dots, q_{K-1-p}$$

será la señal obtenida por la reducción de los últimos p bits de q . Equivalentemente, $q^{(p)}$ es obtenido por el corrimiento a la derecha de la representación binaria de $|q|$ por p bits. Esto es entonces fácilmente verificado por:

$Q_{K-1-p}(x) = q^{(p)}$ donde Q_{K-1-p} es el cuantizador de zona muerta uniforme con una tamaño de paso de $2^p\Delta$. La figura 5.5 ilustra esta inserción para $p=1$ y $p=2$.

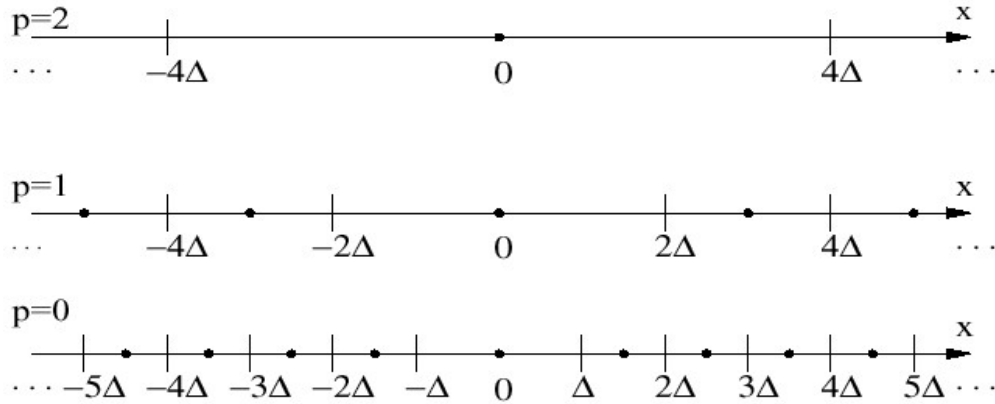


Figura 5.5 Cuantizador escalar uniforme de zona muerta embedded.

De esta discusión se puede deducir que si los LSB's p de $|q|$ no son disponibles en el decodificador, aun es posible obtener una aproximación de x , pero a un nivel bajo de calidad. En particular, el resultado será el mismo si la cuantización se hubiese llevado a cabo usando un tamaño de paso de $2^p\Delta$ (antes que Δ) en primer lugar. En este caso la cuantización inversa es realizada por:

$$x = \overline{Q_p^{-1}}(q^{(p)}) = \begin{cases} 0 & q^{(p)} = 0 \\ \text{sign}(q^{(p)}) (|q^{(p)}| + \delta) 2^p \Delta & q^{(p)} \neq 0 \end{cases} \tag{5.34}$$

Note que cuando $p=0$ se produce una decuantización de calidad completa dada por la ecuación 5.32.

5.2.3.3 Cuantización vectorial

La cuantización vectorial (VQ por sus siglas en inglés) es una generalización de la cuantización escalar. El salto de una a varias dimensiones es un paso más que permite nuevas ideas, nuevos conceptos y técnicas que ofrecen mejoras pero que normalmente no

ofrecen una contrapartida con respecto al caso escalar. Mientras que la cuantización escalar se usa sobre todo para conversión analógico-digital, la cuantización vectorial se usa en un procesamiento más complejo de la señal, en los que la mayoría de los casos tiene alguna forma de representación digital que permite una versión comprimida de la señal original. Realmente, la cuantización vectorial se usa, aunque no exclusivamente, para compresión de datos.

Un vector se puede usar para describir casi cualquier tipo de patrón, por ejemplo un segmento de una forma de onda o una imagen, simplemente formando un vector con sus muestras. Así la cuantización vectorial puede ser vista como una forma de reconocimiento de patrones, donde el patrón de entrada puede ser aproximado a un elemento de un conjunto de patrones predefinidos.

A continuación se muestran algunas definiciones básicas para el estudio de los cuantizadores vectoriales, estas complementan las ya descritas anteriormente en sección 5.2.3.1.

Recordando la función de correspondencia, la resolución y la distorsión de un cuantizador Q . Podemos decir que a cada uno de los N puntos definidos en el codebook está asociada una partición de R^p o celda, R_i , definida como sigue:

$$R_i = \{x \in R^p : Q(x) = y_i\}$$

La idea intuitiva para explicar el concepto de partición dentro del contexto de los cuantificadores es la siguiente. Uno de los parámetros del cuantificador es el número de niveles o tamaño N y otro la dimensión p . El número de niveles indica cuántas regiones, o lo que es lo mismo, cuántos puntos compondrán el cuantificador. Estos puntos formarán unas regiones de dimensión p o particiones; en realidad estos puntos son los centroides de estas regiones, que se definen a continuación.

El centroide de una partición R dada, $y^* = \text{cent}(R)$, como el vector y que minimiza la distorsión media entre los puntos x de R e y^* es:

$$y^* = \text{cent}(R) \text{ si } E[d(x, y^*) | x \in R] \leq E[d(x, y) | \forall y \in R] \quad 5.35$$

Un método sencillo para calcular el centroide de una partición es la media aritmética:

$$\text{cent}(R) = \frac{1}{\|R\|} \sum_{i=1}^{\|R\|} x_i \quad 5.36$$

donde $R = \{x_i : i = 1, \dots, \|R\|\}$ y $\|R\|$ es la cardinalidad de R .

En la figura 5.6 se ha representado un espacio de 2 dimensiones ($p=2$), que se ha dividido en 11 particiones ($N=11$). Cada una de las particiones contiene un punto que representa al centroide de la partición.

Existe una clase especial de cuantizadores vectoriales, denominados cuantizadores vectoriales de *Voronoi* o de vecino más cercano (*Nearest Neighbor Quantizers*). Estos cuantizadores se caracterizan porque las particiones quedan totalmente definidas por un codebook y una medida de distorsión. La principal ventaja que ofrecen es que no requieren ninguna definición explícita de la descripción geométrica de las celdas, sino que esta información va definida implícitamente con el codebook y la medida de distorsión.

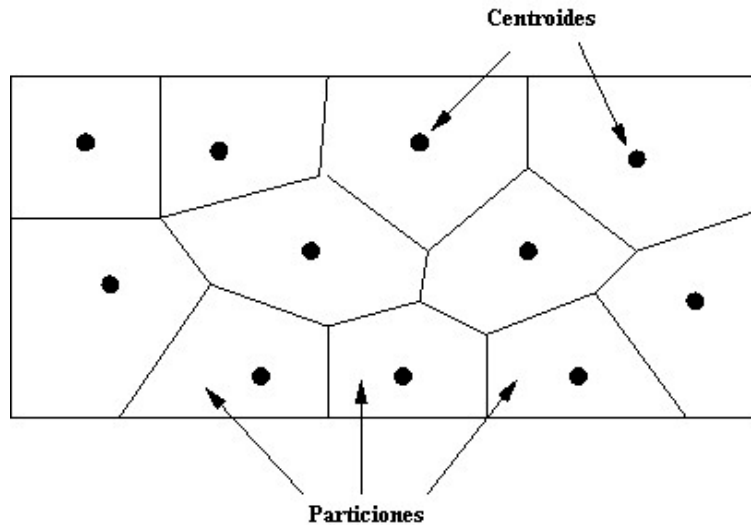


Figura 5.6 Particiones y centroides.

Por lo tanto se define un cuantizador vectorial de Voronoi o *NN* (*Nearest Neighbor*) como aquél en que la partición en celdas viene dada por la siguiente ecuación:

$$R_i = \{x : d(x, y_i) \leq d(x, y_j) \forall y_j \in C\}. \quad 5.37$$

En otras palabras, con un codificador *NN*, cada celda R_i se forma con todos los puntos x que tienen una distorsión cuando son codificados con el vector y_i menor que la que tendrían si se codificaran con cualquier otro vector del codebook. De esta forma, podemos definir la función Q como:

$$Q(x) = \arg \min_{y \in C} \{d(x, y)\} \quad 5.38$$

Normalmente, cuando se habla de cuantizadores vectoriales, se habla de cuantizadores vectoriales *NN*.

El algoritmo de Lloyd generalizado

El algoritmo de Lloyd generalizado, permite diseñar fácilmente cuantizadores que minimizan la distorsión para un conjunto de ejemplos de diseño; es una extensión del Algoritmo de Lloyd para el caso escalar. Dicho algoritmo está basado en el uso iterativo de operaciones de modificación de un codebook inicial, operaciones que se resumen en la denominada Iteración de Lloyd. Ambos casos, el escalar y el vectorial, son paralelos y siguen los mismos principios. Esta sección se centra en el caso vectorial.

Existen dos variantes de la Iteración de Lloyd. En la primera de ellas, se conoce la distribución de la señal de entrada, con lo cual el cálculo de las particiones se hace más fácil (Iteración de Lloyd Conocidas las Estadísticas). En la segunda, esa distribución se desconoce, y se utiliza una señal de entrada que sirve como casos de ejemplo o conjunto de entrada, y a partir de los cuales operará el algoritmo (Iteración de Lloyd para Datos Empíricos). Este apartado se centra en la segunda variante.

La Iteración de Lloyd para Casos Empíricos se aplica directamente sobre una distribución definida por los casos de prueba, T , para obtener un cuantizador para esta distribución. La idea esencial es que si se toma un conjunto finito T de tamaño M suficientemente grande, y se obtiene un cuantizador suficientemente óptimo para este conjunto, este cuantizador será también óptimo para la distribución real. La Iteración de Lloyd para Casos Empíricos consta de dos pasos, y se describe en la tabla 5.2.

Tabla 5.2 Iteración de Lloyd para casos Empíricos.

Iteración de Lloyd para casos Empíricos (C_m, T, N)	
1.	Dado un codebook $C_m = \{y_i; i = 1, \dots, N\}$ partir el conjunto de entrada T en particiones R_i usando la siguiente condición: $R_i = \{x \in T : d(x, y_i) \leq d(x, y_j) \text{ para todo } j \neq i\}$
2.	Usando la definición de centroide dada en la ecuación 5.35, se calculan los centroides de cada partición para calcular el codebook. Hacer $C_{m+1} = \{\text{cent}(R_i)\}$. Si se generó una celda vacía en el paso 1, se asignará un vector alternativo (en vez del cálculo del centroide) para esa celda.

Existen diversas alternativas para solucionar el problema de la celda vacía, como generar un centroide aleatoriamente, o partir otras celdas en dos siguiendo distintas heurísticas para elegir la celda a partir.

En la tabla 5.3 se concretan los pasos de los que consta el Algoritmo de Lloyd Generalizado.

Tabla 5.3 Algoritmo de Lloyd Generalizado.

Algoritmo de Lloyd Generalizado (C_I, T, N)	
1.	Comenzar con un codebook inicial C_I . Sea $m=1$.
2.	Dado un alfabeto, C_m , ejecutar la Iteración de Lloyd para generar un nuevo codebook C_{m+1} .
3.	Calcular la distorsión media para C_{m+1} .
4.	Si ha cambiado en una pequeña cantidad solamente desde la iteración anterior, parar. Sino, hacer $m=m+1$ e ir al paso 2.

Un problema que surge en el primer paso de este algoritmo es el de elegir un codebook inicial. En el paso 3 del algoritmo de Lloyd generalizado, aparece también un concepto nuevo: la distorsión media, que es utilizado como medida de la calidad del

diseño del cuantizador. La distorsión media puede definirse como la media de la distorsión que se produce entre cada vector del conjunto de casos de prueba, y el vector del codebook al que se cuantifica:

$$D = \frac{1}{M} \sum_{j=1}^M \min(d(x_i, y)), \quad y \in C \tag{5.42}$$

La forma más común de utilizar esta medida para generar la condición de fin del algoritmo de Lloyd es la siguiente:

$$(D_m - D_{m+1}) / D_m < \epsilon,$$

donde ϵ es un umbral que se utiliza como parámetro de diseño, $0 \leq \epsilon \leq 1$

VQ de estructura de árbol

La VQ de estructura de árbol (TSVQ por sus siglas en inglés) es una de las técnicas más eficientes y ampliamente usadas para reducir la compleja búsqueda en VQ. La idea básica de TSVQ es buscar una serie de pequeños codebooks dirigidos a la elección de un vector en el codebook C . Al nivel 0 (o raíz) del árbol hay un solo codebook C_0 de longitud p y tamaño $N_0 = 2^{pR_0}$. La señal cuantizada obtenida es $q_0 = Q_0(x)$.

Al nivel 1 del árbol, hay N_0 codebooks C_{1,q_0} $q_0 = 0, \dots, N_0 - 1$. Cada uno de estos codebooks son de longitud p y de tamaño $N_1 = 2^{pR_1}$ y la señal obtenida es $q_1 = Q_{1,q_0}(x)$. Al nivel 2 del árbol hay N_1 codebooks para cada elección de q_0 , denotados por C_{2,q_0,q_1} $q_1 = 0, \dots, N_1 - 1$, cada uno de longitud p y de tamaño $N_2 = 2^{pR_2}$. Si $Q_{1,q_0}(x) = q_1$ entonces x esta sujeto a C_{2,q_0,q_1} y se produce $q_2 = Q_{2,q_1,q_0}(x)$ y así sucesivamente. Esta situación es descrita en la figura 5.7, en esta figura hay $K=3$ niveles con $N_0=N_1=N_2=2$.

