

01170

**UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO**  
**DIVISION DE ESTUDIOS DE POSGRADO**  
**FACULTAD DE INGENIERIA**

**CODIFICACION DE IMAGENES DIGITALES  
UTILIZANDO UNA ESTRUCTURA DE DATOS  
MULTI-RESOLUCION Y  
CUANTIZACION VECTORIAL**

**TESIS**

**QUE PARA OBTENER EL GRADO DE  
MAESTRO EN INGENIERIA ELECTRICA  
(COMUNICACIONES)**

**PRESENTA**

**RUBISEL TOVILLA HEREDIA**

**DIRECTOR DE TESIS:**

**DR. FRANCISCO GARCIA UGALDE**

**MEXICO, D. F.**

**TESIS CON  
FALLA DE ORIGEN**

1997



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

**A mis padres:**

**Jesús Tovilla Cabrales.**

**Rafaela Heredia de Tovilla.**

**A Daniel Arturo.**

## **A G R A D E C I M I E N T O S**

Deseo expresar mi agradecimiento y respeto a todas aquellas personas que dedicaron tiempo, interés y esfuerzo para la realización de esta tesis.

A la Universidad Nacional Autónoma de México.

A la División de Estudios de Posgrado de la Facultad de Ingeniería, por mi formación.

A mi director de tesis **Dr. Francisco García Ugalde** por su motivación, apoyo y paciencia.

A mis sinodales, por el tiempo invertido en la revisión de este documento.

A los profesores de la DEPFI.

Al Consejo Nacional de Ciencia y Tecnología.

# Resumen

La codificación de imágenes forma parte del procesamiento digital y consiste en una representación eficiente de los datos, mientras se mantiene la información esencial contenida en ellos.

La compresión de imágenes tiene dos consecuencias directas. La primera de ellas es la reducción del tiempo de transmisión y del ancho de banda requerido del canal en un sistema de transmisión de imágenes: televisión digital, videoconferencia y facsímil. La segunda, es la reducción de la capacidad de almacenamiento necesario para conservar la información.

La descomposición piramidal en múltiples resoluciones, es un esquema que combina las técnicas de codificación por transformada, con la codificación predictiva e interpolativa/extrapolativa, para eliminar la correlación entre los pixels de una imagen. El hecho de que esta representación se asemeje al sistema de visión humano, constituye una cualidad muy interesante. Además, permite realzar los rasgos característicos de la imagen, lo que la hace adecuada para muchas tareas de análisis y compresión. Otra característica atractiva de esta técnica de codificación híbrida, es su adecuada capacidad para la transmisión progresiva.

En esta tesis se analiza e implanta un algoritmo de codificación de imágenes digitales que combina la representación multiescala con la cuantización vectorial, con el objetivo de alcanzar mayores tasas de compresión y al mismo tiempo conservar una buena calidad subjetiva de los datos reconstruidos.

Los resultados que se obtienen de la implantación, demuestran que como una consecuencia de la representación piramidal en múltiples resoluciones de las imágenes, éstas se pueden codificar a tasas de bit menores que la entropía de la imagen original para la transmisión sin pérdidas. Además, como un esquema de codificación con pérdidas, se alcanza una capacidad de transmisión cercana a 0.55 bits/pixel con una calidad de reconstrucción excelente.

# Indice

<b>1</b>	<b>INTRODUCCION</b>	<b>3</b>
1.1	Clasificación de las técnicas de codificación de imágenes. . . . .	4
	Modulación por pulsos codificados PCM . . . . .	5
	Codificación predictiva DPCM . . . . .	5
	Codificación por transformada . . . . .	6
	Codificación interpolativa/extrapolativa . . . . .	6
	Cuantización vectorial . . . . .	7
1.2	El codificador de dos canales. . . . .	7
1.3	Codificación piramidal. . . . .	8
<b>2</b>	<b>EL SISTEMA DE CODIFICACION</b>	<b>10</b>
2.1	Cuantización Vectorial. . . . .	11
2.2	Medidas del desempeño de un Cuantizador Vectorial . . . . .	13
2.3	Diseño del libro de códigos . . . . .	15
	2.3.1 Condiciones de optimabilidad para un cuantizador vectorial . . . . .	15
	2.3.2 El algoritmo LBG . . . . .	16
	2.3.3 Selección del libro de códigos inicial . . . . .	16
2.4	Algoritmos rápidos para la cuantización vectorial, usando la técnica del vecino más cercano . . . . .	17
2.5	El codificador de Huffman, construcción y restricciones para el diseño . . . . .	21
2.6	Aplicaciones del código de Huffman en la codificación de imágenes de error . . . . .	24
<b>3</b>	<b>DESCOMPOSICION EN MULTIPLES RESOLUCIONES</b>	<b>25</b>
3.1	Procesamiento de señales a múltiples tasas . . . . .	26
	3.1.1 Decimación e interpolación . . . . .	26
3.2	Transmisión progresiva . . . . .	28
3.3	Codificación de dos canales . . . . .	28
3.4	Estructura de árbol subbanda de banda diádica . . . . .	29
3.5	El algoritmo de Burt y Adelson . . . . .	30
3.6	Características de la pirámide Gaussiana . . . . .	34
3.7	Características de la pirámide Laplaciana . . . . .	34

3.8	Reconstrucción de la imagen original . . . . .	35
<b>4</b>	<b>EL ESQUEMA DE CODIFICACION UTILIZADO</b>	<b>39</b>
4.1	Descomposición piramidal de imágenes . . . . .	41
4.1.1	Formación de las pirámides Gaussiana modificada y Laplaciana . . . . .	42
4.2	Recuperación de la imagen original . . . . .	44
4.3	Cuantización vectorial de imágenes representadas mediante una estructura piramidal . . . . .	45
4.4	Transmisión progresiva . . . . .	46
4.5	Codificación del error residual . . . . .	48
<b>5</b>	<b>RESULTADOS</b>	<b>50</b>
5.1	Presentación de los resultados . . . . .	52
5.1.1	Lena256 . . . . .	53
5.1.2	Miss América256 . . . . .	54
5.2	Análisis de los resultados . . . . .	55
<b>6</b>	<b>CONCLUSIONES</b>	<b>65</b>
	<b>BIBLIOGRAFIA</b>	<b>67</b>

# Capítulo 1

## INTRODUCCION

La compresión de los datos presentes en las imágenes ha sido y continúa siendo una de las principales áreas de investigación en el procesamiento digital de señales. El principal objetivo de la compresión de imágenes, que constituye una ciencia/arte de la codificación eficiente de los pixels, es representar una imagen con la menor cantidad de bits posibles, explotando la redundancia presente entre los pixels, mientras se preserva el nivel de calidad e inteligibilidad requerido para una aplicación dada.

La compresión de imágenes tiene dos consecuencias directas. La primera de ellas se enfoca a la reducción del ancho de banda requerido para el canal en un sistema de transmisión de imágenes, como los que se emplean en televisión digital, videoconferencia y facsímil. La segunda es la reducción de la capacidad de almacenamiento necesario para guardar las imágenes.

Con la finalidad de ubicar el objetivo de esta tesis, se muestra en la figura 1.1 un ambiente típico que rodea a un sistema de codificación de imágenes.

En este esquema, la imagen digital de entrada al transmisor es codificada por un codificador de fuente o de imágenes, cuya salida es una cadena de bits, los cuales representan la imagen de entrada. El codificador de canal, transforma la cadena de bits en una forma adecuada para hacer posible la transmisión a través de un canal de comunicaciones, empleando alguna forma de modulación. Esta señal modulada se transmite por un canal de comunicaciones, el cual generalmente introduce algún tipo de ruido. Esta situación se corrige incluyendo algún tipo de código de corrección de errores en la etapa del codificador de canal.

En el receptor, el decodificador de canal tiene la tarea de convertir la señal modulada en una cadena de bits. El decodificador de fuente reconstruye la imagen a partir de esta cadena de bits, la cual se despliega para su visualización, o bien se almacena en una memoria. Cabe señalar que para el caso del procesamiento local, no es necesario incluir en el esquema el codificador y decodificador de canal, así como tampoco el canal mismo.

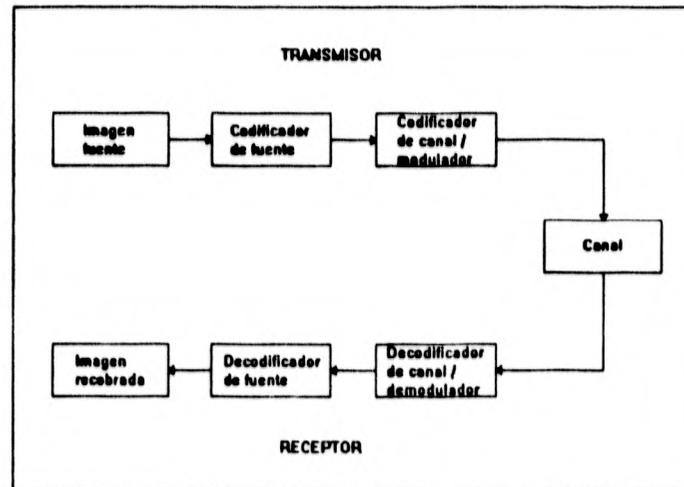


Figura 1.1: Ambiente típico de un codificador de imágenes.

## 1.1 Clasificación de las técnicas de codificación de imágenes.

Existe una gran variedad de métodos propuestos para compresión de imágenes, pero ellos pueden clasificarse en dos grupos fundamentales: sin pérdidas y con pérdidas.

Aquellas técnicas de codificación que permiten una reconstrucción numéricamente idéntica, pixel a pixel, de la imagen original después de la compresión, se dice que son preservadoras de la información, reversibles o sin pérdidas. Por otro lado, las técnicas de compresión con pérdidas o irreversibles, son aquellas en las que la imagen reconstruida presenta degradaciones con respecto a la original. Esto da como resultado mayores tasas de compresión en el sistema. En general, en cuanto mayor sea la calidad y la inteligibilidad que se pueda suprimir, menor será la tasa de bit requerida para codificar una imagen.

Como se mencionó anteriormente, los esquemas de codificación pueden ser sin pérdidas o con pérdidas. A la primera categoría pertenecen los codificadores estadísticos, como el codificador aritmético y el codificador de Huffman, el cual es un código eficiente de longitud variable, libre de prefijos que indiquen el inicio y el final de cada palabra, donde la asignación de bits depende de la probabilidad de ocurrencia de los símbolos de la fuente. Las técnicas de codificación sin pérdidas se usan generalmente para la asignación de bits a las salidas cuantizadas de las señales codificadas, ya sea por un esquema DPCM o bien por transformadas, con el fin de minimizar la tasa promedio de bits.

Por otra parte, los esquemas de codificación con pérdidas pueden subdividirse

en cinco clases diferentes, como se muestra en la figura 1.2.

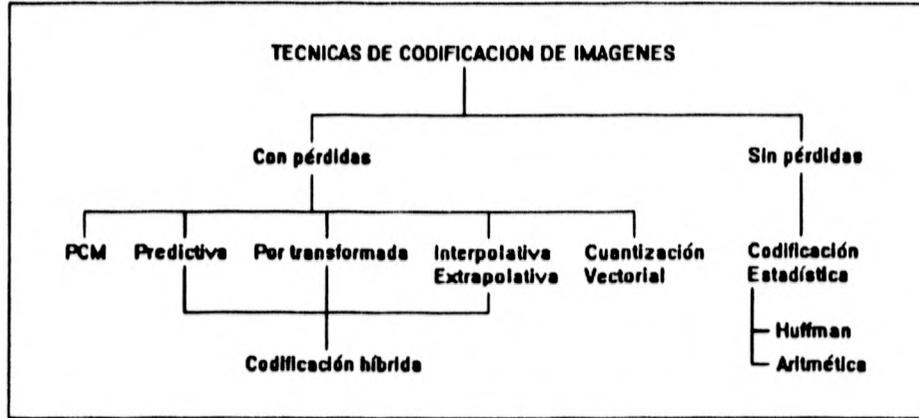


Figura 1.2: Clasificación de los algoritmos de codificación de imágenes.

**La modulación por pulsos codificados (PCM)**, es una representación tiempo discreto-amplitud discreta de los pixels, en la cual se elimina la menor cantidad de redundancia estadística o perceptual de la señal original. Como se muestra en la figura 1.3, la discretización temporal se obtiene muestreando la señal original a la tasa de Nyquist, mientras que la discretización de la amplitud se obtiene usando un número de niveles tal, que la degradación introducida sea tolerable. Un sistema PCM básico, se ocupa frecuentemente en la mayoría de las aplicaciones del procesamiento digital, para obtener una imagen digitalizada "original" a partir de una imagen analógica.

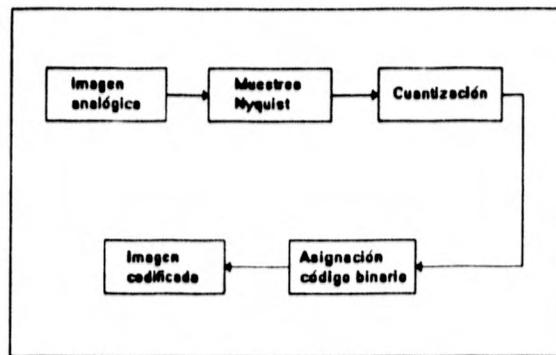


Figura 1.3: Las etapas de un codificador PCM.

**En la codificación predictiva o DPCM**, se predice el valor del pixel actual que se va a codificar, empleando los pixels codificados anteriormente, con lo que

se obtiene el error de predicción entre el pixel actual y su estimación. Este error se cuantiza, usando un codificador PCM y se transmite o se almacena. Con esta operación se intenta eliminar la redundancia existente entre los pixels adyacentes de la imagen. En la figura 1.4 se muestra el diagrama de bloques de un codificador y un decodificador predictivos.

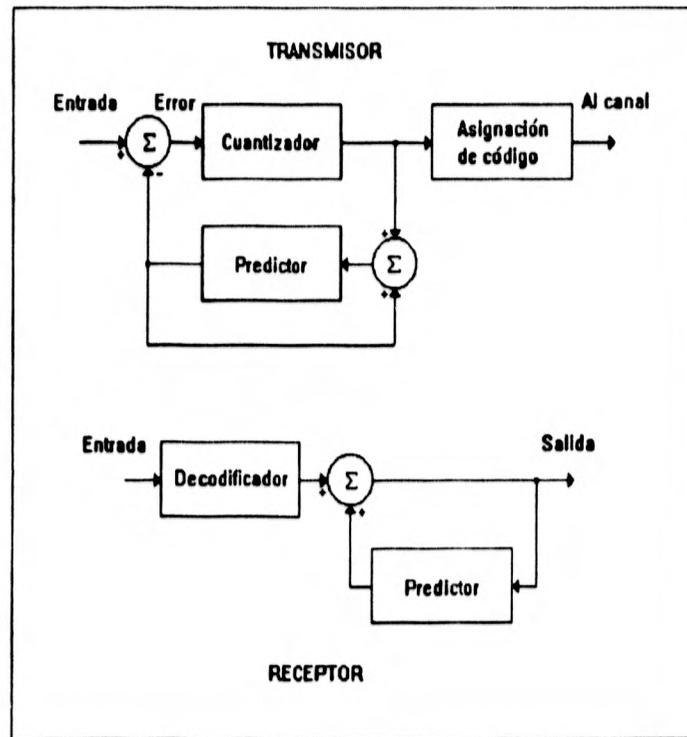


Figura 1.4: Diagrama a bloques de un codificador y un decodificador predictivos.

En la **codificación por transformada**, la imagen de entrada se adecúa, por medio de una transformación lineal, a un dominio diferente al de la intensidad de los pixels, con el fin de concentrar la energía de la imagen sobre unos cuantos coeficientes transformados. Estos coeficientes obtenidos de la transformación se cuantizan para su transmisión o almacenamiento. Las técnicas por transformada han demostrado una gran capacidad en aplicaciones a tasas de bit bajas (por debajo de 1 ó 2 bits/pixel), como las usadas en videoconferencias. Sin embargo, tienen el inconveniente de ser computacionalmente más complejas.

Las **técnicas de codificación interpolativa/extrapolativa**, trabajan bajo un principio diferente a las anteriores. Estas técnicas transmiten un subconjunto de pixels de la imagen al receptor, el cual hace una interpolación o una extrapo-

lación para obtener los pixels no transmitidos. Este tipo de algoritmos se utiliza principalmente en sistemas entre tramas, combinados con codificación predictiva.

**La cuantización vectorial** consiste en un mapeo de un espacio Euclideo de dimensión  $k$ ,  $R^k$ , en un subconjunto finito  $Y$  de  $R^k$ . Esto se puede expresar simbólicamente como:

$$Q : R^k \mapsto Y; \quad Y \subset R^k; \quad Y = \{x_i | i = 1, 2, \dots, N\} \quad (1.1)$$

El principal objetivo de la cuantización vectorial, es encontrar un conjunto de  $N$  vectores, o libro de códigos ( $Y$ ), que minimice la distorsión esperada. En esta técnica, primero se descompone la imagen en vectores de dimensión  $k$ , cada uno de los cuales se representará usando uno de los vectores existentes en el libro de códigos, que sea el que más se le asemeje en base a un criterio de distorsión seleccionado. En el capítulo 2 se presenta un estudio más detallado de esta técnica de codificación.

## 1.2 El codificador de dos canales.

En la práctica, un codificador puede usar una combinación de dos o más de los esquemas vistos, con el propósito de lograr un mejor intercambio costo/desempeño.

En un codificador de dos canales, se combina la técnica de codificación interpolativa/extrapolativa con algún otro esquema de codificación con o sin pérdidas. La imagen de entrada se separa en sus componentes de altas frecuencias, que representan el contraste local y los de bajas frecuencias, que representan la iluminación local media. En este esquema, los componentes de bajas frecuencias constituyen una versión filtrada pasobajas de la imagen original y por lo tanto se pueden submuestrear. Por otro lado, los componentes de altas frecuencias se pueden cuantizar burdamente, ya que no contienen información de la luminancia local y como las regiones de amplitud grande tienen un contraste local alto, entonces el ruido introducido será menos visible.

En la figura 1.5 se muestra la estructura de un codificador de dos canales tanto para el transmisor como para el receptor. Cuando en este tipo de codificadores la señal de entrada se divide en más de dos canales, se dice entonces que el sistema de codificación es subbanda. Este método de procesamiento se aplicó por primera vez por Crochiere, Webber y Flanagan, en tareas de codificación de voz [CROWEB76], y posteriormente por Vetterli, Woods y O'Neil en tareas de codificación de imágenes [VETTER84][WOONEI86].



vectorial y una sin pérdidas, que es la codificación de Huffman, en la etapa final de asignación de bits.

La presentación de este documento consta de 6 capítulos, a través de los cuales se analizan las diferentes etapas que constituyen el método de codificación empleado. El capítulo 2, que corresponde al sistema de codificación, se concentra en las etapas de cuantización y de asignación de bits. En él se estudian las características del cuantizador vectorial, las condiciones de optimalidad que se deben satisfacer, así como las métricas empleadas para medir su desempeño y los parámetros que se deben de considerar durante su diseño. También se presentan tres algoritmos rápidos de codificación vectorial, los cuales alcanzan un desempeño igual al de la búsqueda exhaustiva. Además, se incluye una sección dedicada al codificador de Huffman, el cual constituye la etapa de asignación de bits en el esquema de codificación empleado.

En el capítulo 3, se analizan en detalle las características de la descomposición a múltiples escalas. Se presenta el esquema piramidal de Burt y Adelson, y se estudian las características de las pirámides Gaussiana y Laplaciana. En el capítulo 4, se presenta el esquema de codificación empleado durante este trabajo, en él se analizan las ventajas que se pueden obtener al combinar la representación a múltiples resoluciones con la cuantización vectorial. En el capítulo 5, se muestran los resultados obtenidos usando nuestro esquema de codificación y se hace una comparación con otros esquemas de codificación similares. El capítulo 6, está dedicado a las conclusiones y en él se hace un análisis final de los resultados obtenidos. Finalmente, en la bibliografía se incluye una lista de las referencias consultadas durante el desarrollo de esta tesis. La forma en la que se presentan las referencias es usando las tres primeras letras del apellido del primer autor y las tres primeras letras del apellido del segundo autor, seguidas del año en el que se publicó dicha referencia. El objetivo de usar este formato, es hacer más fácil la localización de cada una de ellas.

## Capítulo 2

# EL SISTEMA DE CODIFICACION

Un sistema de codificación de imágenes consiste principalmente de tres etapas: una etapa de transformación de los datos, una etapa de cuantización y una etapa de asignación de bits; como se muestra en la figura 1.

