

24

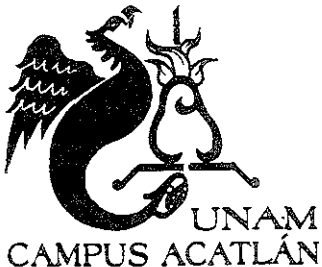


UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ACATLÁN"

"TECNICAS DE ALTA COMPRESION
DE IMAGEN FIJA"

T E S I S A
QUE PARA OBTENER EL GRADO DE
LICENCIADO EN MATEMATICAS
APLICADAS Y COMPUTACION
P R E S E N T A :
ANGEL JAIR CONTRERAS



ASESOR: M.C. SARA CAMACHO CANCINO



MARZO - 2001



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

Agradezco a mi madre y mi padre, a todas las personas que de una u otra forma, durante distintos momentos, han recorrido este camino junto conmigo para alcanzar este logro que también es suyo.

Gracias a Dios y a todos ustedes.

ÍNDICE

Índice.	1
Introducción.	5
1. Principios y conceptos de la compresión de imagen fija.	9
1.1. Imagen e imagen digital	9
1.1.1 Clasificación de las imágenes digitales	9
1.1.2. Las formas de representación del color	10
1.1.3. Resolución de la imagen digital	12
1.1.4. Entidades de la imagen digital	12
1.2. Procesamiento de imagen digital	13
1.2.1. Campos de estudio del procesamiento de imagen digital	13
1.3. Compresión de datos y compresión de datos de imagen	14
1.3.1. Breves antecedentes de la compresión de datos	14
1.3.2. Definición de compresión de datos	14
1.3.3. Clasificación de las técnicas compresión de datos	16
1.3.4. Breves antecedentes de la compresión de datos de imagen	19
1.3.5. Definición de compresión de datos de imagen	20
1.3.6. Clasificación de las técnicas de compresión de imagen	20
1.4. Digitalización de imágenes	22
1.4.1. Muestreo en la digitalización de imagen	23
1.4.2. Cuantificación en la digitalización de imagen	27
1.5. Redundancia de las imágenes digitales	28
1.5.1. Redundancia de codificación	28
1.5.2. Redundancia interpixel	29
1.5.3. Redundancia psicovisual	30
1.6. Principios de la compresión de imagen digital	30
1.6.1. Unidades de información de la imagen	31
1.6.2. Entropía de la imagen	32
1.6.3. Principios de la compresión de imagen	33
1.7. Medidas de desempeño de la compresión de imagen digital	33
1.7.1. Eficiencia de la compresión	34
1.7.2. Complejidad de la compresión	34
1.7.3. Medición de la distorsión	34
1.7.3.1. Criterio de fidelidad objetiva	35
1.7.3.2. Criterio de fidelidad subjetiva	36
1.8 Resumen de principios y conceptos de la compresión de imagen fija.	37
2. Técnicas basadas en forma de onda.	39
2.1. Definición y clasificación de las técnicas de compresión basadas en forma de onda	39

2.2. Compresión predictiva	40
2.2.1. Codificación DPCM	40
2.2.2. Codificación adaptativa	44
2.3. Compresión basada en transformadas	45
2.3.1. Definición de las transformadas	46
2.3.2. Tipos de transformadas, transformadas KL y DCT	48
2.3.3. Algoritmo de codificación de transformada	51
2.4. Compresión sub-banda	53
2.4.1. Definición de sub-banda	53
2.4.2. Tipos de filtros digitales, filtros QMF	55
2.4.3. Algoritmo de codificación sub-banda.	57
2.5. Compresión wavelet.	69
2.5.1. Definición de wavelet	69
2.5.2. Tipos de wavelets.	71
2.5.3. Algoritmo de codificación wavelet	74
2.6. Compresión de cuantificación por vector	75
2.6.1. Definición de cuantificación por vector	77
2.6.2. Tipos de cuantificadores de vector	77
2.6.3. Algoritmo de cuantificación por vector	79
2.7. Resumen de la compresión de forma de onda	82
3. Técnicas de segunda generación.	85
3.1. Definición y clasificación de las técnicas de compresión de segunda generación.	85
3.2. Compresión por sistema alto sintético.	86
3.2.1. Descripción de la técnica.	87
3.3. Compresión piramidal.	88
3.3.1. Descripción de la técnica	88
3.3.2. Codificación de los datos	89
3.4. Compresión predictiva no estacionaria anisotrópica.	90
3.4.1. Descripción de la técnica.	90
3.4.2. Codificación de los datos.	94
3.5. Compresión basada en formación de región.	95
3.5.1. Descripción de la técnica.	95
3.5.2. Codificación de los datos.	101
3.6. Compresión basada en descomposición direccional.	103
3.6.1. Descripción de la técnica.	103
3.6.2. Codificación de los datos.	109
3.7. Compresión basada en reconocimiento de patrones.	113
3.7.1. Descripción de la técnica.	115
3.8. Compresión basada en segmentación de estructura de árbol.	117
3.8.1. Descripción de la técnica.	117
3.9. Resumen de la compresión de segunda generación.	118
4. Técnicas basadas en fractales.	121
4.1. Definición y clasificación de las técnicas de compresión fractal.	121
4.2. Definición de fractal.	122
4.2.1. Métrica.	124
4.2.2. Punto fijo.	124

4.2.3. Mapeo contractivo.	124
4.3. Compresión basada en el sistema de funciones iteradas (IFS).	125
4.3.1. Transformaciones afines.	126
4.3.2. Algoritmo de iteración aleatorio.	129
4.3.3. El teorema del collage.	131
4.3.4. Asignación de probabilidades.	132
4.3.5. Descripción de la técnica.	133
4.4. Compresión basada en transformada fractal.	135
4.4.1. Codificación de transformada fractal.	135
4.4.2. Decodificación de transformada fractal.	138
4.5. Resumen de la compresión fractal.	140
5. Estándares comerciales y áreas de aplicación de la compresión.	141
5.1. Estándares de compresión comerciales.	141
5.2. Estándares de compresión T.4 del CCITT.	142
5.2.1. Estándar T.4 unidimensional.	142
5.2.1.1. Descripción del algoritmo.	142
5.2.2. Estándar T.4 bidimensional.	145
5.2.2.1. Descripción del algoritmo.	145
5.3. Estándar de compresión T.6 del CCITT.	152
5.3.1. Descripción del algoritmo.	154
5.4. Estándar de compresión JBIG.	155
5.4.1. Definición de imagen JBIG.	157
5.4.2. Compresión JBIG secuencial.	161
5.4.3. Compresión JBIG progresiva.	162
5.5. Estándar de compresión JPEG.	163
5.5.1. Definición de imagen JPEG.	165
5.5.2. Compresión JPEG secuencial basada en DCT.	168
5.5.3. Compresión JPEG progresiva basada en DCT.	180
5.5.4. Compresión JPEG sin pérdida.	183
5.5.5. Compresión JPEG jerárquica.	184
5.6. Áreas de aplicación de la compresión de imágenes.	187
5.7. Aplicaciones computacionales.	188
5.7.1. Computadoras multimedia.	188
5.7.2. Sistemas de imágenes de documentos.	188
5.7.3. Servidores multimedia.	192
5.8. Aplicaciones para red.	195
5.8.1. Facsímile.	195
5.9. Aplicaciones para aparatos electrodomésticos.	198
5.9.1. Fotografía digital.	198
5.9.2. Cámaras digitales.	200
5.9.3. Photo CD.	202
5.10. Aplicaciones editoriales.	204
5.10.1. Publicaciones tradicionales.	204
5.10.2. Publicaciones electrónicas.	205
5.11. Aplicaciones para el cuidado de la salud.	207
5.11.1. Sistemas de imágenes médicas.	207
5.11.2. Operación de las IMACS.	209

5.12. Resumen de estándares comerciales y áreas de aplicación de la compresión.	212
Conclusiones.	215
Recomendaciones.	217
Bibliografía.	219
Anexo A. Técnicas de compresión sin pérdida de información.	221
A.1. Codificación Run-Length.	221
A.2. Codificación Shannon-Fano.	222
A.3. Codificación Huffman.	224
A.4. Codificación Ziv-Lempel.	226
A.5. Codificación Aritmética.	229

INTRODUCCIÓN

Las inquietudes más importantes para mí como estudiante de Matemáticas Aplicadas y Computación son precisamente los campos de aplicación donde se utilicen actualmente los conocimientos adquiridos durante los estudios de la carrera. Si bien la licenciatura cuenta con preespecialidades que buscan dar una orientación profesional a los alumnos, el campo de estudio es mucho más amplio, y las opciones profesionales mucho más variadas.

En el transcurso de los estudios, los alumnos de todas las carreras nos valemos de muchos medios para realizar nuestro trabajo y en los últimos años las telecomunicaciones a través de redes digitales han incrementado su importancia como herramienta alternativa. Si bien los temas de interés que pueden encontrarse a través de redes como Internet son de una gran variedad, la acción que funge como común denominador para todos los que las utilizan es el intercambio de información, así como su posterior almacenamiento. La información puede presentarse de distintas formas, sin embargo, para la mayoría de las personas, la presentación de la información en forma visual es mucho más fácil de comprender, ya que, además de estimular el pensamiento, estimula uno de los sentidos más importantes para el ser humano: la visión.

“Una imagen dice más que mil palabras” es una sentencia que está fuera de discusión ya que se respalda en la naturaleza misma del ser humano. En este sentido, también la orientación reciente de la tecnología de la computación sirve para apoyarla. El desarrollo de aplicaciones que utilizan, en cierta medida o de forma absoluta, a las imágenes manipuladas o generadas por medio de herramientas computacionales, es un indicador de la trascendencia de la información visual digital en muchas actividades como la educación, la industria, la ciencia, el comercio o el entretenimiento.

La compresión de imágenes es un ejemplo práctico de la manera en que las matemáticas se aplican hacia la generación de soluciones y alternativas para un problema relacionado con la computación. La necesidad de que las imágenes digitales requieran de una menor cantidad de información para representarlas tiene una trascendencia de actualidad, en virtud de que las comunicaciones por medios digitales se basan, contienen o proyectan contener material visual. Esto plantea una situación de exigencia tanto en dispositivos de

almacenamiento como en capacidades de transmisión, debido a la gran cantidad de datos que este tipo de información genera.

Como un antecedente del interés hacia un tema relacionado con el expuesto en este trabajo, el licenciado Alejandro Rubio presentó una tesina en la ENEP Acatlán sobre los formatos de archivos de audio, imagen y video digitales, en la que menciona las técnicas de compresión utilizadas por dichos formatos, así como algunas definiciones acerca de la estructura básica de las imágenes digitales. Dichas técnicas de compresión para imágenes corresponden a las primeras desarrolladas en este campo, siendo su mayoría importadas del área de la compresión de texto, las cuales no explotan por sí solas lo suficiente, las características particulares que la representación digital de una imagen involucra.

La generación de efectos especiales o la animación digital cinematográfica, los servicios de videoconferencia, las transmisiones de televisión, el “bajar” y retocar una imagen de Internet, o realizar diseños originales en una computadora personal, son algunos de los resultados más conocidos de la aplicación intensiva de métodos matemáticos propios del procesamiento digital de imágenes. Sin embargo, para la mayoría de las personas que los utilizamos, existe una “caja negra” que nos hace transparente su operación. Considero que al ser el manejo y generación de imágenes digitales un área de la computación que está extendiendo su alcance sobre un mayor número de personas y de formas mucho más variadas, es conveniente tratar una de las cuestiones más importantes del procesamiento al que se somete dicha información visual: la compresión. Este trabajo tiene como fin, develar uno de los vértices de esa “caja negra” que es el procesamiento digital de imágenes, específicamente en el área de la compresión de imágenes digitales, en proporción a los conocimientos adquiridos en la licenciatura de Matemáticas Aplicadas y Computación orientada hacia la preespecialidad de Sistemas Computacionales.

Existe una cantidad de técnicas que aprovechan la similitud de colores, el contraste entre los objetos o la semejanza entre dichos objetos con figuras geométricas básicas. Tales técnicas de procesamiento de información visual encuentran sus fundamentos tanto en métodos originalmente diseñados para el audio digital como en métodos generados especialmente para imágenes, auxiliándose del análisis y del reconocimiento de patrones.

El trabajo de investigación se limita a las técnicas denominadas de alta compresión, las cuales procesan la información de la imagen de forma que la compresión lograda supere el límite mínimo establecido por la Teoría de la Información para unidades digitales. Además, como se mencionó anteriormente, las técnicas expuestas se orientan específicamente hacia la compresión de imágenes digitales fijas, excluyendo al video digital.

La investigación realizada en este trabajo es de tipo documental. Los medios utilizados son los libros y artículos publicados disponibles que tengan como objetivo la introducción al tema del procesamiento digital de imágenes en lo general, y al de la compresión de imágenes digitales en lo particular. Los recursos disponibles fueron el material bibliográfico y hemerográfico de las bibliotecas de la Dirección General de Cómputo Académico (DGSCA), del Instituto de Investigación en Matemáticas Aplicadas y Sistemas (IIMAS), ambas de la Universidad Nacional Autónoma de México (UNAM); y de la biblioteca del Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional (IPN).

El trabajo está estructurado en cinco capítulos que describen las diferencias entre las técnicas para manejar los datos que componen las imágenes, aunque todas ellas tienen como objetivo la reducción en la cantidad de información necesaria para representarlas sin modificar su apariencia final notablemente. Sin embargo, en las implementaciones reales, muchas veces llegan a interactuar unas técnicas con otras en distintas etapas de la compresión, lo que dificulta su clasificación de forma contundente.

El primer capítulo ubica a la compresión de imágenes en el contexto del procesamiento de imágenes digitales y la muestra como una de sus áreas de estudio más importantes. También explica algunos de los conceptos fundamentales para la compresión de imágenes, así como para la extracción y manipulación de los datos que las componen y algunas medidas de evaluación de la eficiencia de las técnicas de compresión en términos de la reducción obtenida por su aplicación.

El segundo capítulo expone las técnicas de compresión correspondientes a la clasificación de forma de onda. Esta denominación busca generalizar los procedimientos que aprovechan las características de frecuencia de los valores que componen una imagen, sin dar importancia a su contenido visual, con el fin de reorganizarla o descomponerla de manera que la información menos frecuente se conserve y que la información más frecuente sea descartada sin deteriorar la apariencia de la imagen.

El tercer capítulo describe a las técnicas denominadas de segunda generación, las cuales corresponden a los procedimientos que, si bien algunas explotan igualmente las características de frecuencia de la imagen, lo realizan bajo un criterio de selección de información que considere a los componentes visuales contenidos en ella, como contornos, bordes y texturas.

El cuarto capítulo realiza una introducción hacia el tema de la teoría fractal desde el punto de vista de la compresión de imágenes y explica el funcionamiento de dos métodos de

compresión. Ambos procedimientos constituyen aproximaciones hacia la implementación práctica de conceptos que permitan representar los componentes visuales en una imagen en términos de ecuaciones matemáticas que generen figuras geométricas básicas.

El quinto capítulo consiste en las descripciones de los estándares de compresión más difundidos en el mercado de la comercialización y distribución de imágenes digitales. A pesar de que los estudios e investigaciones acerca de las técnicas de compresión de imagen datan desde finales de los años setenta y principios de los ochenta, la mayoría de ellos aún no se concretan en un estándar de compresión distribuido de manera comercial. Esto se debe a factores como la complejidad de sus operaciones y a la exigencia de recursos para los equipos de cómputo, pero principalmente a la preferencia de los fabricantes y consumidores por adoptar sólo algunos estándares de funcionamiento simple que faciliten el intercambio y manipulación de las imágenes digitales. En este capítulo también se comentan algunas de las aplicaciones más populares donde la compresión de imágenes es una herramienta básica para su funcionamiento o en las que se podría utilizar para mejorar su desempeño.

Los conocimientos de utilidad que faciliten la comprensión de este trabajo se relacionan con la arquitectura de computadoras, las unidades información digital, la estadística y el análisis de algoritmos.

CAPÍTULO I:

PRINCIPIOS Y CONCEPTOS DE LA COMPRESIÓN DE IMAGEN FIJA.

1.1. IMAGEN E IMAGEN DIGITAL.

Una imagen natural se define como una señal bidimensional conformada por el conjunto de ondas electromagnéticas que se encuentran en el rango de percepción del sistema visual humano. En otras palabras, es cualquier cosa que se capta a través de la vista por efecto de la acción de la luz. Esta imagen puede presentarse en la forma de una transparencia, diapositiva, fotografía, dibujo o incluso tomarse directamente de la realidad. Para obtener una versión digital de una imagen con el fin de manipularla por medio de una computadora digital se requiere del uso de un dispositivo digitalizador (generalmente scanners o cámaras digitales). El resultado de la digitalización se denomina imagen digital, la cual consiste en un arreglo de números reales o de conjuntos de números reales representado por una cantidad finita de bits. Cada elemento de dicho arreglo se denomina elemento de cuadro o pixel (picture element) y para el campo de estudio del procesamiento de imágenes digitales también se denomina muestra. La figura 1.1 ilustra este proceso:

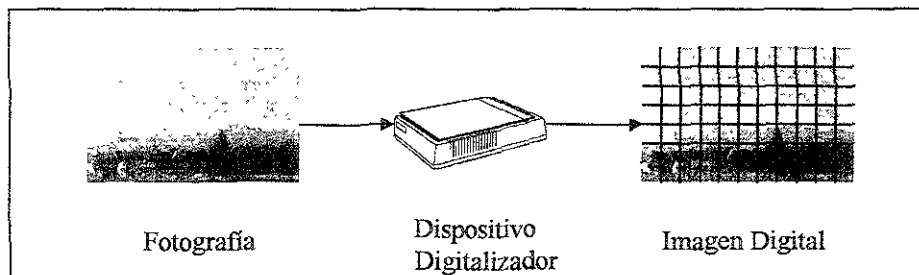


Figura 1.1. Obtención de una imagen digital a partir de una imagen del mundo real por medio de su digitalización.

1.1.1 Clasificación de las imágenes digitales.

La precisión en el valor de cada pixel o muestra determina cuantos niveles de intensidad de color pueden representarse. Es decir, a cada valor distinto de los bits que representan a la muestra se asocia un color distinto que adquirirá dicho elemento, dependiendo del tipo de imagen digital a la que corresponda.

En base a la exactitud de dicho valor es posible clasificar a las imágenes en tres categorías:

- Imágenes blanco y negro. En las que cada pixel se representa utilizando un solo bit que puede identificar el color negro o el color blanco. Estas imágenes también se conocen como binivel, binarias o bitonales.
- Imágenes de escala de gris. Donde cada pixel se representa por medio de un nivel de luminancia. Es decir, una gama de valores que pueden lograrse utilizando 8 bits por muestra para su representación, alcanzando una colección de 256 variaciones en el tono de gris desde el negro hasta el blanco. Para imágenes de documentos como las transmitidas por facsímile regularmente se usan de 4 a 7 bits.
- Imágenes a color. Tienen 3 componentes por pixel, mismos que ocupan 16, 24 o más bits para su representación cada uno. El tipo de información que proporciona cada componente para la integración de la imagen a color depende del formato de representación utilizado.

La figura 1.2 muestra una ilustración de las tres diferentes categorías de imágenes:

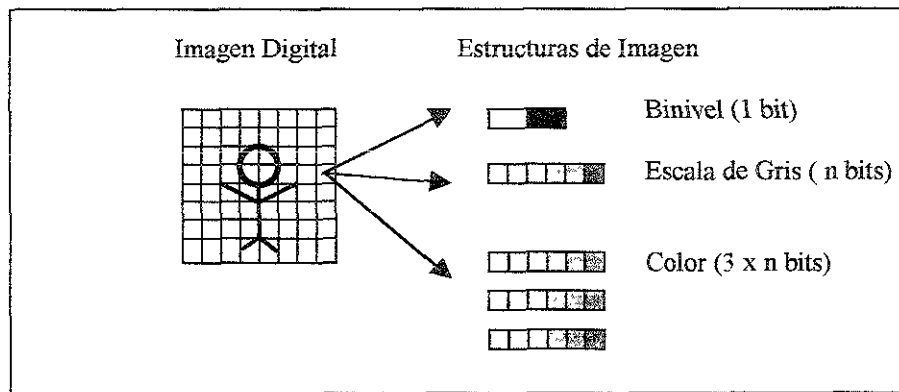


Figura 1.2. Estructuras de imagen digital blanco y negro, a escala de gris y a color (HOFFMAN, Roy. Data compression in digital systems).

1.1.2. Las formas de representación del color.

Los formatos de representación de color usados comúnmente son el RGB (Red:rojo, Green:verde y Blue:azul), el YUV o YIQ, y el YC_bC_r . Si bien los dos últimos formatos dependen directa o indirectamente del RGB debido a la forma en que se determinan sus valores, los formatos YUV y YC_bC_r se usan con más frecuencia al procesar de forma digital una imagen a color debido a que aligeran la carga de trabajo computacional.

El sistema de representación RGB se basa en la teoría tricromática que establece que la combinación apropiada de los tres colores básicos: rojo, verde y azul, puede formar cualquier color. Esto implica que una imagen con el formato de representación RGB está compuesta por píxeles de tres componentes cada uno. En las imágenes a color de 24 bits por muestra, las cuales pueden tener hasta 16,777,215 colores distintos basándose en la combinación planteada, cada componente utiliza 8 bits para almacenar un valor correspondiente al tono del color que representa. Puede decirse que existen tres versiones de la misma imagen, una representando el contenido de rojo, otra de verde y otra de azul, y que la fusión y la variación en los tonos de las mismas proporcionan la imagen a color final.

Otro sistema de representación es el formato YUV o YIQ el cual se forma a partir de un componente de luminancia (Y) y dos componentes de cromancia (U y V). Existen varias aproximaciones propuestas para el cálculo de los valores de Y, U y V, siempre a partir de una representación RGB anterior. La siguiente aproximación corresponde al estándar CCIR 601 (también conocido como formato YUV 4:2:2):

$$Y = R/4 + G/2 + B/2 \quad U = (B - Y)/2 \quad V = (R - Y)/2$$

Es decir, la representación YUV es una conversión del RGB hacia un conjunto de valores que concentran la información más importante de la imagen en los componentes de luminancia de cada muestra, dejando en los componentes de cromancia la información adicional. Puede decirse que el conjunto de todos los componentes decromancia formaría una versión de la imagen original a escala de gris con todos sus detalles y en los componentes de cromancia se encontraría la información necesaria para transformar la versión en escala de gris a una imagen a color.

Esta representación se utiliza generalmente para el procesamiento de imágenes digitales ya que sólo es necesario procesar uno de los tres componentes que integran las muestras para poder incidir en la imagen completa, a diferencia del formato RGB en el que es necesario procesar los tres componentes para modificar a la imagen en su conjunto.

Finalmente, en el formato YC_bC_r , el cual a su vez parte de una estimación previa en formato YUV de la imagen a representar, el componente Y es el mismo que en la representación YUV, pero los componentes C_b y C_r se obtienen a partir de los siguientes cálculos:

$$C_b = (U/2) + 0.5 \quad \text{y} \quad C_r = (V/1.6) + 0.5$$

Esta interpretación mejora el desempeño de la YUV, reduciendo la cantidad de información necesaria para representar la imagen al limitar el rango de valores de los

componentes C_b y C_r al intervalo de $[0,1]$. La Figura 1.3 muestra la relación de dependencia entre estas tres representaciones anteriores:

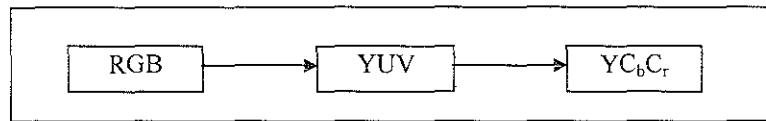


Figura 1.3. Dependencia de las representaciones de color, la YCbCr de la YUV y la YUV de la RGB.

1.1.3. Resolución de la imagen digital.

Otro factor importante que constituye a una imagen digital es la resolución de imagen. La resolución en un sistema de imágenes se refiere a su capacidad para reproducir los detalles finos, expresada en el número de píxeles que conforman un renglón (N) y el número de renglones (M) que integran el arreglo bidimensional en que consiste una imagen digital, y que se denota como $M \times N$. Entre mayor resolución se utilice para digitalizar una imagen, ésta tendrá mejor y más natural definición de los objetos que contenga al momento de desplegarse en un monitor. Las resoluciones a las que se digitalizan las imágenes comúnmente son: 320×240 , 640×480 , 800×600 , 1024×768 , 1280×1200 y 1600×1200 .

1.1.4. Entidades de la imagen digital.

Una imagen digital puede describirse de diversas formas dependiendo del tipo de proceso al cual será sometida. En este sentido, una imagen puede describirse en términos de las entidades mínimas que la componen, es decir en píxeles y las magnitudes de cada uno. Otra alternativa es describirla en términos de bloques de píxeles, de los cuales es posible estimar un valor promedio de la magnitud del color para el bloque completo. Una alternativa más es describirla en términos de los contornos y texturas que contiene.

Los contornos en imágenes digitales se definen comúnmente como cambios abruptos en el nivel de color o de escala de gris en un área de la imagen. De acuerdo a la contundencia del cambio, pueden definirse como contornos fuertes o débiles. La textura, por otro lado, en el contexto de las imágenes digitales puede interpretarse como el conjunto de píxeles que se encuentran rodeados por contornos y que si bien sí presentan ligeros cambios en su valor con respecto a los píxeles circundantes, lo que se denomina como granularidad, no llegan a considerarse propiamente como contornos.

1.2. PROCESAMIENTO DE IMAGEN DIGITAL.

El término de procesamiento de imagen digital generalmente se refiere al procesamiento de una imagen bidimensional por medio de una computadora digital. Una vez que se ha digitalizado una imagen y asignado una representación como las descritas en la sección 1.1.2, ésta se almacena como una matriz de dígitos binarios en la memoria de la computadora. Esta imagen entonces puede procesarse y/o desplegarse en un monitor.

La imagen natural que se somete a digitalización por medio de un dispositivo digitalizador y la imagen digital resultante que se obtiene en un monitor o impresora, conforman la entrada y la salida, a través de sus respectivas terminales, de un sistema de procesamiento digital de imagen típico como el que se muestra en la figura 1.4.

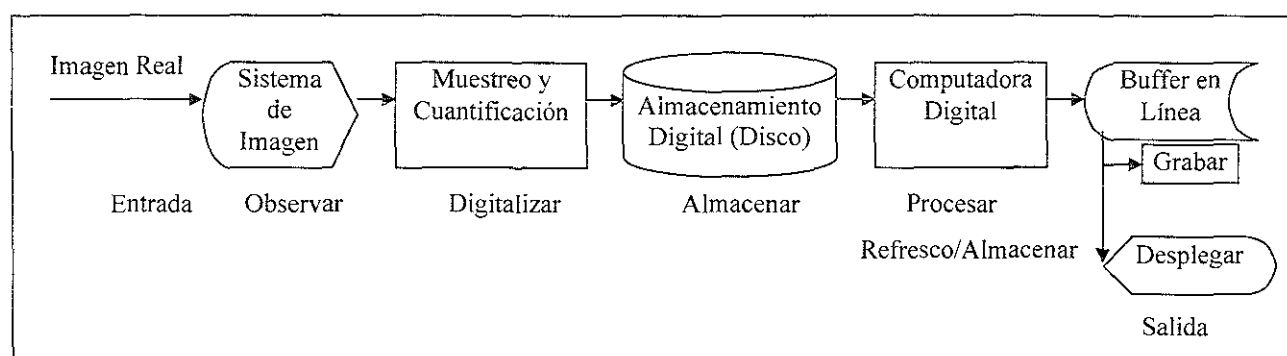


Figura 1.4. Una secuencia de procesamiento de imagen digital típica (JAIN, Anil K. Fundamentals of image processing).

El procesamiento de imágenes digitales tiene un amplio espectro de aplicaciones tales como localización remota vía satélite u otros equipos aéreos, transmisión y almacenamiento de imágenes para aplicaciones de negocios, procesamiento médico, utilización de radares y sonares, procesamiento de imágenes creadas a partir de la acústica, robótica e inspección automatizada de partes industriales. En pocas palabras, donde quiera que un ser humano, una máquina o cualquier otra entidad reciba datos de dos o más dimensiones, puede decirse que se está procesando una imagen.

1.2.1. Campos de estudio del procesamiento de imagen digital.

Las clases de problemas a los que se dirige el procesamiento de imágenes principalmente son:

- Representación y modelado de imágenes.
- Intensificación de imágenes.
- Restauración de imágenes.
- Análisis de imágenes.
- Reconstrucción de imágenes.
- Compresión de datos de imágenes.

Este trabajo expone algunas técnicas de procesamiento de imagen que se dirigen a la clase de problemas relacionados con la compresión de datos de imágenes.

1.3. COMPRESIÓN DE DATOS Y COMPRESIÓN DE DATOS DE IMAGEN.

Antes de explicar en qué consiste la compresión de datos de imagen es conveniente establecer que la compresión de datos en su sentido general, no surge como aplicación del área del procesamiento de imágenes, sino que es un campo de estudio que ha tenido su propio desarrollo y que encuentra sus comienzos a principios de los años cuarenta. La palabra "dato" involucra el medio por el cual se representan ciertos tipos de información significativa para un usuario en las formas de imagen, texto, video, audio y voz. Por lo que la compresión de datos es una herramienta que tiene influencia sobre el procesamiento de cualquiera de estas modalidades de los datos.

1.3.1. Breves antecedentes de la compresión de datos.

La compresión de datos remonta sus orígenes a la época inicial de los sistemas de comunicación electrónicos analógicos, siendo una de sus primeras técnicas el código Morse, ya que éste se valía de un conjunto limitado de símbolos y de ciertas reglas de secuencia para representar mensajes textuales completos. Posteriormente, con la conversión de los sistemas de comunicación hacia la forma digital y con la creación de dispositivos de almacenamiento computacionales, la compresión de datos ha buscado orientar sus técnicas hacia el aprovechamiento de esta representación de la información.

1.3.2. Definición de compresión de datos.

La compresión de datos es el nombre genérico que identifica al conjunto de técnicas que tienen como objetivo la reducción en la cantidad de espacio que es necesario asignar a un conjunto de mensajes o de datos. Este espacio puede presentarse en la forma de volumen, en el caso de un medio de almacenamiento como una cinta magnética; un intervalo de tiempo, como el tiempo requerido para transmitir el conjunto de mensajes dado; o en una porción del espectro electromagnético, como el ancho de banda requerido para transmitir el conjunto de datos dado. Estas tres formas de espacio están interrelacionadas por la siguiente fórmula:

$$\text{Volumen} = f(\text{tiempo} \times \text{ancho de banda})$$

Por lo que una reducción en el volumen puede interpretarse como una reducción en el tiempo de transmisión o en el ancho de banda.

La importancia de estos parámetros ha cambiado a través del tiempo, sin embargo, la reducción del volumen de espacio para almacenamiento sigue siendo uno de los factores más trascendentales en lo que a los sistemas digitales se refiere.

Para ilustrar los grandes requerimientos de almacenamiento de información, en una situación que considere la manipulación de diversos tipos de datos digitales como texto, imágenes, audio y video, considere una aplicación como una enciclopedia electrónica, que podría requerir:

- 500,000 páginas de texto (2 Kilobytes por página) – total 1 Gigabyte.
- 3,000 imágenes a color (en promedio de $640 \times 480 \times 24$ bits = 1Megabyte/imagen) – total 3 Gigabytes.
- 500 mapas (en promedio $640 \times 480 \times 16$ bits = 0.6 Megabyte/mapa) – total 0.3 Gigabytes.
- 60 minutos de sonido stereo (176 Kilobytes/seg.) – total 0.6 Gigabytes.
- 30 animaciones de 2 minutos de duración ($640 \times 320 \times 16$ bits x 16 cuadros/segundo = 6.5 Megabytes/segundo) – total 23.6 Gigabytes.
- 50 películas digitalizadas de 1 minuto de duración ($640 \times 480 \times 24$ bits x 30 cuadros/segundo = 27.6 Megabytes/segundo) – total 82.8 Gigabytes.

La enciclopedia requeriría 111.3 Gigabytes para almacenarse. Si se considera que sobre estos datos se aplican diversos algoritmos de compresión para los distintos tipos de datos de la enciclopedia, podrían obtenerse las siguientes proporciones de compresión:

- Texto 2:1.
- Imágenes a color 15:1.
- Mapas 10:1.
- Sonido stereo 6:1.
- Animación 50:1.
- Video 50:1.

Donde la notación n:1 expresa las veces en que el archivo ve reducido su tamaño original.

Haciendo uso de la compresión, los requerimientos de espacio se reducirían de 111.3 Gigabytes a sólo 2.96 Gigabytes, lo que resulta en menor costo por concepto de dispositivos e infraestructura de almacenamiento.

Existen tres razones principales por las que se requiere de la compresión de datos. Dichas razones están relacionadas con:

- Grandes requerimientos de almacenamiento de datos.
- Dispositivos de almacenamiento relativamente lentos que no permiten la recuperación de datos sin comprimir en tiempo real o al menos muy breve.
- Los anchos de banda de las redes actuales, que limitan la transmisión y distribución de grandes cantidades de datos en un tiempo igualmente razonable.

Conforme los sistemas digitales han entrado en los rangos de los gigabytes de información, los requerimientos de almacenamiento, recuperación y distribución de datos han incrementado los costos de las bases de datos y de los enlaces de comunicación hacia terminales remotas. Por lo que, aún contando con mayores anchos de banda en la parte de transmisión como tal en un sistema digital, la compresión de datos es una operación fundamental en otras partes del mismo.

La compresión de datos ha adquirido otros nombres a través del tiempo. Uno de los más comunes es el de Codificación de Fuente. Este nombre es un término de la denominada teoría de la información y se utiliza para distinguir este tipo de codificación de la Codificación de Canal. En este trabajo se hará uso indistinto de los términos compresión y codificación ya que para el ámbito al que se limita el tema resultan equivalentes.

1.3.3. Clasificación de las técnicas de compresión de datos.

En su caso más general, la compresión de los datos de una señal puede esquematizarse de la manera en que se presenta en la figura 1.5.

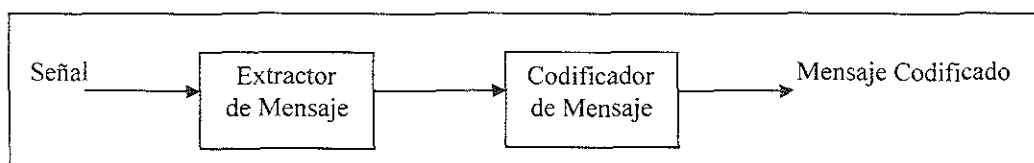


Figura 1.5. Modelo general del proceso de compresión de señales.

De acuerdo a este diagrama, dicho sistema está compuesto por un Extractor de Mensaje seguido por un Codificador de Mensaje. La extracción de mensaje está relacionada ya sea con el Modelo de Representación de la Información, el cual es una herramienta que se utiliza para calcular la probabilidad de que cierto dato se encuentre en el mensaje a codificar; y/o con las transformaciones a la representación inicial de la señal, las cuales se enfocan en la extracción de las características más relevantes de los datos originales, dependiendo de la transformación aplicada. Por otra parte, el Codificador de Mensaje es en

general, un procedimiento que asigna palabras de bits a cada mensaje extraído, ya sea por un método sistemático o haciendo uso del comportamiento estadístico de los mensajes extraídos.

Debido a que la compresión de datos es todavía un área relativamente nueva dentro de la teoría de la comunicación, aún no existe un criterio de clasificación oficial acerca de las técnicas que se emplean. Sin embargo existe cierto acuerdo entre los investigadores para definir dos grandes categorías basándose en el modelo de la figura 1.5: las técnicas de compresión de datos sin pérdida de información y las técnicas de compresión de datos con pérdida de información. Estas categorías han adquirido distintos sinónimos como: técnicas reversibles o irreversibles; técnicas de codificación sin ruido o de codificación con reducción de la fidelidad; y técnicas de reducción de la redundancia o de reducción de la entropía. Si bien determinar solamente las clasificaciones con pérdida y sin pérdida de información no es la postura más exacta debido a que algunas técnicas podrían encajar en ambas ramas, sí es conveniente para el propósito de diferenciar la manera en que las técnicas de compresión manipulan los datos para la disminución de sus requerimientos de espacio.

Las técnicas de compresión sin pérdida de información tienen por objetivo la eliminación parcial o total de los datos que resulten repetitivos dentro de un mensaje transmitido o almacenado (de ahí el nombre de reducción de la redundancia), para reinsertarlos nuevamente ya sea en el lado receptor del sistema de transmisión o en algún dispositivo de recuperación. Es decir que estas técnicas efectúan siempre un proceso reversible. Estas técnicas se orientan principalmente a los datos de los cuales no es posible prescindir como texto o instrucciones de programas. Este tipo de dato se denomina dato simbólico.

Las técnicas de compresión con pérdida de información eliminan de forma irreversible cierta cantidad de datos que sobrepasan un valor límite establecido o que no cumplen determinadas condiciones. Los datos que han sido descartados durante la compresión no pueden ser reconstruidos en la etapa de la recuperación del mensaje transmitido o almacenado. Estas técnicas se aplican generalmente sobre la información digital que, una vez recuperada, será percibida por un usuario humano, presumiblemente sin que la pérdida sea notoria como en el caso de las imágenes, la voz, el audio o el video. Este tipo de dato se denomina dato difuso. Como estas técnicas efectúan una mejor compresión de la información también se les denomina como técnicas de alta compresión. La figura 1.6 muestra la clasificación de las técnicas de compresión de datos en su sentido general.

Las técnicas sin pérdida de información ilustradas en la figura 1.6 varían entre sí respecto al procedimiento que utilizan para efectuar la compresión de datos, sin embargo todas giran en torno al mismo principio de conservación íntegra del mensaje procesado.

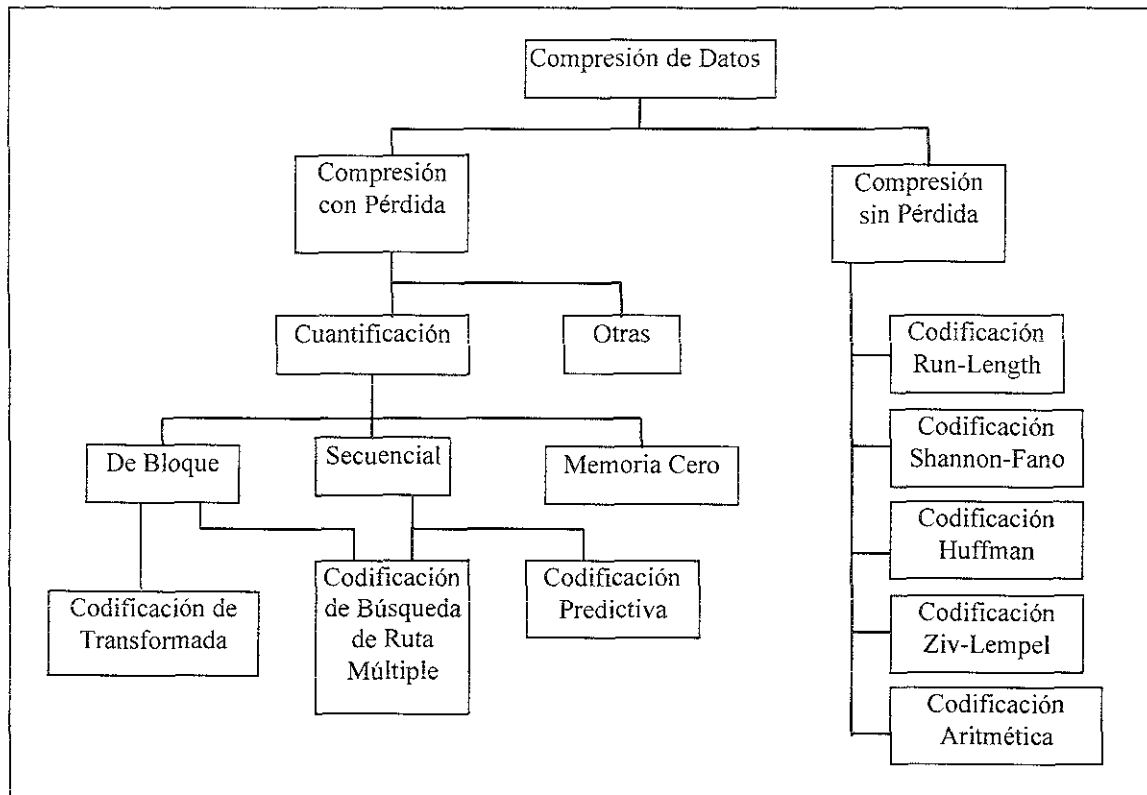


Figura 1.6. Clasificación de las técnicas de compresión de datos (KESPRET, Irene. PKZIP LHARC & Co. The ultimate data compression book).

Las técnicas con pérdida de información se clasifican en dos categorías: Cuantificación y Otras técnicas. El término cuantificación se refiere a la reducción en el valor de un dato con el fin de que ocupe menos bits para representarse. Si bien muchas de las técnicas que por el momento se generalizan con el nombre de Otras también efectúan cuantificación de datos, suelen ser técnicas especialmente diseñadas para preservar sólo alguna parte del contenido de la información de la fuente, dependiendo de la aplicación en particular.

También en la figura 1.6, debajo de cuantificación, hay tres categorías: de bloque, secuencial y memoria cero. La cuantificación de memoria cero fue la primera en usarse. En ella, cada muestra se cuantifica individualmente usando el mismo cuantificador sin importar los valores de las muestras cercanas. Los otros dos tipos de cuantificación, de bloque y secuencial, toman ventaja de la dependencia (similitud de valores) entre muestras, lo cual mejora la compresión obtenida. La forma más popular de la cuantificación de bloque es la codificación de transformada, debido a que proporciona un método de baja complejidad para procesar los datos y para recuperar los datos originales después de su procesamiento. Bajo la cuantificación secuencial, se encuentran dos tipos: la codificación predictiva y la codificación de búsqueda de ruta múltiple. En la codificación predictiva se hace una predicción de la muestra siguiente y se cuantifica la diferencia entre la predicción

y el valor original. La codificación de búsqueda de ruta múltiple pretende lograr la menor distorsión posible entre un conjunto de datos de entrada y un conjunto de valores previamente almacenados. Al conservar dicha diferencia se logra mayor eliminación de información, y al mismo tiempo mejor reducción de datos.

Hasta el momento se ha descrito a la compresión de datos en términos generales. Las dos grandes clasificaciones hacen distinción entre las técnicas en base al tratamiento que realizan sobre la información, por lo que también existe una distinción entre las aplicaciones a las que se orientan y para las que son funcionales. Aunque en términos de bits no hay diferencia entre la información que contiene texto y la que representa imágenes, a la hora de seleccionar una técnica de compresión es indispensable considerar la fuente que se pretende codificar. Si se pretende comprimir información de texto, no sería aceptable la eliminación de datos, como letras o frases completas ya que perdería totalmente el sentido y el valor. Por ello la aplicación de técnicas sin pérdida de información es lo más adecuado, aún a costa del precio pagado por espacio de almacenamiento.

El caso de la compresión de datos de imagen es particular en virtud de que las aplicaciones que la utilizan están diseñadas para que dichas imágenes sean observadas por una persona. Esta situación hace, como se explica en secciones posteriores, que las técnicas sin pérdida de información no sean la herramienta básica sino más bien auxiliar para la codificación de imágenes, por lo que éstas se remiten al anexo A de este trabajo. Las técnicas de compresión con pérdida enfocadas hacia la codificación de imágenes se exponen durante el desarrollo del trabajo. A continuación se hace una introducción general de la compresión de datos de imagen.

1.3.4. Breves antecedentes de la compresión de datos de imagen.

El interés en la compresión de datos de imagen o compresión de imagen data desde hace más de 25 años. El centro de atención inicial de las investigaciones en este campo fue sobre el desarrollo de métodos analógicos para reducir el ancho de banda necesario para la transmisión de video analógico o compresión de ancho de banda. Sin embargo, con la llegada de la computadora digital y el subsecuente desarrollo de circuitos integrados avanzados este interés se movió del ambiente analógico hacia aproximaciones de compresión digital. Con el reciente surgimiento de varios estándares de compresión de imagen internacionales, este campo actualmente tiene un crecimiento significativo en virtud de la aplicación práctica del trabajo teórico que comenzó con C. E. Shannon y otros investigadores que formularon la visión probabilística de la información a principios de los años cuarenta.

A través de los años la compresión de imagen ha crecido de forma sostenida, al punto de considerarse un factor crucial en el desarrollo de otros campos computacionales como la multimedia, y en el desempeño de aplicaciones como la videoconferencia, las imágenes médicas y de documentos, la transmisión de imágenes por facsímil y en general cualquier aplicación que dependa del manejo, almacenamiento o transmisión eficiente de imágenes.

1.3.5. Definición de compresión de datos de imagen.

La compresión de datos de imagen o compresión de imagen consiste en la reducción de la cantidad de datos requeridos para representar una imagen digital. Esta compresión, como ya se mencionó, es realizada por aplicaciones orientadas hacia los usuarios/observadores humanos, que son los únicos para los que una imagen desplegada en un monitor tiene sentido. La característica de “complementación de imagen” y la limitada capacidad de percepción a los colores detallados y una alta sensibilidad a los contrastes, bordes y texturas del sistema visual humano, hace factible la eliminación de alguna cantidad de información en una imagen que de cualquier forma, hasta cierto límite, no se podría apreciar si se conservara íntegra. Este hecho también influye directamente en sus costos de almacenamiento y/o transmisión. A este conjunto de datos extra innecesarios se le denomina redundancia psicovisual. Para procesar este tipo de información, las técnicas de codificación con pérdida son las que pueden aprovechar mejor estas circunstancias tan particulares para lograr una alta compresión de datos.

Para tener una apreciación de la importancia de la compresión de imágenes considere los requerimientos de almacenamiento de una imagen a color de 3 por 5 pulgadas explorada a 256 colores y a 300 dpi (puntos por pulgada). La imagen consiste en 15 pulgadas cuadradas con 90 mil puntos por cada una. Así, se requieren 1,350,000 píxeles para representar la imagen completa sin considerar los datos necesarios para representar el color. Como son necesarios 8 bits para lograr 256 colores distintos por píxel, esto resulta en un almacenamiento de datos mínimo requerido de 1.35 MB (megabytes) para una imagen de estas dimensiones. Con este ejemplo es posible considerar a la compresión como una herramienta indispensable cuando se trabaja con imágenes.

1.3.6. Clasificación de las técnicas de compresión de imagen.

La compresión imagen es el punto de convergencia entre el procesamiento de imágenes y la compresión de datos. Esto implica que aunque la compresión de imágenes utiliza las técnicas de compresión sin pérdida tradicionales, también toma algunas herramientas de otras áreas del procesamiento de imagen para desarrollar nuevas técnicas orientadas específicamente al tratamiento de información visual digital, y que dan pie a una

clasificación de técnicas de compresión de datos más particular. Dichas técnicas de compresión se basan en varios algoritmos que pueden implementarse ya sea por software o por hardware.

Al igual que las técnicas de compresión de datos en general, una de las clasificaciones más utilizadas para las técnicas de compresión de imagen se compone de dos categorías: Sin Pérdida de Información y Con Pérdida de Información. Las técnicas sin pérdida de información son capaces de recuperar la representación original perfectamente, sin descartar ni un solo dato de la misma. Las técnicas con pérdida de información involucran algoritmos que recuperan una representación similar a la original, descartando alguna información que no es imprescindible para la representación fidedigna de la imagen. Las técnicas con pérdida proporcionan mayor compresión y por lo tanto se aplican más a menudo en la compresión de imagen debido a la gran cantidad de información redundante que generan. La figura 1.7 muestra algunas de las técnicas de compresión sin pérdida más importantes:

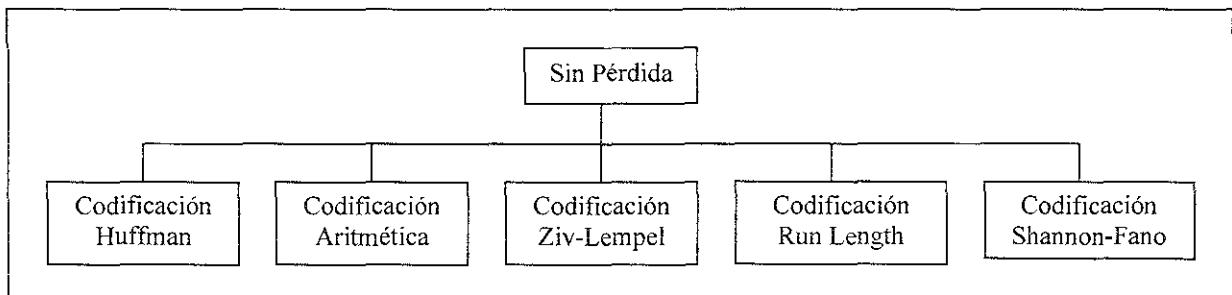


Figura 1.7. Técnicas de compresión de imagen sin pérdida de información.

La codificación de alta compresión (con pérdida de información) para imágenes fijas puede dividirse en tres categorías: Técnicas de Forma de Onda, Técnicas de Segunda Generación y Técnicas de Codificación Fractal. La primera categoría abarca la ramificación correspondiente a la de Cuantificación de la figura 1.6 y corresponde a las técnicas de compresión de datos difusos en general pero aplicadas hacia la información de imagen digital. Las dos últimas categorías, aunque también utilizan la cuantificación, toman el lugar de la de Otras que se mostró en la figura 1.6 ya que se han generado a partir de las condiciones particulares de las imágenes digitales y de algunas herramientas utilizadas en otras áreas del procesamiento de imágenes digitales. La clasificación anterior obedece a la diferencia entre los métodos que cada categoría utiliza para procesar los datos que componen la imagen, además de los métodos y conceptos matemáticos que utilizan para manejar y reducir la información a comprimir.

De forma general, puede decirse que las técnicas de forma de onda toman la información que representa a la imagen digital en bloques de dimensiones fijas y modifican los valores de cada bloque hacia alguna representación conveniente. Las técnicas de segunda

generación se ocupan de encontrar una forma adecuada de agrupar la información original en base a su estructura visual como los contornos o las texturas que contiene. La técnica de codificación fractal también utiliza la segmentación de la imagen, pero a diferencia de las dos anteriores, utiliza técnicas de codificación de la información que le permiten representarla en forma de sistemas de ecuaciones que generen figuras o fracciones elementales de la imagen (de ahí el nombre de fractales) para una compresión más eficiente. Los métodos de codificación basados en fractales pueden verse como una subclase, ya sea de las técnicas de forma de onda o de las de segunda generación, dependiendo de sus implementaciones. Sin embargo, se considerarán como una clase separada, ya que se basan en el principio de representar una imagen por el conjunto de transformaciones que la crearán a partir de cualquier punto inicial, en lugar de representarla a partir de una forma codificada de los resultados de las transformaciones de la imagen inicial. Ésta no es la única clasificación que puede hacerse sobre las técnicas de alta compresión o compresión con pérdida de información, sin embargo, sirve para ilustrar de forma más clara las distintas etapas del proceso general de compresión. Esta investigación consiste en una exposición amplia de cada una de estas clases de técnicas y sus subcategorías correspondientes. La figura 1.8 muestra un cuadro con esta clasificación.

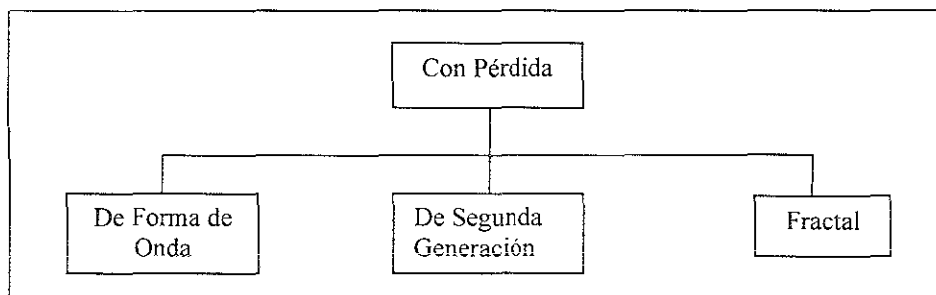


Figura 1.8. Técnicas de compresión de imagen con pérdida de información.

1.4. DIGITALIZACIÓN DE IMÁGENES.

Antes de comenzar la descripción de los principios de la compresión de imagen digital, es necesario exponer la forma en que las imágenes reales son convertidas a versiones digitales y las características particulares que presentan como resultado de este proceso, con el fin de comprender la constitución básica de la información que será manipulada para reducir sus requerimientos de almacenamiento. A continuación se describen los procesos de muestreo de la imagen real para su transformación hacia un arreglo de datos, y de cuantificación para reducir la cantidad de bits utilizados en su representación digital. Si bien se utilizan procedimientos de muestreo y cuantificación similares durante la compresión de la imagen digital, aquí se ocupan como la primera etapa indispensable para cualquier procesamiento de imagen.

1.4.1. Muestreo en la digitalización de imagen.

Una imagen es una representación de un objeto. Técnicamente hablando, las imágenes son señales bidimensionales percibidas por el sistema visual humano. Cuando una imagen es digitalizada, se vuelve una imagen digital. La digitalización de una imagen involucra dos procesos, Muestreo y Cuantificación. El proceso de muestreo ubica a una imagen física en un arreglo de píxeles por medio de puntos muestreados espacialmente a partir de la imagen física.

Se puede usar una función bidimensional $f(x, y)$ para representar una imagen. La imagen $f(x, y)$ también puede representarse por medio de su transformada de Fourier $F(\alpha, \beta)$ en el dominio de frecuencia, la cual está definida como:

$$F(\alpha, \beta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp \{ -j (\alpha x, \beta y) \} dx dy$$

La transformada de Fourier se utiliza para representar a la señal bidimensional de la imagen debido a que cuenta con características tales como:

- Los valores α y β son frecuencias espaciales que representan los cambios lumínicos con respecto a las distancias espaciales. Las unidades α y β son recíprocas de x e y , respectivamente.
- Para funciones continuas, $f(x, y)$ y $F(\alpha, \beta)$ son únicas una respecto de la otra. No existe pérdida de información si en lugar de preservar la imagen, se preserva su transformada de Fourier. Esta característica también se utiliza en la llamada codificación de transformada.

Antes del muestreo, una imagen es una señal continua bidimensional. El proceso de muestreo produce un número de muestras discretas a partir de dicha señal. De estas muestras, se puede reconstruir una imagen que puede ser o no idéntica a la imagen original. Si la imagen reconstruida es idéntica a la imagen original, se dice que la imagen es recuperada a partir de estas muestras. El proceso de reconstrucción es inverso al proceso de muestreo, y se ocupa de convertir las muestras discretas generadas por el muestreo de vuelta a una señal continua bidimensional. El proceso de muestreo se utiliza para procesar una imagen de forma digital. El proceso de reconstrucción se utiliza para mostrar una imagen ya digitalizada en algún dispositivo de despliegue (aparato de televisión o monitor).

Una imagen $f(x, y)$ se dice que es de banda limitada si su transformada de Fourier $F(\alpha, \beta)$ es cero fuera de una región limitada en el plano de frecuencia, como se muestra en la figura 1.9.

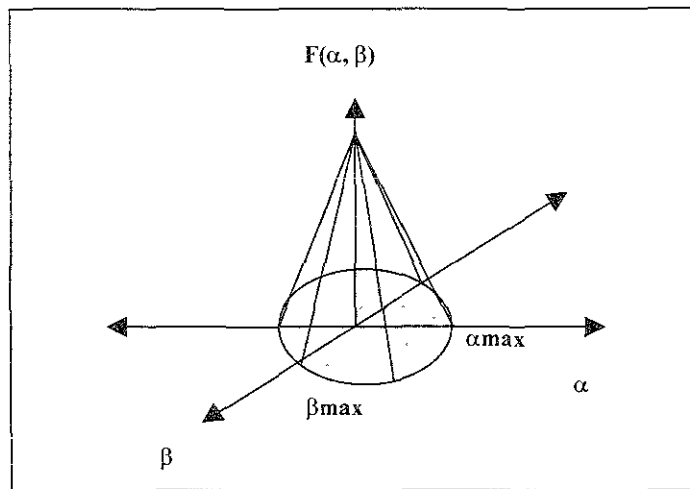


Figura 1.9. Región limitada de la transformada de Fourier $F(\alpha, \beta)$ en el plano de la frecuencia (KOU, Weidong. Digital image compression. Algorithms and standards).

Por la figura 1.9 puede expresarse la condición anterior como:

$$F(\alpha, \beta) = 0, \quad |\alpha| > \alpha_{\max}, \quad |\beta| > \beta_{\max}$$

En la práctica, es posible tratar a todas las imágenes como imágenes de banda limitada por promedios de aproximación a aquellas imágenes que no son de banda limitada. Las cantidades α_{\max} y β_{\max} se denominan los anchos de banda de x e y de la imagen. Si el espectro es circular simétrico como el de la figura entonces sólo una de las dos cantidades son el ancho de banda ya que $\alpha_{\max} = \beta_{\max}$.

La teoría de muestreo bidimensional establece que una imagen de banda limitada muestreada, horizontalmente a una frecuencia de muestreo mayor al doble de la mayor frecuencia en la dirección horizontal (α_{\max}), y verticalmente a una frecuencia mayor al doble de la mayor frecuencia en la dirección vertical en la imagen (β_{\max}), ésta puede recuperarse sin error.

Las frecuencias de muestreo que satisfacen esta condición se denominan Frecuencias Nyquist. Estas frecuencias pueden interpretarse como los límites mínimos para que el proceso de muestreo genere muestras discretas que reproduzcan con la mayor fidelidad posible la señal bidimensional de la imagen muestreada. La importancia de las frecuencias Nyquist radica en que si se quiere recuperar una imagen a partir de los puntos de muestreo

discretos sin errores, las frecuencias de muestreo deben ser mayores a las frecuencias Nyquist.

De la figura 1.9 es claro que las frecuencias Nyquist son $2\alpha_{\max}$ y $2\beta_{\max}$. Correspondiendo a las frecuencias de muestreo en el dominio de frecuencia, existe una red de muestreo en el dominio espacial. La red de muestreo es una red que determina los puntos de muestreo en el dominio espacial. Existe una relación entre las frecuencias de muestreo en el dominio de frecuencia y la red de muestreo en el dominio espacial correspondiente. Si tomamos a las frecuencias de muestreo como α_s y β_s , la red de muestreo en el dominio espacial sería como la de la figura 1.10, donde $\Delta x = 1/\alpha_s$ y $\Delta y = 1/\beta_s$.

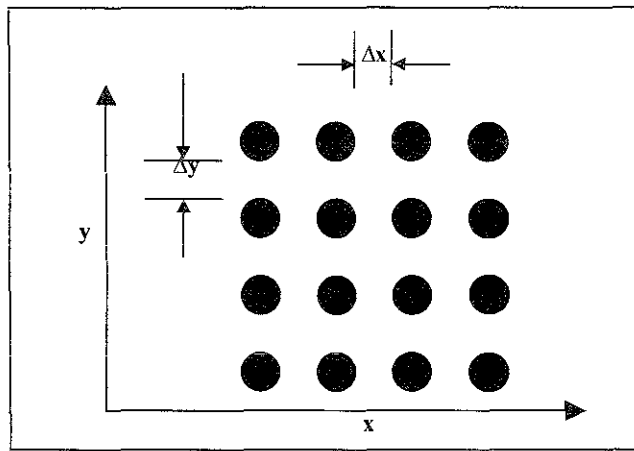


Figura 1.10. Red de muestreo determinada por las regiones limitadas de las transformadas de Fourier y por la distancia establecida entre ellas por Δx y Δy en el plano del espacio (KOU, Weidong. Digital image compression. Algorithms and standards).

Del análisis de Fourier se sabe que aplicando el proceso de muestreo, el espectro de la imagen se repetirá a sí mismo en el plano de la frecuencia de acuerdo a las frecuencias de muestreo. El proceso de reconstrucción usa un filtro de paso bajo para obtener uno de estos espectros repetidos y eliminar el resto de los espectros. Cuando las frecuencias de muestreo α_s y β_s , son mayores que las frecuencias Nyquist, $2\alpha_{\max}$ y $2\beta_{\max}$, el espectro de imagen muestreado luce como en la figura 1.11. En este caso, la transformada de Fourier de la imagen puede recuperarse por un filtro de paso bajo, con respuesta de frecuencia:

$$H(\alpha, \beta) = \begin{cases} 1, & \text{si } (\alpha, \beta) \in \mathfrak{R} \\ \frac{1}{(\alpha_s, \beta_s)}, & \\ 0, & \text{de otra forma} \end{cases}$$

donde $H(\alpha, \beta)$ identifica la salida del filtro de paso bajo y \mathfrak{R} es cualquier región cuyo límite $\partial\mathfrak{R}$ está contenido dentro del anillo entre los rectángulos \mathfrak{R}_1 y \mathfrak{R}_2 .

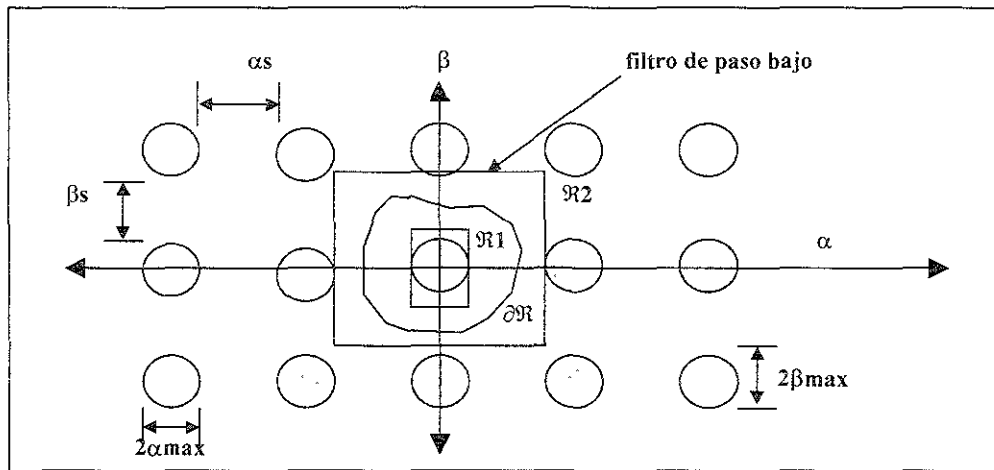


Figura 1.11. Espectro de la imagen en el plano de la frecuencia a la que se ha aplicado el muestreo con las transformadas de Fourier respetando la distancia establecida por las frecuencias de muestreo αs y βs (KOU, Weidong, Digital image compression. Algorithms and standards).

Por el contrario, si las frecuencias de muestreo son menores que las frecuencias Nyquist, entonces la transformada de Fourier no puede recuperarse. En este caso la imagen es submuestreada, como se muestra en la figura 1.12.

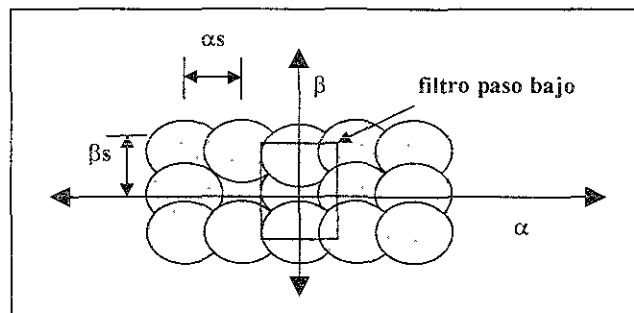


Figura 1.12. Espectro de una imagen submuestreada en el plano de la frecuencia. Las frecuencias de muestreo son menores a los límites de las regiones de las transformadas (KOU, Weidong, Digital image compression. Algorithms and standards).

El submuestreo produce una superposición del espectro. Así, serán introducidas frecuencias espaciales superiores, y serán creadas frecuencias bajas artificiales en la imagen reconstruida, por lo que la imagen reconstruida estará distorsionada. Este fenómeno se conoce como **aliasing**.

1.4.2. Cuantificación en la digitalización de imagen.

Después del muestreo, el siguiente paso en la digitalización de imagen es la cuantificación. La cuantificación es un proceso que ubica a una variable con una precisión ilimitada, en una variable con una precisión limitada. El objetivo es producir una imagen digital en la que cada pixel se represente por un número limitado de bits.

Una función escalera como la de la figura 1.13 se usa para el proceso de cuantificación. Para una muestra de entrada $t : t_k \leq t \leq t_{k+1}$, $k = 0, 1, 2, \dots, N-1$, la salida del cuantificador es q_k , $k = 0, 1, \dots, N-1$.

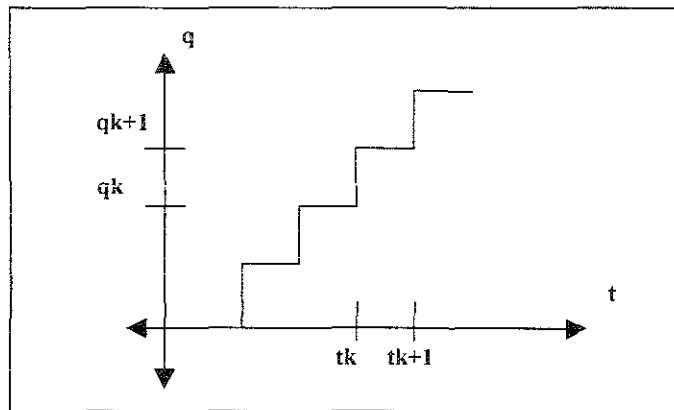


Figura 1.13. Ilustración el comportamiento general del proceso de cuantificación (KOU, Weidong. Digital image compression. Algorithms and standards).

Un esquema de cuantificación consiste en una clase de conjuntos de partición $\{Q_1, Q_2, \dots, Q_N\}$, y un conjunto de salidas de cuantificación $\{q_1, q_2, \dots, q_N\}$. Lloyd y Max probaron de forma independiente que hay dos condiciones necesarias para un cuantificador óptimo:

- *Condición del Vecino más Cercano:*

Para un conjunto dado de niveles de salida $\{q_1, q_2, \dots, q_N\}$, los conjuntos de partición $\{Q_1, Q_2, \dots, Q_N\}$ satisfacen:

$$Q_i \subset \{x : d(x, y_i) \leq d(x, y_j); \text{ para toda } i \neq j\}$$

donde $d(x, y)$ es una medida de la distorsión entre x e y .

- *Condición de Centroide:*

Para una partición dada $\{Q_1, Q_2, \dots, Q_N\}$, el conjunto óptimo de niveles de salida $\{q_1, q_2, \dots, q_N\}$ satisface:

$$q_i = E [x | x \in Q_i]$$

donde E denota la esperanza.

Si una imagen es de banda limitada, entonces se puede recuperar la imagen a través de un filtro de paso bajo. El muestreo es un proceso sin pérdida si las frecuencias de muestreo

son mayores que las frecuencias Nyquist. Sin embargo, la cuantificación es un proceso con pérdida debido a que nunca se podrá recuperar la imagen original. Desde el punto de vista de la digitalización, no existe la compresión sin pérdida. Los algoritmos sin pérdida son relativos a imágenes ya digitalizadas. Hay que asumir que las imágenes de que se hablará de aquí en adelante son imágenes digitalizadas.

En la práctica, el muestreo y la cuantificación pueden ser componentes de sistemas de captura de imágenes. Por ejemplo, un scanner produce una imagen digitalizada muestreando una imagen a través de una cámara y cuantificando las muestras obtenidas.