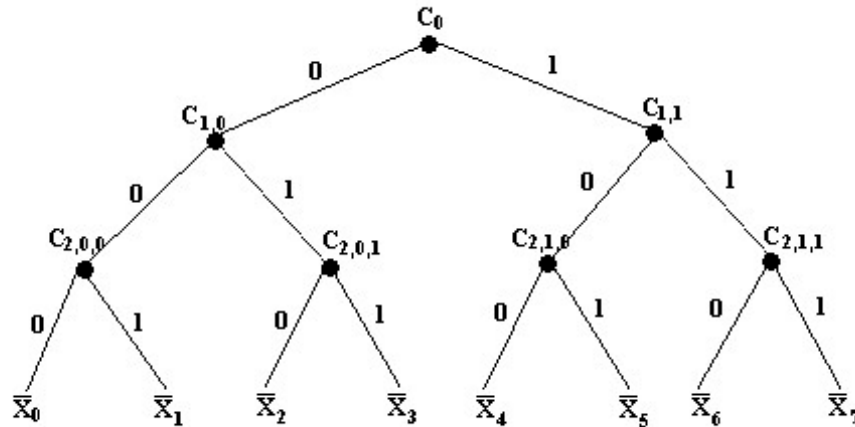


Figura 5.7 VQ de estructura de árbol binario con $K=3$ niveles

Mas generalmente, si para el nivel 0, hay un codebook de tamaño $N_0 = 2^{pR_0}$. Al nivel 1 hay N_0 codebook, cada uno de tamaño $N_1 = 2^{pR_1}$ para un total de $N_0 N_1 = 2^{p(R_0+R_1)}$

codebooks. Al nivel 2 hay N_0N_1 codebook, cada uno de tamaño N_2 para un total de $N_0N_1N_2 = 2^{p(R_0+R_1+R_2)}$ codebooks. Para un árbol de K niveles, lo anterior continua hasta

$K-1$, donde hay $N = \prod_{k=0}^{K-1} N_k = 2^{p \sum_{k=0}^{K-1} R_k}$ codebooks. El índice del TSVQ resultante es

$$R = \frac{1}{p} \log_2 N = \sum_{k=0}^{K-1} R_k .$$

Como un codebook puede ser buscado en cada nivel, la complejidad de la búsqueda de TSVQ es proporcional a $\sum_{k=0}^{K-1} N_k$. Si los codebooks del mismo tamaño son usados en cada nivel entonces:

$$R_k = \frac{R}{K} . \quad 5.43$$

Con esta elección, $N_k = 2^{pR_k} = 2^{pR/K}$ por lo que la complejidad de la búsqueda total es proporcional a:

$$\sum_{k=0}^{K-1} N_k = K 2^{pR/K} . \quad 5.44$$

La complejidad más pequeña es lograda por un árbol binario con $N_k=2$, o $R_k=1/p$.

Muchos algoritmos han sido propuestos para el diseño de TSVQ. Una aproximación razonable es el algoritmo de Lloyd generalizado para diseñar C_0 .

Cuantización de codificación de malla.

La cuantización de codificación de malla (TCQ por sus siglas en inglés) esta basada sobre las ideas de un conjunto de señales expandidas y un conjunto dividido de modulación codificada, y ha sido mostrado que es un método eficiente con complejidad modesta para la codificación de fuentes sin memoria.

Una malla no es más que un estado de transición para una máquina de estado finito. Las mallas son usadas para el estudio de secuencias de estados de transición, o equivalentemente, secuencias de estados. Una malla típica con 8 estados es mostrada en la figura 5.8. En la figura cada columna de puntos (o nodos) representa los ocho posibles estados en cualquier punto dado en el tiempo. Esos estados son rotulados del 0 al 7, de arriba a abajo. Cada rama en la malla representa una transición de un estado a otro, en el próximo punto en el tiempo (nuevo estado).

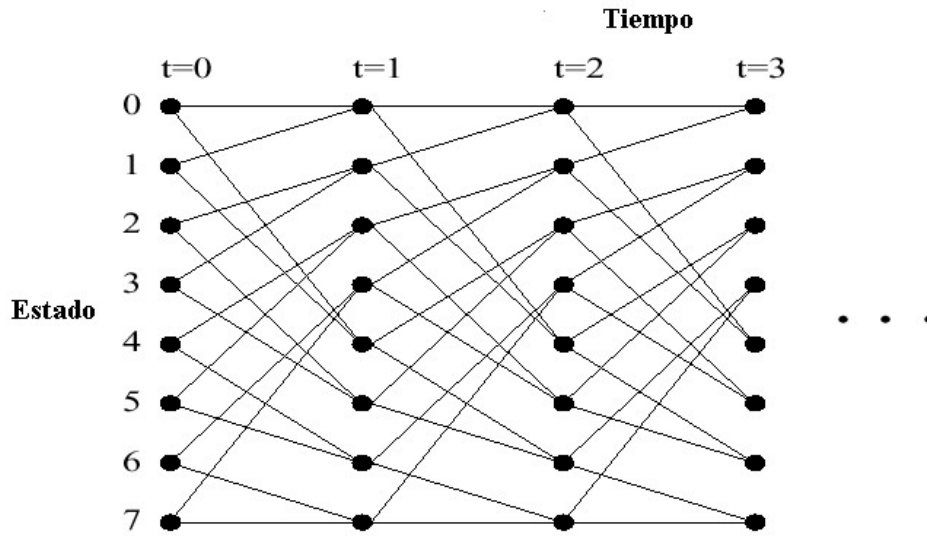


Figura 5.8 Diagrama típico de malla.

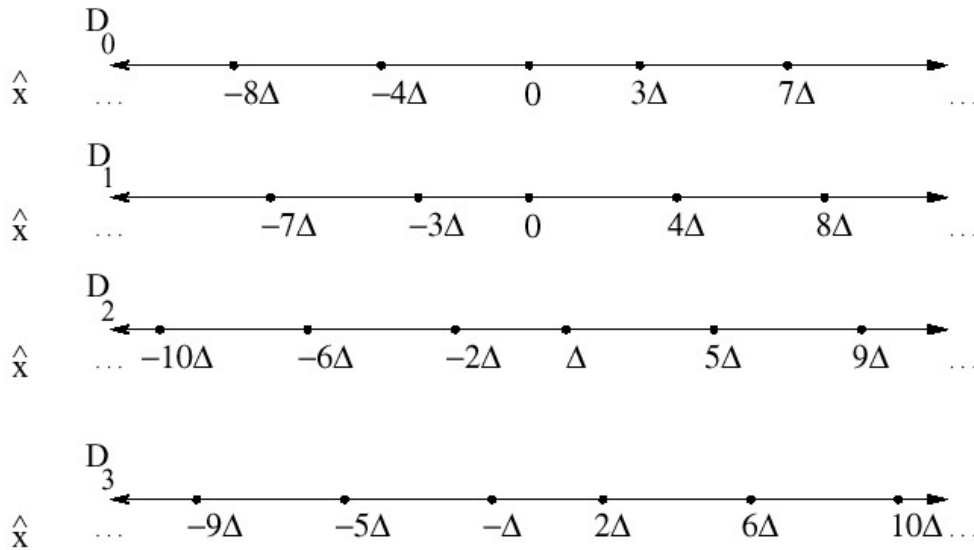
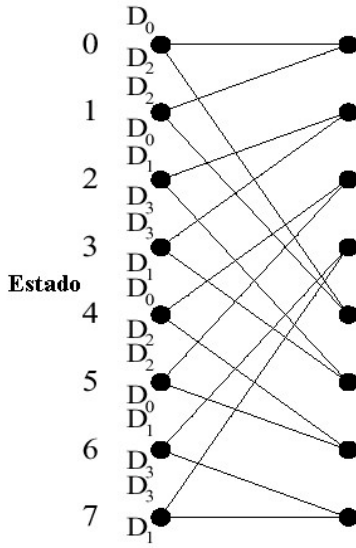
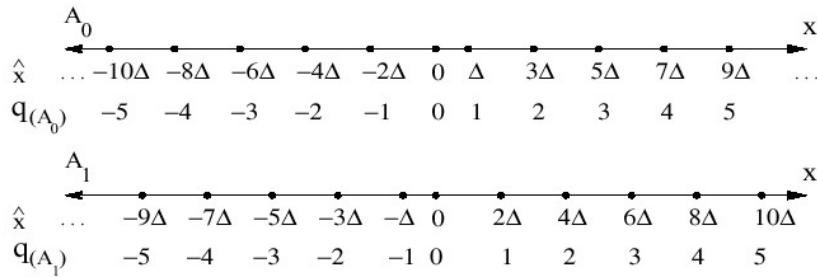


Figura 5.9 Cuantizadores escalares usados para TCQ

Para un cuantizador escalar uniforme se divide dentro de cuatro subconjuntos llamados D_0, D_1, D_2 y D_3 . Esto es ilustrado en la figura 5.9. Los subconjuntos D_i son usados para rotular las ramas de la malla. La figura 5.10a muestra un estado simple de la malla de 8-estados. La unión de los cuantizadores asociados con cada estado es llamado un cuantizador de unión. Note que los dos cuantizadores de unión usados para la malla en la figura 10a son $A_0 = D_0 \cup D_2$ y $A_1 = D_1 \cup D_3$. Estos cuantizadores son ilustrados en la figura 10b, además son mostrados también los valores correspondientes de reconstrucción \hat{x} de cada cuantizador de unión y la correspondiente señal del cuantizador de unión $q(A_i)$.



(a)



(b)

Figura 5.10 (a) Ramas rotuladas para una TCQ (b) Cuantizadores de unión para TCQ

La estructura de la malla permite la codificación usando el algoritmo de Viterbi. Para codificar una secuencia de datos $x \in \mathfrak{R}^p$, una malla de N -estados de m estados es empleada. Note que tal malla tiene $m+1$ columnas de estados. Permita que $S_{i,l}$, $i = 0, 1, \dots, m$, $l = 0, 1, \dots, N-1$ denoten el estado l y la columna i . Para cada estado $S_{i+1,l}$, $S_{i,l'}$ y $S_{i,l''}$ serán los dos estados que tengan ramas finales en $S_{i+1,l}$. También, $D^{l',l}$ y $D^{l'',l}$ serán los conjuntos asociados con esas ramas, respectivamente. Denote $c_{l',l}$ y $c_{l'',l}$ como los codewords en $D^{l',l}$ y $D^{l'',l}$ que minimizan $\rho(x_i, c) = (x_i - c)^2$, y denote $d^{l',l} = (x_i - c_{l',l})^2$ y $d^{l'',l} = (x_i - c_{l'',l})^2$. Finalmente $s_{i+1,l}$ será la “distorsión sobreviviente” asociada con la trayectoria sobreviviente del estado $S_{i+1,l}$.

El i -enésimo paso ($i=0, \dots, m-1$) en el algoritmo de Viterbi entonces consiste de fijar $s_{i+1,l} = \min\{s_{i,l'} + d_{l',l}, s_{i,l''} + d_{l'',l}\}$, preservando la rama que alcanza este mínimo, mientras se eliminan las otras ramas de la malla. Si dos valores comparados con la mínima distorsión sobreviviente son iguales, la igualdad puede ser resuelta arbitrariamente sin impactar sobre el MSE.

Cuando el fin de los datos es alcanzado ($i=m-1$), la malla se remonta del estado final que tiene la distorsión sobreviviente más baja, y el correspondiente conjunto de señales TCQ es producido. Para secuencias largas de datos la elección del estado inicial tiene un efecto insignificante sobre el MSE. Así se fija el estado inicial arbitrariamente a cero.

Una simple modificación puede ser aplicada a las señales de cuantización obtenidas como se describe arriba, permitiendo una codificación de entropía más eficiente. Si se asume que la secuencia de entrada x_i tiene una distribución de probabilidad simétrica.

La decuantización de las señales de TCQ en el decodificador es fácil. La secuencia de señales especifica que codeword fue escogido del cuantizador de unión apropiado a cada estado y esto es suficiente para permitir al decodificador reproducir los valores de reconstrucción, dados en el estado inicial.

5.2.4 Uso eficiente de los recursos del bit.

Frecuentemente los sistemas de codificación de señales contienen diferentes cuantizadores, cada uno de los cuales tiene la tarea de codificar un parámetro diferente que es necesario para caracterizar la señal. Cada uno de estos parámetros puede tener un significado físico diferente y puede requerir de un promedio relativo diferente de reproducción para lograr una calidad global deseada en la imagen de reconstrucción. El número de bits disponibles para describir colectivamente este conjunto de parámetros es inevitablemente limitado. Consecuentemente, una gran preocupación para el diseñador de un sistema de codificación es la asignación de bits, la tarea de distribuir a una cuota dada de bits los diversos cuantizadores para optimizar el funcionamiento global del codificador.