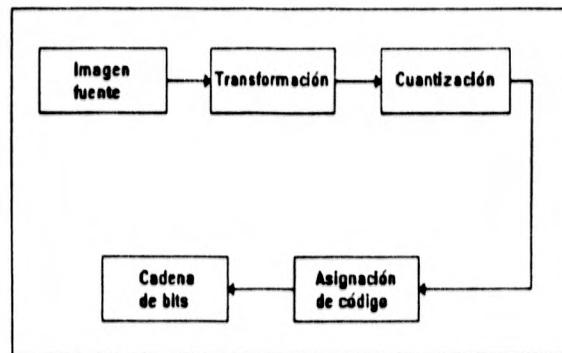


Figura 2.1: Sistema de codificación de imágenes.

El primer elemento, que es el más importante, es la adecuación o transformación de la imagen a un dominio más adecuado para efectuar la cuantización y la asignación de bits. Este elemento es el que determina esencialmente, lo que se codifica.

El segundo elemento es el cuantizador, el cual realiza una representación o cuantización, ya sea de las variaciones de intensidad en la imagen, de los coeficientes de la transformación o de los parámetros del modelo utilizado, empleando un número finito de bits. En nuestro trabajo, esta etapa está formada por un cuantizador vectorial, dado que presenta características muy favorables para el tipo de datos a

codificar. Finalmente, en la tercera etapa, es donde se asignan cadenas de bits a cada uno de los niveles de cuantización usados. Es importante que en esta etapa se emplee un método de codificación sin pérdidas, ya que las cadenas de bits que se tienen a la salida, son las que se transmiten a través de un canal de comunicaciones, o bien son las que se guardan para procesamientos posteriores.

Cada uno de estos elementos presentes en un codificador de imágenes, intenta explotar la redundancia presente entre los pixels de la imagen original, las limitaciones de los dispositivos físicos de despliegue y las características del sistema de visión humana. Cada uno de ellos constituye por separado un campo de estudio, que se relaciona con la representación eficiente de los datos.

Este capítulo se concentra en las etapas de cuantización y de asignación de bits, mientras que en el capítulo 3 se considera en detalle la etapa de adecuación de los datos, que para nuestro caso corresponde a la representación de las imágenes, usando una estructura piramidal multi-resolución.

## 2.1 Cuantización Vectorial.

El caso más sencillo de cuantización, es cuando se tiene un pixel de una imagen y se requiere encontrar un representante de él, que sea el más cercano bajo alguna métrica, dentro de un conjunto finito de valores numéricos permitidos. En forma más precisa, se define un cuantizador escalar unidimensional  $Q$ , de  $N$  puntos, como un mapeo  $Q : R \mapsto C$ ; donde  $R$  es el dominio de los números reales y  $C$  es el conjunto de las posibles salidas o libro de códigos de tamaño  $N$ , y está contenido en  $R$ , es decir,

$$Q : R \mapsto C; \quad C = \{y_i | i = 1, 2, \dots, N\} \subset R \quad (2.1)$$

Para el caso de imágenes en niveles de gris (8 bits),  $R$  representa el subconjunto de los números reales entre 0 y 255, incluidos. A los valores  $y_i$  se les conoce como los niveles de salida o de reproducción. Cada uno de estos valores de reproducción, determina una región,  $R_i \subset R$ , que puede ser de igual tamaño para cada uno de ellos si se trata de un cuantizador uniforme, o de diferente tamaño si se trata de un cuantizador no uniforme. Así, si un dato a cuantizar cae dentro del dominio de un cierto nivel de salida, entonces éste será su representante.

La cuantización vectorial es una generalización de la cuantización escalar. Las principales aplicaciones del caso unidimensional, son básicamente en tareas de conversión analógica/digital, mientras que la cuantización vectorial, por otra parte, se emplea usualmente en tareas de compresión de datos.

En cuantización vectorial, se tiene un diccionario  $C$  que contiene todos los vectores de reproducción permitidos. El proceso de codificación se puede ver como

una forma de reconocimiento de patrones, en el que la imagen original se descompone en vectores o patrones de imagen de dimensión  $k$ . Para cada uno de estos patrones se encuentra un vector de reproducción que mejor se le aproxime en base a un criterio de distorsión elegido, usualmente el error cuadrático, y se transmite la dirección de esta palabra del diccionario. Esto también se puede ver como la combinación de dos funciones: un codificador, que recibe un vector de entrada  $\mathbf{x}$  y genera la dirección de la palabra del código especificada por  $Q(\mathbf{x})$  y un decodificador, el cual usa esta dirección para generar el vector de reproducción  $\hat{\mathbf{x}}$ . En la figura 2.2, se ilustra la estructura básica de un codificador vectorial sencillo, en el cual, tanto en la etapa de codificación como en la de decodificación, se usa la regla del vecino más cercano, para encontrar la palabra del código que represente al vector de entrada. En la parte superior de la figura se muestran los pasos necesarios para encontrar el libro de códigos. En los años recientes, la cuantización vectorial se ha convertido en una técnica importante en tareas de reconocimiento y compresión de voz e imágenes, y su importancia y aplicaciones continúan creciendo [GERGRA91].

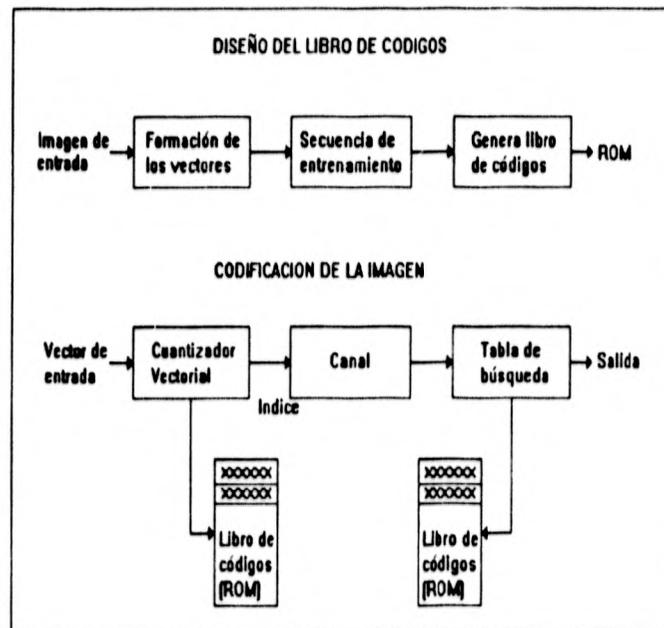


Figura 2.2: Diagrama a bloques de un cuantizador vectorial. Construcción del libro de códigos y codificación.

Un cuantizador vectorial de dimensión  $k$  y de tamaño  $N$  se define como un mapeo de un espacio vectorial Euclideo de dimensión  $k$ ,  $R^k$ , en un conjunto  $C$ , que contiene  $N$  posibles salidas de reproducción, llamadas vectores código o palabras del código, es decir,

$$Q : R^k \mapsto C; \quad C \subset R^k; \quad C = \{y_i | i = 1, 2, \dots, N\} \quad (2.2)$$

En este caso, el conjunto  $C$  se conoce como el diccionario o el libro de códigos de tamaño  $N$ , entendiéndose ésto como  $N$  elementos distintos, cada uno un vector que pertenece al espacio  $R^k$ .

Asociado a cada uno de los  $N$  vectores del diccionario, existe una región  $R_i$  que pertenece al espacio  $R^k$  y que en su conjunto dividen este espacio en  $N$  regiones de decisión. Cada una de las regiones  $R_i$  está definida como:

$$R_i = \{y_i | i = 1, 2, \dots, N\} \quad (2.3)$$

y cumplen con las siguientes condiciones:

$$\bigcup_{i=0}^N R_i = R^k \quad y \quad R_i \cap R_j = \emptyset \quad \forall i \neq j \quad (2.4)$$

Por lo que estas celdas forman una partición del espacio Euclideo  $R^k$ .

Entre las ventajas potenciales de la cuantización vectorial están su capacidad para explotar la dependencia lineal y no lineal de los vectores, la libertad para escoger la forma de las celdas de la partición, la flexibilidad para determinar el tamaño del libro de códigos y la simplicidad de la estructura del receptor, que consiste únicamente de una tabla de búsqueda.

## 2.2 Medidas del desempeño de un Cuantizador Vectorial

La medida de la distorsión o criterio de fidelidad, es una medida de la calidad de un sistema de codificación de imágenes. Algunos de los criterios más usados en el procesamiento digital de imágenes son el error cuadrático medio (MSE), el error cuadrático medio normalizado (NMSE), el error raíz cuadrada del MSE (RMSE) y la relación señal a ruido pico (PSNR). El objetivo principal de un cuantizador vectorial, es generar la mejor secuencia de reproducción posible para una tasa de bits ( $r$ ) dada.

Una medida de distorsión  $\mathbf{E}$ , es una función de costo, no negativa,  $\mathbf{E}(\mathbf{x}, \hat{\mathbf{x}})$  asociada con un cuantizador que representa un vector de entrada  $\mathbf{x}$ , por medio de un vector de reproducción  $\hat{\mathbf{x}}$ .

## El error cuadrático medio, error cuadrático medio normalizado, error raíz cuadrada y la relación señal a ruido pico

Suponga que se tiene una imagen digital  $f(i,j)$  de tamaño  $N \times N$  pixels, donde  $i$  y  $j$  varían entre 0 y  $N-1$ . Cada pixel de la imagen  $f$  es una palabra de  $m$  bits ( $m = 8$  para el caso de imágenes en la escala de gris tipo byte), que corresponde a uno de los  $2^m$  valores posibles en la escala de gris. El codificador reduce esta enorme cantidad de  $N \times N \times m$  bits a una cantidad significativamente menor. En el decodificador, el proceso de reconstrucción genera una estimación  $\hat{f}(i,j)$  del mismo tamaño que la imagen original. Para cada uno de los pixels reproducidos se define el error de reconstrucción como:

$$E(i,j) = \hat{f}(i,j) - f(i,j) \quad (2.5)$$

El error cuadrático medio está dado por

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [\hat{f}(i,j) - f(i,j)]^2 \quad (2.6)$$

el error cuadrático medio normalizado es

$$NMSE = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [\hat{f}(i,j) - f(i,j)]^2}{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j)} \quad (2.7)$$

y el error raíz cuadrada está dado como

$$RMSE = \sqrt{(MSE)^2} = \sqrt{\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [\hat{f}(i,j) - f(i,j)]^2} \quad (2.8)$$

Es usual, que para imágenes fijas la medida de la fidelidad se exprese en términos de la relación señal a ruido pico dada en decibeles como

$$PSNR = 10 \log \left( \frac{255^2}{MSE} \right) \quad dB \quad (2.9)$$

Para medir el número de bits necesarios en la representación de cada vector de la imagen, se toma en cuenta la tasa del código o resolución  $r$  que se define como:

$$r = \frac{\log_2(N)}{k} \quad (2.10)$$

Este parámetro es un indicador de la precisión que se alcanza, con un cuantizador vectorial para un libro de códigos determinado.

Por razones de conveniencia matemática, las medidas de la fidelidad que más se usan en cuantización vectorial son las métricas en diferencias, que son aquellas que dependen tanto de la imagen original como de su estimación. La forma de estas medidas es  $E(\mathbf{x}, \hat{\mathbf{x}}) = L(\hat{\mathbf{x}} - \mathbf{x})$ . Resulta de particular interés la medida error cuadrático medio, porque cumple con la desigualdad del triángulo que establece que

$$E(\mathbf{x}, \hat{\mathbf{x}}) \leq E(\mathbf{x}, \mathbf{y}) + E(\mathbf{y}, \hat{\mathbf{x}}) \quad \forall \mathbf{y} \quad (2.11)$$

Este hecho nos permite, como se verá mas adelante, diseñar algoritmos rápidos para encontrar el libro de códigos de un cuantizador vectorial, usando una cantidad bastante menor de operaciones y obtener distorsiones iguales a las que se obtienen empleando el método de búsqueda exhaustiva.

## 2.3 Diseño del libro de códigos

En cuantización vectorial se necesita determinar un libro de códigos o diccionario de reproducción  $\mathbf{C}$ , que contenga todos los vectores permitidos durante la codificación y decodificación. Este diccionario se debe de conocer por anticipado, tanto en el transmisor como en el receptor y su construcción debe de ser anterior al proceso de codificación. Este diseño representa la esencia de la cuantización vectorial y es la etapa más difícil tanto conceptual como computacionalmente. En la figura 2.2 se pueden ver los pasos involucrados durante la formación del libro de códigos.

### 2.3.1 Condiciones de optimabilidad para un cuantizador vectorial

El diseño óptimo del diccionario debe de satisfacer tres condiciones necesarias de optimabilidad.

- **La condición del vecino más cercano.** Establece que para un libro de códigos dado, el vector a codificar debe de quedar representado por aquella palabra del código que minimice la distorsión. Es decir, dado un libro de códigos  $\mathbf{C} = \{\mathbf{y}_i | i = 1, 2, \dots, N\}$ , y un vector de entrada  $\mathbf{x}$ , la partición óptima debe satisfacer la condición

$$Q(\mathbf{x}) = \mathbf{y}_i \Leftrightarrow E(\mathbf{x}, \mathbf{y}_i) \leq E(\mathbf{x}, \mathbf{y}_j) \quad \forall j \quad (2.12)$$

Es decir, dado un decodificador, el codificador vectorial es un mapeo de mínima distorsión, por lo que

$$E(\mathbf{x}, Q(\mathbf{x})) = \min_{y_i \in \mathbf{C}} E(\mathbf{x}, y_i) \quad (2.13)$$

- **La condición del centroide.** Establece que para una partición dada  $\{R_i : i = 1, 2, \dots, N\}$ , los vectores código deben de ser precisamente los centroides de cada una de las regiones  $R_i$  definidas en la partición.
- **La condición de probabilidad cero para las fronteras.** Esta condición es necesaria para que un diccionario sea óptimo y establece que la probabilidad de que se presenten puntos que pertenezcan a dos regiones distintas de la partición, es cero, es decir,

$$P(\mathbf{x} : E(\mathbf{x}, y_i) = E(\mathbf{x}, y_j)) = 0 \quad \forall i \neq j \quad (2.14)$$

### 2.3.2 El algoritmo LBG

El algoritmo LBG (de las iniciales de los autores: Linde, Buzo y Gray [LIN-BUZ80]) o de las K medias, es un algoritmo iterativo que se basa en las condiciones de optimabilidad necesarias para diseñar el libro de códigos y opera de la siguiente manera:

1. Dado un libro de códigos inicial, se mapean todos los vectores de entrenamiento dentro de las regiones definidas por las palabras del código, usando la condición del vecino más cercano. Si la distorsión obtenida de este mapeo es suficientemente pequeña, se detiene el algoritmo con el libro de códigos resultante. En caso contrario,
2. Se encuentra el centroide de todos los vectores mapeados en cada una de las regiones definidas por los vectores del diccionario. Entonces, se substituye el vector código por este centroide, que idealmente minimiza la distorsión para el grupo de entrenamiento dado. Regresa al paso 1.

El diagrama de flujo de este algoritmo se muestra en la figura 2.3. El algoritmo se detiene cuando el cambio en la distorsión es menor que un cierto umbral determinado para el diseño. Obsérvese que este algoritmo produce una distorsión mínima local únicamente y no se puede asegurar la obtención de un libro de códigos globalmente óptimo.

### 2.3.3 Selección del libro de códigos inicial

Existen varios métodos para la selección del libro de códigos inicial. Uno de ellos es tomar como diccionario inicial un subconjunto de los vectores de entrenamiento, con la única condición de que sean lo más diferentes posible. Otra opción,

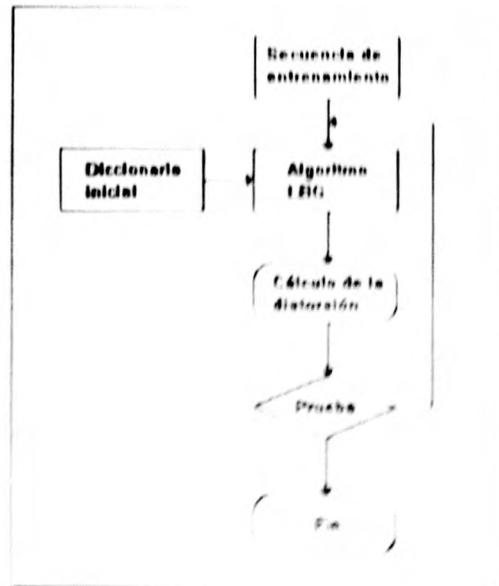


Figura 2.3: Diagrama de flujo del algoritmo LBG.

es considerar como suposición inicial una región verdaderamente cuantizada de los componentes individuales, eliminando aquellos que se repitan. Una técnica rápida de usar la técnica de partición. En esta propuesta primero se obtiene el contenido de la secuencia de entrenamiento, el cual representa el libro de códigos de dimensión cero. Para obtener el diccionario de dimensión 2 se distorsiona el libro de códigos del nivel cero y con esto se obtienen los nuevos índices con los cuales se aplica el algoritmo LBG y se obtienen las regiones de partición y los nuevos valores del código. Se procede de la misma manera hasta llegar al libro de códigos del tamaño deseado.

## 2.4 Algoritmos rápidos para la cuantización vectorial, usando la técnica del vecino más cercano

Una de las principales dificultades que se presentan cuando se trata del uso de códigos es la enorme cantidad de operaciones que se deben realizar. Se han desarrollado un gran número de propuestas con la finalidad de disminuir este problema. Algunas de las aportaciones que se han hecho a partir de algoritmos como CHEGERSA, CHEGERSA REIMORAT, SOLIMORAT, etc. son PALM. El objetivo principal de esta línea de investigación es disminuir el tiempo del cómputo que se requiere para la codificación de un mensaje.

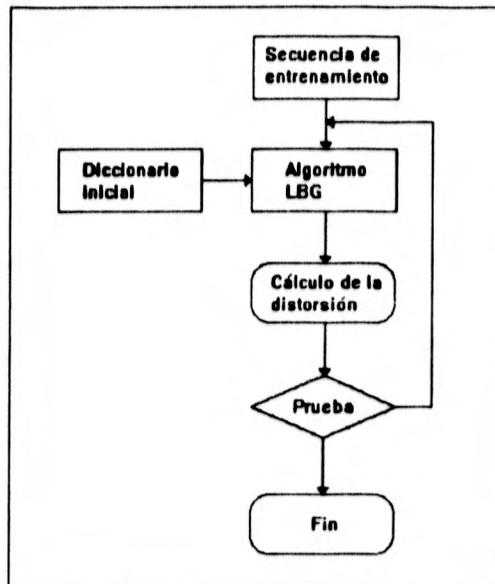


Figura 2.3: Diagrama de flujo del algoritmo LBG.

es considerar como suposición inicial una versión burdamente cuantizada de los componentes individuales, eliminando aquellos que se repitan. Una tercera opción, es usar la técnica de partición. En esta propuesta primero se obtiene el centroide de la secuencia de entrenamiento, el cual representa el libro de códigos de dimensión cero. Para obtener el diccionario de dimensión 2 se distorsiona el vector del código del nivel cero y con esto se obtienen dos nuevos vectores, con los cuales se aplica el algoritmo LBG y se obtienen dos regiones de partición y dos nuevas palabras del código. Se procede de la misma manera hasta llegar al libro de códigos del tamaño deseado.

## 2.4 Algoritmos rápidos para la cuantización vectorial, usando la técnica del vecino más cercano

Una de las principales dificultades que se presentan durante el diseño del libro de códigos, es la enorme cantidad de operaciones que se tienen que realizar. Se han desarrollado un gran número de propuestas con la finalidad de atenuar este problema. Algunos de los algoritmos que se revisaron se pueden encontrar en [CHEGER84], [CHEGER86], [REZMOR87], [SOLMOR89], [EQUITZ87] y [RAMPAL92]. El objetivo principal en cada uno de estos algoritmos, es encontrar la palabra del código más cercana al vector de entrada sin tener que recurrir a consul-

tar todos los vectores del diccionario. El principal inconveniente es que en la mayoría de estos algoritmos, la ganancia en cuanto al número de operaciones realizadas y el tiempo de ejecución es aleatoria, y usualmente el libro de códigos obtenido genera una mayor distorsión que el algoritmo de búsqueda exhaustiva.

Uno de los métodos más sencillos para la búsqueda rápida, es el de la distorsión parcial propuesto en [BEIGRA85]. Este método usualmente reduce el tiempo promedio de búsqueda por un factor de 4 y sólo requiere de una ligera modificación con respecto al algoritmo de búsqueda exhaustiva. La clave de esta propuesta radica en checar la distorsión que se produce componente por componente al representar cada vector de entrada con una palabra del diccionario. Así, si el cuantizador usa vectores de longitud  $k$ , se puede determinar si una palabra del código no es la correcta para representar al vector de entrada, si para alguna  $i < k$ ,  $i = 1, 2, \dots, k$ , la distorsión acumulada es mayor a la permitida, con lo que se descarta dicha palabra y se continúa con la siguiente.

En [PALRAM89] se discute la mejora del método de la distorsión parcial, que se obtiene al ordenar las palabras del libro de códigos en función de su probabilidad de ocurrencia. Con esto, los vectores del código con mayor probabilidad de aparecer se consideran primero durante el proceso de la búsqueda del vecino más cercano, con lo que se reduce el tiempo de cálculo necesario y el número de operaciones ejecutadas.