1.5. REDUNDANCIA DE LAS IMÁGENES DIGITALES.

Una vez que se obtiene la imagen digital, es necesario analizar una de sus características, derivada de la observación de los valores que la constituyen, y que se relaciona directamente con la posibilidad de reducir los datos necesarios para su representación.

Los valores que una imagen digital contiene, identifican niveles de luminancia derivados de las etapas de muestreo y cuantificación, los cuales varían en magnitud dependiendo de la representación utilizada (de 0 a 1, a 256 o a 3×256). Sin embargo, por la similitud en la intensidad de brillo de los objetos en una imagen real, y por el intervalo del espectro electromagnético de la luz, existen zonas de la imagen digital cuyos valores son muy parecidos. Es decir, hay datos que es posible omitir si se toman en cuenta otros datos del mismo valor revisados previamente. Esta característica se conoce como redundancia de datos.

La redundancia de datos es una cuestión central en la compresión de imágenes digitales, para la que existen tres tipos básicos de redundancia de datos: redundancia de codificación, redundancia interpixel y redundancia psicovisual. La compresión de datos se logra al reducir o eliminar uno o más de estos tipos de redundancias.

1.5.1. Redundancia de codificación.

Para ilustrar esta clase de redundancia supóngase que se tiene una imagen de escala de gris de 8 niveles con una distribución como la de la tabla 1.1, donde r_k corresponde a una variable aleatoria discreta en el intervalo $[0,1]$ representando alguno de los 8 niveles de gris y $p_r(r_k)$ corresponde a la probabilidad de ocurrencia del símbolo r_k .

r_k	$p_r(r_k)$	Código 1	$l_1(r_k)$	Código 2	$l_2(r_k)$
0	0.19	000	3	11	2
1/7	0.25	001	3	01	2
2/7	0.21	010	3	10	2
3/7	0.16	011	3	001	3
4/7	0.08	100	3	0001	4
5/7	0.06	101	3	00001	5
6/7	0.03	110	3	000001	6
1	0.02	111	3	000000	6

Tabla 1.1. Probabilidades de los símbolos r_k , junto con códigos de longitud fija y variable para su codificación (VETTERLI, Martin. Wavelets and subband coding).

Si se utiliza un código binario de 3 bits para representar los niveles de gris, como el Código 1 de la tabla, la longitud de palabra de código promedio L_{avg} será de 3. Sin embargo, si se utiliza el Código 2, el cual asigna palabras de bits más cortas para los niveles más probables y más largas para los menos probables, la longitud de palabra promedio se reduce a:

$$\begin{aligned}
 L_{avg} &= \sum_{k=0}^7 l_2(r_k) p_r(r_k) \\
 &= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + 4(0.08) + 5(0.06) + 6(0.03) \\
 &\quad + 6(0.02) \\
 &= 2.7 \text{ bits.}
 \end{aligned}$$

Como puede apreciarse, en base al aprovechamiento de las probabilidades de ocurrencia de los valores de nivel, es posible reducir la cantidad promedio de bits necesaria para representar a la imagen, lo que implica reducción en la redundancia de codificación. Esta técnica de codificación se denomina codificación de longitud variable (VLC).

1.5.2. Redundancia interpixel.

La redundancia interpixel está directamente relacionada con la correlación entre los píxeles dentro de una imagen. Debido a que el valor de un píxel dado puede predecirse de forma considerable a partir de los valores de sus píxeles vecinos, la información proporcionada por cada píxel de forma individual es relativamente pequeña. Mucha de la contribución visual de un solo píxel a una imagen es redundante, ya que puede calcularse a partir de los valores de sus píxeles vecinos. A este tipo de redundancia se le conoce como redundancia interpixel.

Con el fin de reducir las redundancias interpixel dentro de una imagen, el arreglo bidimensional de píxeles usado para representar a las imágenes digitales, debe

transformarse en un formato no visual más eficiente. Por ejemplo, pueden utilizarse las diferencias entre los píxeles adyacentes para representar una imagen. Las transformaciones de este tipo se denominan mapeos.

1.5.3. Redundancia psicovisual.

El brillo de una región percibido por el ojo humano depende de otros factores además de la luz reflejada por una región. Por ejemplo, Pueden percibirse variaciones en la intensidad de la luz aún en regiones que tienen una luminosidad constante, como en el caso de un monitor. Dicho fenómeno se deriva del hecho de que el ojo no tiene la misma sensibilidad para toda la información visual. Para el sistema visual humano, cierta información tiene menos importancia que otra. Esta información se dice que es psicovisualmente redundante. Esta información puede eliminarse sin deteriorar la apariencia de la imagen observada.

La percepción humana de la información en una imagen normalmente no involucra un análisis cuantitativo de cada píxel o del valor de luminancia en la imagen. En general, un observador busca distinguir ciertas características tales como bordes o regiones de textura y las combina mentalmente en agrupamientos reconocibles. Posteriormente el cerebro relaciona estos agrupamientos con un conocimiento previo para complementar el proceso de interpretación de la imagen.

A diferencia de la redundancia de codificación e interpixel, la redundancia psicovisual está asociada con información visual real y cuantificable. La eliminación de ésta es posible debido a que la información en sí misma no es esencial para el procesamiento visual humano. Como la eliminación de datos psicovisualmente redundantes resulta en una pérdida de información cuantitativa, a esta acción se denomina cuantificación. Este término generalmente implica el mapeo de un rango amplio de valores de entrada hacia un número limitado de valores de salida. Como la cuantificación es una operación irreversible, resulta en una pérdida de información.

1.6. PRINCIPIOS DE LA COMPRESIÓN DE IMAGEN DIGITAL.

Ya se ha establecido cuáles son los tipos de información redundante presentes en una imagen digital, mismos que hay que eliminar o al menos disminuir para lograr su compresión. El sentido de la codificación de datos es lograr la cantidad mínima de datos necesarios para representar a la imagen sin pérdida de la información esencial. Para determinar cuantos datos son indispensables conservar para lograr este objetivo, la

compresión de imágenes se vale de los resultados de la teoría matemática conocida como teoría de la información.

Ideada por Claude Shannon en 1948, la teoría de la información propone un modelo matemático para un sistema de comunicación como el que se ilustra en la figura 1.14. Existen dos aspectos a considerar a partir de este modelo:

- La manera en que un sistema de comunicación puede transmitir eficientemente la información que una fuente produce.
- La manera en que un sistema de comunicación puede lograr una comunicación confiable a través de un canal con ruido.

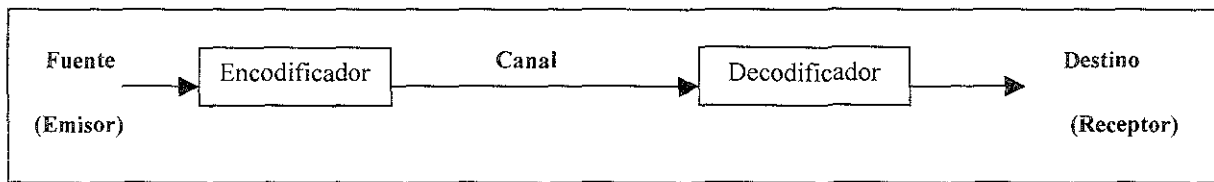


Figura 1.14. Modelo de sistema de comunicación de Shannon (KOU, Weidong. Digital image compression. Algorithms and standards).

El primer aspecto se refiere a las etapas del modelo donde se efectúa la compresión y la descompresión de la información, denotadas por los cuadros de Encodificador y Decodificador. El segundo se dirige hacia la codificación adecuada de datos para tener control sobre los niveles de error que pudieran introducirse en la información a través del medio físico de transmisión, denotado por el Canal. Estos dos aspectos se conocen como codificación fuente y codificación de canal.

1.6.1. Unidades de información de la imagen.

La premisa fundamental de la teoría de la información es que la generación de la información fuente puede modelarse como un proceso probabilístico susceptible de medición. De acuerdo con esta suposición, un evento aleatorio E que ocurre con una probabilidad $P(E)$ se dice que contiene:

$$I(E) = \log(1/P(E)) = -\log P(E)$$

unidades de información. La cantidad $I(E)$ también se denomina auto-información de E . La cantidad de auto-información atribuida al evento E es inversamente proporcional a la probabilidad de E . Si $P(E) = 1$ (si el evento es seguro), $I(E) = 0$ por lo que no se atribuye información a éste. Es decir, ya que no hay ningún nivel de incertidumbre atribuible al evento, no es necesario transmitir ninguna unidad de información para comunicar que el

evento ocurrió. Sin embargo, si $P(E) = 0.99$, si se requiere una pequeña cantidad de información para comunicar que E ha ocurrido. Por lo tanto, debido a que E es casi seguro, se ocupa más información para comunicar que E no ha ocurrido.

La base del logaritmo determina la unidad utilizada para medir la información, lo que en este caso se trata de unidades binarias o bits. Por ejemplo, si $P(E) = \frac{1}{2}$, $I(E) = -\log_2 \frac{1}{2}$ o un bit. Esto significa que un bit es la cantidad de información requerida para comunicar la ocurrencia de uno de dos eventos igualmente posibles.

1.6.2. Entropía de la imagen.

En el ámbito de las imágenes digitales, una fuente es una imagen digitalizada. Los símbolos fuente para representar las imágenes digitales son píxeles en la forma de bits o secuencias de bits. El usuario obtiene la información que contiene una imagen digital recibiendo todos los píxeles que representan la imagen. Cada uno de estos símbolos fuente ocupa una cantidad distinta de unidades de información, y al promedio de unidades de información por símbolo se conoce como entropía. Para cada símbolo, existe un producto de la probabilidad del símbolo y su logaritmo. La entropía es una suma negativa de los productos de todos los símbolos en un conjunto de símbolos determinado.

Matemáticamente hablando, una fuente puede modelarse como un proceso aleatorio discreto. Para un conjunto de símbolos dado $S = \{ s_j \mid j = 0, 1, \dots, M-1 \}$, si una fuente produce una secuencia de variables aleatorias, r_0, r_1, r_2, \dots donde $r_k \in S$, siguiendo un modelo de probabilidad $P = \{ p(s_j) \mid j = 0, \dots, M-1 ; s_j \in S \}$, la información de la fuente se define como su entropía:

$$H(S) = - \sum_{j=0}^{M-1} p(s_j) \log p(s_j)$$

Por ejemplo, si una fuente tiene un conjunto de símbolos de $\{1,0\}$ como en el caso de las imágenes binivel, y si éste produce una secuencia binaria siguiendo el modelo de probabilidad:

- Se produce '1' con probabilidad de 0.8
- Se produce '0' con probabilidad de 0.2

Entonces la cantidad promedio de información producida por la fuente es:

$$\text{Entropía} = - (0.8 \log_2 0.8 + 0.2 \log_2 0.2) = 0.7219 \text{ bits/símbolo.}$$

La entropía puede interpretarse como la cantidad mínima de bits por símbolo necesaria para representar la información de la imagen digital sin pérdida de información, en base al modelo probabilístico empleado para dichos símbolos. Su resultado indica que existe un esquema de codificación de palabras de código binarias (como en la sección 1.5.1) capaz de reducir la asignación de bits de cada símbolo hasta la tasa propuesta por la entropía.

Los algoritmos de compresión son métodos que reducen el número de símbolos utilizados para representar una información fuente, por lo que en última instancia, buscan lograr una asignación de palabras de código cercana a su entropía. En consecuencia, también reducen la cantidad de espacio necesario para almacenar la información fuente o la cantidad de tiempo necesario para transmitirla por un canal de cierta capacidad. La transformación de los símbolos finales de vuelta a símbolos fuente representando la información original o una forma muy cercana a ésta se denomina descompresión. El método de transformación que efectúa esta operación se conoce como algoritmo de descompresión.

1.6.3. Principios de la compresión de imagen.

De las definiciones anteriores se derivan los principios de los algoritmos de compresión de imágenes, los cuales son:

1. Reducir la redundancia en los datos de la imagen y(o)
2. Producir una imagen reconstruida de la imagen original con la introducción de un cierto nivel de distorsión que es insignificante para la aplicación en cuestión.

El objetivo es obtener una representación más eficiente de los datos de imagen mientras que se preserva la información esencial contenida en ellos.

1.7. MEDIDAS DE DESEMPEÑO DE LA COMPRESIÓN DE IMAGEN DIGITAL.

Un algoritmo de compresión de imagen es eficiente, dependiendo de la relación que guarde con las medidas de desempeño diseñadas para evaluarlo. Existen tres criterios de evaluación básicos para un algoritmo de compresión de imagen digital:

- Eficiencia de la compresión.
 - Complejidad de la compresión.
 - Medición de la distorsión.
-

1.7.1. Eficiencia de la compresión.

La eficiencia de la compresión se estima en base a la proporción de compresión, la cual se define como la proporción, C_R , del número de bits de la imagen original, $S_{original}$, con respecto al tamaño de la imagen comprimida, $S_{comprimida}$:

$$C_R = \frac{S_{original}}{S_{comprimida}}$$

A partir de C_R , es posible determinar una medida que evalúe el grado de redundancia que guardan dos conjuntos de datos que representan a la misma imagen, es decir $S_{original}$ y $S_{comprimida}$. Esta medida se denomina redundancia de datos relativa R_D y se define como:

$$R_D = 1 - (1 / C_R)$$

Si $S_{original} = S_{comprimida}$, $C_R = 1$ y $R_D = 0$, lo que indica que la imagen original no contiene datos redundantes. Cuando $S_{comprimida} \ll S_{original}$, $C_R \rightarrow \infty$ y $R_D \rightarrow 1$, lo que indica que se logró una alta compresión de datos redundantes. Finalmente, si $S_{comprimida} \gg S_{original}$, $C_R \rightarrow 0$ y $R_D \rightarrow -\infty$, lo que indica que $S_{comprimida}$ contiene más datos que la imagen original, lo cual es el caso contrario al propósito de la compresión.

1.7.2. Complejidad de la compresión.

La complejidad de un algoritmo de compresión de imagen, se mide por el número de operaciones requeridas para efectuar los procesos de encodificación y decodificación. Las operaciones con datos incluyen sumas, restas, multiplicaciones, divisiones y desplazamientos. Usualmente los algoritmos más complejos producen mayor reducción de datos y tardan más en ejecutarse debido a la cantidad de cálculos que realizan.

1.7.3. Medición de la distorsión.

Esta medición tiene por objetivo evaluar el grado de fidelidad que guarda una imagen generada a partir de datos que fueron comprimidos con respecto a la imagen original antes de la compresión. Es decir, Existen dos clases de criterios para evaluar la cantidad de información perdida: el criterio de fidelidad objetiva y el criterio de fidelidad subjetiva.

1.7.3.1. Criterio de fidelidad objetiva.

Cuando el nivel de información perdida puede expresarse como una función de la imagen original y la imagen comprimida y subsecuentemente descomprimida, se dice que ese nivel está basado en un criterio de fidelidad objetiva. Un ejemplo de esto es el error de raíz media cuadrada (root-mean-square o rms) entre una imagen de entrada y una de salida. Para calcularlo, supóngase que $f(x, y)$ representa una imagen de entrada y $f'(x, y)$ representa una aproximación de $f(x, y)$ que resulta de la compresión y descompresión de la imagen de entrada. Para cualquier valor de x e y , el error $e(x, y)$ entre $f(x, y)$ y $f'(x, y)$ se define como:

$$e(x, y) = f'(x, y) - f(x, y)$$

por lo que el error absoluto entre las dos imágenes es:

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x, y) - f(x, y)]$$

donde las imágenes son de tamaño $M \times N$. A partir de esta medida es posible calcular la pérdida de información promedio entre los componentes de las imágenes. Esta medida se conoce como el error cuadrado medio, denotado como MSE. El MSE se define como:

$$\text{MSE} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x, y) - f(x, y)]^2$$

Entonces, el error de raíz de media cuadrada, e_{rms} entre $f(x, y)$ y $f'(x, y)$ es la raíz del error MSE sobre el arreglo $M \times N$:

$$e_{\text{rms}} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x, y) - f(x, y)]^2 \right]^{1/2}$$

Una medida de fidelidad muy relacionada es la proporción de media cuadrada señal a ruido de la imagen comprimida-descomprimida. Si se considera a la imagen $f'(x, y)$ como la suma de la imagen original más una señal de ruido $e(x, y)$, la proporción de media cuadrada señal a ruido, denotada por SNR_{ms} sería:

$$\text{SNR}_{\text{ms}} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f'(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f'(x, y) - f(x, y)]^2}$$

1.7.3.2. Criterio de fidelidad subjetiva.

El factor de la apariencia de imagen también es de primera importancia en la mayoría de las aplicaciones que requieren compresión de datos visuales. De hecho, si bien la mayoría de las aplicaciones requieren altas proporciones de compresión, este requerimiento está en general en conjunción con el deseo de una apariencia excelente de las imágenes resultantes. Por eso, el objetivo es que las imágenes resultantes del proceso de descompresión brinden la calidad más alta posible, en términos de no introducir ninguna distorsión visible. En consecuencia, resulta apropiado medir dicha calidad de una imagen por medio de evaluaciones subjetivas. Esto puede lograrse al mostrar una imagen descomprimida a una selección determinada de observadores y promediar la evaluaciones. Las evaluaciones pueden efectuarse utilizando una escala de clasificación o por comparaciones de las imágenes original y descomprimida una contra otra. La tabla 1.2 muestra una posible escala de clasificación, mientras que la tabla 1.3 muestra una posible escala de comparación una contra otra. En cualquier caso, se dice que las evaluaciones se basan en un criterio de fidelidad subjetivo.

Valor	Evaluación	Descripción
1	Excelente	Una imagen de muy alta calidad, tanto como se pueda desear.
2	Buena	Una imagen de alta calidad, agradable a la vista. La interferencia no es objetable.
3	Admisible	Una imagen de calidad aceptable, aunque podría ser mejor. La interferencia no es objetable.
4	Marginal	Una imagen de calidad pobre que podría mejorarse. La interferencia es algo objetable.
5	Inferior	Una imagen de calidad pobre, aunque se puede ver. La interferencia es muy objetable.
6	Inadmisible	Una imagen tan mala que no puede verse.

Tabla 1.2. Escala de clasificación de calidad de imagen de la Asociación de Estudios de Transmisiones de Televisión (E.U.).

Comparación	Descripción
-3	Mucho peor
-2	Peor
-1	Ligeramente peor
0	La misma
1	Ligeramente mejor
2	Mejor
3	Mucho mejor

Tabla 1.3. Escala de comparación de calidad una contra otra entre imágenes originales y descomprimidas.

1.8. RESUMEN DE PRINCIPIOS Y CONCEPTOS DE LA COMPRESIÓN DE IMAGEN FIJA.

Las imágenes se componen de señales electromagnéticas bidimensionales perceptibles al ojo humano a través de la acción de la luz. Las diversas formas en que los seres de la realidad reaccionan al ser iluminados y la manera en que la vista interpreta esas reacciones se conoce como color. Los dispositivos digitalizadores convierten dichas señales en información digital susceptible de manipularse por medio de una computadora digital. Existen tres tipos de imágenes digitales: binivel, a escala de gris y a color. En virtud de la cantidad de información digital que contengan pueden ser de baja o alta resolución.

El caso de las imágenes digitales se sujeta a criterios de valoración distintos a los de otros tipos de información digital, siendo uno de los más importantes el de la apreciación visual que un observador obtenga de la imagen, y restando trascendencia a la integridad o completitud de los datos que la conforman. Debido a que el proceso de digitalización no tiene la capacidad de discernir la información estrictamente necesaria de la información adicional, existen dos aproximaciones para resolver el exceso de información. La eliminación de los datos digitales que se encuentren repetidos, lo que se conoce como compresión sin pérdida de información; y la conservación de los contornos, figuras y contrastes en la imagen por sobre el fondo o las tonalidades de color suaves, que se conoce como compresión con pérdida de información. Esta segunda es más conveniente para las imágenes digitales ya que se rige igualmente por un criterio de percepción.

Los criterios para evaluar una técnica de compresión de información digital son la eficiencia con la que comprimen, es decir, cuanta información se descartó con respecto de la cantidad original; la complejidad con la que se lleva a cabo la compresión; y la distorsión de la información contenida en los datos comprimidos con respecto a la original, que para el caso de la compresión de imágenes, se involucra la apreciación que un observador humano tenga de la misma.

CAPÍTULO II: TÉCNICAS BASADAS EN FORMA DE ONDA.

2.1. DEFINICIÓN Y CLASIFICACIÓN DE LAS TÉCNICAS DE COMPRESIÓN BASADAS EN FORMA DE ONDA.

Estas técnicas se refieren a los métodos que utilizan los valores de cada pixel que conforman la imagen digital para efectuar operaciones que aprovechen su similitud o correlación. Para su ejecución se ocupan métodos matemáticos que proporcionan una representación más eficiente para identificar el contenido de una imagen, en términos de la cantidad de valores necesarios para tal efecto. Estas técnicas se denominan de forma de onda debido a que sus métodos aprovechan el comportamiento estadístico, que puede esquematizarse como una onda, de los valores de una imagen en un área determinada, entre pixeles vecinos, cercanos, dentro de un bloque o dentro de un vector. Estas técnicas pueden dividirse en dos categorías: codificación predictiva y codificación basada en transformación. De ellas se derivan las técnicas estudiadas en este capítulo, las cuales se esquematizan en la figura 2.1.

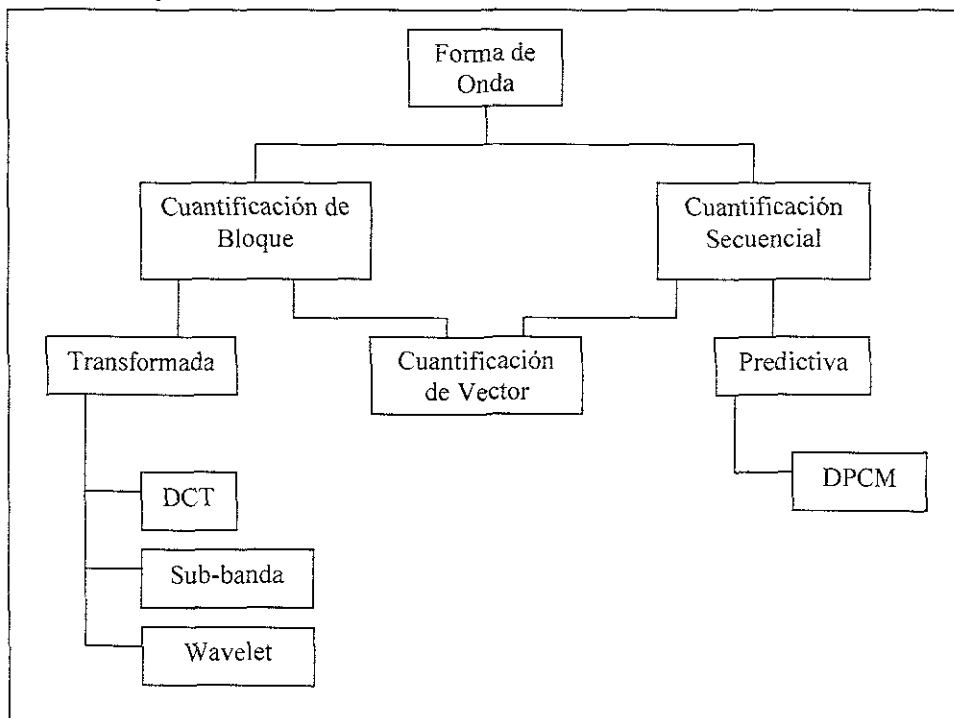


Figura 2.1. Clasificación general de la codificación de forma de onda.

2.2. COMPRESIÓN PREDICTIVA.

La filosofía básica de las técnicas de codificación predictiva es la de remover la redundancia mutua entre pixeles sucesivos, denominada redundancia interpixel (sección 1.5.2), y codificar solamente la información nueva resultante. La técnica de codificación más representativa de esta categoría es la Modulación para Codificación de Pulso Diferencial (DPCM), la cual se expone a continuación.

2.2.1. Codificación DPCM.

Considérese una secuencia de valores originales $i(l)$, la cual se ha codificado hasta $l=k-1$, y sean $i'(k-1)$, $i'(k-2)$,..., $i'(k-N)$ los valores de la secuencia reproducida resultante (decodificada). En $l = K$, cuando llega $i(k)$, se predice una cantidad $'(k)$, la cual es un estimado de $i(k)$, a partir de las muestras $i'(k-1)$, $i'(k-2)$,..., $i'(k-N)$ decodificadas anteriormente. Esto es,

$$'(k) = f(i'(k-1), \dots, i'(k-N))$$

donde $f(\dots)$ denota la regla de predicción de $i(k)$. Posteriormente, para codificar su valor sólo es necesario codificar el error de predicción,

$$e(k) = i(k) - '(k)$$

Si se cuantifica $e(k)$, obteniendo así el valor $e'(k)$, entonces se toma al valor reproducido de $i(k)$ original como:

$$i'(k) = '(k) + e'(k)$$

El proceso continua de esta manera recursivamente. Este método se denomina modulación para codificación de pulso diferencial (DPCM). La figura 2.2 muestra los diagramas del codificador y del decodificador DPCM.

El codificador tiene que calcular la secuencia reproducida $'(k)$. El decodificador simplemente actúa como ciclo predictor del codificador.

El codificador da salida al error de predicción entre $i(k)$ y $'(k)$. Ya que se existe una etapa de cuantificación, el error sufre una reducción en su valor dependiendo del cuantificador aplicado, dando salida al error cuantificado $e'(k)$ definido por:

$$e'(k) = e(k) - q(k)$$

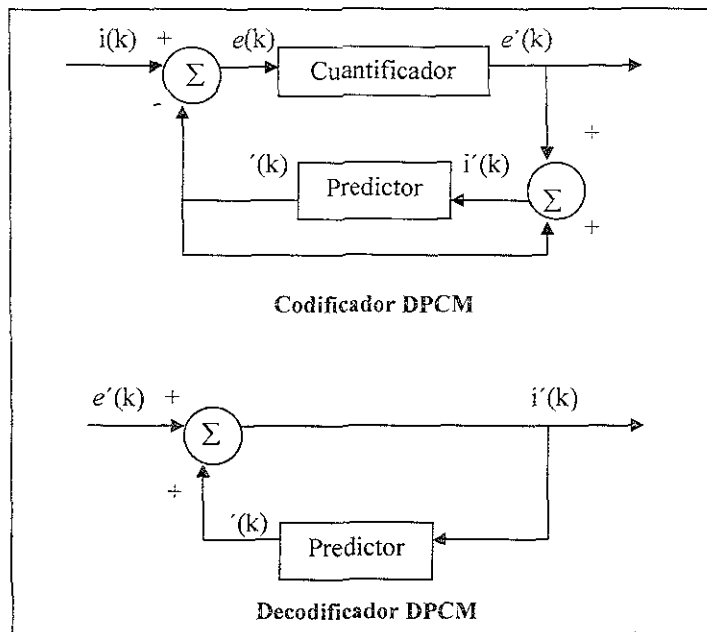


Figura 2.2. Esquema del codificador y del decodificador DPCM (KOU, Weidong, Digital image compression. Algorithms and standards).

donde $e'(k)$ es el error de predicción de $e(k)$ cuantificado con el error de cuantificación $q(k)$.

Supóngase que se tiene una secuencia de píxeles con valores de escala de gris 100, 102, 120, 120, 120, 118 y 116 la cual debe codificarse de forma predictiva utilizando DPCM en base a la siguiente regla de predicción: $i'(k) = i'(k-1)$. Asíumase que se utiliza un cuantificador de dos bits como el de la figura 2.3, excepto para el primer valor de la secuencia, el cual se cuantifica con 7 bits, dando como resultado $i'(0) = i'(0) = 100$.

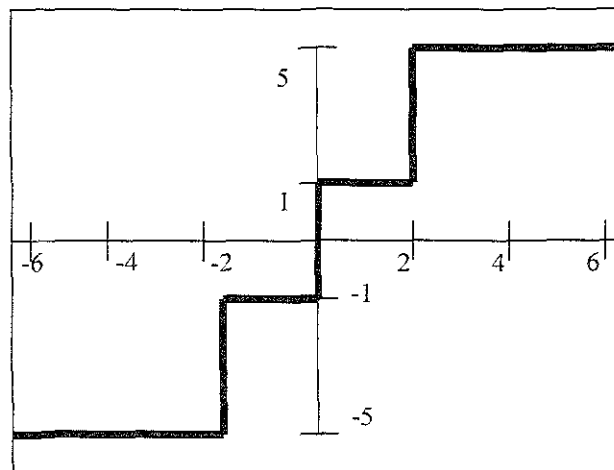


Figura 2.3. Ejemplo de cuantificador de error de predicción para codificación predictiva.

Las tablas 2.1 y 2.2 muestran el resultado del codificador y del decodificador DPCM sobre secuencia propuesta en las columnas $e'(k)$ e $i'(k)$ respectivamente. La figura 2.4 muestra el diagrama del compresor DPCM con la regla de predicción propuesta, la figura 2.5 muestra el diagrama del descompresor DPCM.

La codificación predictiva juega un importante papel en estándares de compresión tales como el JPEG y JBIG debido a su eficiencia de compresión y su simplicidad.

k	$i(k)$ Entrada	$\hat{i}(k)$	$e(k)$	$e'(k)$ Salida	$i'(k)$	$q(k)$
0	100	-	-	-	100	-
1	102	100	2	1	101	1
2	120	101	19	5	106	14
3	120	106	14	5	111	9
4	120	111	9	5	116	4
5	118	116	2	1	117	1
6	116	117	-1	-1	116	0

Tabla 2.1. Resultados de la codificación DPCM.

K	$e'(k)$ Entrada	$\hat{i}(k)$	$i'(k)$ Salida
0	-	-	100
1	1	100	101
2	5	101	106
3	5	106	111
4	5	111	116
5	1	116	117
6	-1	117	116

Tabla 2.2. Resultados de la decodificación DPCM.

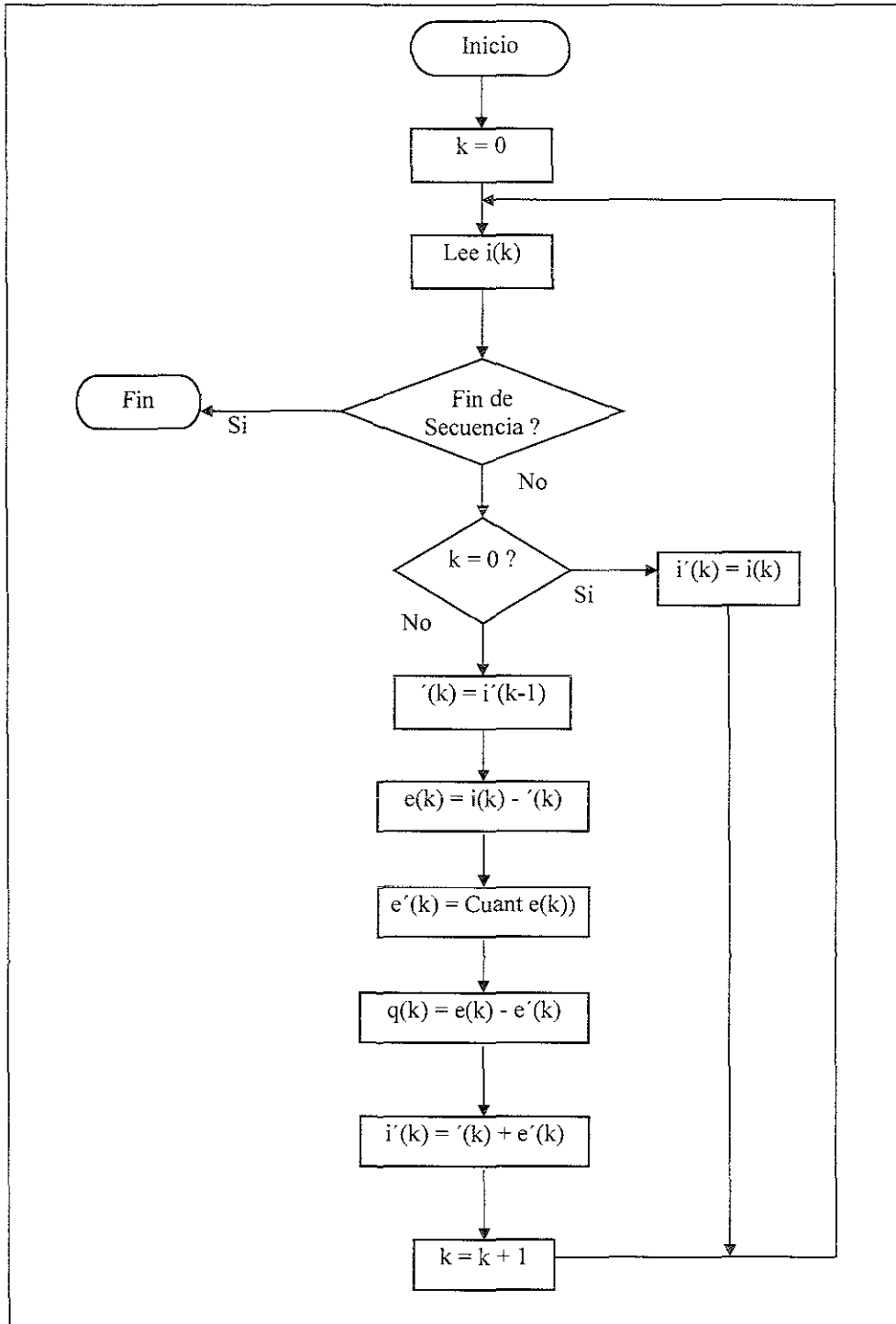


Figura 2.4. Diagrama de flujo del compresor DPCM.

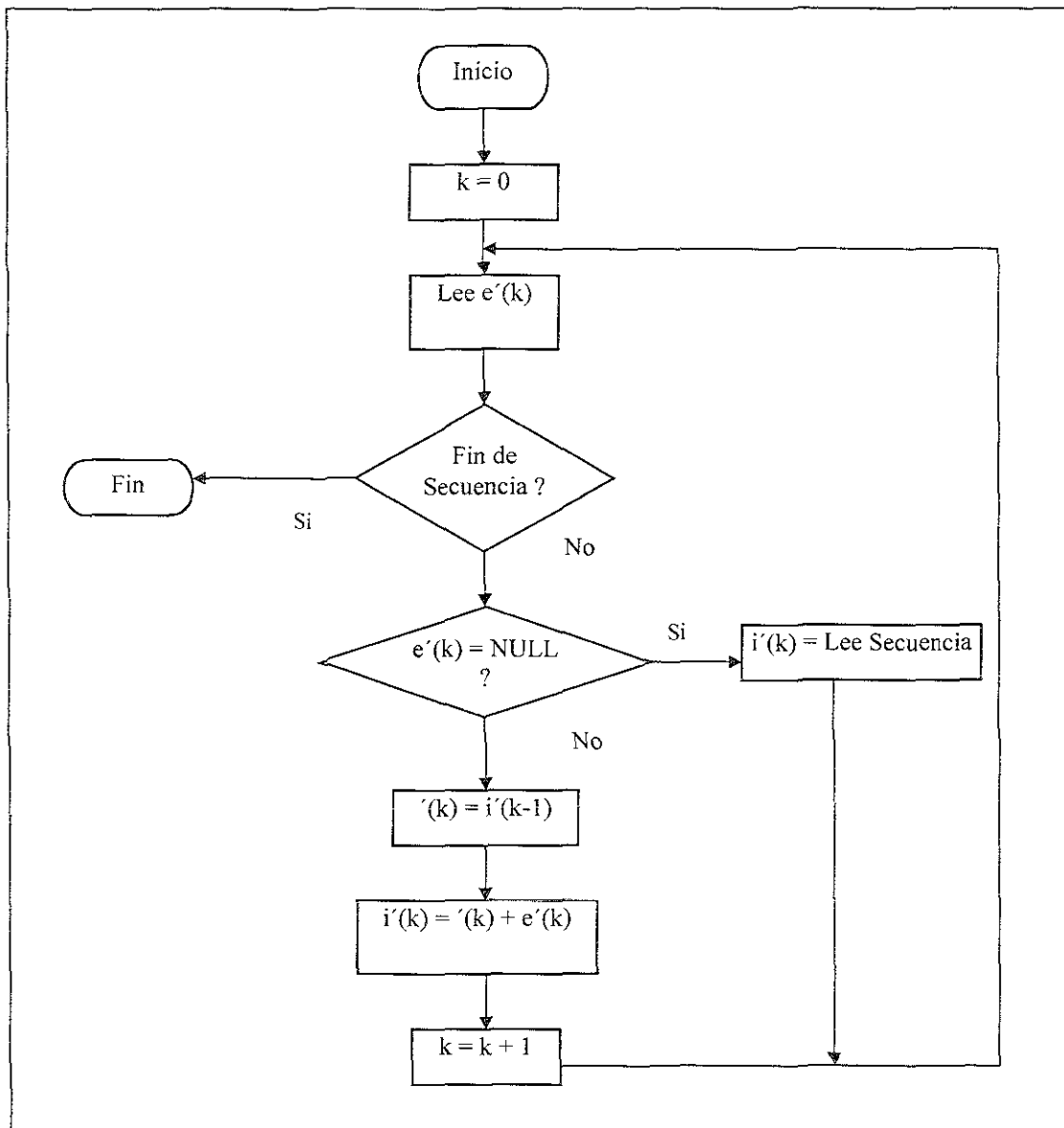


Figura 2.5. Diagrama de flujo del descompresor DPCM.

2.2.2. Codificación adaptativa.

La codificación adaptativa usualmente se ocupa como complemento a la codificación predictiva, y se basa en la predicción del valor de los píxeles venideros basándose en los píxeles vistos previamente en una forma no necesariamente lineal. Por ejemplo, si los últimos diez píxeles en una fotografía a escala de gris tienen todos valores entre 45 y 50,

un sistema de compresión adaptativa podría predecir con alta probabilidad que el valor del siguiente pixel estaría en el mismo rango. Entonces, podría utilizarse un sistema de codificación entrópica, como la codificación Huffman o la aritmética (Anexo A), para asignar probabilidades a los valores venideros.

La mayoría de los esquemas adaptativos se basan en utilizar algunos de los pixeles circundantes como parte de los cálculos para las probabilidades del pixel venidero. En la figura 2.6, el pixel a ser codificado se muestra en la posición 0,0. Los pixeles que se utilizan más comúnmente cuando se calculan las probabilidades se muestran en las posiciones A,B,C y D.

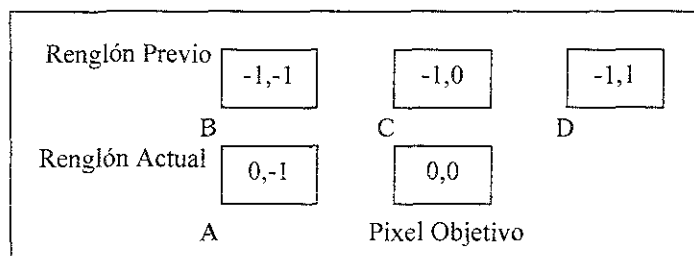


Figura 2.6. Ejemplo de esquema de predicción adaptativo (NELSON, Mark. The data compression book).

Las predicciones acerca del valor del pixel objetivo pueden hacerse basándose en ecuaciones predictivas como: $P = A, B, C, (A+C)/2, (A+D)/2, (A+(C-B)), (A+(C+D)/2)/2, (A+(C-B)/2)$.

Estas técnicas usan los datos previos para calcular el valor más parecido al del pixel objetivo, y ajustan el esquema de codificación coordinadamente. Mientras que estos cálculos producen buenos resultados, nuevamente no son de la magnitud necesaria para efectuar una compresión de alta efectividad aún con cuantificación.

De esta forma, la regla de predicción para el ejemplo de la sección anterior se sustituye por alguna de las reglas predictivas adaptativas propuestas, considerando que debe estar presente un buffer de memoria que vaya almacenando los valores anteriores al pixel objetivo para efectuar la predicción. Dicho buffer se limpiará conforme la regla de predicción vaya descartando los valores que ya no sean necesarios para el cálculo de la predicción.

2.3. COMPRESIÓN BASADA EN TRANSFORMADAS.

El término de transformada de imagen usualmente se refiere a una clase de arreglos o matrices utilizadas para representar imágenes. Dicha representación tiene por objeto el

facilitar el procesamiento y análisis de datos aprovechando el nivel de correlación que exista entre ellos. A la aplicación de una transformada sobre la secuenciabidimensional que implica una imagen, se denomina transformación. Las transformaciones tienen la cualidad de cambiar el dominio de los valores que conforman la imagen hacia otro que permita procesarlos de forma conveniente, para después realizar una transformación inversa sobre los datos procesados y regresarlos al dominio original. La forma en que actúa una transformada sobre una imagen se representa en la figura 2.7.

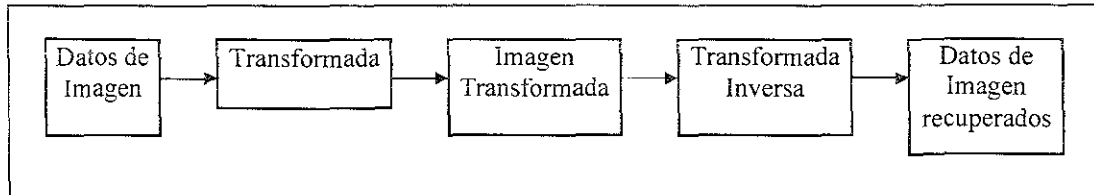


Fig. 2.7. Proceso de transformación y transformación inversa.

La transformación por sí sola no efectúa ninguna modificación sobre los datos originales, sólo modifica la estructura de su representación. El resultado de la transformación es un arreglo de coeficientes, denominados coeficientes de transformada, de las mismas dimensiones que el arreglo de la imagen original.

2.3.1. Definición de las transformadas.

Las transformadas se utilizan en el procesamiento de imágenes debido a ciertas propiedades que facilitan la eliminación de información por efecto de la cuantificación sin deteriorar la apariencia de la imagen decodificada. Las propiedades aprovechadas son las siguientes:

- Conservación y rotación de la energía: implica que los datos de la imagen original pueden recuperarse completamente a partir de los coeficientes de transformada resultantes del proceso de transformación. Los datos simplemente cambian su plano hacia uno que permita derivar los valores de los píxeles en relación con las posiciones que ocupan en la matriz.
- Compactación de la energía: la transformación analiza la correlación entre el valor de las muestras desde los píxeles de las primeras columnas y los primeros renglones hacia el resto de la matriz. Este proceso permitirá determinar los valores de toda la matriz a partir de los primeros píxeles. Esto se interpreta como una compactación de la energía en pocos coeficientes de transformada y muchos coeficientes que no proporcionan información.

- **Decorrelación:** Por las características de redundancia de las imágenes digitales, los valores que la componen están fuertemente correlacionados. Es decir, que hay una alta probabilidad de predecir el valor de un pixel a partir de cualquier otro dentro de la matriz que los contenga debido a la similitud de su valor. Una vez que se ha aplicado la transformada, esta circunstancia desaparece, ya que los pixeles superiores izquierdos de la matriz tendrán valores mayores a los del resto del arreglo. Esta característica se denomina decorrelación.

Para propósitos de descripción, una imagen digital de $N \times N$ puede verse como una secuencia bidimensional $u(m,n)$, sobre la que se aplica una transformada para obtener los coeficientes de transformada $v(k,l)$. Igualmente, puede aplicarse una transformada inversa sobre los coeficientes de transformada para obtener la secuencia de imagen recuperada. Ambas transformaciones pueden expresarse como:

$$v(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m,n) a_{k,l}(m,n), \quad 0 \leq k,l \leq N-1$$

$$u(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k,l) a_{k,l}^*(m,n), \quad 0 \leq m,n \leq N-1$$

donde $\{a_{k,l}(m,n)\}$ es la transformada de imagen, la cual puede verse como un conjunto de funciones que satisfacen las propiedades de:

- **Ortonormalidad:** Asegura que siempre se podrá recuperar cualquier parte de la imagen original a partir de los coeficientes de transformada correspondientes, aplicando la transformada inversa $\{a_{k,l}^*(m,n)\}$.
- **Completitud:** Asegura que los datos recuperados al aplicar la transformada inversa $\{a_{k,l}^*(m,n)\}$ serán valores aproximadamente idénticos a los de la imagen original.

La cualidad de invertibilidad de la transformada es lo que la hace útil para el procesamiento de imágenes ya que es posible devolver los datos transformados a su forma original. En este sentido, las transformadas son matrices unitarias. Es decir, que su matriz inversa tiene una matriz equivalente que puede calcularse más fácilmente. Si se toma a la transformada como A , su inversa corresponde a A^{-1} y ésta a su vez equivale a A^{*T} que corresponde a la conjugada transpuesta de la matriz A . Existen muchos tipos de transformadas, pero las que son útiles para el procesamiento de imágenes son las que proporcionan coeficientes de tipo real, ya que se pueden manejar de forma más simple. Esta circunstancia restringe el uso de las transformadas a las transformadas reales y ortogonales, esto es que $A = A^*$ y que $A^{-1} = A^T$. Estos resultados simplifican enormemente los cálculos necesarios para determinar los coeficientes de las transformadas.

Una característica más a las que se restringen las transformadas usadas en el procesamiento de imágenes es la de separabilidad. Esta cualidad implica que la transformada $a_{k,l}(m,n)$ pueda expresarse de la forma:

$$a_{k,l}(m,n) = a_k(m)a_l(n) = a(k,m)a(l,n)$$

y que por las propiedades de ortonormalidad y de completitud sea válido que, usando de nuevo la notación, $AA^{*T} = A^*A^T = I$. Utilizando esta característica, es posible que las ecuaciones de transformación originales adquieran una nueva representación:

$$v(k,l) = \sum_{m,n=0}^{N-1} a(k,m) u(m,n) a(l,n) \leftrightarrow V = AUA^{*T}$$

$$u(m,n) = \sum_{k,l=0}^{N-1} a^*(k,m) v(k,l) a^*(l,n) \leftrightarrow U = A^{*T}VA$$

donde U y V representan a las matrices de la imagen original y la de coeficientes de transformada, respectivamente.

Complementando esta representación con las equivalencias de las matrices conjugadas y conjugadas transpuestas, se obtiene la representación más simplificada:

$$V = AUA^T$$

$$U = A^TVA$$

Estas ecuaciones se denominan transformaciones bidimensionales separables. Esta nueva estructura derivada de las características anteriores implica que la transformación se llevará a cabo primero sobre cada columna de U y después sobre cada renglón de U para finalmente obtener la matriz V .

Esta representación final de las transformaciones reduce de manera importante la cantidad de sumas y multiplicaciones necesarias para efectuar la transformación y es la que se utiliza para las implementaciones de este procedimiento sobre datos de imagen digital.

2.3.2. Tipos de transformadas, transformadas KL y DCT.*

Las transformadas que se utilizan con más frecuencia en el campo del procesamiento de imágenes son:

* Para mayor información ver: JAIN, Anil K. Fundamentals of image processing. Prentice-Hall.

- Transformada Karhunen-Loeve.
- Transformada Coseno.
- Transformada Seno.
- Transformada DFT unitaria.
- Transformada Hadamard.
- Transformada Haar.
- Transformada Slant.

Todas las transformadas anteriores cumplen las características de conservación y compactación de la energía y de decorrelación mencionadas anteriormente. Sin embargo, varían en el desempeño de esas operaciones debido a la forma en que se generan. De entre las transformadas anteriores se conoce a la transformada Karhunen-Loeve (KL) como la transformada óptima y al resto como transformadas subóptimas.

Para explicar a que se debe esta diferencia es necesario establecer algunos conceptos. La correlación ($r(n) = \rho^{|n|}$ con $0 \leq \rho \leq 1$ para toda n) es una medida de la fuerza con la que los valores de una secuencia $u(n)$ se encuentran linealmente relacionados independientemente de la escala de medición. En el caso de las imágenes digitales fuente esta medida es alta para grandes conjuntos de valores en virtud de la redundancia interpixel descrita en 1.5.2, siendo ρ con valor cerca de 1. A partir de esta medida es posible determinar una matriz de correlación para la secuencia $u(n)$ con $1 \leq n \leq N$ de la forma:

$$R = \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & \rho^2 & \rho^3 & \dots & \rho^N \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \rho^{N-1} & \dots & \rho & \dots & 1 \end{pmatrix}$$

La transformada busca generar una representación de la secuencia $u(n)$ lo menos correlacionada posible, es decir, reducir ρ cerca de cero. Al generar la matriz R (con ρ cerca de cero) de la matriz de coeficientes de transformada, ésta debe asemejarse a una matriz diagonal y con ceros fuera de la misma, por lo que sus coeficientes han reducido su dependencia lineal por efecto de la aplicación de la transformada.

En este sentido, la transformada KL es óptima debido a que se define como:

$$Y = \Phi^T X$$

donde X es la matriz de datos original, Y es la matriz de datos transformados y Φ^T es la transpuesta de la matriz unitaria Φ , la cual reduce la matriz de correlación R de X a su forma diagonal. Esto puede interpretarse como que la transformada KL elimina la dependencia lineal de la matriz de datos original X en base a correlación de sus propios datos.

La definición de la transformada KL la hace óptima para las características de compactación de la energía y de decorrelación. Sin embargo, debido a que depende totalmente de los valores particulares y del tamaño de cada secuencia de valores y a que no es posible estimarla de forma analítica, su complejidad es elevada.

Para algunos tipos de procesamiento como la compresión es posible utilizar transformadas subóptimas que no dependan de los valores de las secuencias, debido a que son generadas por operaciones preestablecidas, a cambio de una menor compactación y decorrelación entre los datos.

La tabla 2.3 muestra las varianzas σ^2 de los coeficientes de transformada de una secuencia aleatoria (como una línea de imagen) con factor de correlación $\rho = .95$ y de tamaño $N = 16$.

Secuencia	Transformadas						
	KL	Coseno	Seno	DFT Unitaria	Hadamard	Haar	Slant
0	12.442	12.406	11.169	12.406	12.406	12.406	12.406
1	1.946	1.943	1.688	1.100	1.644	1.644	1.904
2	0.615	0.648	1.352	0.292	0.544	0.487	0.641
3	0.292	0.295	0.421	0.139	0.431	0.487	0.233
4	0.171	0.174	0.463	0.086	0.153	0.144	0.173
5	0.114	0.114	0.181	0.062	0.152	0.144	0.172
6	0.082	0.083	0.216	0.051	0.149	0.144	0.072
7	0.063	0.063	0.098	0.045	0.121	0.144	0.72
8	0.051	0.051	0.116	0.043	0.051	0.050	0.051
9	0.043	0.043	0.060	0.045	0.051	0.050	0.051
10	0.037	0.037	0.067	0.051	0.051	0.050	0.051
11	0.033	0.033	0.040	0.062	0.051	0.050	0.051
12	0.030	0.030	0.042	0.086	0.051	0.050	0.031
13	0.028	0.028	0.031	0.139	0.051	0.050	0.031
14	0.027	0.027	0.029	0.292	0.050	0.050	0.031
15	0.026	0.026	0.026	1.100	0.043	0.050	0.031

Tabla 2.3. Varianzas de los coeficientes de transformada de una secuencia aleatoria con $\rho = .95$ y $N = 16$ (JAIN, Anil K.

Fundamentals of image processing).

De la tabla anterior es claro que la transformada KL compacta mejor la energía en menos datos debido a que los decorrelaciona de manera óptima. Como se mencionó, esta transformada es muy compleja porque depende de la secuencia de entrada para su cálculo. Como puede verse en la tabla 2.3 la transformada subóptima que más se aproxima al desempeño de la KL es la transformada de Coseno. Para el caso del procesamiento de imágenes digitales, las cuales se expresan en valores enteros positivos correspondientes a

los niveles de la imagen, se utiliza la transformada de coseno discreta (DCT), misma que se expresa como:

$$C_{ij} = \begin{cases} (1/(\sqrt{N})) & \text{si } i = 0 \\ (\sqrt{(2/N)) \cos \left| \frac{(2i+1)j\pi}{2N} \right|} & \text{si } i > 0 \end{cases}$$

donde N es el tamaño del bloque sobre el que se aplicará la transformada DCT, i y j son los índices de los renglones y las columnas respectivamente, del bloque a transformar. Esta transformada cumple con las condiciones de separabilidad expuestas anteriormente, por lo que la transformada de una matriz de pixeles P puede obtenerse a partir de la ecuación:

$$\text{DCT}(P) = C \cdot P \cdot C^T$$

2.3.3. Algoritmo de codificación de transformada.

Una vez definida la operación que realiza la transformación sobre los datos de las imágenes digitales, es posible exponer el algoritmo de codificación de transformada bidimensional el cual se esquematiza en la figura 2.8.

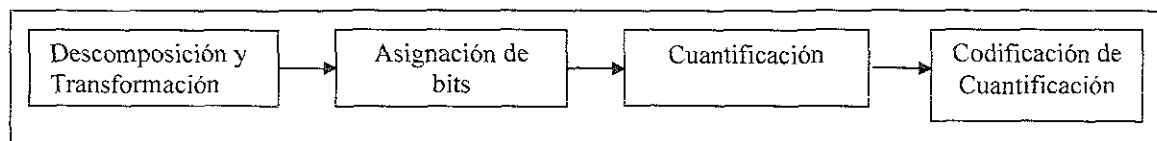


Figura 2.8. Esquema del algoritmo de codificación de transformada.

- Paso 1. Dividir la imagen dada en pequeños bloques cuadrados de tamaño N y aplicar a cada uno la transformada generada para el tamaño del bloque N .
- Paso 2. Determinar la asignación de bits. Comúnmente se utilizan matrices de asignación de bits que proporcionan más espacio para expresar los valores donde se encuentra compactada la energía de la matriz de coeficientes de transformada, es decir, la parte superior izquierda. La figura 2.9 muestra una matriz de asignación de bits para un bloque de coeficientes DCT de 16×16 , obteniendo una asignación promedio de 1 bit por pixel.

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

Figura 2.9. Asignación de bits para una matriz de coeficientes DCT de 16x16 (JAIN, Anil K. Fundamentals of image processing).

- Paso 3. Diseñar el cuantificador. Por lo general se utilizan matrices de divisores que buscan reducir el valor de los coeficientes en la medida de que esta reducción no afecte la apariencia de la imagen comprimida.
- Paso 4. Codificar la salida del cuantificador. En este paso los valores de los coeficientes de transformada, reducidos por efecto de la cuantificación, se convierten a palabras de código utilizando técnicas de codificación sin pérdida, como la codificación Huffman o la aritmética expuestas en el Anexo A, para transmitirse a través de un canal o para almacenarse en un dispositivo.
- Paso 5. Reproducir los coeficientes. Consiste en recuperar los datos de la imagen original, naturalmente reducidos por la cuantificación, por medio de la transformada inversa a la utilizada para el proceso de cuantificación. En el caso de la transformada DCT su inversa se expresa como:

$$P = C^T \cdot \text{DCT}(P) \cdot C$$

En la práctica, la DCT se utiliza en el estándar de compresión de imágenes JPEG, el cual se expone en la sección 5.5. En la misma sección se encuentra el ejemplo del efecto de la transformada DCT sobre un bloque de píxeles de una imagen muestra, así como de los pasos de cuantificación y de codificación de la salida del cuantificador. La figura 2.10 muestra el resultado de la aplicación de la DCT sobre una imagen original, y la figura 2.11 muestra el diagrama de la compresión por transformada.



Figura 2.10. Imagen original e imagen codificada con DCT (JAIN, Anil K. Fundamentals of image processing).

2.4. COMPRESIÓN SUB-BANDA.

El denominado sistema de análisis/síntesis sub-banda se utilizó por primera vez para datos unidimensionales (1-D) tales como señales de audio por Croisier en 1976. Smith propuso soluciones con la propiedad de reconstrucción perfecta de la señal codificada. La extensión a señales bidimensionales (2-D) fue presentada por Vetterli, y posteriormente se aplicó en la compresión de imágenes.

2.4.1. Definición de sub-banda.

El concepto de codificación sub-banda se basa en dividir los datos de entrada de la imagen en varias bandas de frecuencia, de forma similar a la codificación de transformada pero aplicada a toda la imagen, obteniéndose en su forma más simple, dos bandas de frecuencia. Una banda para representar los datos de baja frecuencia y otra banda para los datos de mayor frecuencia. Estas sub-bandas son posteriormente codificadas separadamente, transmitidas o almacenadas y después decodificadas y sintetizadas para obtener una imagen reconstruida a partir de ellas.

Un sistema de codificación sub-banda típico como el de la figura 2.12 muestra como mediante el uso de filtros de frecuencia digitales se divide la imagen de entrada en dos bandas, cada una con un cuantificador particular denotado por las Q's. Así mismo, cada banda tiene un codificador entrópico particular que le asigna palabras de código, mismo que se identifica por las E's. Los bloques D's denotan a los decodificadores entrópicos

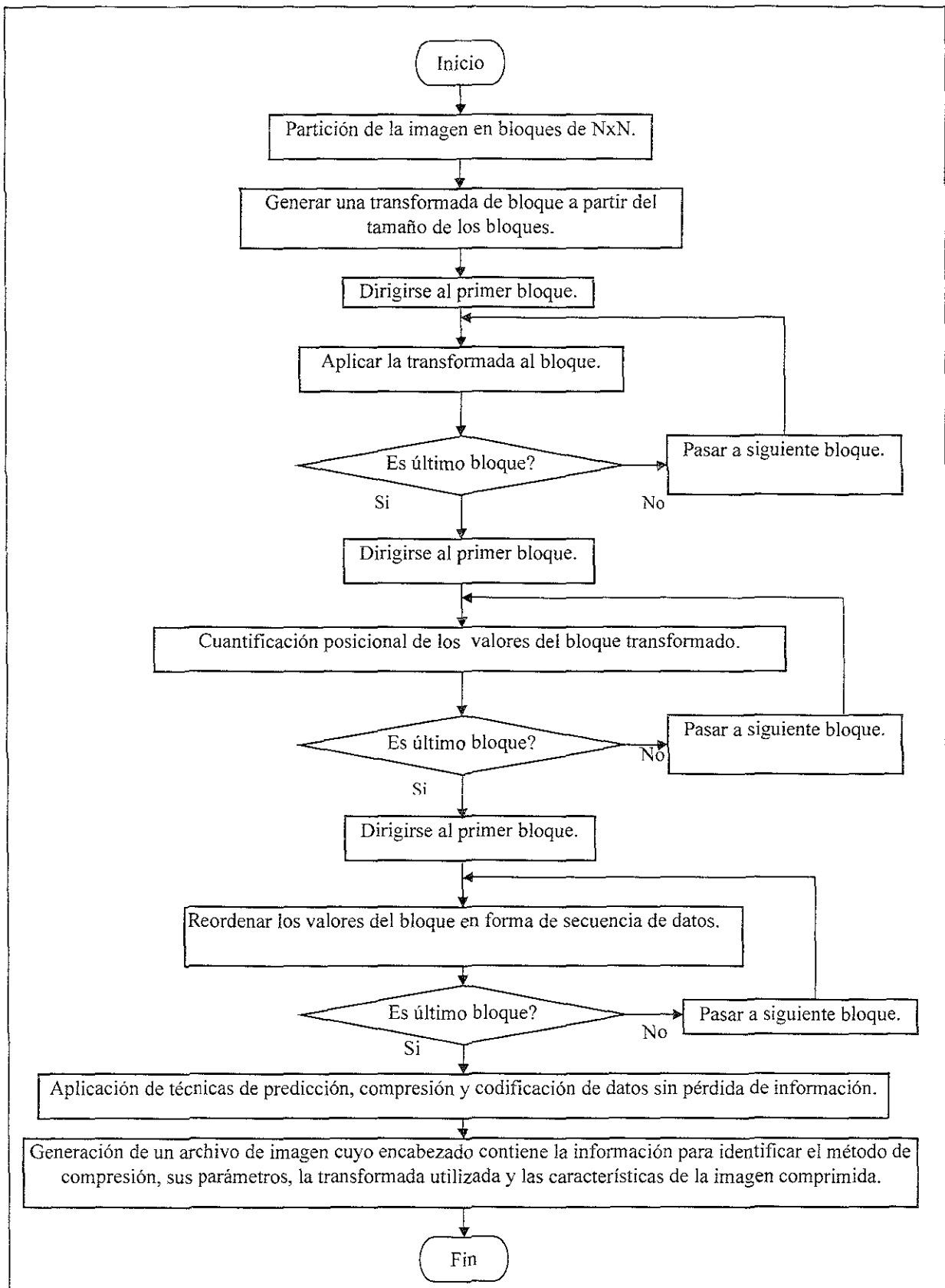


Figura 2.11. Diagrama de flujo de la compresión por transformada.

utilizados en el proceso de descompresión de las sub-bandas, las cuales son filtradas de forma inversa para obtener los datos de la imagen original y así obtener la imagen descomprimida.

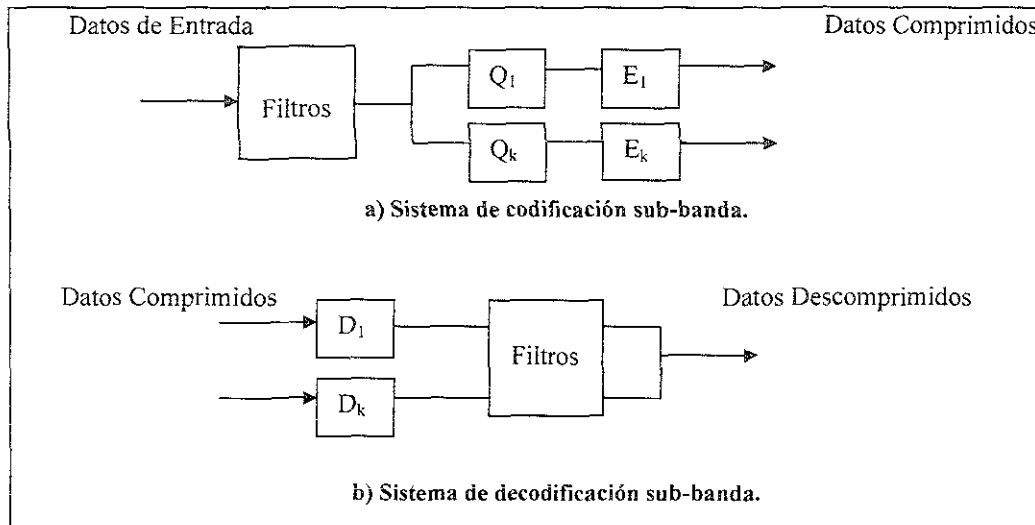


Figura 2.12. Esquemas de la codificación y la decodificación sub-banda (KOU, Weidong. Digital image compression. Algorithms and standards).

2.4.2. Tipos de filtros digitales, filtros QMF.

Para dividir una banda completa (es decir, la imagen) en varias sub-bandas, se necesitan filtros digitales. Sin embargo, ningún filtro tiene una respuesta de frecuencia verdaderamente rectangular. Esta situación propicia un tipo de superposición de frecuencias como la expuesta en la sección 1.4.1 la cual ocasiona un efecto denominado **aliasing**. La cuestión en la selección de los filtros a utilizar es si permiten la eliminación de este efecto. Esteban y Galand proporcionaron una solución a este problema con la introducción de los denominados bancos de filtros de espejo de cuadratura (QMF).

Los sistemas de filtros en su sentido general pueden esquematizarse de la manera ilustrada en la figura 2.13 y describirse mediante la siguiente secuencia de pasos:

- Paso 1. Los datos de una imagen original se descomponen en dos bandas de frecuencia, lo que se denomina como análisis, por medio de filtros de paso bajo y de paso alto expresados por $H_0(z_1, z_2)$ y $H_1(z_1, z_2)$ respectivamente.
- Paso 2. Posteriormente al filtrado de la imagen, se efectúa un submuestreo de las señales obtenidas por medio de la función $f\downarrow$ la cual conserva una de cada dos

muestras en la dirección horizontal y la vertical. A este proceso se le denomina muestreo descendente (downsampling).

- Paso 3. Para la etapa de síntesis o reconstrucción de la imagen, los datos de frecuencia se muestrean de forma ascendente (upsampling) por medio de la función recíproca $f\uparrow$ la cual inserta ceros en los lugares donde el muestreo descendente desprecia valores.
- Paso 4. Finalmente se filtran las bandas de frecuencia alta y baja con los filtros $F_0(z_1, z_2)$ y $F_1(z_1, z_2)$ para regresar los datos del dominio de frecuencia hacia el dominio espacial, fusionándolos y generando la imagen reconstruida $x'(z_1, z_2)$.

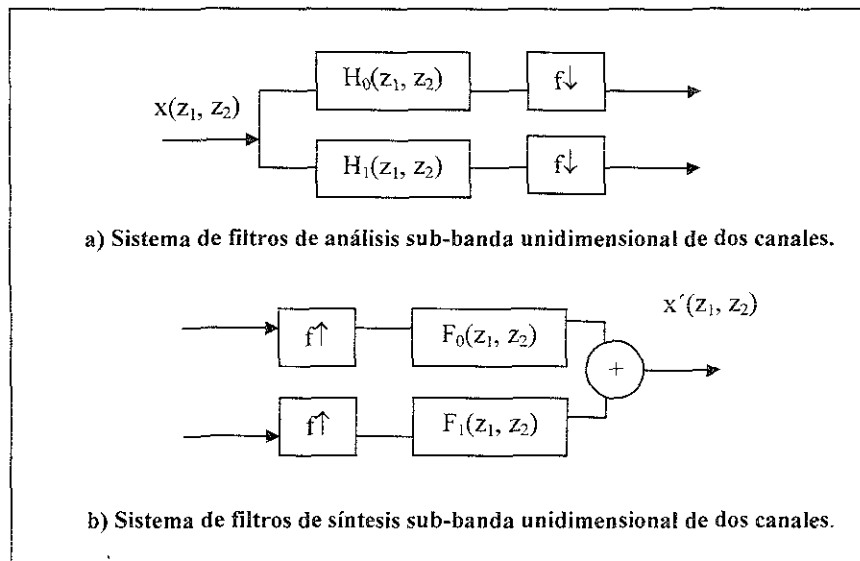


Figura 2.13. Esquemas de sistemas de filtros de análisis/síntesis bidimensionales (VETTERLI, Martin. "Multi-dimensional sub-band coding: Some theory and algorithms").

La estructura de los filtros QMF permiten eliminar el error de **aliasing**, que puede interpretarse como una imagen no deseada generada por el efecto de los filtros sin espectro de frecuencia rectangular. Si se sustituyen los filtros $H_0(z_1, z_2)$, $H_1(z_1, z_2)$ y $F_0(z_1, z_2)$, $F_1(z_1, z_2)$ generales por los filtros QMF $H(z_1, z_2)$, $H(-z_1, -z_2)$ y $H(z_1, z_2)$, $-H(z_1, z_2)$ respectivamente puede obtenerse la señal de la imagen reconstruida a partir de que:

$$\begin{aligned} x'(z_1, z_2) &= \frac{1}{2} [H(z_1, z_2) \cdot x(z_1, z_2) + H(-z_1, -z_2) \cdot x(-z_1, -z_2)] \cdot H(z_1, z_2) \\ &\quad + \frac{1}{2} [H(-z_1, -z_2) \cdot x(z_1, z_2) + H(z_1, z_2) \cdot x(-z_1, -z_2)] \cdot H(-z_1, -z_2) \\ &= \frac{1}{2} [H^2(z_1, z_2) - H^2(-z_1, -z_2)] \cdot x(z_1, z_2) \end{aligned}$$

De las ecuaciones anteriores se aprecia que el término de **aliasing** al que corresponde $x(-z_1, -z_2)$ desaparece automáticamente.