5.2.4.1 El problema de la asignación de bit

Asumiendo que una señal consiste de K componentes $\{x_i, i=1,2,\dots,K\}$ formando un vector \mathbf{x} , donde la variancia del número de componente i es igual a $\sigma_{x_i}^2$ y todas las componentes tiene media cero. Suponiendo que la pdf de cada x_i es conocida y una medida de distorsión es seleccionada, se puede diseñar un cuantizador óptimo Q_i para x_i , basado sobre una medida de distorsión, para cualquier elección de N_i niveles de cuantización, donde N_i es un entero no negativo. Es frecuentemente deseable que N_i sea una potencia entera de 2 por lo que la resolución $b_i=\log_2 N_i$ es un número entero de bits.

Si $W_i(b)$ denota el error cuadrático medio incurrido en la cuantización óptima de x_i con b bits de resolución. Tratando cada variable aleatoria aisladamente se tiene por lo menos en principio, una solución completa para el funcionamiento que puede ser logrado en la cuantización de una variable aleatoria. La función $W_i(b)$ dice cuanto se puede reducir la distorsión promedio por el incremento de la resolución o en otras palabras, se

tiene el funcionamiento –precio de intercambio disponible. El precio es el número de bits que se asignan y el funcionamiento es determinado por la distorsión media que resulte.

El funcionamiento global de un sistema de codificación puede ser simplemente determinado de la suma de las distorsiones medias. Así se puede definir la distorsión global D como una función de un vector de asignación de bits, $b=(b_1, b_2, \dots, b_k)$ de acuerdo a:

$$D = D(b) = \sum_{i=1}^k W_i(b). \quad 5.45$$

El problema de asignación de bit es determinar los valores óptimos de b_1, b_2, \dots, b_k sujetos a una cuota fija dada, B , de bits disponibles. La limitante de índice de bits es dada por:

$$B = \sum_{i=1}^K b_i. \quad 5.46$$

En este punto hay dos aproximaciones básicas para encontrar una asignación óptima o cercanamente óptima. La primera es intentar encontrar un algoritmo que exactamente minimice 5.45 sobre todas las opciones de b . Y la segunda aproximación en la cual se invocan aproximaciones de cuantizaciones a alta resolución para escribir expresiones explícitas para distorsiones óptimas $W_i(b_i)$. La expresión resultante para $D(b)$ puede entonces ser minimizada con la familiar técnica de Lagrange de cálculo o desigualdades comunes.

Si se asume que la distorsión de sobrecarga es insignificante y que las aplicaciones de alta resolución aplican, entonces de 5.11 se obtiene:

$$W_i(b_i) \approx h_i \sigma_i^2 2^{-2b_i}, \quad 5.47$$

donde la constante h_i es determinada por la pdf $f_i(x)$ de la variable aleatoria normalizada x_i/σ_i . Los valores de éste se pueden ver en la tabla 5.1.

El uso de aproximaciones de alta resolución no es correcto ciertamente para valores pequeños de b_i como 1 o 2 bits, pero frecuentemente se vuelven aproximaciones razonables aun en el caso de resoluciones bajas y medias.

Por simplicidad de notación, primero se tratará el caso donde cada x_i tiene la misma pdf normalizada, es decir, x_i/σ_i es idénticamente distribuida. En este caso $h_i = \epsilon_*^2$, es una constante independiente de i .

Soluciones clásicas al problema de asignación de bits.

La asignación de bits usando la aproximación de índice alto con límites enteros y no negativos sobre las asignaciones de bits es dada por la siguiente formula cuando las variables aleatorias tienen una pdf normalizada idénticamente distribuida y las distorsiones por componente no son pesadas:

$$b_j = \bar{b} + \frac{1}{2} \log_2 \frac{\sigma_{x_j}^2}{\rho^2}, \quad 5.48$$

donde $\bar{b} = \frac{B}{K}$ es el número promedio de bits por parámetro, K es el número de parámetros y $\rho^2 = \left[\prod_{n=1}^K \sigma_{x_n}^2 \right]^{1/K}$ es la media geométrica de las variancias de las variables aleatorias.

La distorsión global mínima obtenida con esta solución es dada por:

$$D = K \varepsilon_*^2 \rho^2 2^{-2\bar{b}}$$

y cada cuantizador incurrirá la misma distorsión promedio, esto es, $W_i(b_i) = \varepsilon_*^2 \rho^2 2^{-2\bar{b}}$. Esta fórmula dará en general valores no enteros y negativos de la cuenta de bits. Por lo que se utilizarán algoritmos ambiciosos para resolver este problema.

Ahora si se considera un caso más general donde las variables aleatorias x_i son no idénticamente distribuidas, entonces las constantes de distorsión distintiva h_i , son determinadas de la pdf normalizada correspondiente a las variables aleatorias. El problema de minimización es cambiado con la tarea de minimizar:

$$\sum_{i=1}^k h_i \sigma_i^2 2^{-2b_i}$$

sujeto a la misma cuota de bits B . Es fácilmente visto que esta cantidad es la misma a la previa función objetivo, excepto que ahora el valor de la varianza es escalado $h_i \sigma_i^2$ por lo que la solución es la siguiente:

$$b_j = \bar{b} + \frac{1}{2} \log_2 \frac{\sigma_{x_j}^2}{\rho^2} + \frac{1}{2} \log_2 \frac{h_j}{H} \quad 5.49$$

donde H es la media geométrica de los coeficientes h_i . Y la distorsión global mínima obtenida con esta solución es dada por:

$$D = KH \rho^2 2^{-2\bar{b}}.$$

Como una generalización general, si se considera el caso donde la asignación de bits es optimizada para una medida de distorsión global pesada, donde los pesos g_i son especificados y continuando con una pdf normalizada no distribuida idénticamente, la solución al problema de optimización es dada por:

$$b_j = \bar{b} + \frac{1}{2} \log_2 \frac{\sigma_{x_j}^2}{\rho^2} + \frac{1}{2} \log_2 \frac{h_j}{H} + \frac{1}{2} \log_2 \frac{g_j}{G}, \quad 4.50$$

donde G es la media geométrica de los valores de peso g_i . Y la distorsión global mínima obtenida con esta solución es dada por:

$D = KHGF \rho^2 2^{-2\bar{b}}$. En este caso cada cuantizador contribuye una distorsión promedio inversamente proporcional al valor correspondiente de peso g_i . Para esta asignación óptima, el cuantizador tendrá una distorsión global relativamente pequeña.

Una mejora al problema de asignación de bits es la adición de un límite a la resolución de cada cuantizador para evitar que sean negativos (o que el número de niveles N_i sea al menos la unidad). Este requerimiento podría eliminar la ocurrencia de asignaciones de bits negativas sin requerir la solución para tener valores enteros de resolución. Esta resolución fue tomada por Segall quien usó algunas técnicas básicas de análisis funcional para obtener una solución a este problema. Él también generalizó el problema para aplicar a cualquier función de distorsión convexa $W(b)$ que es estrictamente convexa con una primera derivada continua $W'_i(b)$ tal que $W'_i(\infty) = 0$. Esta condición implica que la distorsión del cuantizador decrezca ligeramente como el número de bits incrementa y el índice de decrecimiento de distorsión tiende a cero como el número de bits incrementa sin límite.

Asume que cada X_i tiene una pdf idénticamente normalizada. Ponga $w(b) = W_i(b) / \sigma_i^2$; la cual es independiente de i , para ser el error cuadrático medio incurrido en la cuantización óptima X_i / σ_i^2 . Ponga la inversa de $w(b)$ como $J(w)$. Ya que la función de distorsión es decreciente con el incremento de la resolución b , la inversa existe y está bien definida.

De esta forma se declara, sin probar, la solución de Segall al problema de asignación, donde la asignación óptima es dada por:

$$b_i = \begin{cases} b_i^* = J\left(\frac{\theta^*}{\sigma_i^2} w'(0)\right), & \text{si } 0 < \theta^* < \sigma_i^2, \\ 0, & \text{si } \theta^* \geq \sigma_i^2 \end{cases},$$

donde θ^* es la única raíz de la ecuación:

$$S(\theta) = \sum_{i: \sigma_i^2 \geq \theta} J\left(\frac{\theta}{\sigma_i^2} w'(0)\right) = B,$$

donde B es la cuota total de bits.

5.2.4.2 Codificación de subbanda

En la codificación de subbanda, un banco de filtros opera sobre la señal de entrada para generar un conjunto de señales de ancho de banda angosto cada una representando a una subbanda diferente del espectro de entrada. El ancho de banda angosto de cada señal de subbanda permite que el submuestreo se lleve a cabo, reduciendo las necesidades de codificación del bit stream de cada subbanda. En esta parte primero se definirá la teoría básica de la codificación de subbanda en una forma ideal para posteriormente mostrar un caso real.

Supóngase que se tiene un proceso estacionario aleatorio gaussiano $\{X_n\}$ con media cero, variancia σ_X^2 , índice de muestreo f_s , y densidad espectral $S_X(f)$. Se define un banco de filtros paso banda ideal:

$$H_m(f) = \begin{cases} 1 & \text{para } \frac{(m-1)f_s}{2M} < f < \frac{mf_s}{2M} \\ 0 & \text{otro caso} \end{cases}$$

Así cada filtro M tiene un ancho de banda de f_s/M . Cuando X_n es aplicado a cada filtro, la salida obtenida son las señales de subbanda X_{mn} para $m=1,2,\dots,M$. La naturaleza ideal de los filtros implica que:

$$X_n = \sum_{m=1}^M X_{mn}.$$

También es fácil mostrar que las señales de subbanda son decorrelacionadas una con otra, es decir: $E[X_{mn} X_{ls}] = 0$ para $m \neq l$. Por lo que para todos los enteros n y s se tiene:

$$X_n = \sum_{m=1}^M X_{mn},$$

$$\sigma_X^2 = \sum_{m=1}^M \sigma_m^2,$$

donde σ_m^2 es la variancia de la m enésima señal de subbanda.

Para codificar las señales de subbanda, un cuantizador Q_m con b_i bits opera sobre cada muestra sucesiva X_{mn} de la m enésima señal de subbanda produciendo los valores $\hat{X}_{mn} = Q_m(X_{mn})$. El valor final de reproducción \hat{X}_n para una señal compuesta es dado por operación de reconstrucción o síntesis:

$$\hat{X}_n = \sum_{m=1}^M \hat{X}_{mn}.$$

Para este esquema de codificación se tiene que el error cuadrático medio es:

$$D = E[(\hat{X}_n - X_n)^2] = \sum_{m=1}^M (\hat{X}_{mn} - X_{mn})^2,$$

lo cual muestra que la distorsión global es la suma de las distorsiones promedio en cada subbanda. Con una asignación de bits óptima entre las subbandas usando una aproximación de alta resolución, la distorsión mínima alcanzable promedio es dada por:

$$D_{sbc} = M h_g \rho_s^2 2^{-2\bar{b}}$$

donde \bar{b} es el promedio de la asignación de bits para las muestras de subbandas. Así $\bar{b} = B/M$ donde $B = \sum_{m=1}^M b_i$, ρ_s^2 es la media geométrica de las variancias de la subbandas y h_g es la constante de cuantización dada.

El índice general de f_s muestras por segundo de las señales de subbanda colectivamente consiste de Mf_s muestras por segundo. Para hacer una transición a un esquema más significativo, si tomamos cada señal de subbanda de $f_s/2M$ (es decir, de igual ancho) esto puede ser submuestreado por un factor de M mientras se permite una recuperación de la señal de subbanda original de la secuencia de índice reducido de muestras. Específicamente, se extrae cada M enésima muestra de cada subbanda para

formar el muestreo de las señales de subbanda dadas por: $Y_{mk} = \hat{X}_{mk}$ donde $n=kM$, y k es ahora la señal de tiempo para las señales de submuestra.

Los mismos cuantizadores son usados para las submuestras Y_{mk} que producen valores cuantizados **Error! Objects cannot be created from editing field codes.**, donde \hat{X}_{mk} y \hat{Y}_{mk} tienen la misma variancia. Así el índice de asignación de bits y la distorsión global promedio resultan donde el índice reducido de f_s/M implica que el índice de bits global es $R = \bar{b}f_s$ bits por segundo.