Los tres algoritmos que se presentan a continuación, tienen la propiedad de poseer un desempeño igual al de la búsqueda exhaustiva, mientras que reducen el tiempo de ejecución del algoritmo de cuantización vectorial, tomando ventaja de usar como medida de la distorsión el error MSE, o alguna otra medida que satisfaga la desigualdad del triángulo.

### 2.4.1 El método de la esfera

Si se define  $\mathbf{C} = \{\mathbf{Y}_i, i = 1, 2, \dots, N\}$  como un libro de códigos de tamaño  $N$  donde  $\mathbf{Y}_i = (y_{i1}, y_{i2}, \dots, y_{ik})$  es un vector del código de dimensión  $k$ . Para un vector de entrada  $\mathbf{X} = (x_{11}, x_{12}, \dots, x_{1k})$ , debemos encontrar el vector del código  $\mathbf{C}(\mathbf{X})$  que sea el más cercano a  $\mathbf{X}$ , en base a alguna métrica.

La clave del algoritmo de la esfera se muestra en la figura 2.4 (a), y se puede establecer de la siguiente manera: dada una medida de distancia apropiada, si se conoce que la distancia entre el vector de entrada  $\mathbf{X}$  y la palabra del código  $\mathbf{Y}_i$  es  $h_i$ , entonces, la mejor palabra del código se debe de localizar dentro del círculo centrado en  $\mathbf{Y}_i$  y cuyo radio es  $2h_i$ , por lo que aquellos vectores que queden afuera de esta región se pueden excluir de la búsqueda, con lo que se reduce el número de operaciones y el tiempo de ejecución.

El algoritmo es el siguiente:

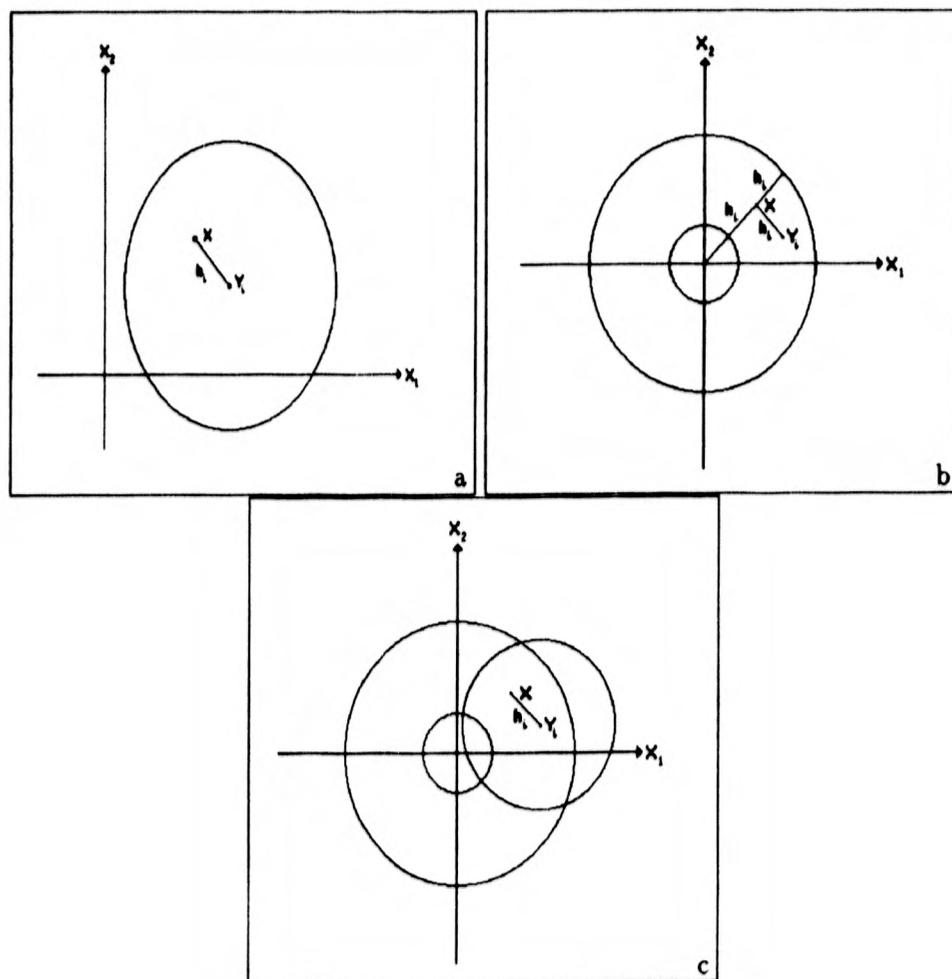


Figura 2.4: (a) Método de la esfera. La región contenida en la esfera de radio  $2h$ , centrada en  $Y$ , define el espacio de búsqueda. (b) Método del anillo. La región contenida entre los límites del anillo definen el espacio de búsqueda. (c) Método esfera-anillo.

1. Dado el libro de códigos  $C = \{Y_i, i = 1, 2, \dots, N\}$ , determine la norma  $r_i$ , para cada una de las palabras del código.
2. Asigne a cada vector del código un índice determinado por el orden ascendente de  $r_i$ .
3. Encuentre la matriz de distancias  $D$  cuyos elementos son

$$d_{ij} = d(Y_i, Y_j) = \|Y_i - Y_j\| \quad (2.15)$$

como esta matriz es simétrica, con ceros en la diagonal, entonces el espacio de memoria requerido para su almacenamiento, se reduce a  $N(N-1)/2$ , donde  $N$  es el tamaño del libro de códigos.

4. Para cada vector  $X$  de la entrada
  - (a) Calcule  $r_x$ .
  - (b) Identifique el vector del código  $Y_i$ , cuya  $r_i$  sea la más cercana a  $r_x$ .
  - (c) Calcule la mínima distancia,  $h_i$  entre  $X$  y  $Y_i$ .
  - (d) Identifique el subconjunto  $S$  del diccionario, que contiene todas las palabras del código que satisfacen la condición

$$d(Y_k, Y_j) \leq 2h_i \quad (2.16)$$

- (e) Busque en  $S$  el vector más cercano a  $X$ , si se encuentra un vector del código más cercano a  $X$  designelo como  $Y_i$  y regrese al paso 4(d), de otro modo el vector  $Y_i$  actual es el mejor representante de  $X$ .

**DESVENTAJA:** Este algoritmo requiere de una tabla, que contenga las distancias entre las palabras del código y que ocupa un espacio de memoria de  $N(N-1)/2$ , donde  $N$  es el tamaño del libro de códigos.

## 2.4.2 El método del anillo

Este algoritmo tiene la ventaja de que no requiere la tabla de distancias del método anterior. Los vectores del código se ordenan como antes, y  $r_i$ , así como la distancia, se definen de la misma manera. Para cada vector de entrada  $X$  se calcula  $r_x$ . Dado un vector del código  $Y_i$ , que sea el mejor representante de  $X$  y que está a una distancia  $d(X, Y_i) = h_i$  de  $X$ , se puede demostrar que el mejor representante de  $X$ ,  $Y_k$ , debe de satisfacer la condición

$$r_x - h_i \leq r_k \leq r_x + h_i \quad (2.17)$$

La ecuación (2.17) define el espacio de búsqueda mostrado en la figura 2.4 (b), que generalmente será mucho menor que el libro de códigos entero, siempre que se pueda encontrar una adecuada palabra del código  $Y_i$ , al inicio. El algoritmo es el mismo que para el caso anterior, salvo que se elimina el paso 3 y en el paso 4(d) se usa la ecuación (2.17) en lugar de la (2.16).

### 2.4.3 El método de la esfera y el anillo

Si se observa que una vez que se encuentra un vector candidato  $Y_i$  para el vector de entrada  $X$ , se deben de satisfacer al mismo tiempo tanto la ecuación (2.16) como la ecuación (2.17), entonces se pueden combinar los dos algoritmos anteriores. En este caso el espacio de búsqueda, que se muestra en la figura 2.4 (c), será mucho menor que el definido por cualquiera de estas ecuaciones por separado, pero tiene el inconveniente, al igual que el primer algoritmo, de que requiere de una tabla de distancias de tamaño  $N(N-1)/2$ . El algoritmo es el mismo que antes, donde ahora en el paso 4(d) se usan las ecuaciones (2.16) y (2.17) para definir el espacio de búsqueda.

## 2.5 El codificador de Huffman, construcción y restricciones para el diseño

La tercera etapa de un sistema de codificación corresponde a la asignación de bits. La salida de este elemento es la que finalmente se entrega al codificador de canal para realizar la transmisión de la información codificada. La principal característica de esta etapa es que debe de ser un proceso reversible, en el sentido de que se pueda obtener la reconstrucción exacta de la información después de la asignación de bits.

Resulta de interés determinar el mínimo número de bits que se requieren para codificar un cierto mensaje. En este sentido, el teorema de la codificación de fuente, establece que, en promedio, el número mínimo de bits que se requieren para codificar una fuente con entropía  $H$ , está dado por la propia entropía en bits, la cual se define como

$$H = - \sum_{i=1}^M p_i \log_2(p_i) \text{ bits} \quad (2.18)$$

con

$$\sum_{i=1}^M p_i = 1 \quad (2.19)$$

donde  $p_i$  son las probabilidades de ocurrencia de cada uno de los  $M$  símbolos.

El código de Huffman es un método óptimo para el diseño de códigos compactos o de mínima redundancia, el cual es simple de usar y unívocamente decodificable.

Como resultado de esto, se obtienen las más bajas tasas de bit promedio, las cuales se encuentran entre  $H$  y  $H + 1$ . La construcción de este código requiere usar una estructura de árbol, a través de la cual se realiza la asignación de los bits a cada uno de los símbolos a codificar.

Por razones de simplicidad en el diseño de un código de Huffman, frecuentemente se restringe al caso binario. La metodología que se sigue para obtener un código de Huffman a partir de un conjunto de  $M$  datos, los cuales pueden tomar valores entre 0 y  $N$ , es la siguiente:

1. Encuentre el histograma de los  $M$  datos a codificar.
2. Ordene el histograma en función de la probabilidad de ocurrencia de cada símbolo. Si el número de símbolos en el histograma es 1, termina. En caso contrario,
3. Sume la probabilidad de los dos símbolos menos probables del histograma. Asigne un bit 1 al símbolo cuya probabilidad sea la más baja y un 0 al otro símbolo. Si uno o los dos símbolos fusionados en este punto, es el resultado de haber juntado dos o más símbolos en los niveles anteriores, asigne el bit correspondiente a todos ellos.
4. Forme un nuevo histograma de nivel  $N-1$ , que contiene  $N-1$  símbolos, asígnele la probabilidad obtenida en el paso anterior, al nuevo símbolo generado de la fusión en el paso 3. Haga  $N=N-1$  y regrese al paso 2.

Al salir de este procedimiento, se tendrá una cadena de bits para cada uno de los símbolos del histograma que tenga una probabilidad de ocurrencia diferente de cero. Para obtener el código de Huffman todo lo que se tiene que hacer es leer esas cadenas de bits de derecha a izquierda.

En la figura 2.5 (a) se ilustra el procedimiento para generar un código de Huffman usando como secuencia a codificar el conjunto de las vocales. En este ejemplo, cada uno de los símbolos se muestra en el orden convencional de aparición. Siguiendo los pasos descritos anteriormente, se tiene una secuencia de  $N$  etapas, donde  $N$  es igual al número de símbolos a codificar. Estas etapas están representadas por las columnas de la figura 2.5(a). En la primera etapa se inicia con los  $N$  símbolos o nodos que contienen su respectiva frecuencia relativa. En cada una de las etapas se identifican las dos probabilidades menores, señaladas con un pequeño cuadro a su derecha, las cuales se suman para generar un nuevo nodo y se eliminan de la lista de nodos activos. En este paso, se le asigna un uno al nodo cuya probabilidad es menor y un cero al otro nodo. Cada uno de los nodos activos se reordena en la siguiente etapa, en función de su frecuencia relativa y se continúa de este modo hasta obtener un solo nodo de probabilidad 1.

Etapa	1	2	3	4	5
1	A 0.12	0.12			
2	E 0.42	0.42	0.42	0.42	
3	I 0.09				
4	O 0.30	0.30	0.30		
5	U 0.07				
6		UI 0.16			
7			AIU 0.20		
8				AIUO 0.50	
9					EAIUO 1.00

Símbolo	Probabilidad	Longitud	prom L	Código
A	0.12	3	0.36	011
E	0.42	1	0.42	1
I	0.09	4	0.36	0100
O	0.30	2	0.60	00
U	0.07	4	0.28	0101

Figura 2.5: (a) Construcción del código de Huffman, usando como secuencia a codificar el conjunto de las vocales, con las probabilidades relativas asociadas que se muestran. (b) Código asignado a cada una de las vocales y longitud promedio requerida para su codificación

Ahora, para ver el código, se lee de derecha a izquierda la palabra binaria asignada a cada símbolo, donde el resultado de esta operación es el código de Huffman para este conjunto de datos. En la figura 2.5 (b), se muestra el código de Huffman obtenido para este ejemplo, y también se muestran las longitudes promedio para cada uno de los símbolos. Finalmente, la longitud promedio requerida para codificar el mensaje es

$$L_{prom} = \sum_{i=1}^N p(i)L(i) = 2.02 \text{ bits} \quad (2.20)$$

Como puede verse, la longitud promedio es menor al número de bits que se necesitarían en un código de longitud fija para representar a este grupo de datos, que para el ejemplo considerado serían 3 bits. A la relación entre el número de bits requeridos en un código de longitud fija y la longitud promedio del código de Huffman se le conoce como la ganancia del código, que para este ejemplo es de  $1.485(3/2.02)$ . Se define la eficiencia del código, como la relación entre la entropía del mensaje y la longitud promedio. Esta cantidad, nos indica el porcentaje de compresión lograda por el código, que para este ejemplo es de  $98.76(1.995/2.02)$  por ciento.

Se tienen las siguientes restricciones básicas en la construcción de un código de Huffman:

1. No se pueden tener dos o más palabras del código que sean iguales.

2. Las palabras del código deberán formarse de tal modo que no se necesiten indicadores adicionales para señalar el inicio y el final de cada una de ellas, una vez que sea conocido el punto de inicio de una secuencia de palabras del código.
3. El código deberá construirse de tal modo que ninguna de las palabras del código sea prefijo de alguna otra palabra.

## 2.6 Aplicaciones del código de Huffman en la codificación de imágenes de error

Las imágenes de diferencia ( $E$ ), que se obtienen en las técnicas predictivas, usualmente tienen una varianza muy reducida, y además, están significativamente menos correlacionadas, en comparación con la imagen original. El histograma de estas señales está fuertemente concentrado alrededor de cero y se asemeja a una distribución Laplaciana de la forma:

$$p(E) = \frac{1}{\sqrt{2}\sigma_E} \exp\left(\frac{-\sqrt{2}|E|}{\sigma_E}\right) \quad (2.21)$$

También, se ha demostrado, que los histogramas de las imágenes de diferencia, en general tienen la misma forma y únicamente difieren en el parámetro de la varianza  $\sigma_E^2$  [RABJO91]. Estas imágenes de diferencias, usualmente se codifican usando un codificador de entropía para obtener un proceso sin pérdidas, y en este sentido, el código de Huffman representa una opción atractiva para este fin.

## Capítulo 3

# DESCOMPOSICION EN MULTIPLES RESOLUCIONES

Una característica de las imágenes del mundo físico, es la correlación que existe entre los pixels, si se consideran éstos dentro de una vecindad. Esto hace que, representar una imagen directamente en términos de los valores de los pixels, sea ineficiente, porque la mayor parte de la información codificada es redundante. El primer paso en la construcción de un sistema de codificación de fuente, consiste en diseñar la etapa de adecuación de los datos, para obtener una representación de ellos en la cual, ciertamente, se elimine la correlación presente en la imagen. Este objetivo se logra usando las técnicas de codificación predictivas y por transformadas, vistas en el capítulo 1. Ambas técnicas tienen sus ventajas y sus inconvenientes, en la primera, la estructura del codificador es muy sencilla de implantar y se obtiene una buena adaptabilidad a las características locales de la imagen. Por otro lado, como se mencionó anteriormente, las técnicas de codificación por transformadas generalmente proveen mayores tasas de compresión, pero éstas las logran a expensas de una mayor complejidad computacional.

La distribución no homogénea de la energía de las imágenes en el dominio frecuencial, ha convertido la descomposición en múltiples resoluciones de las señales, en un problema práctico [AKAHAD92]. El concepto básico es, dividir el espectro de la señal en subespectros, y entonces tratar estos subespectros individuales para propósitos de procesamiento. Desde el punto de vista de la codificación de señales, se puede apreciar que los subespectros que posean mayor contenido de energía, son los que tendrán mayor prioridad en procesamientos posteriores. En imágenes digitales, se tienen predominantemente componentes de bajas frecuencias, por lo que las subbandas pasobajas contendrán la mayor parte de la energía. Si únicamente se ocupan estas subbandas durante el proceso de síntesis, entonces es de esperarse, que el error de reconstrucción sea pequeño, para fines prácticos.

La descomposición del espectro de una señal en subbandas, provee una base matemática para tres características importantes en el análisis y procesamiento.

- Se hace posible el monitoreo de los componentes de energía de las señales, dentro de cada subbanda.
- La descomposición en subbandas del espectro de la señal, conduce a la representación en múltiples resoluciones, por medio del procesamiento de señales a múltiples tasas de acuerdo al teorema del muestreo de Nyquist.
- El procesamiento a múltiples tasas, nos habilita para la transmisión progresiva de la información.

### 3.1 Procesamiento de señales a múltiples tasas

En un sistema multi-tasa, las muestras de una señal se procesan y manipulan a diferentes tasas de reloj en varios puntos dentro de la configuración. Típicamente, una imagen analógica limitada en banda se muestrea a la tasa de Nyquist para obtener una señal de banda completa, la cual posee un contenido espectral desde cero hasta la mitad de la frecuencia de muestreo. Esas muestras de la señal se pueden procesar, ya sea a mayores o menores tasas de reloj, usando interpolación o decimación. Estas operaciones constituyen el marco de trabajo de la descomposición en múltiples resoluciones.

#### 3.1.1 Decimación e interpolación

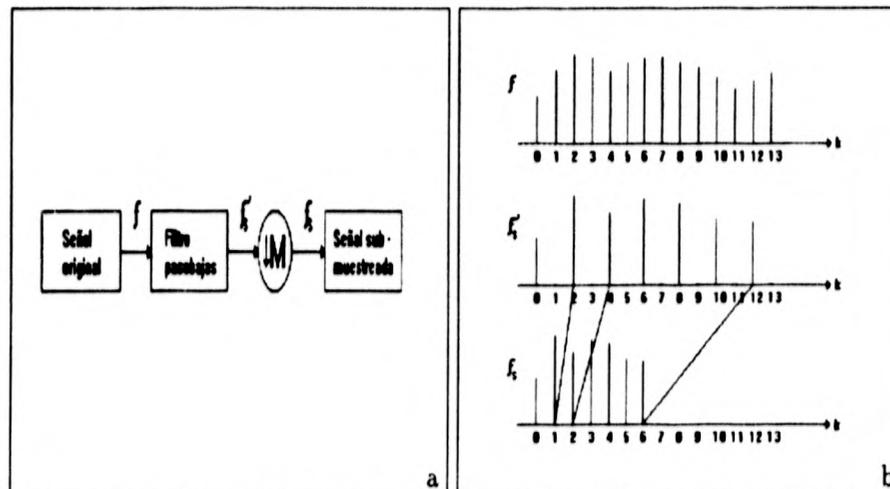


Figura 3.1: Decimación (a) Representación esquemática. (b) Señales original, intermedia y submuestreada.

**La decimación** es el proceso de reducir la tasa de muestreo de una señal, por un factor entero  $M$ . Este proceso se lleva a cabo pasando la señal original, a través

de un filtro pasobajas sin traslape, y entonces, submuestreando la señal filtrada por el factor  $M$ . Para el caso unidimensional, esto se puede expresar matemáticamente de la siguiente manera: dada una señal  $f$ , se define una señal intermedia  $f'$  como

$$f_s'(k) = \begin{cases} f(k) & \text{para } k = 0, \pm M, \pm 2M, \dots \\ 0 & \text{en otro caso} \end{cases} \quad (3.1)$$

y la señal submuestreada  $f_s$  se obtiene entonces, tomando sólo una muestra de cada  $M$  posibles

$$f_s(k) = f'(kM) = f(kM) \quad (3.2)$$

En la figura 3.1 (a) se muestra la representación de la decimación, y en (b) se ilustran las señales original, intermedia y submuestreada que se obtienen de esta operación para  $M = 2$ .