Si se asume que $H(z_1, z_2)$ es un filtro simétrico bidimensional de dimensiones $L_1 \times L_2$, su función de transferencia, es decir la transformada que aplica sobre los datos, puede escribirse como:

$$H(e^{jw_1}, e^{jw_2}) = \exp[-j(((L_1 - 1)/2)w_1 + ((L_2 - 1)/2)w_2)] \cdot H^*(w_1, w_2)$$

donde $H^*(w_1, w_2)$ es real.

A partir de esta definición, y considerando que $H(-e^{jw_1}, -e^{jw_2}) = H(e^{jw_1+\pi}, e^{jw_2+\pi})$, es posible expresar los datos de la imagen reconstruida en términos del filtro utilizado como:

$$x'(w_1, w_2) = \frac{1}{2} \cdot \exp[-j(((L_1 - 1)w_1 + (L_2 - 1)w_2)] \\ \times [H^{*2}(w_1, w_2) + H^{*2}(w_1+\pi, w_2+\pi)] \cdot x(w_1, w_2).$$

2.4.3. Algoritmo de codificación sub-banda.

El método de transformación de los filtros de espejo de cuadratura bidimensionales (QMF 2-D) usado para codificación sub-banda de imagen permite una reconstrucción casi perfecta de la señal de entrada libre de **aliasing**, sin contar aún con la operación de cuantificación. A partir de esto, es posible establecer los pasos para la operación de un codificador sub-banda los cuales se describen a continuación:

- Paso 1. Diseñar los QMF's.
- Paso 2. Asignar bits para las sub-bandas.
- Paso 3. Diseñar un cuantificador especial para cada sub-banda.

Paso 1. Diseñar los QMF's.

Un diseño de filtros QMF bidimensionales se muestra en la figura 2.14.

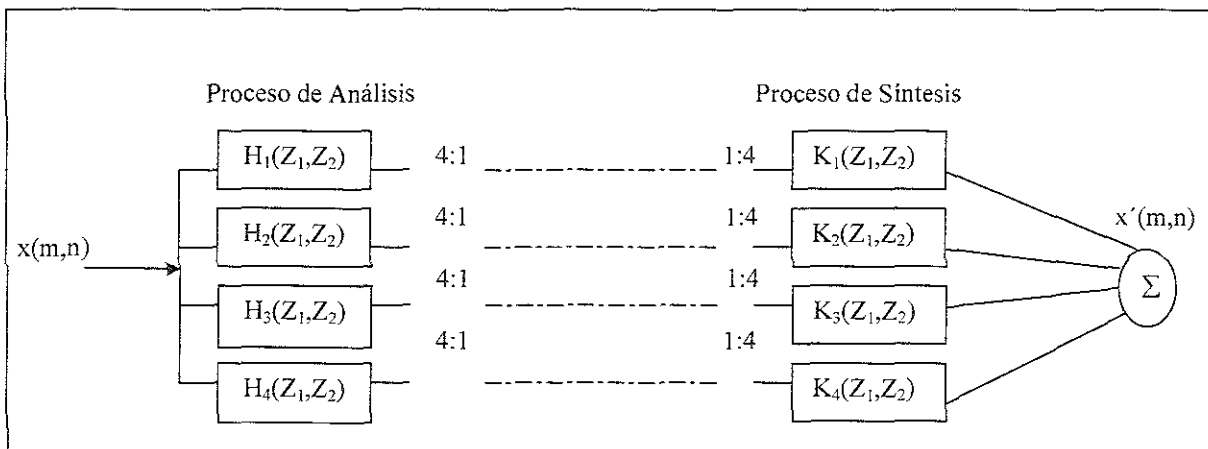


Figura 2.14. Modelo de banco de filtros QMF bidimensionales (VETTERLI, Martin. "Multi-dimensional sub-band coding: Some theory and algorithms").

En el sistema de la figura 2.14, $H_i(Z_1, Z_2)$ y $K_i(Z_1, Z_2)$ son transformaciones efectuadas por los filtros $h_i(n, m)$ y $k_i(n, m)$ para $i = 1, 2, 3, 4$ y $4:1$ denota un submuestreo de 1 de cada 4 muestras. Es decir, que el modelo propuesto separa la imagen de entrada $x(m, n)$ en cuatro sub-bandas, siendo la banda de baja frecuencia $H_1(Z_1, Z_2)$ y $H_4(Z_1, Z_2)$ la banda de mayor frecuencia. En este sentido, deben satisfacerse las siguientes condiciones para lograr una reconstrucción libre de **aliasing** de la señal de entrada $x(n, m)$

$$H_2(Z_1, Z_2) = -K_2(Z_1, Z_2) = H_1(-Z_1, Z_2) \tag{1}$$

$$H_3(Z_1, Z_2) = -K_3(Z_1, Z_2) = H_1(Z_1, -Z_2) \tag{2}$$

$$H_4(Z_1, Z_2) = K_4(Z_1, Z_2) = H_1(-Z_1, -Z_2) \tag{3}$$

$$H_1(Z_1, Z_2) = H_{11}(Z_1) H_{12}(Z_2) \tag{4}$$

donde la última condición (4) denota el uso de filtros de tipo separable. Esta característica, similar a la propiedad de separabilidad de las transformadas de la sección 2.2, simplifica el problema de filtrado QMF de ser bidimensional a un filtrado QMF unidimensional para el que es válido que

$$|H^{*2}(w_1) H^{*2}(w_2) + H^{*2}(w_1 + \pi) H^{*2}(w_2 + \pi)| = 1.$$

La característica de separabilidad de $H_1(z_1, z_2)$ proporciona un método alternativo para implementar los bancos QMF de la figura 2.14. Por ejemplo, se pueden obtener las mismas cuatro bandas efectuando el filtrado primero a lo largo de cada renglón de la imagen de entrada y después a lo largo de cada columna. El sistema obtenido se muestra en la figura 2.15.

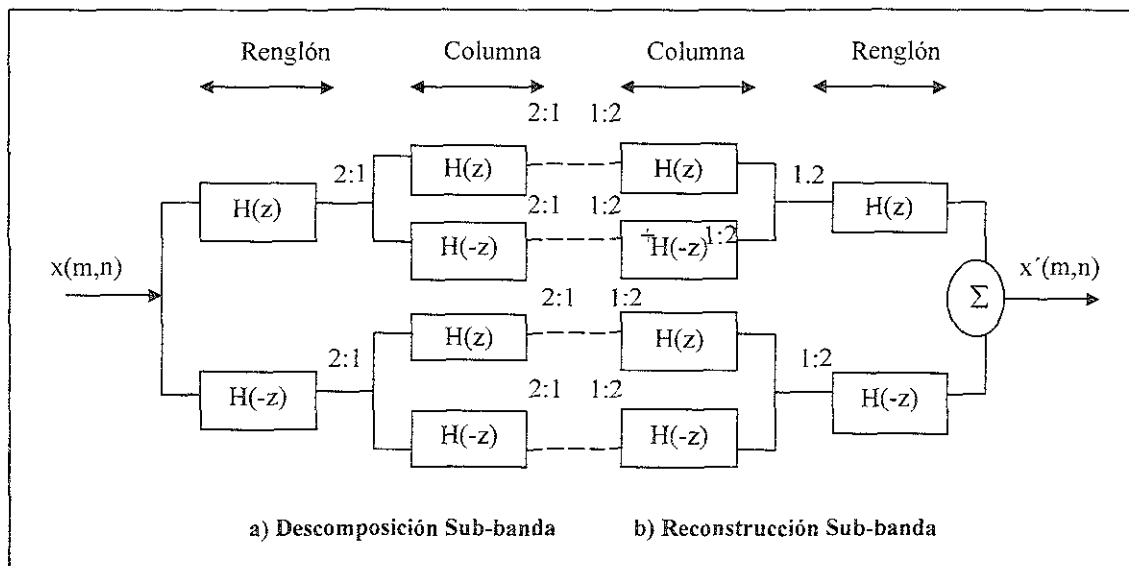


Figura 2.15. Sistema de análisis/síntesis sub-banda con filtros QMF bidimensionales separables (VETTERLI, Martin. "Multi-dimensional sub-band coding: Some theory and algorithms").

En la figura 2.15(a) se muestra cómo la imagen de entrada se descompone en dos bandas, de frecuencia baja y alta sobre los renglones y después sobre las columnas. Ambos pasos van seguidos de muestreo descendente de dos muestras. Posteriormente, cada sub-banda se descompone nuevamente en bandas de paso bajo y paso alto y se realiza el mismo submuestreo. La recuperación de la imagen esquematizada en 2.15(b) se lleva a cabo procediendo de forma inversa, muestreando ascendentemente los datos, transformándolos con su respectivo filtro primero sobre las columnas, sintetizando la información y procediendo de igual forma sobre los renglones. Este esquema de codificación, debido a la característica de separabilidad de los filtros QMF, brinda un resultado como el que se muestra en la figura 2.16.

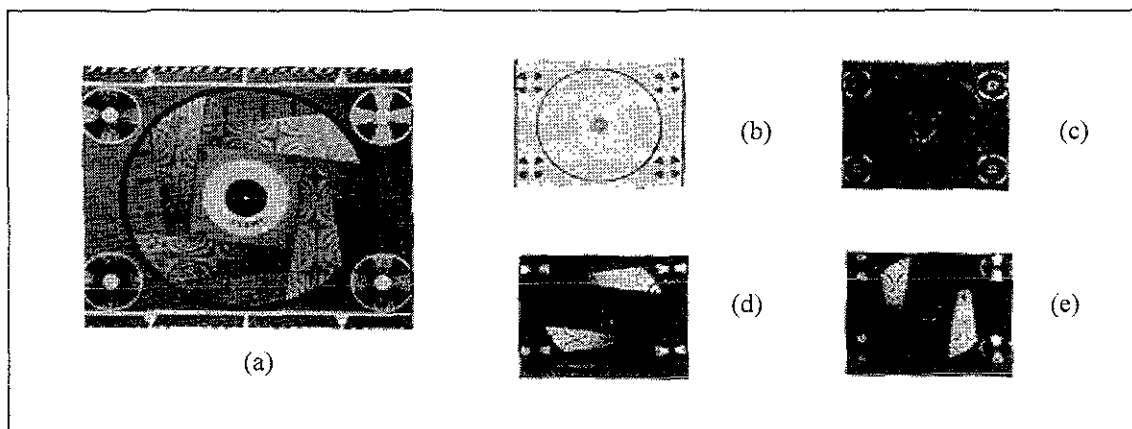


Figura 2.16. Descomposición de una imagen en cuatro sub-bandas. (a) Imagen original. (b) Banda más baja. (c) Banda diagonal. (d) Banda horizontal. (e) Banda vertical (GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images").

Como puede verse en la figura 2.16, los bordes contundentes horizontales y verticales se han extraído por medio de sus dos bandas respectivas. (c) representa la banda diagonal ya que sólo son visibles en ella los bordes diagonales. Note que la descomposición de la señal de entrada puede extenderse a más de cuatro bandas (7,10,13,16,32,...) repitiendo el proceso anterior en cada señal sub-banda en una forma de estructura de árbol. La figura 2.17 muestra un ejemplo de un sistema de banco QMF que proporciona una descomposición de n bandas de la señal de entrada.

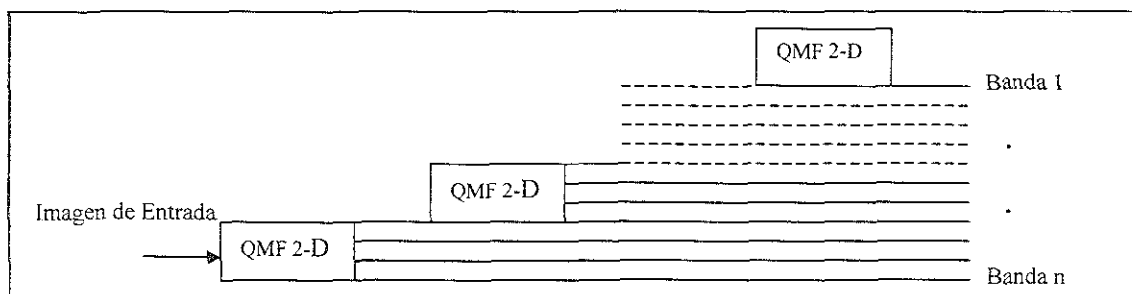


Figura 2.17. Sistema QMF de n sub-bandas (GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images").

Esta es una descomposición no uniforme del espectro de la imagen donde la banda más baja sufre una mayor descomposición por medio de otro QMF 2-D.

Paso 2. Asignar bits para las sub-bandas.

De forma similar a la expuesta en la sección 2.2.3, la asignación de bits en la codificación sub-banda busca asignar una mayor cantidad de bits a los componentes que contienen más información relevante para la imagen completa. En el caso de este esquema de codificación, la parte más importante se encuentra en los datos de salida del filtro de paso bajo, mientras que la salida del resto de los filtros puede almacenarse con menor precisión.

Paso 3. Cuantificación para cada sub-banda.

Una vez que se ha separado a la imagen en las sub-bandas determinadas por el modelo utilizado y que se les ha asignado una cantidad de bits de precisión, es necesario seleccionar un método de codificación de la información adecuado para las características particulares de cada sub-banda. La sub-banda de las frecuencias bajas debe codificarse con un método que permita ajustar el nivel de cuantificación de forma tal que la eliminación que realice sea mínima. Esto en virtud de que dicha sub-banda es la que contiene la información más representativa de la imagen. En cambio, las sub-bandas de las frecuencias superiores pueden codificarse con un método que realice mayor cuantificación de datos.

Como ejemplo, puede considerarse el uso de dos métodos de codificación para cuantificar las sub-bandas de un modelo de 4 sub-bandas, en particular la codificación escalar y la codificación DPCM como la expuesta en la sección 2.2.1. Excepto para las banda más baja, la correlación entre pixeles de las señales de banda mayor es muy baja. Por lo tanto, el método DPCM sólo se considera para codificar la banda más baja, mientras que las señales de banda superior se codifican con cuantificación escalar directamente, es decir que el valor de entrada se cuantifica con un escalar independiente. La figura 2.18 esquematiza este proceso.

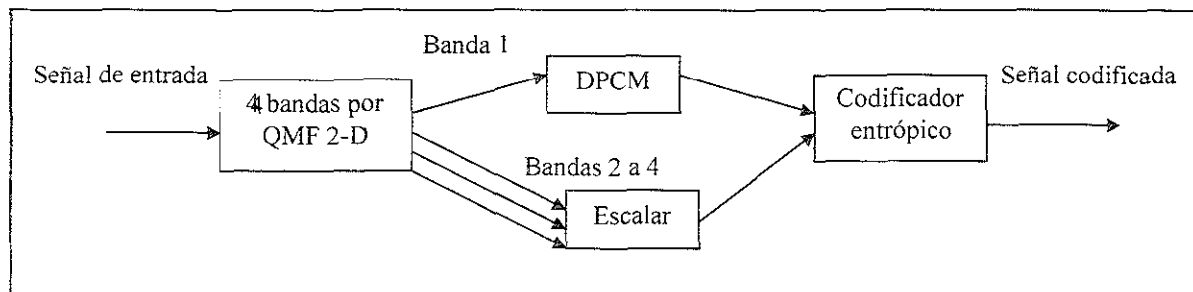


Figura 2.18. Esquema de codificación de sub-bandas de 4 bandas.

Codificación DPCM de la banda más baja.

Como se mencionó anteriormente, la banda más baja contiene el área de brillo general, y ya que la correlación entre píxeles de esta banda es alta se aplica por lo tanto, un método de codificación diferencial a la banda más baja de la señal de entrada. En el codificador DPCM usado en este ejemplo, un predictor lineal construye su píxel actual como una suma ponderada de los píxeles previos. La forma en que se obtiene el valor del píxel en la predicción bidimensional se muestra en la figura 2.19.

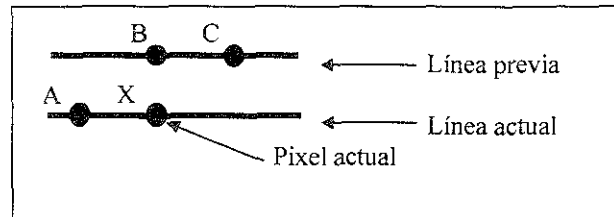


Figura 2.19. Configuración de los píxeles usados para la predicción (GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images").

La ecuación utilizada para determinar el valor de predicción del píxel es:

$$X = 0.5 A + 0.25 B + 0.25 C$$

donde A, B y C son los píxeles reconstruidos previamente y X es la predicción del píxel actual. La señal del error de predicción se cuantifica con un cuantificador no uniforme simétrico.

Codificación escalar de las señales de banda alta.

Una aproximación óptima es diseñar un cuantificador que se ajuste a la función de densidad de probabilidad de los datos en la señal de entrada. Sin embargo, esta opción no es conveniente para codificar las señales de banda de frecuencia alta. Esto se debe principalmente a la existencia de ruido en la imagen (ruido provocado por el movimiento del dispositivo de captura como una cámara) el cual se manifiesta como una señal de bajo nivel dentro de estas bandas, y resultaría en una cuantificación alta para el ruido, y en una cuantificación poco efectiva para los puntos de contorno que caen dentro del rango del cuantificador. Esto implicaría que el cuantificador no actúe convenientemente sobre la información relevante de las bandas de paso alto debido al ruido de baja frecuencia.

Esta circunstancia lleva al diseño de un cuantificador simétrico con las siguientes características: a) el cuantificador tiene una zona muerta central, d , la cual descartará los valores comprendidos en su rango para eliminar el ruido de la imagen; b) el rango del

cuantificador se fija por los límites inferior y superior, $\pm t$, para cubrir un cambio moderado en la luminosidad en los puntos de bordes; c) la señal que cae dentro del rango activo se cuantifica uniformemente en L niveles; d) las señales más allá del límite $+t$ o por debajo del límite $-t$ son mapeadas hacia los valores máximos de $+y$ e $-y$, respectivamente. Lo anterior se esquematiza en la figura 2.20.

El criterio a la hora de seleccionar los parámetros d , t , y L se basa en la disyuntiva entre los requisitos de fidelidad y de proporción de compresión. Note que algunas áreas de una imagen pueden contener pequeños cambios en la luminosidad (como en el caso del sombreado detallado de un rostro) que podrían ser eliminados por la zona muerta central del cuantificador. De forma similar, la señal puede ir más allá del límite del rango del cuantificador debido a los bordes contundentes. Sin embargo, este tipo de distorsión puede ser tolerada por el ojo humano. Además, la zona muerta central puede reducir de forma importante el ruido y mejorar la apariencia de imagen reconstruida. Esto también ayuda a la codificación eficiente de las señales de bandas altas en la etapa final del proceso de codificación.

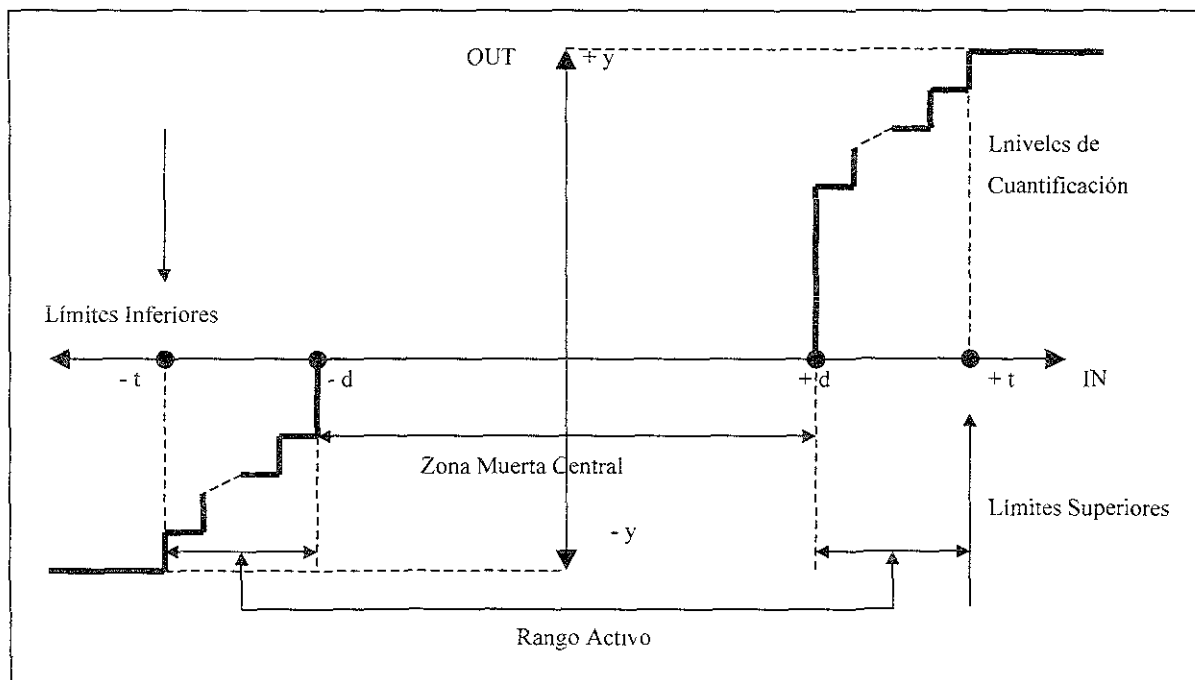


Figura 2.20. Cuantificador para bandas de alta frecuencia (GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images").

Como se mencionó anteriormente, las señales de banda alta contienen la información de los contornos y el ruido de baja energía de la imagen. Una vez que se ha eliminado el ruido por medio del cuantificador, el número de puntos de contorno se reduce drásticamente. Este hecho puede incrementar en mucho la eficiencia de la codificación cuando los puntos de contorno cuantificados se transmiten al receptor junto con su información posicional

respectiva. La información posicional se codifica con codificación run-length, considerando cada línea explorada como una secuencia de corridas blancas y negras donde las corridas blancas corresponden a la longitud de los puntos de contorno. La solución óptima es diseñar dos conjuntos de palabras de código que puedan acoplarse a los estadísticos de las corridas blancas y negras. Se afirma que para un conjunto dado de probabilidades de mensaje, el código más eficiente es el código Huffman. Sin embargo, por lo general los códigos Huffman carecen de estructura y su implementación podría ser complicada cuando el número de mensajes es grande. En un punto de equilibrio entre compresión y facilidad de implementación, se opta por el uso de una clase especial de códigos unidimensionales llamados códigos B_1 .

En general, los códigos B_1 de longitud de bloque (n, m) se construyen de la forma siguiente: Las palabras de código correspondientes a una longitud de corrida dada consisten en una secuencia de bloques de longitud $(n + 1)$ y $(m + 1)$ bits para corridas negras y blancas, respectivamente. El primer bit de cada bloque, denominado el bit de color, se utiliza para indicar el color de la corrida codificada (corridas negras o blancas). Los bloques subsecuentes con el mismo bit de color forman una palabra de código. En el proceso de decodificación, el bit de color se usa para examinar si un nuevo bloque pertenece a la palabra de código que se está codificando o es el inicio de una nueva palabra de código. En la tabla 2.4 se muestra un ejemplo de palabras de código B_1 correspondientes a corridas negras o blancas $(n = m = 1)$ con longitudes de 1 a 4.

Longitud de Corrida	Palabra de Código
1	C 0
2	C 1
3	C 1 C 0
4	C 1 C 1

Tabla 2.4. Ejemplo de Tabla de Código B_1 de corridas negras y blancas (GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images").

Como una visión sintetizada del método descrito hasta ahora, en la figura 2.21 se muestra un diagrama general de la compresión sub-banda y posteriormente la descripción del proceso de compresión sub-banda aplicado a una imagen monocromática y a una a color.

Para ejemplificar el proceso de codificación sub-banda en imágenes monocromáticas, considérense las imágenes de la figura 2.22. Para estas imágenes, la entrada se dividió en cuatro bandas usando un sistema de bancos QMF bidimensionales. La imagen de banda más baja la cual contiene el área de brillo general de la imagen completa se codificó con DPCM utilizando predicción bidimensional con un cuantificador no uniforme de 31 niveles. El análisis de la señal de error para varias imágenes ha mostrado que un método

eficiente para codificar la señal de error es usar un esquema de codificación de longitud variable (VLC) como el expuesto en la sección 1.5.1.

En base al análisis estadístico de los datos, se derivaron palabras de código VLC de 31 niveles, lo cual concuerda de forma razonable con los estadísticos de una imagen cuantificada típica. La tabla 2.5 muestra las palabras de código con sus respectivas longitudes de palabra.

Debe enfatizarse que el grado de exactitud para codificar la primera banda juega un papel importante para mejorar la apariencia de la imagen reconstruida. Como se mencionó anteriormente, debe diseñarse un cuantificador especial para la banda de baja frecuencia para permitir la asignación de valores de entre un conjunto mayor de posibilidades. En el caso de que se hayan determinado 2^n niveles para el cuantificador de las bandas altas, puede usarse el mismo cuantificador para la banda baja con 2^{n+1} niveles de cuantificación. Exceptuando el parámetro de zona muerta central, el cual es el mismo para todas las bandas (por ejemplo, $d=8$), los parámetros restantes del cuantificador pueden seleccionarse de formas distintas. Por ejemplo, para las bandas 3 y 4, las cuales representan las bandas horizontal y vertical, el número de niveles de cuantificación en el rango activo fue de $L = 14$. Los límites superior e inferior se seleccionaron como ± 36 y ± 42 . En la transmisión de las señales de las bandas altas sólo se transmiten los valores DPCM diferentes de cero cuantificados junto con su información de posición. Esta información se codificó con la técnica run-length línea por línea usando palabras de código B_1 .

Codificación sub-banda de imágenes a color.

En general, las imágenes a color contienen los tres colores primarios: rojo, verde y azul, lo que equivale a la representación RGB. Para aplicaciones de compresión de imagen, estos tres colores primarios se transforman a la representación YUV como se mostró en la sección 1.1.2. Esta representación explota mejor la correlación entre los valores RGB para imágenes digitales de escenas naturales con baja saturación de color. La transmisión digital de señales a color por medio de la representación YUV resulta más conveniente debido a la reducción que realiza de los componentes de cromancia y a la mejor representación de la luminancia diagonal. La compactación de energía que se logra gracias a la transformación de los componentes RGB a YUV es casi tan óptima como la lograda por la transformación KL sobre la información de los píxeles. Además, dicha transformación en la representación de color tiene la ventaja de ser compatible con las señales monocromáticas en virtud de que el componente Y corresponde a una versión a escala de gris de la imagen a color. En la transformación YUV la mayoría de la energía se concentra en la luminancia, dejando poca energía a los componentes de cromancia (U y V). Como resultado, el componente de alta energía necesita codificarse con mayor exactitud que los componentes de baja energía.

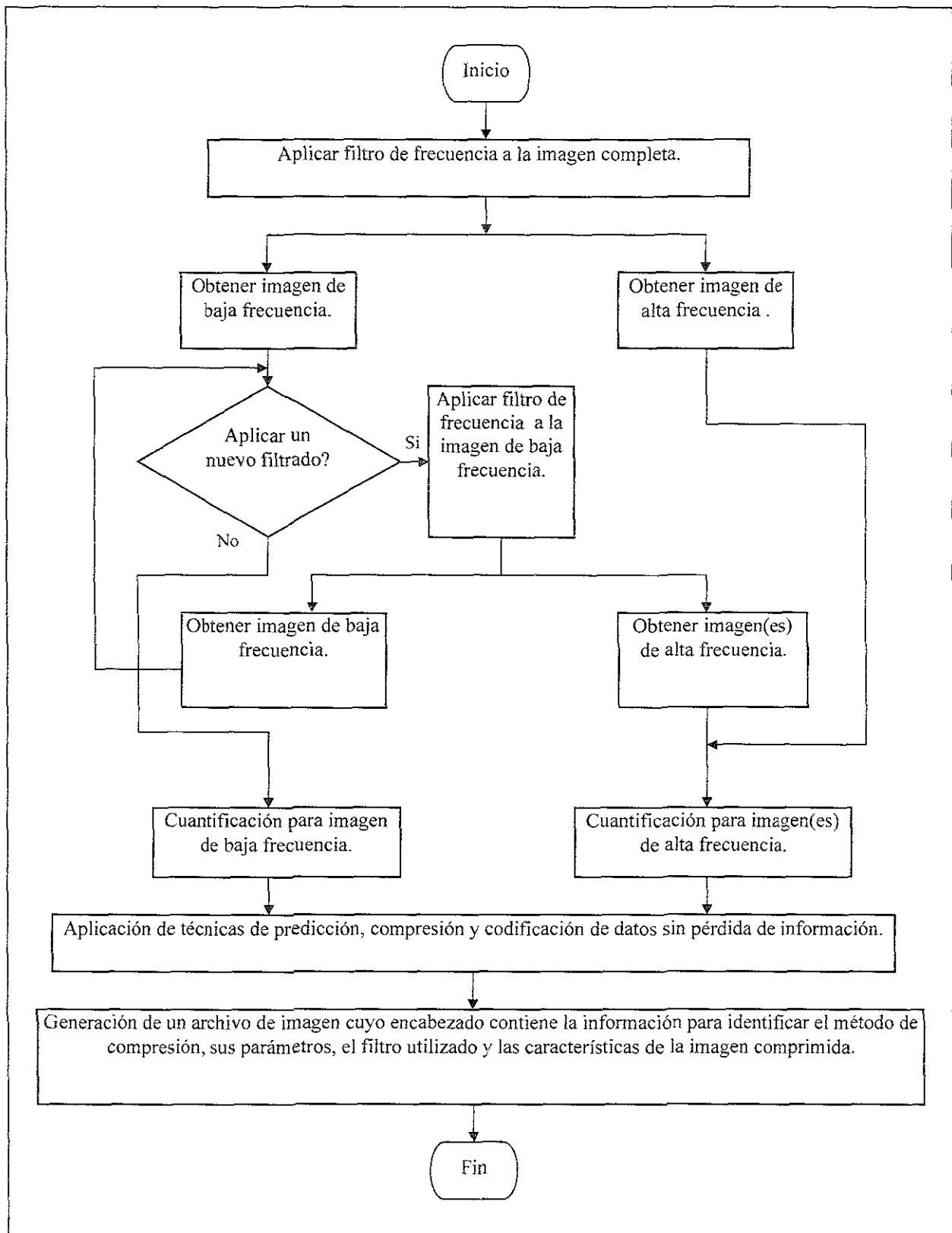


Figura 2.21. Diagrama de flujo de la compresión sub-banda.



Figura 2.22. Imágenes originales comprimidas con codificación sub-banda. Imagen reconstruida de Karen con SNR de 33.78 e imagen reconstruida de Lena con SNR de 33.95 (GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images").

Nivel	Palabra de Código	Bits
1	0	1
2	100	3
3	1110	4
4	1010	4
5	11110	4
6	11001	5
7	10111	5
8	10110	5
9	111111	6
10	110111	6
11	110110	6
12	110101	6
13	110000	6

14	1111100	7
15	1101001	7
16	1101000	7
17	1100011	7
18	11000101	8
19	11000100	8
20	111110110	9
21	111110101	9
22	11111011101	11
23	11111011100	11
24	11111010011	11
25	11111010010	11
26	11111010001	11
27	11111010000	11
28	111110111111	12
29	111110111110	12
30	111110111101	12
31	111110111100	12

Tabla 2.5. Palabras de código de longitud variable para codificación sub-banda (GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images").

Esto también concuerda con las características del sistema visual humano donde el ojo es menos sensible a la degradación en la cromancia que a la degradación en la luminancia. Por lo tanto, la eficiencia de codificar imágenes depende principalmente de la eficiencia con la que se codifique la señal de luminancia.

El mismo método de codificación sub-banda descrito anteriormente para imágenes monocromáticas se aplica para codificar el componente Y. Un método de codificación similar también puede aplicarse para codificar los componentes de cromancia, cuantificando las señales de banda alta con mayor rigor que la señal de luminancia. En el caso de una descomposición de 4 a 16 bandas, la señal de la banda más alta puede descartarse sin provocar ninguna diferencia notoria para el observador. Como resultado, las señales de cromancia sólo se filtran con paso bajo en las direcciones horizontal y vertical antes del submuestreo en cada dirección. La figura 2.23 muestra un ejemplo de imagen a color original y recuperada.



Figura 2.23. Imagen a color original comprimida con codificación sub-banda. Imagen reconstruida de pimientos con SNR de 32.95 para el componente Y, 32.22 para el componente U, y 32.67 para el componente V (GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images").

Las señales reconstruidas se producen interpolando los píxeles faltantes en las direcciones horizontal y vertical. Esto se logra insertando píxeles con valor cero antes de filtrar con paso bajo en cada dirección. El principio de filtrado QMF requiere que, para una reconstrucción libre de **aliasing**, también se incluyan las señales interpoladas de las bandas más altas. En una aplicación donde no se consideren estas bandas podría esperarse alguna degradación debido al **aliasing**. Sin embargo, gracias al ancho de banda menor de los componentes U y V con respecto a la señal de luminancia, al descartar los componentes de cromancia de las bandas más altas se puede tener un efecto mínimo en la apariencia final de la imagen reconstruida.

La imagen a color RGB primeramente se transformó a su representación YUV. El componente Y se codificó de la misma forma que las imágenes monocromáticas. En este caso, sin embargo, la banda más baja se codificó con DPCM usando un cuantificador no uniforme de 15 niveles. La señal de error cuantificada entonces se codificó con VLC utilizando las primeras 15 palabras de código de la tabla 2.5.

En el caso de los componentes de cromancia U y V, las señales altas se descartaron aplicando un filtro de paso bajo y submuestreo con un factor de 16 a 1 sobre dichos componentes. Las señales de cromancia submuestreadas posteriormente se codificaron usando un cuantificador no uniforme de siete niveles. Se asignó una palabra de código de longitud variable obtenida de las primeras siete palabras de código a la señal de error cuantificada. Para la reconstrucción de la imagen, las señales de cromancia fueron interpretadas aplicando el mismo filtro de paso bajo utilizado en el lado transmisor. Se efectuó también una interpolación 1 a 16 en dos etapas de filtrado bidimensional, agregando ceros en el lugar de los píxeles faltantes. Un paso similar se realiza para

interpolan las señales de luminancia, con la excepción de que en cada etapa también se agregaron las señales de banda alta a la señal de banda baja.

2.5. COMPRESIÓN WAVELET.

La Transformada Wavelet es un tipo de representación de señal que puede proporcionar el contenido de frecuencia de la señal en un instante particular de tiempo o en una situación particular en el espacio. Esta técnica analiza el contenido de frecuencia de una imagen completa y las separa en regiones bajas y altas en base a ese criterio. Una vez descompuesta la imagen, la región de más baja frecuencia puede analizarse nuevamente con las transformadas wavelet para ajustarse a los nuevos valores de la imagen de baja frecuencia y realizar una nueva segmentación. Una de las posibles implementaciones de la Transformada wavelet puede ser usando Filtros de Espejo de Cuadratura (descritos en la sección 2.4), y por lo tanto la compresión basada en wavelet puede verse como un tipo de codificación sub-banda.

La teoría wavelet no es una teoría nueva. En 1910, Haar descubrió un conjunto de wavelets llamado funciones Haar. En los años 30's, la teoría wavelet fue desarrollada más ampliamente por varios matemáticos como Levy, Lusin, Littlewood, Paley y Franklin. Sin embargo, las aplicaciones de la teoría wavelet en el procesamiento de señales e imágenes sólo tienen algunos años. La técnica de compresión de imagen llamada codificación wavelet es una de ellas.

2.5.1. Definición de wavelet.

Una wavelet es una función de una variable, de la cual se conoce su localización y es oscilante. Para entender una wavelet, tomemos como referencia una onda como una onda eléctrica u onda de agua. Dicha onda es una función del tiempo, del espacio o de ambos. Una wavelet es una pieza de onda. Es oscilante pero dentro de un período o una distancia/región limitados. Una ilustración de una onda y de una wavelet se muestra en la figura 2.24.

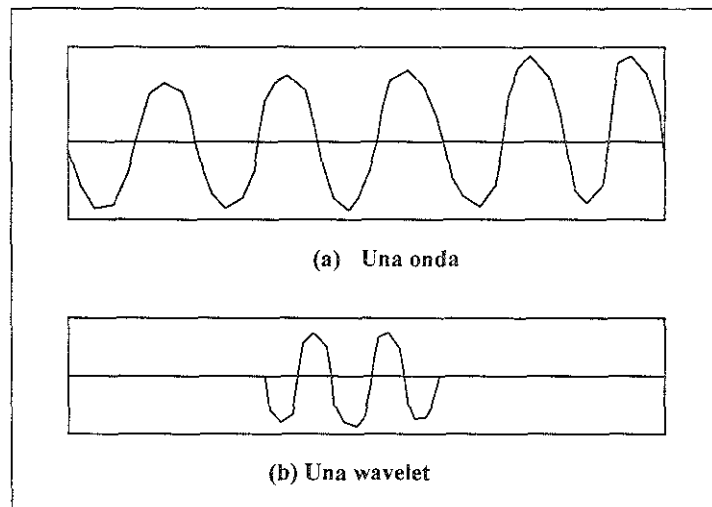


Figura 2.24. Diagramas de una onda y de una wavelet a partir de dicha onda (KOU, Weidong. Digital image compression. Algorithms and standards).

Existen dos clases de wavelets: las wavelets con escala en el tiempo, que en el campo del procesamiento de imágenes deben llamarse wavelets con escala en el espacio; y las wavelets con frecuencia en el tiempo, o para el mismo campo deben denominarse wavelets con frecuencia en el espacio.

Antes de discutir estas dos clases, primero hay que examinar la funcionalidad de las wavelets para el procesamiento de señales e imágenes. Un ejemplo es el alfabeto común, el cual constituye una herramienta para representar palabras escritas. Los símbolos del 0 al 9 permiten representar cualquier número entero. De forma similar, las wavelets proporcionan una forma para representar una señal. El hecho de que sea posible describir información por medio de un conjunto de símbolos o funciones es de mucha utilidad ya que dicha descripción puede facilitar la comprensión y el manejo de esa información. Con una representación wavelet, ciertas señales pueden analizarse y comprimirse más fácilmente. La representación wavelet es adecuada para señales no estacionarias que no son predecibles estadísticamente aún contando con el conocimiento estadístico de las muestras anteriores.

El funcionamiento de la representación wavelet puede explicarse tomando como ejemplo la representación decimal de un número. Para un entero dado 3456, este expresa una representación de $3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 6$. Esto es, el número se representa por medio de los coeficientes 3, 4, 5 y 6 con las bases 10^3 , 10^2 , 10^1 y 10^0 . De forma similar, una señal dada $f(x)$, puede representarse por los coeficientes $\alpha(j, k)$ y las wavelets asociadas:

$$\psi_{j,k}(x) : f(x) = \dots + \alpha(0,0)\psi_{0,0}(x) + \alpha(0,1)\psi_{0,1}(x) + \dots$$

2.5.2. Tipos de wavelets.

Existen dos clases de wavelets, las wavelets con escala en el espacio y las wavelets con frecuencia en el espacio. Las wavelets con escala en el espacio se generan a partir de una sola wavelet llamada wavelet madre por medio de operaciones de escalado y traslación. Para una wavelet madre dada $\psi(t)$, la cual satisface las siguientes dos condiciones:

1) Está bien localizada
$$\frac{\psi(t)}{|t|} \rightarrow 0 \text{ cuando } |t| \rightarrow \infty$$

2) Es oscilante
$$0 = \int_{-\infty}^{\infty} \psi(t) dt = \dots = \int_{-\infty}^{\infty} t^{m-1} \psi(t) dt$$

es posible generar el resto de las wavelets $\psi_{a,b}(t)$ de la familia wavelet con escala en el espacio por medio de la siguiente ecuación:

$$\psi_{a,b}(t) = (1/\sqrt{a})\psi((t - b)/a), a > 0, b \in \mathbb{R}$$

Esta clase de wavelet se utiliza para la descomposición a multiresolución de una señal. Dada una secuencia de resoluciones con un orden creciente, el contenido de información de una imagen a una resolución dada puede extraerse a partir de la diferencia entre su aproximación en la resolución actual y su aproximación en la siguiente resolución más baja. Las resoluciones están asociadas fuertemente con el concepto de escala. Usando las wavelets con escala en el espacio, una imagen es segmentada en regiones de frecuencia y cada región equivale a una versión de menor resolución de la imagen original. Posteriormente, la imagen de baja frecuencia puede tomarse como la nueva imagen original para aplicar sobre ésta la familia de wavelets que vuelva a segmentar la imagen.

Las wavelets con frecuencia en el espacio fueron descubiertas por Malvar. Él utilizó una aproximación del análisis de Fourier denominada "ventaneado" que consiste en cortar una señal $s(t)$ en partes y después aplicar el análisis de Fourier sobre ellas. Esta operación lleva a un conjunto de wavelets $w_{m,n}(t) = e^{iamt} w(t - bn)$, donde $w(t)$ es la ventana, $b > 0$ es la longitud promedio de las partes $w(t - bn)$ $s(t)$, y $a > 0$ está en función de b . Las wavelets Malvar son particularmente útiles para codificar voz y música.

Funciones Haar.

Las funciones Haar son un conjunto wavelets simples que fueron descubiertas por Haar en 1910, y aplicadas en los campos de la compresión y filtrado de imagen en los 70's. Como se muestra en la figura 2.25, las wavelets Haar son simplemente funciones rectangulares.

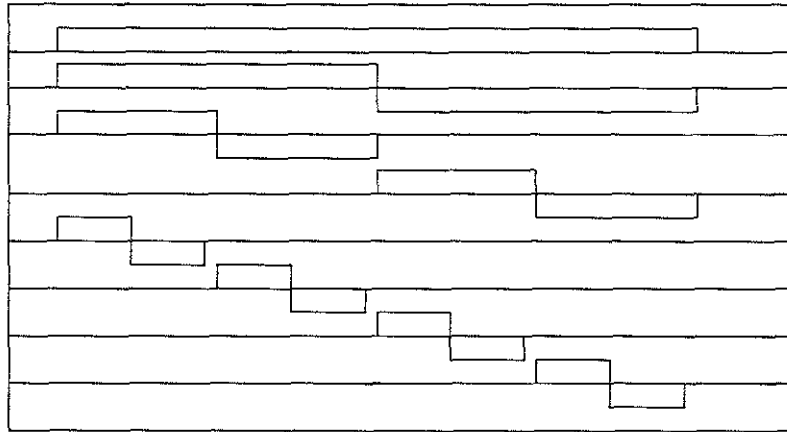


Figura 2.25. Las primeras ocho wavelets Haar (KOU, Weidong. Digital image compression. Algorithms and standards).

Las wavelets Haar son wavelets con escala en el espacio. La wavelet madre de las funciones Haar es $\psi(t)$, la cual es igual a 1 si $0 \leq t < 1/2$, a -1 si $1/2 \leq t < 1$, e igual a 0 si $t < 0$ o $t > 1$. El resto de las wavelets Haar pueden generarse por medio del escalado y la traslación con $a = 2^{-j}$ y $b = 2^{-j}k$. La figura 2.26 muestra una transformada Haar discreta de 8×8 .

1	1	1	1	1	1	1	1
1	1	1	1	-1	-1	-1	-1
$\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$	0	0	0	0
0	0	0	0	$\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$
2	-2	0	0	0	0	0	0
0	0	2	-2	0	0	0	0
0	0	0	0	2	-2	0	0
0	0	0	0	0	0	-2	-2

Figura 2.26. Matriz de transformada Haar (KOU, Weidong. Digital image compression. Algorithms and standards).

El principio básico consiste en aplicar la transformada wavelet Haar a la imagen completa para dividirla en sus componentes de frecuencia decorrelacionados de forma similar a la codificación sub-banda. Así se obtienen cuatro regiones agrupadas desde la imagen de baja frecuencia hasta la de alta frecuencia. Posteriormente puede utilizarse la wavelet madre para generar nuevamente una familia de wavelets completa que se aplique a los coeficientes de transformada de baja frecuencia, ahora considerando las características de sus estadísticos particulares. Estas regiones tienen un ordenamiento que depende de su aportación a la imagen, siendo los bloques superiores izquierdos los primeros en

codificarse y a partir de ellos, extenderse hacia el exterior de la imagen en una secuencia padre-hijo como la que se muestra en la figura 2.27. El resultado de este proceso se ilustra en la figura 2.28.

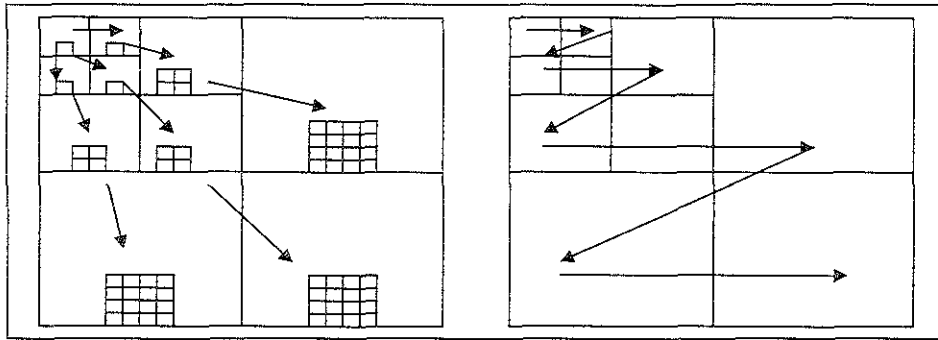


Figura 2.27. Padre-Hijo y su correspondiente ruta de exploración para codificación de coeficientes Wavelet (KOU, Weidong. Digital image compression. Algorithms and standards).

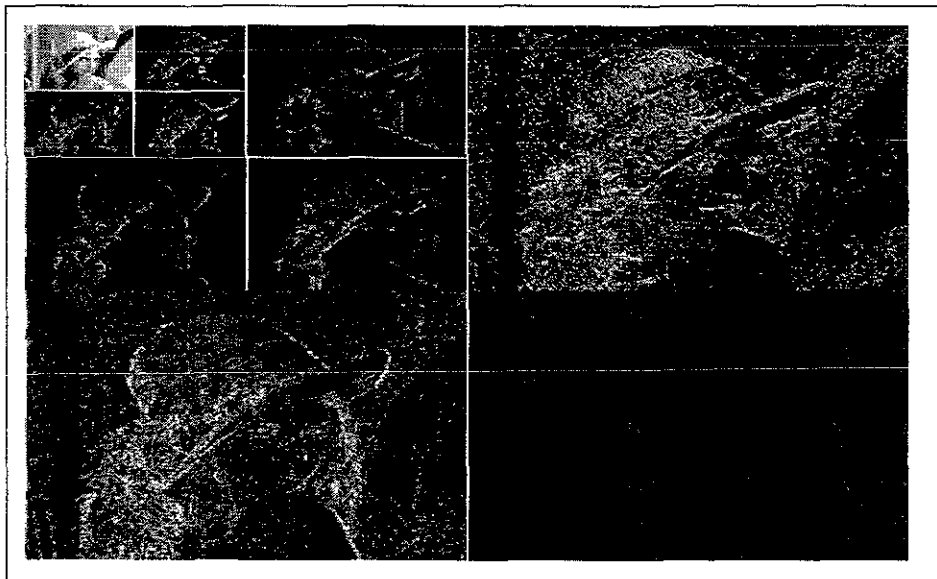


Figura 2.28. Ilustración de la descomposición de transformada wavelet sobre la imagen de Lena (KOU, Weidong. Digital image compression. Algorithms and standards).

Existen otras wavelets además de las wavelets Haar que pueden utilizarse para compresión de imágenes. Algunas de estas wavelets llevan a proporciones de compresión mayores que otras. El beneficio obtenido del análisis wavelet es que las características locales de una imagen pueden capturarse usando menos coeficientes. Codificando estos coeficientes, pueden comprimirse imágenes de forma eficiente.

2.5.3. Algoritmo de codificación wavelet.

Los algoritmos de compresión de imagen basados en wavelets pueden generalizarse como se muestra en la figura 2.29.

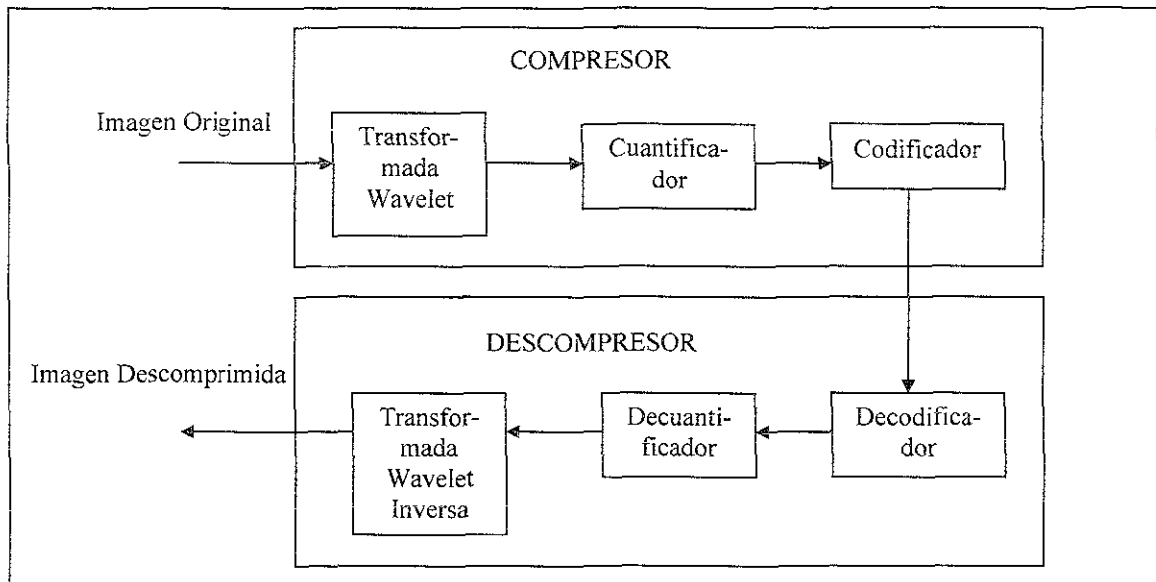


Figura 2.29. Esquema del compresor y descompresor de la codificación wavelet (KOU, Weidong. Digital image compression. Algorithms and standards).

Considerando las descripciones de la operación de las wavelets sobre las imágenes y la estructura derivada de la misma, puede establecerse que la compresión wavelet consiste en tres pasos:

- Paso 1. Transformada Wavelet para decorrelacionar los datos de la imagen.
- Paso 2. Cuantificación de los coeficientes transformados.
- Paso 3. Codificación de los coeficientes cuantificados.

Una vez que la transformada wavelet ha decorrelacionado los valores de la imagen transformada, reorganizando la información más importante en pocos coeficientes, lo que resta por establecer es la cantidad de bits a signar para cada sector de la imagen. Un posible modelo de asignación de bits se muestra en la figura 2.30.

Finalmente, en la figura 2.31 se muestra el diagrama general del proceso de compresión por transformada wavelet.

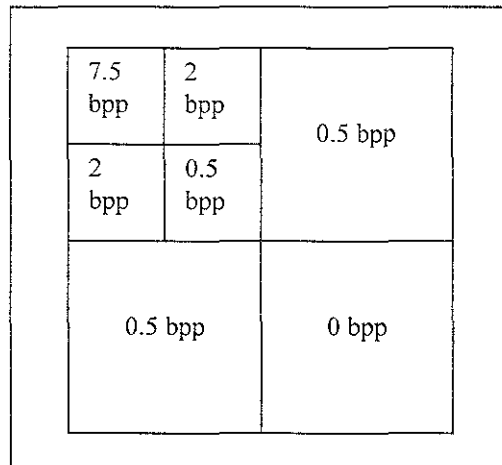


Figura 2.30. Esquema de asignación de bits para una imagen de 256x256 (bpp=bits por píxel. KOU, Weidong. Digital image compression. Algorithms and standards).

2.6. COMPRESIÓN DE CUANTIFICACIÓN POR VECTOR.

La cuantificación es uno de los pasos más importantes en cualquier esquema de compresión. Este proceso ya ha sido referido en secciones anteriores y es la etapa donde los algoritmos de compresión descartan la información que no es imprescindible para representar la imagen y así reducir su tamaño de manera importante. El objetivo de este proceso es proyectar los valores continuos de los coeficientes transformados resultantes en la forma de un conjunto finito de símbolos.

El esquema de cuantificación más popular es la cuantificación uniforme, la cual es ampliamente usada debido a su simplicidad y eficiencia. Como ya se ha mencionado anteriormente, esta cuantificación reduce el valor de los coeficientes obtenidos a partir de alguna transformación o cálculo que aproveche su correlación, generalmente asignando un divisor o un valor fijo para un rango de valores dado, cuyo efecto reductor aumentará conforme los coeficientes dejen de ser significativos. Este tipo de cuantificador sólo toma en cuenta valores individuales sin considerar sus valores vecinos, además de que se aplica de forma directa a todo el arreglo determinado por el codificador. Un ejemplo de este cuantificador, utilizado por el estándar JPEG, se muestra en la sección 5.5. Para una eficiencia de alta compresión, la cuantificación uniforme puede combinarse con una delimitación de zona muerta como la revisada en la sección 2.4.3 para aumentar el número de coeficientes cuantificados con valor cero. Recientemente se utilizan tamaños de paso de cuantificación mayores para cuantificar los coeficientes transformados de alta frecuencia tomando en cuenta la sensibilidad del sistema visual humano al contenido de frecuencia de una imagen.

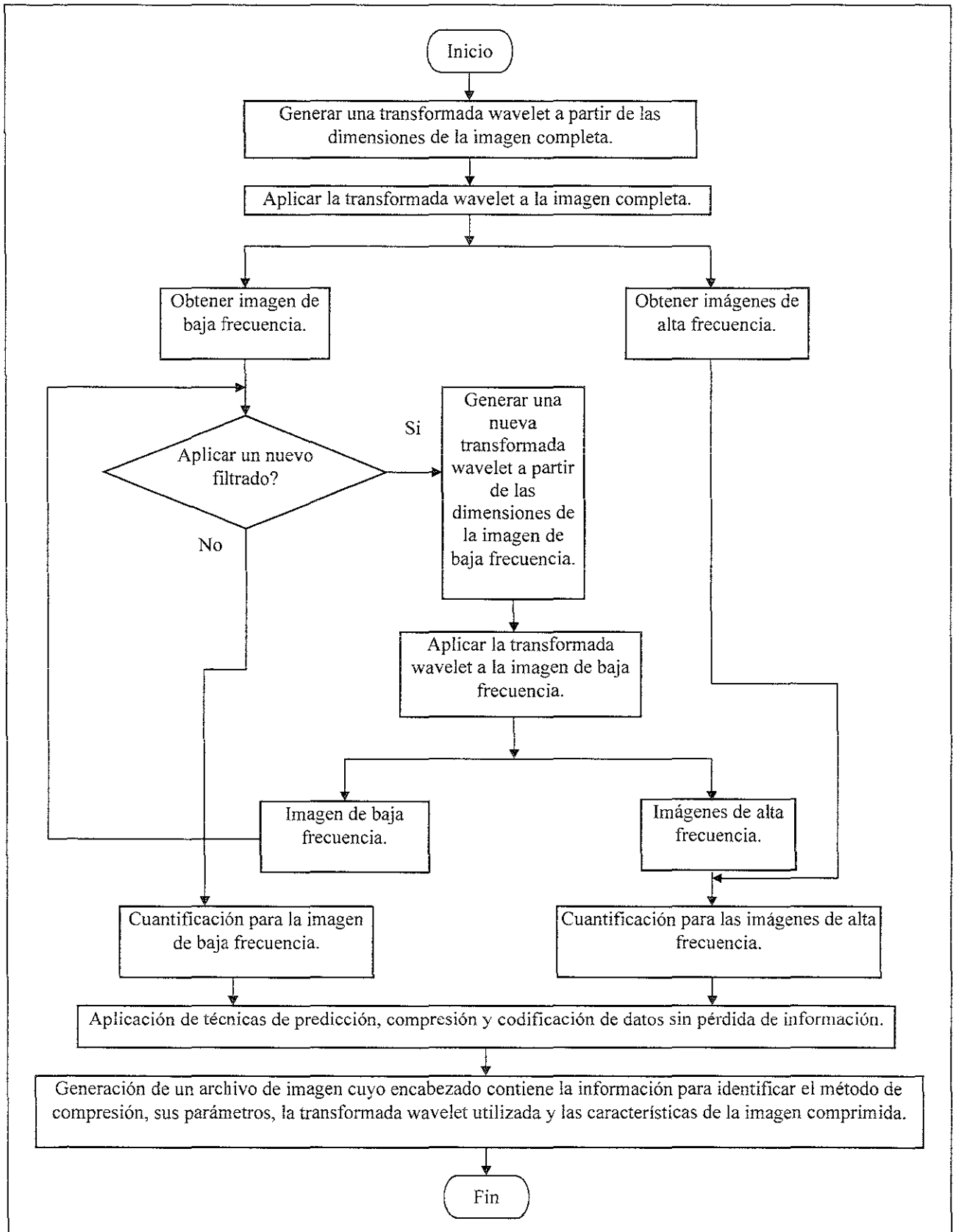


Figura 2.31. Diagrama de flujo de la compresión por transformada wavelet.

2.6.1. Definición de cuantificación por vector.

En contraste a la cuantificación escalar, la cuantificación por vector (VQ) es el proceso de “mapear” conjuntos de valores en forma de vectores dentro de un conjunto predefinido de patrones. De manera similar a la codificación de transformada, la cuantificación de vector (VQ) es una técnica de codificación de bloque. En lugar de codificar muestras escalares, el VQ opera en vectores, un conjunto ordenado de números reales. El principio detrás del VQ se basa en el trabajo de Shannon quién mostró que codificar vectores es mejor que codificar muestras escalares en términos de eficiencia de compresión. Un sistema VQ básico se muestra en la figura 2.32.

Desde el punto de vista de la codificación del bloque, la VQ explota la correlación entre píxeles en un bloque local, permitiendo asignar menos bits a cada píxel, llevando a una técnica de compresión eficiente. La operación de descompresión del VQ es muy simple y rápida debido a que no necesita procesamiento intensivo de información excepto para la localización de los vectores en la tabla de vectores.

2.6.2. Tipos de cuantificadores de vector.

La figura 2.32 muestra los componentes más importantes en un sistema de cuantificación de vector: la librería de código y el bloque de acoplamiento de patrones. La VQ es esencialmente una técnica de acoplamiento de patrones contenidos en un bloque de píxeles. Para un vector de entrada dado, el codificador VQ encuentra el vector de palabra de código contenido en su librería mejor acoplado al vector de entrada y genera el índice del vector en la librería como salida. Para efectuar esta operación, debe establecerse una regla de acoplamiento de patrones. Para diseñar un cuantificador de vector, es necesario considerar los siguientes tres aspectos:

- a) Escoger una regla de acoplamiento de patrones.
- b) Diseñar una librería de código, y posiblemente.
- c) Reducir la complejidad del proceso de acoplamiento de patrones.

De la teoría de la información de Shannon, se establece que dada una dimensión del bloque analizado lo suficientemente grande, el desempeño del VQ puede ser arbitrariamente cercano al límite teórico, o entropía, para una tasa de codificación dada. Sin embargo, no es práctico tener un VQ de gran dimensión ya que la complejidad del VQ crece exponencialmente con respecto a la dimensión del vector. El tercer aspecto importante para diseñar un cuantificador de vector es el cómo reducir la complejidad. Se han definido varias aproximaciones al respecto, y de entre ellas, las más comunes son:

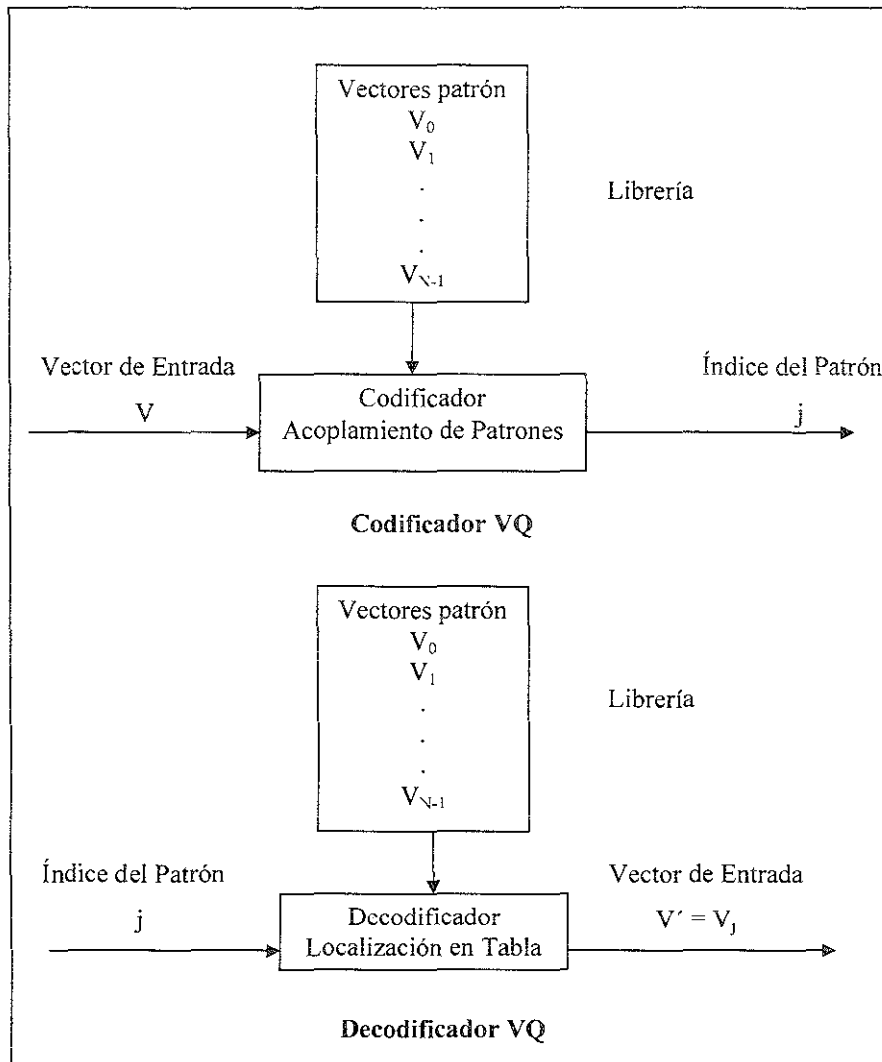


Figura 2.32. Esquema del codificador y del decodificador VQ (KOU, Weidong, Digital image compression. Algorithms and standards).

- VQ con estructura de árbol.
- VQ clasificados.
- VQ de etapas múltiples.
- VQ de código producto.
- VQ de transformada.
- VQ Jerárquico.
- VQ predictivo.
- VQ de estado finito.
- VQ con algoritmo de búsqueda rápida.

La idea básica de estas aproximaciones es reducir la complejidad y lograr un desempeño subóptimo diseñando un VQ con algún orden de estructura o utilizando algunas características de la fuente en el diseño. Por ejemplo, buscar un árbol con N vectores hoja

es más rápido que buscar un conjunto de N vectores ya que el primero requiere $\log N$ operaciones de búsqueda y el segundo requiere N operaciones de búsqueda. Basándose en esta idea, se puede diseñar una librería con una estructura de árbol para poder efectuar una búsqueda rápida. Esto resulta en un VQ con estructura de árbol. Un VQ clasificado es aquel que tiene múltiples librerías diseñadas por la clasificación de los vectores de entrada en diferentes clases. Otros esquemas de librería múltiple incluyen al VQ de etapas múltiples, VQ jerárquico, VQ de estado finito, y VQ de código producto. Si bien las aproximaciones y los algoritmos resultantes son muy diferentes, el objetivo principal es el mismo: lograr una reducción en la complejidad usando más de una librería. Explotando la correlación entre vectores sucesivos se llega al VQ predictivo, y permitiendo asignaciones de bits fraccionales por cuantificación de vector de coeficientes transformados se obtiene el VQ de transformada.

2.6.3. Algoritmo de cuantificación por vector.

Para un vector de entrada dado, el mejor vector acoplado es el que resulte en la menor distorsión. Así, escoger una regla de acoplamiento de patrones requiere escoger una medida de distorsión $d(v, v')$ donde v es el vector de entrada y v' es el vector de salida. Algunas estimaciones de la distorsión utilizadas comúnmente son:

1.-Medida de distorsión de error cuadrada:

$$d(v, v') = \|v - v'\|^2 = \sum_{j=1}^k (v_j - v'_j)^2$$

2.-Medida de distorsión de error cuadrada ponderada:

$$d(v, v') = (v - v')^T W (v - v')$$

donde W es una matriz de ponderación definida y simétrica.

3.-Medida de distorsión máxima:

$$d(v, v') = \max_j |v_j - v'_j|$$

Después de escoger una medida de distorsión, lo siguiente por hacer es diseñar una librería de código. Para compresión de imagen digital, una distorsión comúnmente usada es la distorsión de error cuadrada, y un algoritmo de diseño de librería comúnmente usado es el algoritmo generalizado de Lloyd. El algoritmo es iterativo y constituye la generalización del algoritmo de diseño de cuantificador Lloyd-Max para cuantificación escalar, el cual cumple con las características revisadas en la sección 1.4.2.

Para un cuantificador de vector óptimo $Q(v)$, puede mostrarse que éste satisface las siguientes dos condiciones:

-*Condición de Vecino más Cercano*: la i -ésima región de la partición del espacio del vector de entrada deberá consistir de todos los vectores de entrada más cercanos al i -ésimo vector de salida que a cualquier otro vector de salida:

$Q(v) = v_j'$ sólo si $d(v, v_j') \leq d(v, v_i')$ para toda i .

-*Condición de Centroide*: el vector de salida óptimo para la i -ésima región es el centroide de esa región:

$$v_j' = \text{centroide}(R_j) = \left(1 / \|R_j\| \right) \sum_{n=1}^{\|R_j\|} u_n$$

para una partición dada $\{R_j; j=1, \dots, N\}$ del espacio k -dimensional R^k , donde $R_j = \{u_n^{(j)}; n=1, 2, \dots, \|R_j\|\}$ es el número de elementos en R_j .

De estas condiciones, para un conjunto grande dado S_M de vectores muestra de tamaño $M > N$, puede derivarse un algoritmo Lloyd generalizado para diseñar un cuantificador de vector de tamaño N :

Paso 1. Establecer una librería inicial $C_m = \{u_n, n=0, 1, \dots, N-1\}$, $m=0$ y el límite de distorsión final $e_{\text{límite}}$.

Paso 2. Para la librería dada C_m , particionar el S_M en N subconjuntos usando la condición de vecino más cercano.

$$R_n = \{u \in S_M : d(u, v_n') \leq d(u, v_j'); \text{ para toda } j \neq n\}$$

Paso 3. Usando la condición centroide, computar los centroides para cada uno de los N subconjuntos, y establecer estos N centroides como la nueva librería,

$$C_{m+1} = \{\text{centroide}(R_n); n=0, 1, \dots, N-1\}$$

Paso 4. Calcular la distorsión promedio e_{m+1} para C_{m+1} . Si $e_{m+1} \leq e_{\text{límite}}$, entonces salir, de otra forma, establecer $m+1 \rightarrow m$ e ir al paso 2.

Para un conjunto finito de vectores muestra, una secuencia de cuantificadores de vector es producida por el algoritmo generalizado de Lloyd. Las distorsiones promedio de esta secuencia de cuantificadores de vector convergen en un número finito de iteraciones. Así, se puede obtener siempre un cuantificador de vector con el nivel de cuantificación deseado mediante el algoritmo generalizado de Lloyd.