Supóngase que es llevado acabo el PCM sobre la señal de entrada X_n con una cuantización óptima a un índice de $\bar{b}f_s$ b/s. Entonces, el mínimo error cuadrático medio es dado por:

$$D_{PCM} = h_g \sigma_X^2 2^{-2\bar{b}}. \quad 5.51$$

Ahora, la ventaja del funcionamiento de la codificación de subbanda comparada con el PCM puede ser medida por la proporción:

$$\frac{D_{PCM}}{D_{sbc}} = \frac{h_g \sigma_X^2 2^{-2\bar{b}}}{M h_g \rho_X^2 2^{-2\bar{b}}} = \frac{\frac{1}{M} \sum_{m=1}^M \sigma_X^2}{(\prod_{m=1}^M \sigma_X^2)^{1/M}}. \quad 5.52$$

Así la ganancia de funcionamiento es simplemente la proporción aritmética de la media geométrica de las variancias de las subbandas. Consecuentemente la ganancia es más grande cuando hay una gran variación en los niveles de potencia de las diferentes subbandas. Claramente para una gran variación de la densidad espectral de entrada y un suficiente número de bandas se logra una ganancia substancial de funcionamiento.

5.2.5 Efectos de transmisión de error

Para la codificación de canal y fuente conjunta, se debe de considerar además de la codificación de fuente o errores de cuantización, los efectos de los bits de fuente corrompidos, debido a los errores de canal, sobre la imagen reconstruida.

Generalmente conociendo los parámetros de la distribución estadística de la fuente descompuesta como en [3,4], se puede evaluar la contribución del error de reconstrucción cuadrático medio debido a un error en cualquier posición de bit dada. Esas contribuciones pueden ser expresadas en términos de sensibilidades de bits individuales en cualquier posición de bit dada y conduce a una aproximación útil y manejable para los efectos combinados de los errores de codificación de canal y fuente. Estas medidas de error son contabilizadas en la distorsión total de sistema y de esta forma se logra una asignación de bit global de tal forma que con un código de protección de errores se evite la propagación de errores en la decodificación.

En otros esquemas como en [1] para prevenir la propagación infinita de errores de decodificación, se paquetizan las secuencias de codewords antes de la transmisión. Donde

los paquetes consisten de dos partes: el indicador de longitud que indica la longitud de la porción de información en bits y la porción de información que consiste de una secuencia de codewords binarios. Además se consideran medidas correctivas para que el sistema tenga menos sensibilidad al ruido del canal.

Un método posible para mitigar los efectos de los errores de canal es el uso de la codificación de control de errores. De esta forma, de todos los bits usados para codificar la imagen, algunos serán usados en la codificación de fuente mientras el resto será guardado para proveer protección contra el ruido del canal.

5.2.5.1 Índice de codificación de canal

Para un sistema del mundo real, con diferentes sensibilidades al error se pueden asignar diferentes índices de códigos de canal para diferentes componentes de salida del codificador, resultando diferentes niveles de protección, es decir índices de bits de error (BER's por sus siglas en inglés). Entonces se puede asignar un solo índice de canal, resultando probabilidades de bits de error idénticas para todos los codewords de una componente de la fuente o podrían asignarse diferentes índices de código de canal, resultando en probabilidades de bits de error para un bit que pertenece a una componente de la fuente. Este último esquema requiere de una clase de códigos que permitan diferentes índices.

Denotando $R_{i,j}$ como el índice del código de canal (en bits por el uso del canal) asignado a la posición j de un bit de una componente i , se puede escribir el índice de código de canal promedio (en bits fuentes por uso del canal) de la componente i como:

$$R_{c,i} = \frac{n_i}{\sum_{j=1}^{n_i} 1/R_{i,j}}.$$

5.2.6 Índice y distorsión global

Siendo posible calcular la distorsión introducida por codificación de fuente y la distorsión global incluyendo los errores de canal, (es decir, la suma de la distorsión de la codificación de fuente D_s y la distorsión debido a los errores de canal D_c), se puede describir la distorsión global D_{s+c} (en MSE) para todas las asignaciones posibles de diferentes bits de cuantización por componente de una fuente e índices de códigos de canal diferentes $R_{i,j}$ para diferentes posiciones de las muestras de una componente i (con un BER resultante) como:

$$D_{s+c} = D_s + D_c = \sum_{i=1}^K D_{s+c_i}.$$

El índice global resultante es:

$$R_{s+c} = \frac{R_s}{R_c}.$$

El problema es entonces similar al problema de asignación de bit para la codificación de fuente pura, para encontrar el número de bit de cuantización para cada componente de la fuente junto con los índices del canal para las diferentes clases de sensibilidad a error para obtener así una distorsión global D_{s+c} bajo la limitante de un índice global máximo R_{s+cj} dado. Es posible resolver este problema para igualar una distorsión global específica D_{s+c} con el índice global mínimo R_{s+c} .

5.2.6.1 Asignación de bit de un canal fuente conjunto

La tarea de asignar los bits correspondientes a un cierto índice global R_{s+c} de tal forma que se obtiene una distorsión mínima para un cierto BER o SNR sobre un canal es por medio de un algoritmo de asignación del bit óptimo.

En [1] el algoritmo usado es un algoritmo de programación entera descrita por Shoham y Gersho en [11], y en [3, 4,12] se usa el algoritmo de Westerink [13].

En [3] para mejorar el funcionamiento global del sistema propuesto, llevan a cabo una asignación de código dentro de cada componente de la fuente.

Referencias

- [1] N.Tanabe and N.Farvardin, "Subband image coding using entropy-coded quantization over noisy channels," IEEE Journal on Selected Areas in Communications vol.10, pp.926–942, June.1992.
- [2] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channels: an approach to combined source-channel coding," IEEE Trans. on Info. Theory, vol. 33, No.6, pp. 827–838, Nov. 1987.
- [3] M.J. Ruf and J.W. Modestino, "Operational rate-distortion performance for joint source and channel coding of images," IEEE Trans. on Image Processing, pp. 305–320, March 1999.
- [4] J. Cai and C. W. Chen, "Operational rate-distortion design for joint source-channel coding over noisy channels", in Proc. IEEE WCNC'99, New Orleans, LA, Sept. 1999.
- [5] A. Gersho and R.M Gray, "Vector quantization and signal compression", Kluwer Academic Publishers, 1991.
- [6] A. Gersho "Asymptotically optimal block quantization", IEEE Trans. Infor. Theory, IT-25, 373-380, July, 1979.
- [7] N.S. Jayant and P. Noll, "Digital Coding of waveforms", Principles and applications to Speech and Video, Prentice-Hall, Englewood Cliffs, NJ, 1984.
- [8] J. Max, "Quantizing for minimum distortion", IEEE Trans. Infor. Theory, 7-12, Mar. 1960.
- [9] D.A. Huffman, "A method for the construction for minimum redundancy codes", Proc. IRE, 40(9), 1098-1101, Sept. 1952.
- [10] J.M Lervik, "Subband image communication over digital transparent and analog waveform channels", Ph.D. Thesis, Norwegian University of Science and Technology, Norway, June 1993.

- [11] Y. Shoham and A. Gersho, “Efficient bit allocation for arbitrary set of quantizers”, IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-36, pp. 1445-1453, Sept. 1988.
- [12] J. Cai and C. W. Chen, “Robust Joint source-channel coding for image transmission over wireless channels”, IEEE Trans. on Circuits and Systems for Video Tech., Vol. 10, No. 6, Sept. 2000.
- [13] P. H. Westerink, J. Biemond, and D. E. Boekee, “An optimal bit allocation algorithm for subband coding,” in ICASSP, pp. 757–760, 1988.
- [14] H. Coward, “Joint Source-Channel Coding: Development of methods and utilization in image communication”, Ph.D. Thesis, Norwegian University of Science and Technology, Norway, 2001.
- [15] V.K. Madisetti and D.B. Williams, “The digital signal processing handbook”, IEEE Press, USA 1998.
- [16] A.K. Jain, “Fundamentals of digital image processing”, Prentice Hall Information and systems sciences series, USA 1998.
- [17] A. Ambardar, “Analog and digital signal processing”, Brooks/Cole Publishing Company, 2da edition, USA 1999.
- [18] “Digital Coding of Waveform: Principles and Applications to Speech and Video”
- [19] “Vector Quantization and signal Compression”
- [20] D.S. Taubman y M. W. Marcellin, “JPEG2000 Image Compression Fundamentals Standards and Practice”, Kluwer Academic Publishers, London, 2002

Capítulo 6

Codificación de imágenes.

La codificación de imágenes juega un role importante dentro de las comunicaciones modernas de multimedia y la compresión de imágenes probablemente representa una fuente dominante dentro del tráfico en una red actualmente. Por tal motivo el estándar JPEG2000 intenta ser el sucesor de JPEG en muchas áreas de aplicación. Esto, es motivado principalmente por la necesidad de representaciones de imágenes comprimidas, las cuales ofrezcan características incrementales demandadas por las aplicaciones modernas; además de un funcionamiento de compresión superior. En este capítulo se verán algunas de estas características incrementales que ofrece el estándar JPEG2000 además de algunos ejemplos de transmisión utilizando JPEG2000. La demostración de las características se hará utilizando las utilerías del software Kakadu, el cual es una implementación de la parte 1 de JPEG 2000.

6.1 Cualidades de la codificación de JPEG2000

JPEG2000 brinda un nuevo paradigma a los estándares de compresión de imágenes. En este estándar se incluyen bien los cuatro modos de operación de JPEG (secuencial, progresivo, jerárquico y sin pérdidas). El compresor puede decidir la calidad máxima de calidad hasta una descompresión sin pérdidas siempre y cuando se haya codificado de esta forma. También se escoge en el tiempo de compresión la resolución máxima y el tamaño.

Por ejemplo suponga una imagen (en este caso la imagen de Lena de 512x512 pixeles y 8 bpp) es comprimida sin pérdidas a su tamaño completo. Suponga que el archivo resultante es de B_0 bytes de tamaño. Entonces es posible de extraer B_1 bytes del archivo, ($B_1 < B_0$) y descomprimir esos B_1 bytes para obtener una imagen descomprimida con pérdidas. Esta imagen será idéntica a la imagen obtenida si la compresión hubiese sido llevada a cabo para B_1 bytes desde un principio. En la siguiente serie de figuras 6.1 y 6.2 se puede observar lo anterior.



6.1 (a) Compresión JPEG2000 sin pérdidas



6.2 (a) Descompresión JPEG2000 sin pérdidas



6.1 (b) Compresión JPEG2000 con rate = 1bps



6.2 (b) Descompresión JPEG2000 con rate = 1bps



6.1 (c) Compresión JPEG2000 con rate = 0.5bps



6.2 (c) Descompresión JPEG2000 con rate = 0.5bps



6.1 (d) Compresión JPEG2000 con rate = 0.1bps



6.2 (d) Descompresión JPEG2000 con rate = 0.1bps

Serie de figuras 6.1 Todas las imágenes se obtuvieron comprimiendo al rate indicado por el compresor JPEG2000.

Serie de figuras 6.2 Todas las imágenes se obtuvieron a partir de la descompresión de la imagen 6.1a sin pérdidas al rate indicado.

NOTE. Que el funcionamiento del descompresor a índices muy bajos no es similar al del compresor; esto se debe a que la serie de imágenes de descompresión JPEG2000 se obtuvo de la imagen comprimida inicialmente sin pérdidas, mientras que en la otra serie cada imagen se comprimió al rate indicado y se descomprimió tal cual.

Similarmente es posible extraer B_2 bytes del archivo y descomprimir para obtener una imagen de resolución reducida. Al igual que la anterior, la imagen resultante será exactamente la misma si la versión de resolución baja de la imagen hubiese sido comprimida para B_2 bytes desde un principio. Ver figura 6.3.