**La interpolación** es el proceso de incrementar la tasa de muestreo, por un factor entero  $M$ . Este proceso se puede analizar en dos pasos: (1) sobremuestrear, llenando con  $(M - 1)$  ceros los espacios entre muestras, y (2) filtrar pasobajas la señal obtenida en el paso (1). Esto se puede expresar como:

$$f_o(k) = \begin{cases} f(\frac{k}{M}) & \text{para } k = 0, \pm M, \pm 2M, \dots \\ 0 & \text{en otro caso} \end{cases} \quad (3.3)$$

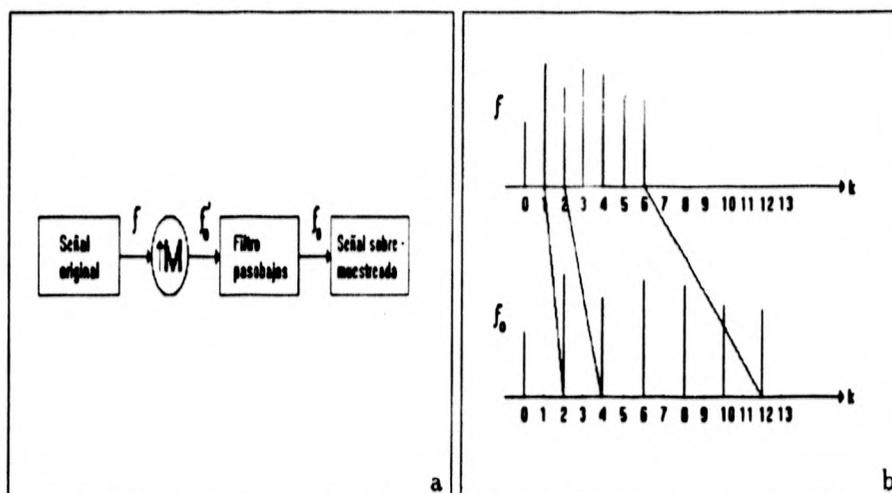


Figura 3.2: Interpolación (a) Representación esquemática. (b) Señales original e interpolada.

En la figura 3.2 (a) se muestra la representación de la interpolación, y en (b) se ilustran las señales original e interpolada que se obtienen de esta operación.

## 3.2 Transmisión progresiva

Una transmisión progresiva, consiste en enviar sucesivamente partes de la información contenida en una imagen, por etapas, en cada una de las cuales, en el receptor se reconstruye una aproximación de ésta. La imagen reconstruida se mejora progresivamente, conforme se recibe mayor cantidad de datos, hasta obtener la reconstrucción final.

La motivación para transmitir progresivamente la información contenida en una imagen, se encuentra en la necesidad de usar canales de ancho de banda reducido, como son las líneas telefónicas. Una de las propiedades que posee este esquema, es el reconocimiento temprano. La transmisión puede detenerse en algún momento si la calidad de la imagen reconstruida es suficientemente buena, o si la información contenida en ella no es de interés. Dentro de las características que debe de tener un esquema de transmisión progresiva están:

- Transmisión a tasas de bit bajas para las primeras aproximaciones, en las cuales se toma la decisión de conservar o rechazar la imagen. Esto nos permite una compresión efectiva grande.
- Hacer uso de todos los datos procesados previamente, por lo que la información adicional que se tiene que transmitir en la siguiente etapa, se mantiene en un mínimo.
- La capacidad de transmisión total de una imagen, a una tasa baja de bit y con excelente calidad de la reconstrucción.
- Emplear algoritmos de codificación y decodificación que sean relativamente rápidos, y adecuados para implantaciones en arquitecturas de procesamiento.

## 3.3 Codificación de dos canales

La codificación de dos canales constituye el punto de partida para el estudio de los sistemas de codificación en múltiples resoluciones. El objetivo de esta operación, es separar la señal en dos bandas de frecuencias, y entonces codificar cada una de estas bandas en base a su contenido de energía.

### 3.4 Estructura de árbol subbanda de banda diádica

Una estructura de árbol de banda diádica, es una estructura irregular. En ella, el espectro de la imagen se divide en dos subbandas iguales: una de altas frecuencias y una de bajas frecuencias. Después, se subdivide nuevamente la parte baja del espectro en dos bandas iguales, por lo que el espectro alto permanece sin cambios. En la figura 3.3 se ilustra la partición resultante de aplicar esta estructura. En esta descomposición, la banda baja, representada como BB, proporciona una versión burda de la señal original, mientras que la banda alta, BA, contiene la información de los detalles presentes en la misma.

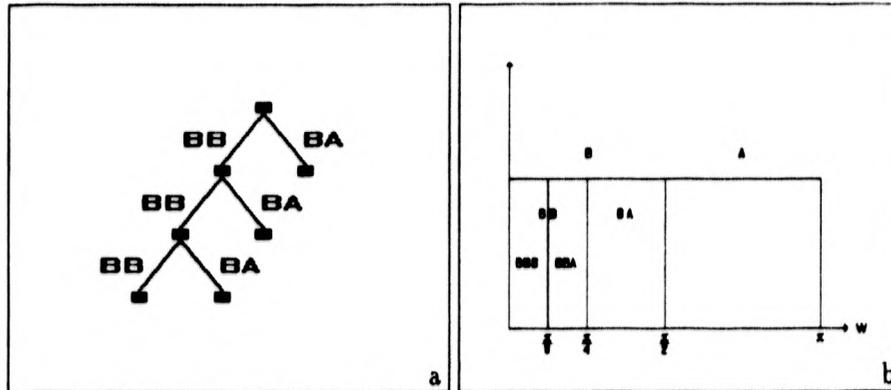


Figura 3.3: Descomposición del espectro de frecuencias en bandas diádicas.

La descomposición en múltiples resoluciones es una metodología que pusieron por primera vez Burt y Adelson[BURADE83], para problemas de codificación de imágenes y de visión. En este capítulo se describe el algoritmo piramidal de Burt y Adelson, el cual constituye la base del objetivo principal de este trabajo. El propósito de este algoritmo es, proveer un proceso automático en el contexto de imágenes digitales, para calcular la imagen en la escala  $2^{j+1}$  a partir de la imagen en la escala  $2^j$ . Si la imagen original corresponde a un arreglo bidimensional de  $256 \times 256$  pixels, el algoritmo piramidal primero nos da una imagen de  $128 \times 128$ , después una de  $64 \times 64$  y así sucesivamente hasta alcanzar el límite de un pixel. La importancia de los algoritmos piramidales se deriva de su estructura iterativa, la cual usa el resultado de un nivel a una escala dada para pasar a la siguiente.

### 3.5 El algoritmo de Burt y Adelson

Este es un algoritmo que combina las características de los métodos predictivos, por transformadas e interpolativos/extrapolativos, con la finalidad de eliminar o reducir la correlación presente en la imagen. Esta es una técnica no causal, que se basa en vecindades simétricas centradas en cada pixel de la imagen, lo que permite obtener predicciones más precisas, y por consiguiente una mayor compresión de los datos. Además, emplea cálculos locales y sencillos.

La forma de operación del algoritmo es la siguiente:

- Dada una imagen inicial  $G_0$  de tamaño  $R_k$  renglones  $\times$   $C_k$  columnas, y  $k = 0$ , se hace una predicción de cada uno de los pixels de la imagen, calculada ésta, como un promedio local ponderado, usando una función  $W$ , de forma Gaussiana unimodal, centrada en el pixel mismo. Como resultado, se obtiene una imagen filtrada pasobajas  $\hat{G}_0$ , la cual representa una estimación de  $G_0$ .
- La imagen pasobajas obtenida se decima a una tasa 2:1, en base al esquema mostrado en la figura 3.4, para producir una imagen de menor resolución espacial y temporal.

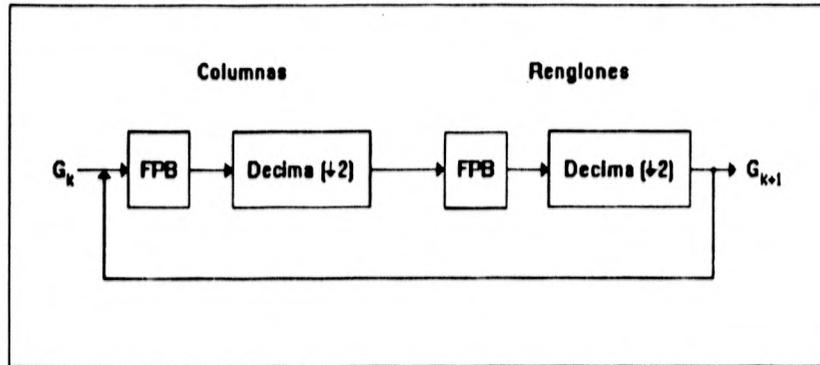


Figura 3.4: Diagrama para la construcción de un nivel de la pirámide Gaussiana.

Esto equivale a aplicar una función de filtrado pasobajas de la siguiente manera:

$$G_k(i, j) = \sum_{m=-2}^2 \hat{W}(m) \sum_{n=-2}^2 \hat{W}(n) G_{k-1}(2i + m, 2j + n) \quad (3.4)$$

$$\begin{aligned} \text{para } i &= 1, 2, \dots, R_k/2, \\ j &= 1, 2, \dots, C_k/2. \end{aligned}$$

- Se expande la imagen pasobajas obtenida en el punto anterior, de acuerdo a la función que se define en el apartado 3.5.2 y se le resta a la imagen original para obtener el error de predicción  $L$ , dado como

$$L_{k-1}(i, j) = G_{k-1}(i, j) - \hat{G}_{k-1}(i, j) \quad (3.5)$$

$$\text{para } i = 1, 2, \dots, R_k/2, \\ j = 1, 2, \dots, C_k/2.$$

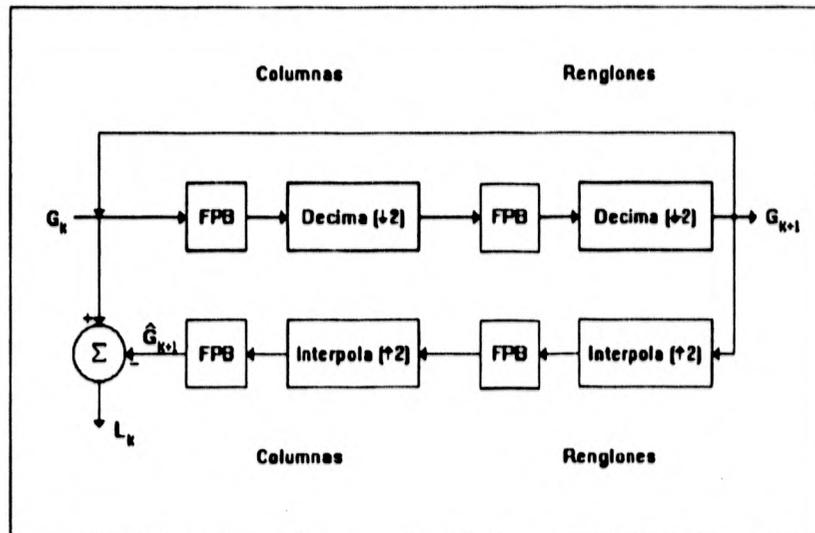


Figura 3.5: Diagrama para la construcción de un nivel de la pirámide Laplaciana.

Este resultado, representa una versión filtrada pasoaltas de la imagen original, que contiene las variaciones de altas frecuencias presentes en la imagen. En la figura 3.5, se muestra el esquema empleado para producir un nivel de la pirámide Laplaciana.

- Haga  $k = k + 1$  y regrese al primer punto.

La aplicación iterativa de estos pasos generará una pirámide de imágenes Gaussianas y otra de imágenes Laplacianas. Para cada una de estas estructuras y dados los anchos de banda de los filtros utilizados, existe una relación de muestreo de 2 a 1 entre niveles. En la figura 3.6 se muestra la representación típica de una estructura piramidal, para tres niveles, así como la partición correspondiente del plano de frecuencias.

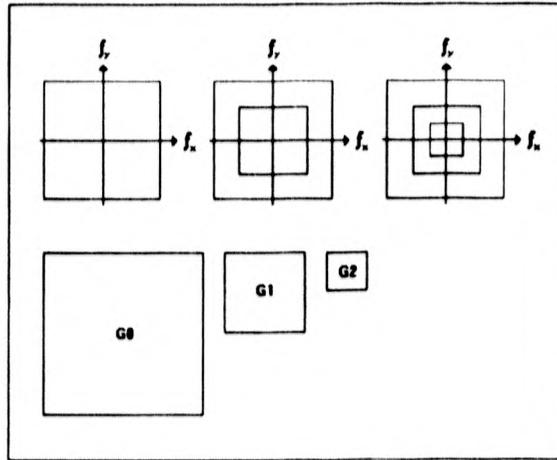


Figura 3.6: Representación espacial de una estructura piramidal de tres niveles, y su correspondiente partición del plano de frecuencias.

### 3.5.1 La función de filtrado Gaussiano

Si se tiene una imagen de tamaño  $C_k$  columnas  $\times$   $R_k$  renglones, se obtiene su versión filtrada pasobajas y submuestreada, de acuerdo al algoritmo original, usando una ventana de filtrado de tamaño  $5 \times 5$ , de la siguiente manera

$$G_k(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) G_{k-1}(2i + m, 2j + n) \quad (3.6)$$

$$\begin{aligned} \text{para } i &= 1, 2, \dots, R_k/2, \\ j &= 1, 2, \dots, C_k/2. \end{aligned}$$

donde la función Gaussiana  $W$ , que por simplicidad se supone separable,  $W(m, n) = \hat{W}(m)\hat{W}(n)$ , debe de cumplir con ciertas condiciones:

- La función unidimensional,  $\hat{W}$  debe de estar normalizada.

$$\sum_{m=-2}^2 \hat{W}(m) = 1 \quad (3.7)$$

- Debe de ser simétrica, es decir,

$$\hat{W}(m) = \hat{W}(-m) \quad \text{para } m = 0, 1, 2. \quad (3.8)$$

- Se debe satisfacer la condición de igual contribución, que establece que, todos los nodos de un nivel dado deben contribuir con el mismo peso total para

generar los nodos en el siguiente nivel superior. Esto da como resultado que si  $\hat{W}(0) = a$ ,  $\hat{W}(1) = \hat{W}(-1) = b$ , y  $\hat{W}(2) = \hat{W}(-2) = c$ , entonces se requiere que  $a + 2c = 2b$ , lo que significa que el kernel generador es

$$\begin{aligned}\hat{W}(0) &= a \\ \hat{W}(1) &= \hat{W}(-1) = \frac{1}{4} \\ \hat{W}(2) &= \hat{W}(-2) = \frac{1}{4} - \frac{a}{2}\end{aligned}$$

En la figura 3.7 se muestran las formas que toma la función  $\hat{W}(m)$  en términos de los valores del parámetro  $a$ . Como se puede ver, cuando  $a = 4$ , la función tiene una forma Gaussiana, mientras que para  $a = 6$ ,  $\hat{W}$  es trimodal.

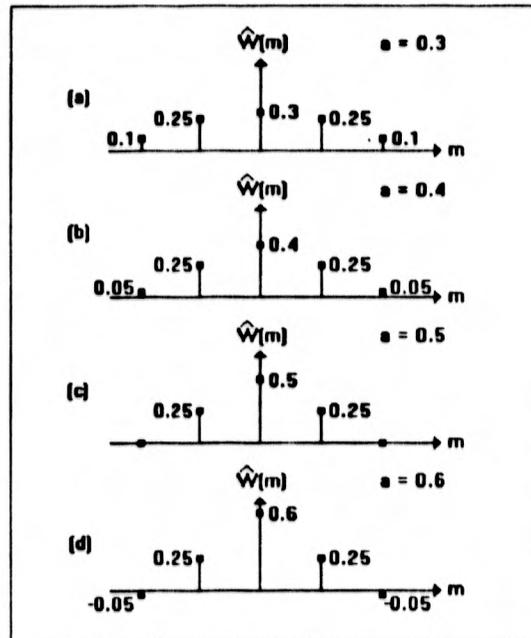


Figura 3.7: Respuesta al impulso de  $\hat{W}$  como una función del parámetro  $a$ .

### 3.5.2 Interpolación Gaussiana

Para obtener una imagen de nivel  $k-1$  a partir del nivel  $k$ , se tiene que realizar una interpolación Gaussiana, para generar los valores de nuevos nodos entre los nodos dados. La función de expansión que se utiliza para este propósito está dada como:

$$G_{k-1}(i, j) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 W(m, n) G_k \left( \frac{i-m}{2}, \frac{j-n}{2} \right) \quad (3.9)$$

$$\begin{aligned} \text{para } i &= 1, 2, \dots, R_{k-1}, \\ j &= 1, 2, \dots, C_{k-1}. \end{aligned}$$

Donde  $W$  es la misma función empleada en la generación de la pirámide Gaussiana.

En este punto, se consideran sólo aquellos términos cuyos índices sean enteros, para generar el valor de los pixels interpolados.

### 3.5.3 La función de filtrado Laplaciano

La función de filtrado Laplaciano se obtiene de forma indirecta. La imagen Laplaciana del nivel  $k-1$ , se obtiene como el error que existe entre los niveles  $k-1$  y  $k$  de la pirámide Gaussiana de la siguiente manera:

$$L_{k-1}(i, j) = G_{k-1}(i, j) - \text{EXPANSION}[G_k(i, j)] \quad (3.10)$$

$$\begin{aligned} \text{para } i &= 1, 2, \dots, R_{k-1}, \\ j &= 1, 2, \dots, C_{k-1}. \end{aligned}$$

donde la función EXPANSION está dada por la ecuación (3.9).

El conjunto de imágenes generadas al aplicar (3.10) a todos los niveles de la pirámide Gaussiana, nos da como resultado la pirámide Laplaciana.

## 3.6 Características de la pirámide Gaussiana

Las principales características del filtrado Gaussiano son:

- Se presenta una alta atenuación de los componentes de altas frecuencias, que están más allá de la banda de paso, en cada uno de los niveles de la pirámide.
- Cada uno de los niveles superiores, representa una estimación del nivel anterior, donde se tienen sólo una cuarta parte de los coeficientes estimados.

## 3.7 Características de la pirámide Laplaciana

Además de la simplicidad en su estructura, la pirámide Laplaciana nos ofrece las siguientes ventajas, importantes desde el punto de vista de la codificación de imágenes:

- La información contenida está significativamente descorrelacionada, lo cual resulta en que los valores de los pixels estén concentrados alrededor de cero.
- La entropía total de la pirámide es menor que la entropía de la imagen original, lo cual nos garantiza poder codificar los pixels a una tasa de bits menor.
- Se puede introducir un lazo para la compensación del error.
- Es posible la reconstrucción exacta.
- Esta estructura es particularmente adecuada para la transmisión progresiva de imágenes.

La entropía total de la pirámide Laplaciana  $H_T$ , se calcula como un promedio ponderado de las entropías parciales  $H_K$ , en cada uno de los niveles de ella. La fórmula que se emplea para este cálculo es:

$$E_T = \sum_{k=0}^{N-1} \frac{H_k}{2^{2k}} \quad (3.11)$$

donde N es el número de niveles en la pirámide.

### 3.8 Reconstrucción de la imagen original

La imagen original se puede recobrar sin pérdidas usando únicamente la pirámide Laplaciana de N etapas, cuyo nivel más alto coincide con el de la imagen Gaussiana. El procedimiento que se ilustra en la figura 3.8 es el siguiente:

- Se usa la expansión Gaussiana para obtener la imagen Laplaciana del nivel  $k = N$ , del doble de su tamaño. El resultado se suma con la imagen del nivel  $k-1$ , con lo que se obtiene la imagen Gaussiana reconstruida del nivel  $k-1$ .
- Para los niveles  $N-1$  hasta cero, se expande la imagen Gaussiana reconstruida del nivel  $k-1$ , y se suma con la imagen Laplaciana del siguiente nivel  $k-2$ , para obtener la imagen Gaussiana recobrada del nivel  $k-2$ . Es decir,

$$G_{k-2}(i, j) = L_{k-2}(i, j) + \hat{G}_{k-1}(i, j) \quad (3.12)$$

$$\begin{aligned} \text{para } i &= 1, 2, \dots, R_{k-2}, \\ j &= 1, 2, \dots, C_{k-2}. \end{aligned}$$

- Se hace  $k = k-1$  y se repite el paso anterior hasta que  $k = 0$ . Cuando se cumple esta condición, se tiene la reconstrucción sin pérdidas de la imagen original.

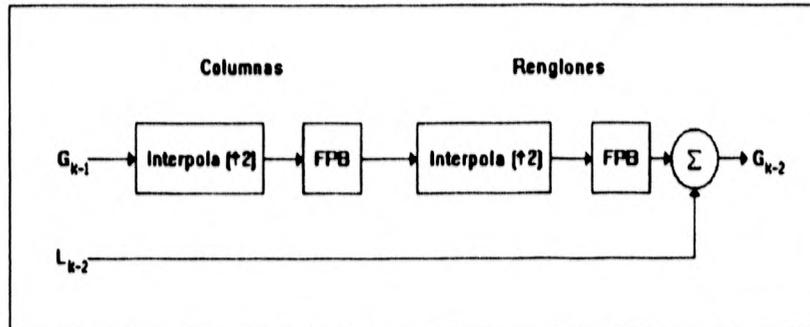


Figura 3.8: Diagrama de reconstrucción de la imagen original a partir de su representación mediante una pirámide Laplaciana.