Librerías de código para el VQ.

En principio, el diseño de un cuantificador de vector requiere una librería de código para definición de patrones comunes, usada tanto para el codificador como para el decodificador. La elección de la librería de código depende de las propiedades estadísticas de los datos fuente a cuantificar, y por lo tanto juega un papel esencial en el desempeño del VQ. Se han desarrollado muchas estrategias para diseñar librerías de código apropiadas de forma eficiente que requieran una carga computacional mínima. La aproximación mejor conocida para el diseño de una librería de código es el Algoritmo Generalizado de Lloyd, también conocido como el Algoritmo LBG. Otras variantes de Cuantificación de Vector pueden actualizar la librería inicial conforme el proceso de cuantificación está en marcha. Dichos cuantificadores son conocidos como Cuantificadores de Vector Adaptativos. Otra clase de cuantificadores de vector utiliza librerías de código no explícitas para evitar los requerimientos de almacenamiento. La Cuantificación de Vector Lattice (LVQ) es una de dichas aproximaciones.

Formación de los vectores.

La formación de vectores es otro punto importante en el diseño del VQ. En las primeras versiones de los cuantificadores de vector, los vectores se formaban por valores de pixel de una imagen en bloques cuadrados. En las técnicas más recientes, los vectores se crean a partir de los coeficientes de una imagen transformada. Los coeficientes pueden formarse ya sea a partir de aquellos de una sub-banda dada alrededor de la misma posición espacial o a partir de coeficientes de diferentes sub-bandas en la misma posición espacial. Si bien desde el punto de vista teórico, es posible mostrar que el VQ es superior a la cuantificación escalar, desde el punto de vista de la complejidad, la sobrecarga adicional debida a la implementación del VQ no parece favorecer dichas aproximaciones para la mayoría de las aplicaciones. La figura 2.33 muestra el resultado de la aplicación de la cuantificación por vector sobre una imagen original.



Figura 2.33. Imagen de Lena original e imagen reconstruida a partir de la cuantificación por vector (CHELLAPPA, Rama. Digital image processing).

2.7. RESUMEN DE LA COMPRESIÓN DE FORMA DE ONDA.

Las técnicas comprendidas en esta clasificación son utilizadas por la mayoría de los estándares de compresión y formatos gráficos comerciales en la actualidad debido a la baja complejidad de su procesamiento y a que logran una representación eficiente de la información contenida en una imagen digital. Si bien el medio de análisis utilizado por estas técnicas no toma en cuenta las entidades visuales que se encuentran en la imagen a comprimir y se limita a verificar su importancia estadística, estos procedimientos, sobre todo los de compresión por transformadas, gozan de gran aceptación entre las empresas y organizaciones relacionadas con las imágenes digitales, debido principalmente a que promueven la compatibilidad entre las tecnologías de compresión que las utilizan.

En la tabla 2.6 se muestra un cuadro comparativo de las técnicas de compresión revisadas en este capítulo y cuyos criterios de comparación se utilizarán igualmente para comparar las técnicas expuestas en los capítulos siguientes. Para este caso, el criterio de la herramienta de selección se refiere al método utilizado para que la técnica seleccione la información que requiere; el área de selección inicial se refiere al elemento o conjunto de elementos de la imagen a partir del cual la técnica comenzará a seleccionar la información; si una técnica es iterativa con respecto al área de selección implica que la herramienta de selección, original o modificada, será utilizada nuevamente sobre el área de selección, inicial o posterior; la redefinición del tamaño para el área de selección implica que el área de selección definida inicialmente sufrirá una reducción en sus dimensiones, como en el caso de la aplicación de la transformada wavelet sobre la imagen de paso bajo de la transformación inmediata anterior; la redefinición de los parámetros para la herramienta en cada iteración implica que con cada nueva iteración la herramienta sufrirá un cambio en sus parámetros para ajustarse a una nueva área de selección, como en el caso de los filtros QMF de la compresión sub-banda al ajustarse a las características de la imagen de paso bajo del filtrado inmediato anterior; la consideración del contenido particular del área de selección implica que la herramienta de selección de información utilizada por la técnica de compresión plantea una modificación en su comportamiento selectivo al encontrar cambios abruptos en los valores de color en la imagen, lo que implica la presencia de algún objeto definido por contornos, característica que no se aplica en el caso de las técnicas de compresión de forma de onda revisadas en este capítulo.

Técnica de compresión	Herramienta de selección de información	Área de selección inicial	¿Iterativa con respecto al área de selección?	¿Redefinición del tamaño para área de selección en cada iteración?	¿Redefinición de parámetros para la herramienta en cada iteración?	¿Consideración del contenido particular del área de selección?
Predictiva	Ecuaciones de Predicción	Pixel	No	No	No	No
Transformada	Transformadas bidimensionales	Bloque de píxeles	No	No	No	No
Sub-banda	Filtros digitales	Imagen completa	Sí	No	Sí	No
Wavelet	Transformadas bidimensionales	Imagen completa	Sí	Sí	Sí	No
Vector	Vectores	Bloque de píxeles	Sí	No	Sí	No

Tabla 2.6. Cuadro comparativo de las técnicas de compresión de forma de onda.

CAPÍTULO III:

TÉCNICAS DE SEGUNDA GENERACIÓN.

3.1. DEFINICIÓN Y CLASIFICACIÓN DE LAS TÉCNICAS DE COMPRESIÓN DE SEGUNDA GENERACIÓN.

El objetivo al que cualquier sistema de procesamiento de imagen orienta sus resultados es el ojo humano. Si bien nuestro sistema visual es por mucho el mejor sistema de procesamiento de imágenes, también está lejos de ser perfecto. De esta forma, si un esquema de codificación se acopla al sistema visual humano e intenta imitar la parte conocida de sus funciones, es posible alcanzar altas proporciones de compresión. Por lo tanto, resulta útil efectuar un preprocesamiento sobre los datos de la matriz de píxeles que conforman a la imagen antes de aplicar las técnicas de codificación de primera generación como las basadas en forma de onda o las de compresión sin pérdida de información. Por lo tanto, una forma opcional para codificar imágenes es representarlas en términos de regiones texturizadas rodeadas por contornos de manera que las regiones definidas correspondan, tanto como sea posible, a los objetos en la escena.

Los progresos en el estudio de los mecanismos de visión humana han abierto nuevas expectativas para la codificación de imagen. La sensibilidad direccional de las neuronas combinada con el procesamiento separado de contornos y texturas de la imagen digital ha creado una nueva clase de métodos de codificación capaces de alcanzar altas proporciones de compresión (70:1). Se denominan técnicas de segunda generación, al conjunto de procesos capaces de lograr altas compresiones más allá del límite entrópico establecido por el modelo de probabilidades utilizado, considerando los elementos que contiene una imagen. Pueden definirse dos grupos en esta clasificación: los métodos que utilizan operadores locales y métodos que utilizan descripciones de contorno-textura. A lo largo de este capítulo se describen siete métodos que se ubican en ambos grupos, los cuales se muestran en la figura 3.1.

Las técnicas basadas en operador local descomponen una imagen original en versiones de la misma que están en función del espectro de frecuencia de sus valores, mediante filtros como los revisados en la sección 2.4 y 2.5 pero efectuados de forma iterativa, modificando sus parámetros y ajustándose a las características locales de la nueva imagen generada con cada iteración.

Las técnicas orientadas a contorno-textura intentan segmentar la imagen en regiones texturizadas rodeadas por contornos de forma tal que éstos correspondan, tanto como sea posible, a los objetos contenidos en la imagen. Las informaciones de contorno y de textura se codifican separadamente. Los contornos pueden extraerse en dos formas: por formación de región o usando técnicas de extracción de contorno o de detección de bordes. En el primer caso, se obtienen contornos parecidos a los originales, lo que facilita la tarea de listar las regiones y sus propiedades. La imagen segmentada luce como una imagen de rompecabezas. En el segundo caso, los contornos que se obtienen no son tan parecidos.

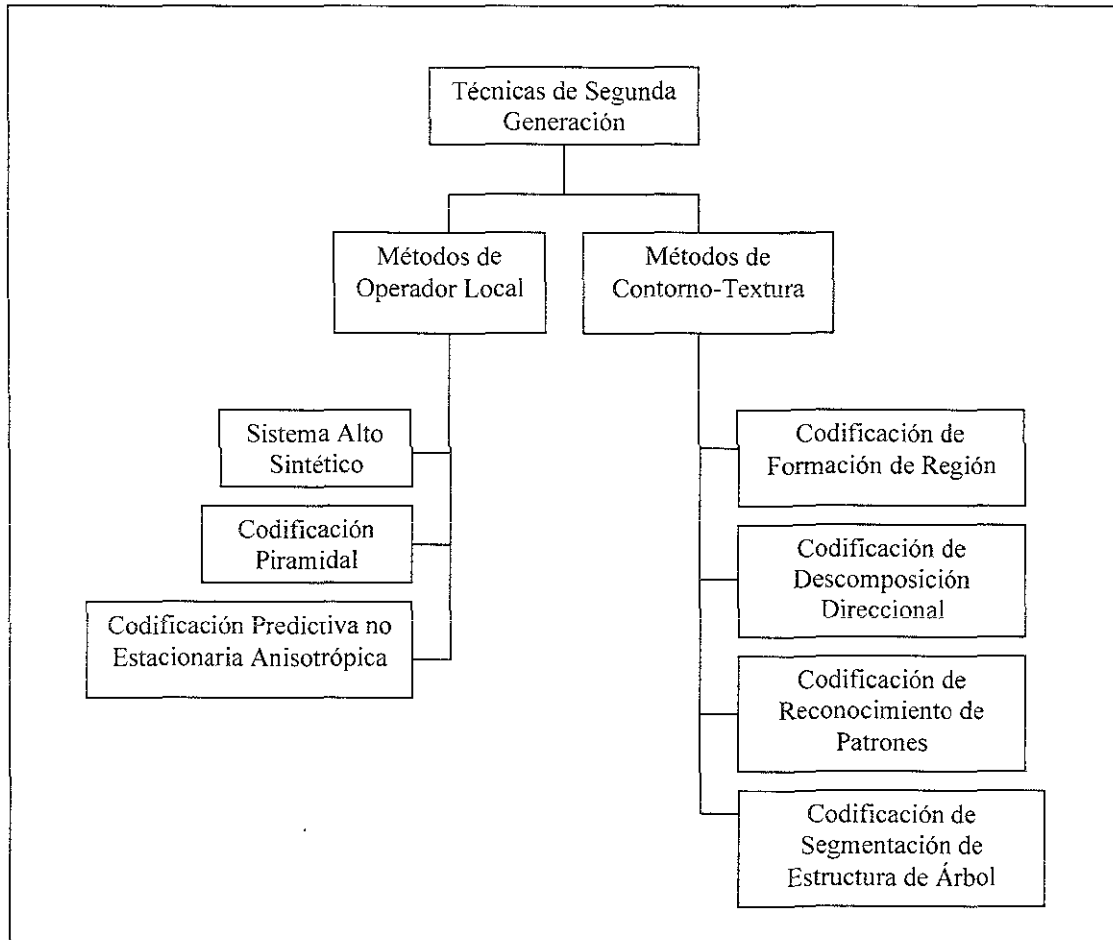


Figura 3.1. Clasificación general de la codificación de segunda generación.

3.2. COMPRESIÓN POR SISTEMA ALTO SINTÉTICO.

Esta técnica es uno de los primeros métodos que propusieron la descomposición de una imagen en sus componentes de alta y baja frecuencia y se conoce desde hace más de

veinte años. En los primeros años de la codificación de imagen se consideró un método complejo, pero hoy en día es fácil de implementar con la tecnología actual.

3.2.1. Descripción de la técnica.

La secuencia de mensajes para el sistema alto sintético se selecciona al dividir la imagen original en dos partes: la imagen de paso bajo que contiene la luminosidad general sin contornos contundentes, y la imagen de paso alto que contiene la información de los bordes fuertes, de forma semejante a la codificación sub-banda de la sección 2.4. La imagen de paso bajo puede representarse con muy pocas muestras. Posteriormente se realiza una operación que detecte los límites de los bordes en la imagen de paso alto para determinar si un punto de borde es importante. Entonces, al descartar los puntos de borde que no proporcionen mucha información a la imagen, el método se convierte en un proceso con pérdida de información. Finalmente, se codifica la ubicación y la magnitud de cada punto de borde. Estas variables son los mensajes que conforman la imagen de paso alto.

Para la decodificación de la imagen, se utiliza un filtro de reconstrucción bidimensional, cuyas propiedades son determinadas únicamente por el filtro usado al generar la imagen de paso bajo, y así poder sintetizar la parte de alta frecuencia a partir de la información de los bordes. Entonces, esta señal “alta sintética” se suma a la imagen de paso bajo para dar la salida final. Un diagrama del procedimiento de este sistema se muestra en la figura 3.2.

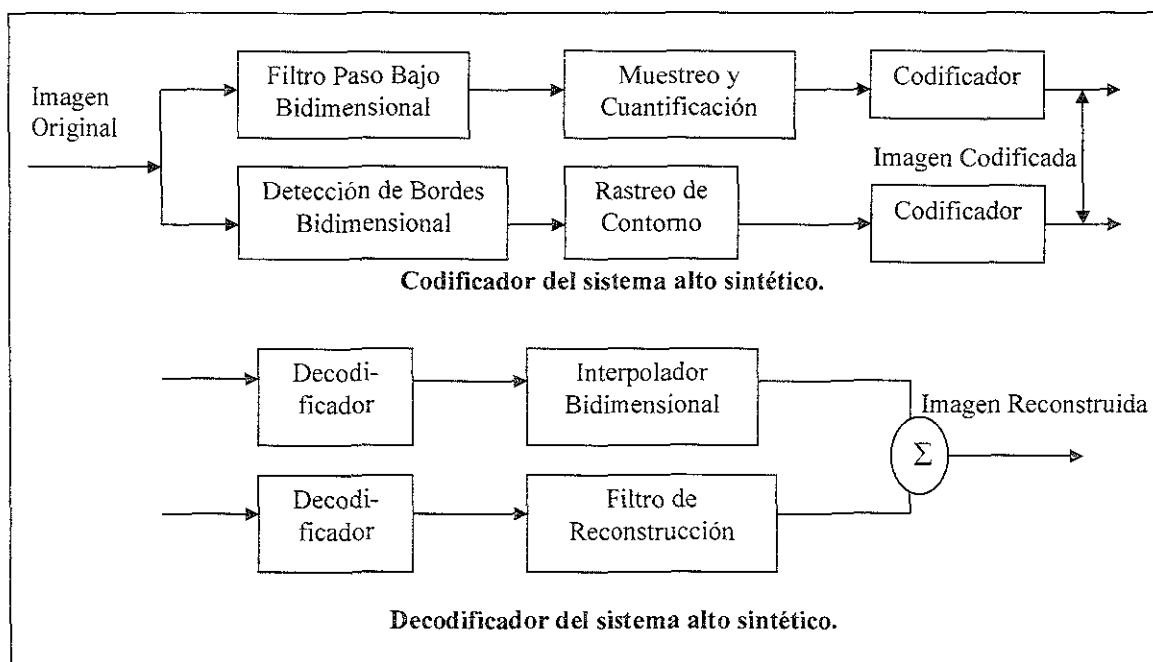


Figura 3.2. Esquema del codificador y del decodificador para el sistema alto sintético (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

El sistema alto sintético explota las propiedades del sistema visual humano desde sus primeros niveles de procesamiento. Por lo tanto, permite una reducción considerable en la cantidad de redundancia. Los valores de los límites para determinar la información a descartar utilizados en este sistema son de principal importancia para la proporción de compresión y para la apariencia de las imágenes reconstruidas. Es posible obtener una proporción de compresión alta, pero a cambio se pierde una gran cantidad de información de textura y detalles. Debido a la ausencia de métodos teóricos para evaluar la pérdida de textura en una imagen, su apariencia sólo puede evaluarse con un procedimiento de corte y prueba. El método basado en descomposición direccional, expuesto en la sección 3.6 puede verse como un refinamiento del sistema alto sintético, donde la habilidad para extraer y codificar bordes se mejora con el uso de filtros direccionales. La figura 3.3 muestra la imagen recuperada a partir de la codificación con el sistema alto sintético de una imagen original.

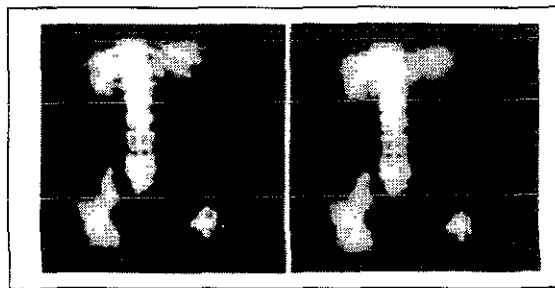


Figura 3.3. Imagen de rayos X original y su versión reconstruida a partir del sistema alto sintético con una proporción de compresión de 8.37:1 (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

3.3. COMPRESIÓN PIRAMIDAL.

Debido a que combina características de los métodos de codificación predictiva y de transformada y debido a que las compresiones obtenidas no son muy altas (10:1), esta técnica de codificación podría considerarse como un método híbrido de la primera generación. Sin embargo, se incluye en los de segunda generación debido a que la estructura jerárquica de algunas de sus funciones son similares a las del sistema visual humano. Esta técnica también se le conoce como codificación progresiva o codificación jerárquica.

3.3.1. Descripción de la técnica.

Comenzando a partir de la imagen original $x(k, \zeta)$ a codificar, se procesa mediante un filtro dado, una versión de paso bajo $x_1(k, \zeta)$ de la misma usando los promedios locales de los

valores de las muestras con una correlación alta. La imagen de paso bajo obtenida, limitada por una frecuencia de corte f_1 , puede verse como una predicción de $x(k, \zeta)$. Entonces, el error de predicción se define como:

$$e_1(k, \zeta) = x(k, \zeta) - x_1(k, \zeta) \quad (1)$$

Codificar $e_1(k, \zeta)$ y $x_1(k, \zeta)$ es equivalente a codificar directamente la imagen original. La compresión esperada depende de dos factores: $e_1(k, \zeta)$ por su naturaleza, es una imagen de paso alto y debido a la baja sensibilidad del ojo humano a estas frecuencias, pueden utilizarse menos bits por muestra para representarla en comparación con los utilizados para la imagen original. Además $x_1(k, \zeta)$ es una imagen de paso bajo, por lo que puede representarse con menos muestras que las necesarias para representar la imagen original.

A partir de lo anterior, además de las interpretaciones particulares y las funciones específicas usadas, este método es conceptualmente idéntico al sistema alto sintético. La diferencia consiste en que las operaciones básicas descritas se realizan de forma iterativa. Si $x_1(k, \zeta)$ se filtra nuevamente con paso bajo, con una frecuencia de corte f_2 , el resultado $x_2(k, \zeta)$, se utiliza como la predicción de $x_1(k, \zeta)$. El error de esta predicción es entonces:

$$e_2(k, \zeta) = x_1(k, \zeta) - x_2(k, \zeta) \quad (2)$$

Nótese que esta segunda imagen de error es más pequeña que $e_1(k, \zeta)$ por un factor, el cual es la razón de las dos frecuencias de corte f_1/f_2 . Después de n iteraciones, se obtiene una serie de imágenes de error de predicción $e_1(k, \zeta)$, $e_2(k, \zeta)$, ..., $e_n(k, \zeta)$. En cada iteración, las dimensiones de estas imágenes se reducen por un factor f_i/f_{i+1} . Para una implementación simple, puede usarse un factor de 2 en cada iteración. Si se ve a estas imágenes como una apilada sobre la otra, el resultado es una estructura de datos piramidal. Este tipo de funciones son utilizadas ampliamente en el procesamiento digital de imágenes para detectar cambios en la intensidad de los valores que la componen.

3.3.2. Codificación de los datos.

Estas imágenes de error son cuantificadas y codificadas. Para reconstruir la imagen decodificada, se diseñan y utilizan filtros de interpolación para compensar el submuestreo realizado en cada nivel. La imagen decodificada se obtiene a partir de la suma de las imágenes de error decodificadas e interpoladas. La figura 3.4 muestra el diagrama de este procedimiento.

Una característica de este sistema es que la apariencia de la imagen decodificada puede mejorarse a conveniencia si se reduce la proporción de compresión esperada. La figura 3.5 muestra el diagrama general de la compresión piramidal.

A partir de esta sección se utilizarán tres imágenes originales, las cuales se muestran en la figura 3.6, para señalar el efecto comparativo de las técnicas de compresión a exponer en este capítulo.

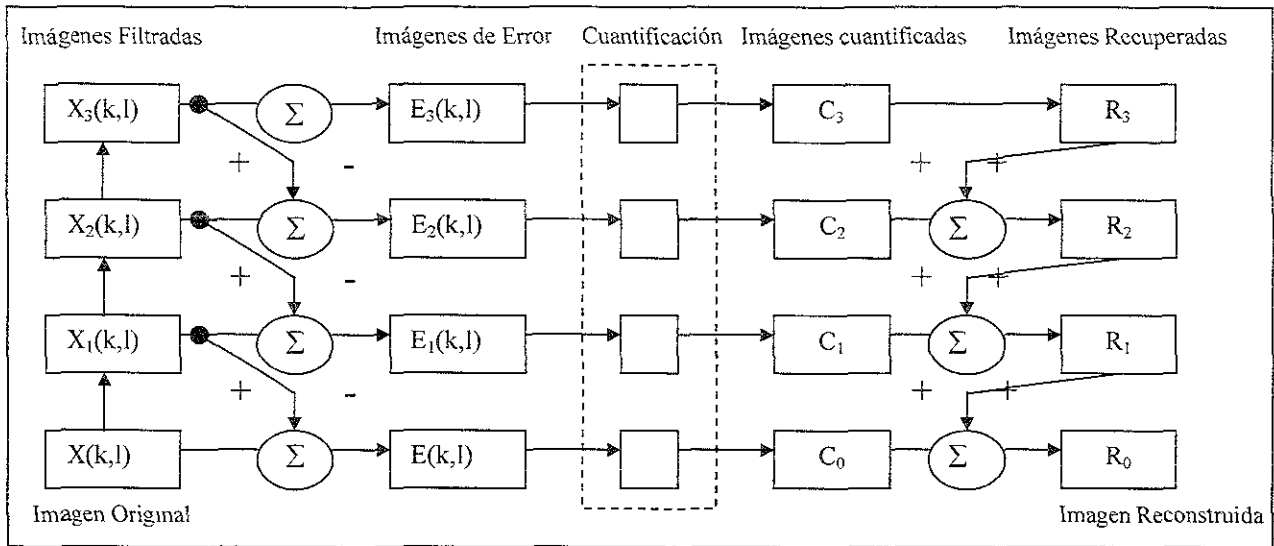


Figura 3.4. Diagrama del sistema de codificación piramidal (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

En el caso de la codificación piramidal, la figura 3.7 muestra el ejemplo de la pirámide generada a partir de una imagen filtrada con paso bajo por medio de funciones de ponderación de correlación separables de 5×5 . Las frecuencias de corte para el submuestreo tienen un valor de 2 en cada dirección, por lo que cada imagen tiene cuatro veces menos píxeles que su antecesora. La figura 3.8 muestra la imagen recuperada a partir de la codificación piramidal con proporción de compresión de 6:1.

3.4. COMPRESIÓN PREDICTIVA NO ESTACIONARIA ANISOTRÓPICA.

Este método pone especial énfasis en la selección de las muestras de la imagen antes de ser codificadas, una característica importante en los métodos de segunda generación.

3.4.1. Descripción de la técnica.

Los principios de la codificación predictiva no estacionaria anisotrópica pueden verse como el subconjunto de un esquema para restauración de imagen. Se comienza con el problema de reconstruir una imagen estimada $x'(k, \zeta)$ a partir de una imagen original $x(k, \zeta)$, distorsionada por un ruido que se interpreta como las señales de alta frecuencia en la imagen. A partir de esta circunstancia se deriva la utilización de filtros para la separación de ese ruido de la señal de baja frecuencia, que para esta técnica de codificación se trata de los denominados filtros Wiener.

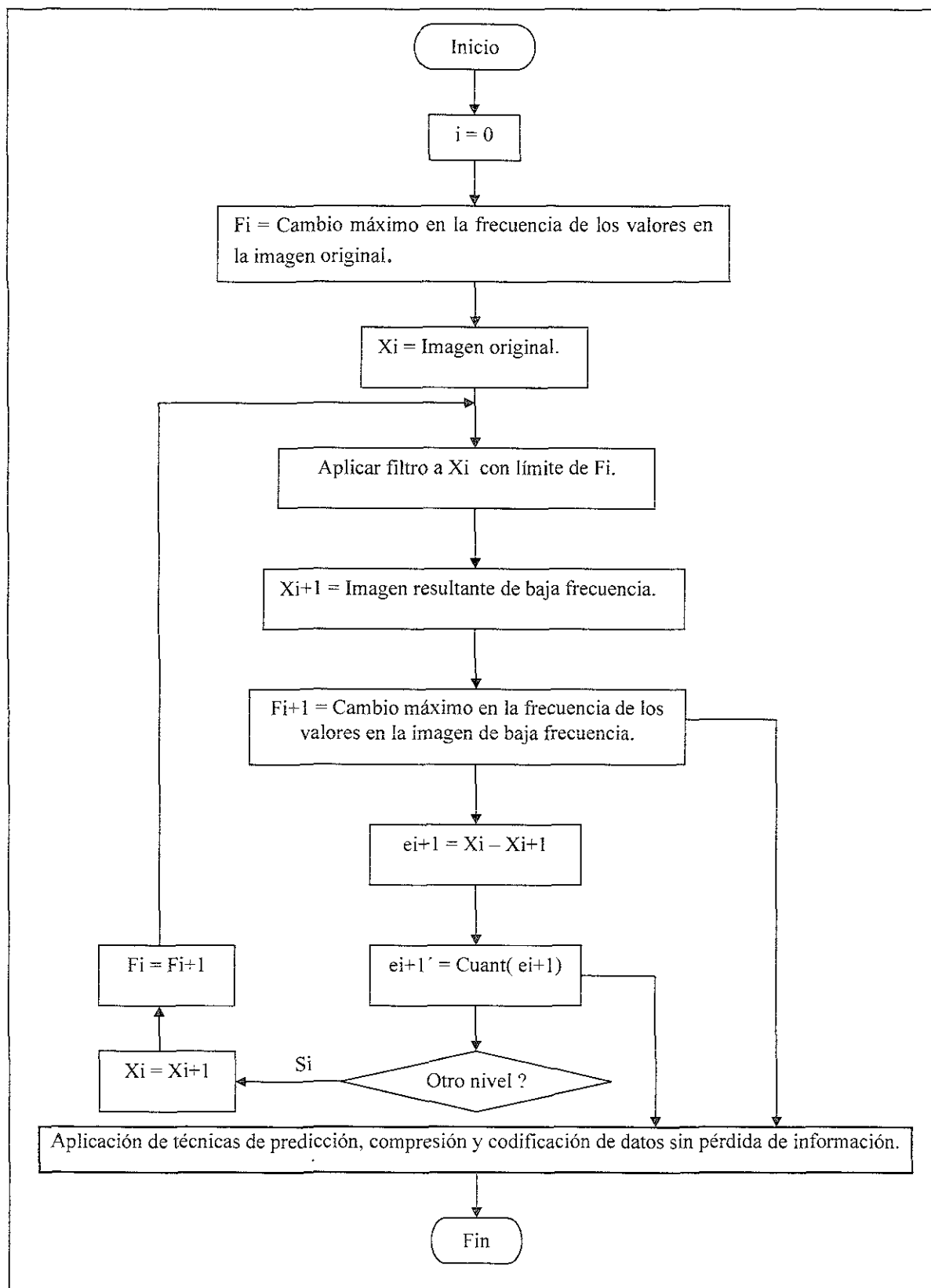


Figura 3.5. Diagrama de flujo de la compresión piramidal.



Figura 3.6. Imágenes originales para comparar algunas técnicas de compresión de segunda generación (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").



Figura 3.7. Cuatro niveles de la pirámide generada por la codificación piramidal.



Figura 3.8. Imágenes resultantes del proceso de codificación piramidal con compresión de 6:1.

(IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

Filtros Wiener.

El filtro Wiener tiene tres componentes. El primer componente es el filtro Wiener isotrópico clásico expresado como:

$$H_1(\rho) = \frac{\phi_x(\rho)}{\phi_x(\rho) + \sigma_n^2} \quad (3)$$

donde ρ es la frecuencia radial en el dominio de Fourier, $\phi_x(\rho)$ el espectro de magnitud de la imagen original, y σ_n^2 la magnitud del ruido. Este filtro tiene una característica de paso bajo ya que la imagen tiene más energía de baja frecuencia que ruido.

El segundo componente también es un filtro isotrópico dado por:

$$H_2(\rho) = \frac{\sigma_n^2}{\phi_x(\rho) + \sigma_n^2} \quad (4)$$

En contraste con el primer componente, este tiene una característica de paso alto.

El tercer componente es un filtro anisotrópico dado por:

$$H_3[\rho, \theta, \varphi(k, \zeta)] = \frac{\cos^2[\varphi(k, \zeta) - \theta] \sigma_n^2}{\phi_x(\rho) + \sigma_n^2} \quad (5)$$

donde θ es la frecuencia angular polar en el dominio de Fourier. Este filtro se usa para ponderar el ruido local de acuerdo con su dirección con respecto al de una anisotropía local indicada por $\varphi(k, \zeta)$. Los componentes de ruido local alineados con $\varphi(k, \zeta)$ se atenúan para una mejor restauración.

El filtro de restauración final es la suma ponderada de estos tres componentes:

$$H(\rho, \theta, k, \zeta) = H_1(\rho) + \beta(k, \zeta) \tau(k, \zeta) H_2(\rho) + \beta(k, \zeta)[1 - \tau(k, \zeta)] H_3[(\rho, \theta, \varphi(k, \zeta))] \quad (6)$$

La función de ponderación $\beta(k, \zeta)$ está relacionada con la magnitud de las características rectilíneas locales (bordes y líneas) en la vecindad del punto (k, ζ) , mientras que $\tau(k, \zeta)$ representa la variabilidad en el ángulo del mismo punto. La ponderación $\beta(k, \zeta)$ puede verse como una estimación dirección-intensidad de una no estacionaridad local. Así, junto con la información direccional $\varphi(k, \zeta)$, se forma una imagen vectorial la cual indica la magnitud y la dirección de las características rectilíneas de forma local en la imagen. Todas

las funciones desconocidas en (6) como $\beta(k, \zeta)$, $\tau(k, \zeta)$, $\varphi(k, \zeta)$ y las salidas de los filtros dependen directamente de los datos de entrada. Para reducir la carga computacional de estas estimaciones, se han sugerido filtros limitados espacialmente (operadores locales) en la forma de arreglos de al menos 15 elementos. La restauración puede efectuarse iterativamente por medio de varias pasadas de los filtros de reconstrucción a través de los datos.

3.4.2. Codificación de los datos.

La forma de utilizar el filtro de restauración de imagen descrito para el campo de la codificación de imagen es transformarlo en un filtro de predicción y usar el error de predicción como los mensajes a codificar. En este esquema, se aplica primero la transformada de coseno discreta (DCT) al error de predicción antes de cuantificarlo y codificarlo. Las funciones de ponderación $\beta(k, \zeta)$, $\tau(k, \zeta)$, $\varphi(k, \zeta)$, estimadas a partir de los datos de la imagen original, también deben codificarse, transmitirse y/o almacenarse.

En general, el filtro de restauración no necesita ser recursivo. Sin embargo, el filtro de predicción que se utiliza para esta técnica de compresión si debe ser recursivo. Esto requiere del truncamiento de las salidas generadas en (6) para obtener una ecuación diferencial bidimensional recursiva para implementar la predicción. Cuando se procesan la predicción y su error, se aplica una transformada DCT unidimensional a cada línea de la imagen de error de predicción. Los coeficientes transformados son cuantificados en un proceso de dos niveles: codificación adaptativa de la parte estacionaria y una codificación sólo de los valores de baja frecuencia de la parte no estacionaria.

Las funciones de ponderación que constituyen los filtros de frecuencia se codifican igualmente. Estas funciones tiene un bajo ancho de banda, por lo que es posible codificarlas con un muestreo descendente de 1:6 en ambas direcciones. Las magnitudes, ángulos y variaciones en los ángulos se codifican con 2, 4 y 1 bit respectivamente. Esto lleva a una compresión muy alta, mayor a 70:1, para representar las funciones de ponderación. Tomando en cuenta el número de bits usados para codificar el error de predicción, se obtienen compresiones globales alrededor de 35:1 con imágenes decodificadas de apariencia aceptable. La figura 3.9 muestra los resultados de la compresión predictiva no estacionaria anisotrópica aplicada sobre las imágenes originales de la figura 3.6, y la figura 3.10 muestra el diagrama general de esta técnica.



Figura 3.9. Imágenes recuperadas a partir de la codificación predictiva no estacionaria anisotrópica a proporción de compresión de 20:1 (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

3.5. COMPRESIÓN BASADA EN FORMACIÓN DE REGIÓN.

En la primera etapa de este método, la imagen se segmenta para clasificar sus píxeles en píxeles de contorno y píxeles de textura. Este procedimiento particiona la imagen en un conjunto de regiones adyacentes bajo la restricción de que la variación en el nivel de gris en la región no contiene ninguna discontinuidad contundente, es decir, contornos. La segmentación se efectúa en tres pasos: preprocesamiento, formación de región y eliminación de los artificios.

3.5.1. Descripción de la técnica.

Preprocesamiento.

El preprocesamiento intenta reducir la granularidad local de la imagen original sin afectar sus contornos, para que no se obtengan regiones demasiado pequeñas después de la formación de región. Además, estas pequeñas regiones no corresponden en general, a objetos reales en la imagen original y se convierten en contornos falsos. El problema clave en el preprocesamiento es la reconciliación de dos objetivos aparentemente contradictorios; la remoción de granularidad y la preservación de bordes. La mayoría de los filtros de remoción de granularidad tienen características de paso bajo y por lo tanto

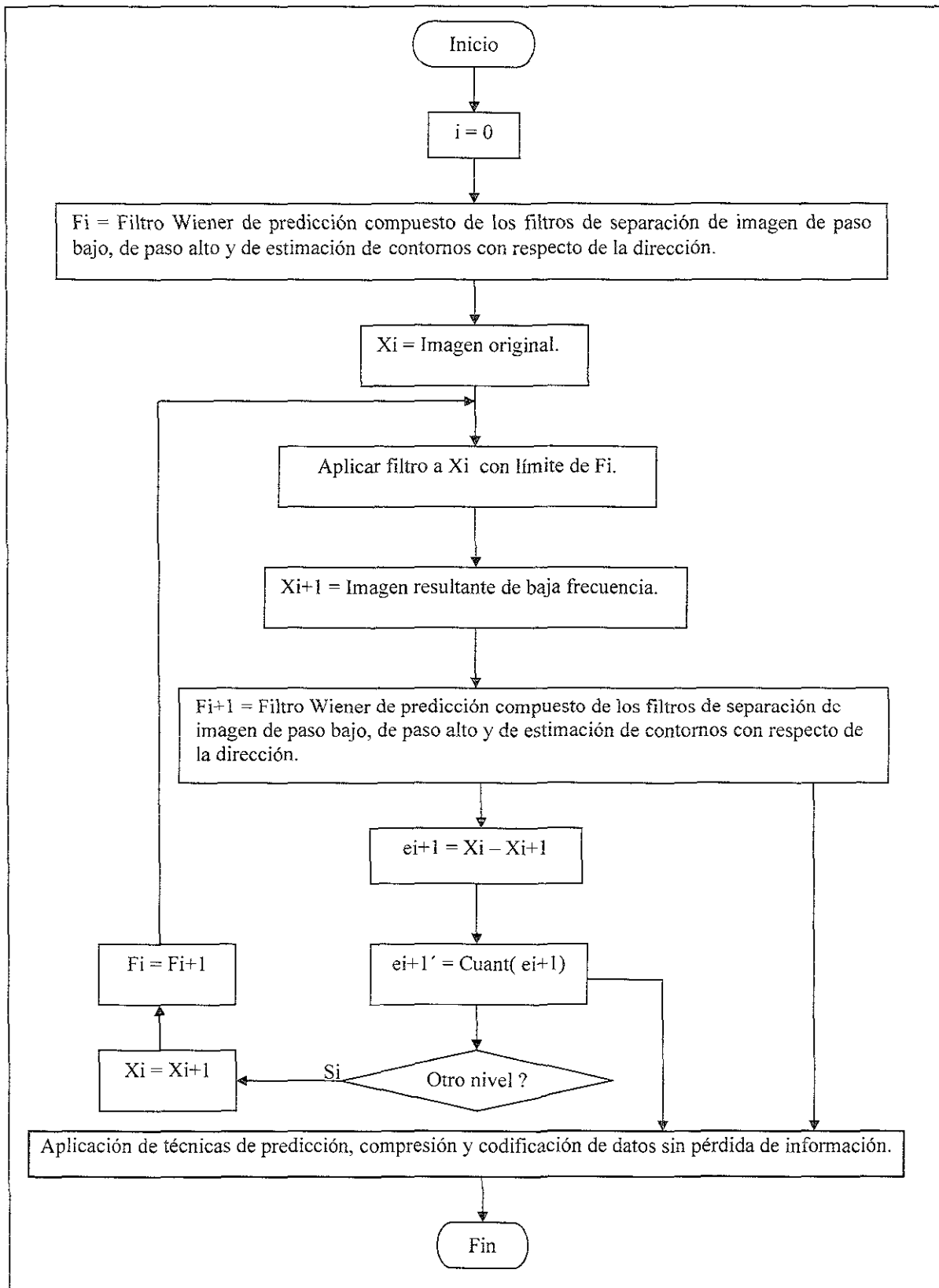


Figura 3.10. Diagrama de flujo de la compresión predictiva no estacionaria anisotrópica.

su operación provoca que se filtren igualmente bordes suaves. Para esta operación se escoge un filtro denominado filtro gradiente inverso debido a su habilidad para adaptar sus coeficientes de acuerdo al contraste local, por lo que se comporta como un filtro de paso bajo en áreas altamente contrastantes. Como sus características de frecuencia no son perfectamente de paso bajo o paso completo, no es muy eficiente en remover la granularidad, al menos en una pasada. Esta es la causa por lo que su aplicación debe ser iterativa hasta que la granularidad se haya removido lo suficiente. En la figura 3.11 se muestra el resultado de aplicar este filtro a una señal unidimensional.

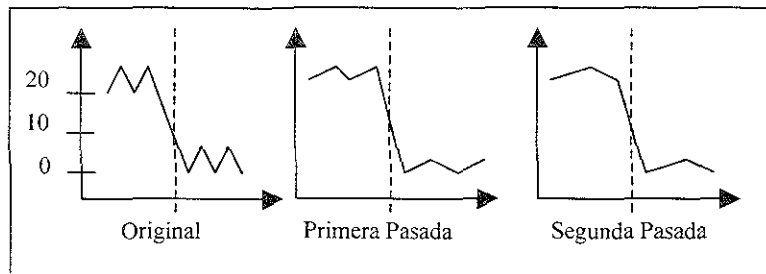


Figura 3.11. Resultados del filtrado gradiente inverso sobre una señal unidimensional en dos iteraciones

(IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

Formación de región.

El mecanismo de formación de región es el siguiente. Las regiones a extraer deben caracterizarse por alguna propiedad en el paso de preprocesamiento. La propiedad puede ser, por ejemplo, el nivel de gris de un pixel, la variación en el nivel de gris o la energía dentro de una banda de frecuencia dada. La selección de esta propiedad juega un papel muy importante en la complejidad del método y en la exactitud de los contornos obtenidos después de la segmentación. Posteriormente, comenzando con un pixel dado en la imagen, se examinan sus pixeles vecinos para ver si comparten la misma propiedad. Si este es el caso, ese pixel se incluye en la región, y de nuevo, sus pixeles vecinos son examinados y así sucesivamente. Cuando ya no quedan más pixeles conectados a la región y compartiendo la misma propiedad, el procedimiento se detiene y reinicia en cualquier otro pixel que no se incluya en la primera región. La segmentación se completa cuando todos los pixeles de la imagen están asignados a una región. Como ejemplo puede utilizarse la propiedad de un intervalo fijo de nivel de gris. Aunque posee una cantidad de niveles constante, este intervalo se hace adaptativo moviéndolo hacia arriba y hacia abajo en la escala de niveles de gris para interceptar el máximo número de pixeles. Sin embargo, este desplazamiento está restringido a que los pixeles incluidos previamente en alguna región siempre permanezcan en ella. Después de este proceso de formación de región, se obtienen dos tipos de artificios: contornos que no separan completamente dos regiones y contornos que son de dos pixeles de ancho. La figura 3.12 muestra un esquema de estos artificios.

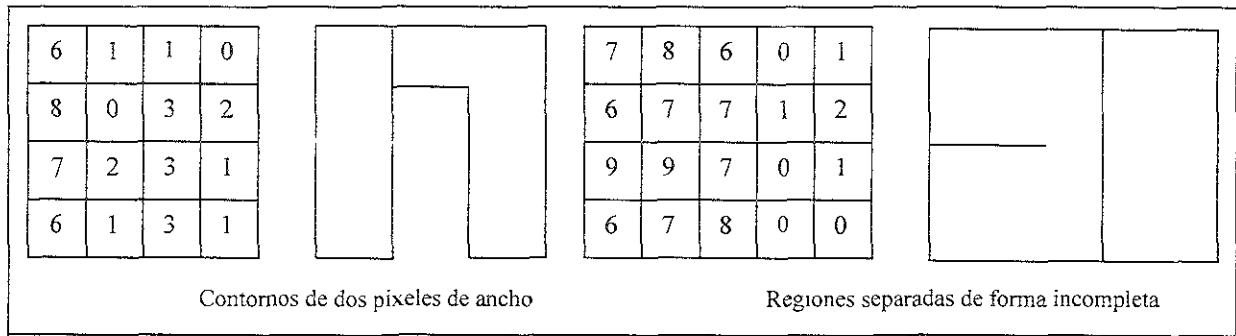


Figura 3.12. Artificios resultantes del proceso de la formación de región (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

Para el caso que ilustra la figura 3.12, la intención es definir regiones cuadradas y limitadas. En el primer cuadro, el contraste de valores 3 impide tomar en cuenta los valores a su lado y formar un cuadro mayor. En el segundo cuadro, no se incluyen los pixeles de la región inferior en la superior, pero como comparten valores el algoritmo no establece un contorno entre ellas. Estos artificios se eliminan fácilmente con un procedimiento apropiado. Dos imágenes originales y los resultados de la formación de región se muestran en la figura 3.13 con 1001 y 1111 regiones respectivamente.

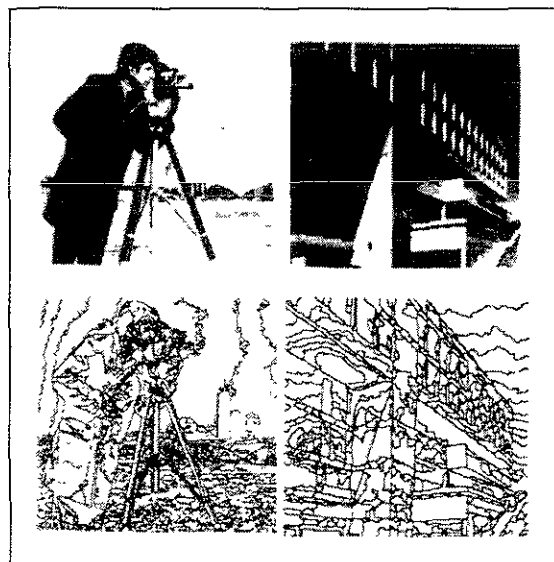


Figura 3.13. Dos imágenes originales y el resultado de la formación de región con un intervalo de 10 niveles de gris (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

A este nivel, la imagen puede verse como un rompecabezas con contornos adyacentes de un punto de ancho. Desafortunadamente, debido a la propiedad tan simple que fue utilizada (los niveles de gris), la cantidad de contornos es mucho mayor que los objetos en la imagen original. Dos procedimientos para solucionar este problema son: La introducción de

algunas distorsiones eliminando las regiones insignificantes y sus contornos, o utilizar una propiedad más refinada. La primera alternativa, se describe a continuación.

Eliminación de artificios.

Existen dos soluciones para reducir el número de regiones obtenidas por formación de región a partir de la introducción de alguna distorsión en los valores de la imagen: la eliminación de las regiones pequeñas y la fusión de las regiones adyacentes débilmente contrastantes. En la figura 3.14 se muestra el histograma del número de regiones del edificio de la figura 3.13 como una función de su tamaño expresado en términos del número de puntos internos resultantes.

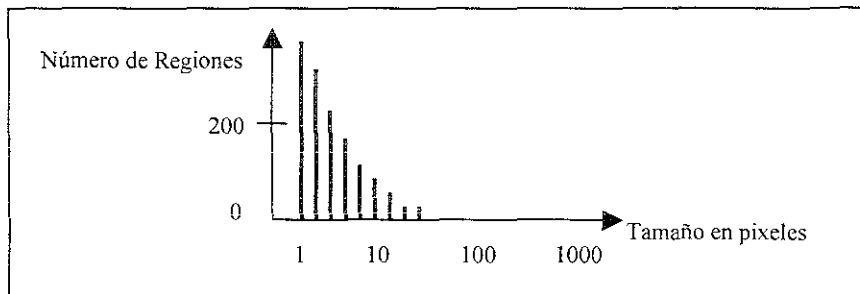


Figura 3.14. Histograma de las regiones obtenidas por formación de región (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

Esta gráfica indica que aproximadamente el 70 por ciento de las regiones tienen menos de 15 píxeles. Existen dos razones para este resultado. Primero, la granularidad no se elimina completamente en el preprocesamiento; y segundo, áreas pequeñas de alta variación de luminancia en la imagen original se segmentan en otras tantas regiones más pequeñas debido a la propiedad utilizada. Si se asume que las regiones que contengan un número de píxeles menor a un límite no sean significativas, su eliminación reduce drásticamente el número de regiones resultantes. Para evitar la creación de huecos en la imagen, estas regiones se incluyen en alguna de sus regiones adyacentes. Para minimizar la distorsión resultante, la región receptora se escoge como la región adyacente cuyo nivel de gris promedio sea el más parecido al de la región pequeña a incluirse. Al observar las áreas de luminancia constante en la figura del edificio (el cielo por ejemplo), puede notarse que están subdivididas en regiones aún cuando no existe un contorno real. Esto se debe a la propiedad usada en la formación de región, la cual divide la imagen en regiones de rango dinámico para un nivel de gris fijo.

La segunda posibilidad para reducir el número de regiones es fusionar las regiones adyacentes cuyo contraste esté por debajo de un cierto nivel. El contraste entre regiones adyacentes está definido como la diferencia de nivel de gris promediocalculada a lo largo de su borde común. Nótese que este procedimiento no introduce ninguna distorsión. El

siguiente es el histograma del número de regiones adyacentes en la imagen del edificio de la figura 3.15 como una función de su contraste común.

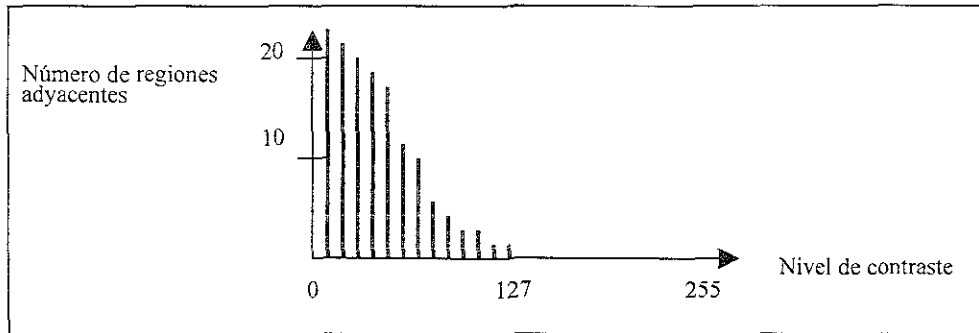


Figura 3.15. Histograma de los niveles de contraste entre regiones adyacentes (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

Esta gráfica indica que aproximadamente el 35 por ciento de los bordes de región en la imagen tienen un contraste local de menos de 15 niveles de gris. Por lo tanto puede asumirse que las regiones adyacentes cuyo contraste local sea menor que cierto límite se fusionen sin introducir discontinuidades, reduciendo así el número de regiones. Los esquemas de estos procesos de eliminación se muestran en la figura 3.16.

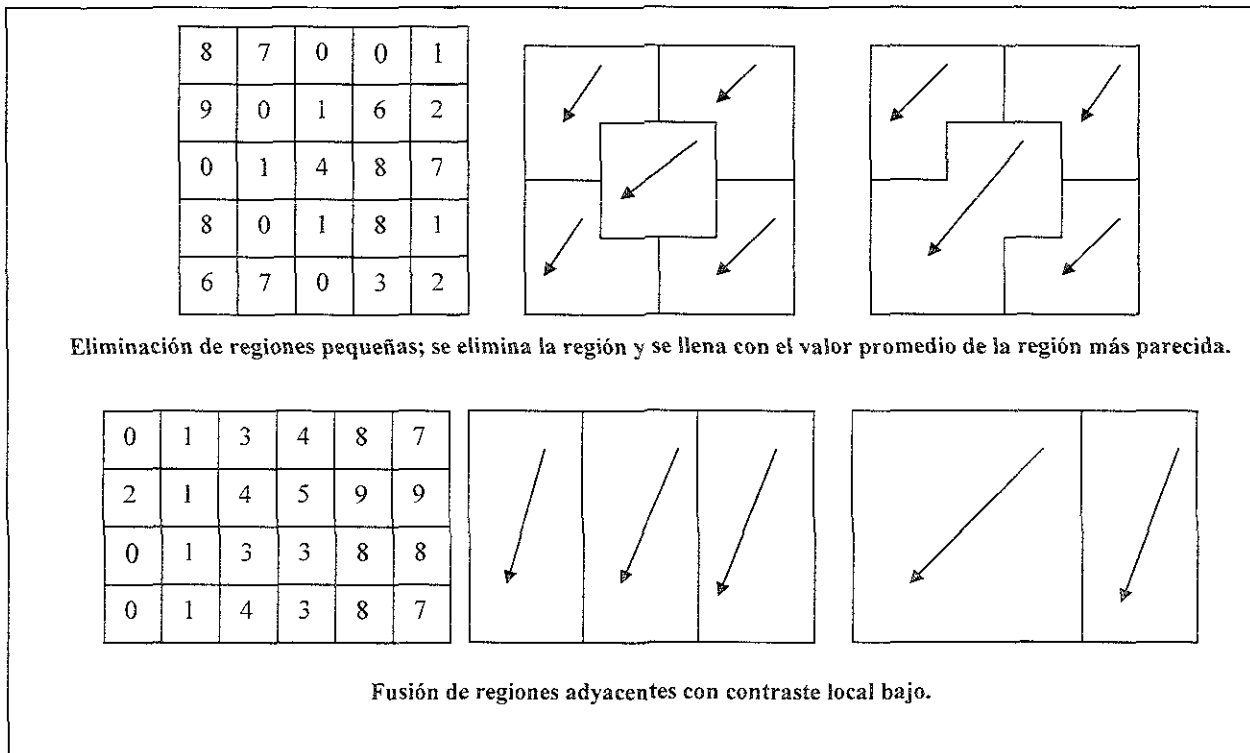


Figura 3.16. Esquemas de los procedimientos de eliminación de regiones.

El resultado de aplicar este procedimiento a las imágenes segmentadas se muestra en la figura 3.17. El número de regiones ahora es de 195 y 164 respectivamente.

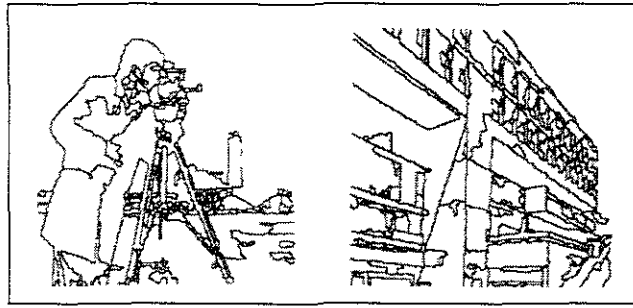


Figura 3.17. Resultado final del proceso de segmentación sobre dos imágenes originales (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

El proceso de segmentación completo tiene dos características importantes: esparamétrico en el sentido de que el número de regiones puede fijarse a priori y no hay discontinuidad en el nivel de gris dentro de las regiones, lo cual permite el uso de funciones analíticas globales (como funciones polinomiales) para su descripción.

3.5.2. Codificación de los datos.

Codificación de contorno.

Finalmente, una de las partes de los mensajes a codificar para su transmisión o almacenamiento la constituye una descripción eficiente de los contornos obtenidos después de la segmentación. La descripción precisa de los contornos es esencial para el sistema visual humano. En esta técnica la codificación de contornos se efectúa en dos pasos.

Como las regiones están juntas, los puntos de contorno a lo largo de los bordes de dos regiones adyacentes se describen dos veces, una para cada región. En el primer paso estos puntos se remueven de una de las regiones para describirse y codificarse sólo una vez.

En el segundo paso, los segmentos de contorno restantes se describen en un proceso de tres modos: 1) aproximación por segmentos de línea, 2) aproximación por segmentos circulares y, 3) sin aproximación. Comenzando a partir del primer punto de contorno, el segmento rectilíneo y el segmento circular más largos se dibujan bajo la restricción de que el error máximo entre los datos originales y su aproximación no exceda un límite dado. Se evalúa el costo asociado con cada modo, en términos de número de bits para codificar y se selecciona el modo más "barato". Este proceso lleva a un promedio de 1.6 bits por punto de contorno para imágenes naturales si el límite de error es de 1.1 píxeles.

Codificación de textura.

La parte restante de los mensajes después de la codificación de contorno es la codificación de textura. Nótese que dentro de cada región ya no hay ninguna discontinuidad contundente y por lo tanto, la variación en el nivel de gris dentro de una región puede describirse con funciones polinomiales bidimensionales. La codificación de textura también se efectúa en dos pasos.

En el primer paso, la estructura general del nivel de gris en cada región se aproxima utilizando una función polinomial bidimensional. El orden del polinomio se determina como una función del error de aproximación y del costo involucrado en codificar coeficientes polinomiales. El criterio de aproximación usado es el error promedio cuadrado (MSE), el cual se minimiza a través de cada región para polinomios de orden 0, 1 y 2. Un esquema de estas aproximaciones se muestra en la figura 3.18. Para este caso particular, la mejor aproximación se obtiene con una función polinomial de primer orden.

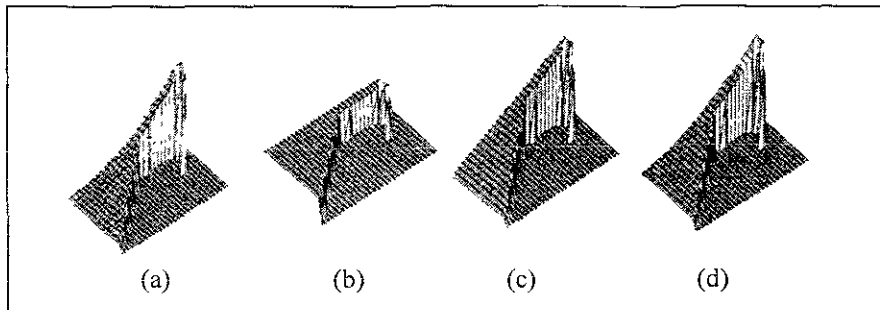


Figura 3.18. Aproximación del nivel de gris de una región con funciones polinomiales bidimensionales. (a) Datos originales.

(b) Aproximación de orden cero. (c) Aproximación de primer orden. (d) Aproximación de segundo Orden (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

En el segundo paso, la granularidad removida con el preprocesamiento se agrega de nuevo en la forma de un ruido pseudo-aleatorio para representar a la imagen con una apariencia más natural. El error promedio cuadrado entre la imagen segmentada y la imagen reconstruida por funciones polinimiales se calcula para cada región. Este error se usa para controlar la varianza de la señal pseudo-aleatoria agregada como microtextura.

La figura 3.19 muestra el resultado de la compresión por formación de región sobre dos imágenes originales con una compresión de 50:1, y la figura 3.20 muestra el diagrama general de esta técnica.



Figura 3.19. Imágenes recuperadas en base a la codificación de formación de región con proporción de compresión de 50:1 (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

La técnica basada en formación de región se constituye como una buena aproximación para la codificación de imagen de alta compresión. Su debilidad radica en la selección de la propiedad usada en la formación de región, la cual lleva a un número mucho mayor de regiones cuyos bordes no necesariamente son los contornos de los objetos en la imagen original.

3.6. COMPRESIÓN BASADA EN DESCOMPOSICIÓN DIRECCIONAL.

En este método, se hace mucho énfasis en la detección de bordes para preservar su información de la mejor manera posible. De acuerdo a las propiedades del sistema visual humano con respecto a la codificación, las características requeridas para un detector de bordes son: precisión en la posición del borde, economía en el sentido de que sea detectado el conjunto más pequeño posible de puntos de borde representativos, y la habilidad para reconstruir el borde original. Sin embargo, la mayoría de los operadores de detección de bordes no resultan óptimos simultáneamente con respecto a estas características.

3.6.1. Descripción de la técnica.

La definición que se utiliza comúnmente para determinar un punto de elemento de borde lleva a la generación de información redundante. La debilidad de esta clase de operador se hace evidente por la cantidad de tareas que se realizan (adelgazamiento de bordes, rastreo de bordes) para mejorar sus resultados. Esa es la razón por la que es preferible definir un elemento de borde (EE) para actuar como una función de pasobidimensional de un ancho y dirección dadas. Por facilidad de notación, dicho elemento de borde se representará con un vector \vec{e} tal que

$$|\vec{e}| = \zeta \quad \text{y} \quad a_1 < \arg(\vec{e}) < a_2 \quad (7)$$

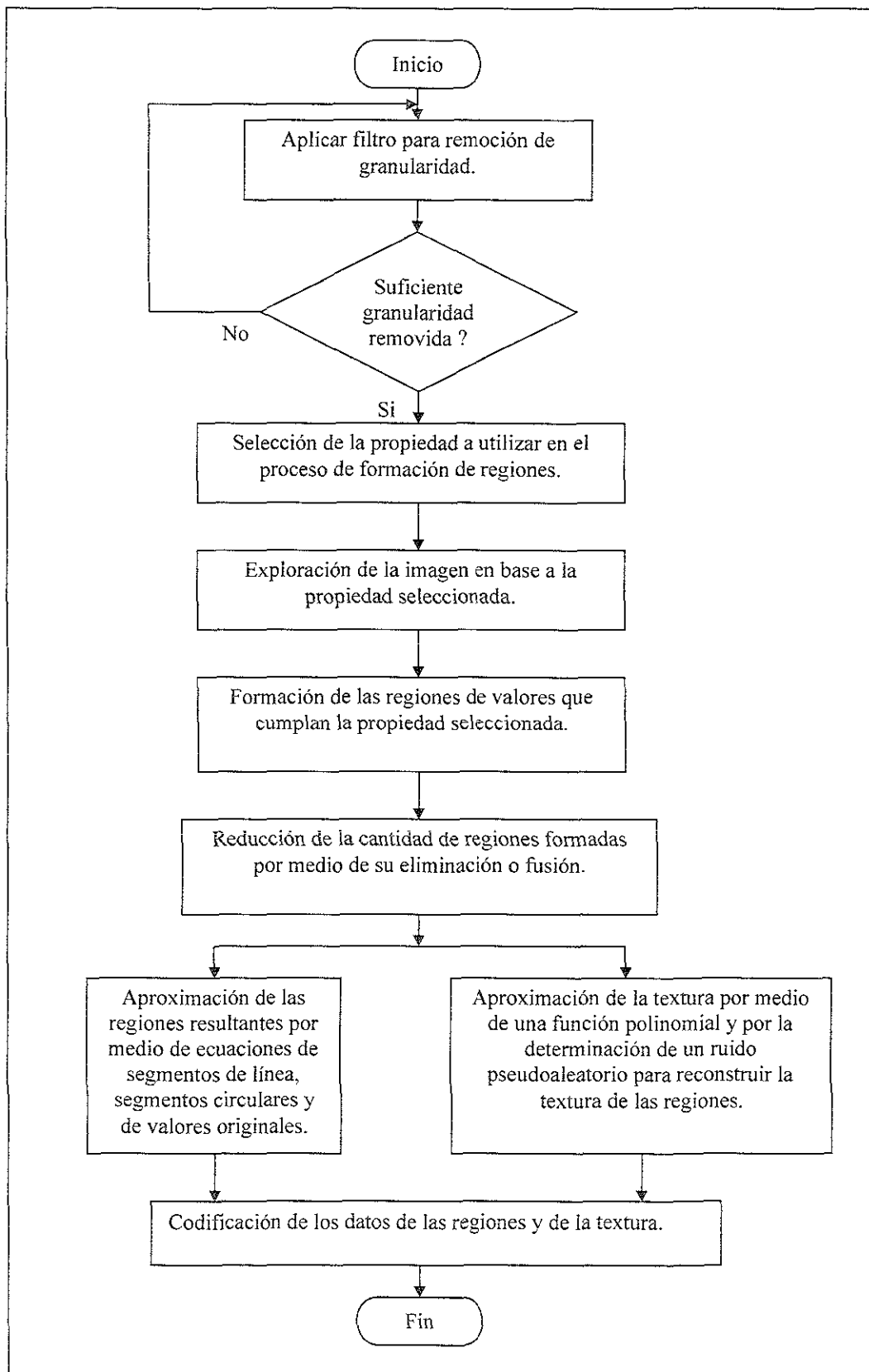


Figura 3.20. Diagrama de la compresión por formación de región.

El objetivo es obtener una representación de los bordes de una imagen por medio de dichos elementos de borde. El conjunto de elementos de borde que proporciona esta representación constituye el conjunto de elementos de borde (ES). Un esquema de detección de borde que utiliza esta representación y produce la información necesaria para reconstruir los bordes, satisface todos los requerimientos mencionados anteriormente. Hay que definir un subconjunto de elementos de bordes (EES) a partir del conjunto original ES, el cual contenga los elementos de borde restringidos a direcciones determinadas, que tienen como límites los ángulos a_1 y a_2

$$S_k = \{ \vec{e}_k \mid \|\vec{e}_k\| = \zeta, \arg(\vec{e}_k) \in (a_1, a_2) \} \quad (8)$$

A cada EES se asocia una dirección principal correspondiente al bisector del ángulo (a_1, a_2) y una dirección ortogonal la cual es perpendicular a la dirección principal. Usando estas definiciones, una imagen direccional se define como una imagen que contiene elementos de borde que pertenecen sólo a un subconjunto de elementos de borde.

Filtros direccionales.

El filtrado direccional se basa en la relación entre la presencia de un borde en una imagen y su contribución al espectro de frecuencia de la imagen. Este filtrado está respaldado por la existencia de neuronas sensibles a la dirección en el sistema visual humano. La transformada discreta de Fourier^{*} de una imagen $x(k, \zeta)$ está dada por:

$$X(m, n) = \sum_{k=0}^{N-1} \sum_{\zeta=0}^{N-1} x(k, \zeta) \exp[-j2\pi (mk + n\zeta) / N] \quad (9)$$

Si se examina el contenido del espectro de frecuencia a lo largo de una línea que pasa a través del origen del dominio de Fourier, escogiendo (por facilidad computacional) la línea $X(0, n)$ se tiene que

$$\begin{aligned} X(0, n) &= \sum_{k=0}^{N-1} \sum_{\zeta=0}^{N-1} x(k, \zeta) \exp[-j2\pi n\zeta / N] \\ &= \sum_{\zeta=0}^{N-1} \exp[-j2\pi (mk + n\zeta) / N] \sum_{k=0}^{N-1} x(k, \zeta) \end{aligned} \quad (10)$$

Esta ecuación puede verse como la transformada de Fourier de una señal unidimensional que representa los valores promedio de la imagen $x(k, \zeta)$ a lo largo del eje ζ . Si existe un

^{*} Para mayor información ver: JAIN, Anil K. Fundamentals of image processing. Prentice-Hall.

borde paralelo a este eje, la señal correspondiente a la segunda igualdad de (10) es una función de filtrado para el borde paralelo. La figura 3.21 esquematiza lo anterior.

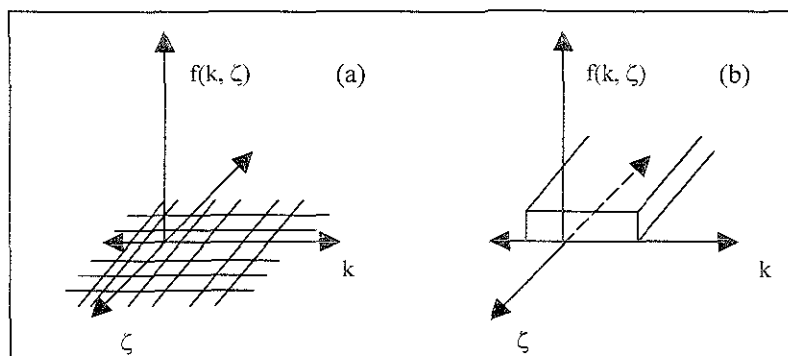


Figura 3.21. Sistema de coordenadas para proyectar una imagen a lo largo del eje ζ (a) y un borde ideal paralelo a él (b).

Por lo tanto, la contribución del borde a lo largo del eje ζ en el dominio de Fourier puede deducirse a partir de la transformada de Fourier de la señal del borde. De forma similar, puede encontrarse el resultado para otras direcciones girando el eje k hacia la izquierda o hacia la derecha hasta 90° y calcular la proyección de la imagen en el eje ζ . Las series de señales que se obtienen se muestran en la figura 3.22.

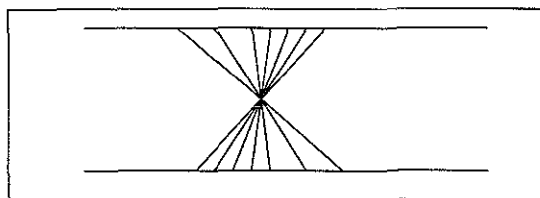


Figura 3.22. Proyecciones de un borde ideal paralelo al eje ζ en varias direcciones.

Puede lograrse una versión teórica de las transformadas de Fourier de estas señales aproximándolas por medio de las siguientes series de funciones de error:

$$g_n(z) = (2\pi / \sigma_n) A \int_{-z}^z \exp(-x^2 / 2\sigma_n^2) dx \tag{11}$$

las cuales se asemejan a las mostradas en la figura 3.22 y tienen un aspecto como el que se muestra en la figura 3.23.

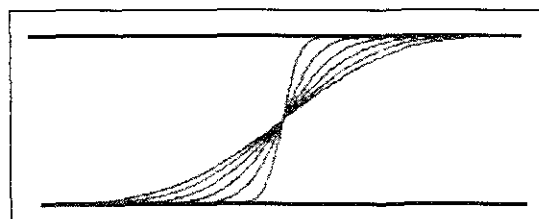


Figura 3.23. Aproximación de las proyecciones de la figura 3.22.