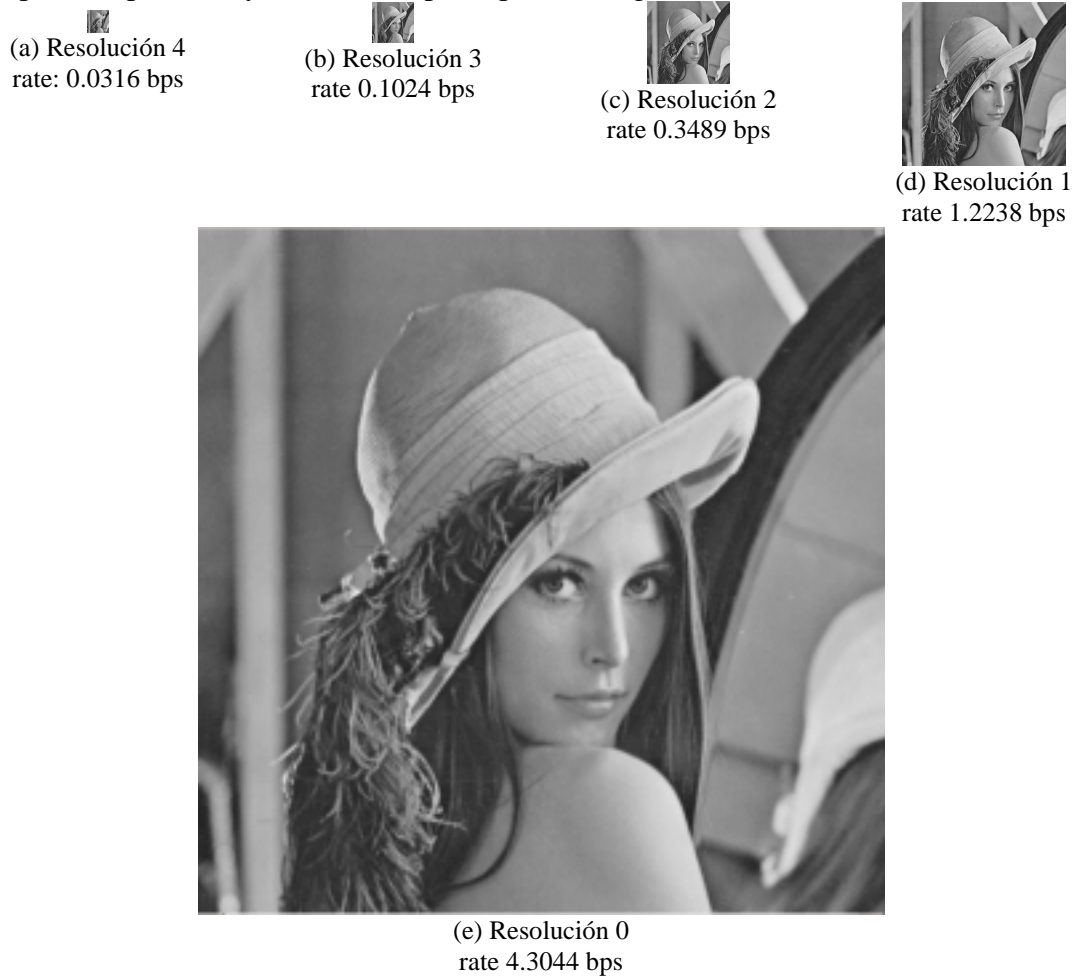


Figura 6.3 Niveles de resolución en la descompresión de un bitstream JPEG2000.

Además de la escalabilidad de calidad y de resolución, los codestream de JPEG2000 soportan acceso aleatorio espacial. Por medio de este acceso aleatorio se pueden recuperar y descomprimir datos del codestream correspondientes a regiones espaciales de una imagen, ver figura 6.4. Este acceso aleatorio se extiende a componentes también. Por ejemplo una componente en escala de grises se puede obtener de una imagen a color. Similarmente aquellas componentes que contengan textos o gráficos pueden ser extraídas cuando se presenten. Esto puede ser hecho región por región con diferentes calidades y resoluciones.

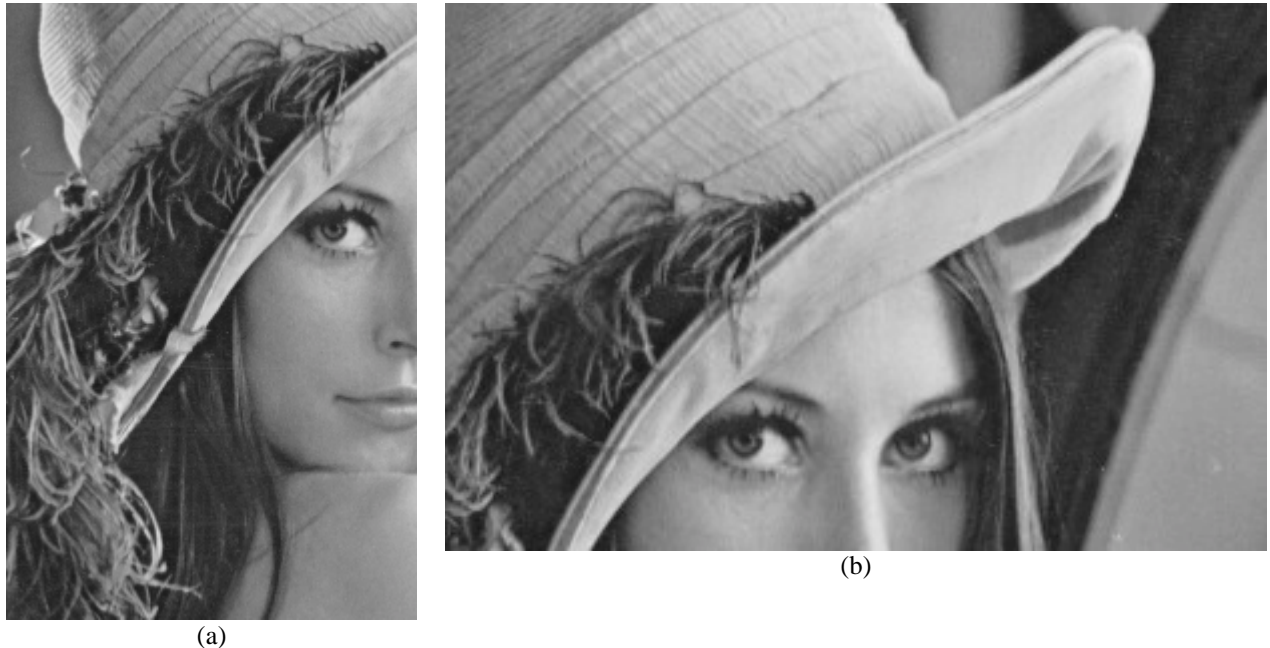


Figura 6.4 Descompresión JPEG2000 de partes de la imagen de Lena comprimida con el formato de JPEG200 sin pérdidas.

Es importante notar que para la localización, extracción y decodificación de los bytes requeridos para el producto de una imagen deseada, no necesariamente se decodifica el codestream entero y/o imagen.

Cualquiera de las imágenes de arriba puede ser extraída de un code-stream JPEG2000 para crear un nuevo code-stream JPEG2000. Claramente, cualquier cosa que puede ser descomprimida y desplegada puede ser recomprimida y almacenada. Sin embargo, con JPEG2000 los bits comprimidos relevantes pueden ser extraídos y reacomodados dentro de un code-stream de conformidad sin descomprimir. Específicamente la resolución reducida y/o la calidad reducida de las imágenes comprimidas pueden ser producidas sin un ciclo de descompresión y recompresión, por lo cual en algunos casos se evita la construcción del ruido de compresión.

Además para reducir la calidad y la resolución, el cortado (cropping) del dominio de la imagen comprimida es posible. El cortado en el dominio de compresión es realizado por el acceso a los datos comprimidos asociados con una región espacial dada y rescribiéndola como un code-stream de conformidad.

Las manipulaciones geométricas son también parcialmente soportadas en el dominio de compresión. Las rotaciones de 90,180 y 270 grados son posibles. Ver figura 6.5. El reflejo inverso (mirroring) o el volteo (flipping) (de arriba a bajo o de izquierda a derecha) puede ser también llevado a cabo (ver figura 6.6). Estos procedimientos no se pueden realizar enteramente en el dominio de compresión. Alguna transcodificación de datos es requerida, pero el ciclo de la transformación inversa y hacia delante es evitado, y solamente pequeños cambios en la distorsión y en el bit-rate incurren.



(a) Rotada 90°



(b) Rotada 180°



(c) Rotada 360°



(e) Rotada 270°

Figura 6.5 Rotaciones en una imagen comprimida JPEG2000. Las rotaciones pueden realizarse en el proceso de compresión o de descompresión.



Figura 6.6 Reflejo inverso en una imagen comprimida JPEG2000. Los reflejos pueden realizarse en el proceso de compresión y/o descompresión.

Esta idea puede ser también usada para decodificar y procesar una versión rotada o reflejada de la imagen en una manera más aproximada. Esto sería útil para enviar grandes imágenes a impresoras de baja memoria basadas en el escaneo.

Todas las variaciones anteriores pueden ser efectuadas en el tiempo de compresión o en las operaciones de análisis y descompresión. Esta capacidad deriva de la independencia de los code-blocks, y por lo tanto las dimensiones de los code-blocks gobiernan la granularidad de las regiones espaciales que pueden ser elegidas.

JPEG2000 también permite al codificador seleccionar regiones de forma y tamaño arbitrario para un tratamiento preferencial. En este caso la ROI (Region of Interest) puede ser escogida en el tiempo de codificación y no es fácilmente alterada por el análisis o decodificación. Para esta forma de codificación de ROI, los coeficientes wavelet que afectan las muestras de la imagen dentro de la ROI son previamente enfatizados antes de la codificación del plano de bits. La cantidad de pre énfasis es escrita en el code-stream, y es usada apropiadamente para realinear los coeficientes ROI en el tiempo de decodificación. La figura 6.7 muestra una imagen con una ROI.



(a) bitrate 1bps



(b) bitrate 0.5bps



(c) bitrate 0.05bps

Serie de figuras 6.7. Imágenes codificadas con una ROI al bitrate indicado. A la columna derecha se le incluyeron los valores de peso en la codificación de la imagen. Ambas columnas fueron codificadas de tal forma que el fondo se separa de la región de interés.

Otro tipo de ventajas que se pueden realizar durante el proceso de codificación de una imagen en JPEG2000, debido a la independencia que existe en los tiles y code-stream, es la elección de los valores de los parámetros de entrada al codificador que modifican la imagen completa, un determinado tile o componente. La siguiente imagen (serie de imágenes 6.8) fue codificada de tal forma que cada tile tiene un cierto índice de bits, con características diferentes en cuanto al tamaño del codeblock, el tipo de transformada wavelet, los niveles de resolución en la transformada wavelet, la transformada de componentes, el tamaño de paso de cuantización, etc.

La imagen girasol a color con tres componentes de tamaño 495x700 fue codificada de la siguiente manera: precisión de 8bpp para cada componente, sin signo, un submuestreo de 1 por cada componente, .9 tiles de 165x234 cada uno, 6 niveles de calidad (sin pérdidas, 1bps, 0.5bps, 0.25bps, 0.1bps y .05bps), orden de progresión LRCP (capa, resolución, componente y posición), sin uso de recintos, transformada de componentes (a menos que se indique otro), transformada wavelet no reversible (a menos que se indique otro), con 5 niveles de resolución (a menos que se indique otro), cuantización expuesta, tamaño de paso de cuantización base 1/256 (a menos que se indique otro), tamaño del codeblock 64x64 (a menos que se indique otro).

Tile 0: 1bps, transformada wavelet reversible, tamaño del codeblock 32x32. Tile 1: sin pérdidas, con 3 niveles de resolución. Tile 2: 0.05bps. Tile3: 0. 1bps, tamaño de paso de cuantización base 1/128. Tile 4: 0.5bps, sin transformada de componentes, tamaño del codeblock 32x32. Tile 5: 0.25bps, sin transformada de componentes, con 3 niveles de resolución. Tile 6: 0.5bps, Tile 7: 0.25bps, con 3 niveles de resolución, tamaño de paso de cuantización base 1/128. Tile 8: 0.1bps, sin transformada de componentes, transformada wavelet reversible.



Imagen original de girasoles



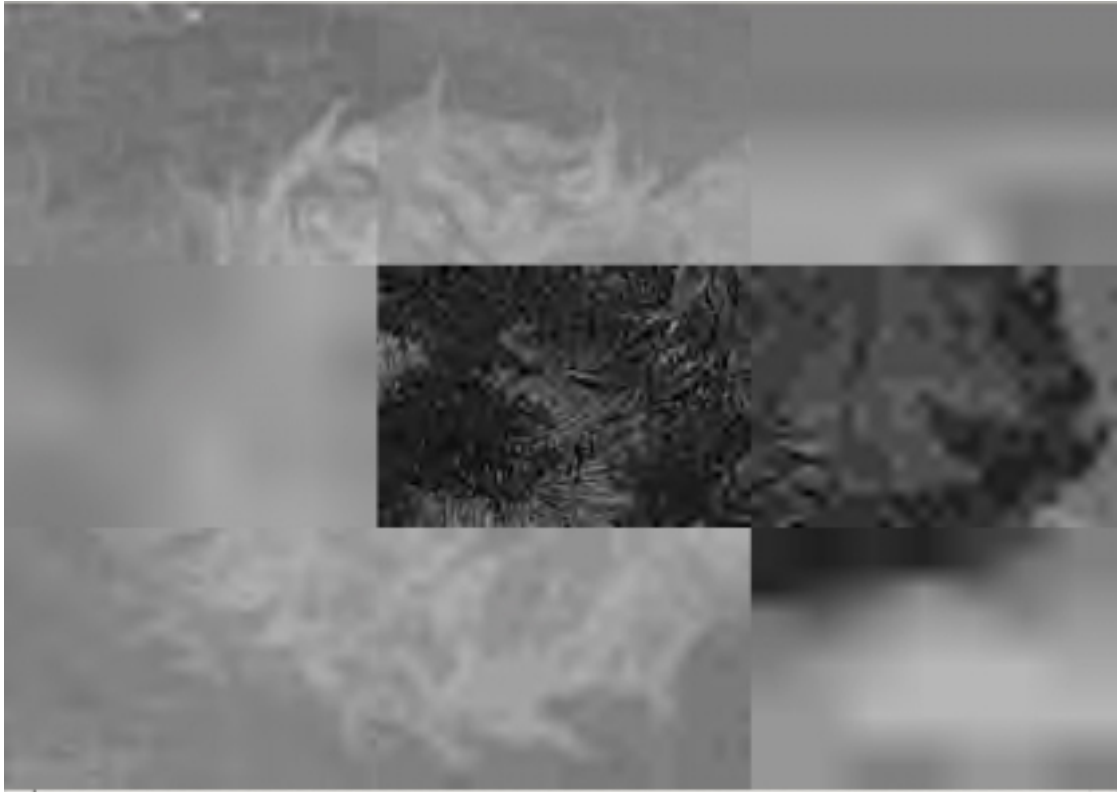
(a) Imagen a color comprimida con JPEG2000



(b) Componente de luminancia



(c) Primera componente de diferencia de color



(d) Segunda componente de diferencia de color

Serie de figura 6.8. Imagen de girasol a color con sus tres componentes codificada con 9 tiles de diferentes características.