En la figura 3.9 (a) se muestra la representación piramidal Gaussiana, con cuatro niveles, para la imagen Lena de  $256 \times 256$ . En 3.9 (b) se presenta la pirámide Laplaciana correspondiente, obtenida con el algoritmo de Burt y Adelson, en este caso, durante el despliegue se invirtió el histograma, para poder apreciar las variaciones de tono en las imágenes de diferencia. En 3.10 (a) se muestra el histograma de la imagen original, en el que se puede ver que los valores de los pixels están distribuidos a lo largo de todo el rango de la escala de gris. En 3.10 (b) a (d), se muestran los histogramas para las tres primeras imágenes de la pirámide Laplaciana. Como se puede apreciar, la distribución de los pixels se concentra alrededor de cero, lo cual representa una característica importante para fines de compresión.



Figura 3.9: Representación piramidal de cuatro niveles para una imagen (a) pirámide Gaussiana. (b) pirámide Laplaciana.

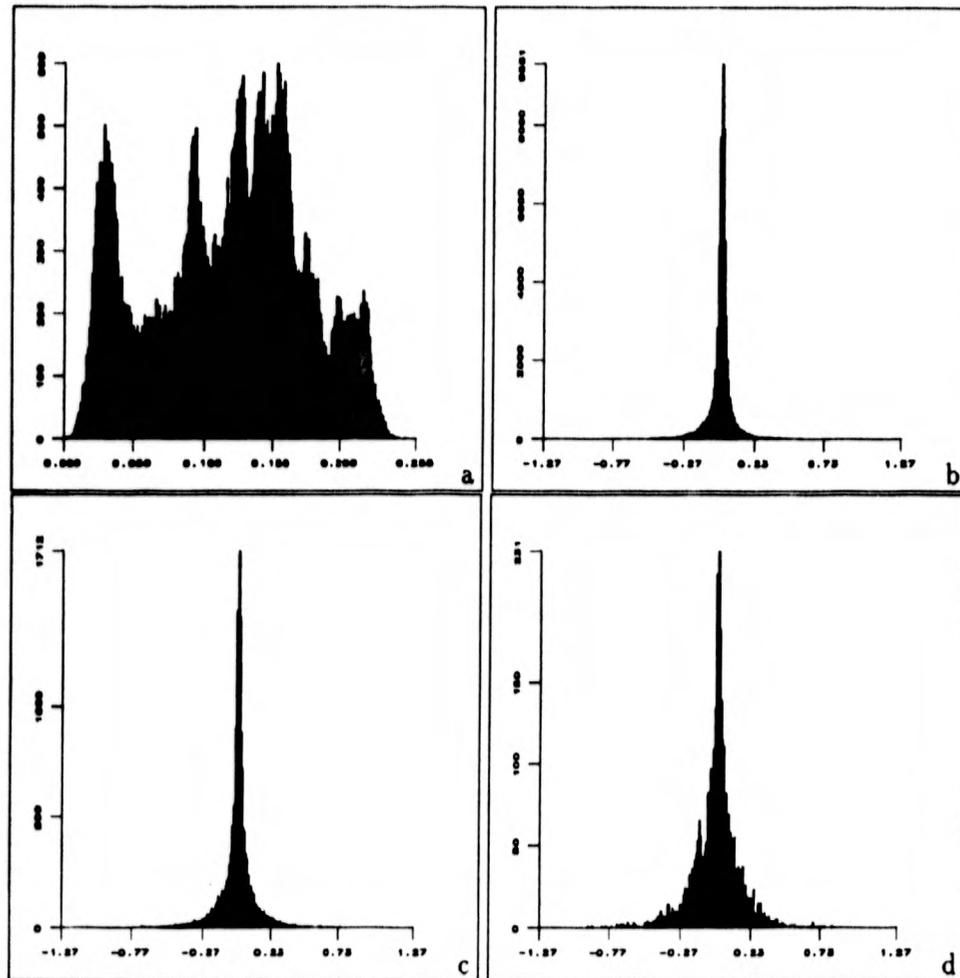


Figura 3.10: Histogramas de la representación piramidal de cuatro niveles para una imagen (a) Histograma de la imagen original. (b), (c) y (d) Histogramas de la pirámide Laplaciana. Niveles 0, 1 y 2 respectivamente.

## Capítulo 4

# EL ESQUEMA DE CODIFICACION UTILIZADO

En este capítulo se presenta el esquema de codificación utilizado. En él, se combinan cuatro técnicas de codificación con pérdidas, que son: predictiva, por transformada, interpolativa/extrapolativa y cuantización vectorial, y un esquema de codificación reversible, como lo es el código de Huffman.

Las categorías generales donde caen la mayor parte de las técnicas de transmisión progresiva de imágenes son tres: piramidal, en el dominio de la transformada y codificación iterativa.

En la propuesta piramidal, los diferentes niveles en la pirámide corresponden a las aproximaciones sucesivas de la imagen original. El contenido de la pirámide puede incluir el valor de la media, los pares media y diferencia, y/o versiones filtradas pasobanda de la imagen. En la propuesta basada en el dominio de la transformada la imagen primero se agrupa en bloques, a los que se les aplica una transformación, y los coeficientes resultantes se transmiten de manera progresiva en algún orden, usualmente del más bajo hasta el orden superior. Así, se logran aproximaciones sucesivas, con una mejor resolución en cada etapa nueva, mediante la transformación inversa de los coeficientes. En la tercera propuesta, se efectúa una codificación iterativa sobre la imagen residual o diferencia, ya sea en el dominio espacial o en el frecuencial. En cada una de las etapas se forma una imagen de diferencia la cual se codifica en la siguiente etapa.

En nuestro caso, nos enfocaremos únicamente al esquema piramidal, en el cual la representación de los datos se hace en forma similar a la vista en el capítulo anterior. Se forman las dos pirámides de datos usando una función de filtrado diferente a la de Burt y Adelson, pero se conservan las propiedades antes vistas.

Para obtener mayor eficiencia durante el proceso de codificación, se plantea usar un cuantizador vectorial sobre la pirámide Laplaciana en una base de nivel por nivel, donde los vectores se forman al particionar cada uno de éstos, en vectores

especialmente contiguos que no se traslapan. Finalmente, se emplea un codificador de Huffman para codificar sin pérdidas las imágenes cuantizadas en la etapa de asignación de bits, a fin de asegurar la reproducción perfecta.

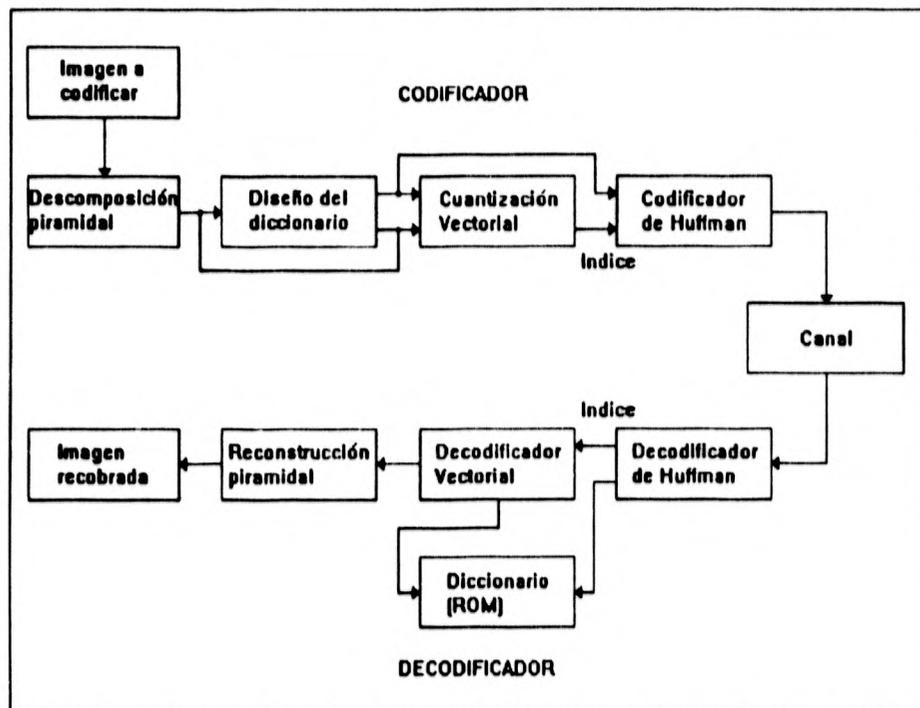


Figura 4.1: Esquema de codificación piramidal utilizado.

En la figura 4.1 se presenta el diagrama a bloques del esquema de codificación utilizado. Como se muestra, en la primera etapa se hace la descomposición en múltiples resoluciones de la imagen de entrada, que para nuestro caso se considera la Gaussiana de nivel  $N$ ,  $G_N$ . En este bloque, se obtiene la imagen Gaussiana del nivel  $N-1$ ,  $G_{N-1}$ , y la imagen Laplaciana del nivel  $N$ ,  $L_N$  como la resta de las dos imágenes Gaussianas. En general, dada la imagen Gaussiana del nivel  $K$ ,  $G_k$ , se obtienen de ella las imágenes  $G_{k-1}$  y  $L_k$ , donde  $k$  toma valores entre  $N$  y  $1$ , con  $N$  el número de niveles de la pirámide.

Una vez que se tiene la pirámide Laplaciana de  $N$  niveles, en la siguiente etapa se aplica la cuantización vectorial nivel por nivel, empezando con  $k = 3$ , para terminar en el nivel  $N$ . En este paso, primero se obtiene el libro de códigos para cada nivel  $k$ , empleando el algoritmo LBG en alguna de sus variantes mostradas en el capítulo 2. El libro de códigos obtenido se codifica usando el código de Huffman y se transmite a través del canal. Por otro lado, se cuantiza vectorialmente la

imagen en el nivel  $k$ ,  $L_k$ , con lo que se obtienen las direcciones de los vectores de reproducción, las cuales se codifican también usando el código de Huffman para su transmisión. Para evitar la propagación del error de cuantización del nivel  $k$ , éste se suma al nivel  $k+1$ , el cual se cuantiza vectorialmente siguiendo el procedimiento descrito.

Por último, se obtiene el error residual al nivel  $N+1$ , que representa la diferencia entre la imagen original y la sintética, y se codifica sin pérdidas, usando un código de Huffman para obtener una reconstrucción sin pérdidas de la imagen.

En el receptor, se ejecutan las operaciones inversas, primero, se decodifica el libro de códigos y se almacena en una memoria ROM. Después, se reciben las direcciones obtenidas durante la cuantización vectorial, las cuales se usan para direccionar los vectores de reproducción almacenados en la ROM, con el fin de recobrar los niveles de la pirámide Laplaciana. Finalmente, se reconstruye la imagen sintética, que cuando se transmite el error residual, es idéntica a la original.

A continuación, se describen cada una de las partes que constituyen nuestro sistema de codificación de imágenes digitales.

## 4.1 Descomposición piramidal de imágenes

Para medir la eficiencia de un esquema de codificación de imágenes digitales se considera la tasa de compresión, la calidad de la imagen decodificada y la complejidad de la implementación del algoritmo de codificación/decodificación en una arquitectura. El algoritmo de Burt y Adelson tiene la desventaja de que no considera los errores introducidos durante la codificación de las imágenes Laplacianas transmitidas. Este tiende a propagarse de los niveles inferiores hacia la imagen final reconstruida. Además, la complejidad de los filtros que se utilizan en los procesos de decimación e interpolación es comparable a la de un sistema de codificación por transformada [YEH89].

En este trabajo se utiliza un esquema modificado para la formación de la pirámide Gaussiana, con el cual se obtiene una mayor rapidez de operación y se mantiene una excelente calidad de la imagen estimada. Más aún, resultan estimaciones más precisas que si únicamente se submuestra la imagen en cada nivel, además de que el error de codificación se puede transmitir progresivamente. En esta representación se obtiene una secuencia de imágenes de tamaño reducido a partir de la imagen original, calculando la media truncada sobre bloques no traslapados de  $2 \times 2$  pixels, con lo que se obtiene al mismo tiempo la decimación de la imagen filtrada. En la siguiente sección se describe este algoritmo.

### 4.1.1 Formación de la pirámide Gaussiana modificada

El algoritmo para obtener la estructura Gaussiana de  $N$  niveles para una imagen, es el siguiente:

1. **Inicialización.** Sea  $k = N$ , con  $N$  el nivel de la imagen original, es decir,  $G_k = G_N$ .
2. **Formación del nivel  $k-1$ .** Para cada uno de los bloques de  $m \times m$  pixels contiguos que no se traslapan en el nivel  $k$ , encuentre la media truncada.

$$G_{k-1, i+\frac{1}{2}, j+\frac{1}{2}} = \left[ \frac{G_{k,i,j} + G_{k,i,j+1} + G_{k,i+1,j} + G_{k,i+1,j+1}}{4} \right] \quad (4.1)$$

para  $i, j = 1, 3, \dots, 2^k - 1$ .

donde  $[a]$  es la truncatura de  $a + 0.5$ .

3. **Terminación.** Sea  $k = k-1$ . Si  $k \neq 0$  regrese al paso 2, de otro modo finaliza.

En otras palabras, como se muestra en la figura 4.2, la pirámide Gaussiana se forma de promediar sucesivamente sobre bloques de  $m \times m$  pixels, donde  $m = 2$ , iniciando desde el nivel  $k = N$ .

### 4.1.2 Formación de la pirámide Laplaciana

Una vez que se ha construido la pirámide media, se forma entonces la pirámide de diferencias. Esto se hace, restando dos niveles sucesivos de la primera, bajo el siguiente esquema.

$$\begin{aligned} L_{k,i,j} &= G_{k-1, i+\frac{1}{2}, j+\frac{1}{2}} - G_{k,i,j} \\ L_{k,i,j+1} &= G_{k-1, i+\frac{1}{2}, j+\frac{1}{2}} - G_{k,i,j+1} \\ L_{k,i+1,j} &= G_{k-1, i+\frac{1}{2}, j+\frac{1}{2}} - G_{k,i+1,j} \\ L_{k,i+1,j+1} &= G_{k-1, i+\frac{1}{2}, j+\frac{1}{2}} - G_{k,i+1,j+1} \end{aligned} \quad (4.2)$$

para

$$\begin{aligned} k &= N, N-1, N-2, \dots, 1 \\ i, j &= 1, 3, \dots, 2^k - 1 \end{aligned}$$

Es decir, la diferencia en el nivel  $k$  es igual a la media, menos el valor de cada uno de los pixels que la generó. En la figura 4.3 se presenta una representación esquemática de la formación de un nivel de la pirámide Laplaciana.

Se observa que, como en las imágenes naturales los pixels vecinos están altamente correlacionados, entonces, en la pirámide de diferencias los valores  $L_{k,i,j}$  tenderán a concentrarse alrededor de cero. Por lo tanto, la entropía de la pirámide de diferencias  $H_T$  será mucho menor que la entropía de la imagen original  $H_0$ .

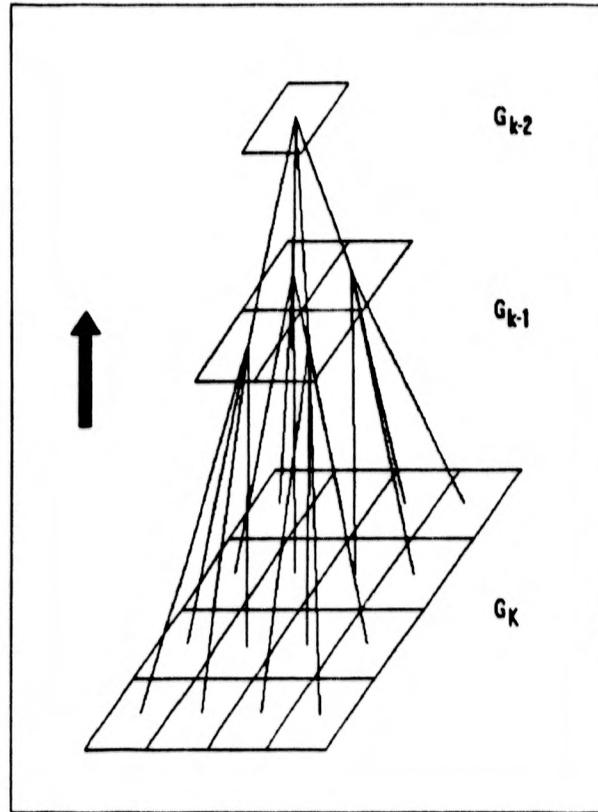


Figura 4.2: Formación de la pirámide Gaussiana, usando bloques de  $2 \times 2$  pixels.

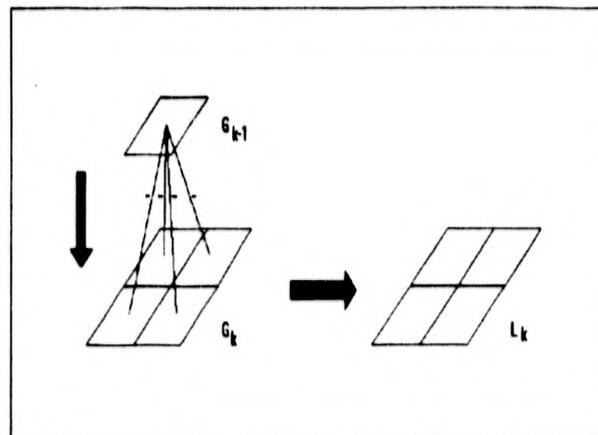


Figura 4.3: Formación de la pirámide Laplaciana, usando bloques de  $2 \times 2$  pixels.

## 4.2 Recuperación de la imagen original

Para lograr la reproducción sin pérdidas de la imagen original, todo lo que se tiene que hacer es, transmitir todos los niveles de la pirámide Laplaciana y seguir los pasos que se describen a continuación:

1. **Inicialización.** Sea  $G_0$  el nivel superior de la pirámide en el nivel 0, y sea  $k = 1$ .
2. **Reconstrucción del nivel  $k$ .** Los valores de los nodos en el nivel  $k$ , se obtienen por medio de las siguientes expresiones

$$\begin{aligned}
 G_{k,i,j} &= G_{k-1,i+\frac{1}{2},j+\frac{1}{2}} - L_{k,i,j} \\
 G_{k,i,j+1} &= G_{k-1,i+\frac{1}{2},j+\frac{1}{2}} - L_{k,i,j+1} \\
 G_{k,i+1,j} &= G_{k-1,i+\frac{1}{2},j+\frac{1}{2}} - L_{k,i+1,j} \\
 G_{k,i+1,j+1} &= G_{k-1,i+\frac{1}{2},j+\frac{1}{2}} - L_{k,i+1,j+1}
 \end{aligned}
 \tag{4.3}$$

para

$$\begin{aligned}
 k &= N, N-1, N-2, \dots, 1 \\
 i, j &= 1, 3, \dots, 2^k - 1
 \end{aligned}$$

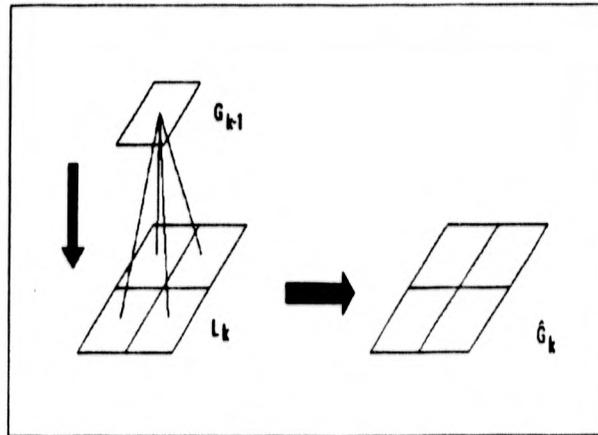


Figura 4.4: Recuperación de un nivel de la pirámide Gaussiana.

que es justamente la operación inversa a la formación de la pirámide de diferencias, como se ilustra en la figura 4.4.

3. **Terminación.** Sea  $k = k+1$ , si  $k \neq N$  regrese al paso 2, de otro modo termina.

A la salida de este proceso, se tendrá la reproducción de la imagen original.

### 4.3 Cuantización vectorial de imágenes representadas mediante una estructura piramidal

Si se cuantizan los valores de los píxels de la pirámide Laplaciana, se puede reducir substancialmente la entropía de ésta. Aunque se introducen errores de cuantización, estos se pueden transmitir progresivamente para eliminar las degradaciones perceptibles al ojo humano.

La teoría de la tasa de la distorsión, indica que se obtiene un mejor desempeño codificando vectores que escalares, inclusive, para el caso de fuentes sin memoria. Más aún, una fuente de datos bien definida, puede comprimirse arbitrariamente cercana a este límite, conforme el tamaño de los bloques codificados se aproxima a infinito.