Las transformadas de Fourier de estas aproximaciones están dadas por:

$$G_n(f) = (A / j2\pi f) \exp[-\sigma_n^2 f^2] \quad (12)$$

Puede deducirse de esta ecuación que conforme σ_n^2 , es decir la varianza entre las señales, se reduce, el contenido de alta frecuencia de la transformada de Fourier incrementa. Como resultado, la contribución de un borde está distribuida a través de todo el espectro de frecuencia. El componente de frecuencia más alta se ubica en la dirección ortogonal a la del borde, y conforme se aleja de esa dirección, la frecuencia de las contribuciones disminuye hasta desaparecer a 90° . Por lo tanto, para detectar un subconjunto de elementos de borde, debe conservarse el sector del espectro de frecuencia que corresponde al intervalo de direcciones del EES. Un filtro cuya salida abarca un sector o una parte de un sector en el dominio de frecuencia se denomina filtro direccional. Para realizar más fácilmente la detección de bordes con estos filtros, se emplea un filtrado de paso alto a lo largo de la dirección principal. Así, el plano de frecuencia completo correspondiente a la imagen completa queda cubierto con n filtros direccionales y un filtro de paso bajo. La salida ideal del i -ésimo filtro direccional está dada por:

$$H_i(f, g) = \begin{cases} 1, & \text{si } [\theta(i) < \tan^{-1}(g/f) < \theta(i+1) \text{ y si } (f^2 + g^2) > \rho_c \\ 0, & \text{de otra forma} \end{cases} \quad (13)$$

$$\begin{aligned} \text{con} \quad & \theta(i) = (i-1)\pi / 2n \\ & \theta(i+1) = (i+1)\pi / 2n \\ \text{y} \quad & |f|, |g| < 0.5 \end{aligned}$$

donde f y g son frecuencias espaciales, ρ_c es la frecuencia de corte del filtro de paso bajo, y donde se asume un tamaño de paso de muestreo de 1.

De acuerdo a esto, un filtro direccional es un filtro de paso alto a lo largo de su dirección principal y un filtro de paso bajo a lo largo de la dirección ortogonal. El ancho de banda a lo largo de esta dirección es $g \tan(\pi / 2n)$. La figura 3.24 esquematiza esta situación.

La salida ideal de los filtros direccionales deberá modificarse por medio de una función ventana apropiada debido a la formación de oscilaciones alrededor de los cruces cero, es decir los puntos cuya frecuencia es cero, correspondientes a los bordes reales. Una de las funciones ventana más apropiada para este propósito es la ventana gaussiana dada por:

$$W(k, \zeta) = \exp[-(k^2 + \zeta^2) / s^2] \quad (14)$$

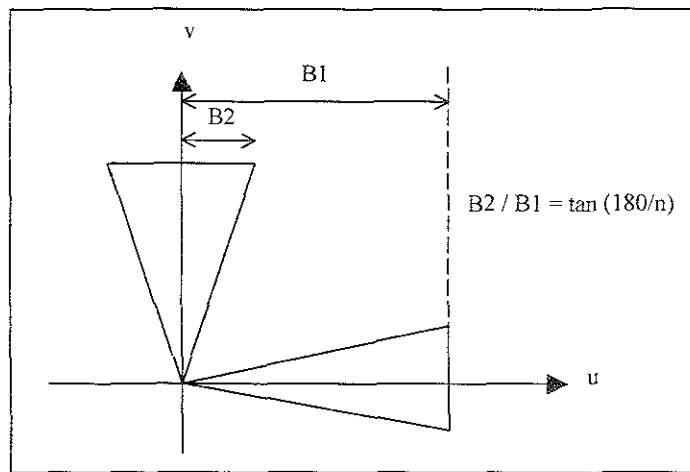


Figura 3.24. Ancho de banda de un filtro direccional.

Después de aplicar la ventana a los filtros, filtrar y luego superponer todas las imágenes direccionales y la imagen de paso bajo, se obtiene como resultado la imagen original. Por lo tanto, el filtrado direccional es una transformación que preserva la información de la imagen. La figura 3.25 es un ejemplo de la descomposición direccional con ocho filtros direccionales.

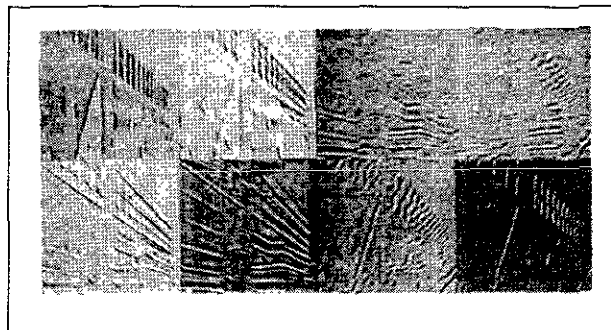


Figura 3.25. Descomposición direccional en ocho componentes.

Existe sin embargo, una cuestión por establecer. Si la varianza general s^2 de la función ventana es muy grande, los filtros direccionales son más selectivos en su sector pero muy pobres como filtro de paso bajo a lo largo de la dirección ortogonal. Este comportamiento introduce oscilaciones indeseables en los bordes detectados. Por el contrario, si este parámetro es muy pequeño, estas oscilaciones se atenúan considerablemente en detrimento de la selectividad direccional por lo que el mismo borde puede aparecer en varias imágenes direccionales. Una solución a esta cuestión puede encontrarse utilizando una transformación no lineal. Esta transformación consiste primero en calcular la suma de los valores de los puntos de imagen ubicados en la misma posición en todas las imágenes direccionales y reteniendo el punto de máximo valor. Entonces, todos los otros puntos se

asignan como cero excepto el punto de máximo valor, el cual toma el valor de la suma. Una ilustración de esta transformación es como la que se muestra en la figura 3.26.

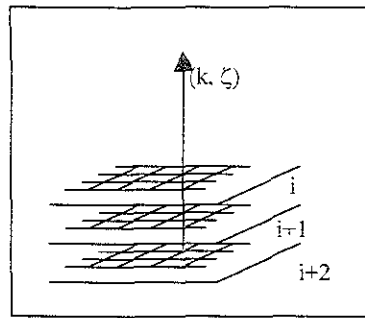


Figura 3.26. Ilustración de la transformación no lineal para mejorar la selectividad direccional.

Esta transformación mejora considerablemente la selectividad direccional y mantiene las oscilaciones indeseables a un nivel aceptable.

Hay otros dos parámetros importantes en el desempeño de los filtros direccionales, su cantidad y la frecuencia de corte del filtro de paso bajo. El número de filtros está directamente relacionado con el ancho mínimo de los elementos de borde que es aceptado a priori en la imagen. Por lo tanto, puede definirse el número de filtros direccionales fijando la longitud mínima de los elementos de borde aceptados. Otro factor a considerar es la segmentación de bordes curvos (cuantificación de curvatura) que resulta de la definición de los elementos de borde. Por otra parte, la elección de la frecuencia de corte tiene influencia directamente sobre la proporción de compresión y la apariencia de la imagen codificada.

3.6.2. Codificación de los datos.

Los mensajes a codificar los constituyen las imágenes direccionales y la imagen de paso bajo. El objetivo principal es alcanzar la mayor compresión para un nivel de degradación de imagen dado. El componente de baja frecuencia es, por su naturaleza, adaptado para codificación de transformada en virtud de su alta compactación de energía. La parte débil de la codificación de transformada es su capacidad para la codificación de bordes. Como no hay bordes en el componente de baja frecuencia, la codificación de transformada es el método más adecuado, por lo que no se muestra degradación por la codificación de este componente.

Las imágenes de alta frecuencia se usarán para detectar y codificar los bordes. A este proceso está asociada una pérdida de información y su consecuente degradación. La pérdida de información proviene de la selección entre los bordes fuertes y los débiles. Si la proporción de compresión se establece a tasas altas, los bordes débiles se eliminarán. Por otra parte, la aproximación indirecta de los bordes por medio de segmentos de línea, como

se asume por la definición de los elementos de borde, introduce algunas degradaciones en las posiciones con mucha curvatura. La detección de bordes en las imágenes direccionales se basa en el carácter de paso alto de los filtros direccionales a lo largo de su dirección principal. Filtrar una señal con un filtro de paso alto proporciona cruces de valor cero en las posiciones de cambios abruptos (bordes). De acuerdo a esto, la detección de bordes en las imágenes direccionales se efectúa buscando los cruces cero a lo largo de la dirección principal de cada imagen.

Para facilidad de implementación, se introduce una normalización de todas las direcciones principales en la dirección horizontal. Esta se basa en operaciones de interpolación, rotación y submuestreo. La detección de bordes en las imágenes normalizadas se realiza buscando los cruces cero a lo largo de las columnas utilizando los siguientes parámetros:

$$\begin{aligned} P_1 &= x_i(k, \zeta) \cdot x_i(k+1, \zeta) \\ P_2 &= x_i(k, \zeta) \cdot x_i(k+2, \zeta) \end{aligned} \quad (15)$$

donde $x_i(k, \zeta)$ representa la i -ésima imagen direccional.

La existencia de un cruce cero se establece por la condición $P_1 < 0$ o $P_1 = 0$ y $P_2 < 0$. Otra variable que debe usarse después de esta etapa es la magnitud de los cruces cero, definida como:

$$P_3 = |x_i(k, \zeta) - x_i(k+k', \zeta)| \quad (16)$$

donde $k' = 1$ o a 2 si $P_1 < 0$ o $P_1 = 0$ y $P_2 < 0$, respectivamente. Estableciendo un límite a esta variable, es posible controlar la intensidad de los bordes a retener.

Debido a la reducción del ancho de banda a lo largo de las líneas (dirección ortogonal) y el submuestreo, la detección de bordes se aplica a imágenes direccionales más pequeñas. De acuerdo a esto, cada cruce cero detectado corresponde a un elemento de borde como se definió en (7). Su dirección pertenece al sector determinado por el filtro direccional ($180/2n$) y su longitud en pixeles es igual a la tasa de submuestreo. Un ejemplo de detección de bordes por cruces cero se muestra en la figura 3.27.

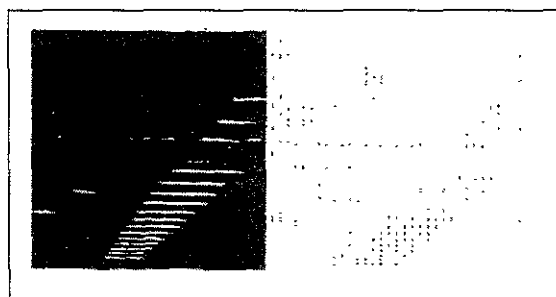


Figura 3.27. Imagen direccional del edificio de la figura 3.6 y la detección de bordes con sus cruces cero.

La figura 3.27 muestra la imagen resultante de la detección efectuada sobre una imagen direccional aplicando un algoritmo de detección para cada quinta columna (tasa de submuestreo 5:1). Cuando se detectan los bordes en todas las imágenes direccionales, el resultado obtenido es como se muestra en la figura 3.28.

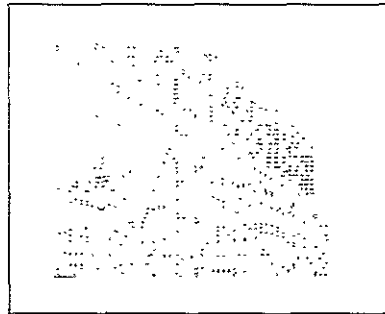


Figura 3.28. Resultado global de la detección de cruces cero en ocho imágenes direccionales.

La imagen de la figura 3.28 se obtuvo rotando y superponiendo cada imagen direccional después de la detección de bordes. La figura 3.29 muestra la imagen de bordes obtenida por interpolación entre los cruces cero de cada imagen direccional antes de la rotación y superposición.

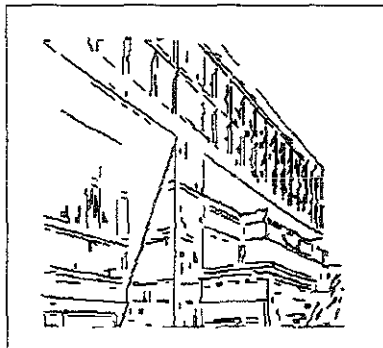


Figura 3.29. Bordes obtenidos después de la interpolación.

Nótese que la figura 3.28, que contiene 1107 puntos, permite la reconstrucción completa de la figura 3.29, la cual tiene 5381 puntos. Este resultado muestra que los requerimientos de detección económica y precisa para la generación de detectores de bordes se satisfacen ampliamente.

Cada imagen direccional se representa por las posiciones y las magnitudes de los cruces cero. Las posiciones se codifican con el método run-length usando código Huffman, lo que requiere un promedio de 4.5 bits por posición. Las magnitudes de los cruces cero se codifican utilizando diferentes niveles de cuantificación, tomando por lo general una palabra de código de 3 bits para codificar las magnitudes de los cruces cero. Las mejoras

obtenidas usando palabras de código mayores no son suficientes para justificar el precio pagado, mientras que palabras más pequeñas llevan a distorsiones notorias.

La imagen de baja frecuencia puede codificarse de por alguna de dos formas. Como la frecuencia máxima de este componente es mucho menor, puede submuestrearse y los pixeles resultantes pueden codificarse con un procedimiento sin pérdida de información. La otra alternativa es la codificación de transformada. La elección de la técnica de transformada depende directamente del tipo de filtrado utilizado. Las posiciones de los coeficientes Fourier se conocen a partir de las características del filtro, y la importancia de todos estos coeficientes obliga a la selección de un método de cuantificación lineal con palabras de longitud fija para codificar la posición y palabras de longitud variable, asignadas por el código Huffman, para codificar las magnitudes.

Decodificación de los datos.

Para reconstruir la imagen original, todos los componentes deben decodificarse y sumarse. El componente de baja frecuencia se obtiene por transformación inversa de los coeficientes codificados. El componente de alta frecuencia se obtiene sintetizando las imágenes direccionales a partir de los cruces cero. La síntesis de los bordes a partir de la información de los cruces cero y la interpolación entre las columnas de las imágenes direccionales normalizadas son los procesos más críticos para la apariencia de la imagen decodificada. El modelo de borde, presentado al comienzo de esta descripción, ofrece la base teórica para la síntesis de las señales unidimensionales a lo largo de las columnas. La ecuación (11) requiere dos parámetros: la magnitud A de los cruces cero representando el contraste del borde, y la desviación estándar relacionada con lo pronunciado de la inclinación del borde. Conforme se codifica la magnitud de los cruces cero, el único parámetro desconocido es la desviación estándar en virtud de que depende directamente de los datos de entrada. Existen dos posibles soluciones para esta situación. La primera consiste en asumir un valor constante para la desviación estándar. Esta solución brinda un borde estándar más o menos regular. La segunda consiste en asumir una dependencia de la desviación estándar en el contraste. Resultados experimentales indican que una variación lineal de la desviación estándar con el contraste proporciona bordes más realistas. La función prototipo adoptada para aproximar los perfiles de los cruces cero es la siguiente:

$$g(u) = (u / Ak) \exp [-u^2 / Ak] \quad (17)$$

donde u es la distancia a partir del cruce cero con $u = 0$, A es la magnitud y k una constante. La familia de estas funciones a diferentes valores de A se ilustra en la figura 3.30.

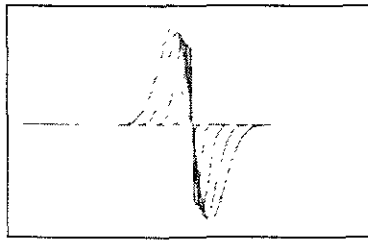


Figura 3.30. Perfiles de cruces cero sintéticos (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

Nótese que la función prototipo de (17) está muy relacionada con la ecuación (11).

Una vez efectuada la síntesis de los perfiles de cruces cero en las posiciones codificadas, la imagen direccional completa se construye por interpolación entre las columnas de las imágenes submuestreadas. Para una interpolación perfecta entre las columnas de estas imágenes, debe tomarse en cuenta el hecho de que los elementos de borde asumidos por la presencia de cada cruce cero pueden tener cualquier dirección dentro de este intervalo. El algoritmo de interpolación busca un punto vecino no sólo en la misma línea sino también en las dos líneas precedentes o siguientes.

La figura 3.31 muestra el diagrama general de la compresión por descomposición direccional, y la figura 3.32 muestra el resultado final de la decodificación de descomposición direccional con una proporción de compresión de 41:1.

3.7. COMPRESIÓN BASADA EN RECONOCIMIENTO DE PATRONES.

La codificación basada en reconocimiento de patrones es una técnica de compresión que aprovecha las propiedades macroscópicas de una imagen. Combina la compresión y el reconocimiento de patrones, llevando a altas proporciones de compresión. Las imágenes fuente tienen ciertos patrones como caracteres, líneas y cajas rectangulares. Reconociendo estos patrones o acoplando los patrones en una imagen fuente con los patrones ya transmitidos o almacenados, la imagen fuente puede describirse en términos de los patrones de figuras que la conforman. Este método de compresión es muy útil para el reconocimiento de caracteres y aplicaciones de entendimiento del contenido para imágenes de documentos y para visión computacional.

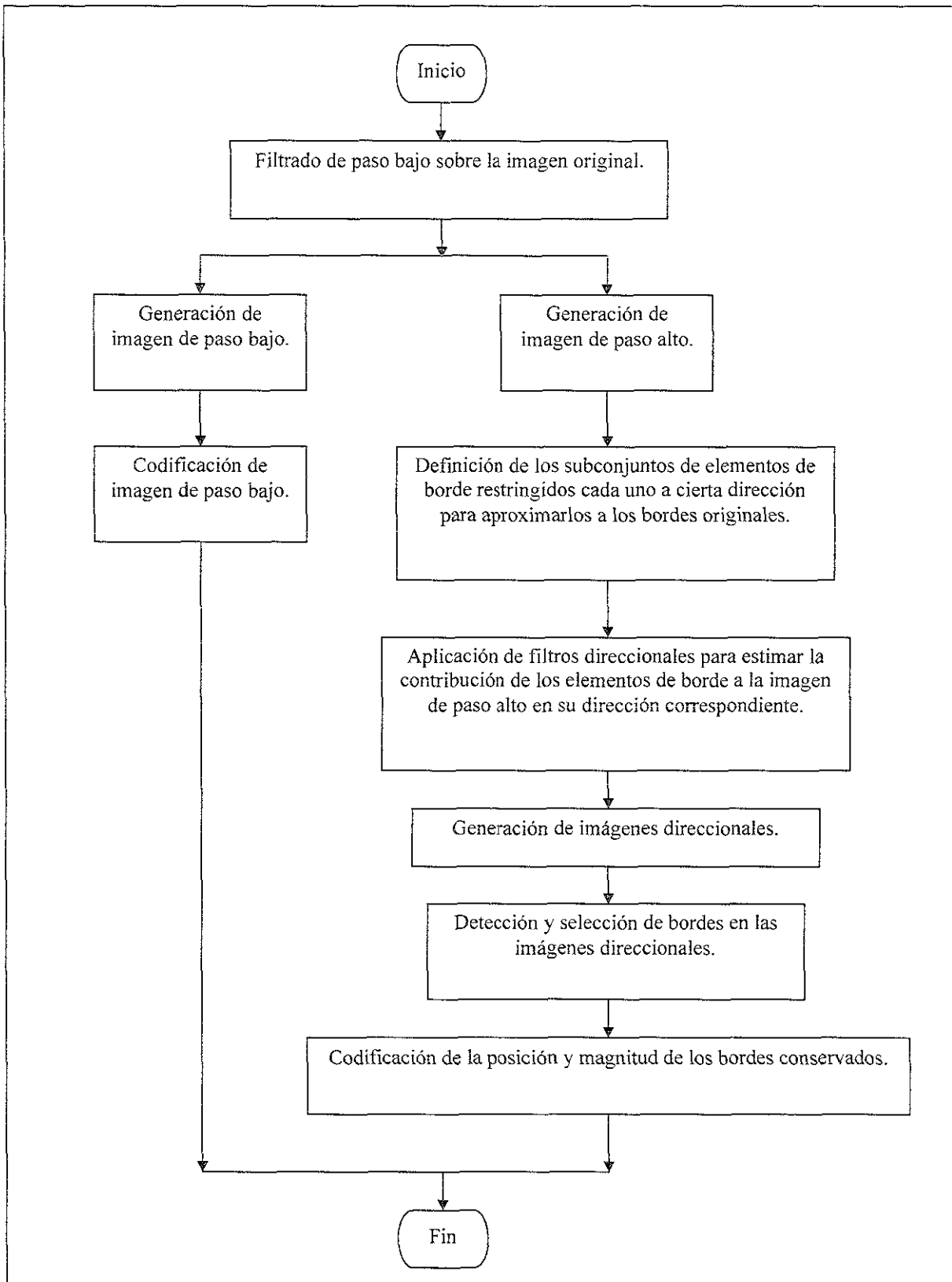


Figura 3.31. Diagrama de la compresión por descomposición direccional.

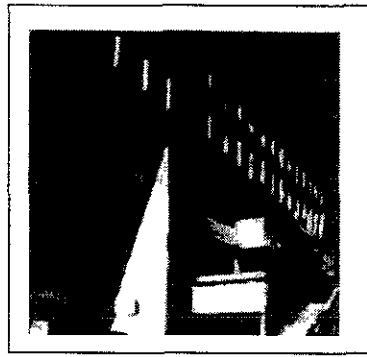


Figura 3.32. Imagen decodificada con descomposición direccional con proporción de compresión de 41:1 (IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques").

3.7.1. Descripción de la técnica.

Hay dos técnicas básicas implicadas en la codificación por reconocimiento de patrones: el entendimiento de patrones y el acoplamiento de patrones. La técnica de entendimiento de patrones reconoce patrones tales como caracteres alfabéticos junto con la información de su tipografía, mientras que la técnica de acoplamiento de patrones simplemente acopla los patrones en la imagen fuente con los patrones ya almacenados o transmitidos previamente, por lo que los patrones en la imagen fuente son reemplazados por los patrones ya conocidos. La ventaja de la técnica de entendimiento de patrones es que combina los procesos de compresión y de reconocimiento, los cuales llevan a una proporción de compresión alta y facilita un procesamiento posterior de la información. La ventaja de la técnica de acoplamiento de patrones es que preserva alguna información detallada de la imagen. La desventaja del acoplamiento de patrones es que su eficiencia de compresión es menor a la técnica de entendimiento de patrones, y puede requerir trabajo adicional para un procesamiento posterior de la información. Un diagrama simplificado del sistema de codificación basado en reconocimiento de patrones se muestra en la figura 3.33.

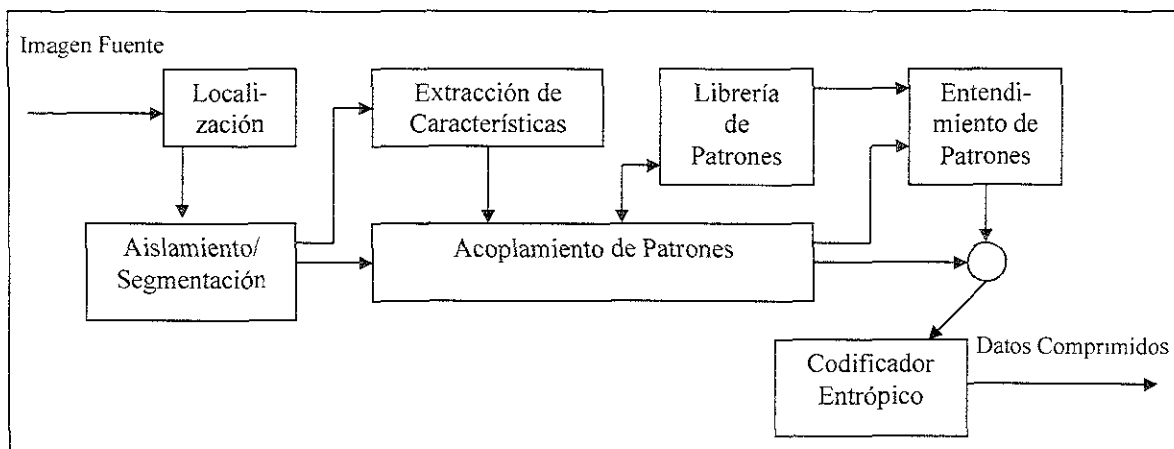


Figura 3.33. Diagrama del sistema de codificación basado en reconocimiento de patrones.

En la figura 3.33 el bloque de localización de patrón localiza un patrón en la imagen fuente. El bloque de aislamiento/segmentación de patrón aísla o segmenta el patrón del resto de la imagen. El patrón se transmite al bloque de acoplamiento de patrón. Con ayuda del bloque de extracción de características, el bloque de acoplamiento de patrones identifica el patrón que se acople mejor de entre los patrones almacenados en la librería de patrones.

La extracción de características durante este proceso de identificación reduce la cantidad de patrones parecidos posibles de la librería de forma tal que se requieran menos patrones de la librería para realizar comparaciones con el patrón de entrada. Si no se encuentra un patrón lo suficientemente similar en la librería, el patrón de entrada se convierte en un nuevo patrón y se almacena en la librería.

El proceso de acoplamiento de patrones produce un índice de la librería además de información adicional como la ubicación del patrón dentro de la imagen fuente. Con la técnica de acoplamiento de patrones, este índice junto con la información adicional se transmite al codificador entrópico. En el caso de la técnica de entendimiento de patrones, este índice junto con la información adicional se transmite al bloque de entendimiento de patrones.

Con la información del índice, el bloque de entendimiento de patrones extrae el patrón de la librería de patrones y produce información descriptiva como una letra con cierta tipografía y fuente. Después, la información descriptiva junto con la información adicional se codifica con el codificador entrópico.

En la técnica de compresión basada en reconocimiento de patrones hay involucradas varias operaciones propias del reconocimiento de patrones:

- Localización del patrón: examina la imagen fuente línea por línea.
- Aislamiento/Segmentación de patrón: Cargado del patrón al sistema.
- Extracción de Características: extraer ciertas características del patrón.
- Acoplamiento de patrones: acoplar un patrón contra los patrones almacenados.
- Entendimiento de patrones: produce información descriptiva de un patrón dado.

Algunos investigadores han usado codificación basada en reconocimiento de patrones para codificar imágenes binivel (blanco y negro) con proporciones de compresión mayores al doble que las logradas por el estándar CCITT T.6. Actualmente, este método se está considerando para desarrollar una extensión del estándar JBIG.

3.8. COMPRESIÓN BASADA EN SEGMENTACIÓN DE ESTRUCTURA DE ÁRBOL.

3.8.1. Descripción de la técnica.

El algoritmo de segmentación de estructura de árbol particiona la imagen en una cantidad de polígonos siguiendo un criterio de uniformidad, donde un polígono se define como cualquier secuencia continua parecida de segmentos de línea. Estos segmentos de línea son los lados del polígono. Estos polígonos se estructuran como una representación de árbol binario. La raíz del árbol corresponde a la imagen completa. Los dos nodos de la raíz representan a dos subimágenes limitadas por polígonos canónicos, los cuales se particionan en cuatro n -gonos convexos con $3 \leq n < 8$ (donde un subconjunto C de R^n se dice que es convexo si $(1 - \lambda)x + \lambda y \in C$ siempre que $x \in C$, $y \in C$ y $0 < \lambda < 1$). La dirección para realizar el corte en cuatro puede tener uno de cuatro valores posibles: horizontal, vertical, 45 grados y 135 grados. La segmentación recursiva produce un árbol binario con k hojas. La imagen se codifica como una secuencia de $k - 1$ códigos de partición y k códigos de llenado. Un código de partición se compone de la información de la dirección y del desplazamiento de los cortes de los cortes de segmentación. El desplazamiento de corte es con respecto a un vértice dado del polígono canónico que limita a la subimagen que se está cortando. Un código de llenado contiene un valor de intensidad de la luminancia de cada región. El proceso de decodificación es simplemente el llenado del polígono.

En la figura 3.34 se muestran ejemplos de polígonos canónicos, donde UL, LR, LL y UR representan los vértices superior más izquierdo, inferior más derecho, inferior más izquierdo y superior más derecho, respectivamente. Estos vértices se denominan los vértices clave de un polígono.

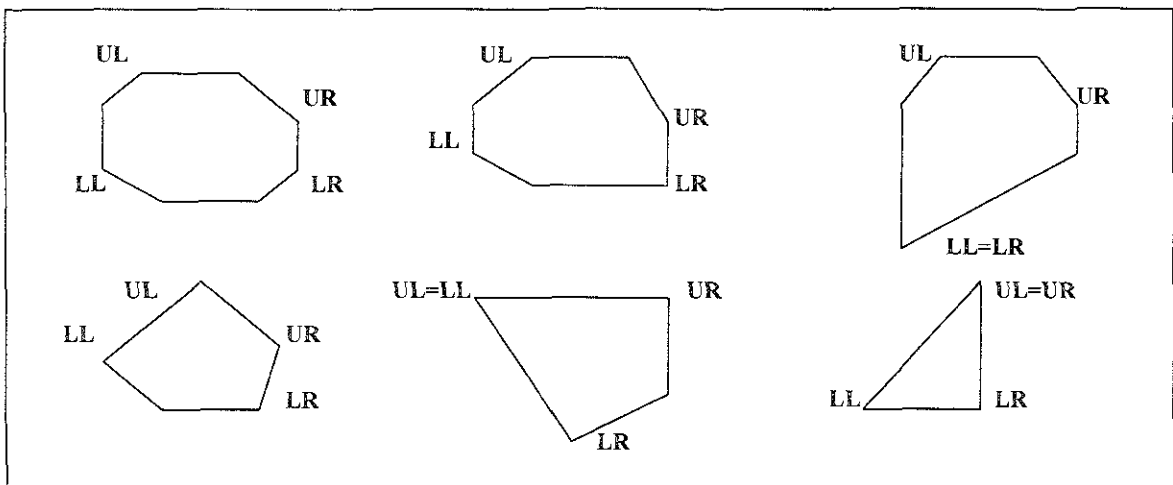


Figura 3.34. Ejemplos de polígonos canónicos(KOU, Weidong. *Digital image compression. Algorithms and standards*).

3.9. RESUMEN DE LA COMPRESIÓN DE SEGUNDA GENERACIÓN.

Las técnicas de compresión de segunda generación toman como referencia para la definición de la imagen los contornos contenidos en ella, ya sea en el contexto de su importancia para el espectro de frecuencia de la imagen o en el contexto de la formación de grupos de píxeles contenidos entre dichos contornos. Las técnicas basadas en operador local descomponen una imagen original en versiones de frecuencia de ella, por medio de filtros que determinan la importancia de los contornos detectados para la imagen ya sea completa o restringida a cierta área. Las técnicas orientadas a contorno-textura dividen la imagen en regiones de píxeles rodeadas por contornos que buscan aproximarse a los contornos reales de la imagen.

La tabla 3.1 muestra un cuadro comparativo de las técnicas de compresión de segunda generación similar al mostrado en la sección 2.7, el cual muestra los mismos criterios de comparación con el fin de establecer la diferencia entre ambas clasificaciones. La definición de cada criterio es la misma, sin embargo es importante hacer el énfasis de que estas técnicas si consideran para su funcionamiento el contenido particular del área de selección que utilizan, es decir los contornos y texturas de la imagen, a diferencia de las técnicas revisadas en el capítulo anterior. Estas técnicas resultan más complejas que las de forma de onda ya que la mayoría efectúan filtrado digital de la información de forma iterativa. Además, algunas de ellas utilizan más de una herramienta de selección de información para efectuar sus procesos lo que eleva su complejidad y la carga de trabajo para el procesador. Este es el motivo por el que, a excepción de la compresión piramidal, la mayoría de las técnicas de esta clasificación aún no se concretan en la forma de un estándar o formato de compresión comercial en la actualidad.

Técnica de compresión	Herramienta de selección de información	Área de selección inicial	¿Iterativa con respecto al área de selección?	¿Redefinición del tamaño para área de selección en cada iteración?	¿Redefinición de parametros para la herramienta en cada iteración?	¿La técnica considera el contenido particular del área de selección?
Sistema Alto Sintético	Filtros digitales	Imagen completa	No	No	No	Sí
Piramidal	Filtros digitales	Imagen completa	Sí	Sí	Sí	No
Anisotrópica	Filtros digitales	Imagen completa	Sí	Sí	Sí	Sí
Formación de Región	Filtros digitales	Imagen completa	Sí	No	No	Sí
	Agrupamiento de píxeles	Imagen completa	No	No	No	

Direccional	Filtros digitales Filtros direccionales	Imagen completa Imagen completa	No Sí	No No	No Sí	Sí
Reconocimiento de Patrones	Patrones de pixeles	Patrón de pixeles	Sí	No	Sí	Sí
Segmentación de Árbol	Árbol de segmentación binario	Imagen completa	Sí	Sí	No	Sí

Tabla 3.1. Cuadro comparativo de las técnicas de compresión de segunda generación.

CAPÍTULO IV:

TÉCNICAS BASADAS EN FRACTALES.

4.1. DEFINICIÓN Y CLASIFICACIÓN DE LAS TÉCNICAS DE COMPRESIÓN FRACTAL.

En los últimos años, el término “fractal” se ha hecho popular en varias aplicaciones computacionales. Una de esas aplicaciones es la compresión fractal de imagen digital. Ciertas técnicas de compresión fractal pueden ofrecer una proporción de compresión mucho mayor (100:1) comparadas con otras técnicas de compresión tradicionales como las de forma de onda.

Considerando una imagen digital, se hace uso de técnicas de procesamiento de imagen como separación de color, detección de bordes, análisis del espectro de frecuencia y análisis de variación de textura, para descomponer la imagen en segmentos denominados fractales. Un segmento puede ser cualquier objeto en una imagen natural como una nube o una hoja. Un segmento también puede ser una colección más compleja de píxeles como la arena, rocas o niebla en la imagen natural de una playa. Posteriormente se almacena una referencia de estos segmentos en una librería de fractales, la cual no contiene partes de imagen literalmente, sino conjuntos compactos de números, llamados códigos de sistema de funciones iteradas (IFS), que servirán para reproducir los fractales correspondientes. Además, el sistema de catálogo de la librería es de una eficiencia tal, que los coeficientes correspondientes a imágenes parecidas están cerca unos de los otros. Esta característica lo hace conveniente para establecer procesos automatizados de búsqueda en la librería con el fin de encontrar fractales que se aproximen a una imagen objetivo dada y así generar una versión muy parecida de ella. Un resultado matemático conocido como el teorema del collage garantiza que siempre se podrá encontrar un código IFS conveniente para aproximar una imagen por medio de segmentos básicos o fractales. El sistema de funciones iteradas considera actuar sobre la imagen completa de forma similar a la técnica sub-banda y a partir de ello generar los fractales necesarios para describirla. Si la aplicación del IFS se restringe a ciertas regiones de dimensión establecida se trata de un sistema de funciones iteradas particionado o también denominado técnica de transformada fractal. La figura 4.1 muestra un diagrama de dicha clasificación.

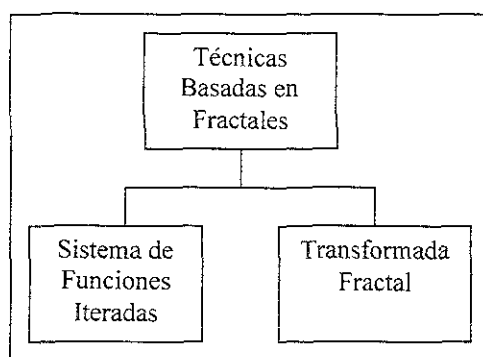


Figura 4.1. Clasificación de las técnicas basadas en fractales.

4.2. DEFINICIÓN DE FRACTAL.

Una definición conveniente de fractal para el área de la compresión de imagen es considerarlo como una colección de píxeles en una imagen que tienen una propiedad de autosimilitud.

Esta definición puede extenderse para el campo de las imágenes digitales, definiéndolo como una imagen infinitamente magnificable que puede generarse por medio de un pequeño conjunto de instrucciones y datos. Con un fractal, entre más acercamientos se realicen sobre una imagen, la imagen amplificada se hará más detallada. En cambio, si se realizara un acercamiento sobre una imagen codificada con cualquier otra técnica, eventualmente se llegarán a ver únicamente bloques de los píxeles que integran la imagen.

La teoría fractal fue presentada por Benoit Mandelbrot para definir una estructura fracturada que posee formas similares a muchos tamaños distintos. Las imágenes generadas con fractales tienen una apariencia agradable al ojo humano debido a su bajo contenido de información en el sentido matemático, lo cual se deriva de que su formación es producto de un sistema de ecuaciones finito, aún cuando la imagen pareciera ser infinita. Gracias a las características de integración del sistema visual humano, el cerebro percibe una imagen completa formada a partir de segmentos codificados separadamente. Mandelbrot creó algunas de las primeras imágenes fractales abstractas, y observó que existen estructuras matemáticas similares detrás de la geometría de la naturaleza. La figura 4.2 muestra un ejemplo de dichas imágenes.

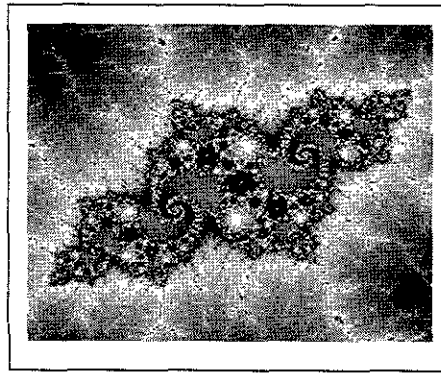


Figura 4.2. Ejemplo de imagen fractal del conjunto generado por Mandelbrot.

Esta definición sólo abarca el caso de las imágenes, y mediante ella es posible definir a la compresión fractal de imagen como un esquema de compresión que codifica una parte de una imagen por medio de otras partes más pequeñas y simples de una imagen original, a través de algunas técnicas de transformación como el escalado y la rotación.

La clave detrás de la compresión fractal de imagen es la autosimilitud de las imágenes. Al particionar una imagen en segmentos, uno de esos segmentos puede ser similar a otros. Puede ser que dos segmentos sean idénticos o que un segmento pueda aproximarse a otro después de escalarlo o rotarlo. Una ilustración de estos procesos se muestra en la figura 4.3.

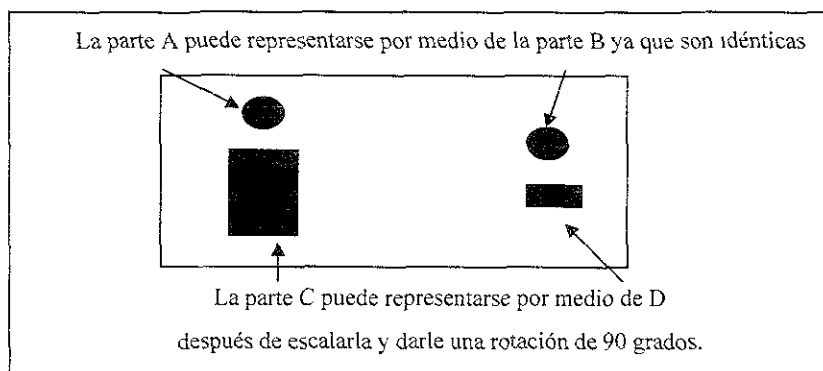


Figura 4.3. Ilustración del escalado y rotación de fractales (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

Para entender el funcionamiento de la codificación fractal es necesario definir términos como métrica, punto fijo y mapeo contractivo, para posteriormente describir las operaciones que efectúa el sistema de funciones iteradas.

4.2.1. Métrica.

Un espacio métrico (G, d) es un conjunto G con una métrica d que mide la distancia entre dos puntos en G , donde d tiene las siguientes propiedades:

- 1.- $d(u, v) = d(v, u), \forall u, v \in G$
- 2.- $0 < d(u, v) < \infty, \forall u, v \in G, u \neq v$
- 3.- $d(u, u) = 0, \forall u, v \in G$
- 4.- $d(u, v) \leq d(u, w) + d(w, v), \forall u, v, w \in G$

Una secuencia $\{x_n\}_{n=1}^{\infty}$ de puntos en un espacio métrico (G, d) se denomina una secuencia Cauchy, si existe un entero $N > 0$ para cualquier número dado $\varepsilon > 0$, tal que $d(x_m, x_k) < \varepsilon$ para toda $m, k > N$.

Un espacio métrico (G, d) es completo si cada secuencia Cauchy en el espacio métrico tiene un límite en ese espacio.

Un subconjunto de un espacio métrico (G, d) es compacto si cada secuencia infinita en el subconjunto contiene una subsecuencia que tiene un límite en el subconjunto.

Un ejemplo de métrica es la medida de distorsión del error cuadrado promedio (MSE) expuesta en la sección 1.7.3.1.

4.2.2. Punto Fijo.

Sea $f : G \rightarrow G$ una transformación en un espacio G . Un punto $U_f \in G$ tal que $f(U_f) = U_f$ se denomina un punto fijo de la transformación.

Un punto fijo es el punto que es mapeado hacia sí mismo bajo un mapeo dado.

4.2.3. Mapeo Contractivo.

Distancia y Métrica de Hausdorff: Sea (G, d) un espacio métrico completo y $H(G)$ el espacio de subconjuntos compactos no vacíos de G . Entonces, para un punto dado $u, v \in H(G)$, la distancia Hausdorff está definida por:

$$h(u, v) = d(u, v) \vee d(v, u)$$

donde la notación $x \vee y$ significa el máximo de los dos números x o y .

La función h es una métrica y $(H(G), h)$ es un espacio métrico compacto. La métrica h se denomina la métrica Hausdorff.

Un ejemplo de mapeo contractivo es el mapeo de dos puntos en otros dos puntos distintos, con valores en las coordenadas de un medio de los valores de los dos puntos originales.

Transformación Contractiva: Una transformación $f: G \rightarrow G$ en un espacio métrico (G, d) es contractiva si existe una constante $0 \leq s < 1$ tal que $d(f(u), f(v)) \leq s \cdot d(u, v) \forall u, v \in G$. La constante se denomina factor de contractividad.

Teorema de una Transformación Contractiva: Una transformación contractiva f en un espacio métrico completo (G, d) posee un punto fijo único:

$$U_f = \lim_{n \rightarrow \infty} f^{on}(u), \forall u \in G$$

donde f^{on} es la n -ésima composición de f consigo misma.

Un ejemplo de dicha transformada contractiva son las denominadas transformadas afines, las cuales son de la forma:

$$w(x) = Ax + b \text{ con } x, b \in \mathbb{R}^2.$$

Las transformadas afines pueden rotar, escalar y trasladar un conjunto de coeficientes. Para una colección w_1, \dots, w_n de dichas transformaciones afines, la transformación $W = \bigcup_{i=1}^n w_i$ es una transformada afín. De la teoría de la transformación contractiva, se determina que para el punto fijo x_f ,

$$x_f = W(x_f) = w_1(x_f) \cup \dots \cup w_n(x_f).$$

Con los factores de contractividad respectivos, una colección w_1, \dots, w_n y el espacio euclidiano \mathbb{R}^2 , se forma un sistema de funciones iteradas (IFS). El IFS es la herramienta básica de la codificación de fractales. Mediante el IFS, es posible buscar las transformaciones w_1, \dots, w_n necesarias para describir y codificar un fractal dado.

4.3. COMPRESIÓN BASADA EN EL SISTEMA DE FUNCIONES ITERADAS (IFS).

En base a las definiciones de la sección anterior, un IFS consiste en un espacio métrico completo y en un conjunto finito de transformaciones contractivas además de sus factores

de contractividad respectivos. Un ejemplo de sistema de funciones iteradas es la colección de todos los valores posibles de cualquier imagen bidimensional junto con su medida de distorsión de error cuadrado promedio y un conjunto de mapeos contractivos.

La teoría IFS es una extensión de la geometría clásica. Utiliza transformaciones afines para expresar relaciones entre partes de una imagen. Utilizando solamente estas relaciones, es posible definir y transmitir imágenes complejas. Con la teoría IFS, es posible describir algún objeto complejo dentro de una imagen natural de forma tan exacta como si se tuviera un plano del mismo.

4.3.1. Transformaciones afines.

Las transformaciones afines, como se mencionó en la sección anterior, pueden describirse como combinaciones de rotaciones, escalados y traslaciones de los ejes de coordenadas en el espacio n -dimensional. Un ejemplo de esta descripción en dos dimensiones es:

$$W(x, y) = (1/2x + 1/4y + 1, 1/4x + 1/2y + 2)$$

Lo que también puede expresarse en forma matricial:

$$W \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} .5 & .25 \\ .25 & .5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Esta transformación mueve el punto $(0,0)$ al $(1,2)$ y mueve $(-1,0.5)$ a $(0.625,2)$. Se denota esta transformación con W ; la notación $W(S)$ denota la subimagen de W en un conjunto de puntos S . En la figura 4.4 se ilustra el efecto de una transformación afín sobre una imagen F , con base en el plano x, y .

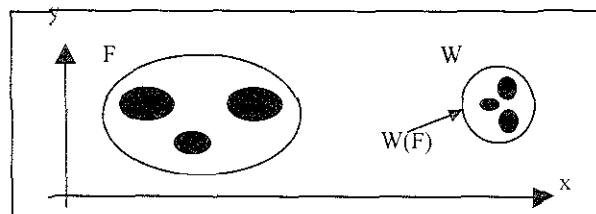


Figura 4.4. Una transformación afín W mueve la imagen F hacia una nueva imagen $W(F)$ (BARNSELEY, Michael F., SLOAN, Alan D. "A better way to compress images").

El resultado es una nueva cara estirada $W(F)$. La transformación afín ha deformado y movido la cara. Nótese que los ojos en la cara transformada $W(F)$ están más juntos de lo que están en F , como efecto de su cualidad de transformación contractiva, ya que siempre mueve los puntos que conforman a una imagen de manera en que los acerca.

Otro ejemplo de una transformación afín contractiva se muestra en la figura 4.5 Esta vez se efectúa sobre una hoja para producir una hoja más pequeña.

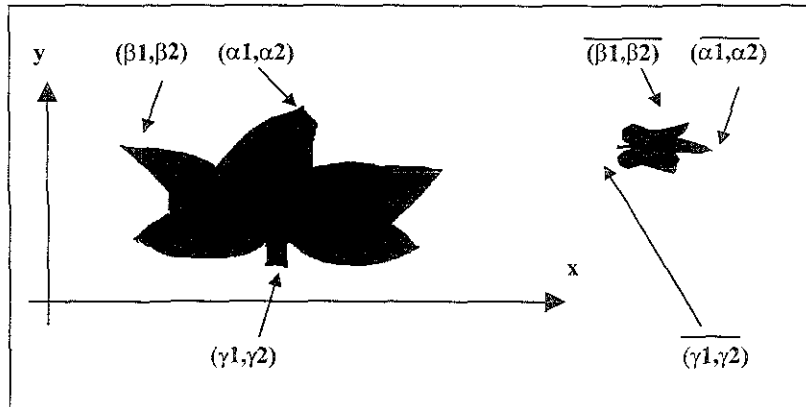


Figura 4.5. Ilustración de la transformación afín contractiva sobre la figura de una hoja (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

La forma general de una transformación afín es:

$$W \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} ax + by + e \\ cx + dy + f \end{bmatrix}$$

donde los coeficientes a , b , c , d , e y f son números reales.

Si conocemos de antemano las traslaciones, rotaciones y escalados que se combinan para producir W , es posible generar valores de coeficientes de la forma siguiente:

$$\begin{aligned} a &= r \cos \theta, & b &= -s \sin \phi \\ c &= r \sin \theta, & d &= s \cos \phi \end{aligned}$$

donde r es el factor de escalado en x , s es el factor de escalado en y , θ es el ángulo de rotación en x , ϕ es el ángulo de rotación en y , e es la traslación en x , y f es la traslación en y .

Para encontrar la transformación afín que convierta a la hoja grande en la pequeña es necesario despejar los valores de los coeficientes afines a , b , c , d , e y f para los cuales la transformación W tiene la propiedad:

$$W(\text{hoja grande}) \approx \text{hoja pequeña.}$$

Puede comenzarse considerando el plano bidimensional descrito por los ejes x e y , como se ve en la figura 4.5. La hoja grande tiene tres puntos de coordenadas (α_1, α_2) , (β_1, β_2) y (γ_1, γ_2) . Se han marcado también los puntos correspondientes en la hoja pequeña, también con sus coordenadas determinadas $(\overline{\alpha_1}, \overline{\alpha_2})$, $(\overline{\beta_1}, \overline{\beta_2})$ y $(\overline{\gamma_1}, \overline{\gamma_2})$, respectivamente.

Es necesario determinar los valores para los coeficientes a , b y e por medio de la resolución de las tres ecuaciones lineales siguientes:

$$\alpha_1 a + \alpha_2 b + e = \overline{\alpha_1} \tag{1}$$

$$\beta_1 a + \beta_2 b + e = \overline{\beta_1} \tag{2}$$

$$\gamma_1 a + \gamma_2 b + e = \overline{\gamma_1} \tag{3}$$

y los valores para c , d y f en forma similar:

$$\alpha_1 c + \alpha_2 d + f = \overline{\alpha_2} \tag{4}$$

$$\beta_1 c + \beta_2 d + f = \overline{\beta_2} \tag{5}$$

$$\gamma_1 c + \gamma_2 d + f = \overline{\gamma_2} \tag{6}$$

Ahora que se ha descrito a la transformación afin contractiva y cómo determinar alguna que mapee una imagen fuente hacia una imagen objetivo, es posible describir un sistema de funciones iteradas. Un IFS es una colección de transformaciones afines contractivas. En la figura 4.6 se muestra un ejemplo de un IFS de tres transformaciones.

$$\begin{array}{l}
 W_1 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} .5 & 0 \\ 0 & .5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
 W_2 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} .5 & 0 \\ 0 & .5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
 W_3 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} .5 & 0 \\ 0 & .5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} .25 \\ .5 \end{bmatrix}
 \end{array}$$

Figura 4.6. Ejemplo de IFS de tres transformaciones afines (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

Cada transformación debe tener también una probabilidad asociada p_i , que determine su importancia con respecto a las otras transformaciones. En este caso, pueden establecerse las probabilidades p_1 , p_2 y p_3 , las cuales deben sumar 1.

La tabla 4.1 expresa la misma información del IFS de la figura 4.6 en forma tabular. En las tablas 4.2 y 4.3 se presentan otros dos ejemplos de códigos IFS. Nótese que un IFS puede contener cualquier cantidad de transformaciones afines.

W	A	b	c	D	e	F	p
1	.5	0	0	.5	0	0	.33
2	.5	0	0	.5	1	0	.33
3	.5	0	0	.5	.5	.5	.34

Tabla 4.1. Códigos IFS para un triángulo de Sierpinski (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

W	A	b	c	D	e	f	P
1	.5	0	0	.5	0	0	.25
2	.5	0	0	.5	.5	0	.25
3	.5	0	0	.5	0	.5	.25
4	.5	0	0	.5	.5	.5	.25

Tabla 4.2. Códigos IFS para un cuadrado (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

W	A	B	c	d	e	F	p
1	0	0	0	.16	0	0	.01
2	.2	-.26	.23	.22	0	1.6	.07
3	-.15	.28	.26	.24	0	.44	.07
4	.85	.04	-.04	.85	0	1.6	.85

Tabla 4.3. Códigos IFS para un helecho (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

4.3.2. Algoritmo de iteración aleatorio.

Una vez que se han obtenido, almacenado y/o transmitido los coeficientes para las transformadas afines que describen a la imagen original, el siguiente paso es decodificar el código IFS por medio de un método denominado método de iteración aleatorio. Considerando que un IFS puede contener cualquier cantidad m de transformaciones afines $W_1, W_2, W_3, \dots, W_m$, cada cual con una probabilidad asociada, es posible ilustrar este método mediante el siguiente algoritmo:

- 1) Inicialice: $x = 0, y = 0$.
- 2) Para $n = 1$ hasta 2500, hacer los pasos (3-7).
- 3) Seleccione k para tener un valor entre 1 y m , con probabilidad p_k .
- 4) Aplique la transformación W_k al punto (x, y) para obtener (x', y') .
- 5) Establezca (x, y) para ser igual al nuevo punto : $x = x'$ e $y = y'$.
- 6) Si $n > 10$, trace el punto (x, y) .
- 7) Regrese.

Aplicando este procedimiento a las transformaciones de la tabla 4.1 se produce una imagen como la de la figura 4.7.

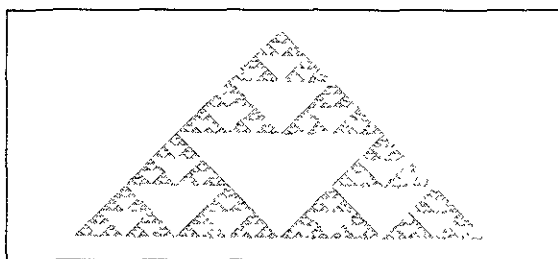


Figura 4.7. Resultado de aplicar el algoritmo de iteración aleatorio al código IFS de la tabla 4.1. Esta figura se conoce como el triángulo de Sierpinski (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

Incrementando el número de iteraciones n en el algoritmo de iteración aleatorio se agregan nuevos puntos a la imagen. La figura 4.8 muestra varias etapas del proceso y el resultado del algoritmo aplicado a los coeficientes de la tabla 4.3.

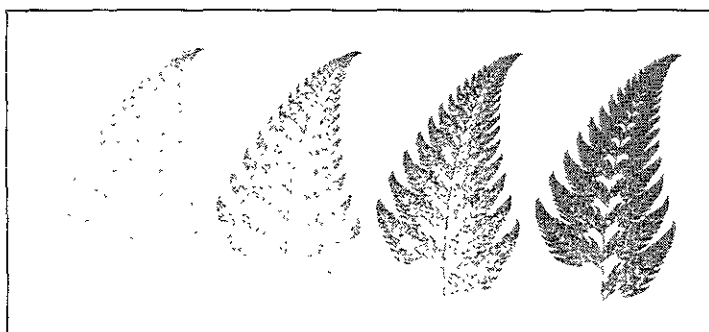


Figura 4.8. Imagen de un helecho generada al aplicar el algoritmo de iteración aleatoria a los códigos IFS de la tabla 4.3 (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

Incrementando el factor de escala utilizado en la decodificación de la imagen, puede hacerse un acercamiento a ella, como se muestra en la figura 4.9.

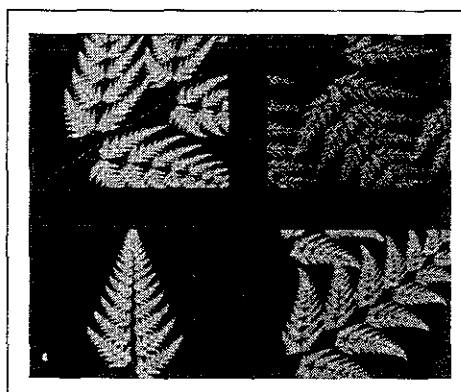


Figura 4.9. Acercamientos sucesivos sobre partes de un helecho codificado con IFS (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

La trascendencia del algoritmo de iteración aleatorio radica en la seguridad de que se producirá una imagen idéntica a la original, independientemente de la secuencia particular de elecciones aleatorias de las transformaciones afines que se hayan hecho. Esta certeza la proporciona el teorema del collage, el cual es resultado de experimentos matemáticos gráfico-computacionales y de un fundamento teórico desarrollado por el matemático John Elton.

4.3.3. El teorema del collage.

A continuación se expondrá un método sistemático para encontrar las transformaciones afines necesarias para producir un conjunto de códigos IFS que representen una imagen original, utilizando el denominado teorema del Collage.

Para ilustrar este método, considérese una imagen original de un cuadro relleno S en el plano x,y , con sus vértices en las coordenadas $(0,0)$, $(1,0)$, $(1,1)$ y $(0,1)$. El objetivo es seleccionar un conjunto de transformaciones afines contractivas, en este caso w_1 , w_2 , w_3 y w_4 , de forma tal que S sea aproximado tanto como sea posible por la unión de las cuatro subimágenes $w_1(S) \cup w_2(S) \cup w_3(S) \cup w_4(S)$. Una ilustración de este ejemplo se muestra en la figura 4.10.

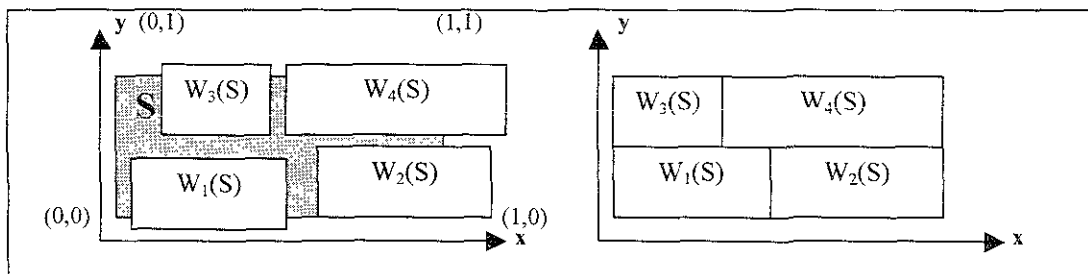


Figura 4.10. Se usa el teorema del Collage para codificar un cuadro S . Se obtiene el código IFS correcto cuando las cuatro transformaciones afines de S cubren a S (BARNESLEY, Michael F., SLOAN, Alan D. "A better way to compress images").

La figura 4.10 muestra a la izquierda, a S junto con las cuatro transformaciones afines de S que no lo cubren; a la derecha, las transformaciones afines se han ajustado para que al unir las imágenes de las transformaciones, éstas cubran todo el cuadro.

Para encontrar los coeficientes de estas transformaciones se utiliza el método descrito anteriormente por medio de las ecuaciones (1) a (3) y (4) a (6). Los valores encontrados para este ejemplo se ubican en la tabla 4.2. Cuando se aplica el algoritmo de iteración aleatoria sobre dichos coeficiente se obtiene un cuadro reconstruido.

El ejemplo anterior describe el proceso de codificación fractal en general, donde se necesita encontrar un conjunto de transformaciones afines que reduzca al mínimo la diferencia entre imagen original y reconstruida, misma que se genera a partir de la unión de las transformaciones afines de la imagen. El teorema del Collage dice que entre mayor exactitud se emplee para describir a la imagen original por medio de transformaciones afines de ésta, dichas transformaciones proporcionarán una codificación IFS más exacta de la imagen original. La figura 4.11 muestra una ilustración del teorema del Collage.



Figura 4.11. El teorema del collage aplicado a la generación de la imagen de una hoja (BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

En la figura 4.11, en la parte inferior izquierda, se muestran los bordes de una hoja poligonalizada con cuatro transformaciones afines de esos bordes. Las hojas transformadas unidas no forman una buena aproximación de la hoja. En consecuencia, la imagen IFS correspondiente (abajo derecha), generada por medio del algoritmo de iteración aleatoria no luce como una hoja real. Sin embargo, conforme el collage se hace más exacto (arriba izquierda) la imagen decodificada (arriba derecha) se hace más exacta igualmente.

Por medio de este ejemplo se muestra que no tiene que conseguirse necesariamente el código IFS exacto para lograr una buena réplica de la imagen original. Por el contrario, el código IFS se considera una técnica de codificación robusta, ya que si se presentan pequeñas perturbaciones en el código, éstas no resultarán en un daño notorio en la imagen descomprimida. En los ejemplos anteriores se han usado cuatro transformaciones para codificar la imagen, sin embargo, puede usarse cualquier cantidad de ellas.

4.3.4. Asignación de probabilidades.

Una vez que se han definido las transformaciones, es necesario asignarles un factor de probabilidad a cada una con el fin de determinar la importancia de dicha transformación

para el proceso de reconstrucción de la imagen original. En general, la elección de diferentes probabilidades no produce imágenes diferentes, pero si afecta la proporción a la que se “llenan” las regiones o se reproducen los atributos de la imagen. Si las transformaciones afines w_i correspondientes a una imagen I son:

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

donde $i = 1, 2, 3, \dots, n$, entonces, la cantidad de tiempo que el algoritmo de iteración aleatorio invertirá en la subimagen w_i es aproximadamente igual a:

$$\frac{\text{área de } w_i}{\text{área de } I}$$

Así que mientras $ad - cb$ no sea cero, el resultado de la proporción anterior será igual al determinante de la matriz de transformación para w_i . Por lo anterior, una elección adecuada para determinar la probabilidad p_i es:

$$\frac{a_i d_i - c_i b_i}{\sum_{k=1}^n |a_k d_k - c_k b_k|}$$

ya que ninguno de estos resultados p_i será igual a cero. Si se presenta un valor de cero, deberá reemplazarse con un valor positivo muy pequeño como 0.001 y las otras probabilidades deben ajustarse para mantener la suma de todas las probabilidades igual a 1.

4.3.5. Descripción de la técnica.

Por las definiciones anteriores es posible hacer una breve descripción de compresión y descompresión por sistema de funciones iteradas. Una imagen de entrada se descompone en segmentos a través de técnicas de procesamiento de imágenes. Estos componentes de imagen son referenciados en la librería IFS usando el teorema del collage, y sus códigos IFS son grabados en ella. Cuando la imagen va a reconstruirse, se introducen los códigos IFS al algoritmo de iteración aleatoria. La exactitud de la imagen reconstruida depende solamente de la configuración para la tolerancia asignada durante la etapa demapeo del collage. Las figuras 4.12, 4.13, 4.14 y 4.15 se obtuvieron utilizando la técnica de codificación de IFS a proporciones de compresión de 10,000 a 1. En ellas se indican el número de transformaciones afines utilizadas y el tamaño comprimido obtenido.

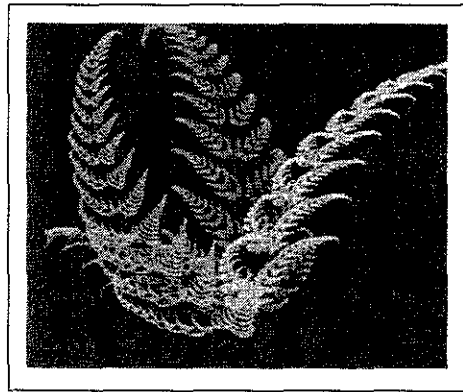


Figura 4.12. Imagen a color codificada con IFS de helechos tridimensionales (4 transformaciones, 100 bytes. BARNSELEY, Michael F., SLOAN, Alan D. "A better way to compress images").

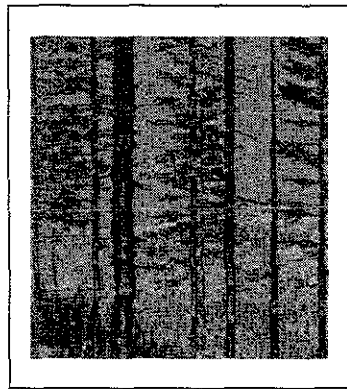


Figura 4.13. Fotografía a color codificada con IFS de un bosque oscuro (120 transformaciones, 2000 bytes. BARNSELEY, Michael F., SLOAN, Alan D. "A better way to compress images").



Figura 4.14. Fotografía a color codificada con IFS de una chica boliviana (120 transformaciones, 2000 bytes. BARNSELEY, Michael F., SLOAN, Alan D. "A better way to compress images").

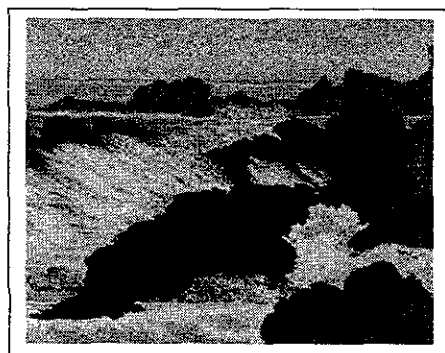


Figura 4.15. Fotografía a color codificada con IFS de una costa rocosa (60 transformaciones, 500 bytes. BARNSELY, Michael F., SLOAN, Alan D. "A better way to compress images").

La técnica de compresión IFS exige un alto rendimiento computacional, tanto en la fase de codificación como de decodificación. Para aplicaciones gráficas, muchas veces se necesita hardware especializado que pueda acelerar el proceso de codificación y decodificación.

4.4. COMPRESIÓN BASADA EN TRANSFORMADA FRACTAL.

El objetivo de un algoritmo de compresión fractal es realizar la búsqueda de un conjunto de mapeos de una imagen original. Al Comenzar el proceso de iteración en cualquier parte de la imagen inicial, estos mapeos poseen un punto o fractal fijo que se aproxima a la imagen a codificar. En la práctica, no es trivial encontrar dicho conjunto de mapeos.

El sistema de codificación basado en IFS funciona bien para imágenes simples cuyos elementos puedan fragmentarse fácilmente, pero en el caso de imágenes del mundo real, éstas también tendrían que seguir el proceso de codificación donde se modela a la imagen como una colección de segmentos fractales, buscando el mejor conjunto de transformaciones afines para cada uno. Antes de que la compresión fractal pudiera utilizarse comercialmente, se requería de un método que pudiera efectuar automáticamente dicho proceso en una cantidad de tiempo razonable y con resultados exactos y predecibles.

4.4.1. Codificación de transformada fractal.

Considerando este problema, Barnsley hizo la observación de que todas las imágenes del mundo real tienen redundancia afin. Esto significa que, bajo transformaciones afines convenientes, las grandes cantidades de bits necesarias para representar las imágenes del mundo real, pueden reducirse en una cantidad promedio. Esta observación, junto con el

fundamento matemático del teorema del collage, lo llevó a la invención del proceso de transformación fractal para compresión de imágenes. El esquema de la figura 4.16 ilustra el algoritmo de este proceso.

El primer paso en el proceso de compresión por transformada fractal es particionar la imagen en regiones de dominio no superpuestas. Al reunir las, el conjunto de regiones de dominio debe cubrir la imagen completamente, pero pueden ser de cualquier tamaño o forma. Posteriormente, el algoritmo define una colección de regiones de rango posibles, que deben ser mayores en tamaño que las regiones de dominio, las cuales se pueden superponer y no necesitan cubrir la imagen completamente.

Para cada región de dominio, el programa debe escoger la región de rango que, después de aplicar una transformación afín apropiada, se acople lo mejor posible a la región de dominio. Las transformaciones afines no sólo escogen y deforman la imagen dentro de la región de rango, también reducen el contraste y cambian el brillo y su intensidad. Posteriormente, cada transformación afín puede describirse por medio de sus coeficientes afines.

El siguiente ejemplo ilustra el funcionamiento del algoritmo descrito: Supóngase una imagen de 256×256 píxeles de tamaño con una precisión por píxel de 8 bits. Para codificar esta imagen, primero se particiona en 1024 rangos no superpuestos de 8×8 . Dentro de la imagen hay una cantidad total de $(256-15) \times (256-15)$ bloques posibles, denominados dominios, de tamaño 16×16 que pueden superponerse. Entonces, para cada rango, se busca en todos los dominios hasta encontrar un dominio que se parezca a la imagen rango. Esto se realiza minimizando la diferencia entre el rango y el resultado del mapeo sobre todo el dominio.

A través de este ejemplo, puede verse que la complejidad de la compresión fractal es importante. Esto se debe a que hay 8 orientaciones en cada dominio de 16×16 . Esto significa que hay 8 formas de mapear un cuadro en otro. Para un total de $(256-15) \times (256-15)$ dominios, hay que realizar $8 \times (256-15) \times (256-15) = 464,648$ comparaciones entre cuadros. Además se requiere de una operación de escalado de un dominio de 16×16 en un rango de 8×8 . Estos números reflejan el costo del proceso de minimización de distorsión para codificar la imagen.