6.2 Transmisión de imágenes JPEG2000 sobre canales ruidosos

Se considera un sistema de codificación conjunta de fuente canal (JSCC por sus siglas en inglés) con un método de asignación de índice para la transmisión de flujos de códigos (codestreams) JPEG-2000 (J2K), los cuales son protegidos contra los efectos de ruido del canal por una protección de error única (UEP por sus siglas en inglés).

Los esquemas UEP aplican diferentes cantidades de protección a diferentes secciones del bitstream, por lo cual son capaces de reducir la protección de secciones menos importantes del bitstream para proveer mayor protección en las secciones de mayor importancia.

Así los índices de codificación del canal y la fuente son conjuntamente optimizados para formar un bitstream UEP J2K, utilizando las características de los codestream J2K para formar un bitstream codificado conjuntamente de canal fuente de multi capas a un índice de bits total deseado.

En esta parte del capítulo se mostraran dos trabajos que utilizan esta estructura para la transmisión de bitstreams JPEG2000 u otro tipo de codificación de fuente por capas sobre canales simétricos binarios. En el primer caso [1] se utilizan los códigos

RCPC para proveer la protección única de error y en el segundo [2] toma un código de canal sistemático Reed-Solomon.

6.2.1 Estructura de la fuente

En un J2K una señal de entrada primero es dividida en tiles rectangulares no sobrepuestos. Si la imagen tiene múltiples componentes, las muestras de cada componente que caen dentro de un particular tile son referidas como tile-componente. Cada componente tile-componente es entonces transformado usando una transformada wavelet y las subbandas wavelet son divididas dentro de diferentes estructuras geométricas. La más pequeña de estas estructuras es un codeblock. Los codeblocks son formados por la fragmentación de las subbandas wavelet. Los codewords de resoluciones particulares son agrupados juntos para formar recintos o zonas.

Una vez que las subbandas wavelet son cuantizadas, cada codeword es comprimido individualmente usando un codificador de bit-plane. El codificador de bitplane realiza tres pasos sobre cada bitplane de un codeblock. Esos pasos son referidos como pasos de codificación.

Los datos comprimidos de cada codeblock pueden ser considerados como un bitstream incrustado. El codificador J2K calcula y usa la información del índice de distorsión asociada con cada paso de codificación de cada codeblock. El codestream es formado incluso por los diferentes números de pasos de codificación de cada codeblock, dependiendo de un criterio dado. Si el decodificador detecta un error dentro de un paso de codificación, este descartará el actual y los futuros pasos de decodificación pertenecientes a un codeblock.

Para el propósito de la formación del codestream, los datos comprimidos de cada recinto son clasificados para formar paquetes. Cada paquete contiene una cabecera y un cuerpo. La cabecera del paquete contiene información acerca de la contribución de cada codeword en el recinto del paquete. El cuerpo del paquete contiene los pasos de codificación de los codeblocks en ese recinto. Un paquete de cada recinto de cada resolución de cada tile componente forma una capa. Por lo tanto conceptualmente, una capa es un incremento de calidad para un tile entero. Los paquetes que pertenecen a un tile particular son agrupados juntos para formar un tile-stream, y los tile-stream son agrupados juntos para formar un codestream J2K. Similar a los paquetes los tile-stream constan de una cabecera y de un cuerpo. Hay también una cabecera principal al inicio de cada codestream de cada imagen. La información de cabecera a varios niveles es crucial para la decodificación correcta del codestream.

J2K provee diferentes utilerías para la recuperación de error para combatir la propagación de errores a lo largo del codestream y guarda la sincronización entre el codificador y el decodificador. J2K provee un mecanismo referido como cabeceras de paquete empaquetizadas, con el cual es posible extraer las cabeceras de los paquetes de cada paquete, y almacenarlas dentro de una cabecera principal. Esto puede proveer ventajas significantes para la recuperación de error si la cabecera principal puede ser

transmitida de un modo libre de error. Los experimentos muestran que las utilerías de protección de error provistas dentro del estándar pueden mejorar el funcionamiento, especialmente cuando éstos son combinados con otras técnicas de codificación de errores hacia atrás (FEC por sus siglas en inglés).

6.2.2 Descripción general del algoritmo de asignación de índice para JSCC

El objetivo de optimización de la codificación conjunta de fuente canal es encontrar un esquema de asignación de índice V para minimizar la distorsión esperada sujeta a un índice de bits designado. En el caso de la no existencia de ruido de canal, esto es equivalente a minimizar la distorsión contribuida de cada codeword mientras se guarda la suma de las longitudes del bitstream dentro del índice designado:

$$\min \sum_b D_b \quad \sum_b L_b \leq L_T, \quad 6.1$$

donde D_b y L_b son la distorsión y la longitud del bitstream del codeblock b respectivamente; L_T es el tamaño de archivo asignado.

Cuando se toman los efectos de la codificación de canal y el ruido dentro de la cuenta, para cada codeblock b , la distorsión esperada llega a ser:

$$E[D_b(V_b)] = D_{b,0} - E[\Delta D_b(V_b)], \quad 6.2$$

donde $D_{b,0}$ es la distorsión inicial, V_b es el esquema de asignación de índice para el codeblock b . $E[\Delta D_b(V_b)]$ es la reducción de distorsión esperada cuando el esquema de asignación de índice V_b es empleado, resultando un codeblock de longitud $L_b(V_b)$ en el bitstream.

Con la ecuación 6.2, en el caso de presencia de ruido de canal, la ecuación 6.1 puede ser escrita:

$$\min \sum_b (D_{b,0} - E[\Delta D_b(V_b)]) \quad \text{si} \quad \sum_b L_b(V_b) \leq L_T, \quad 6.3$$

o equivalentemente

$$\min \sum_b -E[\Delta D_b(V_b)] \quad \text{si} \quad \sum_b L_b(V_b) \leq L_T.$$

Adoptando los resultados de la minimización de Langrange [3], el problema puede ser resuelto por un método iterativo

$$\min \left\{ \sum_b -E[\Delta D_b(V_b)] + \lambda \sum_b L_b(V_b) \leq L_T \right\}, \quad 6.4$$

por la minimización de cada término independientemente. El barrido de λ sobre el rango de cero a infinito, dará conjuntos de $\{V_b\}$ y $\{\sum_b L_b\}$. Si durante este proceso se encuentra un $\{\sum_b L_b\}$ igual a L_T , entonces se encontrará la solución deseada.

6.2.3 Ejemplos de transmisión.

Caso 1.

En [1] para minimizar cada término en la ecuación 6.5 se realiza una tarea de optimización a nivel de los pasos de codificación. Así, para cada codeblock b , se denota a N_c como el número total de pasos de codificación de éste. Para cada paso de codificación i , hay una reducción de distorsión asociada Δd_i con longitud de l_i bytes, donde $i \in [1, N_c]$. Sin ninguna codificación de canal se tiene:

$$\Delta D_b = \sum_{i=1}^{N_c} \Delta d_i \quad L_b = \sum_{i=1}^{N_c} l_i$$

Para encontrar el esquema de asignación de índice que minimice 6.4. Se denota a r_i como el índice de codificación de canal para el paso de codificación i , donde $0 \leq r_i \leq 1$ y $P\left(r_i, \frac{l_i}{r_i}\right)$ como la probabilidad de que haya uno o más errores incorregibles en el paso de codificación i si el índice del canal es empleado. Finalmente, suponga N_c' como el número de pasos de codificación incluidos por V_b . Ya que un decodificador J2K puede decodificar todos los pasos de codificación correctos antes de que uno contenga el primer bit erróneo en un codeblock, la reducción de distorsión esperada para N_c' pasos de codificación usando índices de código de canal $r_1, r_2, \dots, r_{N_c'}$, es:

$$E[\Delta d] = \sum_{j=1}^{N_c'-1} \left(\sum_{k=1}^j d_k \right) \left(\prod_{k=1}^j \left[1 - P\left(r_k, \frac{l_k}{r_k}\right) \right] \right) P\left(r_{j+1}, \frac{l_{j+1}}{r_{j+1}}\right) + \left(\sum_{k=1}^{N_c'} d_k \right) \left(\prod_{k=1}^{N_c'} \left[1 - P\left(r_k, \frac{l_k}{r_k}\right) \right] \right). \quad 6.5$$

Con la ecuación 6.5, dada una λ , la ecuación 6.4 puede ser evaluada para cualquier esquema de asignación de índice $V_b = (r_1, r_2, \dots, r_{N_c'})$. Así r_i denota los índices de codificación de canal aplicados al paso de codificación i , N_c' denota el último paso de codificación incluido en el codeblock b ($N_c' \leq N_c$). Para reducir el espacio de búsqueda para una solución óptima se toman en cuenta algunas consideraciones como: que $r_i \leq r_{i+1}$ ($1 \leq r_i \leq N_c'$) ya que la importancia de los bits reduce a lo largo del bitstream y que para obtener un nivel de protección óptimo suficiente en un canal simétrico binario al menos se necesitan 3 índices de canal disponibles.

Resultados

En este trabajo se eligió un canal simétrico binario (BSC) con $\varepsilon \leq 10^{-2}$ como canal experimental, y se uso un código de canal RCPC. En este experimento se encontró que un código RCPC con índice 2/3 es lo suficientemente fuerte para producir una pequeña probabilidad de error en el decodificador para este canal, y se seleccionaron los índices de código (8/9, 4/5, 2/3) de [4]. Las imágenes de Lena y Bárbara (512x512) fueron utilizadas como imágenes de prueba. Las simulaciones fueron hechas a índices de bits totales de 0.1, 0.25, 0.5, 0.75 y 1 bpp con 1000 pruebas para cada uno. Los resultados se presentan en la tabla 6.1.

Índice de bits (bpp)	Lena	Bárbara
0.1	27.87	23.38
0.25	32.00	26.45
0.5	35.32	29.84
0.75	37.17	32.25
1	38.39	34.11

Tabla 6.1 Resultados de PSNR (en dB) de la transmisión de los codestreams JPEG2000 de Lena y Barbara con un código RCPC con índice de 2/3.

Caso 2

En este trabajo [2] las diferentes cadenas de código para las diferentes capas son completamente fijas y no existe dependencia entre otras capas. Por lo cual se aplica un código sistemático RS(N, K_l) sobre la l enésima capa del bitstream de B_l bytes de longitud. Se define (esquema de V_b) $\mathbf{K}=(K_1, \dots, K_L)$ como el vector de asignación de bits, y el índice de transmisión total es entonces dada por:

$$R(\mathbf{K}) = \sum_{l=1}^L R_l \frac{N}{K_l}. \quad 6.6$$

Cada vector \mathbf{K} corresponde a una estrategia del UEP. Para encontrar el UEP óptimo para un bitstream por capas dado con índice R_s , se utiliza el concepto de minimización de 6.4. En este caso la distorsión de reconstrucción esperada es dada por $D(\mathbf{K}, R_s)$, y esta sujeta por la limitante del índice de transmisión total R_0 . Por lo tanto el problema de codificación conjunta de fuente por capas y UEP puede ser formulado como: $\min_{R_s, \mathbf{K}} \{ \min_{\mathbf{K}} D(\mathbf{K}, R_s) \}$ sujeto a $R(\mathbf{K}) \leq R_0$ 6.7

Para resolver este problema de optimización se utiliza la técnica de Langrage, para obtener el mejor índice de codificación de índice R_s^* y su correspondiente asignación de índice $\mathbf{K}^*=(K_1^*, \dots, K_L^*)$ o UEP.

Aplicando un código RS(N, K_l) sobre la l enésima capa, éste es capaz de corregir $(N-K_l)/2$ errores, así cualquier byte en la l enésima capa puede ser recuperado correctamente en el decodificador RS con una probabilidad de:

$$p_l = \frac{1}{K_l} \sum_{s=0}^{K_l} \binom{K_l}{s} \varepsilon_b^{K_l-s} (1-\varepsilon_b)^s \times \sum_{c=0}^{N-K_l} \binom{N-K_l}{c} \varepsilon_b^{N-K_l-c} (1-\varepsilon_b)^c \times \begin{cases} K_l, & \text{si } s+c \geq (N-K_l)/2 \\ s, & \text{en otro caso} \end{cases}$$

donde el símbolo de probabilidad de error ε_b es dado por $\varepsilon_b = 1 - (1-\varepsilon)^8$ y ε es el índice de error de bits (BER) del BSC.