En nuestro esquema se plantea aplicar cuantización vectorial a la estructura de datos Laplaciana, sobre bloques de píxels de tamaño  $2 \times 2$  y  $4 \times 4$ , en una base nivel por nivel. En este algoritmo, los errores de cuantización producidos en un nivel  $k$ , se incorporan al nivel  $k+1$ , para reducir los efectos de la distorsión introducida. Finalmente, los errores generados en el nivel más alto de la pirámide se codifican usando un código de Huffman, para asegurar el recobrado sin pérdidas de la imagen original.

Para diseñar el libro de códigos y hacer posible la cuantización vectorial, se utiliza el algoritmo LBG en cada etapa en base al siguiente procedimiento:

1. Se divide la imagen del nivel  $k$  en vectores de tamaño  $2 \times 2$  ó  $4 \times 4$  píxels. Todos los vectores que se obtienen se toman como la secuencia de entrenamiento.
2. Se calcula el centroide de la secuencia de entrenamiento, el cual constituye el único vector del libro de códigos de tamaño cero.

Para obtener los libros de código de tamaño  $n = 2, 4$ , hasta  $N$ ,

3. Se calcula la distorsión final que resulta de codificar la imagen, usando el libro de códigos de tamaño  $n/2$ . Si ésta es cero o menor o igual que la distorsión deseada, se detiene el algoritmo con el diccionario resultante. En caso contrario,
4. Se distorsiona cada uno de los vectores del libro de tamaño  $n/2$ , con lo que se obtienen dos nuevos vectores por cada uno de ellos, lo que resulta en el diccionario de tamaño  $n$ .
5. Se mapean todos los vectores de entrenamiento, dentro de las regiones definidas por los vectores del código, en base al criterio de error MSE.
6. Se calcula el centroide de los vectores mapeados en cada una de las regiones de la partición. Los centroides resultantes serán las nuevas palabras del diccionario de tamaño  $n$ , se incrementa  $n$  y se regresa al paso 3.

Una vez que se tiene el libro de códigos de tamaño  $N$  para la imagen del nivel  $k$ , se cuantiza vectorialmente ésta última. Después, se codifican usando un código de Huffman tanto el libro de códigos como las direcciones resultantes de los vectores, y se transmiten a través del canal.

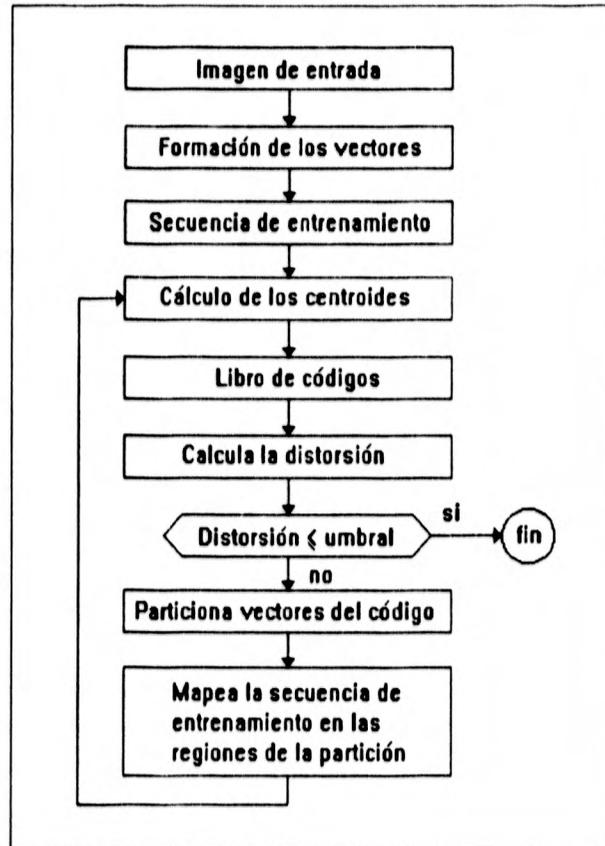


Figura 4.5: Algoritmo para el diseño del libro de códigos.

En la figura 4.5 se muestra el algoritmo utilizado para el diseño del libro de códigos.

## 4.4 Transmisión progresiva

Los pasos que se siguen para la transmisión progresiva de la imagen se describen a continuación:

1. Se genera la pirámide Laplaciana usando las ecuaciones (4.1) y (4.2).

2. **Inicialización.** Se transmiten los primeros tres niveles (de menor resolución) codificados con un código Huffman.
3. Para  $k = 3$ , se cuantiza vectorialmente la imagen  $L_k$  y se transmiten el libro de códigos y los índices de los vectores de reproducción, codificados con un código de Huffman.
4. **Transmisión del error de cuantización.** Para los niveles  $K = 4, 5, \dots, N$ , se aplican iterativamente los pasos (a) y (b)

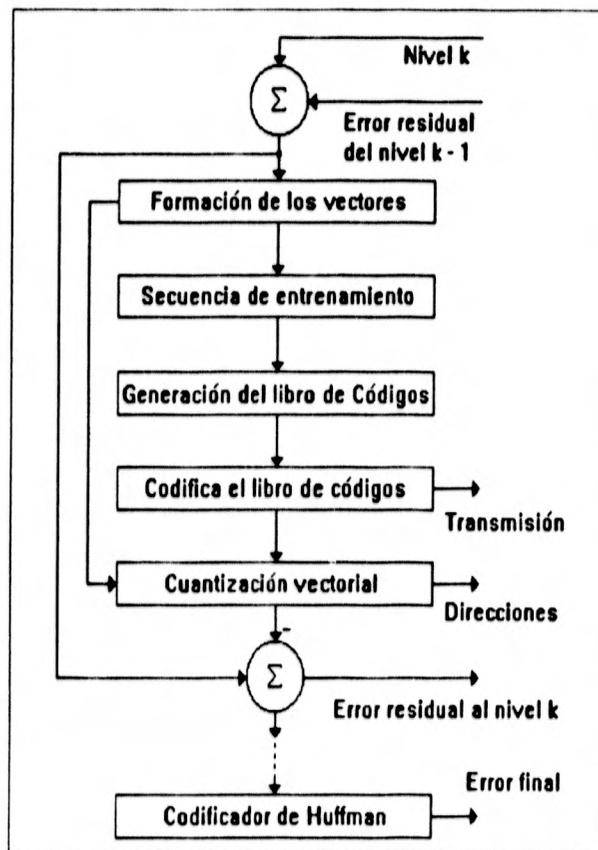


Figura 4.6: Pasos involucrados en la transmisión progresiva aplicando cuantización vectorial a la pirámide Laplaciana.

- (a) Se obtiene la imagen de error generada por la cuantización vectorial en el nivel  $k-1$ , y se expande al doble de su tamaño, usando repetición de pixels, para sumársela a la imagen Laplaciana del nivel  $k$ ,  $L_k$ , con lo que se obtiene  $L_{k+}$ .

(b) Se cuantiza vectorialmente la imagen  $L_{k+}$  y se transmiten el libro de códigos y los índices de los vectores de reproducción, codificados con un código de Huffman. Si  $k \leq N$ , haga  $k = k+1$  y regrese a (a), en caso contrario termina.

5. **Codificación del error residual.** Se obtiene el error residual en el nivel  $k = N$ , y se usa un codificador de Huffman, operando en una base pixel a pixel, para codificarlo sin pérdidas.

En la figura 4.6 se muestran los pasos involucrados en la transmisión progresiva de imágenes aplicando cuantización vectorial a la pirámide Laplaciana. Como se puede ver del diagrama, el error residual del nivel  $k-1$  se suma al nivel  $k$  antes de la cuantización vectorial, con lo que se puede recobrar sin pérdidas cada uno de los niveles.

## 4.5 Codificación del error residual

El codificador de Huffman se ha usado tradicionalmente como un codificador final en una variedad de esquemas de compresión. Como se mencionó en el capítulo 2, este código de longitud variable asigna menor cantidad de bits a aquellos símbolos que tienen mayor probabilidad de ocurrencia y más bits a aquellos con poca probabilidad de presentarse, con esto, se obtiene una reducción en la tasa promedio de bits de los datos codificados.

En la figura 4.7 se muestra el histograma de una imagen típica de error, que resulta de la descomposición en múltiples resoluciones. Como se puede ver, los datos que están alrededor de cero tienen una probabilidad de ocurrencia alta, mientras que conforme se alejan de este valor, disminuye significativamente su probabilidad de aparición. Esta característica del histograma se asemeja a una distribución exponencial con varianza reducida de la forma indicada por la ecuación (4.4) y nos lleva a pensar en aplicar el código de Huffman para codificar sin pérdidas el error residual.

$$p(E) = \frac{1}{\sqrt{2}\sigma_E} \exp\left(\frac{-\sqrt{2}|E|}{\sigma_E}\right) \quad (4.4)$$

La construcción del código de Huffman se lleva a cabo usando el algoritmo descrito en la sección 2.5, tanto para codificar la imagen de error residual, como para codificar el libro de códigos y las direcciones de los vectores de reproducción que se generan durante la cuantización vectorial de los niveles de la pirámide Laplaciana.

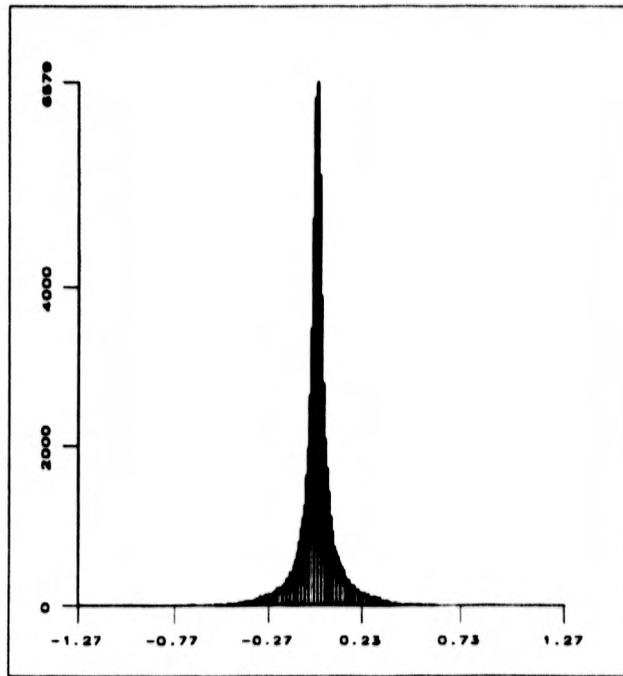


Figura 4.7: Forma típica del histograma de la imagen de error residual.

## Capítulo 5

# RESULTADOS

Una vez establecido el modelo de codificación a utilizar, en este capítulo nos enfocaremos a evaluar el desempeño del mismo. Para este fin, se consideran las medidas objetivas de la distorsión, el error cuadrático medio normalizado (NMSE) porcentual, la relación señal a ruido pico (RSRP) y el error cuadrático medio (MSE), definidas en el capítulo 2.



Figura 5.1: Imágenes originales de prueba de tamaño  $256 \times 256$  a 8 bpp (a) Lena256. (b) Miss América256.

La evaluación subjetiva del algoritmo, se hace observando la imagen codificada, a una distancia de cinco veces el tamaño de la diagonal de la misma. A esta distancia, un observador puede determinar la calidad de la imagen en una escala que va de excelente, muy buena, regular, mala, a muy mala.

Para llevar a cabo las pruebas, se consideran las imágenes, con variación en la escala de gris, Lena256 y Miss América256 (trama), que son de tamaño  $256 \times$

256 pixels, codificadas en 8 bits. Estas imágenes se muestran en la figura 5.1 (a) y (b) respectivamente. La razón que justifica la selección de este par de imágenes, radica en su diferente contenido de frecuencias. Para la primera de ellas, los pixels se encuentran distribuidos a lo largo de la escala de gris, mientras que para Miss América256, los pixels están concentrados en una banda estrecha de valores. Esto se ilustra en las figuras 5.2 (a) y (b). Como se puede ver, las entropías correspondientes de estas imágenes son de 7.638 bits/pixels y 5.971 bits/pixels, lo cual nos indica el límite inferior de la tasa, para la transmisión sin pérdidas.

La codificación de cada una de las imágenes se hace de la siguiente manera:

- Se obtiene la pirámide Laplaciana de 8 niveles.
- La codificación se puede hacer de dos formas
  - Se codifica con un código de Huffman la pirámide completa, en una base nivel por nivel, o bien,
  - Se cuantizan vectorialmente los niveles del 3 al 8 en bloques de  $2 \times 2$  o de  $4 \times 4$ , siguiendo el procedimiento establecido en el apartado 4.3, mientras que los niveles 0, 1 y 2, así como los libros de códigos y las direcciones de los vectores, se codifican con un código de Huffman.
- Se efectúa la transmisión de la imagen codificada.

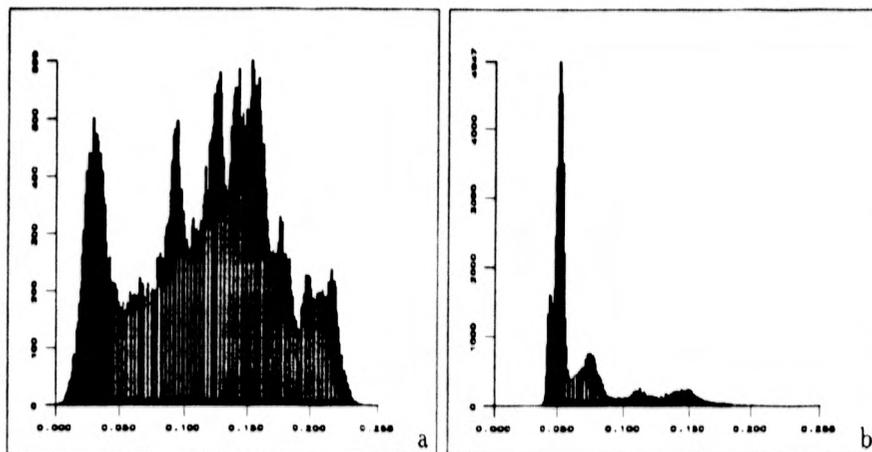


Figura 5.2: Histogramas de las imágenes originales de prueba (a) Lena256.  $H=7.638$ ,  $\sigma^2=3057.27$ . (b) Miss América256.  $H=5.971$ ,  $\sigma^2=1070.14$ .

## 5.1 Presentación de los resultados

Para cada imagen de prueba, se obtienen tres tablas, las cuales contienen la tasa de bit que representa cada nivel de la pirámide transmitido, la tasa de bit acumulada y las métricas de la distorsión. Todas estas estadísticas se obtienen para cada nivel, e indican la tasa de transmisión, el error de codificación y la calidad, que se obtienen, si se representa la imagen original usando el nivel correspondiente, es decir, si la imagen al nivel  $k$  se expande y se toma como la versión final.

La primera tabla, contiene los resultados que se obtienen cuando la estructura de datos se codifica sólo con un código de Huffman, la segunda, contiene los resultados generados cuando se usa cuantización vectorial con bloques de  $2 \times 2$ , sobre los niveles del 3 al 8 de la pirámide, y la tercera, contiene los resultados de aplicar cuantización vectorial con bloques de  $4 \times 4$ . Para ambas imágenes de prueba se presentan únicamente los niveles del 5 al 8, primero, porque los niveles más bajos de la pirámide ya no se pueden visualizar fácilmente y, segundo, porque el nivel más alto de la pirámide es numéricamente idéntico, pixel a pixel, a la imagen original.

También, se presentan los histogramas tanto para la pirámide Laplaciana, como para su versión codificada vectorialmente. Aquí, se incluyen el valor de la entropía por cada nivel, la cual será utilizada para medir la eficiencia del algoritmo de codificación. Finalmente, se presentan las gráficas del error NMSE porcentual y la relación señal a ruido pico, contra la tasa de bit total transmitida.

### 5.1.1 Lena256

NIVEL k	$TASA_{HUFF}(k)$ BPP	$TASA_{ACUM}(k)$ BPP	NMSE (%)	RSRP (DB)	MSE
0	0.0001068	0.0001068	17.9637	13.2607	3069.043
1	0.0002441	0.0003509	15.7495	13.8320	2690.744
2	0.0010986	0.0014495	14.2380	14.2702	2432.515
3	0.0054932	0.0069427	10.2184	15.7109	1745.777
4	0.0245361	0.0314788	6.2565	17.8414	1068.898
5	0.0964355	0.1279143	3.7961	20.0113	648.552
6	0.3662109	0.4941252	1.8829	23.0564	321.686
7	1.3315430	1.8256682	0.6800	27.4798	116.169
8	4.8419189	6.6675871	0.0000	$\infty$	0.0000

Tabla 5.1 (a): Resultados de la simulación para la imagen de prueba Lena256, aplicando codificación de Huffman sobre cada uno de los niveles de la estructura de datos.

NIVEL k / N	$TASA_{HUFF}(k)$ (BPP)		$TASA_{ACUM}(k)$ BPP	NMSE (%)	RSRP (DB)	MSE
	LIBRO	DIRECC				
0	0.0001068		0.0001068	17.9637	13.2607	3069.043
1	0.0002441		0.0003509	15.7495	13.8320	2690.744
2	0.0010986		0.0014495	14.2380	14.2702	2432.515
3 / 2	0.0004883	0.0003662	0.0023040	11.8224	15.0776	2019.813
4 / 4	0.0009765	0.0019531	0.0052337	8.5606	16.4796	1462.555
5 / 4	0.0010986	0.0078125	0.0141448	6.2079	17.8753	1060.592
6 / 8	0.0024414	0.041137	0.0577239	3.2862	20.6378	561.428
7 / 8	0.0023193	0.1497803	0.2098235	1.6568	23.6120	283.059
8 / 8	0.0021972	0.5754395	0.7874606	0.6276	27.8280	107.220
9	5.0687256		5.8561862	0.0000	$\infty$	0.0000

Tabla 5.1 (b): Resultados de la simulación para la imagen de prueba Lena256, aplicando cuantización vectorial, sobre los niveles del 3 al 8, con bloques de  $2 \times 2$ , usando libros de código de tamaño N.

NIVEL k / N	$TASA_{HUFF}(k)$ (BPP)		$TASA_{ACUM}(k)$ BPP	NMSE (%)	RSRP (DB)	MSE
	LIBRO	DIRECC				
0	0.0001068		0.0001068	17.9637	13.2607	3069.043
1	0.0002441		0.0003509	15.7495	13.8320	2690.744
2	0.0010986		0.0014495	14.2380	14.2702	2432.515
3 / 2	0.0023193	0.0001220	0.0038908	11.4885	15.2020	1962.779
4 / 4	0.0052490	0.0006103	0.0097501	9.1939	16.1697	1570.754
5 / 8	0.0117187	0.0028076	0.0242764	6.4741	17.6929	1106.082
6 / 16	0.0243582	0.0147705	0.0637051	4.0367	19.7444	689.664
7 / 32	0.0519033	0.0723876	0.186996	2.0407	22.7070	348.640
8 / 64	0.1043701	0.3267822	0.6181483	0.6879	27.4292	117.542
9	5.1905518		5.8087001	0.0000	$\infty$	0.0000

Tabla 5.1 (c): Resultados de la simulación para la imagen de prueba Lena256, aplicando cuantización vectorial, sobre los niveles del 3 al 8, con bloques de  $4 \times 4$ , usando libros de código de tamaño N.

### 5.1.2 Miss América256

NIVEL k	TASA <sub>HUFF</sub> (k) BPP	TASA <sub>ACUM</sub> (k) BPP	NMSE (%)	RSRP (DB)	MSE
0	0.0001068	0.0001068	17.9637	13.2607	3069.043
1	0.0002441	0.0003509	15.7495	13.8320	2690.744
2	0.0010986	0.0014495	14.2380	14.2702	2432.515
3	0.0046386	0.0060881	5.2304	23.1386	315.657
4	0.0186767	0.0247648	2.7080	25.9975	163.429
5	0.125000	0.1497648	1.3406	29.0511	80.904
6	0.2412109	0.3909757	0.6195	32.4037	37.385
7	0.8669433	1.257919	0.2289	36.7272	13.815
8	3.5732422	4.8311612	0.0000	∞	0.0000

Tabla 5.2 (a): Resultados de la simulación para la imagen de prueba Miss América256 (trama1), aplicando codificación de Huffman sobre cada uno de los niveles de la estructura de datos.

NIVEL k / N	TASA <sub>HUFF</sub> (k) (BPP)		TASA <sub>ACUM</sub> (k) BPP	NMSE (%)	RSRP (DB)	MSE
	LIBRO	DIRECC				
0	0.0001068		0.0001068	17.9637	13.2607	3069.043
1	0.0002441		0.0003509	15.7495	13.8320	2690.744
2	0.0010986		0.0014495	14.2380	14.2702	2432.515
3 / 2	0.0003662	0.0003662	0.0021819	9.5625	20.5182	577.103
4 / 4	0.0009765	0.0017089	0.0048673	5.5375	22.8909	334.191
5 / 4	0.0009765	0.0063476	0.0121914	3.0799	25.4386	185.873
6 / 8	0.0023193	0.0404052	0.0549159	1.3746	28.9421	82.959
7 / 8	0.0021972	0.1516113	0.2087244	0.6584	32.1384	39.737
8 / 8	0.0021972	0.5662841	0.7772057	0.2590	36.1908	15.631
9	3.670166		4.4473717	0.0000	∞	0.0000

Tabla 5.2 (b): Resultados de la simulación para la imagen de prueba Miss América256 (trama 1), aplicando cuantización vectorial, sobre los niveles del 3 al 8, con bloques de  $2 \times 2$ , usando libros de código de tamaño N.