Finalmente, con todos los datos generados, se escribe un archivo FIF (Formato de Imagen Fractal). Este consiste en un encabezado con información acerca de la selección específica de las regiones de dominio, seguido de la lista empacada de coeficientes afines seleccionados para cada región de dominio. Este proceso genera un archivo que es independiente de la resolución de la imagen original. Es decir, se ha obtenido una "ecuación" para la imagen. En el caso de una línea recta que puede representarse por la

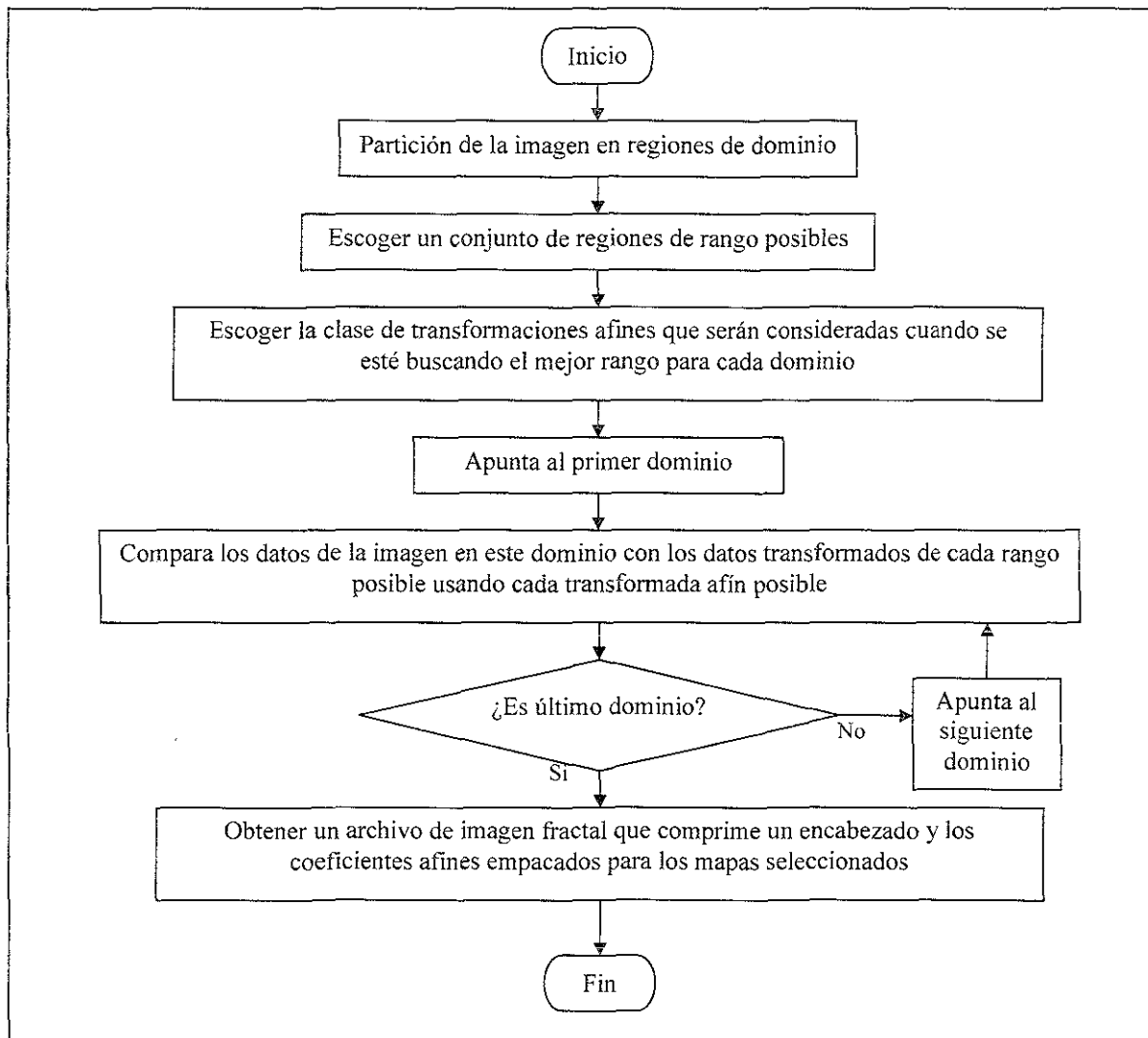


Figura 4.16. El algoritmo para compresión por transformada fractal de imagen.

ecuación $y = ax + b$, si se conocen los valores de los coeficientes a y b es posible dibujar dicha línea a cualquier resolución.

Las aplicaciones comerciales de la transformada fractal tienen algunos problemas al seleccionar las regiones de dominio, de rango y las transformaciones a utilizar. Entre más grandes sean las regiones de dominio, el número de transformaciones necesarias para modelar la imagen será menor, y por lo tanto el archivo fractal será más pequeño. Sin embargo, si no se logra un acoplamiento lo suficientemente bueno entre cada una de las regiones de dominio y una región de rango transformada, la apariencia de la imagen comprimida al recuperarse se deteriora.

El compresor considera regiones de dominio de varios tamaños para encontrar la mejor región de rango para cada una en el menor tiempo posible, y utiliza un proceso matemático para evaluar el conjunto óptimo de regiones de dominio para el tamaño de archivo deseado.

Para mantener el tiempo de compresión en un nivel razonable, es necesario incluir límites prácticos en la colección de regiones de rango posibles y en las transformaciones permitidas. Por ejemplo, en el sistema Poem Color de Iterated Systems, el compresor tiene cuatro posibles modelos que controlan el tiempo permitido para buscar la mejor región de rango para cada dominio. Cuando trabaja en los modos de búsqueda más exacta, los cuales toman más tiempo, es posible extender la clase de transformaciones y el tamaño de regiones de rango posibles para lograr una imagen de mejor resolución en el mismo tamaño de archivo comprimido.

4.4.2. Decodificación de transformada fractal.

Considerando los coeficientes afines contenidos en el archivo FIF, el proceso de descompresión puede crear una réplica fractal que luzca como la imagen original a cualquier resolución. El proceso de descompresión comienza cuando se asigna espacio de memoria en el equipo donde se realizará la descompresión para dos imágenes de igual tamaño A y B. El tamaño de estas imágenes puede ser menor o mayor que el de la imagen original antes de la compresión, y el contenido inicial es irrelevante. Pueden ser datos o una imagen. El diagrama de la figura 4.17 muestra los pasos del proceso de descompresión de transformada fractal.

Para la primera iteración del proceso de descompresión, hay que referirse a la imagen A como la imagen rango y a la imagen B como la imagen dominio. Se particiona la imagen dominio en regiones de dominio especificadas en el encabezado del archivo FIF. Para cada región de dominio en la imagen dominio, se leen los coeficientes afines para este dominio del archivo FIF. Se localiza la región de rango especificada por esta transformación afín en la imagen rango, y se mapea el contenido de esta región de rango a partir de la imagen rango.

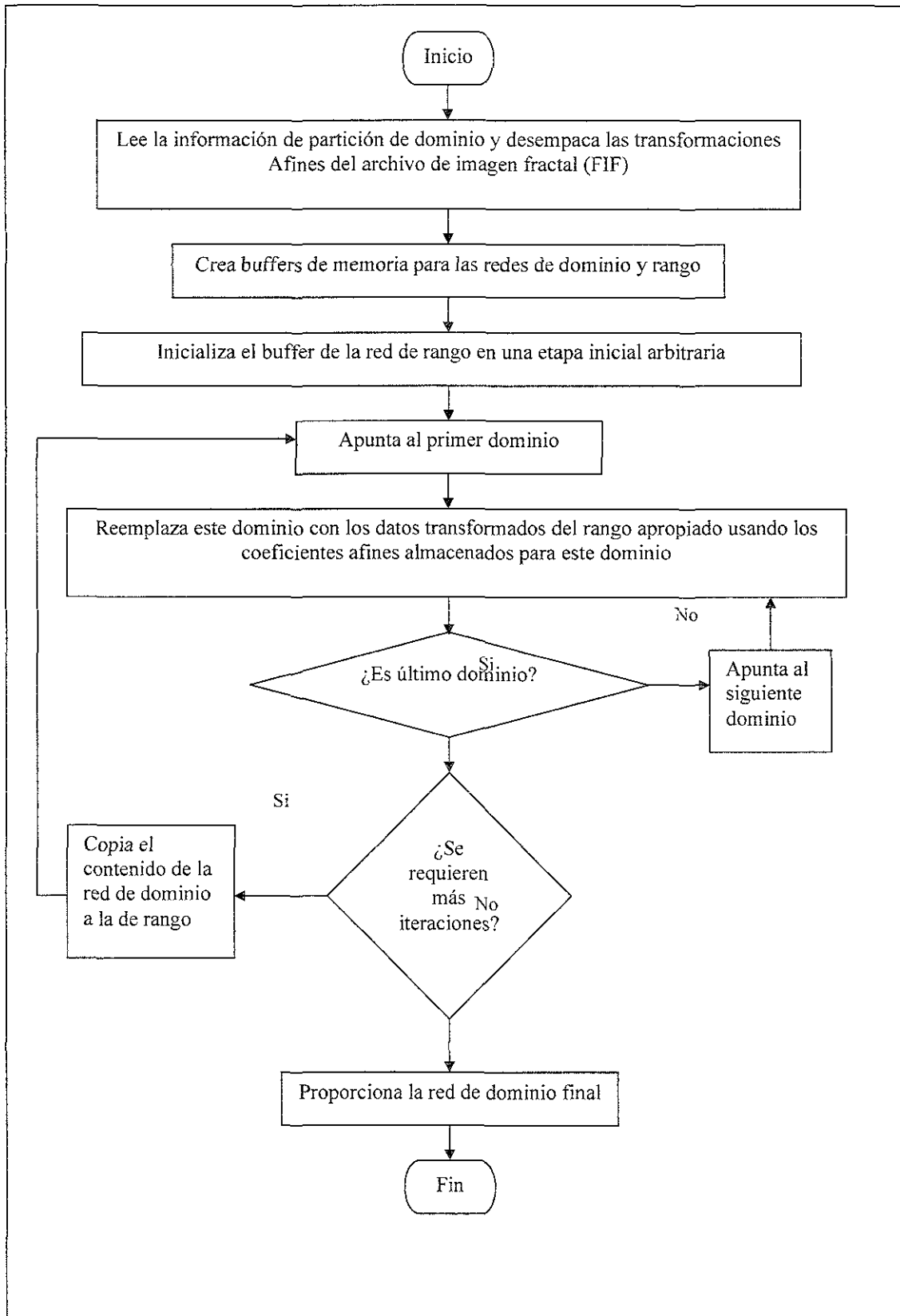


Figura 4.17. El algoritmo para descomprimir una imagen almacenada en la forma de datos de transformada fractal.

4.5. RESUMEN DE LA COMPRESIÓN FRACTAL.

En general, puede decirse que la compresión fractal es diferente de los métodos de compresión tradicionales, y es una nueva aproximación para la compresión de imágenes, sin embargo, aún está a debate su utilidad. La ventaja de la compresión fractal es que ofrece altas proporciones de compresión para imágenes que contengan dibujo. Las desventajas son que el proceso de codificación es muy caro y no hay suficientes pruebas de que sea buena para otro tipo de imágenes como imágenes de documentos, médicas o espaciales. El hecho de que la operación de compresión sea de un costocomputacional tan alto la hace prohibitiva para aplicaciones simétricas como las comunicaciones basadas en imágenes.

Cuando se observa a la compresión fractal desde el punto de vista de la tecnología de compresión tradicional, la compresión fractal puede verse como un sistema de cuantificación de vector que toma a la imagen completa como su librería y cuyas palabras de código son partes de la imagen después de algunas operaciones geométricas.

En la tabla 4.4 se muestra un cuadro comparativo de las técnicas de compresión fractal, siguiendo los criterios utilizados en los mismos cuadros para las técnicas de capítulos anteriores. En ella se destaca que su principal diferencia radica en el área de selección de información inicial que manejan, por motivos de carga de trabajo computacional.

Técnica de compresión	Herramienta de selección de información	Área de selección inicial	¿Iterativa con respecto al área de selección?	¿Redefinición del tamaño para área de selección en cada iteración?	¿Redefinición de parametros para la herramienta en cada iteración?	¿Consideración del contenido particular del área de selección?
Funciones Iteradas	Transformaciones afines	Región de píxeles	Sí	No	Sí	Sí
Transformada Fractal	Transformaciones afines	Rango de píxeles	Sí	No	Sí	Sí

Tabla 4.4. Cuadro comparativo de las técnicas de compresión fractal.

CAPÍTULO V:

ESTÁNDARES COMERCIALES Y ÁREAS DE APLICACIÓN DE LA COMPRESIÓN.

5.1. ESTÁNDARES DE COMPRESIÓN COMERCIALES.

La tecnología de compresión es una herramienta que permite un mejor aprovechamiento de los recursos de comunicación y procesamiento disponibles, al convertir la información digital en una representación que requiere menor espacio de almacenamiento y menor tiempo de transmisión. Ambos factores son determinantes para aplicaciones dirigidas a un gran número de usuarios localizados en sitios cercanos o remotos accedendo información de forma simultánea. El tiempo de respuesta, es decir, el que la imagen requiere para desplegarse en pantalla, es un parámetro para determinar si el método usado para comprimir la imagen resulta adecuado o no. Como ha podido apreciarse a lo largo del trabajo, existen muchas aproximaciones para el tratamiento de una imagen digital con el fin de comprimir sus datos sin perder información significativa para su representación descomprimida. Sin embargo, debido al crecimiento de las redes de información a nivel mundial y a la proliferación de aplicaciones que manejan imágenes y gráficos de mayores resoluciones, se hace indispensable el establecimiento de métodos internacionales, denominados estándares, que sustenten prácticas y técnicas de compresión que puedan interpretarse en cualquier parte del mundo. Dichos estándares son diseñados por organizaciones internacionales que tienen como propósito promover una comunicación abierta entre plataformas de diversos fabricantes y hacer disponible la información codificada bajo dichos métodos para un gran número de personas. Los estándares buscan un equilibrio entre la proporción de la compresión, la apariencia de la imagen descomprimida, las facilidades de transmisión y de implementación. En este sentido, se han generado diversos estándares que si bien no utilizan en su mayoría las tecnologías más modernas de representación de las imágenes, si son muy populares debido precisamente a su simplicidad de diseño y a su presencia en las aplicaciones gráficas comerciales.

5.2. ESTÁNDARES DE COMPRESIÓN T.4 DEL CCITT.

El estándar CCITT T.4 fue desarrollado por el Comité Consultivo de Telefonía y Telegrafía (CCITT) en 1980 para la transmisión de facsímile analógico y corregido en 1984 y 1988 para transmisión de facsímile digital sobre la red telefónica pública (PSTN) y para circuitos para arrendamiento internacional. El objetivo de este estándar era el de poder transmitir documentos bitonales hasta de un tamaño de A4 explorados a una resolución de 100 puntos por pulgada (DPI) o a 3.85 líneas por milímetro y muestreados a 1728 muestras por línea, y transmitidos a una velocidad de 4800 bits por segundo en un tiempo promedio de un minuto para todo el documento a través de la PSTN. Este estándar hizo popular el facsímile digital comprimido con este estándar denominado Grupo 3. Los esquemas de codificación especificados en el estándar CCITT T.4 se conocen como los algoritmos de compresión Grupo 3 del CCITT. Existen dos esquemas de codificación en el CCITT T.4, el esquema unidimensional CCITT Grupo 3 y el esquema bidimensional CCITT Grupo 3.

5.2.1. Estándar T.4 unidimensional.

Como ejemplo para ilustrar la forma de trabajo del método de codificación unidimensional CCITT Grupo 3, puede examinarse una línea de una imagen binivel como la que se muestra en la figura 5.1.

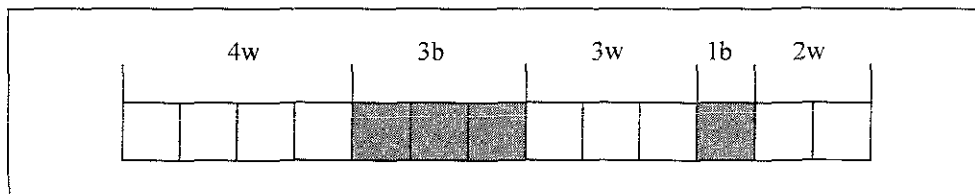


Figura 5.1. Una línea de una imagen binivel (w equivale a muestra de color blanco y b equivale a muestra de color negro. KOU, Weidong. Digital image compression. Algorithms and standards).

En este ejemplo se tiene un total de 13 píxeles en una línea de una imagen binivel. La secuencia de píxeles es 4 blancos, 3 negros, 3 blancos, 1 negro y 2 blancos. Como las probabilidades de las secuencias son diferentes, puede utilizarse un código de longitud variable (VLC) para aprovechar esta característica y mejorar la eficiencia de la compresión.

5.2.1.1. Descripción del algoritmo.

El esquema de codificación unidimensional CCITT Grupo 3 utiliza un código de longitud variable denominado Código Huffman Modificado (MH). Cada línea de la imagen se codifica como una serie de palabras de código de longitud variable y cada palabra de

código representa una longitud de corrida ya sea de todas blancas o todas negras. El código Huffman modificado tiene dos tipos de palabras de código: las palabras de código terminales y las palabras de código de construcción. Las palabras de código terminales se usan para representar las longitudes de corridas en el rango de 0 a 63 píxeles. Cada palabra de código de construcción en conjunción con la palabra de código terminal se usa para codificar una longitud de corrida en el rango de 64 a 1728 píxeles. Para longitudes de corrida mayores a 1728, puede utilizarse un conjunto de palabras de código de construcción opcionales, con lo que se soportan longitudes de corridas en el rango de 1792 a 2560 píxeles. Siguiendo a cada línea de datos, existe una palabra de código de fin de línea (EOL). Esta palabra de código EOL también se requiere antes de las palabras de código de la primera línea de la imagen. Una longitud variable de una cadena de ceros puede insertarse entre un segmento de datos de línea y un EOL. Esta cadena cero se denomina Relleno (Fill). Seis EOL's consecutivos forman una palabra de código especial llamada regreso de control (RTC), indicando el final de la imagen. Los códigos terminales, de construcción y de construcción adicionales se listan en las tablas 5.1 a 5.3 respectivamente.

El esquema de codificación unidimensional CCITT Grupo 3 puede resumirse como un diagrama de flujo de codificación como el de la figura 5.2.

Longitud de Corrida Blanca	Palabra de Código	Longitud de Corrida Negra	Palabra de Código
0	00110101	0	000110111
1	000111	1	010
2	0111	2	11
3	1000	3	10
4	1011	4	011
5	1100	5	0011
6	1110	6	0010
7	1111	7	00011
8	10011	8	000101
9	10100	9	000100
10	00111	10	0000100
11	01000	11	0000101
12	001000	12	0000111
13	000011	13	00000100
14	110100	14	00000111
15	110101	15	000011000
16	101010	16	0000010111
17	101011	17	0000011000
18	0100111	18	0000001000
19	0001100	19	00001100111
20	0001000	20	00001101000
21	0010111	21	00001101100
22	0000011	22	00000110111
23	0000100	23	00000101000
24	0101000	24	000000010111
25	0101011	25	00000011000

26	0010011	26	00011001010
27	0100100	27	000011001011
28	0011000	28	000011001100
29	00000010	29	000011001101
30	00000011	30	000001101000
31	00011010	31	000001101001
32	00011011	32	000001101010
33	00010010	33	000001101011
34	00010011	34	000011010010
35	00010100	35	000011010011
36	00010101	36	000011010100
37	00010110	37	000011010101
38	00010111	38	000011010110
39	00101000	39	000011010111
40	00101001	40	000001101100
41	00101010	41	000001101101
42	00101011	42	000011011010
43	00101100	43	000011011011
44	00101101	44	000001010100
45	00000100	45	000001010101
46	00000101	46	000001010110
47	00001010	47	000001010111
48	0001011	48	000001100100
49	01010010	49	000001100101
50	01010011	50	000001010010
51	01010100	51	000001010011
52	01010101	52	000000100100
53	00100100	53	000000110111
54	00100101	54	000000111000
55	01011000	55	000000100111
56	01011001	56	000000101000
57	01011010	57	000001011000
58	01011011	58	000001011001
59	01001010	59	000000101011
60	01001011	60	000000101100
61	00110010	61	000001011010
62	00110011	62	000001100110
63	00110100	63	000001100111

Tabla 5.1. Códigos terminales (KOU, Weidong, Digital image compression. Algorithms and standards).

Longitud de Corrida Blanca	Palabra de Código	Longitud de Corrida Negra	Palabra de Código
64	11011	64	0000001111
128	10010	128	000011001000
192	010111	192	000011001001
256	0110111	256	000001011011
320	00110110	320	000000110011
384	00110111	384	000000110100
448	01100100	448	000000110101
512	01100101	512	0000001101100
576	01101000	576	0000001101101
640	01100111	640	0000001001010
704	011001100	704	0000001001011
768	011001101	768	0000001001100
832	011010010	832	0000001001101
896	011010011	896	0000001110010

960	011010100	960	0000001110011
1024	011010101	1024	0000001110100
1088	011010110	1088	0000001110101
1152	011010111	1152	0000001110110
1216	011011000	1216	0000001110111
1280	011011001	1280	0000001010010
1344	011011010	1344	0000001010011
1408	011011011	1408	0000001010100
1472	010011000	1472	0000001010101
1536	010011001	1536	0000001011010
1600	010011010	1600	0000001011011
1664	011000	1664	0000001100100
1728	010011011	1728	0000001100101
EOL	000000000001	EOL	000000000001

Tabla 5.2. Códigos de construcción (KOU, Weidong. Digital image compression. Algorithms and standards).

Longitud de corrida (Blanca y Negra)	Códigos de Construcción
1792	00000001000
1856	00000001100
1920	00000001101
1984	000000010010
2048	000000010011
2112	000000010100
2176	000000010101
2240	000000010110
2304	000000010111
2368	000000011100
2432	000000011101
2496	000000011110
2560	000000011111

Tabla 5.3. Códigos de construcción adicionales (KOU, Weidong. Digital image compression. Algorithms and standards).

5.2.2. Estándar T.4 bidimensional.

La codificación bidimensional CCITT Grupo 3 utiliza el código Huffman modificado y el código de designación de dirección de elemento relativo modificado (MR). La ventaja principal de la codificación bidimensional CCITT Grupo 3 sobre la codificación unidimensional CCITT Grupo 3 es que se mejora la eficiencia de la compresión. Esto se debe al hecho de que la codificación bidimensional explora la correlación de píxeles entre dos líneas sucesivas en lugar de hacerlo solamente sobre la línea actual.

5.2.2.1. Descripción del algoritmo.

Para entender cómo trabaja la codificación bidimensional CCITT Grupo 3, se necesitan las definiciones de los términos de la tabla 5.4.

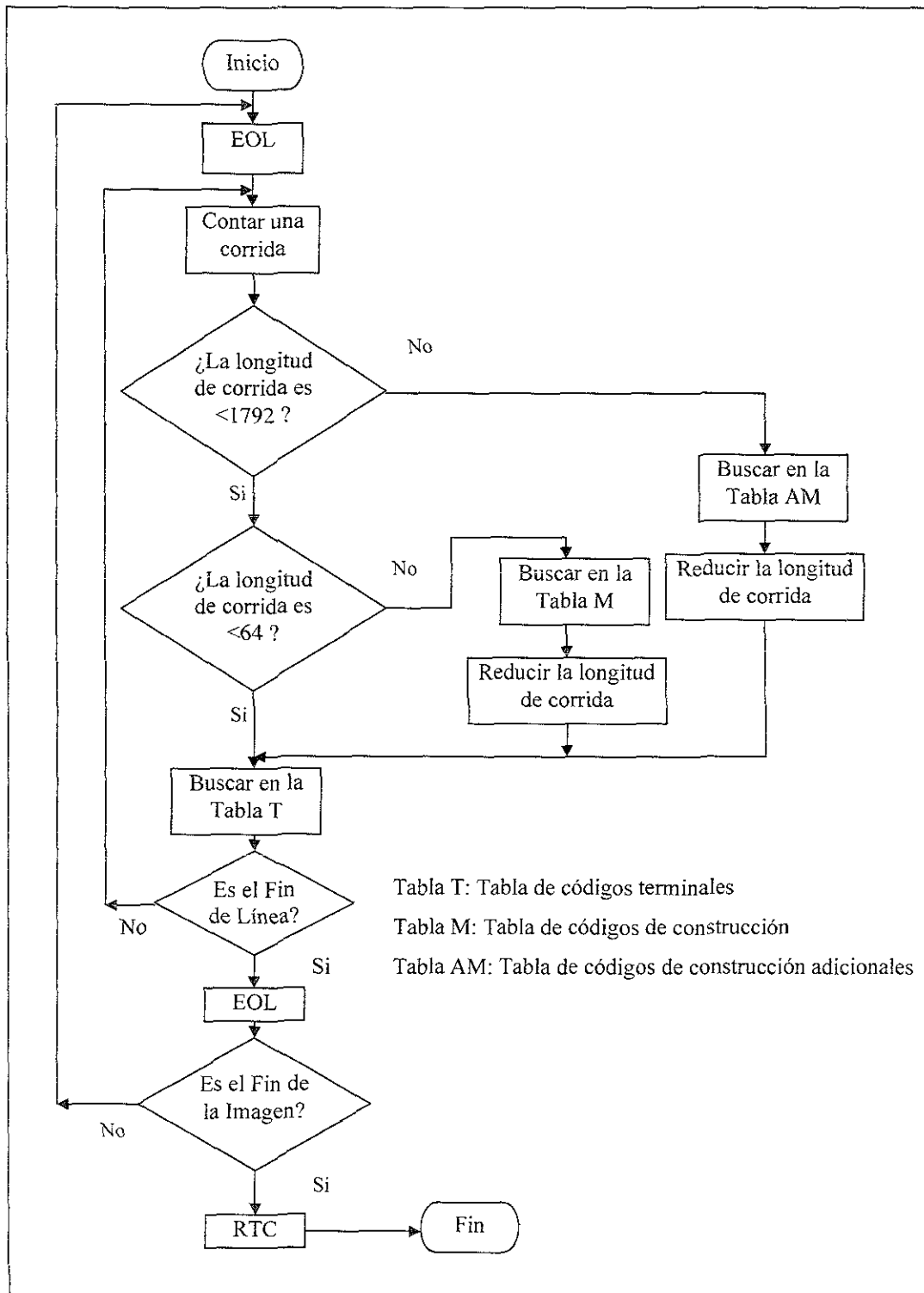


Figura 5.2. Diagrama de flujo de la codificación unidimensional CCITT Grupo 3.

Término	Definición
Color de Elemento	Negro o Blanco
Elemento Cambiante	Un elemento cuyo color es diferente del elemento previo en la misma línea explorada
Línea de Codificación	La línea explorada actual
Línea de referencia	La línea explorada previa
Modo de Código	Un método para codificar la posición de cada Elemento Cambiante en la Línea de Codificación
Elemento de referencia	Un elemento cuya posición determina un Modo de Código

Tabla 5.4. Términos utilizados el estándar CCITT T.4 bidimensional (KOU, Weidong. Digital image compression. Algorithms and standards)

La codificación bidimensional CCITT Grupo 3 es un método de codificación de línea por línea. La posición de cada elemento cambiante en la línea de codificación se codifica con respecto a la posición de un elemento de referencia correspondiente. El elemento de referencia se localiza ya sea en la línea de codificación o en la línea de referencia. Después de codificar la línea de codificación, ésta se convierte en la línea de referencia para la siguiente línea de codificación. Para limitar el error durante la transmisión del facsímile, se envía periódicamente una línea codificada con código Huffman modificado (es decir una línea codificada unidimensional). Este método se denomina Factor K. Para facsímile, a la resolución vertical estándar, K es igual a 2; a la resolución vertical más alta, k es igual a 4. Para imágenes digitales, K puede ser cualquier entero positivo diferente de cero. Para cualquier grupo de K líneas, el esquema bidimensional CCITT Grupo 3 codifica una línea con la codificación unidimensional y las otras K-1 líneas con la codificación bidimensional.

Elementos de cuadro cambiantes.

En la codificación bidimensional CCITT Grupo 3 existen cinco elementos de cuadro cambiantes como se lista en la tabla 5.5.

Elemento Cambiante	Definición
a_0	El elemento de referencia está en la línea de codificación.
a_1	El siguiente elemento cambiante está a la derecha de a_0 en la línea de codificación.
a_2	El siguiente elemento cambiante está a la derecha de a_1 en la línea de codificación.
b_1	El siguiente elemento cambiante está en la línea de referencia a la derecha de a_0 y es de color opuesto al de a_0 .
b_2	El siguiente elemento cambiante está a la derecha de b_1 en la línea de referencia.

Tabla 5.5. Lista de los Elementos de cuadro cambiantes (KOU, Weidong. Digital image compression. Algorithms and standards).

Al comenzar la línea de codificación, al elemento cambiante a_0 se le asigna primero un elemento cambiante blanco imaginario localizado justo antes del primer elemento de la línea de codificación. Durante la codificación de la línea de codificación, la posición de a_0 se determina por medio del modo de codificación anterior. Un ejemplo de estos cinco elementos cambiantes se muestra en la figura 5.3.

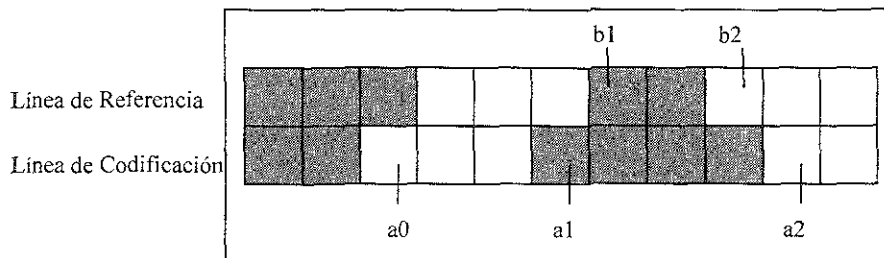


Figura 5.3. Un ejemplo de elementos de cuadro cambiantes (KOU, Weidong. Digital image compression. Algorithms and standards).

Modos de codificación.

En el estándar CCITT T.4 existen tres modos de codificación: Modo de Paso (P), Modo Vertical (V) y Modo Horizontal (H). Dependiendo de la posición de un elemento cambiante en la línea de codificación, puede identificarse uno de los tres modos de codificación, los cuales se listan en la tabla 5.6.

Modo de Codificación	Condición
Modo de Paso	Cuando la posición de b_2 lleve a la izquierda de a_1 .
Modo Vertical	Cuando la distancia relativa entre a_1 y b_1 sea menor o igual a 3.
Modo Horizontal	Cuando no ocurran ni el Modo de Paso ni el Modo Vertical.

Tabla 5.6. Modos de codificación del estándar CCITT T.4 bidimensional (KOU, Weidong. Digital image compression. Algorithms and standards).

En el modo vertical, dependiendo de la distancia relativa entre a_1 y b_1 pueden definirse siete casos posibles:

- $V(0)$: a_1 justo abajo de b_1 .
- $V_R(1)$: a_1 un pixel a la derecha de b_1 .
- $V_R(2)$: a_1 dos pixeles a la derecha de b_1 .
- $V_R(3)$: a_1 tres pixeles a la derecha de b_1 .
- $V_L(1)$: a_1 un pixel a la izquierda de b_1 .
- $V_L(2)$: a_1 dos pixeles a la izquierda de b_1 .
- $V_L(3)$: a_1 tres pixeles a la izquierda de b_1 .

Las palabras de código para los tres modos de codificación (de paso, vertical y horizontal) se muestran en la tabla 5.7 junto con la extensión opcional. En esta tabla $M(a_i a_j)$ representan las palabras de código de las tablas 5.1 a 5.3 para la corrida $a_i a_j$.

Modo	Elementos a Codificar		Palabra de Código
De Paso P	b_1, b_2		0001
Horizontal H	$a_0 a_1, a_1 a_2$		$001 + M(a_0 a_1) + M(a_1 a_2)$
Vertical:			
$V(0)$	A_1 justo bajo b_1	$a_1 b_1 = 0$	1
$V_R(1)$	a_1 a la derecha de b_1	$a_1 b_1 = 1$	011
$V_R(2)$		$a_1 b_1 = 2$	000011
$V_R(3)$		$a_1 b_1 = 3$	0000011
$V_L(1)$	a_1 a la izquierda de b_1	$a_1 b_1 = 1$	010
$V_L(2)$		$a_1 b_1 = 2$	000010
$V_L(3)$		$a_1 b_1 = 3$	0000010
Extensión	2-D (extensiones)		0000001xxx
	1-D (extensiones)		000000001xxx

Tabla 5.7. Tabla de Códigos bidimensionales (KOU, Weidong. Digital image compression. Algorithms and standards.)

Ejemplos de estos tres modos de codificación se muestran en las figuras 5.4 a 5.6.

El diagrama de flujo del procedimiento de codificación del esquema CCITT T.4 bidimensional se muestra en la figura 5.7.

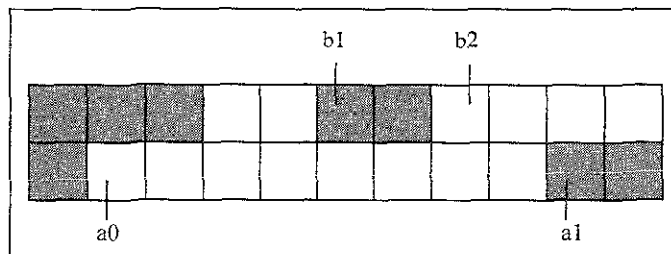


Figura 5.4. Ejemplo de modo de paso, P (KOU, Weidong. Digital image compression. Algorithms and standards.)

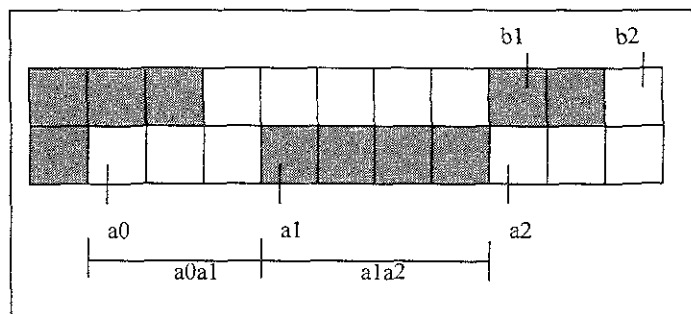


Figura 5.5. Ejemplo de modo horizontal, H (KOU, Weidong. Digital image compression. Algorithms and standards.)

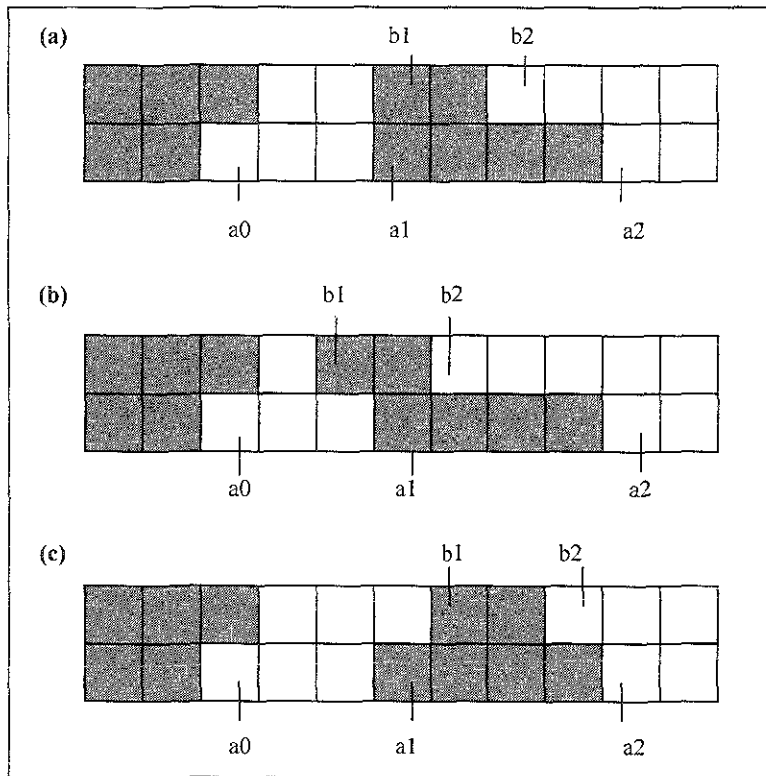


Figura 5.6. Ejemplos de modos verticales, (a) del $V(0)$, (b) del $VR(1)$ y (c) del $VL(1)$ (KOU, Weidong. Digital image compression. Algorithms and standards).

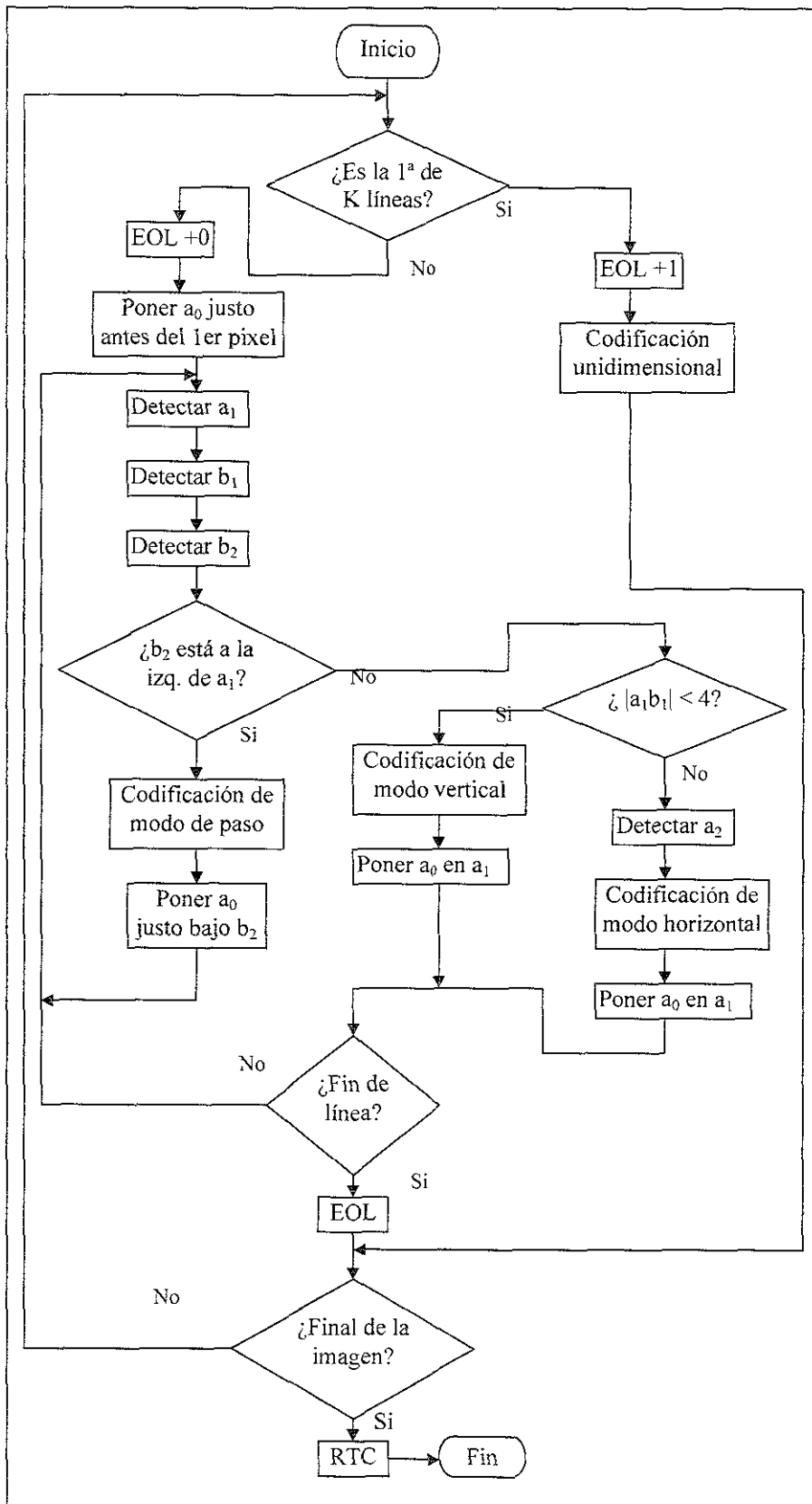


Figura 5.7. Diagrama de flujo de la codificación bidimensional CCITT T.4.

Algunos ejemplos de codificación del CCITT T.4 bidimensional se muestran en las figuras 5.8 y 5.9. En estos esquemas los pixeles marcados con puntos negros indican el elemento cambiante a codificar.

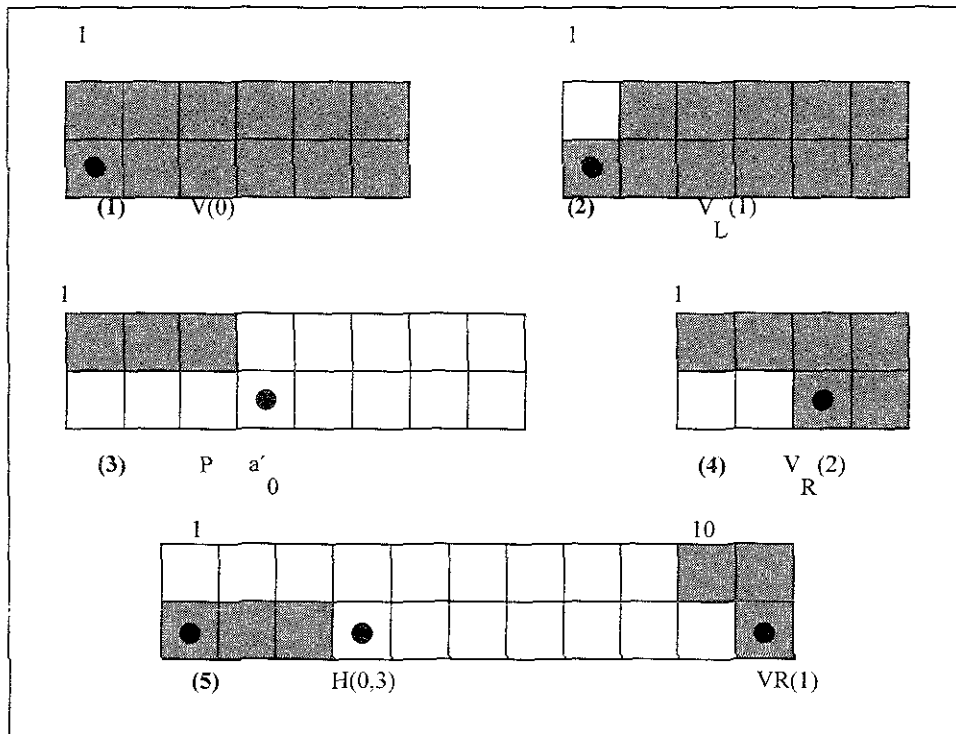


Figura 5.8. Ejemplos de codificación: primera parte de una línea explorada (KOU, Weidong. Digital image compression. Algorithms and standards).

Descompresión de datos.

El proceso de descompresión puede implementarse a través de la búsqueda de un árbol de código para código de asignación de dirección de elemento relativo (READ) modificado como se muestra en la figura 5.10.

5.3. ESTÁNDAR DE COMPRESIÓN T.6 DEL CCITT.

En 1984, CCITT desarrolló un estándar que define los esquemas de codificación de facsímile a utilizar en la transmisión de facsímile blanco y negro del Grupo 4. El estándar conocido como el estándar CCITT T.6, está diseñado para transmisión digital de facsímile libre de errores a través de redes de datos públicas. Esta especificación se ha utilizado ampliamente en sistemas de imágenes de documentos para comprimir imágenes blanco y negro. Se denomina también como el estándar de compresión CCITT Grupo 4.

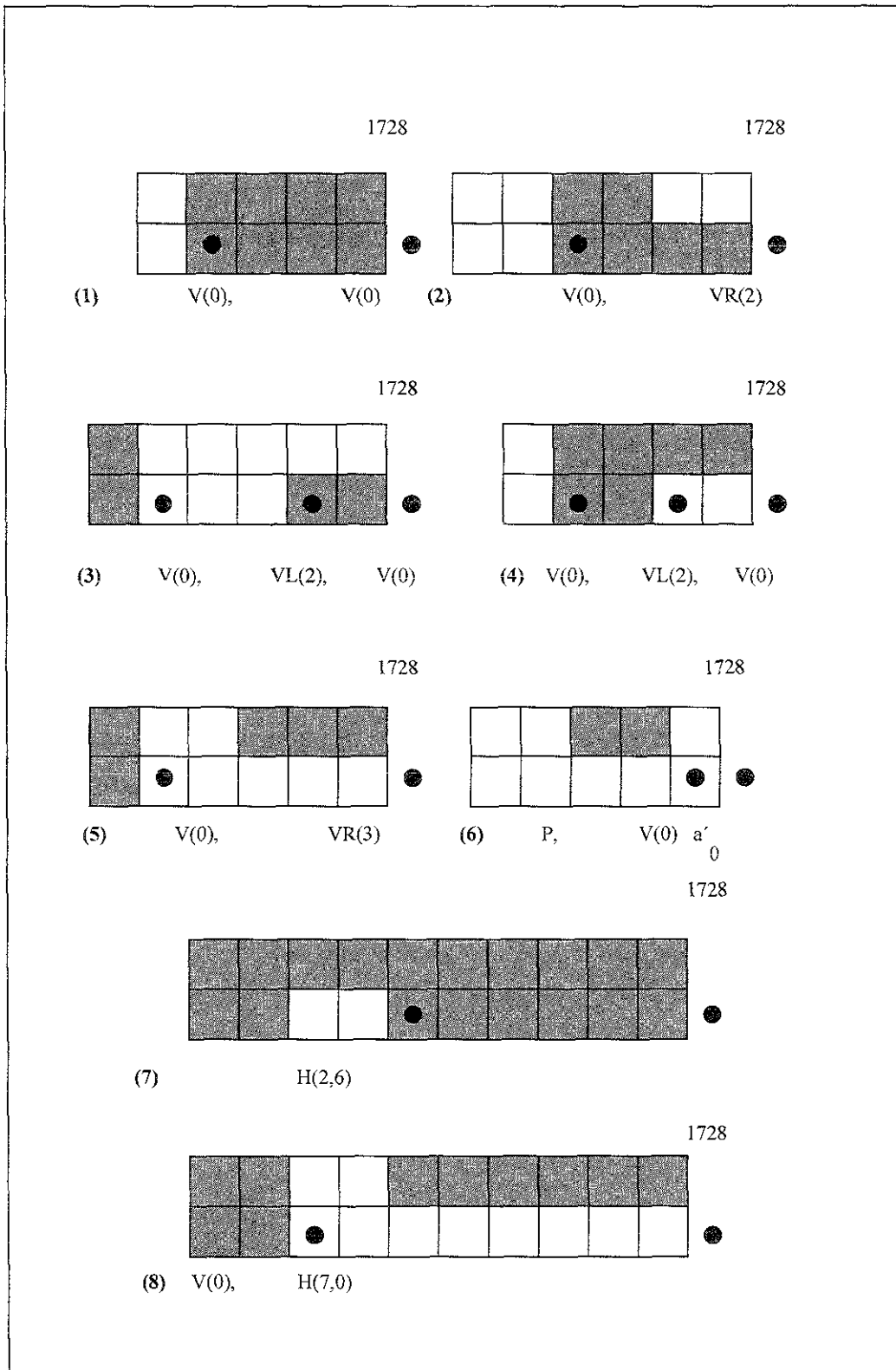


Figura 5.9. Ejemplos de codificación: última parte de una línea explorada (KOU, Weidong. Digital image compression. Algorithms and standards).

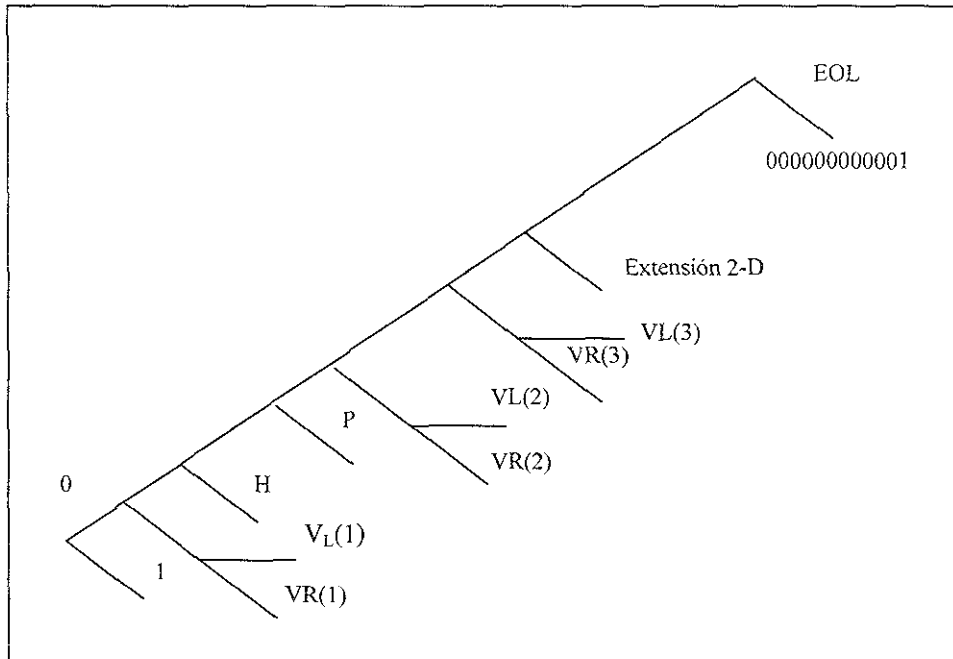


Figura 5.10. Árbol de código para Código READ modificado (KOU, Weidong. Digital image compression. Algorithms and standards).

5.3.1. Descripción del algoritmo.

El estándar de codificación de facsímile CCITT T.6 consiste en un esquema de codificación de facsímile básico y esquemas de codificación opcionales adicionales. Para este estándar se asume que los errores de transmisión se corrigen por medio de procedimientos de control en una etapa de procesamiento anterior a la compresión, por lo tanto, el CCITT T.6 es un esquema de codificación sin ruido.

Este esquema utiliza un código que se conoce como código de designación de dirección de elemento relativo modificado (MMR o READ modificado). En el esquema de codificación de facsímile clásico se utiliza un método de codificación línea por línea bidimensional. En este método de codificación, la posición de cada elemento de cuadro cambiante en la línea de codificación actual se codifica con respecto a la posición de un elemento de referencia correspondiente, localizado ya sea en la línea de codificación o en la línea de referencia. La línea de referencia está inmediatamente arriba de la línea de codificación. Después de que la línea de codificación ha sido codificada, se convierte en la línea de referencia para la siguiente línea de codificación. La referencia para la primera línea de una página es una línea en blanco imaginaria. Este esquema de codificación es el mismo que el esquema de codificación bidimensional del Grupo 3 especificado en la recomendación CCITT T.4, excepto que la primera línea se codifica de diferente forma, y

que en el esquema de codificación bidimensional Grupo 3, cada K-ésima línea ($K=2$ o 4) de los datos de la imagen deben codificarse con el esquema de codificación unidimensional Grupo 3 para asegurar la recuperación adecuada de la imagen. La ventaja del esquema de codificación CCITT T.6 es que éste explota tanto las dependencias horizontales como las verticales, lo que resulta en una eficiencia de compresión mayor, ya que el esquema de codificación bidimensional CCITT T.4 requiere que cada K-ésima línea de los datos de la imagen se codifiquen por medio del esquema de codificación unidimensional Grupo 3.

Para documentos tamaño A4 de 200 DPI (puntos por pulgada), la proporción de compresión típica del esquema de codificación CCITT T.6 es de 15:1 a 20:1. Para imágenes de documento de 400 DPI, ésta varía de 30:1 a 40:1 debido a que la correlación entre los píxeles aumenta.

El procedimiento de codificación CCITT T.6 puede representarse con un diagrama de flujo como el de la figura 5.11.

Como el esquema de codificación CCITT T.6 está diseñado para transmisión libre de ruido, un solo bit defectuoso provoca la corrupción de la imagen completa. Comparado con el esquema de codificación CCITT T.4, este esquema es menos robusto ya que es muy sensible a los errores.

5.4. ESTÁNDAR DE COMPRESIÓN JBIG.

El estándar de compresión JBIG (ISO/IEC 11544 o también conocido como CCITT T.82) diseñado por el Grupo de Expertos Unidos de Imágenes Binivel (JBIG), es un estándar para codificación progresiva de imágenes binivel.

El estándar JBIG es un estándar de compresión sin pérdida, y soporta tanto operaciones de compresión secuencial como progresiva. La capacidad progresiva tiene dos beneficios:

- Una base de datos común puede servir eficientemente a dispositivos de salida con resoluciones muy diferentes.
- Proporciona búsqueda de imágenes a través de enlaces de comunicación de tasas baja y media.

El estándar JBIG también puede utilizarse para comprimir imágenes a escala de gris o a color por medio de la codificación de niveles de gris, donde cada nivel de gris se procesa como una imagen binivel.

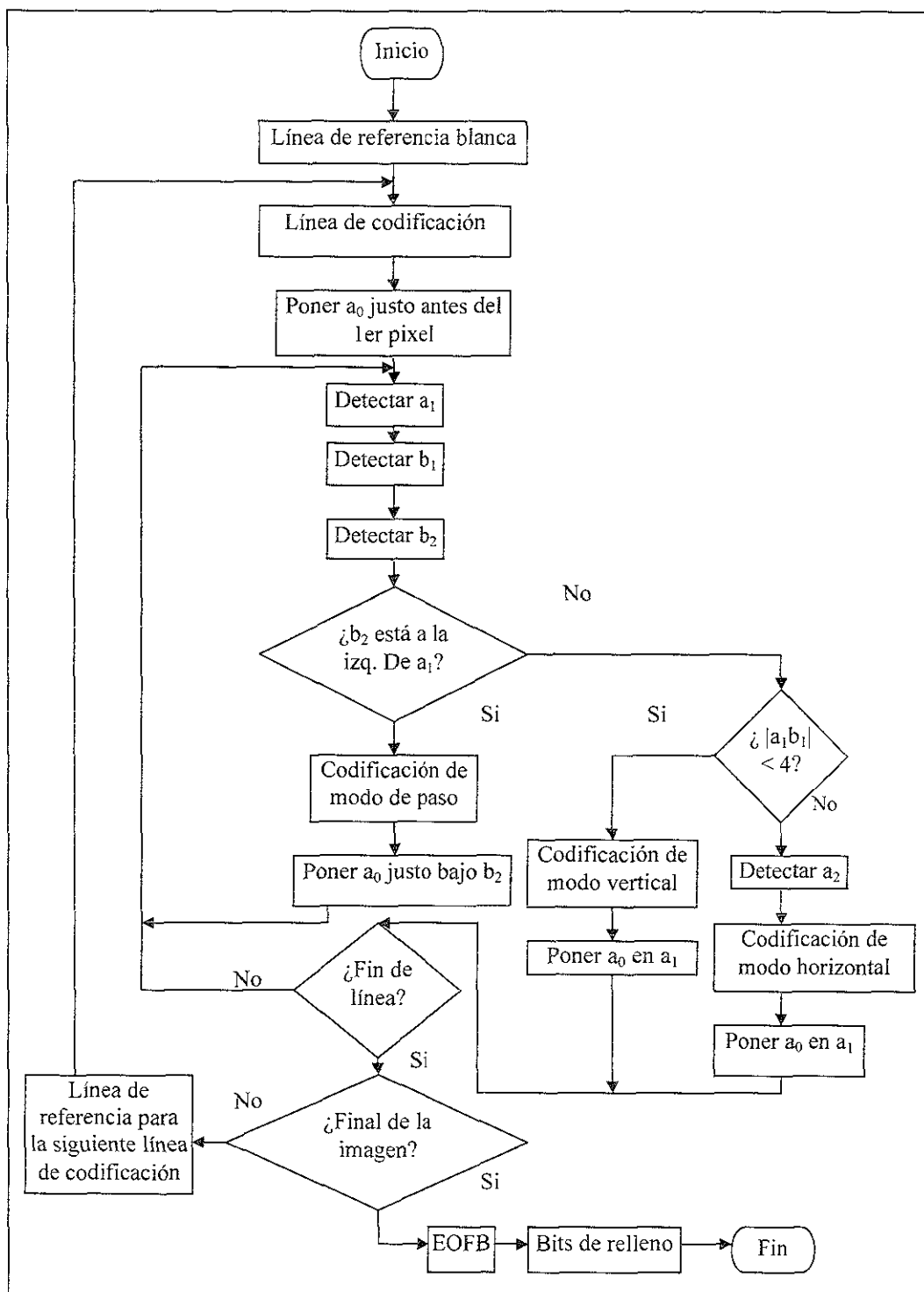


Figura 5.11. Diagrama de flujo de la codificación CCITT T.6.

El mayor beneficio del estándar JBIG es que logra una eficiencia de compresión mayor a la de los estándares CCITT T.4 y T.6, particularmente para comprimir imágenes de escala de gris, con proporciones de compresión de 2 a 30 veces mejor.

5.4.1. Definición de imagen JBIG.

Franjas (stripes) y ordenamiento de datos.

Las franjas y el ordenamiento de datos son componentes importantes para el funcionamiento del JBIG. Para definirlos es posible utilizar un ejemplo. Supóngase que se tiene una imagen binaria de 100 DPI, y se utilizan tres capas de resolución, 25, 50 y 100 DPIs respectivamente. Cada capa de resolución representa una imagen binivel. Cada una de estas tres imágenes se particiona en piezas más pequeñas, por ejemplo, en tres piezas. Cada una de estas tres piezas consiste en una banda horizontal que se denomina franja (stripe). Finalmente se obtiene un total de nueve franjas para tres imágenes como se muestra en la figura 5.12.

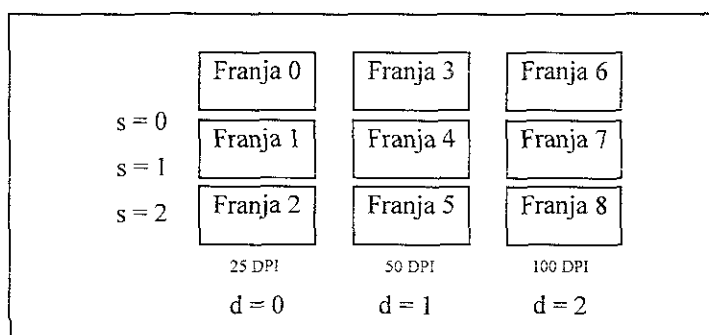


Figura 5.12. Franjas (stripes) de imágenes de resolución de tres capas (KOU, Weidong. Digital image compression. Algorithms and standards).

Ordenamiento de datos.

Estas nueve franjas pueden almacenarse en diferentes órdenes de acuerdo a su capa de resolución d, y al número de franja s. Esto se conoce como ordenamiento de datos. Existen cuatro ordenamientos de datos posibles como se muestra en la figura 5.13.

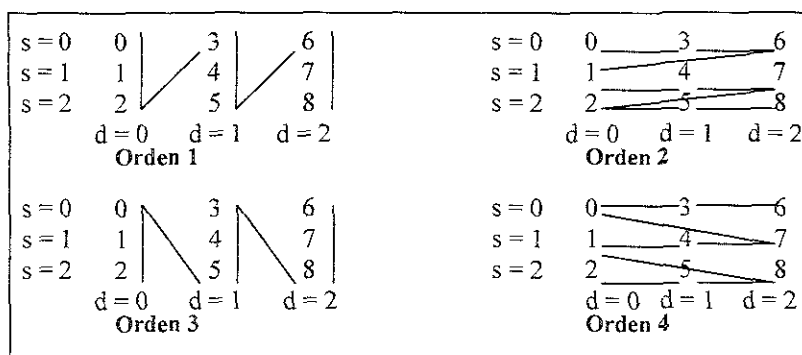


Figura 5.13. Posibles ordenamientos de datos binivales (KOU, Weidong. Digital image compression. Algorithms and standards).

Cuando existe más de plano de bits (nivel de gris), las franjas se almacenarán de acuerdo a la franja s, la capa d, y el plano de bit p, como se muestra en la figura 5.14.

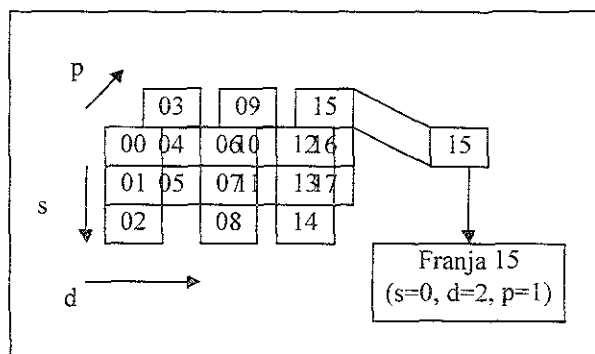


Figura 5.14. Esquema de una descomposición de 3 franjas por capa, 3 capas por plano de bits y 2 planos de bits (KOU, Weidong. Digital image compression. Algorithms and standards).

Al descomponer la imagen en franjas, el JBIG es capaz de proporcionar la característica de que el dato comprimido $C_{s,d,p}$, de la imagen de franja $I_{s,d,p}$ para la franja s de resolución d del plano de bit p , es independiente del ordenamiento de las franjas. Esta característica es importante debido a que la información total que describe a una imagen es independiente del método de codificación y del método prospectado para la descompresión. Una imagen puede comprimirse ya sea progresivamente o secuencialmente. Esta característica se denomina la característica de compatibilidad Progresiva/Secuencial.

Reducción de la resolución.

El JBIG soporta codificación progresiva, la cual trabaja con resoluciones múltiples. En el JBIG existe una relación entre pixeles en una imagen a alta resolución y pixeles en una imagen a baja resolución. La figura 5.15 muestra una representación gráfica tridimensional de dicha relación, la cual consiste en la generación de una imagen de menor resolución para el JBIG a partir de la imagen de alta resolución original.

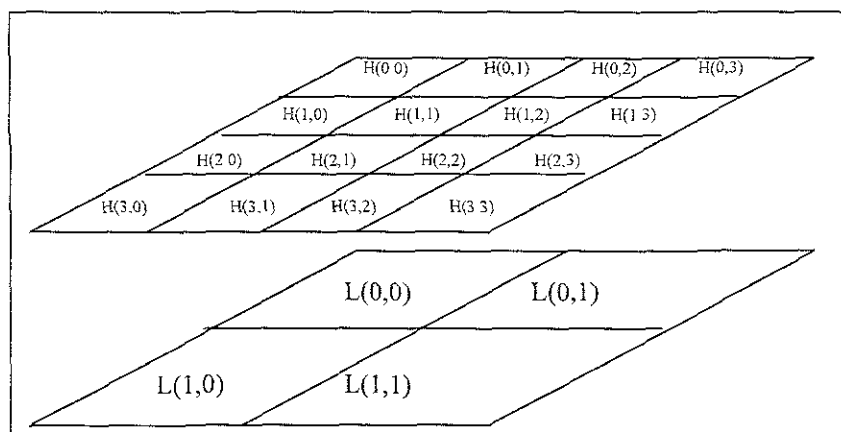


Figura 5.15. Pixeles de alta y baja resolución en una gráfica tridimensional (KOU, Weidong. Digital image compression. Algorithms and standards).

Patrones y adaptación.

Un patrón JBIG, también conocido como patrón de codificación o patrón de modelo, es un grupo de píxeles vecinos que proporcionan al codificador aritmético JBIG un número entero llamado contexto. En la figura 5.16 se muestran dos patrones JBIG donde el píxel denotado por la marca “?” corresponde al píxel que necesita codificarse y que no es parte del patrón. Los píxeles denotados por una “X” son los píxeles que sí pertenecen al patrón, y los píxeles denotados por “A” son píxeles especiales denominados píxeles adaptativos, o píxeles AT.

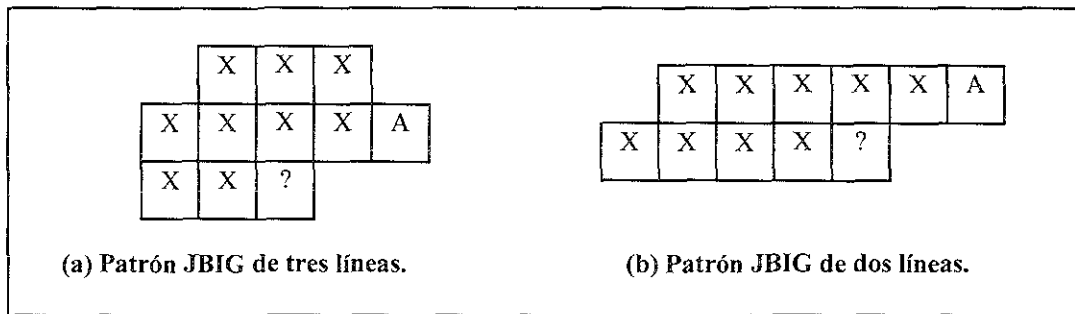


Figura 5.16. Dos patrones JBIG (KOU, Weidong. Digital image compression. Algorithms and standards).

Los valores de píxel de un patrón determinan el contexto. El contexto es utilizado por el codificador aritmético JBIG para establecer un estimado de la probabilidad condicional del píxel a codificar. Si esta probabilidad estimada es exacta y cercana a 0 o a 1, será posible calcular el valor de dicho píxel a partir del patrón utilizado para determinar su probabilidad, obteniéndose así una eficiencia de compresión mayor. El propósito de los patrones JBIG es lograr que el valor del píxel a codificar sea altamente predecible.

El JBIG utiliza varios esquemas de adaptación para manipular los patrones mostrados en la figura 5.16. Además de estos patrones, el JBIG también utiliza ciertos tipos de predicciones con el propósito igualmente de mejorar la adaptación de patrones. Existen dos clases de predicciones en el JBIG. Una se denomina predicción típica (TP) y la otra se denomina predicción determinística (DP).

Predicciones típicas.

Existen dos clases de predicciones típicas: la predicción típica en la capa diferencial (capa de baja resolución), y la predicción típica en la capa inferior (capa de alta resolución original).

La predicción típica de la capa diferencial busca las regiones que contengan un solo color. Verificando el color de todos los píxeles en una vecindad de ocho píxeles, como se ve en la figura 5.17, es posible hacer una predicción de los cuatro píxeles de alta resolución asociados con el píxel X de la figura. Si los ocho píxeles tienen el mismo color de X, entonces los cuatro píxeles de alta resolución asociados con X muy probablemente tendrán el mismo color de X. En los casos de no ser así, se utiliza información adicional para indicarlos.

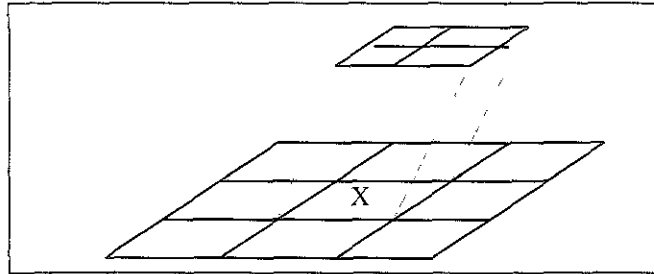


Figura 5.17. Predicción típica de capa diferencial (KOU, Weidong. Digital image compression. Algorithms and standards).

Las ventajas de la predicción típica de capa diferencial son:

- La implementación se acelera ya que tanto el codificador como el decodificador pueden omitir los píxeles de alta resolución asociados con X si la predicción es exitosa.
- La eficiencia de compresión se mejora ya que no se necesita codificar los píxeles de alta resolución asociados con X si la predicción es exitosa.