La reducción esperada en la distorsión después de la codificación de la l enésima capa es:

$$D(K_l) = (1-p_l) \sum_{n=1}^{B_l-1} p_l^n \left(D\left(\sum_{i=1}^{l-1} B_i\right) - D\left(\sum_{i=1}^{l-1} B_i + n\right) \right) + p_l^{B_l} \left(D\left(\sum_{n=1}^{l-1} B_i\right) - D\left(\sum_{i=1}^l B_i\right) \right)$$

y la distorsión de reconstrucción total esperada es:

$$D(\mathbf{K}) = D(K_1, \dots, K_L) = D_0 - \sum_{l=1}^L D(K_l), \quad 6.8$$

donde D_0 es la distorsión esperada cuando el índice total es cero. (Por simplicidad de notación se omite el R_s en $D(\mathbf{K}, R_s)$).

Si el modelo de una función de índice de distorsión operacional de fuente como $D(R) = D_0 2^{-2R}$, $D(K_l)$ puede ser explícitamente expresada como:

$$D(K_l) = (1 - p_l) \left(\frac{p_l - p_l^{B_l}}{1 - p_l} - \frac{\frac{p_l}{4} - \left(\frac{p_l}{4}\right)^{B_l}}{1 - \frac{p_l}{4}} \right) \times D_0 2^{-2 \sum_{i=1}^{l-1} B_i} + p_l^{B_l} D_0 \left(2^{-2 \sum_{i=1}^{l-1} B_i} - 2^{-2 \sum_{i=1}^l B_i} \right)$$

Alternativamente, se puede usar el modelo $1/R$ en [5] para la función de índice de distorsión.

Para minimizar la distorsión (ecuación 6.8) dada la limitante en el índice de transmisión (ecuación 6.6), por medio de Lagrange

$$J(\mathbf{K}) = D(\mathbf{K}) + R(\mathbf{K})$$

se utiliza un método iterativo para minimizar $J(K_1, \dots, K_L)$ sobre una variable de tiempo, mientras se guardan las otras variables constantes hasta converger.

Resultados

Se utilizaron las imágenes Lena y Goldhill de 512x512 y se asumió que $\varepsilon \leq 0.01$ para el BSC. Para este valor de ε se obtuvo $\mathbf{K}^* = (167, 171, 175, 177, 181, 183)$ para los índices 1, 0.5 y 0.25bpp para ambas imágenes. La tabla 6.2 muestra los resultados del PSNR (en dB) con este esquema de codificación. En cada caso, se realizaron 1000 pruebas.

R_0 (b/p)	Lena	Goldhill
1	38.83	34.82
0.5	35.73	31.83
0.25	32.56	29.51

Tabla 6.2 Resultados de PSNR (en dB) de la transmisión de los codestreams JPEG2000 de Lena y Goldhill con un código RS.

Se puede observar que en ambos ejemplos los resultados obtenidos son muy semejantes, esto se debe a que ambos están basados en una codificación conjunta de fuente canal, y solo difieren en el código que utilizan para la protección única de error. La ventaja de uno a otro simplemente será la facilidad de implementación y decodificación del UEP en el sistema de transmisión.

Referencias

- [1] Z. Wu, A. Bilgin y M.W. Marcellin, “Unequal error protection for transmission of JPEG2000 codestreams over noise channels”, IEEE ICIP 2002
- [2] T. Chuu, Z. Liu, Z. Xiong y X. Wu, “Joint UEP and layered source coding with application to transmission of JPEG-2000 Coded images”, IEEE 2001
- [3] Y. Shoham and A. Gersho, “Efficient bit allocation for an arbitrary set of quantizers,” IEEE Trans. Acous-tics, Speech, and Signal Proc., vol. 36, pp. 1445-1453, September 1988.
- [4] J. Hagenauer, “Rate-compatible punctured convolutional for noisy channels”, IEEE Signal Processing Letters, vol. 4, pp. 189-191, July 1997.
- [5] S. Mallat and F. Falzon, “Analysis of low bit rate image transform coding,” IEEE Trans. Signal Proc., vol. 46, pp. 1027-1042, April 1998.
- [6] D.S. Taubman y M. W. Marcellin, “JPEG2000 Image Compression Fundamentals Standards and Practice”, Kluwer Academic Publishers, London, 2002
- [7] Kakadu V2.2 software

Capítulo 7

Otras aplicaciones.

Debido al aumento del Internet y de los productos de multimedia inalámbricos, la mayoría de los trabajos se han dedicado recientemente al diseño de sistemas eficientes para la transmisión de imágenes y vídeo sobre canales ruidosos. En este capítulo se describirá a grandes rasgos la transmisión progresiva de imágenes y de video.

7.1 Transmisión progresiva de imágenes.

El uso de los esquemas de codificación de imágenes para transmisiones progresivas es principalmente concernido a canales de ancho de banda pequeño tal como las líneas telefónicas. El principal objetivo es habilitar al receptor para obtener un borrador de reconocimiento rápido del contenido de una imagen entera donde después se puede decidir si se aborta o se continúa la transmisión del resto de la imagen. Las aplicaciones típicas de la transmisión progresiva son las llamadas telebúsquedas y telecompras. En ambos casos el contenido de la base de datos de la imagen almacenada remotamente es escaneada y así el receptor que desea tener una imagen reconocible tan pronto como le sea posible pueda cancelar la transmisión de las imágenes no deseadas inmediatamente.

Diferentes esquemas de transmisión progresiva han sido descritos en la literatura. En [1], Hofmann y Troxel emplean el método de submuestreo. Ellos usan un índice de submuestreo de 14:1 en ambas direcciones, produciendo una imagen reconocible en muchos casos. Sin embargo para imágenes con mayor información de alta frecuencia, se exige usar un índice de submuestreo de 8:1 como máximo, para poder reconocer la versión submuestreada de la imagen. Por supuesto, la complejidad de los métodos de submuestreo es mínima. Los problemas de reconocimiento que ocurren usando los métodos de submuestreo son discernidos por Knowlton en [2]. Él propone un método que esta basado sobre el promedio de cada dos píxeles vecinos. Como resultado menos bits son necesitados para el reconocimiento de imágenes cuando se compara con [1]. Una comparación directa con otras técnicas de transmisión progresiva es difícil, ya que los ejemplos mostrados son solamente para imágenes originales de 4 bits/píxel y prácticamente se usan en dimensiones de imágenes no cuadradas. Otro método basado sobre el promedio que no tiene esas desventajas es descrito por Tanimoto en [3]. La cantidad de gastos generales puede variar de un 5% hasta un 33%. Las técnicas que se describen en [1,2,3] producen una reconstrucción perfecta, mientras no se aplique una compresión de datos.

Sistemas más complejos para una transmisión progresiva son basados sobre métodos de transformadas de bloque. Takikawa [4] muestra métodos para rápidas reconstrucciones progresivas de imágenes que son transformadas por bloque usando la transformada de Fourier o de Hadamard, ambas con un tamaño de bloque de 16x16. Esta parece que usa al menos 16 de los 256 coeficientes de transformada para producir una imagen reconocible. Estos resultados son mostrados sin la codificación de los

coeficientes. En [5] una Transformada Coseno Discreta (DCT) es usada para la transmisión progresiva de imágenes. El documento meramente trata con el problema del orden de transmisión de coeficientes DCT, considerando un esquema adaptativo y cuatro esquemas fijos para el orden de transmisión. Los ejemplos son mostrados para dos imágenes diferentes de 128x128 con 8 bits/píxel, pero de nuevo para coeficientes sin codificar. Otro documento sobre transmisión progresiva fue publicado por Wang y Goldberg [6]. El método está basado sobre una cuantización vectorial y por lo tanto incorpora la compresión de datos. La aplicación de un codificador de entropía para la imagen de error final en el último estado, todavía, permite la codificación de esquemas sin pérdidas. El primer estado produce ya imágenes reconocibles a índices de bits de aproximadamente 0.06bit/píxel para imágenes de 256x256.

En [7] Westerink, Biemond y Boeke proponen un esquema de transmisión progresiva que está basado sobre el concepto de codificación de subbanda (SBC), en el cual se usan las subbandas codificadas, estableciéndose la compresión de datos. La superioridad de las modernas técnicas de compresión wavelet supera la compresión de bloques. Además las wavelet proveen una representación de multiresolución. Las aproximaciones de imágenes a escalas finas o gruesas (o componentes de baja y alta frecuencias) pueden ser naturalmente sintetizadas en el dominio wavelet. Esta propiedad, conocida como propiedad de empotramiento (sucesiva refinación) o escalabilidad en el índice de bits, es muy atractiva para la transmisión progresiva y almacenamiento de imágenes.

Algunos modos de progresión o escalabilidad en el índice de bits han sido incluidos en diversos estándares de imágenes y de video, tal como el JPEG, JPEG-2000, MPEG-2 y MPEG-4. Pero los bitstreams progresivos sufren de una alta sensibilidad a los errores de canal debido a la propagación de error en el receptor. Por lo que la codificación conjunta de fuente canal ha emergido como una forma efectiva de direccionar este problema y por lo tanto este principio ha sido una aplicación exitosa para la transmisión progresiva de imágenes.

7.2 Transmisión de video

La transmisión de la información de video sobre medios inalámbricos es ahora un área madura, en la cual un rango de servicios interactivos y de distribución tal como el videoteléfono, la videoconferencia, el video en demanda, etc., son ofrecidos por varios proveedores a través del mundo. La mayoría de los servicios adoptaron la primera generación de estándares de codificación de video, tal como, H.261 para videoconferencia de 64kbits/s a 2Mbits/s, H.263 para videoteléfono por lo menos a 64kbits/s, y los estándares del Grupo de Expertos en Figuras en Movimiento (MPEG-1 o 2) a índices de bits arriba de 100Mbits/s. En la escena de la distribución inalámbrica de video, un rango de servicios de transmisión broadcast aparece desde hace pocos años para aplicaciones estacionarias. En Europa, los esquemas de distribución broadcast para video digital (DVB por sus siglas en inglés) emplean MPEG-2 para la compresión de video y la

corrección de errores hacia delante (FEC) Reed-Solomon (RS) como protección contra los errores de canal.

El funcionamiento del video se caracteriza por su índice de bits. El rango de formatos de video estándar es resumido en la tabla 7.1, junto con sus índices de bits descomprimidos a índices de escaneo de cuadros de 10 y 30 cuadros/seg para ambas señales de color y en escala de grises.

Formato de video	Dimensiones de luminancia	No de pels por cuadro	Sin compresión			
			Índice de bits (Mbit/s)			
			10 cuadros/s		30 cuadros/s	
			Escala de grises	Color	Escala de grises	Color
SQCIF	128x128	12288	0.983	1.47	2.95	4.42
QCIF	176x176	25344	2.03	3.04	6.09	9.12
CIF	352x288	101376	8.1	12.2	24.3	36.5
4CIF	704x576	405504	32.4	48.7	97.3	146.0
16CIF	1408x1152	1622016	129.8	194.6	389.3	583.9
CCIR 601	720x480	345600	27.65	41.472	82.944	124.416
HDTV 1440	1440x960	1382400	110.592	165.888	331.776	497.664
HDTV	1920x1080	2073600	165.9	248.832	497.664	746.496

SQCIF: Sub-Quarter Common Intermediate Format
 QCIF: Quarter Common Intermediate Format
 CIF: Common Intermediate Format
 HDTV: High Definition Television

Tabla 7.1 Varios formatos de video y sus índices de bits sin compresión.

A continuación se muestran las características principales de algunos estándares de codificación de video, usados actualmente en los sistemas de telecomunicaciones.

Estándar de Codificación de Video H.261

Introducido en 1990 y revisado en 1993 el estándar de codificación H.261 [9] del ITU (Internacional Telecommunication Union) fue diseñado para la transmisión de video sobre canales ISDN (Integrated Services Digital Network). Un algoritmo de codificación híbrida es empleado y la predicción inter-frame es usada para reducir la redundancia temporal inter-frame, mientras que la codificación por transformada es usada para reducir la redundancia espacial. La información de traslación de movimiento es extraída por la compensación de movimiento representada por vectores de movimiento (MV por sus siglas en inglés). Este sistema fue diseñado para índices de transmisión de bits en el rango de 40Kbps a 2Mbps aproximadamente.

Estándar de Codificación de Video H.263

El grupo de expertos de la ITU-T para el video telefónico a muy bajas tasas de transmisión (LBC) produjo un conjunto de recomendaciones para video telefónico sobre PSTN (Public Switched Telephony Network) apuntado a tasas de transmisión de 64 Kbits/s y más bajas. Este estándar es el diseño de recomendaciones de la ITU-T. H.263 formalmente aceptado en 1996 [10].