NIVEL k / N	TASA <sub>HUFF</sub> (k) (BPP)		TASA <sub>ACUM</sub> (k) BPP	NMSE (%)	RSRP (DB)	MSE
	LIBRO	DIRECC				
0	0.0001068		0.0001068	17.9637	13.2607	3069.043
1	0.0002441		0.0003509	15.7495	13.8320	2690.744
2	0.0010986		0.0014495	14.2380	14.2702	2432.515
3 / 2	0.0021972	0.0001220	0.0037687	8.0645	21.2582	486.695
4 / 4	0.0050048	0.0004882	0.0092617	5.3265	23.0595	321.459
5 / 8	0.0111083	0.0024414	0.0228114	2.7839	25.8773	168.012
6 / 16	0.0220947	0.0129394	0.0578455	1.4738	28.6395	88.945
7 / 32	0.043456	0.0653076	0.1666101	0.6871	31.9538	41.466
8 / 64	0.0847167	0.3127441	0.5640709	0.2366	36.5842	14.277
9	3.729126		4.2931969	0.0000	∞	0.0000

Tabla 5.3 (c): Resultados de la simulación para la imagen de prueba Miss América256 (trama 1), aplicando cuantización vectorial, sobre los niveles del 3 al 8, con bloques de  $4 \times 4$ , usando libros de código de tamaño N.

## 5.2 Análisis de los resultados

De los resultados presentados se pueden hacer las siguientes observaciones:

- De las tablas 5.1 y 5.2, mostradas, para Lena256 y Miss América256, se puede apreciar que una imagen cuyos pixels se encuentran distribuidos sobre toda la escala de niveles de gris, se puede codificar a 0.61 bpp, con una RSRP de 27.4 dB y un NMSE de 0.68 por ciento, con una calidad de reconstrucción muy buena. Por otra parte, una imagen con poca variación frecuencial, se puede codificar a una tasa de 0.56 bpp, con una RSRP de 36.5 dB y un NMSE de 0.23 por ciento, con una excelente calidad de reconstrucción. Esto se puede corroborar, viendo las imágenes sintéticas que se muestran en las figuras 5.3 y 5.7, respectivamente.
- Las figuras 5.4 y 5.8 corresponden a las expansiones de los niveles 7, 6 y 5 de la pirámide Gaussiana recobrada. Estas expansiones ponen de manifiesto el contenido de información que se obtiene de los niveles de la estructura multi-resolución de los datos.
- Las figuras 5.5 y 5.9, muestran que se obtiene una buena reducción de la entropía, como resultado de aplicar cuantización vectorial sobre los niveles del 3 al 8, de la pirámide Laplaciana. Esto nos permite reducir la tasa de codificación, aún para recobrar la imagen sin pérdidas.

Si calculamos la entropía de la pirámide Laplaciana original, obtenida en los dos casos, empleando la ecuación (3.11) con  $k' = N - k$ , ésta resulta ser igual a 6.63 y 4.71 bits/pixel respectivamente, lo cual representa la primer ventaja de esta representación a múltiples resoluciones, ya que las correspondientes entropías originales son de 7.63 y 5.97 bits/pixel. Como puede verse en las tablas, para el caso del codificador de Huffman, la tasa de transmisión es ligeramente mayor (6.66 y 4.83 bpp), mientras que para los casos en los que se aplica cuantización vectorial, el hecho de sumar progresivamente el error que se genera en un nivel, al siguiente nivel, hace que la ecuación (3.11) ya no sea válida. Esto, más que representar un inconveniente, representa una ventaja, ya que la transmisión se puede realizar a una tasa de bit, aún menor a la de la estructura de los datos utilizada. Esto se debe, principalmente, a que el error de cuantización se corrige progresivamente.

Como resultado de aplicar el algoritmo de codificación, se puede obtener un ahorro en la tasa de bit para transmisión sin pérdidas, con respecto a la entropía de la imagen original, de 1.83 bpp, para Lena256, y 1.68 bpp, para Miss América256, en el mejor de los casos.

Otra característica que podemos apreciar, es que la entropía aumenta conforme se avanza hacia los niveles inferiores, sin embargo, esto no repercute significativamente en la entropía final de la estructura, porque como se ve de

la ecuación (3.11), el aporte de cada nivel inferior va ponderado por un factor, el cual representa una relación de 4 a 1 de un nivel a otro.

- Para los dos casos en los que se aplica cuantización vectorial, la transmisión del error residual (que corresponde al nivel 9), el cual representa la mayor parte de la tasa de bit, garantiza la reconstrucción sin pérdidas de la **imagen** original.
- De las gráficas mostradas en las figuras 5.6 y 5.10, se puede apreciar que se obtiene una mayor compresión cuando se aplica cuantización vectorial sobre bloques de  $4 \times 4$  pixels que cuando se usan bloques de  $2 \times 2$ , mientras el error se mantiene dentro de un rango adecuado. Esto se ajusta a lo establecido por la teoría de la tasa de la distorsión, que indica, que se puede reducir la tasa de bit tan cercana a la entropía de la imagen, mientras se mantiene una cierta distorsión, conforme el tamaño de los bloques procesados es más grande.
- Para el caso del cuantizador vectorial, la tasas de los **códigos** generados, que se obtienen con la ecuación (2.10), se encuentran entre 0.5 y 0.75, cuando se usan bloques de  $2 \times 2$ , y entre 0.0625 y 0.375, para bloques de  $4 \times 4$ .
- En cuanto al codificador de Huffman aplicado sobre la imagen de error residual, el porcentaje de compresión alcanzado está alrededor de 99.4 por ciento para Lena<sup>256</sup>, y 98.8 por ciento para Miss América<sup>256</sup>. Esto es de esperarse, porque la distribución de los pixels está muy concentrada alrededor de cero, y por el hecho de que el código de Huffinan tiende a obtener un mejor desempeño cuando el conjunto de símbolos a codificar es grande.



Figura 5.3: Imágenes de los niveles 8 al 5, usando la estructura multi-resolución. (a) Pirámide Gaussiana recobrada, imágenes codificadas a 0.618, 0.186, 0.0637 y 0.024 bpp respectivamente. (b) Pirámide Laplaciana. Cuatro niveles usados durante la codificación. (c) Versiones cuantizadas vectorialmente y transmitidas, de la pirámide mostrada en (b).

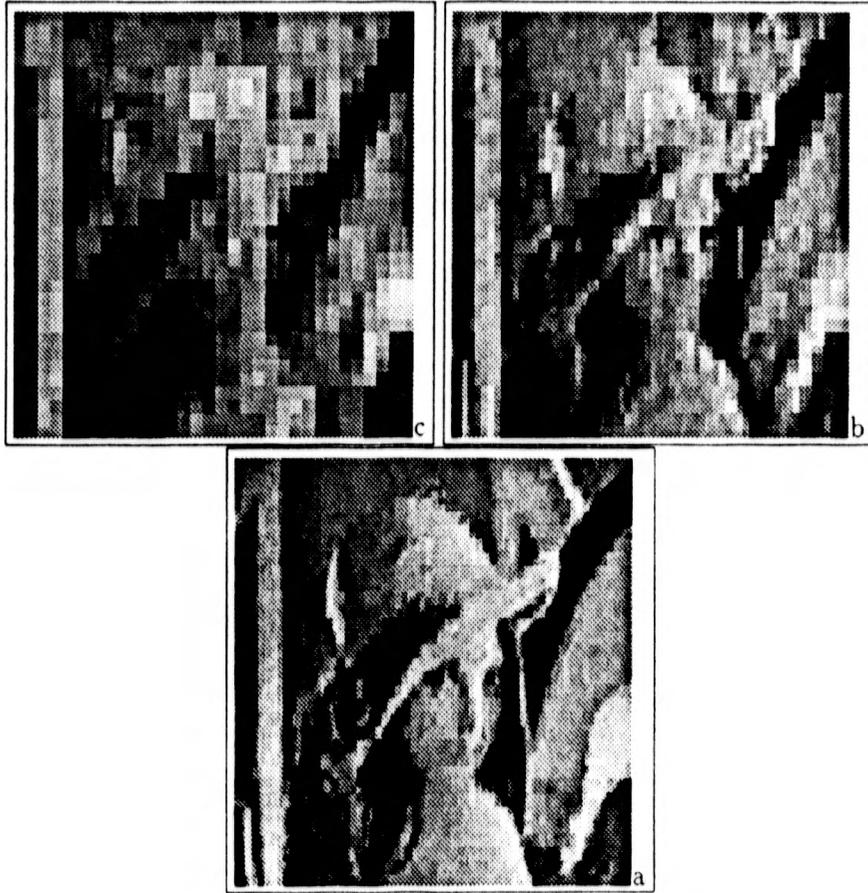


Figura 5.4: Imágenes de los niveles 7 al 5 de la pirámide Gaussiana recobrada, expandidos hasta un tamaño  $256 \times 256$ . (a) Nivel 7 codificado a 0.186 bpp. (b) Nivel 6 codificado a 0.063 bpp. (c) Nivel 5 codificado a 0.024 bpp.

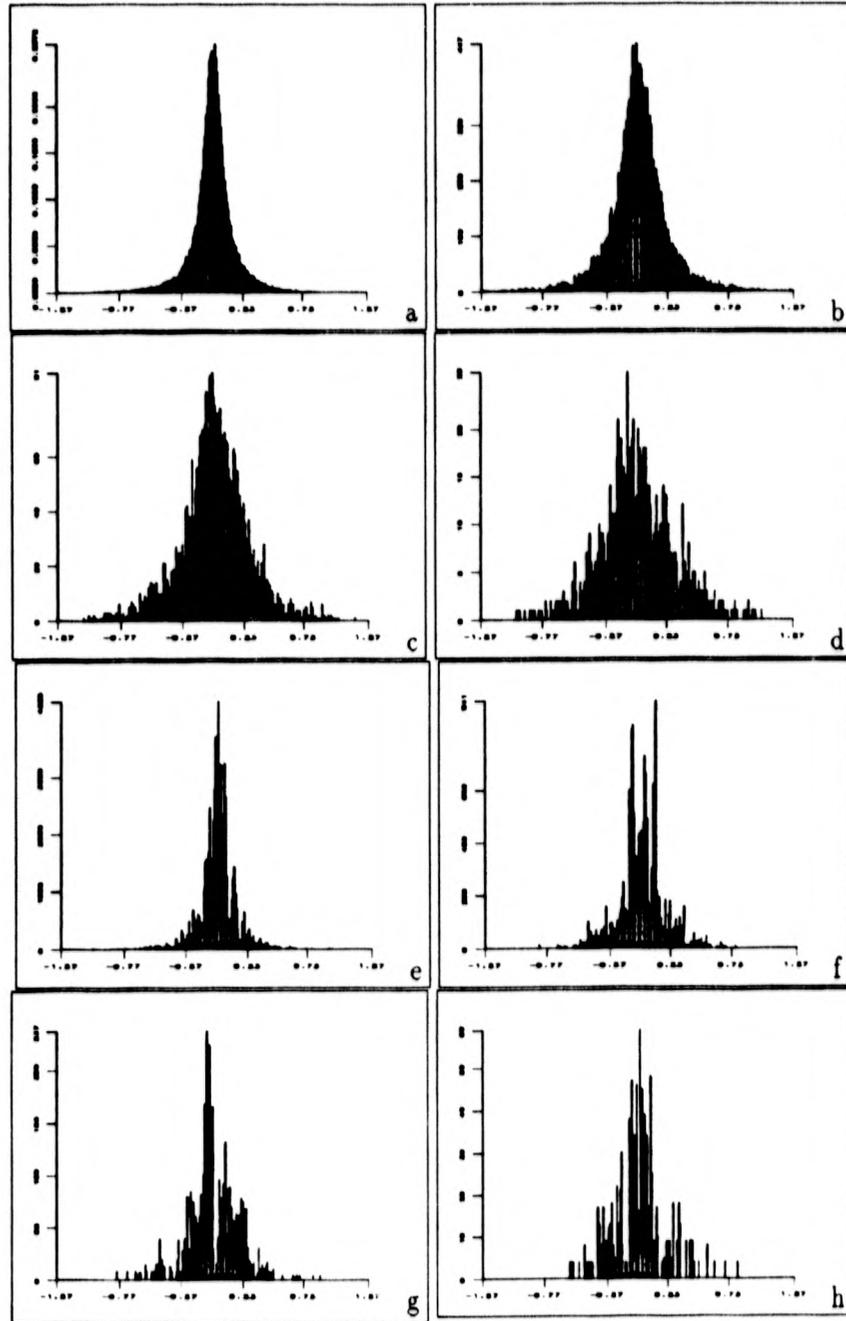


Figura 5.5: Histogramas de la pirámide Laplaciana para los niveles del 8 al 5, incluido el error de codificación del nivel anterior: (a)  $H=6.02$ ,  $\sigma^2=348.343$ . (b)  $H=6.49$ ,  $\sigma^2=573.173$  (c)  $H=6.75$ ,  $\sigma^2=784.321$ . (d)  $H=6.80$ ,  $\sigma^2=923.487$ . Pirámide Laplaciana codificada vectorialmente con bloques de  $4 \times 4$ : (e)  $H=5.63$ ,  $\sigma^2=228.907$  (f)  $H=5.86$ ,  $\sigma^2=338.206$  (g)  $H=5.95$ ,  $\sigma^2=411.095$  (h)  $H=5.68$ ,  $\sigma^2=464.888$ . La entropía se da en bits/píxel.

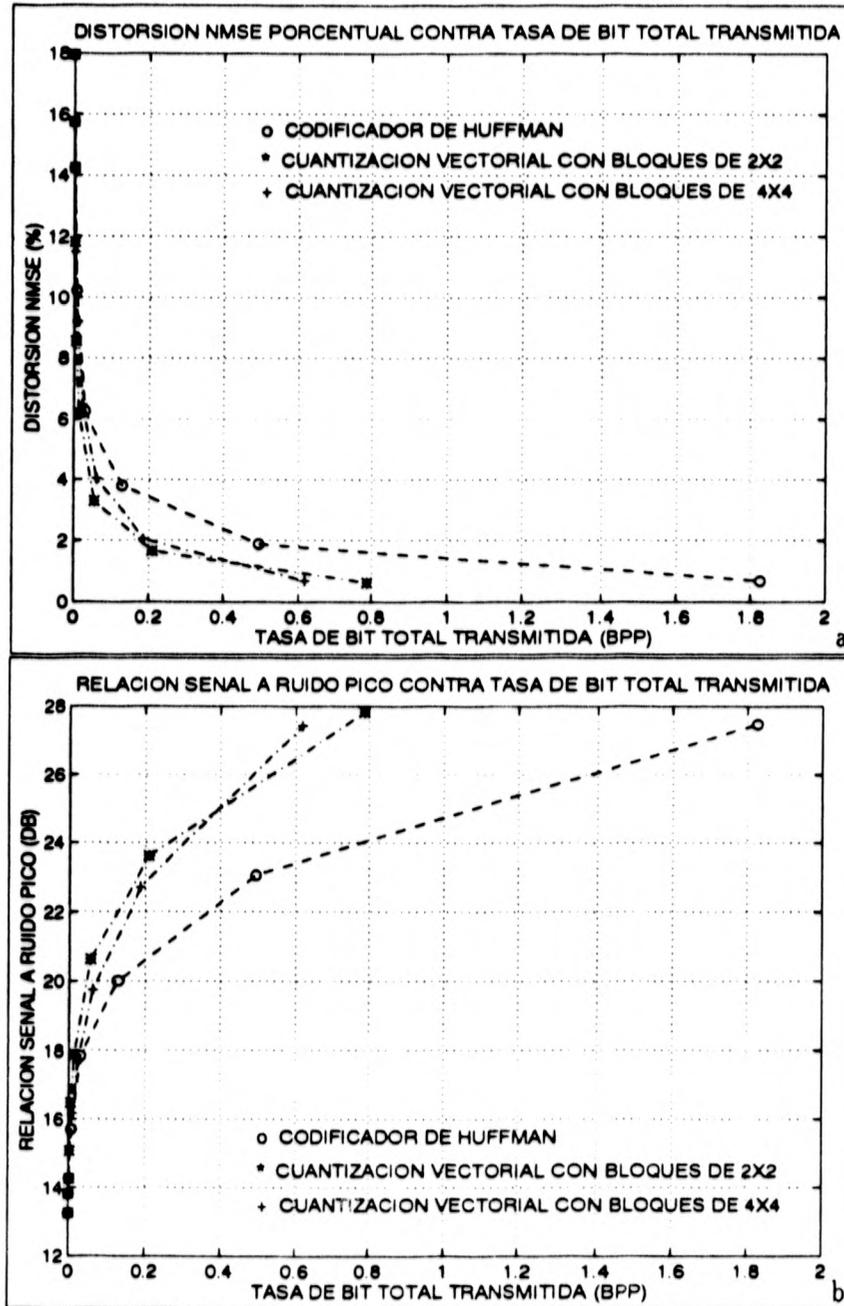


Figura 5.6: NMSE porcentual y RSRP (en dB) contra la tasa de bit total transmitida, para la imagen de prueba Lena256.

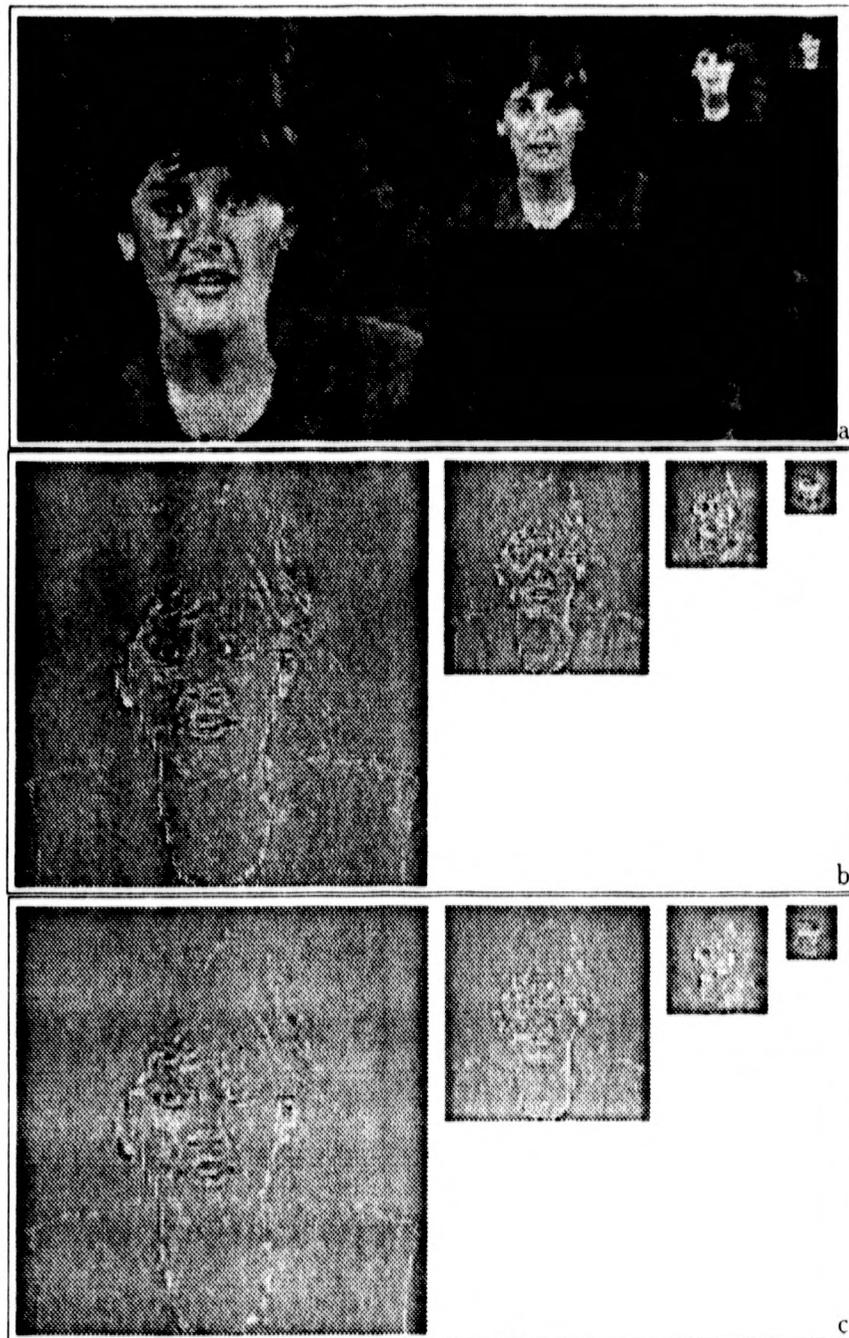


Figura 5.7: Imágenes de los niveles 8 al 5, usando la estructura multi-resolución. (a) Pirámide Gaussiana recobrada, imágenes codificadas a 0.564, 0.166, 0.057 y 0.022 bpp respectivamente. (b) Pirámide Laplaciana. Cuatro niveles usados durante la codificación. (c) Versiones cuantizadas vectorialmente y transmitidas, de la pirámide mostrada en (b).