La predicción típica de capa inferior es más simple que la de capa diferencial debido a que no son necesarias la reducción de resolución y la predicción determinística y en su lugar se utiliza una técnica de salto de línea. Si una línea es idéntica a la línea anterior a ella, la línea se denomina línea típica. Todos los píxeles en la línea típica no son codificados, y en lugar de eso son generados tanto por el codificador como por el decodificador. Utilizando la técnica de salto de línea se logra acelerar el procesamiento, además de que se mejora la eficiencia de la compresión.

A diferencia de las predicciones típicas usadas principalmente para acelerar la implementación, el propósito de la predicción determinística es el de mejorar la eficiencia de la compresión. Cuando las imágenes sufren una reducción en la resolución debido a un algoritmo de reducción de resolución en particular, el valor de un píxel de alta resolución dado es predecible a partir de los píxeles ya conocidos tanto por el codificador como por el decodificador. En este caso, dicho píxel se dice que es determinísticamente predecible, por lo que puede evitarse la codificación del mismo.

La predicción determinística es un algoritmo basado en tablas. Este utiliza los valores de los pixeles circundantes particulares en la imagen de baja resolución y en la imagen de alta resolución original, indexándolos en una tabla para verificar su posibilidad de determinarse. Cuando existe esta posibilidad, se obtiene el resultado de la predicción determinística.

Las tablas de predicción determinística son fuertemente dependientes del algoritmo de reducción de resolución utilizado. El JBIG permite al usuario tener tablas de predicción determinística personalizadas asociadas también con un algoritmo de reducción de resolución personalizado. Si el usuario decide utilizar tablas de predicción determinística personalizadas en lugar de las tablas sugeridas por el JBIG, entonces dichas tablas deben cargarse tanto al decodificador como al codificador.

5.4.2. Compresión JBIG secuencial.

La codificación secuencial JBIG es una codificación de capa única del estándar JBIG. Algunas veces se denomina también JBIG progresivo único. En la figura 5.18 se muestra un sistema de codificación secuencial JBIG, donde $I_{s,0}$ es la imagen de entrada a codificarse, $C_{s,0}$ es la imagen comprimida, "ATMOVE" significa movimiento de patrones adaptativos, "TPVALUE" significa valor de predicción típica y "SLNTP" significa misma línea no típica.

En la figura 5.18(a), la imagen de entrada $I_{s,0}$ se introduce al codificador aritmético. El codificador aritmético produce la imagen comprimida $C_{s,0}$ basándose en los siguientes valores: los valores TPVALUE y SLNTP de la predicción típica de la capa de resolución única, y del contexto a partir de los patrones de modelo. Se envía una secuencia de control ATMOVE desde los patrones adaptativos hacia el decodificador como información adicional para ayudar al proceso de descompresión para que así el decodificador no necesite realizar ninguna búsqueda de la configuración correcta para los patrones adaptativos.

En la figura 5.18(b), la imagen reconstruida $i_{s,0}$ se produce por medio del decodificador aritmético basándose en la imagen comprimida $C_{s,0}$, en el contexto de los patrones de modelo, y en la información de TPVALUE y SLNTP de la predicción típica de la capa de resolución única. Como el JBIG es un proceso sin pérdida de información, la imagen reconstruida es idéntica a la imagen original.

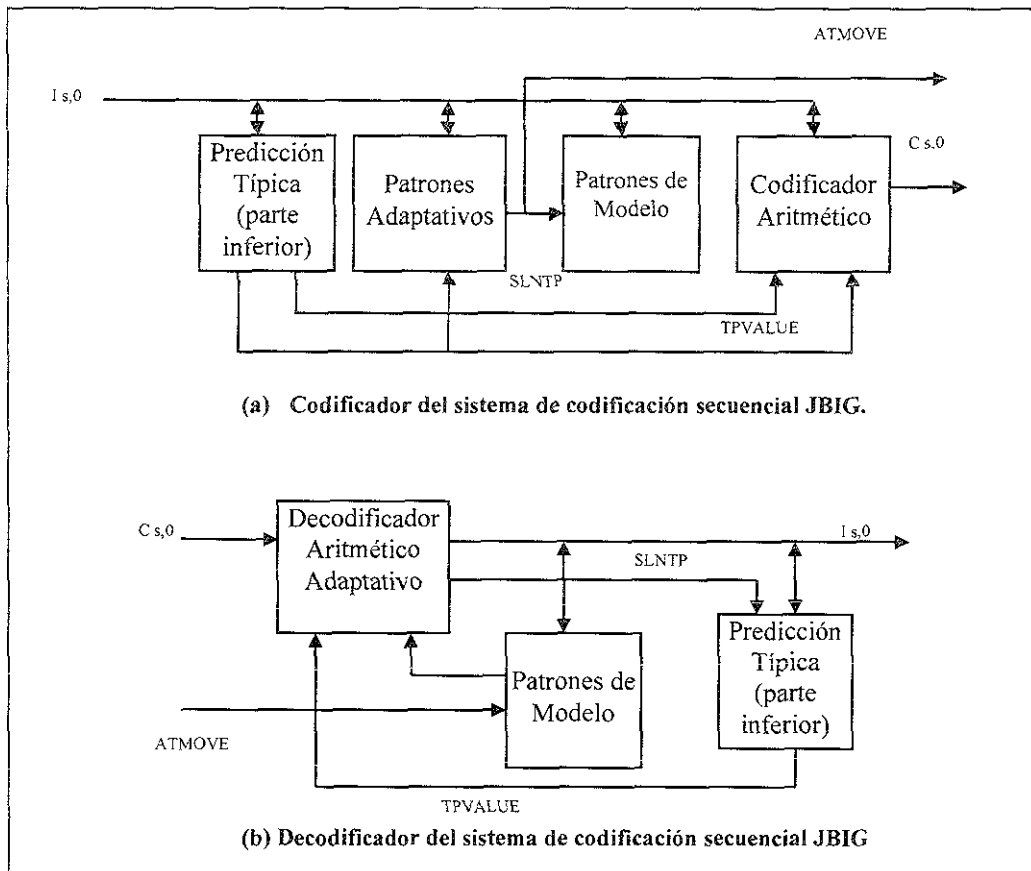


Figura 5.18. Sistema de codificación secuencial JBIG: (a) codificador y (b) decodificador (KOU, Weidong. Digital image compression. Algorithms and standards).

5.4.3. Compresión JBIG progresiva.

A diferencia de la codificación secuencial JBIG de capa única, la codificación progresiva JBIG es un esquema de codificación de múltiples capas. La codificación JBIG secuencial trabaja sobre franjas de una imagen mientras que la codificación JBIG progresiva trabaja tanto con franjas como con las capas de resolución de una imagen. La codificación JBIG secuencial es un caso especial de la codificación JBIG progresiva.

Debido a que la codificación JBIG progresiva trabaja sobre múltiples capas de una imagen, un sistema de codificación JBIG progresivo tiene múltiples codificadores para codificar imágenes de múltiples capas. También cuenta con múltiples decodificadores para decodificar imágenes comprimidas de múltiples capas. El codificador de la capa más baja consiste en el mismo codificador utilizado para la capa única en el sistema de codificación JBIG secuencial, y de la misma forma corresponde a su decodificador. En la figura 5.19 se muestra un sistema de codificación JBIG progresivo. El codificador y el decodificador de la capa diferencial se muestran en la figura 5.20.

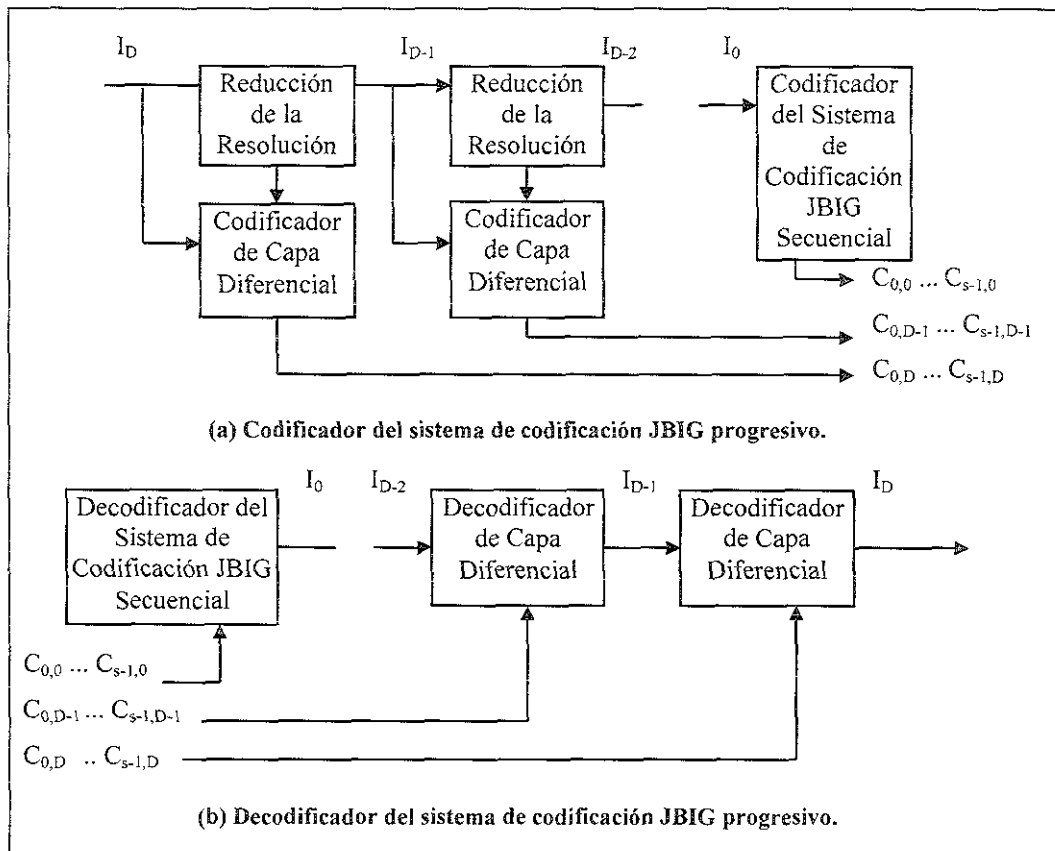


Figura 5.19. Sistema de codificación JBIG progresivo: (a) codificador y (b) decodificador (KOU, Weidong, Digital image compression. Algorithms and standards).

La codificación progresiva requiere un buffer de cuadro para almacenar la imagen de resolución más alta siguiente debido a que estas se utilizan en el proceso de codificar las imágenes de mayor resolución siguientes.

El JBIG especifica la codificación de múltiples resoluciones de forma que la información total que describe a una imagen es independiente del método de codificación y del método esperado de decodificación. A esto se denomina codificación compatible progresiva/secuencial o codificación CP/S.

5.5. ESTÁNDAR DE COMPRESIÓN JPEG.

A finales de los 70's y principios de los 80's, comenzó la investigación de nuevos tipos de compresión de imágenes que planteaban mejorar el desempeño de las técnicas de codificación sin pérdida de información. A fines de los 80's, estas investigaciones encontraron una aplicación comercial en el campo del procesamiento de imágenes para sistemas personales y de microcomputadoras, la mayoría en la forma de tarjetas

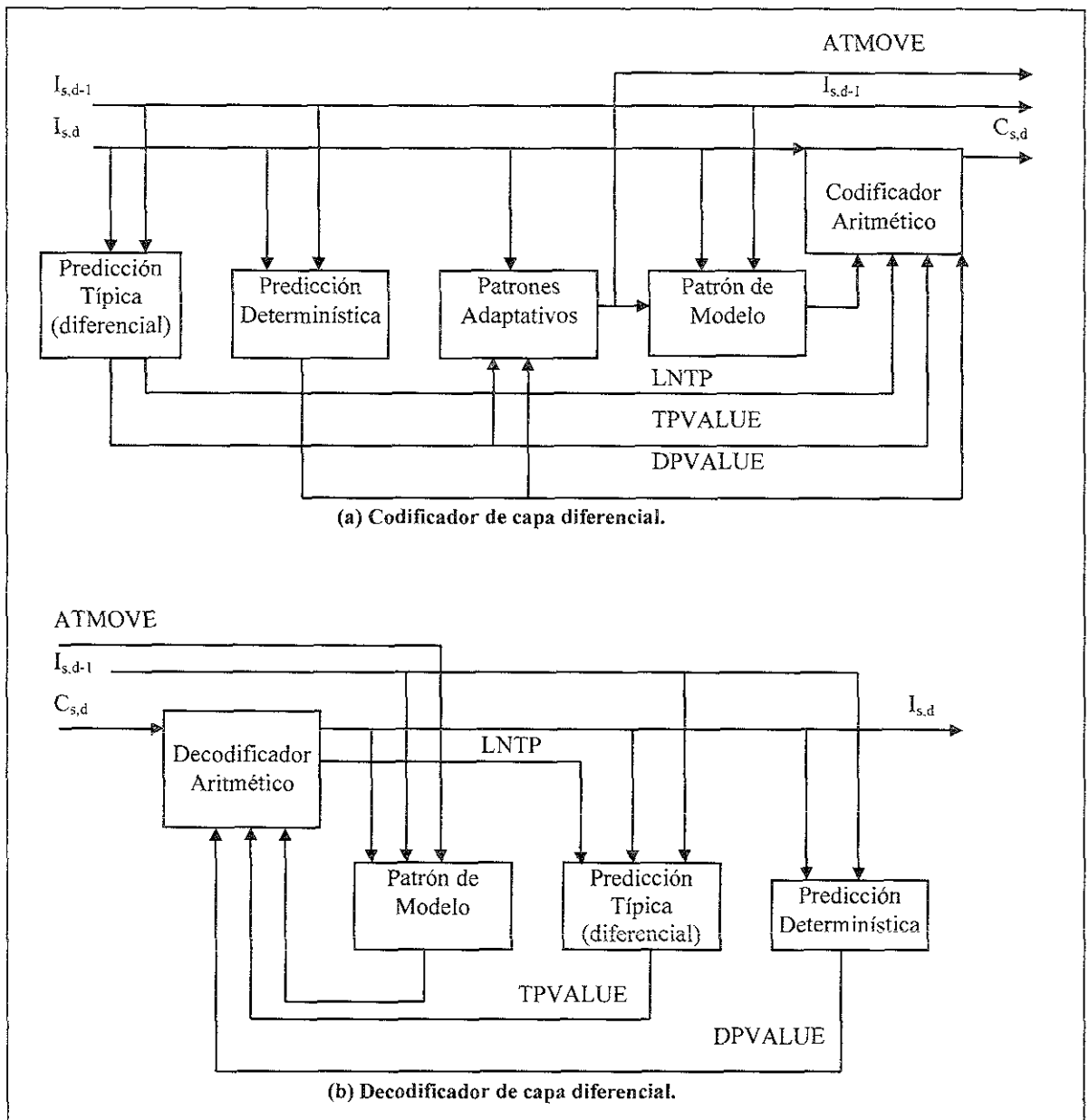


Figura 5.20. Codificador y decodificador diferencial (KOU, Weidong. Digital image compression. Algorithms and standards).

coprocesadoras agregadas para estaciones de trabajo UNIX y Macintosh. Estas tarjetas efectuaban compresión con pérdida en imágenes, en sus principios, a niveles hasta del 95 por ciento sin degradación visible de la calidad de la imagen. Grupos de investigación a nivel mundial comenzaron el desarrollo de un estándar internacional que abarcara estas nuevas modalidades de compresión con pérdida. Los estándares propuestos debían cumplir el requisito de permitir un intercambio fácil entre los distintos formatos gráficos existentes para distintas plataformas. Dos de las primeras organizaciones internacionales que pretendían diseñar un estándar de compresión gráfico, el entonces Comité Consultivo Internacional de Telegrafía y Telefonía (CCITT) y la Organización de Estándares Internacionales (ISO), trabajaron para involucrar a los grupos industriales y académicos en

la formación de un estándar de compresión de imágenes digitales. Estas dos organizaciones formalizaron la creación del organismo que daría sustento a este nuevo estándar, el Grupo de Expertos Fotográficos Unidos (JPEG). El estándar de compresión de imagen JPEG proporciona cuatro modos de operación:

- Codificación Basada en DCT Secuencial, en la que cada componente de imagen es codificado en una exploración única de izquierda a derecha y de arriba hacia abajo. Este modo también se conoce como JPEG básico.
- Codificación Basada en DCT Progresiva, en el que la imagen es codificada en múltiples exploraciones, para producir una imagen decodificada preliminar cuando el tiempo de transmisión es largo.
- Codificación Sin Pérdida, en el que la imagen es codificada de forma que se garantice su reproducción exacta.
- Codificación Jerárquica, en la cual la imagen es codificada a múltiples resoluciones.

5.5.1. Definición de imagen JPEG.

El estándar JPEG es aplicable a imágenes de escala de gris o a color. La forma en que el estándar JPEG define una imagen fuente es de gran importancia para su funcionamiento.

Componentes de imagen.

En el estándar JPEG, una imagen fuente se define como una estructura de N_f componentes C_i para $i = 1, \dots, N_f$ como es muestra en la figura 5.21.

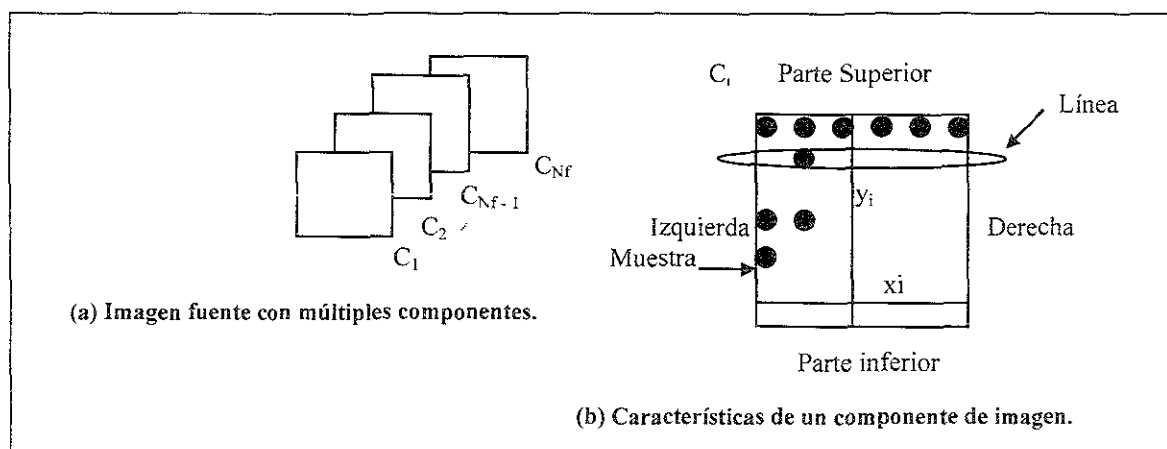


Figura 5.21. Características de la imagen fuente para el estándar JPEG (KOU, Weidong, Digital image compression. Algorithms and standards).

Una imagen fuente a color consiste de tres componentes: rojo, verde y azul. Cada componente está definido como un arreglo rectangular de muestras de x_i columnas por y_i renglones. Un solo paso a través de los datos de uno o más de los componentes en una imagen se denomina exploración (scan). Un conjunto de una o más exploraciones a través de los datos de uno o más componentes en una imagen se denomina cuadro (frame). Para todos los componentes de un cuadro, el número máximo de los valores x_i define el número de muestras por línea X , y el número máximo de los valores y_i define el número de líneas Y , en la imagen fuente.

Muestreo de la imagen.

Para cada componente, existen dos factores de muestreo, uno para la dirección horizontal y otro para la dirección vertical. Éstos se denotan como H_i y V_i para el i -ésimo componente. Estos dos factores de muestreo se definen por las siguientes expresiones:

$$x_i = \text{ceiling}(X * (H_i/H_{\max}))$$

$$y_i = \text{ceiling}(Y * (V_i/V_{\max}))$$

donde x_i es el número de columnas y y_i es el número de renglones del i -ésimo componente, X es el número de muestras por línea y Y es el número de líneas en la imagen fuente, H_{\max} y V_{\max} son los factores de muestreo máximo para todos los componentes en la imagen fuente, y $\text{ceiling}(r)$ es una función de redondeo del número real r hacia el entero más cercano.

Además de los parámetros de dimensión de imagen y de muestreo, existen otros parámetros para describir una imagen fuente, como lo son:

- Precisión de muestra: Una muestra se define como un entero S con precisión de P bits ($2 \leq P \leq 16$), con cualquier valor en el rango de $0 \leq S \leq 2^P - 1$. Dentro de la misma imagen fuente, la precisión es la misma para todas las muestras de todos los componentes.
- Orden de la codificación de los datos de imagen fuente: Intercalado o no intercalado. Para una exploración dada, si el número de componentes (N_s) en la exploración es mayor a 1, entonces los datos de los N_s componentes están presentes dentro de la exploración. Es decir que es de orden intercalado. Si $N_s = 1$, entonces sólo los datos de un componente fuente están presentes dentro de la exploración y se dice que no está intercalado o que es de orden no intercalado.
- Unidad de datos: Una unidad de datos es una muestra para los procesos sin pérdida y también un bloque de 8×8 de muestras continuas en los procesos basados en DCT.

- Orientación: Izquierda o derecha, parte superior o inferior dentro de un componente de imagen.
- Unidad codificada mínima (MCU): Para datos no intercalados el MCU es una unidad de datos. Para datos intercalados la MCU es la secuencia de unidades de datos definida por los factores de muestreo de los componentes en la exploración.

En el estándar JPEG, el encabezado de cada cuadro utiliza los parámetros listados en la tabla 5.8.

N_f	Número de componentes de una imagen
X	Número de muestras por línea en una imagen
Y	Número de líneas en una imagen
H_i	Factor de muestreo horizontal para el i -ésimo componente
V_i	Factor de muestreo vertical para el i -ésimo componente
P	Precisión por muestra en bits
N_s	Número de componentes en una exploración

Tabla 5.8. Parámetros para el encabezado de cuadro definido en el JPEG

Ordenamiento de las muestras.

Otra cuestión importante tiene que ver con la forma en que se ordenan las unidades de datos. Existe un ordenamiento para los datos intercalados y otro para los no intercalados. Si solamente hay un componente en una exploración, es decir que es una exploración de orden no intercalado, el orden de exploración debe ser de izquierda a derecha y de la parte superior hacia la parte inferior, como se muestra en la figura 5.22.

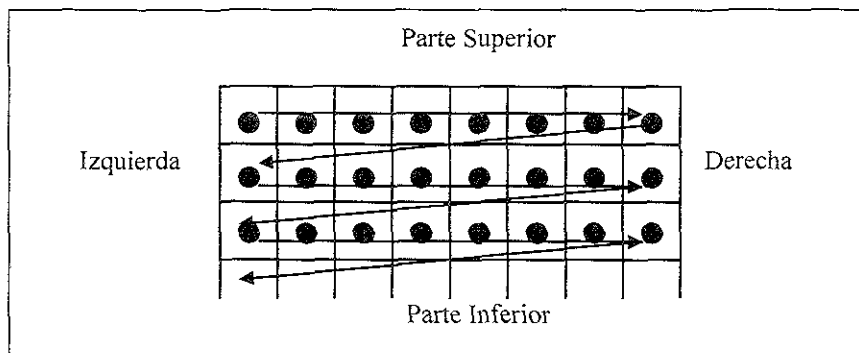


Figura 5.22. Ordenamiento de datos no intercalados (KOU, Weidong. *Digital image compression. Algorithms and standards*).

Cuando hay más de un componente en una exploración, cada componente se particiona en pequeños arreglos rectangulares. Las dimensiones de estos arreglos se determinan basándose en la información del componente contenida en el encabezado del cuadro. Dentro de cada arreglo, las unidades de datos se ordenan de izquierda a derecha y de la parte superior a la parte inferior. Los arreglos en sí mismos también se ordenan en los mismos sentidos dentro de cada componente. La figura 5.23 muestra un ejemplo de ordenamiento intercalado con $N_s = 4$, donde cada componente C_{s_i} se particiona en $H_k \times V_k$ arreglos.

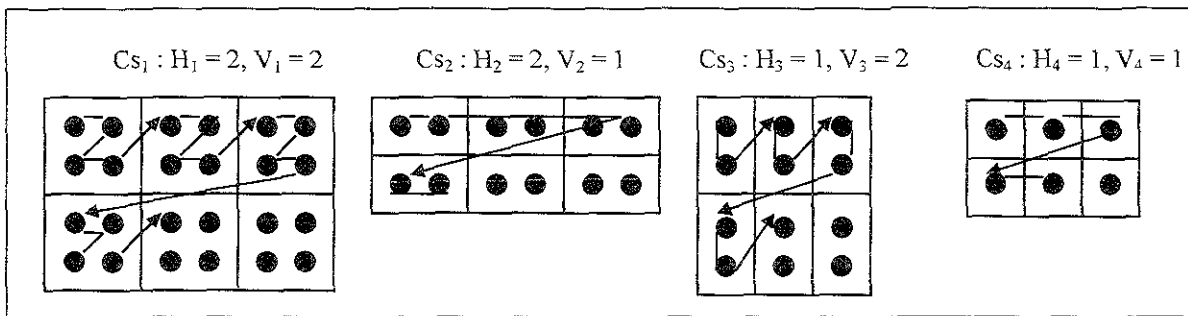


Figura 5.23. Ejemplos de ordenamiento de datos intercalados (KOU, Weidong. Digital image compression. Algorithms and standards).

5.5.2. Compresión JPEG secuencial basada en DCT.

El algoritmo de compresión secuencial JPEG opera en tres etapas sucesivas, las cuales se muestran en la figura 5.24.

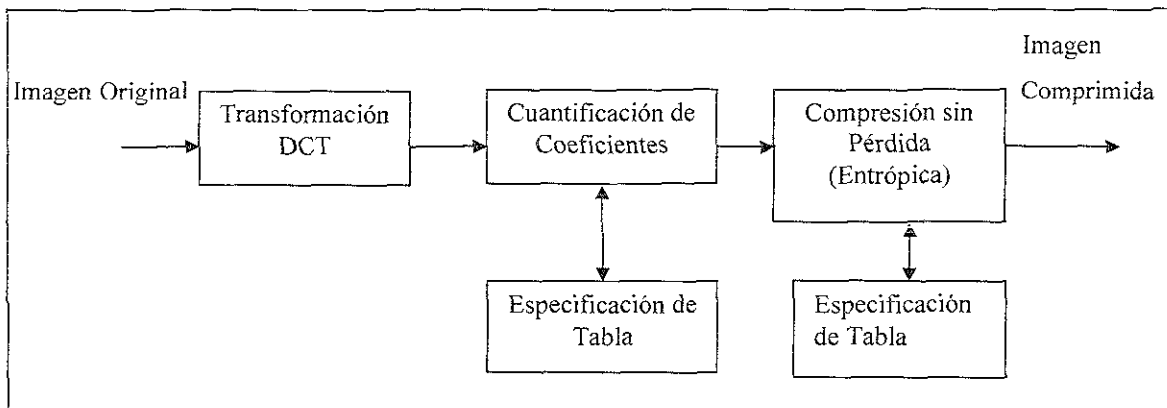


Figura 5.24. Diagrama del proceso de codificación JPEG secuencial (KOU, Weidong. Digital image compression. Algorithms and standards).

El estándar JPEG especifica cinco procesos de codificación para la codificación secuencial basada en DCT:

- Secuencial básica (JPEG básico).
- Secuencial extendido, con codificación Huffman, con precisión de 8 bits/muestra.
- Secuencial extendido, con codificación aritmética, con precisión de 8 bits/muestra.
- Secuencial extendida, con codificación Huffman, con precisión de 12 bits/muestra.
- Secuencial extendida, con codificación aritmética, con precisión de 12 bits/muestra.

La clave del proceso de compresión secuencial es el uso de la transformada de coseno discreta (DCT) discutida en la sección 2.3.2. Esta transformada cambia los datos que componen una imagen del plano espacial en el que se encuentran, hacia el plano de frecuencia de esos valores con respecto al resto de un espacio determinado. La DCT reduce la correlación entre los píxeles, acumulando los valores de menor frecuencia para un espacio dado en pocos coeficientes de transformada resultantes.

Implementación de la DCT.

Una de las primeras características que se muestran al examinar el algoritmo DCT es que el tiempo de cálculo requerido para cada elemento de la DCT depende fuertemente del tamaño de la matriz y conforme éste aumenta, la cantidad de tiempo necesario para procesar cada elemento en el arreglo DCT aumenta también.

Una consecuencia de esta situación es que en cuanto a procesamiento digital, es impráctico efectuar una DCT sobre una imagen completa. Para solucionar este problema, las implementaciones de la DCT como el estándar JPEG, generalmente descomponen la imagen en bloques más pequeños y manejables. El grupo JPEG seleccionó un tamaño de bloque de 8x8 para efectuar la DCT sobre los datos. A pesar de que si se incrementa el tamaño de bloque se obtendría una mejor compresión, esta mejora tiende a hacerse estacionaria. La correlación entre píxeles tiende a disminuir conforme aumenta la distancia entre ellos, tanto que los píxeles alejados 15 o 20 posiciones son de poca utilidad como predicciones del píxel de referencia. Esto significa que un bloque DCT de 64x64 podría no comprimir mucho mejor que si se descompone en cuatro bloques de 16x16, y su tiempo de computación sería mucho mayor. Por estas razones el Comité JPEG decidió apegarse al tamaño de 8x8.

Multiplicación de matrices.

La definición de la DCT permite estructurar una forma muy eficiente de implementación a través de operaciones de matrices. Para efectuar este proceso, primero se crea una matriz $N \times N$ denominada como la matriz de transformada de coseno (C) a partir de la fórmula expuesta en la sección 2.3.2. Una vez que se ha calculado la matriz transformada de coseno, se traspone girándola alrededor de la diagonal principal. La matriz resultante se denomina matriz de transformada de coseno transpuesta (C_t). Los cálculos de ambas matrices se realizan una sola vez al comienzo de la compresión. El siguiente código construye ambas matrices[^]:

[^] NELSON, Mark. The data compression book. Estados Unidos, Ed. M & T Books, 1992.

```

For (j = 0; j < N; j++) {
    C[0][j] = 1.0 / sqrt(N);
    C_t[j][0] = C[0][j]; }
For (i = 0; i < N; i++) {
    For (j = 0; j < N; j++) {
        C[i][j] = sqrt(2.0 / N) *
            Cos((2 * j + 1) * i * pi
                / (2.0 * N));
        C_t[j][i] = C[i][j]; } }

```

Una vez determinadas estas dos matrices, es posible aprovechar la definición de la transformada DCT por medio de la siguiente fórmula:

$$DCT = C \cdot P \cdot C_t$$

donde C es la matriz de coseno, C_t es la matriz de coseno transpuesta y P es la matriz de 8x8 de datos original.

El siguiente es un código que implementa la DCT vía aritmética de matrices*. Nótese que esencialmente, el código no es más que un conjunto de dos ciclos triplemente anidados. El primer conjunto de ciclos multiplica la matriz de transformada de coseno transpuesta por el conjunto de pixeles de entrada, creando una matriz temporal. La matriz temporal se multiplica entonces por la matriz de transformada de coseno, lo que resulta en la matriz DCT de salida.

```

/* MatrixMultiply ( temp.input * C_t ) */
For (i = 0; i < N; i++) {
    For (j = 0; j < N; j++) {
        Temp[i][j] = 0.0;
        For (k = 0; k < N; k++)
            Temp[i][j] += (pixel[i][k] * C_t[k][j]); } }
/* MatrixMultiply ( output.C * temp ) */
For (i = 0; i < N; i++) {
    For (j = 0; j < N; j++) {
        Temp1 = 0.0;
        For (k = 0; k < N; k++)
            Temp1 += C[i][k] * Temp[k][j]);
        DCT[i][j] = Temp1; } }

```

Salida de la DCT.

La figura 5.25 muestra un bloque de entrada representativo de una imagen de escala de gris. A la entrada del codificador las muestras originales, las cuales están en el rango de $[0, 2^p - 1]$ donde p es igual al número de niveles de gris, son desplazadas a enteros dentro del rango de $[-2^{p-1}, 2^{p-1} - 1]$. Para un bloque de una imagen a escala de gris como el de la figura 5.25, donde $p=8$, las muestras originales con rango $[0, 255]$ son desplazadas al rango de $[-128, +127]$.

* NELSON, Mark. The data compression book. Estados Unidos, Ed. M & T Books. 1992

Matriz de Pixeles de Entrada	140	144	147	140	140	155	179	175
	144	152	140	147	140	148	167	179
	152	155	136	167	163	162	152	172
	168	145	156	160	152	155	136	160
	162	148	156	148	140	136	147	162
	147	167	140	155	155	140	136	162
	136	156	123	167	162	144	140	147
	148	155	136	155	152	147	147	136
Bloque de Pixeles Desplazado P	12	16	19	12	11	27	51	47
	16	24	12	19	12	20	39	51
	24	27	8	39	35	34	24	44
	40	17	28	32	24	27	8	32
	34	20	28	20	12	8	19	34
	19	39	12	27	27	12	8	34
	8	28	-5	39	34	16	12	19
	20	27	8	27	24	19	19	8

Figura 5.25. Matriz de pixeles originales y desplazados para la transformación DCT.

En base a la fórmula de la DCT que utiliza multiplicación de matrices y tomando a $n=8$ por el tamaño del bloque a codificar, se tiene que la matriz de transformada de coseno resulta como la de la figura 5.26.

Matriz C	.353	.353	.353	.353	.353	.353	.353	.353
	.49	.415	.277	.097	-.097	-.277	-.415	-.49
	.462	.191	-.191	-.462	-.462	-.191	.191	.462
	.415	-.097	-.49	-.277	.277	.49	.097	-.415
	.353	-.353	-.353	.353	.353	-.353	-.353	.353
	.277	-.49	.097	.415	-.415	-.097	.49	-.277
	.191	-.462	.462	-.191	-.191	.462	-.462	.191
	.097	-.277	.415	-.49	.49	-.415	.277	-.097

Figura 5.26. Matriz de transformada de coseno para un bloque de 8x8.

La matriz transpuesta correspondiente a la anterior se muestra en la figura 5.27(con truncamiento a 3).

Matriz C_t	.353	.49	.462	.415	.353	.277	.191	.097
	.353	.415	.191	-.097	-.353	-.49	-.462	-.277
	.353	.277	-.191	-.097	-.353	.097	.462	.415
	.353	.097	-.462	-.277	.353	.415	-.191	-.49
	.353	-.097	-.462	.277	.353	-.415	-.191	.49
	.353	-.277	-.191	.49	-.353	-.097	.462	-.415
	.353	-.415	.191	.097	-.353	.49	-.462	.277
	.353	-.49	.462	-.415	.353	-.277	.191	-.097

Figura 5.27. Matriz de transformada de coseno transpuesta para un bloque de 8x8.

En la figura 5.28 se muestra un paso intermedio del cálculo de la matriz de coeficientes transformados y en la figura 5.29 se muestra el resultado final de la transformación.

Matriz P·C _t	68.635	-33.794	20.643	-7.487	-10.943	7.094	-2.826	2.49
	68.129	-24.912	22.553	-11.089	1.059	-.216	-7.446	-5.99
	96.722	-15.369	-1.053	3.041	17.297	-7.872	-5.304	-15.521
	73.424	8.708	1.662	-.259	16.944	1.223	16.916	-5.222
	61.775	6.731	17.205	-12.113	8.825	4.77	5.49	4.103
	62.834	5.515	3.931	-9.232	12.708	-19.345	-10.817	-10.042
	53.303	-4.082	-15.713	2.788	17.297	-10.849	-22.184	-16.664
	53.656	6.444	-6.997	8.763	2.118	-.418	-13.171	-7.087

Figura 5.28. Matriz intermedia del proceso de transformación DCT.

Matriz DCT	185	-17	14	-8	23	-9	-13	18
de Salida	20	-34	26	-9	-10	10	13	6
(C·P·C _t)	-10	-23	-1	6	-18	3	-20	0
	-8	-5	14	-14	-8	-2	-3	8
	-3	9	7	1	-11	17	18	15
	3	-2	-18	8	8	-3	0	-6
	8	0	-2	3	-1	-7	-1	-1
	0	-7	-2	1	1	4	-6	0

Figura 5.29. Matriz resultante del proceso de transformación DCT de la matriz original.

La matriz de salida muestra la concentración de valores en el plano de la frecuencia que se ha obtenido por medio de la transformación DCT. La matriz C evalúa la correlación entre los pixeles del arreglo de 8x8 sobre cada columna, comenzando con el cálculo de un promedio general del bloque y asignándolo al primer coeficiente de la nueva matriz. Posteriormente se van comparando las muestras, desde las que se encuentran a los extremos de la columna y terminando con las que se encuentran más cerca de la parte media, acumulando los resultados en los coeficientes de la parte superior de cada columna. La matriz C_t realiza la misma operación pero sobre cada renglón, acumulando los resultados en los primeros coeficientes de cada renglón. A este proceso se le denomina compactación de energía como se describió en la definición de las transformadas.

En la posición 0,0 se encuentra el denominado coeficiente DC. Este valor representa un promedio de la magnitud total de la matriz de entrada tanto en su sentido vertical como horizontal. Nótese que el coeficiente DC es de magnitud mayor al resto de los valores de la matriz DCT, los cuales se denominan coeficientes AC. Además, conforme los elementos se alejan del coeficiente DC, tienden a hacerse más y más pequeños en su magnitud. La DCT es una transformación sin pérdida que no efectúa ninguna compresión, sólo prepara los datos para la etapa con pérdida o de cuantificación del proceso.

Cuantificación.

La cuantificación es el proceso de reducir el número de bits necesarios para almacenar un valor entero, reduciendo su precisión. En el caso de la matriz DCT, entre más lejos se esté de la posición 0,0 menos contribuyen los elementos a la representación de la imagen, y por tanto no es necesario mantener una precisión rigurosa en su valor.

El algoritmo JPEG implementa la cuantificación usando una matriz de cuantificación. Para cada posición de elemento de la matriz DCT, se asigna un valor correspondiente en la matriz de cuantificación denominado valor cuanto. El valor cuanto indica qué tamaño de paso (factor que dividirá a cada coeficiente de la matriz DCT) se ocupará para cada elemento en la representación comprimida de la imagen, con valores que varían de 1 a 255.

Los elementos que importan más al cuadro se codificarán con un tamaño de paso pequeño, ofreciendo el tamaño 1 la mayor precisión con que se conservará el coeficiente DCT. Los valores cuanto pueden elevarse conforme el recorrido se aleja del origen. La fórmula para la cuantificación se define como:

$$F_q(u, v) = \text{Round} \left\lfloor \frac{F(u, v)}{Q(u, v)} \right\rfloor$$

donde $Q(u, v)$ son los coeficientes de cuantificación especificados por una tabla de cuantificación. Como la matriz de cuantificación puede definirse en tiempo de ejecución cuando se efectúa la compresión, el JPEG permite el uso de cualquier matriz de cuantificación. Sin embargo, el ISO ha desarrollado un conjunto estándar de valores de cuantificación para ser usados por implementadores de código JPEG. Estas tablas se basan en muchas pruebas efectuadas por miembros del Comité JPEG, y proporcionan una buena base para alcanzar altos niveles de compresión sin un deterioro importante en la apariencia de la imagen recuperada. El estándar JPEG especifica cuatro tablas de cuantificación, las cuales se muestran en la figura 5.30.

Como ejemplo para elaborar tablas de cuantificación propias, pueden generarse tablas de cuantificación a partir de un algoritmo que se efectúa por medio del siguiente código[^]. Para determinar el valor de los tamaños de paso, se introduce un factor de calidad (quality), el cual varía de 1 a 25.

```
For ( i = 0; i < N; i ++ )
  For ( j = 0; j < N; j ++ )
    Quantum [ i ] [ j ] = 1 + (( 1 + i + j ) * quality );
```

[^] NELSON, Mark. The data compression book. Estados Unidos. Ed. M & T Books, 1992.

8	6	5	8	12	20	26	30	9	9	12	24	50	50	50	50
6	6	7	10	13	29	30	28	9	11	13	33	50	50	50	50
7	7	8	12	20	29	35	28	12	13	28	50	50	50	50	50
7	9	11	15	26	44	40	31	24	33	50	50	50	50	50	50
9	11	19	28	34	55	52	39	50	50	50	50	50	50	50	50
12	18	28	32	41	52	57	46	50	50	50	50	50	50	50	50
25	32	39	44	52	61	60	51	50	50	50	50	50	50	50	50
36	46	48	49	56	50	52	50	50	50	50	50	50	50	50	50
16	17	18	19	20	21	22	23	16	16	19	22	26	27	29	34
17	18	19	20	21	22	23	24	16	16	22	24	27	29	34	37
18	19	20	21	22	23	24	25	19	22	26	27	29	34	34	38
19	20	21	22	23	24	25	26	22	22	26	27	29	34	37	40
20	21	22	23	24	25	26	27	22	26	27	29	32	35	40	48
21	22	23	24	25	26	27	28	26	27	29	32	35	40	48	58
22	23	24	25	26	27	28	29	26	27	29	34	38	46	56	69
23	24	25	26	27	28	29	30	27	29	35	38	46	56	69	83

Figura 5.30. Tablas de cuantificación propuestas por el estándar JPEG (FURHT, Boiko, SMOLIAR, Stephen. Video and image processing in multimedia systems).

El factor de calidad establece la diferencia entre bandas adyacentes del mismo nivel de cuantificación. Estas bandas están orientadas en líneas diagonales a través de la matriz, de forma tal que los niveles de cuantificación del mismo valor están todos de alguna forma a la misma distancia del origen. Un ejemplo de cómo luce una matriz de cuantificación con un factor de calidad 2 generada a partir de este código se muestra en la figura 5.31.

Matriz de Cuantificación	3	5	7	9	11	13	15	17
	5	7	9	11	13	15	17	19
	7	9	11	13	15	17	19	21
	9	11	13	15	17	19	21	23
	11	13	15	17	19	21	23	25
	13	15	17	19	21	23	25	27
	15	17	19	21	23	25	27	29
	17	19	21	23	25	27	29	31

Figura 5.31. Matriz de cuantificación generada a partir del código propuesto.

Como resultado de esta configuración, el coeficiente DCT en 7,7 tendría que alcanzar un valor de 16 para no ser sustituido por un cero. Esto establece el límite para el valor de un elemento antes de aportar cualquier información significativa para el bloque. Cualquier coeficiente por debajo del valor de este límite es despreciado. Este es el punto en que toma lugar la pérdida de información. La transformación de los datos originales hacia la representación DCT es un proceso sin pérdida de información excepto por una pérdida menor por la precisión matemática. El paso que sigue a la cuantificación (formación de

palabras de código) es un paso de codificación sin pérdida de información. Así que esta es la única etapa donde hay oportunidad de descartar datos dentro de este esquema de compresión.

Al tomar a cada coeficiente de la matriz de cuantificación como el divisor del coeficiente de la matriz DCT correspondiente se obtiene un resultado como el mostrado en la figura 5.32.

Bloque después de la Cuantificación	61	-3	2	0	2	0	0	-1
	4	-4	2	0	0	0	0	0
	-1	-2	0	0	-1	0	-1	0
	0	0	1	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	-1	0	0	0	0	0
	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0

Figura 5.32. Matriz DCT resultante del proceso de cuantificación.

Codificación.

El paso final en el proceso JPEG es codificar la imagen cuantificada. La fase de codificación JPEG combina tres pasos distintos para realizar esta tarea. El primero cambia al coeficiente DC en 0,0 de un valor absoluto a un valor relativo. Como los bloques adyacentes en una imagen muestran un alto grado de correlación, aplicar la codificación predictiva como la revisada en la sección 2.2 a los coeficientes DC de cada bloque, regularmente produce una fuerte reducción en sus valores absolutos, como se mostrará durante la descripción de la codificación.

El segundo paso consiste en arreglar todos los coeficientes de la imagen en una secuencia Zig-Zag y codificarlos usando dos técnicas diferentes. La primera es la codificación run-length de los valores cero. La segunda es la codificación entrópica de los coeficientes diferentes de cero, ya sea con codificación Huffman o codificación aritmética, dependiendo de la selección del implementador.

La secuencia Zig-Zag es un tipo de reordenamiento de coeficientes que incrementa la efectividad de la compresión. En lugar de comprimir los coeficientes en un orden por renglones, como en el que se encuentran, el algoritmo JPEG se mueve a través del bloque a lo largo de sus franjas diagonales, seleccionando sus elementos de mayor valor primero, y dejando al final los de menor valor. La ruta de la secuencia Zig-Zag se muestra en la figura 5.33.

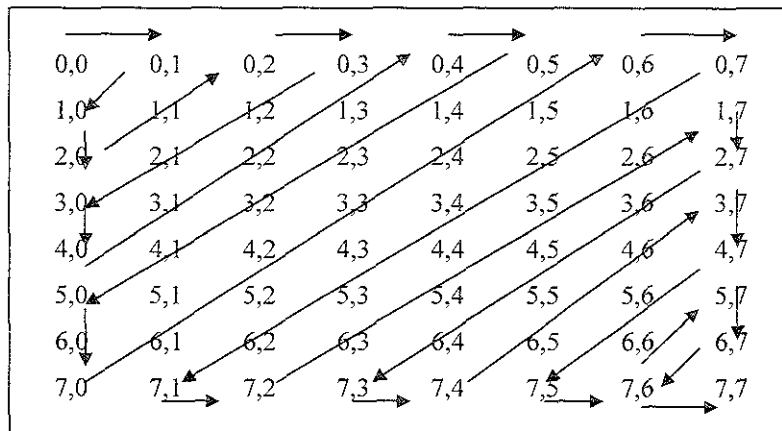


Figura 5.33. Secuencia de reordenamiento Zig-Zag del estándar JPEG FURHT, Bolko, SMOLIAR, Stephen. Video and image processing in multimedia systems).

El siguiente código implementa la secuencia Zig-Zag como parte de una estructura que puede accederse secuencialmente para determinar qué renglón y columna codificar⁴:

```
Struct ZigZag { int row;
               Int col; }      ZigZag [N*N] =
{
  (0,0),
  (0,1),(1,0),
  (2,0),(1,1),(0,2),
  (0,3),(1,2),(2,1),(3,0),
  (0,4),(1,3),(2,2),(3,1),(4,0),
  (0,5),(1,4),(2,3),(3,4),(4,3),(5,0),
  (6,0),(5,1),(4,2),(3,3),(2,4),(1,5),(0,6),
  (0,7),(1,6),(2,5),(3,4),(4,3),(5,2),(6,1),(7,0),
  (7,1),(6,2),(5,3),(4,4),(3,5),(2,6),(1,7),
  (2,7),(3,6),(4,5),(5,4),(6,3),(7,2),
  (7,3),(6,4),(5,5),(4,6),(3,7),
  (4,7),(5,6),(6,5),(7,4),
  (7,5),(6,6),(5,7),
  (6,7),(7,6),
  (7,7) };
```

A continuación se presenta el código que envía cada coeficiente DCT al codificador⁴. Nótese que en lugar de ubicar cada resultado, se determina qué renglón y columna usar y después se ubica en la estructura Zig-Zag. Posteriormente se codifica el elemento definido por el renglón y la columna de la estructura Zig-Zag.

```
For ( i = 0; i < ( N * N ); i ++ )
Row = ZigZag [ i ] . row;
Col = ZigZag [ i ] . col;
Result = DCT [ row ] [ col ] / Quantum [ row ] [ col ];
OutputCode ( output_file, ROUND ( Result ) ); }
```

⁴ NELSON. Mark. The data compression book. Estados Unidos. Ed. M & T Books. 1992.

El codificador Huffman convierte los coeficientes DCT después de la cuantificación en una secuencia binaria compacta utilizando dos etapas: (1) formar una secuencia de símbolos intermedios, y (2) convertir dicha secuencia en una secuencia binaria usando tablas Huffman.

En la secuencia de símbolos intermedia, cada coeficiente AC se representa por un par de símbolos[1]:

- Símbolo 1 (RUNLENGTH, TAMAÑO)
- Símbolo 2 (AMPLITUD)

RUNLENGTH es el número de coeficientes AC con valor cero consecutivos que preceden al coeficiente AC diferente de cero. Este valor está en el rango de $[0,15]$, el cual requiere 4 bits para su representación.

TAMAÑO es el número de bits usados para codificar la AMPLITUD. Este valor está en el rango de $[0,10]$, por lo que se necesitan 4 bits para su representación.

AMPLITUD es el valor del coeficiente AC diferente de cero en el rango de $[-1023,+1024]$, lo que requiere 10 bits para codificarse.

Ejemplo: 0,0,0,0,0,0,476 se codificaría como (6,9)(476) con RUNLENGTH=6, TAMAÑO=9 y AMPLITUD=476.

Si RUNLENGTH es mayor a 15, entonces el Símbolo 1 (15,0) es interpretado como un símbolo extensión con RUNLENGTH =16 y puede haber hasta tres extensiones (15,0) consecutivas.

Ejemplo: (15,0)(15,0)(7,4)(12) equivale a RUNLENGTH=16+16+7=39, TAMAÑO=4 y AMPLITUD=12.

El símbolo (0,0) significa “Fin de Bloque” y está al final de cada bloque de 8×8 .

Para los coeficientes DC la representación de símbolo intermedio consiste en:

- Símbolo 1 (TAMAÑO)
- Símbolo 2 (AMPLITUD)

Debido a que los coeficientes DC son codificados diferencialmente, al ser recuperados se requiere que su rango sea del doble del de los coeficientes AC, es decir $[-2048,+2047]$

La secuencia de símbolos intermedios del bloque de la figura 5.32 sería:

(6)(61),(0,2)(-3),(0,3)(4),(0,1)(-1),(0,3)(-4),(0,2)(2),(1,2)(2),(0,2)(-2),(0,2)(-2),
(5,2)(2),(3,1)(1),(6,1)(-1),(2,1)(-1),(4,1)(-1),(7,1)(-1)(0,0)

El segundo paso en la codificación Huffman es convertir la secuencia de símbolos intermedios a una secuencia binaria. En esta fase, cada Símbolo 1 (de los DC o los AC) se codifica utilizando un Código de Longitud Variable (VLC) que se obtiene a partir de el conjunto de tablas Huffman especificado para cada componente de la imagen. Los Símbolos 2 son codificados usando Código Entero de Longitud Variable (VLI), cuya longitud en bits se muestra en la tabla 5.9.

Longitud de Bits	Amplitud
1	-1, 1
2	-3 a -2, 2 a 3
3	-7 a -4, 4 a 7
4	-15 a -8, 8 a 15
5	-31 a -16, 16 a 31
6	-63 a -32, 32 a 63
7	-127 a -64, 64 a 127
8	-255 a -128, 128 a 255
9	-511 a -256, 256 a 511
10	-1023 a -512, 512 a 1023

Tabla 5.9. Longitudes de bits correspondientes a la codificación VLI de los Símbolos 2 (FURHT, Bolko, SMOLIAR, Stephen. Video and image processing in multimedia systems).

Ejemplo: para un coeficiente AC de la forma (1,4)(12) la representación binaria sería (1111101101100), donde (111110110) es el VLC obtenido de la tabla Huffman, y (1100) es el código VLI de longitud 4 para la amplitud 12.

La secuencia de bits del bloque ejemplo sería finalmente:

(110)(111101)(01)(00)(100)(100)(00)(0)(100)(001)(01)(10)(11011)(10)(01)(01)(01)(01)(111110111)(10)(111010)(1)(1111011)(0)(11100)(0)(111011)(0)(11111010)(0)(1010).

Descompresión JPEG secuencial.

En la decodificación JPEG secuencial, todos los pasos del proceso de codificación se invierten y se implementan en ese sentido. Primero, se aplica el decodificador entrópico en los datos de imagen comprimidos. La secuencia binaria se convierte a una secuencia de símbolos de coeficientes VLC y VLI usando tablas Huffman, y después los símbolos se convierten a coeficientes DCT. Posteriormente se aplica la decuantificación de los coeficientes DCT usando la siguiente función:

$$F_q'(u, v) = F_q(u, v) \times Q(u, v)$$

donde $Q(u, v)$ son los coeficientes de cuantificación correspondientes de la matriz de cuantificación.

Después se aplica la Transformada de Coseno Discreta Inversa (IDCT) sobre los coeficientes decuantificados para convertir la imagen del dominio de frecuencia al dominio espacial. Esta operación se realiza mediante la operación de matrices expresada como:

$$P' = C_t \cdot DCT \cdot C$$

donde C es la matriz de transformada DCT, C_t es su transpuesta, DCT es la matriz de coeficientes transformados y P' es la matriz de valores de pixel recuperados del proceso de compresión JPEG.

El último paso consiste en desplazar el valor de las muestras descomprimidas de vuelta al rango de $[0, 2^p-1]$.

5.5.3. Compresión JPEG progresiva basada en DCT.

Cuando se codifica una imagen por medio del método secuencial JPEG, no es posible ver la imagen completa recuperada hasta que todos los datos codificados se hayan recibido. Sin embargo, con el método de codificación JPEG progresivo basado en DCT es posible obtener una versión de la imagen completa en una forma de resolución reducida preliminar antes de recibir la imagen refinada final. Este modelo utiliza una secuencia de exploraciones, cada una de las cuales codifica una parte de la información de los coeficientes DCT cuantificados. Existen dos formas complementarias de realizar esta operación, la selección espectral y la aproximación sucesiva. En la selección espectral los coeficientes DCT son agrupados en bandas espectrales de frecuencias espaciales relacionadas y se envían primero las bandas de frecuencia más baja. En la aproximación sucesiva, los coeficientes DCT se dividen entre una potencia de 2 antes de codificarse. En el decodificador, los coeficientes DCT se multiplican por esa misma potencia de 2 antes de efectuar la IDCT. Primero se envía la información con menor precisión y después se refina en exploraciones posteriores. La figura 5.35 ilustra estos dos procedimientos de codificación progresiva.

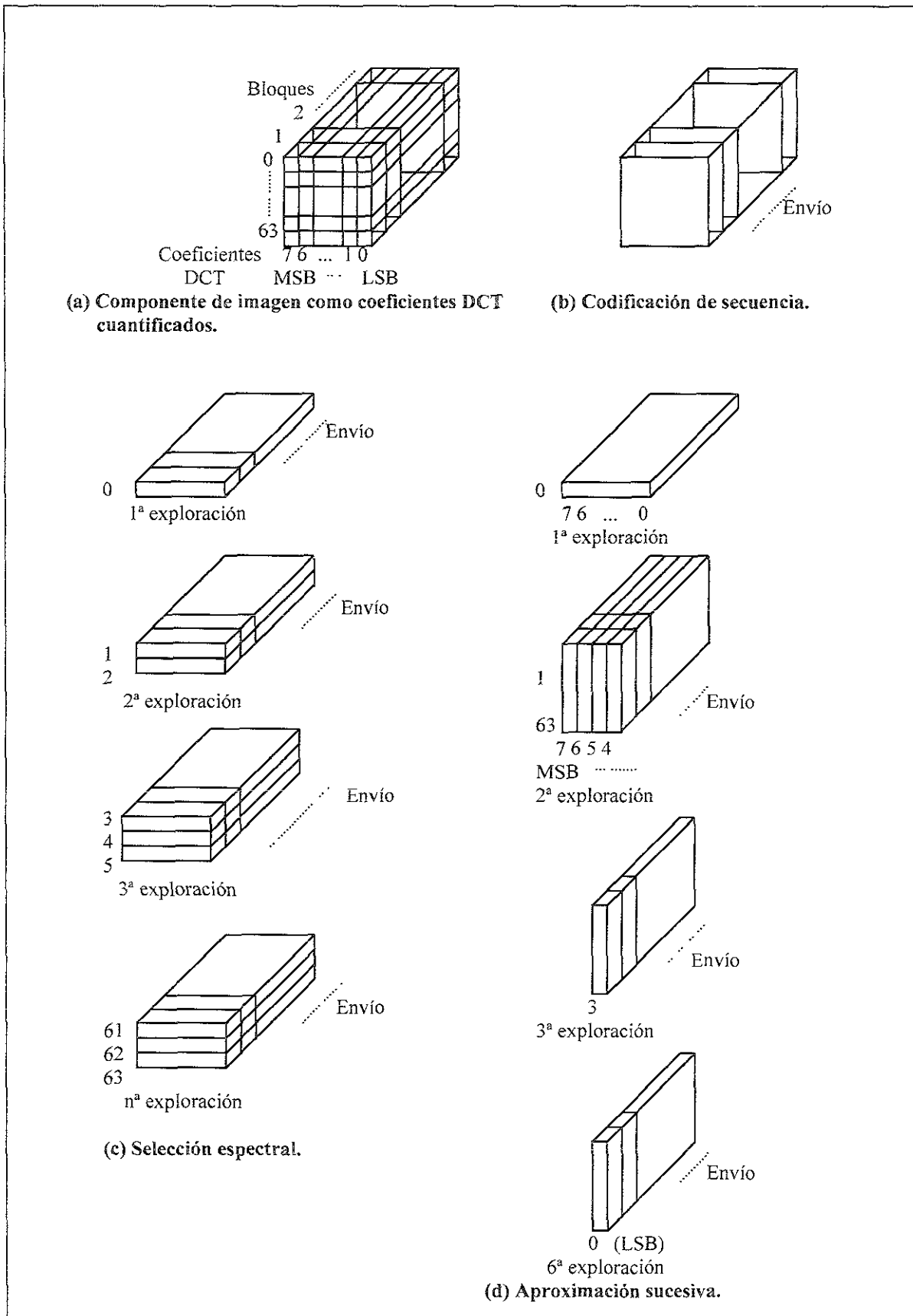


Figura 5.35. Procesos progresivos de selección espectral y de aproximación sucesiva.

Usando la selección espectral dentro de la aproximación sucesiva se logra implementar una progresión completa. Sin embargo, tanto el codificador como el decodificador utilizan únicamente la selección espectral.

El estándar JPEG define ocho procesos de codificación progresivos basados en DCT:

- Únicamente selección espectral, codificación Huffman, con precisión de 8 bits/muestra.
- Únicamente selección espectral, codificación aritmética, con precisión de 8 bits/muestra.
- Progresión completa, codificación Huffman, con precisión de 8 bits/muestra.
- Progresión completa, codificación aritmética, con precisión de 8 bits/muestra.
- Únicamente selección espectral, codificación Huffman, con precisión de 12 bits/muestra.
- Únicamente selección espectral, codificación aritmética, con precisión de 12 bits/muestra.
- Progresión completa, codificación Huffman, con precisión de 12 bits/muestra.
- Progresión completa, codificación aritmética, con precisión de 12 bits/muestra.

Como se ilustra en la figura 5.35, si se escoge la selección espectral, la secuencia Zig-Zag de los coeficientes DCT se divide en bandas. En el encabezado de la exploración, se especifican los índices inicial y final en la secuencia Zig-Zag para una banda dada. Los coeficientes DC y AC siempre se codifican separadamente. Las exploraciones que codifican los coeficientes DC pueden tener bloques intercalados de múltiples componentes. El resto de las exploraciones sólo tienen un componente. Para cada componente, cualquiera de las exploraciones AC deben seguir de la primera exploración DC.

En la aproximación sucesiva, los coeficientes DCT se reducen en su precisión. Ésta operación se realiza por medio de la transformada de punto especificada en el encabezado de la exploración. Una transformada de punto es un proceso de escalado de una muestra o coeficiente DCT. En la aproximación sucesiva hay dos parámetros para la transformada de punto, A_l y A_h . El parámetro de posición de bit A_l especifica la transformada de punto actual, y los cuatro bits superiores (A_h) contienen el valor de la transformada de punto usado en las exploraciones anteriores. A_h es igual a cero si no hay exploraciones sucesivas para la banda. Después de la primera exploración para una banda dada, cada exploración adicional mejora la precisión de los coeficientes de forma progresiva. Se logra la precisión completa después de haber recibido todas las exploraciones.

5.5.4. Compresión JPEG sin pérdida.

Los modos JPEG basados en DCT son técnicas de compresión con pérdida de información ya que la cuantificación y la limitación de la precisión realizadas después de la DCT introducen distorsiones en la imagen resultante. Después de capturar una imagen digital, por ejemplo, existen algunas aplicaciones de la medicina o las ciencias espaciales que requieren que la imagen se codifique sin pérdida de información. En el estándar JPEG se define un modo de operación sin pérdida. El estándar JPEG especifica los dos procesos de codificación siguientes para un modo de operación sin pérdida:

- Proceso sin pérdida con codificación Huffman.
- Proceso sin pérdida con codificación aritmética.

El modo JPEG sin pérdida utiliza para la compresión una técnica de codificación predictiva. Las muestras colindantes reconstruidas se usan para predecir la muestra actual en un proceso similar al que se ilustra en la figura 5.36.

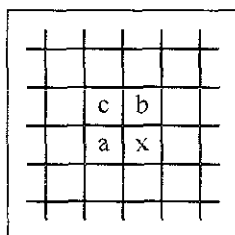


Figura 5.36. Muestras de imagen utilizadas en la predicción sin pérdida JPEG (x es la muestra actual).

En la figura 5.36 hay cuatro muestras de imagen marcadas como a, b, c y x. La cuestión es predecir la muestra de imagen x basándose en a, b, c o sus combinaciones. Si la predicción de x se denota por P_x , existen ocho ecuaciones de predicción especificadas en el estándar JPEG las cuales se muestran en la tabla 5.10.

Número de la Selección	Ecuación de Predicción
0	sin predicción
1	$P_x = a$
2	$P_x = b$
3	$P_x = c$
4	$P_x = a + b - c$
5	$P_x = a + (b - c) / 2$
6	$P_x = b + (a - c) / 2$
7	$P_x = (a + b) / 2$

Tabla 5.10. Ecuaciones de predicción para la codificación sin pérdida JPEG (KOU, Weidong. Digital image compression. Algorithms and standards).

La selección 0 se usa solamente para codificación diferencial en el modo de operación jerárquico JPEG. Las selecciones 1 a 3 son predicciones unidimensionales y las selecciones 4 a 7 son predicciones bidimensionales.

En la tabla 5.10 la selección 1 es una predicción horizontal unidimensional. Se utiliza para la primera línea de muestras al inicio de la exploración y al comienzo de cada intervalo de reinicio de exploración. Una vez que se ha seleccionado una predicción, ésta se utiliza para el resto de las líneas.

Para una precisión de dato de entrada dada P y el parámetro de transformada de punto P_t , el valor de predicción al comienzo de la primera línea y al comienzo de cada intervalo de reinicio es 2^{P-P_t-1} . La diferencia entre la predicción “y” y la entrada “z” se calcula con una operación modulo: $(y-z)\text{modulo } 2^{16}$. En el decodificador, se agrega la diferencia, modulo 2^{16} , a la predicción.

Después de la predicción, el codificador sin pérdida JPEG puede utilizar ya sea codificación Huffman o codificación aritmética para la codificación entrópica. El procedimiento de codificación Huffman para codificación sin pérdida es el mismo que se definió para la codificación de los coeficientes DC en el JPEG básico, con excepción de que se agregan más entradas para codificar las diferencias de las predicciones. La figura 5.37 muestra un diagrama del sistema de codificación sin pérdida JPEG simplificado.

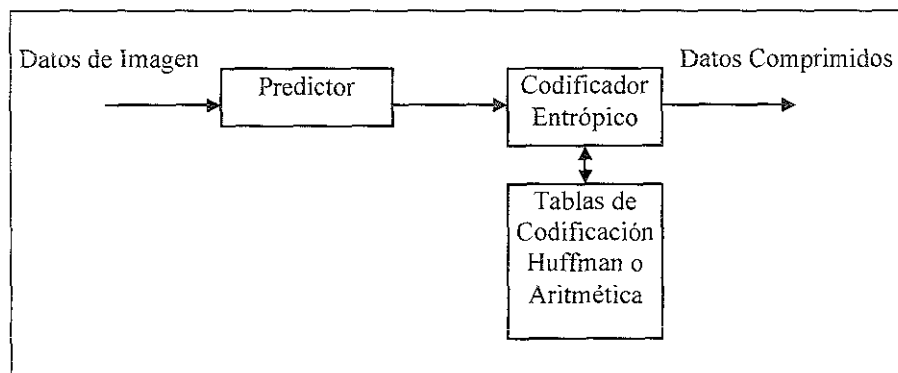


Figura 5.37. Diagrama del sistema de codificación sin pérdida JPEG (KOU, Weidong. Digital image compression. Algorithms and standards).

5.5.5. Compresión JPEG jerárquica.

El modo jerárquico JPEG proporciona codificación progresiva con incremento en la resolución espacial entre las etapas progresivas. La codificación de primera etapa utiliza uno de los modos JPEG ya sea secuencial o progresivo. La salida de cada etapa jerárquica

se procesa con un filtro de muestreo ascendente y ésta se convierte en la predicción de la siguiente etapa. El filtro de muestreo ascendente duplica la resolución espacial horizontal o verticalmente, o en ambas formas si es necesario. El modo jerárquico JPEG también puede utilizar un filtro de muestreo descendente el cual reduce la resolución espacial horizontal o verticalmente o en ambas formas. En el estándar JPEG se especifica un filtro de muestreo ascendente pero no se especifica uno de muestreo descendente. Los filtros de muestreo descendente y de muestreo ascendente pueden utilizarse para generar una pirámide de resolución espacial como la de la figura 5.38.

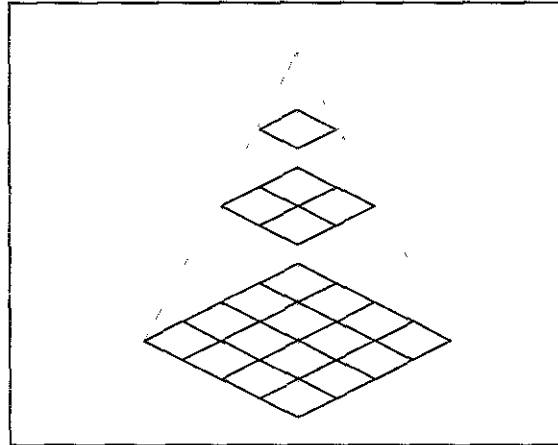


Figura 5.38. Pirámide para la codificación jerárquica (KOU, Weidong. Digital image compression. Algorithms and standards).

La codificación jerárquica puede aplicarse a ambientes que requieran múltiples resoluciones. Por ejemplo, cuando se requiere del uso de imágenes a múltiples resoluciones desde un servidor de imágenes en diferentes dispositivos de imagen. Con la codificación jerárquica, un servidor puede proporcionar varias resoluciones para distintos clientes y aplicaciones.

El modo de operación jerárquico tiene 14 procesos los cuales se listan a continuación:

- Secuencial extendido con codificación Huffman para precisión de 8 bits.
- Secuencial extendido con codificación aritmética para precisión de 8 bits.
- Secuencial extendido con codificación Huffman para precisión de 12 bits.
- Secuencial extendido con codificación aritmética para precisión de 12 bits.
- Selección espectral solamente con codificación Huffman para precisión de 8 bits.
- Selección espectral solamente con codificación aritmética para precisión de 8 bits.
- Progresión completa con codificación Huffman para precisión de 8 bits.
- Progresión completa con codificación aritmética para precisión de 8 bits.
- Selección espectral solamente con codificación Huffman para precisión de 8 bits.
- Selección espectral solamente con codificación aritmética para precisión de 8 bits.

- Progresión completa con codificación Huffman para precisión de 8 bits.
- Progresión completa con codificación aritmética para precisión de 8 bits.
- Sin pérdida (DPCM) con codificación Huffman para precisión de 12 a 16 bits.
- Sin pérdida (DPCM) con codificación aritmética para precisión de 12 a 16 bits.