El estándar del ITU-T H.263 es similar a la recomendación H.261, ya que se basó y complemento la recomendación del H.261. Las principales mejoras en el algoritmo de codificación del H.261 en el H.263 son: el uso de mejores tablas para la codificación de longitud variable que fueron optimizadas para diferentes modos, la predicción de medio píxel para la compensación de movimiento y un vector de movimiento individual por cada bloque de 8x8; adicionando el tamaño de cuadro SQCIF. Además el H.263 incluye cuatro opciones de codificación, que son: vectores de movimiento sin restricción, sintaxis basada en codificación algorítmica, predicción avanzada y cuadros predichos y bidireccionales (P-B). Otros cambios se encuentran en la sintaxis del bitstream del H.263.

Grupo de Expertos en Imágenes en Movimiento (MPEG)

El grupo de expertos de imágenes en movimiento (MPEG) fue formado por la Organización Internacional de Estándares (ISO) para desarrollar un conjunto de estándares para compresión de vídeo digital en concordancia con las siete capas del modelo OSI. El primer intento de MPEG fue para suplir la compresión de datos a velocidades de 1.5 Mbit/s para imágenes almacenadas, llamado MPEG-1. La segunda parte llamada MPEG 2, fue diseñada para generar velocidades de pixel entre 5 y 10 Mbit/s, para videos de mejor calidad en CATV y HDTV, entre otros. MPEG-4 fue lanzado en 1998 para explorar los requerimientos de un conjunto de aplicaciones multimedia más extensas que requerían de interactividad.

MPEG-1

MPEG-1 [11] está diseñado para realizar la codificación y transmisión de datos de audio y video a tasas de transmisión alrededor de los 1.5Mbit/s; de los cuales 1.15 Mbits/s son para el video y los 350 Kbits/s restantes son para el sonido (estéreo) y para datos auxiliares. El estándar consta de 5 partes las cuales son: sistemas, video, audio, pruebas de conformidad y simulación de software.

La parte 2 fue desarrollada principalmente para operar en los medios de comunicación de almacenamiento (como videos CD y contenido de video en laserdisc) ofreciendo tasas de transferencias cerca de los 1.5Mbit/s. Sin embargo este estándar puede ser usado más ampliamente que lo mencionado anteriormente, porque el enfoque que toma es genérico.

El codec de video de MPEG-1 consiste en gran parte de secciones del codec de video H.261 y usa los mismos principios de codificación de video tales como métodos de redundancia temporal y espacial. Cuatro tipos de imágenes son definidos por el estándar MPEG los cuales son las imágenes intra (I), predecible (P) e interpolada o bidireccional (B).

MPEG-2

MPEG-2 [12] es un algoritmo de codificación genérico definido conjuntamente por el MPEG y el grupo de expertos para codificación de video ATM. Este algoritmo es dirigido a muy vastas aplicaciones con alcance de ancho de banda de 2 a 15Mbps. La escalabilidad de MPEG-2 en términos de bit rates, resoluciones, niveles de calidad y

servicios le permite ser usado en televisión por broadcast, TV por cable, cinemas electrónicos, video streaming, multimedia, video comunicaciones, y aplicaciones de computación gráfica.

El estándar MPEG-2 consiste de 9 partes, las tres primeras partes han alcanzado el estatus de estándar internacional, las otras partes son los diferentes niveles de consumación. Actualmente se desarrolla la 10ª parte.

Los formatos de video que pueden ser codificados por MPEG-2 incluyen el formato SIF de MPEG-1, tan bien como formatos adicionales como 4:2:0, 4:2:2 y 4:4:4. Los términos 4:2:2 y 4:2:0 son frecuentemente usados para describir la estructura de muestreo de una imagen digital. 4:2:2 significa que la crominancia es horizontalmente sub-muestreada por un factor de 2 relativo a la luminancia. 4:2:0 significa que la crominancia es horizontalmente y verticalmente sub-muestreada por un factor de 2 relativo a la luminancia.

MPEG-2 también incrementa el rango de diferentes calidades y servicios de bit rate. Esto es posible a través de la escalabilidad. En MPEG-2 la escalabilidad puede tomar la forma de técnica: espacial, temporal, SNR, partición de datos e híbrida. Estos modos rompen el vídeo MPEG 2 en diferentes capas (base, media, y alta) para propósitos de prioritarización de datos de vídeo.

El algoritmo de codificación usado en MPEG-2 difiere un poco del usado en MPEG-1. Este consiste principalmente de la predicción interframe de compensación de movimiento, usando ambas predicciones hacia delante y hacia atrás. Adicionalmente hay diversas técnicas que han sido incluidas para la codificación de video intercalado. Esas técnicas son: una DCT adaptable campo/frame, predicción de compensación de movimiento adaptable campo/frame y predicción espacial.

Para que MPEG-2 maneje varios tipos de aplicaciones que van desde HDTV (anchos de banda grandes) a videoteléfono (anchos de banda pequeños), se deben introducir medidas para definir el tipo de aplicación y promover la compatibilidad entre las diferentes aplicaciones. Estas medidas introducidas fueron definidas como perfiles y niveles.

MPEG-4

MPEG 4 designado ISO/IEC 14496 [13], se finalizó en octubre de 1998 y fue un estándar internacional en los primeros meses de 1999. La versión 2 fue concretizada hasta finales de 1999. Actualmente hay 5 versiones del sistema MPEG-4, llamadas enmiendas por la ISO. MPEG-4 fue sugerido por una necesidad de un nuevo estándar con la convergencia de tres tecnologías separadas: el campo del entretenimiento y comunicaciones (por ejemplo la televisión digital), las aplicaciones interactivas gráficas (p.e. las animaciones) y las aplicaciones de multimedia interactiva (p.e world wide web)

A diferencia de los anteriores estándares de video MPEG-4 adopta una diferente propuesta basada en el contenido. MPEG-4 esta dirigido a las siguientes funcionalidades basadas en el contenido:

- Interactividad basada en el contenido.
 - Utilerías de acceso de datos multimedia basadas en el contenido
 - Manipulación basada en el contenido y edición del bitstream
 - Codificación de datos híbridos, naturales y sintéticos
 - Mejoras del acceso aleatorio temporal
- Compresión
 - Mejora la eficiencia de codificación
 - Codificación de múltiples streams de datos concurrentes
- Acceso universal
 - Robustez en ambientes propensos a errores
 - Escalabilidad basada en el contenido.

MPEG-4 logra estos objetivos proveyendo formas estandarizadas para:

1. representación de unidades de contenido auditivo, visual o audiovisual, llamadas objetos audiovisuales (AVO's por sus siglas en inglés), las cuales pueden ser de origen sintético o natural.
2. descripción de las composiciones de esos objetos para crear una mezcla de AVOs que forman escenas audiovisuales.
3. multiplexación y sincronización de los datos asociados a los AVOs, para que puedan ser transportados sobre canales de red proveyendo una QoS (Quality of Service) apropiada para AVOs específicos; e
4. interacción con las escenas audiovisuales generadas en el receptor final.

Una escena audiovisual está compuesta de AVOs, los cuales pueden ser clasificados dentro de AVOs primitivos y compuestos. Un AVO primitivo puede ser una persona o una voz y un AVO compuesto puede ser una persona hablando. MPEG-4 define la representación codificada de los AVOs como natural o sintética, de 2- o 3-dimensiones. Para satisfacer estas funcionalidades los algoritmos de compresión deben ser más eficientes que los estándares actuales y la representación codificada permite que los AVOs sean manipulados y accedados independientemente e interactivamente.

Las nociones de los objetos audiovisuales y la descripción de la escena representan las ideas principales del estándar MPEG-4 y son completamente nuevas en los sistemas MPEG.

Referencias

- [1] W.D. Hofmann y D.E. Troxel, "Making Progressive Transmission Adaptive", IEEE Trans. on Communications, Vol. COM-34, No. 8, August 1986, pp 806-813.
- [2] K. Knowlton, "Progressive Transmission of Grey-Scale and Binary Pictures by Simple, Efficient and Lossless Encoding Schemes", Proc. IEEE, vol. 68, No. 7, July 1980, pp. 885-896.

-
- [3] S.L. Tanimoto, “Image Transmission whit Gross Information First”, Computer Graphics and Image Processing, Vol. 9, No. 1, January 1979, pp.72-76.
- [4] K. Takikawa, “Fast Progressive Reconstruction of a Transformed Image”, IEEE Trans. on Information Theory, vol. IT-30, No. 1, January 1984, pp. 111-117.
- [5] K.N. Ngan, “Image Display Techniques Using the Cosine Transform”, IEEE Trans. on Acoust., Speech and Signal Proc., Vol. ASSP-32, No. 1, February 1984, pp. 173-177.
- [6] L. Wang and M. Goldberg, “Lossless Progressive Image Transmission by Residual Error Vector Quantization”, IEE Proceeding, Vol. 135, No. 5, October 1988, pp. 421-430.
- [7] Westerink, J. Biemond y D.E. Boekee, “Progressive Transmission of Images Using Subband Coding”, IEEE, 1989.
- [8] K. Chieh y C.J. Kuo, “Retrieval and Progressive Transmission of Wavelet Compressed Images” IEEE International Symposium on Circ. and Systems, June 1997.
- [9] ITU-T, “H.261: Video Codec for audio visual service at $px64$ kb/s,” March 1993.
- [10] ITU-T, “H.263: Video Coding for Low Bitrate Communication,” May 1996.
- [11] ISO/IEC/JTC1/SC29, “Coded representation of picture, audio, and multimedia/ hypermedia information”, November 1992
- [12] Leonardo Chiariglione – Convenor, “Short MPEG-2 description” en ISO/IEC JTC1/SC29/WG11, “Generic coding of moving pictures and associated audio information”, October 2000.
- [13] ISO/IEC 14496-2, “Information technology –generic coding of audio-visual objects (final draft of international standard)”, en ISO/IEC/JTC1/SC29/WG11 N2502, October 1998.

Conclusiones

A lo largo del desarrollo de la presente tesis se tuvo la finalidad de introducir los conceptos más básicos que intervienen en una transmisión ya sea de datos, voz o multimedia; pero en especial de imágenes.

El enfoque que se tomó para la realización de la transmisión de imágenes fue aquella que se efectúa por medio de una codificación conjunta fuente canal. Sobre este aspecto, se puede concluir que los resultados obtenidos al transmitir imágenes bajo este esquema son eficaces. Además, se tiene la certeza de que cuando los parámetros del canal por alguna razón no explicada cambien (es decir, se tengan diferentes capacidades de canal); la información transmitida será más robusta (menos errores) a estos cambios y se asegura la distorsión calculada, cuando se realiza la codificación.

En este esquema la elección del codificador de fuente y de canal es también importante, en este trabajo se eligió el estándar JPEG2000 ya que las técnicas utilizadas durante la codificación de imágenes y el algoritmo de codificación de bloque incrustado con un truncamiento optimizado de bitstreams (EBCOT por sus siglas en inglés) facilitan el empleo de una codificación conjunta de fuente canal. En cuanto al codificador de canal se puede observar que los dos códigos correctores de errores mostrados en este trabajo presentan pequeñas diferencias en las medidas de distorsión obtenidas por cada uno. Por lo que se puede concluir que la elección entre los dos códigos correctores de errores, esencialmente dependerá de la complejidad de su implementación.

En cuanto al estándar JPEG 2000 se puede concluir, que es un estándar destinado a sustituir a su antecesor JPEG debido a su alto rendimiento a bajas tasas de bits. Las características de este estándar permiten su adaptación a diferentes necesidades de canales de transmisión, es decir que se puede obtener de una imagen codificada diferentes tipos de imágenes en el decodificador. Estas imágenes variarán en calidad, resolución y componentes de color; además de que se tiene la opción de decodificar solamente una región espacial dentro de la imagen codificada inicialmente. Hay que aclarar que para tener estas ventajas, cuando se realiza la codificación se tienen que especificar los parámetros que permitan estas permutaciones en la decodificación.

Actualmente a pesar de las ventajas que ofrece el estándar JPEG2000 aun su uso no se ha extendido en las diversas aplicaciones que demandan compresión y calidad en las imágenes. En cuanto a la codificación conjunta de fuente canal no se recomienda cuando la información se va a transmitir sobre varios canales con diferentes características. Ya que la fuente puede ser recodificada en los nodos de transmisión para concordar con los diferentes canales. Esto significa que los nodos deben saber sobre la codificación de fuente para diferentes tipos de fuentes.

Finalmente, se puede decir que los objetivos de esta tesis se cumplieron. Para un estudio futuro se puede mencionar que en la transmisión de imágenes fijas existen más

parámetros a estudiar a los vistos en esta tesis y al mismo tiempo dentro del esquema de transmisión de datos multimedia existe la transmisión de imágenes progresivas y de video.