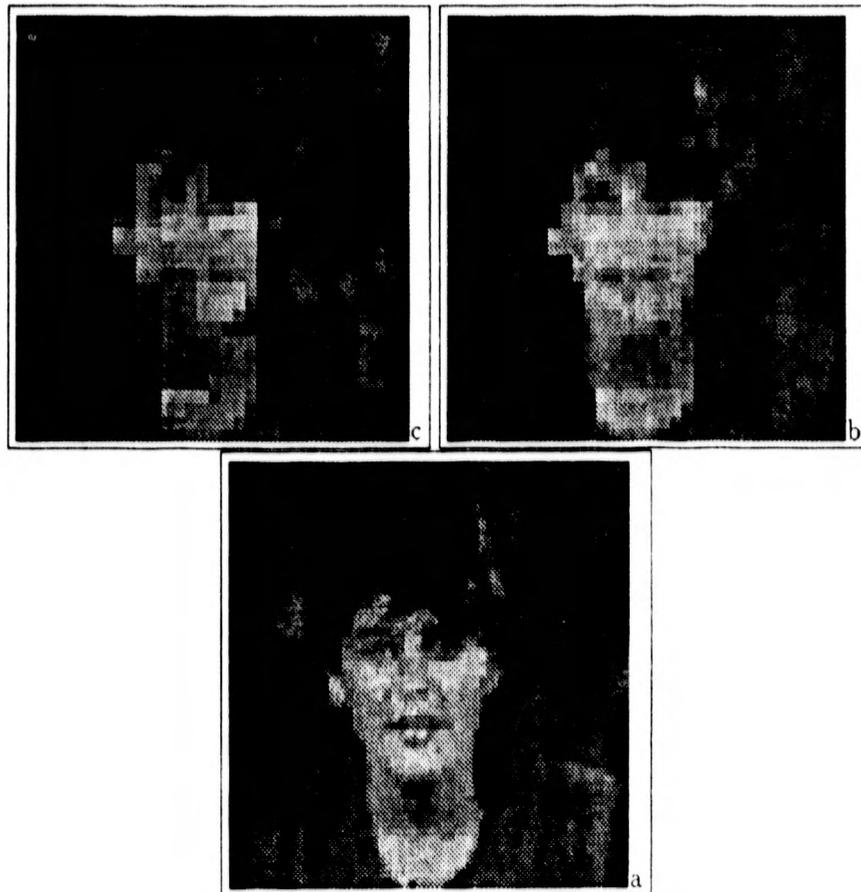


Figura 5.8: Imágenes de los niveles 7 al 5 de la pirámide Gaussiana recobrada, expandidos hasta un tamaño  $256 \times 256$ . (a) Nivel 7 codificado a 0.166 bpp. (b) Nivel 6 codificado a 0.057 bpp. (c) Nivel 5 codificado a 0.022 bpp.

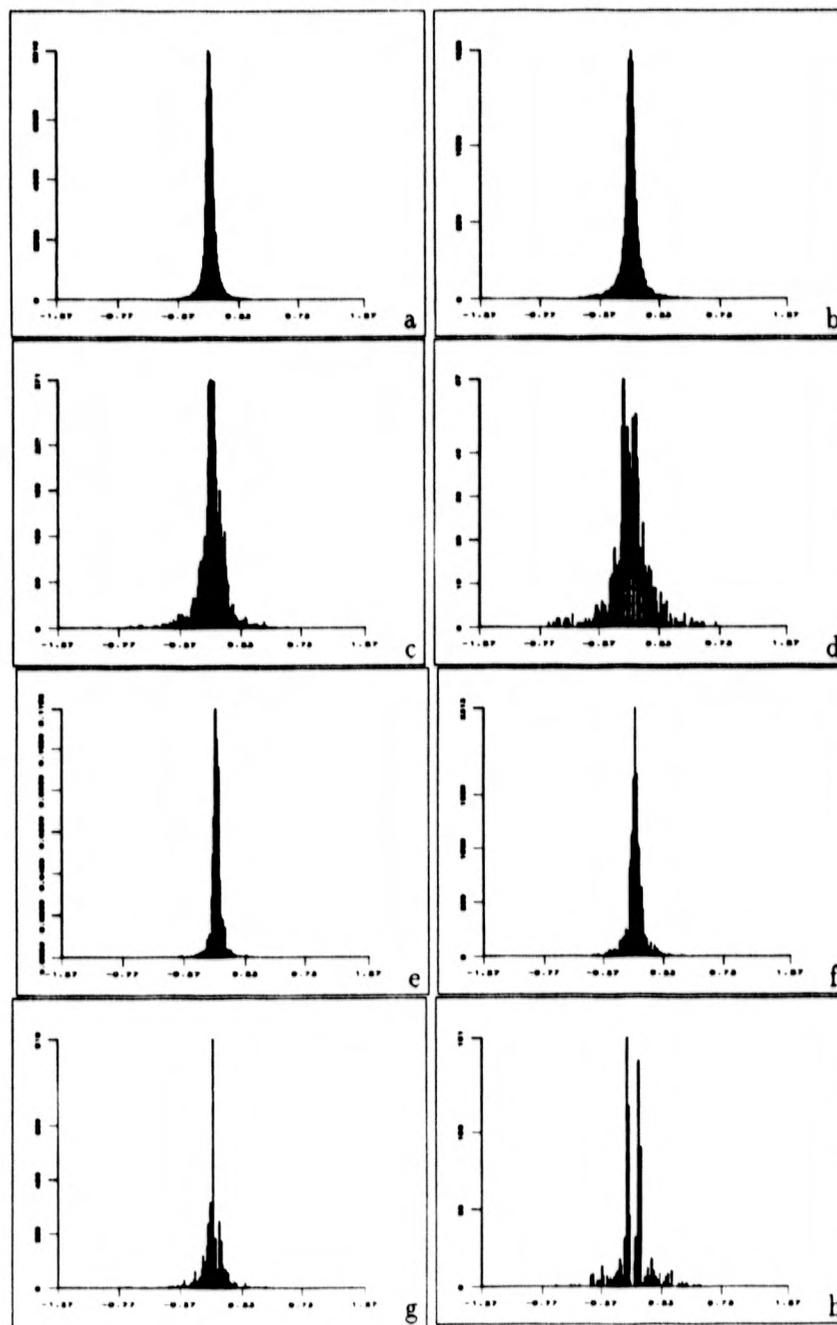


Figura 5.9: Histogramas de la pirámide Laplaciana para los niveles del 8 al 5, incluido el error de codificación del nivel anterior: (a)  $H=4.33$ ,  $\sigma^2=41.278$ . (b)  $H=4.78$ ,  $\sigma^2=74.89$  (c)  $H=5.28$ ,  $\sigma^2=129.727$ . (d)  $H=5.68$ ,  $\sigma^2=240.450$ . Pirámide Laplaciana codificada vectorialmente con bloques de  $4 \times 4$ : (e)  $H=3.93$ ,  $\sigma^2=26.927$  (f)  $H=4.46$ ,  $\sigma^2=47.155$  (g)  $H=4.56$ ,  $\sigma^2=78.051$  (h)  $H=4.44$ ,  $\sigma^2=153.799$ . La entropía se da en bits/pixel.

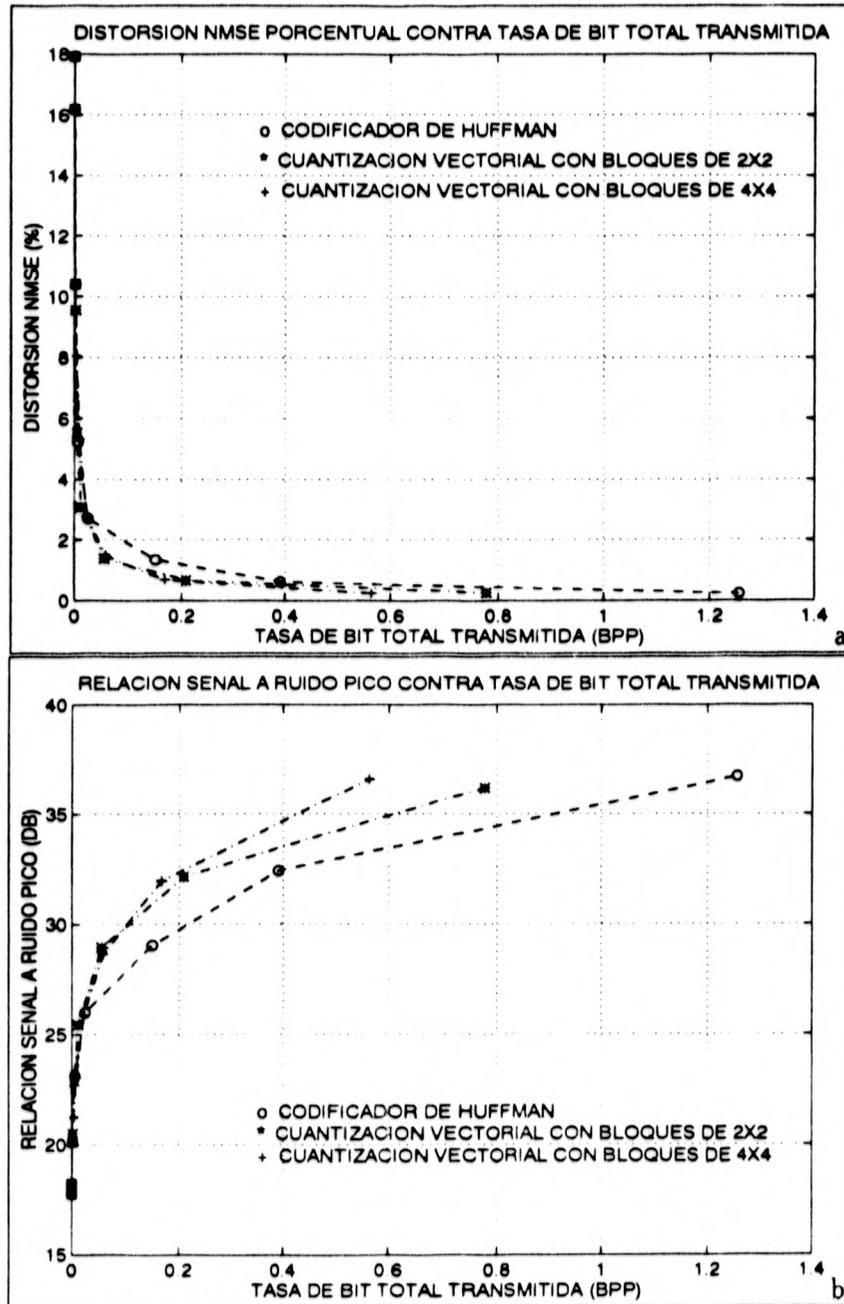


Figura 5.10: NMSE porcentual y RSRP (en dB) contra la tasa de bit total transmitida, para la imagen de prueba Miss América256.

## Capítulo 6

# CONCLUSIONES

La representación de imágenes digitales, mediante una estructura de datos multi-resolución, es una alternativa, que posee muchas características atractivas para el procesamiento digital. Una de las propiedades más interesantes de este esquema, es que resulta adecuado para la transmisión progresiva, en la cual, primero se transmite una parte muy pequeña de los datos codificados, para tener en el receptor una réplica muy burda del contenido de la imagen, después, ésta se va refinando gradualmente, conforme las transmisiones subsecuentes proveen más detalles finos de la imagen. Los cálculos empleados para obtener esta representación son muy simples, y pueden implantarse en una arquitectura de procesamiento en paralelo, para aplicaciones en tiempo real.

Durante el desarrollo de esta tesis, se planteó el objetivo de combinar en un sistema de codificación, la representación a múltiples resoluciones de los datos con la cuantización vectorial. Los resultados obtenidos, indican que este algoritmo logra, al mismo tiempo, la compresión de la información y codificación sin pérdidas, a una tasa de bit menor (en el peor de los casos presentados, 1.6 bits menos), que la entropía de la imagen original. Además, como un esquema de codificación con pérdidas, se alcanzan tasas de compresión considerables (entre 13 y 15 para los casos presentados), con muy buena calidad, objetiva y subjetiva, de la imagen reconstruida, en la cual no se presentan los efectos de bloques.

Se puede decir que este esquema de codificación obtiene mejores resultados, en cuanto a la calidad de la imagen recobrada y a la tasa de transmisión, cuando los pixels están más concentrados alrededor de un cierto valor. Esto se debe a que la principal función de la representación piramidal a múltiples resoluciones, es precisamente, concentrar los datos en una forma Laplaciana, para permitir una mejor codificación de los mismos.

Por otra parte, la aplicación de la cuantización vectorial sobre los diferentes niveles de la estructura piramidal, reduce la entropía de cada uno de ellos y provee una compresión adicional de los datos.

Las tareas en las que se puede utilizar este algoritmo, comprende la transmisión de imágenes fijas sobre canales de baja capacidad, como las líneas telefónicas, para el análisis y procesamiento, así como la reducción de la capacidad de almacenamiento requerido. También, se puede emplear en la transmisión de imágenes en un ambiente multiusuario, en el cual se tenga un cierto número de dispositivos de despliegue con diferentes resoluciones o requerimientos de calidad.

Puedo mencionar, finalmente, que el esquema de codificación utilizado en este proyecto, se puede aplicar para la codificación de secuencias de imágenes en tiempo real. Las pruebas que se han realizado hasta este momento, indican que se pueden obtener tasas de transmisión similares a las mostradas para imágenes fijas, y aún más, éstas se pueden reducir si se explota la redundancia entre tramas, principalmente en secuencias que tienen poca variación frecuencial. En este sentido, un próximo paso que se plantea como posible estudio, sería, aplicar este algoritmo de codificación a secuencias de imágenes, en base a un esquema de codificación entre tramas.

# BIBLIOGRAFIA

- [**AKAHAD92**] Ali N. Akansu y Richard A. Haddad. "Multiresolution Signal Decomposition: Transforms, subbands and wavelets". Academic Press, Inc. 1992.
- [**BARLAU94**] M. Barlaud editor "Wavelets in Image Communication". Elsevier Science B. V. 1994.
- [**BEIGRA85**] Chang-Da Bei y Robert M. Gray. "An Improvement to the Minimum Distortion Encoding Algorithm for Vector Quantization". IEEE Transactions on Communications, Vol. COM-33, No. 10, Oct. 1985, pp 1132 y 1133.
- [**BURADE83**] Peter J. Burt y E. H. Adelson. "The Laplacian Pyramid as a Compact Image Code". IEEE Transactions on Communications, Vol. COM-31, No.4. April 1983., pp 532-540.
- [**CHEGER84**] De-Yuang Cheng, Allen Gersho, et. al. "Fast Search Algorithms for Vector Quantization and Pattern Matching". Proc. ICASSP, Vol. 1, Mar. 1984, pp 9.11.1 - 9.11.4.
- [**CHEGER86**] De-Yuang Cheng, Allen Gersho. "A Fast Codebook Search Algorithm for Nearest-Neighbor Pattern Matching". Proc. ICASSP, Abr. 1986, pp 6.14.1 - 6.14.4.
- [**CROWEB76**] R. E. Crochiere, A. A. Webber y J. L. Flanagan. "Digital Coding of Speech in Subbands". Bell System Tech. J. Vol 55, Octubre 1976, pp 1069 - 1085.
- [**EQUITS87**] William Equitz. "Fast Algorithms for Vector Quantization Picture Coding". Proc. ICASSP., Abr. 1987, pp 18.1.1 - 18.1.4.
- [**FISDIC84**] Thomas R. Fisher y Roy M. Dicharry. "Vector Quantization for Memoryless Gaussian, Gamma and Laplacian Sources". IEEE Transactions on Communications, Vol. COM-32, Sept. 1984. pp 1065-1069.
- [**GERGRA92**] Allen Gersho y Robert M. Gray. "Vector Quantization and Signal Compression". Kluwer Academic Publishers, 1992.
- [**GURORC92**] Gregory C. Gurski, Michael T. Orchard y Andrew W. Hull. "Optimal Linear Filters for Pyramidal Decomposition". ICASSP92 Tomo IV, pp IV-633 - IV-636.
- [**HUABIS92**] C. M. Huang, Q. Bi, G. S. Stiles y R. W. Harris. "Fast Full Search Equivalent Encoding Algorithms for Image Compression Using Vector Quantization". IEEE Transactions on Image Processing, Vol. 1, No. 3, Julio 1992, pp 413 - 416.

- [**HUFFMA52**] David A. Huffman. "A Method for the Construction of Minimum-Redundancy Codes". Proceedings of the I.R.E., Vol. 40, No. 9, pp 1098-1101, Sept. 1952.
- [**LEELIU89**] C. H. Lee y H. T. Liu "Digital Image Coding With High Compression Ratio" Third International Conference on Image Processing and Its Applications. IEEE Warwick, UK, 1989, pp 595 - 598.
- [**LIM90**] Jae S. Lim. "Two-Dimensional Signal and Image Processing". Prentice Hall Signal Processing Series, 1990.
- [**LINBUZ80**] Yoseph Linde, Andrés Buzo, Robert M. Gray. "An Algorithm for Vector Quantizer Design" IEEE Transactions on Communications, Vol. COM-28, Enero de 1980, pp 84-95.
- [**MEYER94**] Yves Meyer. "Wavelets: Algorithms and applications". Society for Industrial and Applied Mathematics, 1994.
- [**NASKIN88**] Nasser M. Nasrabadi y Robert A. King. "Image Coding Usin Vector Quantization: A Review". IEEE Transactions on Communications, Vol. 36, Num. 8, Agosto de 1988, pp 957-971.
- [**NETHAS**] Arun N. Netravali y Barry G. Haskell. "Digital Pictures: Representation and Compression". Plenum Press, July 1989.
- [**PALRAM89**] K. K. Paliwal y V. Ramasubramanian. "Effect of Ordering the Codebook on the Efficiency of the Partial Distance Algorithm for Vector Quantization". IEEE Transactions on Communications. Vol COM-37, No. 5. May. 1989, pp 538 y 539.
- [**PRETEU92**] William H. Press, Saul A. Teukolsky, et. al. "Numerical Recipes in C: The art of Scientific Computing". Second Edition. Cambridge University Press, 1992.
- [**RABJON91**] M. Rabbani y P. W. Jones. "Digital Image Compression Techniques" Bellingham, WA:SPIE Press, 1991.
- [**RAMPAL92**] V. Ramasubramanian y K. K. Paliwal. "An Efficient Aproximation-Elimination Algorithm for Fast Nearest-Neighbor Search". IEEE ICASSP92. 1992, pp I.89 - I.92.
- [**REZMOR87**] Mohammad Reza Soleyman y Salvatore D. Morgera. "A High-Speed Algorithm for Vector Quantization". IEEE ICASSP87. Abr 1987, pp 45.6.1 - 45.6.3.
- [**SOLMOR89**] Mohammad R. Soleymani y Salvatore D. Morgera. "A Fast MMSE Encoding Technique for Vector Quantization". IEEE Transactions on Communications, Vol. 37, No. 6, Junio 1989, pp 656 - 658.
- [**TEKALP95**] A. Murat Tekalp. "Digital Video Processing" Prentice Hall Signal Processing Series, 1995.

[VETTER84] M. Vetterli. "Multi-dimensional Sub-band Coding: Some Theory and Algorithms". IEEE Signal Processing, Vol. 6, Abril 1984, pp 97 - 112.

[VINCEN89] J. M. Vincent. "Application of Arithmetic Coding to Laplacian Pyramid Coding". Third International Conference on Image Processing and Its Applications. IEEE Warwick, UK, 1989, pp 590 - 594.

[WANGOL89] Liming Wang y Morris Goldberg. "Progressive Image Transmission Using Vector Quantization on Images in Pyramid Form". IEEE Transactions on Communications, Vol-37, Num-12, December 1989, pp1339-1349.

[WILLIA91] Ross Neil Williams. "Adaptive Data Compression". Kluwer Academic Publishers, 1991.

[WOONEI86] J. W. Woods y S. D. O'Neil. "Subband Coding of Images". IEEE Trans. on Acoustic Speech and Sig. Proc. Vol ASSP 34, Octubre 1986, pp 1278 - 1288.

[YEH89] C. L. Yeh. "A Lossless Progressive Transmission Technique for Image Browsing". Third International Conference on Image Processing and Its Applications. IEEE Warwick, UK, 1989, pp 599 - 602.