En la codificación jerárquica, cada componente se codifica por medio de una secuencia de cuadros. Los cuadros pueden ser diferenciales o no diferenciales. Los cuadros diferenciales se usan para codificar las diferencias complementarias entre los componentes de entrada fuente y los componentes de referencia después del muestreo ascendente o muestreo descendente necesario. El orden de secuencia de los cuadros consiste en tener primero un cuadro no diferencial, seguido por cuadros diferenciales.

En el estándar JPEG se utiliza un elemento marcador llamado DHP (define proceso jerárquico) para indicar que una imagen se codifica en el modo jerárquico JPEG. El marcador tiene que ubicarse antes del primer segmento de inicio de cuadro en los datos de la imagen comprimida.

El procedimiento de codificación es el siguiente:

- Paso 1. Generar imágenes muestreadas de forma descendente si se requiere; agregar el marcador SOI (inicio de imagen); agregar las tablas y otros segmentos marcadores varios si se requiere; agregar el segmento marcador DHP.
- Paso 2. Si es un cuadro diferencial, ir al paso 3; de otra forma, ir al paso 4.
- Paso 3. Muestrear de forma ascendente los componentes de referencia y agregar un marcador de segmento EXP(componente de referencia de expansión) si se requiere; generar componentes diferenciales y codificar el cuadro diferencial; reconstruir los componentes diferenciales, y reconstruir los componentes para el cuadro subsecuente. Ir al paso 5.
- Paso 4. Codificar el cuadro, reconstruir los componentes usando el proceso de acoplamiento del decodificador.
- Paso 5. Si hay más cuadros, regresar al paso 2; de otra forma, agregar el marcador EOI (fin de la imagen).

El procedimiento de decodificación es el siguiente:

- Paso 1. Si el primer marcador no es el marcador SOI, parar la decodificación y reportar el error. Si no es así, ir al paso 2.

- Paso 2. Interpretar los marcadores. Si el marcador es el marcador EOI, se ha terminado la decodificación. Si la bandera de decodificación no está configurada como “jerárquica” (que el marcador DHP se encuentre antes del primer cuadro), efectuar el modo de decodificación no jerárquico; de otra forma, pasar al paso 3.
- Paso 3. Si es un cuadro diferencial, muestrear de forma ascendente el componente de referencia si es necesario, decodificar el cuadro diferencial y reconstruir los componentes; de otra forma, decodificar el cuadro.
- Paso 4. Regresar al paso 2.

5.6. ÁREAS DE APLICACIÓN DE LA COMPRESIÓN DE IMÁGENES.

La compresión de datos de imágenes encuentra aplicación en diversos sectores de la actividad comercial e industrial actual, principalmente en los relacionados con la generación de productos de contenido o tratamiento de información visual orientados hacia un observador humano. Ejemplos de estos sectores son la fabricación de equipo de computo o electrodoméstico destinado a la captura y tratamiento de imágenes digitales, la elaboración de documentos impresos o electrónicos que contengan información visual o el análisis y almacenamiento de registros gráficos distribuidos en diversas localidades y que requieran de alta disponibilidad. El que sea posible reducir la cantidad de información de las imágenes sin afectar su apariencia implica un fuerte ahorro de recursos como dispositivos de almacenamiento, tiempo de transmisión a través de una red, tiempo de despliegue en pantallas o tiempo de ocupación de los procesadores. Cada uno de estos ahorros se traduce en menos inversión para equipo técnico como en la creación de nuevas posibilidades y métodos de trabajo basados en el uso de las imágenes digitales. A continuación se describen algunas aplicaciones de equipos y procesos que utilizan la tecnología de la compresión de datos de imagen como herramienta importante para extender sus capacidades y beneficios. Estas aplicaciones se enfocan hacia los sectores de la fabricación de equipo de computo, de la transmisión de información de imágenes a través de redes de computadoras, de la modernización de aparatos electrodomésticos, de los procesos de producción de la industria editorial y de la tecnología de imágenes aplicada al cuidado de la salud.

5.7. APLICACIONES COMPUTACIONALES.

5.7.1. Computadoras multimedia.

A principios de los años 90, surgió un nuevo tipo de computadora, la computadora multimedia. Este tipo de computadoras están diseñadas para manejar voz, audio, imagen y video de forma digital, y para ello trabajan tanto con técnicas de codificación sin pérdida como con técnicas de codificación con pérdida de información. Las técnicas de compresión son un factor decisivo para la creación y distribución de estos equipos tanto a nivel empresarial como doméstico, en virtud de la gran cantidad de datos que la representación de estos tipos de información conlleva. Las computadoras multimedia, denominadas también sistemas de imágenes de documentos, tienen actualmente un papel importante como herramienta de trabajo en una amplia gama de campos que ocupan datos difusos y en especial imágenes. Además, también los equipos llamados servidores multimedia, destinados a la distribución y almacenamiento de los datos difusos, efectúan tareas de compresión y descompresión para realizar sus operaciones. A continuación se exponen las características de ambos equipos y la forma en que la compresión de datos influye en su funcionamiento.

5.7.2. Sistemas de imágenes de documentos.

A comienzos de los años 80, mucho antes de que se usará el término computadora multimedia, los sistemas de imágenes de documentos por computadora ya se utilizaban en las grandes empresas y en el gobierno para automatizar los procesos que manejaban grandes cantidades de documentos filmicos y en papel. La información de estos documentos era capturada explorándola con un scanner y creando archivos de imagen que eran almacenados en discos de computadora. Por medio del uso de OCR (reconocimiento óptico de carácter), lectura de código de barras o de la captura por tablero se organizaban dichos documentos para su recuperación posterior. Estos pasos permitían utilizar las herramientas que proporcionaban las computadoras para administrar, almacenar, organizar, recuperar y procesar documentos.

Los sistemas de imágenes de documentos originalmente eran sistemas únicos en su tipo creados para aplicaciones específicas. Muy frecuentemente, eran soluciones autónomas que no integraban otras actividades de procesamiento de datos. A finales de los años 80, los fabricantes de computadoras comenzaron el proceso de incluir herramientas para manejo de imágenes de documento en los equipos. Esto resultó en la producción y distribución de miles de sistemas de imágenes al rededor del mundo. Las mainframes, minicomputadoras y estaciones de trabajo comenzaron a combinarse con componentes especializados de despliegue, scanners e impresoras de alta resolución y formato de gran capacidad. En estos

sistemas se utilizaba tecnología de punta de alto costo, por lo que sólo empresas que manejaran un gran volumen de imágenes podían pagarlos. Posteriormente llegó la revolución de la PC de los años 90, revitalizando totalmente los sistemas de imágenes de documentos. Ahora, a menudo las imágenes de documentos no son más que otra aplicación donde la imagen y el texto coexisten en bases de datos convencionales dentro de computadoras tradicionales. No sólo las imágenes pueden ser procesadas y desplegadas en PC's estándares, sino que los scanners y las impresoras también son accesibles y de bajo costo. Los sistemas de documentos de hoy en día son más rápidos, más baratos, mejor integrados y más estandarizados. Ahora cubren todo el espectro desde los sistemas de gran volumen de información y tecnología de punta hasta los sistemas de escritorio de uso ocasional.

En su sentido estricto, los sistemas de imágenes de documentos involucran la exploración de papel o película y su conversión a formato digital. Sin embargo en la práctica, el mismo sistema puede incluir imágenes de FAX recibidos vía electrónica. También puede incluir imágenes producidas por programas de computadora tales como dibujos de programas de diseño asistidos por computadora. Las fotografías, tanto las exploradas como las generadas electrónicamente pueden estar incluidas igualmente. Algunos ejemplos de aplicaciones de imágenes de documentos incluyen los sistemas de vigilancia de bancos, procedimientos de reclamación para compañías de seguros, documentos legales de los tribunales, y registros del gobierno tales como registros de arresto o licencias de manejo. También, manuales y dibujos para organizaciones de servicio y diseño, almacenes de publicaciones electrónicas como fotografías de periódicos.

Para entender el papel que la compresión de datos juega en los sistemas de imágenes de documentos, considere el sistema que se ilustra en la figura 5.39.

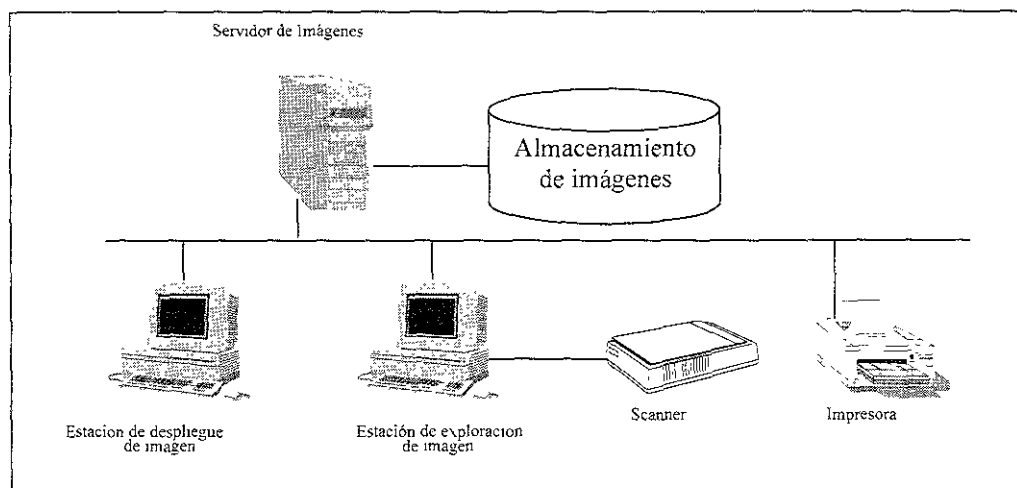


Figura 5.39. Sistema de imágenes de documentos (HOFFMAN, Roy. Data compression in digital systems).

Los documentos se cargan al sistema explorándolos en un formato electrónico en una estación de exploración de imágenes. Se aplica la compresión de datos y se utiliza la OCR, la lectura de código de barras o la captura manual por teclado para capturar la información textual e indexar el documento. El proceso de carga de la imagen también incluye operaciones manuales para asegurar la calidad de la imagen explorada y la exactitud de la información de texto. La información del índice debe crearse de forma tal que posteriormente el documento pueda ser recuperado. Cuando los datos de la imagen dejan la estación de exploración, se envían al servidor de imágenes donde serán almacenados. Después, cuando sea necesario, los datos de la imagen serán recuperados desde el servidor y enviados a la estación de despliegue para visualizarlos o a la impresora para generar una copia en papel. Otros tipos de imágenes que pueden incluirse en el sistema vienen de otras fuentes como los servidores de FAX, estaciones de trabajo de diseño asistido por computadora o de módulos de imágenes médicas. Todos estos elementos de sistema están conectados por redes de comunicación de área local o amplia.

Si bien conceptualmente es simple, los sistemas de imágenes de documentos del mundo real requieren de un cuidadoso diseño para mejorar la productividad de la gente que los utiliza al tiempo que se procesan grandes cantidades de datos. Por ejemplo, a menudo es aceptable permitir que el sistema ocupe varios segundos para recuperar una carpeta electrónica de imágenes de documentos de bienes raíces antes de presentarlas a algún cliente. Sin embargo, cuando la carpeta se presenta en la estación de trabajo, el cliente esperará el despliegue de las páginas en fracciones de segundo para que mejore la búsqueda de un documento en una carpeta real. Para hacer que el sistema sea útil, deben aplicarse técnicas sofisticadas de administración de sistemas para asegurar que los requerimientos de los usuarios se satisfagan. Aquí, el importante papel que juega la compresión de datos se vuelve evidente. Aún estando comprimida, una imagen puede ser de 30 a 90 Kbytes, muchas veces mayor que los registros de base de datos necesarios para las transacciones de negocios típicas. Muchas aplicaciones contienen millones de documentos, los cuales requieren enormes cantidades de almacenamiento, y es necesaria una jerarquía de almacenamiento multinivel que consiste de al menos un elemento de almacenamiento principal, discos magnéticos y discos ópticos en línea. Los archivos de imagen deben arreglarse dentro de la jerarquía de almacenamiento de forma tal que los archivos más activos se ubiquen en los dispositivos de almacenamiento más rápidos. Para asegurar que los usuarios en las estaciones de despliegue puedan recuperar y ver las imágenes con tiempos de respuesta aceptables, deben usarse técnicas de estratificación y monitoreo de los archivos. Los tiempos de transmisión y el ancho de banda de la red también son factores críticos. Para reducir el tráfico en la red, se aplica la compresión de datos en la estación de exploración para que las imágenes comprimidas sean enviadas al servidor de imágenes para su almacenamiento. Para reducir aún más el tráfico de la red y mejorar el tiempo de respuesta, las imágenes comprimidas se envían a la estación de despliegue de imagen y a las impresoras y ahí se descomprimen. Es posible usar software

de PC para implementar completamente la compresión y descompresión de imágenes, ya sea para explorar, desplegar o imprimir las imágenes. Más comúnmente, los scanners contienen funciones de compresión y las impresoras contienen funciones de descompresión, ya sea de fábrica o agregadas como tarjetas de conexión a puerto.

Un amplio rango de opciones de compresión de datos y formatos se utilizan en los sistemas de administración de documentos. El almacenamiento requerido para una imagen comprimida con JPEG, una imagen de FAX o cualquier imagen sin comprimir varía desde algunos miles hasta varios millones de bytes. No sólo existen estos tipos de imágenes con distintos requerimientos de compresión, pero si son los que comúnmente se encuentran en los sistemas de distintas plataformas. Para decidir que clase de técnicas de compresión utilizar, los diseñadores deben tomar en cuenta factores tales como:

- Los diversos estándares que varían de aquellos que se aplican en sistemas de producción de alta tecnología hasta aquellos que se utilizan en sistemas basados en PC de uso casual.
- Los tipos de imágenes tan variados que van desde los FAX hasta las fotografías.
- Muchas representaciones de imagen, incluyendo las binivel, las de escala de gris, y las de calidad fotográfica a color; para las imágenes a color, el RGB el YUV u otros modelos; también, varias profundidades de pixel con color tanto a 8 como a 24 bits.
- Varias resoluciones de imagen que varían desde los 100 o 200 pixeles por pulgada para FAX, hasta los 300, 600, 1200 o más pixeles por pulgada para imágenes complejas.

Todo esto conforma un mismo conjunto a considerar cuando se explora una imagen y se crea una representación electrónica de la misma. Cada imagen explorada se almacena en un archivo de datos de mapa de bits. Cada pixel de la imagen se representa por un bit o un grupo de bits en ese archivo. El número de bits por pixel no sólo depende de los factores anteriores, si también de la estructura del archivo y de la técnica de compresión.

Un ejemplo para ilustrar la diversidad de las técnicas de compresión que se encuentran en un sistema de imágenes de documento es examinar un formato de archivo usado comúnmente, el TIFF (Formato de Archivo Identificado como Imagen). El TIFF soporta varias resoluciones de imagen y representaciones de color. Soporta imágenes sin comprimir y seis métodos de compresión. Cinco sin pérdida y uno con pérdida de información. Los algoritmos de compresión sin pérdida incluyen codificación run-length, MH, MR y MMR del CCITT T.4 y una versión del LZW para imágenes a color. El algoritmo con pérdida es el JPEG secuencial. Así, el saber simplemente que una imagen está almacenada como archivo TIFF dice poco; se requiere la decodificación extensa del encabezado del archivo para saber cómo es la imagen y como está comprimida.

La diversidad de los formatos de archivos y de los esquemas de compresión de imágenes hace que el intercambio de datos entre los sistemas de imágenes de documento sea difícil, ya que se requiere una fuerte inversión en software y herramientas. Para las organizaciones que buscan unificar el manejo de imágenes, una solución es convertir todas las imágenes a un formato de archivo común. Sin embargo, hoy en día esto puede no asegurar la eliminación de los problemas de operación con imágenes. Continuando con el formato TIFF como ejemplo, muchos productos para imágenes soportan este formato de archivo; pero la especificación de archivo TIFF es compleja, con muchas variaciones y opciones, dando lugar a errores y omisiones en su interpretación. El resultado es que se han desarrollado diferencias sutiles entre las diversas implementaciones comerciales del TIFF. Con el fortalecimiento o la simplificación de los estándares de archivos de imagen, y la migración hacia menos estándares de compresión elaborados con mayor cuidado como el JPEG, posiblemente se solucione la problemática entre sistemas de imágenes de documento.

5.7.3. Servidores multimedia.

Los servidores multimedia deben almacenar y manipular grandes cantidades de datos de voz, audio, imagen y video, haciendo de la compresión con pérdida un elemento necesario para controlar el costo de la capacidad de almacenamiento. Estas computadoras se distinguen de las tradicionales debido a que deben manejar los tipos de datos difusos anteriores en tiempo real. Los servidores multimedia se dirigen hacia segmentos de mercado comercial fuera del alcance de las computadoras comunes. Los servidores multimedia se encuentran en redes de computadoras cliente-servidor donde se utilizan para aplicaciones de negocios más o menos tradicionales. Pero también se usan para aplicaciones de entretenimiento, para almacenamiento de datos digitales que las redes de comunicaciones y canales de transmisión envían a los televisores y a una variedad de dispositivos orientados hacia el consumidor. De hecho, las aplicaciones al entretenimiento son responsables del desarrollo de los servidores multimedia. La tabla 5.11 lista algunas de las aplicaciones existentes y potenciales para los servidores multimedia.

Aplicaciones a Negocios	Aplicaciones a Entretenimiento
<ul style="list-style-type: none"> - Entrenamiento de empleados - Bases de datos multimedia - Interconexión para videoconferencia - Bases de datos para monitoreo de seguridad - Servidores para kioskos interactivos - Servidores multimedia para la World Wide Web de Internet 	<ul style="list-style-type: none"> - Video en línea - Televisión interactiva - Compras desde el hogar - Juegos de video para varios jugadores - Servidores para producción de medios - Servidores de audio y video para la industria de los medios de comunicación

Tabla 5.11. Aplicaciones de los servidores multimedia (HOFFMAN, Roy. *Data compression in digital systems*).

Debido a que los servidores multimedia principalmente manejan datos difusos, su operación difiere de las computadoras tradicionales a un nivel tal que se requieren cambios significativos en su diseño. La diferencia clave es que deben almacenar y recuperar flujos de datos de voz, audio, imagen y video en tiempo real, continuamente y sin interrupción. Esto implica la transferencia de grandes cantidades de datos en forma casi inmediata. Los servidores multimedia deben conectarse a redes de alta velocidad. Deben manejar jerarquías de almacenamiento cuya capacidad, en su caso más limitado, debe ser igual o mayor que cualquier computadora convencional de gran capacidad. El construir un verdadero servidor multimedia requiere nuevos hardware y software. Algunos de los cambios en hardware incluyen subsistemas de entrada/salida que puedan trabajar con redes de alta velocidad, dispositivos de almacenamiento en disco magnético que puedan proporcionar no sólo altas tasas de transferencia de datos, sino el soporte necesario para sostener dichas tasas durante largos períodos de tiempo, además de buses internos de amplio ancho de banda para conjuntar todos estos elementos. Los cambios en el software incluyen sistemas de archivos que proporcionen asignación de memoria en tiempo real y optimización del desempeño para el manejo de grandes bloques de datos, a menudo para un gran número de usuarios. Otros cambios en el software incluyen técnicas de reserva de ancho de banda para asegurar tiempos de respuesta consistentes, técnicas para sincronizar la entrega de datos, y administración jerárquica del almacenamiento y técnicas de manejo de buffers que anticipen la demanda continua de voz, audio, imagen y video con un retraso aceptable.

La cuestión es en qué forma encaja la compresión de datos en el diseño de los servidores multimedia. Obsérvese el caso de dos clases de servidores multimedia: los servidores de almacenamiento y despliegue y los servidores multifuncionales. Los servidores de almacenamiento y despliegue esencialmente actúan como controladores de almacenamiento inteligentes y como servidores de base de datos para voz, audio, imagen y video. Se utilizan en aplicaciones donde los datos utilizados para reproducirse en las terminales cliente llegan al servidor ya en formato comprimido. Estos datos se cargan en el servidor y se envían a los clientes donde se descomprimen durante la reproducción. La figura 5.40 muestra una aplicación de servidor de video típica.

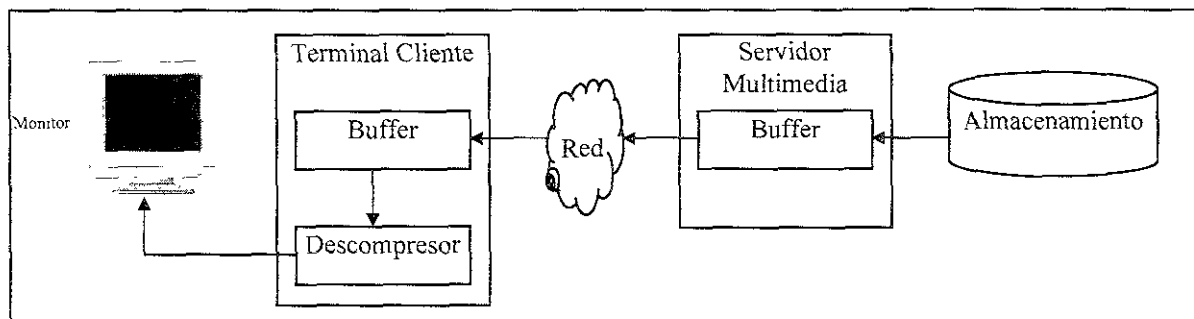


Figura 5.40. Aplicación de servidor de video de almacenamiento y despliegue (HOFFMAN, Roy. *Data compression in digital systems*).

En las aplicaciones de almacenamiento y despliegue, el servidor se encarga de proporcionar al buffer de flujo de cada terminal cliente los datos suficientes para asegurar que el proceso de reproducción no carezca de alimentación. Para realizar esto eficientemente, sólo deben enviarse desde la jerarquía de almacenamiento del servidor la cantidad justa de datos para enviarse por la red. El servidor principalmente administra la base de datos y el tráfico de la red que conecta a los clientes. La compresión de datos afecta a los servidores multimedia de almacenamiento y despliegue en las siguientes formas:

Asúmase que el cliente está reproduciendo datos a una tasa constante, como el video a 30 cuadros por segundo. Cuando los datos están comprimidos con un algoritmo de compresión de tasa variable como el MPEG, la cantidad de bloques de datos que deben recuperarse cambiará de acuerdo con el número de cuadros de video haya empacado el compresor en cada bloque. Obviamente, esto causa un problema de sincronización para el servidor en cuanto a que no puede decidir qué tan seguido enviar un bloque de datos al cliente. Una solución es que el servidor interactúe con el algoritmo de compresión y conozca cuantos cuadros de video contiene cada bloque. Entonces puede etiquetar cada bloque con esta información para sincronizar la entrega de datos con su reproducción. La misma información también es útil para cargar los datos en la jerarquía de almacenamiento del servidor, arreglando los bloques de datos en los dispositivos de almacenamiento de forma tal que facilite la recuperación de un número constante de cuadros de video desde varios bloques de datos.

En el caso de los servidores multifuncionales, éstos se distinguen por estar involucrados activamente en el procesamiento de datos multimedia, haciendo más que sólo almacenarlos, indexarlos y canalizarlos. Algunos ejemplos de las operaciones que realizan estos servidores son las siguientes:

- Conectar y canalizar el audio y video hacia los clientes que participen en una videoconferencia múltiple. Esto puede requerir funciones de compresión y descompresión para administrar y entregar audio y video en diferentes formatos y resoluciones para clientes con distintas capacidades.
- Efectuar la compresión inicial de los datos difusos para permitir su captura directa por servidores usados para producción y distribución de medios. Sin embargo, hoy en día estos datos usualmente se comprimen en estaciones de trabajo que controlan también los dispositivos de captura de datos o en las estaciones de trabajo que se usan para la edición de medios. Esta forma de trabajo seguirá siendo popular debido a que es económica y proporciona a los desarrolladores de medios un control absoluto del proceso.

- Realizar la descompresión para permitir las operaciones de procesamiento sobre los objetos en las bases de datos administradas por el servidor. Estas operaciones pueden incluir interpretación semántica del contenido de los mensajes de voz, audio, imagen o video.
- Efectuar reconocimiento óptico de caracteres de documentos de imagen descomprimidos, permitiendo que las porciones de texto de dichos documentos sean almacenados y procesados junto con otros datos simbólicos.
- Proporcionar compresión y descompresión para datos gráficos y de imagen junto con redimensionamiento, reescalado y conversiones de formato necesarias para controlar los servidores de impresión compartidos que generan documentos compuestos, mapas geográficos e imágenes binarias.

El mercado de los servidores multimedia multifuncionales aún no está bien establecido, como tampoco sus requerimientos ni especificaciones. Por lo tanto, la forma en que se usará la compresión y descompresión de datos tampoco se ha definido de forma precisa.

5.8. APLICACIONES PARA RED.

5.8.1. Facsímile.

Las aplicaciones de negocios para el facsímile comenzaron a aparecer en los años 60, pero los únicos aparatos de FAX disponibles entonces usaban tecnología analógica, eran de costo elevado, de baja velocidad y de operación compleja. La cantidad de datos digitales era demasiada para transmitirse a través de líneas telefónicas ordinarias. El transmitir una página de 8½x11 pulgadas requería 6 minutos. Algunas mejoras en los años 70 redujeron el tiempo casi a la mitad, pero la transmisión seguía siendo muy lenta y cara a través de las líneas telefónicas. Gracias al desarrollo de algoritmos de compresión efectivos, se logró transmitir hasta dos páginas por minuto con la adopción en los años 80 del estándar Grupo 3 del CCITT o ITU, para codificar, comprimir y transmitir facsímile digital. Sin embargo, en ese entonces, la tecnología digital era de alto costo pero los precios fueron cayendo rápidamente. A mediados de los años 80, pasaron de ser equipos poco comunes a herramientas de uso diario en los negocios.

Los estándares para transmisión y compresión de datos son importantes para el mercado del FAX. El estándar Grupo 3 de hoy en día se aplica en todos los equipos para transmisión por facsímile digital de imágenes binivel a través de la PSTN (Redes de Telefonía de Conexión Pública) con las facilidades de la transmisión analógica. Este estándar especifica las operaciones de exploración, codificación (compresión), transmisión e impresión para los equipos dirigidos a comunicaciones de persona a persona. Actualmente están

disponibles los equipos Grupo 3 como aparatos de FAX y conexiones vía módem para computadoras para enviar y recibir FAX por computadora. También los documentos generados por computadora pueden convertirse al formato especificado por el estándar, utilizando una tarjeta adaptadora para FAX, y transmitirse ya sea hacia un equipo similar o hacia un aparato de FAX común. También se cuenta con el estándar Grupo 4 que se aplica a todos los equipos orientados a la transmisión por facsímil digital de imágenes binivel usando las facilidades de la transmisión digital por ISDN (Redes Digitales de Servicios Integrados). Este estándar está dirigido a aplicaciones tales como el correo electrónico, almacenamiento y recuperación de documentos electrónicos o manejo de documentos que integren texto y gráficos. El equipo Grupo 4 aún no tiene una amplia distribución debido a que las ISDN aún no están generalizadas en el mercado y a que la funcionalidad y velocidad del equipo Grupo 3 sigue mejorando. Además, el equipo Grupo 3 ya está adaptado para trabajar sobre líneas digitales ISDN, transmitiendo a 64 Kbps (Kbits por segundo).

En adelante se discutirán las tareas del estándar Grupo 3, las cuales consisten en exploración, transmisión e impresión de FAX, mismas que se ilustran en la figura 5.41.

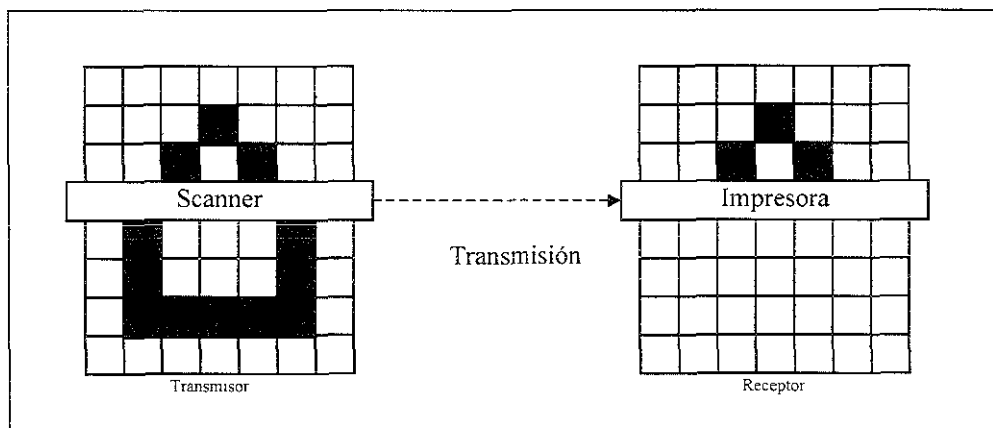


Figura 5.41. Facsímil binivel del Grupo 3 (HOFFMAN, Roy. *Data compression in digital systems*).

Cada página se procesa como una imagen representada por un arreglo bidimensional de píxeles. Se explora una línea a la vez comenzando en la parte superior de la página. Dentro del scanner, la línea explorada se dirige a un chip CCD (dispositivo de acoplamiento de carga) que contiene un arreglo de 1728 fotosensores. Cada fotosensor lee la luminosidad correspondiente a un píxel y produce una señal eléctrica. Para imágenes binivel, la señal puede interpretarse como binaria de 1 o 0 para el negro o el blanco. Después de que se hayan leído, almacenado, codificado y transmitido todas las señales correspondientes a una línea explorada, la exploración pasa a la siguiente línea de la página. La resolución de este proceso es aproximadamente de 200 píxeles/pulgada horizontalmente y de 100 o 200 píxeles/pulgada verticalmente, dependiendo si se selecciona una resolución estándar o alta.

En el lado receptor, las señales son decodificadas y convertidas a un formato conveniente para impresión térmica o de láser. Las señales de sincronización son transmitidas junto con los datos de la imagen para mantener al scanner y a la impresora en el mismo paso.

Por la figura 5.41 es posible darse cuenta por qué transmitir una página de FAX sin comprimir requiere tantos bits. No solamente cada objeto en la página produce bits a transmitir, sino también los espacios en blanco entre los objetos. Dentro de los objetos, hay muchas características que se representan usando bits excesivos considerando que las líneas exploradas adyacentes horizontalmente pueden ser casi idénticas. Los pioneros en el campo del facsímile digital interpretaron estas circunstancias como oportunidades para la compresión de datos. Se desarrollaron algoritmos especializados que no sólo comprimían las largas cadenas de 1's que representan espacios en blanco y las cadenas más cortas de 0's representando componentes negros, sino que los algoritmos también aprovecharon la correlación horizontal y vertical entre las líneas exploradas para lograr mejor compresión.

Hasta ahora, se han descrito los métodos de facsímile convenientes para transmitir imágenes binivel, blanco y negro. También es posible la transmisión por facsímile de imágenes más complejas. El JBIG es un estándar conjunto del CCITT (ITU-T) y el ISO/IEC para imágenes binivel y multinivel. Para imágenes binivel, El JBIG ofrece mejor compresión si se le compara con los algoritmos especificados en las recomendaciones CCITT T.4 y T-6 para FAX Grupo 3. El JBIG también es efectivo cuando se envían imágenes blanco y negro, de escala de gris o a color hacia las impresoras. Otra ventaja del JBIG es que permite utilizar codificación predictiva. Esto da oportunidad, por ejemplo, de enviar una versión de baja resolución de una imagen original rápidamente a un dispositivo de despliegue para una búsqueda rápida. Después si se desea, pueden enviarse el resto de los detalles.

El JPEG es el estándar especificado comúnmente para compresión de facsímile a color. Si bien el JPEG usualmente se ubica como un algoritmo con pérdida de información, también tiene una modalidad sin pérdida para imágenes de facsímile. Sin embargo los equipos de FAX a color no han despertado mucho interés debido a que su costo es mayor que el de FAX blanco y negro y a que las necesidades de los negocios no han llegado a requerirlo. A largo plazo, las proyecciones del facsímile a color pueden no ser alentadoras debido al creciente número de impresoras y fotocopiadoras a color, los cuales son más económicos que los dispositivos de FAX.

En la tabla 5.12 se resumen las características de los algoritmos aplicables a la compresión de facsímile. Todos los algoritmos proporcionan compresión sin pérdida para imágenes de facsímile. El algoritmo JPEG también proporciona compresión con pérdida para imágenes de facsímile a color.

Algoritmo	Recomendación ITU-T	Técnicas de Codificación sin Pérdida empleadas
MH (codificación Huffman modificada)	T.4 (Grupo 3)	Secuencias de píxeles negros o blancos en una línea explorada se codifican con run-length seguido con codificación Huffman estática.
MR (codificación READ modificada)	T.4 (Grupo 3)	Las diferencias entre dos líneas exploradas adyacentes se codifican usando la correlación vertical de píxeles negros o blancos. Periódicamente, se inserta una línea codificada con MH para limitar la propagación de errores.
MMR (codificación READ modificado modificado)	T.6 (Grupo 4)	El MMR adapta el MR por transmisión libre de errores. Las diferencias entre dos líneas exploradas adyacentes se codifican usando la correlación vertical de píxeles blanco y negro.
JBIG	T.82	Las imágenes binivel se codifican con un modelo bidimensional adaptativo que utiliza dos o tres líneas de la imagen seguido por codificación aritmética adaptativa. Las imágenes multinivel, a escala de gris o color se codifican de forma similar con un modelo tridimensional.
JPEG	T.81	Las imágenes de tono continuo, a escala de gris y a color se codifican empleando transformadas, codificación Huffman y aritmética.

Tabla 5.12. Algoritmos de compresión para facsímile (HOFFMAN, Roy. Data compression in digital systems).

5.9. APLICACIONES PARA APARATOS ELECTRODOMÉSTICOS.

5.9.1. Fotografía digital.

A pesar de que la fotografía digital aún no alcanza ni supera la nitidez de la película fotográfica, ésta permite que las imágenes puedan procesarse con métodos que la película nunca podría permitir, además de hacerlo fácil y rápidamente. Hay productos y herramientas domésticas para capturar, procesar, imprimir y desplegar de forma digital fotografías, ya sea para tener versiones miniatura o para publicación, fotoperiodismo y entretenimiento. La figura 5.42 muestra la forma en que las fotografías pueden ser capturadas, procesadas y vistas de forma digital.

Para las personas que aún utilizan cámaras convencionales o tienen un archivo de imágenes en película, pueden utilizarse scanners para digitalizar los negativos, diapositivas y fotografías para almacenarlas y procesarlas. Las cámaras digitales digitalizan directamente las imágenes del mundo real y proporcionan los medios más rápidos para introducirlas a la computadora. La computadora es un elemento vital para la fotografía digital. El software

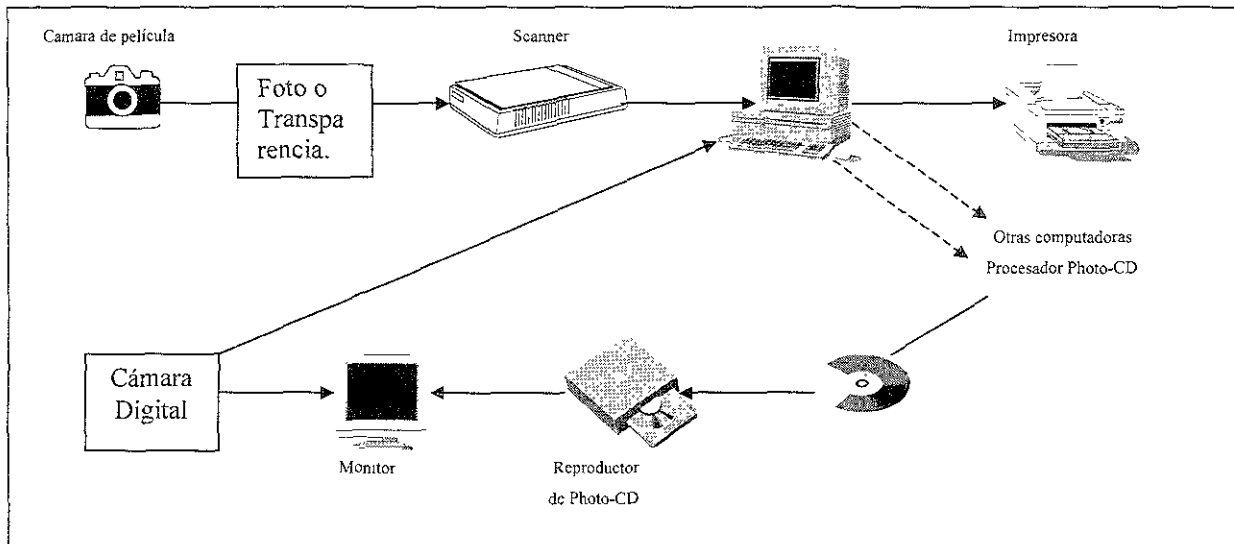


Figura 5.42. Herramientas de la fotografía digital (HOFFMAN, Roy. *Data compression in digital systems*).

de procesamiento de imagen permite manipular y editar las fotografías digitales, incluyéndolas en documentos o para imprimirlas. Las impresoras conectadas a computadoras personales pueden producir impresiones a color que varían en apariencia, desde buena hasta excelente, dependiendo de la calidad de la impresora. Pero existen otras opciones, como la de que los estudios fotográficos estén conectados a una red en línea, permitiéndoles recibir imágenes digitales de los clientes y realizar impresiones de alta calidad o con retoques convenientes para el consumidor. Otra opción es sólo desplegar las imágenes, no imprimirlas. Los fotógrafos pueden desplegar las imágenes en el monitor de su computadora o transmitir las por una red hacia otros usuarios de computadoras personales. Algunas cámaras digitales cuentan con despliegue LCD (Despliegue de Cristal Líquido) de fábrica, y además de enviar las imágenes a la computadora, donde pueden ser desplegadas y manipuladas, proporcionan conexiones para desplegar las imágenes en receptores de televisión o grabarlas en videograbadoras convencionales. Otra opción es enviar las imágenes digitales a un laboratorio fotográfico y tenerlas grabadas en un Foto CD, una técnica que tiene ya algún tiempo disponible.

A pesar de las ventajas de las imágenes digitales, éstas presentaron en sus inicios retos que las compañías fotográficas y electrónicas han ido superando gracias al avance tecnológico. El problema con las imágenes digitales es que requieren una gran cantidad de bits para representarse. Hoy en día, con los avances en los digitalizadores de imagen, la tecnología de almacenamiento, aparatos electrónicos digitales y en la compresión de datos de imagen, la fotografía digital se está haciendo accesible para los fotógrafos profesionales, pequeños empresarios y consumidores en general.

5.9.2. Cámaras digitales.

Las cámaras fotográficas electrónicas y de video se presentaron en el mercado en 1981. Las cámaras operaban como las cámaras de video de mano existentes entonces, pero diseñadas para capturar un cuadro a la vez. Eran de costo elevado, y debido a que tomaron prestada la tecnología analógica de las cámaras de video de mano, sus imágenes no tenían mejor calidad que las de televisión y no podían procesarse de forma digital. Hoy en día, las cámaras electrónicas totalmente digitales siguen siendo de alto costo, pero su calidad de imagen ha mejorado enormemente. Todos los modelos, desde las cámaras digitales simples hasta las más modernas, ofrecen mucho más funcionalidad. Las últimas versiones de cámaras digitales pueden dividirse en tres categorías: Las cámaras digitales profesionales que buscan reproducir la apariencia, textura y operación de las cámaras reflex de película de 35 milímetros. Estas cámaras están dirigidas, y corresponden en precio, a los fotógrafos comerciales profesionales. La segunda categoría la constituyen las cámaras digitales semiprofesionales más baratas que ofrecen muchas funciones de las cámaras profesionales pero con menor resolución de imagen. Estas cámaras resultan atractivas para giros como el de venta de autos, agentes de bienes raíces, agentes de seguros y otros que necesitan pequeñas fotos para propósitos de publicidad y almacenamiento de registros. Finalmente, están las cámaras digitales dirigidas a los fotógrafos ocasionales. Si bien la resolución de la imagen es limitada, incluyen muchas características como una pantalla LCD para composición y revisión de las imágenes, salidas NTSC para conexión a receptores de televisión, videograbadoras, impresoras a color y computadoras domésticas. Aún con esto, todavía falta mucho para que las cámaras digitales puedan igualar la calidad de imagen de las cámaras de película convencionales. La tabla 5.13 proporciona una comparación de los límites de resolución entre la película de 35 mm. y las cámaras digitales.

Productos de Imagen Fija Típicos	Resolución de Imagen (Megapíxeles)	Representación de Color de la Imagen	Tamaño de la Imagen (Sin comprimir)	Almacenamiento para 100 Imágenes (Comprimida 15:1)
Película 35 mm. Máxima resolución	18 M = 5320 x 3550	36 bits	85 MB	567 MB
Película 35 mm. Alta resolución	6 M = 3072 x 2048	36 bits	28 MB	189 MB
Cámara Digital Profesional	3.34 M = 1856 x 1800	36 bits	15 MB	100 MB
Cámara Digital Semiprofesional	2.11 M = 1600 x 1320	24 bits	6.3 MB	42.2 MB
Cámara Digital convencional	1.5 M = 1536 x 1024	24 bits	4.7 MB	31.4 MB

Tabla 5.13. Especificaciones de imagen para equipos fotográficos de película y digitales.

Si bien las cámaras de película económicas no logran toda la resolución posible de la película de 35 mm., sigue siendo superior a la de las cámaras digitales más caras. Sin embargo, lograr una resolución muy alta no es tan importante para los fotógrafos de

instantáneas o incluso para las aplicaciones comerciales donde sólo se ocupan impresiones de tamaño pequeño. Para estas aplicaciones, los fabricantes de cámaras pueden enfocarse hacia las características como la habilidad para ver el cuadro instantáneamente, para digitalizar la imagen, ya sea para hacer ajustes de exposición, cortes o algún tratamiento más sofisticado, y para transmitir las imágenes electrónicamente e incorporarlas en documentos impresos. En la figura 5.43 se ilustra el funcionamiento de una cámara digital.

Operando en un microprocesador, se capturan las imágenes con un sensor de imagen CCD, se procesan y se convierten a formato digital. La imagen digital se almacena temporalmente en el buffer de cuadro mientras se procesa. La etapa de procesamiento de imagen digital efectúa varias funciones: primero comprime una imagen recién capturada y la envía al almacenamiento de imagen, después, con el uso del control, recupera las imágenes comprimidas del almacenamiento de imágenes y las descomprime. Las imágenes reconstruidas pueden enviarse a una computadora vía la función de salida digital y (dependiendo del diseño de la cámara) hacia la pantalla LCD de la cámara, o a un receptor de televisión o videogradora por medio de la función de salida analógica.

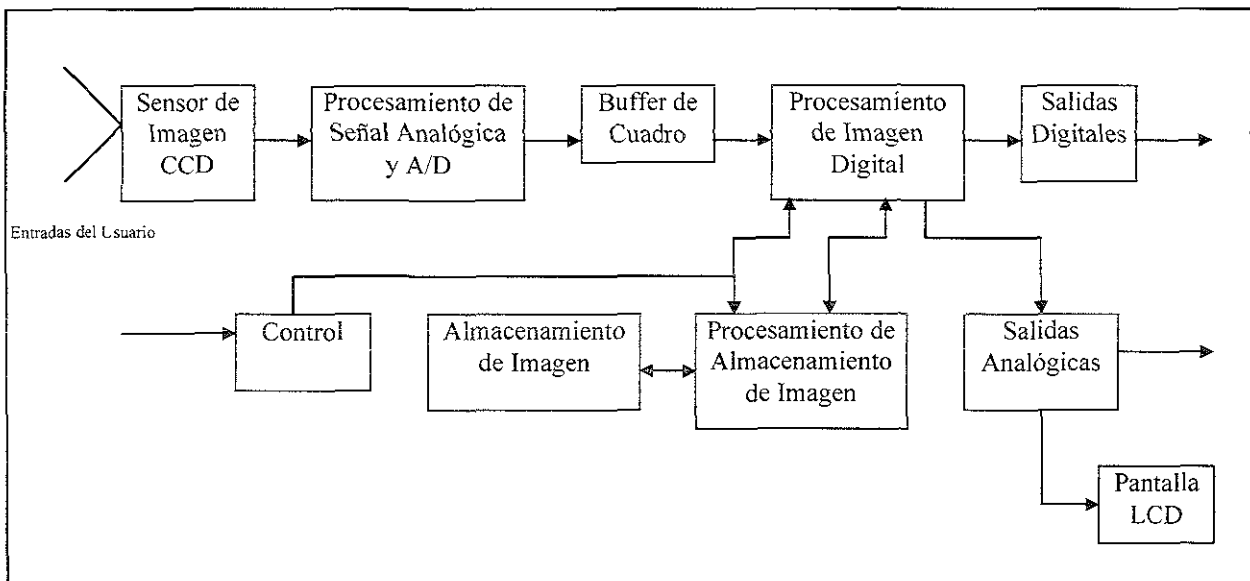


Figura 5.43. Diagrama del funcionamiento de una cámara digital (HOFFMAN, Roy. *Data compression in digital systems*).

La compresión de datos proporciona la herramienta necesaria para hacer que las cámaras digitales sean posibles debido a que, por su portabilidad, la cámara debe almacenar hasta unas 100 imágenes. La reducción de la cantidad de datos a almacenar es crucial, ya que la capacidad de almacenamiento determina grandemente el volumen y el ahorro de energía que tenga la cámara, además de contribuir a establecer los límites de la resolución disponible. El poder comprimir grandes archivos de imágenes en pequeños espacios de almacenamiento permite utilizar medios de almacenamiento como discos magnéticos, tarjetas de disco duro removibles de gran capacidad (formato PCMCIA, tamaño de tarjeta

de crédito) o tarjetas de memoria temporal. El estándar JPEG es el método de compresión utilizado por las cámaras digitales. Las proporciones de compresión alcanzadas son de 15:1 o más.

Las cámaras digitales son productos aún en desarrollo. Su éxito está ligado a los arreglos de CCD de bajo costo que proporcionan altas resoluciones, a la mayor capacidad de almacenamiento, al procesamiento de grandes cantidades de datos a mayor velocidad con menor consumo de energía, y a la mejor compresión de datos de imagen.

5.9.3. Photo CD.

Si bien las cámaras digitales pueden estar fuera del alcance de los fotógrafos ocasionales, con el sistema Photo CD cualquier persona puede experimentar las ventajas de las imágenes electrónicas utilizando las cámaras de película 35 mm. convencionales. El sistema Photo CD fue desarrollado conjuntamente por Kodak y Philips. Permite que los fotógrafos simplemente lleven un rollo de película 35 mm. revelado o no a un laboratorio fotográfico donde, usando una estación de trabajo para imágenes, es posible transferir sus fotos a formato digital y grabarlas en un CD-ROM nuevo. Cada disco puede almacenar alrededor de 100 imágenes y puede actualizarse para agregar nuevas fotos a un disco sin llenar. En 1992, Kodak comenzó a comercializar los productos Photo CD domésticos y profesionales que permiten desplegar en receptores de televisión o en monitores de computadora las imágenes digitales almacenadas o para imprimirlas en los laboratorios fotográficos. Los clientes pueden cargar los discos a un reproductor de CD modificado especialmente y ver imágenes de alta calidad en receptores de televisión convencionales. El reproductor de Photo CD puede pensarse como el remplazo digital del proyector de diapositivas, permitiendo a los observadores hacer secuencias de imágenes en cualquier orden, hacer fundidos, disolvencias, barridos, y girar, agrandar o cortar las imágenes durante su visualización. Sin embargo, el reproductor Photo CD no ha captado gran atención del mercado.

Aquí también la compresión de datos juega un papel importante. Para capturar imágenes de película de 35 mm., el proceso del Photo CD comienza explorando cada imagen a una resolución de 3072x2048 pixeles con color a 24 bits. Esto produce las imágenes digitales de alta resolución necesarias para la impresión. También produce una gran cantidad de datos, cerca de los 19 MB por imagen. Si estas imágenes estuvieran sin comprimir, sólo podrían almacenarse unas 30 imágenes en un CD estándar, mucho menos de los que un proyector de diapositivas puede contener. Los CD's para una sola escritura son de alto precio, por lo que poder almacenar más imágenes en un solo disco resulta atractivo. Además, leer una imagen sin comprimir desde un CD es un proceso lento debido a la velocidad de los dispositivos de lectura. Otra cuestión es que los receptores de televisión y

los monitores de computadora donde se desplegarían las imágenes tienen mucho menos resolución que la usada por el Photo CD para explorar la imagen., así que muchos de esos datos no son necesarios para propósitos de despliegue.

La solución seleccionada para el sistema Photo CD es una tecnología multiresolución que carga los discos con un archivo Image Pac que contiene las imágenes comprimidas almacenadas a cinco diferentes resoluciones. Este rango de tamaños de imagen tiene el propósito de hacer al sistema Photo CD más útil para los usuarios domésticos y profesionales. Como se muestra en la tabla 5.14, cada resolución de imagen está orientada hacia una aplicación distinta y también se utilizan técnicas de compresión diferentes para soportar esas aplicaciones.

Componente de Imagen	Resolución de Imagen	Aplicación
Base/16	192 x 128	Imágenes de Índice Miniatura
Base/4	384 x 256	¼ de Televisión de Super Definición
Base	768 x 512	Televisión de Super Definición
4 Base	1536 x 1024	Televisión de Alta Definición, pequeñas impresiones
16 Base	3072 x 2048	Impresiones en papel (de menos de 16x20 pulgadas)

Tabla 5.14. Resoluciones de imagen del Photo CD (HOFFMAN, Roy. Data compression in digital systems).

La compresión de imagen Photo CD opera de la forma siguiente: Para codificar una imagen, el proceso Photo CD siempre comienza con una imagen positiva (imágenes negativas convertidas a positivas). A través de una serie de transformaciones, la imagen digital se representa por medio de un formato de componentes de color especial Photo YCC (luminancia, cromancia, cromancia) obteniéndose una imagen base. Este formato se diferencia de la representación $YCbCr$ en que utiliza los valores de color primarios RGB como antecedente directo para preservar el rango dinámico de las imágenes de película, soportando tanto la impresión como el desplegado en video.

La compresión utilizada para los tres tipos de resolución de imagen más baja del Photo CD, sólo consiste en submuestrear los componentes de cromancia en un factor de 2 en cada dirección. Esta técnica, que se usa comúnmente en compresión de imagen y video, proporciona una compresión de 2:1 casi sin ninguna pérdida apreciable en la apariencia de la imagen. Se seleccionó este método debido a que las tres resoluciones más bajas representan solamente una pequeña fracción del total de datos y a que permite que estas imágenes estén disponibles rápidamente para su visualización.

Para acomodar unas 100 imágenes en un solo disco, las cinco resoluciones distintas deben poder almacenarse en unos 6 MB de espacio. El submuestreo de los componentes de cromancia por sí solo no puede comprimir las dos resoluciones más altas para lograr este objetivo. Esto significa que estas imágenes deben comprimirse aún más, y para realizar esto, se utiliza una técnica de codificación piramidal o jerárquica. El proceso de

codificación piramidal comienza con el muestreo descendente de la imagen original, que consiste en filtrarla con un filtro de paso bajo y submuestrearla en factor de 2 en cada dirección. Esto crea la primera imagen de resolución reducida. Después esa imagen es muestreada descendentemente de nuevo para producir la siguiente imagen. Este proceso continua hasta que se forma de imagen de menor resolución de la pirámide. El proceso de compresión comienza muestreando ascendentemente (escalando) la imagen de más baja resolución hacia la siguiente imagen de resolución más alta y restándola de la siguiente imagen de resolución más alta para obtener una imagen diferencial. La imagen diferencial se comprime usando un método de compresión apropiado, como las técnicas JPEG, de cuantificación y de codificación entrópica. Esta operación continua hasta que se haya codificado la imagen de mayor resolución en la pirámide.

La codificación piramidal se utiliza para almacenar solamente las dos imágenes de resolución más alta del Photo CD. Estas imágenes contienen la mayoría de los datos, pero debido a que los receptores de televisión aún no cuentan con esas resoluciones, no satisfacen los mismos requerimientos de decodificación en tiempo real que las imágenes de menor resolución.

El método utilizado para almacenar las imágenes diferenciales en el Photo CD no es el JPEG, debido a que cuando se desarrolló esta tecnología el JPEG aún no estaba terminado, por lo que Kodak desarrolló un algoritmo propietario basado en cuantificación y codificación Huffman de longitud variable el cual genera un archivo de tipo PCD.

5.10. APLICACIONES EDITORIALES.

5.10.1. Publicaciones tradicionales.

Sólo en algunos años la industria editorial, una de las más antiguas del mundo, ha revolucionado sus métodos de producción y distribución gracias al advenimiento de la tecnología basada en computadoras. Las aplicaciones de la compresión de datos en esta industria principalmente tienen que ver con el almacenamiento eficiente de todos los tipos de datos que se manejan a través de los distintos medios editoriales como los que se muestran en la tabla 5.15.

Segmento de la Industria	Elementos de Producción	Medios de Distribución	Medios de Entrega
Libros, Revistas y Periódicos tradicionales	Texto, Gráficos e Imágenes	Papel y Microfilm	Tiendas, Correo, Entrega a domicilio
CD-ROM electrónico	Texto, Gráficos, Imagen, Animación, Audio y Video	CD-ROM	PC's Multimedia y aparatos de TV
En línea	Texto, Gráficos, Imagen, Animación, Audio y Video	Red Electrónica (Internet)	PC Multimedia

Tabla 5.15. Medios editoriales (HOFFMAN, Roy. *Data compression in digital systems*).

El proceso de publicación tradicional comienza cuando los autores y los anunciantes tienen información que desean imprimir. El primer paso es editarla y diseñarla para tener el material base. Durante la producción, los elementos visuales se reúnen y se preparan para la impresión. Después de la impresión, se distribuyen las copias a los lectores. La figura 5.44 muestra un esquema de este proceso.

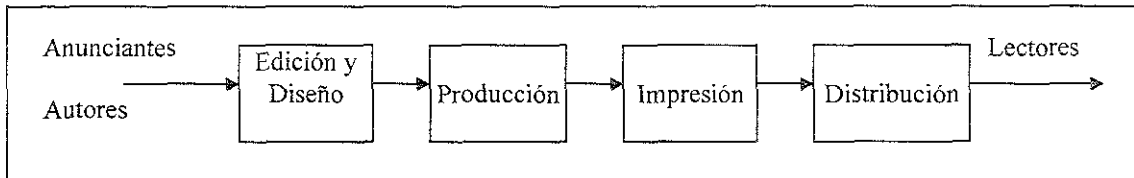


Figura 5.44. Proceso de publicaciones tradicionales (HOFFMAN, Roy. Data compression in digital systems).

Hoy en día, cada etapa de este proceso se realiza utilizando las facilidades de la edición digital computarizada. Las herramientas como procesamiento de palabras, gráficos digitales, imágenes digitales, y retoque fotográfico son muy comunes en esta industria.

Existen dos aplicaciones importantes de la compresión de datos en este proceso: la distribución digital de estos documentos entre los sitios de trabajo y el mantenimiento de un archivo de imágenes digitales. Las publicaciones digitales generan archivos de datos muy grandes. Se necesitan imágenes de alta resolución para impresión y, especialmente para los periódicos, el formato de página es grande, lo que significa que se requieren muchos bits para cada imagen digitalizada. Lo más importante, estos archivos deben transmitirse a través de redes debido a que los pasos individuales del proceso de publicación se realizan en lugares separados físicamente, a menudo en negocios independientes. Por lo tanto, las publicaciones requieren imágenes y gráficos de grandes dimensiones que deben almacenarse en formato digital.

La distribución digital se utiliza para transmitir copias de los anuncios desde las agencias de publicidad hacia las editoriales de periódicos y revistas. También se utiliza para enviar imágenes listas para impresión desde las editoriales a nivel nacional hacia prensas remotas. Regularmente, los textos, gráficos e imágenes listas para impresión se comprimen con algoritmos LZW u otros algoritmos sin pérdida. Las fotografías se comprimen con algoritmos JPEG con o sin pérdida o con algoritmos Photo CD.

5.10.2. Publicaciones electrónicas.

Publicaciones electrónicas es la denominación del giro empresarial que se dedica al desarrollo y distribución de material editorial por medios electrónicos como el CD-ROM o a través de las redes de comunicación como Internet conocido como distribución en línea.

Casi cualquier información procesada con herramientas editoriales asistidas por computadora es factible de ser comercializada gracias a este medio.

Las publicaciones electrónicas involucran etapas de producción similares a las de las publicaciones tradicionales, pero con estructuras y diferencias importantes. La figura 5.45 muestra un esquema de estos pasos.

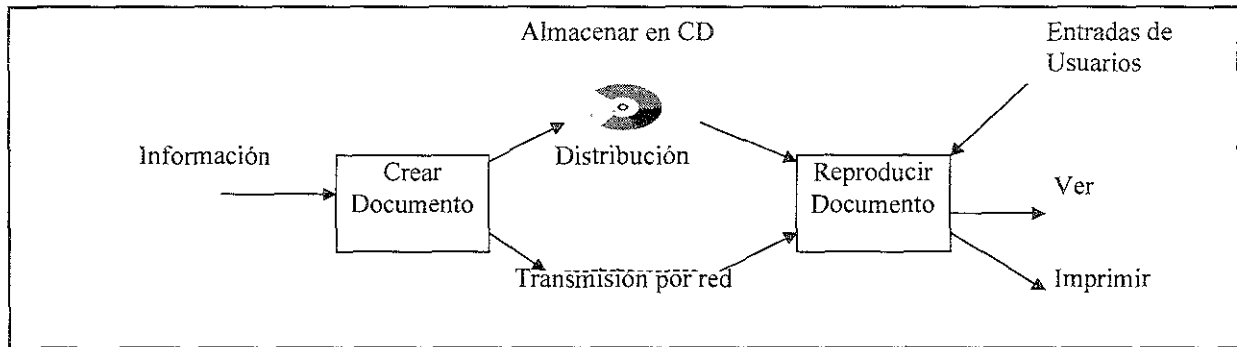


Figura 5.45. Proceso de las publicaciones electrónicas (HOFFMAN, Roy. *Data compression in digital systems*).

Como con las publicaciones tradicionales, las publicaciones electrónicas comienzan procesando la información (editar, diseñar o retocar) para crear el documento. En contraste con las publicaciones tradicionales, la editorial distribuye el documento y el usuario, no la editorial, lo reproduce si lo desea.

Existen un sin número de algoritmos de compresión de texto, imagen y video que pueden utilizarse para auxiliar a la publicación electrónica en la solución de los problemas de almacenamiento y ancho de banda limitados. Considérese las publicaciones en CD-ROM: Estas publicaciones electrónicas gozan de gran flexibilidad en la selección de las tecnologías de compresión de datos a utilizar y fue de las primeras áreas del comercio donde se aplicaron varias técnicas de compresión simultáneamente. La publicación en CD-ROM es un proceso donde las editoriales no sólo crean el contenido sino que también proporcionan el software para reproducir el CD-ROM. Esto implica que tienen libertad de escoger la compresión de datos que sea mejor para su producto en términos de proporción de compresión alta y de tiempo de descompresión muy bajo, ya que la velocidad y la compatibilidad de compresión no son importantes en este caso. Para ayudar a las editoriales a seleccionar una técnica de compresión, un fabricante proporciona una herramienta de software automatizada denominada capa CD, que utiliza un arreglo de compresores para analizar un archivo de documento electrónico y escoger la o las técnicas de compresión que reduzcan los datos en el menor espacio posible. Junto con el archivo comprimido, la herramienta capa CD también proporciona los programas de descompresión que necesitarán los usuarios para leer el archivo.

Un ejemplo de las técnicas de compresión avanzadas que utilizan las editoriales de CD-ROM's es la compresión fractal. Los algoritmos fractales son muy efectivos para reducir imágenes de escenas naturales de alta calidad en la menor cantidad de bits posible, pero a expensas de una compresión más lenta y compleja. Sin embargo, el lado positivo es que la descompresión es más rápida y simple, lo que hace a esta técnica muy atractiva para las editoriales de CD-ROM's.

Considérese ahora las publicaciones en línea: En este campo se utilizan muchos algoritmos de compresión de imágenes, con grandes diferencias de eficiencia y calidad. Un ejemplo es el formato GIF que es líder en los servicios de información en línea y de transmisión por Internet, pero es un formato sin pérdida y que tiene tiempos de descarga de la red muy considerables. Para controlar el tamaño de los archivos, el GIF limita la estructura de sus imágenes a 256 colores. Para reducir el tiempo de descarga de la red, Compuserve desarrolló una modalidad jerárquica de GIF que primero envía una imagen de baja resolución que puede verse mientras se espera a que se descargue la imagen completa. En contraste, los algoritmos de compresión con pérdida son mucho más efectivos para crear archivos de imagen compactos a cualquier resolución, y los desarrolladores de medios han comenzado a poner en Internet imágenes comprimidas con JPEG a 16 bits.

5.11. APLICACIONES PARA EL CUIDADO DE LA SALUD.

5.11.1. Sistemas de imágenes médicas.

Las imágenes son un elemento de creciente importancia para la medicina. No sólo el procesamiento de imágenes digitales ha proporcionado nuevas herramientas para crear formas útiles de información para diagnóstico, sino que las imágenes digitales resultan ser más eficientes para su manejo. En los últimos años se han desarrollado varios sistemas de archivo, procesamiento y transmisión de imágenes digitales para aplicaciones médicas. Si bien aplicadas inicialmente a la radiología y a la patología, ahora se utiliza la misma tecnología para otras áreas de la medicina. Los dispositivos de imágenes médicas radiológicas utilizan un tipo de energía para estimular los tejidos del paciente y al medir la interacción, construyen una imagen que brinda información para diagnosticar heridas y enfermedades.

La industria del cuidado de la salud gasta enormes cantidades de dinero para crear, clasificar y almacenar imágenes generadas durante los exámenes a los pacientes. Existen normas legales para la retención de la información médica que obliga a los centros de salud a conservar sus documentos cierto período de tiempo, usualmente de 3 a 7 años y en algunos casos hasta 30 años, lo que hace un archivo médico algo indispensable. Los

archivos de registros médicos almacenan información muy importante, ya sea en papel o en película, pero dicho almacenamiento resulta costoso. En virtud de esto, las instituciones médicas buscan soluciones de bajo costo que les permitan administrar un creciente número de documentos para hacer más eficientes a los sistemas de salud.

Las imágenes digitales contribuyen a resolver este problema, conjuntamente con las computadoras y las redes de información. Las imágenes digitalizadas pueden manejarse por medio de computadoras interconectadas a las redes de las grandes instituciones médicas. La clave para llevar a cabo esta tarea es un sistema de administración de imágenes y comunicaciones (IMACS) que recibe imágenes en forma digital, con texto asociado y las distribuye a través de una red.

El uso de las computadoras y la tecnología de la información en la administración de las imágenes médicas no es una idea nueva. A mediados de los años 60, las universidades y algunas instituciones en los Estados Unidos comenzaron el desarrollo de lo que denominaron un sistema de archivo de imágenes y comunicaciones (PACS). En los años 80, muchas instituciones médicas comenzaron a utilizar prototipos PACS a nivel mundial. Al encontrar que la tecnología digital era adecuada para el manejo de imágenes médicas e integrando a las PACS a los procedimientos de las instituciones médicas se planteaba una nueva forma de trabajo. Después de varios años de desarrollo de la tecnología de las PACS ahora es parte de las soluciones integradas IMACS, las cuales se están insertando como un elemento más en las operaciones clínicas.

Las cuestiones involucradas en el diseño de estos sistemas incluyen las siguientes:

- Las imágenes médicas digitalizadas consumen una gran cantidad de almacenamiento y de tiempo de transmisión. Una imagen radiológica a escala de gris puede consumir hasta 40 MB de almacenamiento. Dependiendo de la velocidad de la red, el tiempo de transmisión puede ser de segundos, minutos o hasta horas.
- Las imágenes médicas capturadas recientemente deben estar disponibles para desplegarse en cuestión de segundos, y a menudo los doctores necesitan ver varias imágenes simultáneamente.
- La resolución espacial y la profundidad de pixel (número de niveles de gris) de las imágenes médicas a menudo es mayor a la capacidad de despliegue de computadoras convencionales.
- Los usuarios de la comunidad médica tienen diferentes exigencias de la resolución de la imagen. Los radiólogos necesitan que las imágenes se desplieguen a su máxima resolución para hacer sus diagnósticos, mientras que otros médicos pueden trabajar con versiones de baja resolución y con despliegues más económicos.

- El procesamiento de imágenes, como la compresión de datos, introduce artificios (detalles no presentes en la imagen original) que son inaceptables en el caso de que puedan afectar el diagnóstico médico.
- Las redes que conectan a los componentes IMACS deben manejar texto y muchos tipos de imágenes (y en ocasiones voz y video para consultas entre médicos).
- Deben integrarse una gran variedad de productos para construir las IMACS. La base instalada de modalidades de imágenes médicas requiere atención especial. Muchas modalidades de imagen están diseñadas para proporcionar imágenes sólo hacia impresoras en película, no hacia despliegues de computadoras, y en ausencia de estándares, sus salidas digitales a menudo se codifican en formatos propietarios.
- Un reemplazo conveniente de las modalidades de imágenes de rayos X basadas en película, sería el completar la transición de las instituciones médicas para dejar de trabajar con material de película.
- Cada día en las instituciones médicas, se recolectan decenas de gigabytes de imágenes. Estos datos deben estar disponibles para su visualización inmediata y archivados en dispositivos de almacenamiento a largo plazo.

Con excepción de los rayos X convencionales, no es muy difícil recolectar las imágenes médicas debido a que las modalidades de imagen tienen memoria local adecuada y hardware dedicado para adquirir las imágenes eficientemente y la mayoría tienen salidas digitales. La cuestión principal para las IMACS es cómo almacenar las imágenes de forma económica y transmitir y desplegar las imágenes almacenadas en el mismo instante en que se solicitan. La compresión de datos es una pieza clave de la solución.

5.11.2. Operación de las IMACS.

Para entender el funcionamiento de las IMACS considere el esquema de la figura 5.46. Los componentes de un IMACS están conectados por redes. Las LAN's proporcionan enlaces entre los elementos de un sitio. Las WAN's proporcionan los enlaces hacia sitios remotos de una institución médica o hacia diferentes instituciones para consultas teleradiológicas. Un administrador de bases de datos del IMACS controla el funcionamiento general de las IMACS, incluyendo el flujo de trabajo, los datos, y los dispositivos conectados a la red.

Los componentes IMACS efectúan las siguientes funciones: las modalidades de imágenes capturan, procesan, almacenan, y despliegan las imágenes médicas digitales, y la mayoría de las modalidades pueden hacer copias en película. Las modalidades de imagen son sistemas complejos que utilizan ampliamente la tecnología de procesamiento de señal para crear imágenes médicas. La salida digital de una modalidad de imagen se introduce al IMACS por medio de un procesador compuerta (gateway processor), también denominado Unidad de Adquisición de Imagen (IAU). Hay distintos tipos de modalidades de imagen.

Para cada modalidad de imagen, debe programarse una IAU para capturar sus datos de imagen, convertirlos a un formato de archivo estándar para imágenes médicas, y transmitir los datos sobre la red IMACS. La IAU envía los datos de imagen al archivo óptico, donde se almacena permanentemente, y hacia un servidor de imagen para almacenamiento temporal. Cuando las imágenes se han almacenado exitosamente, la IAU crea un registro de estas transacciones para el administrador de base de datos del IMACS.

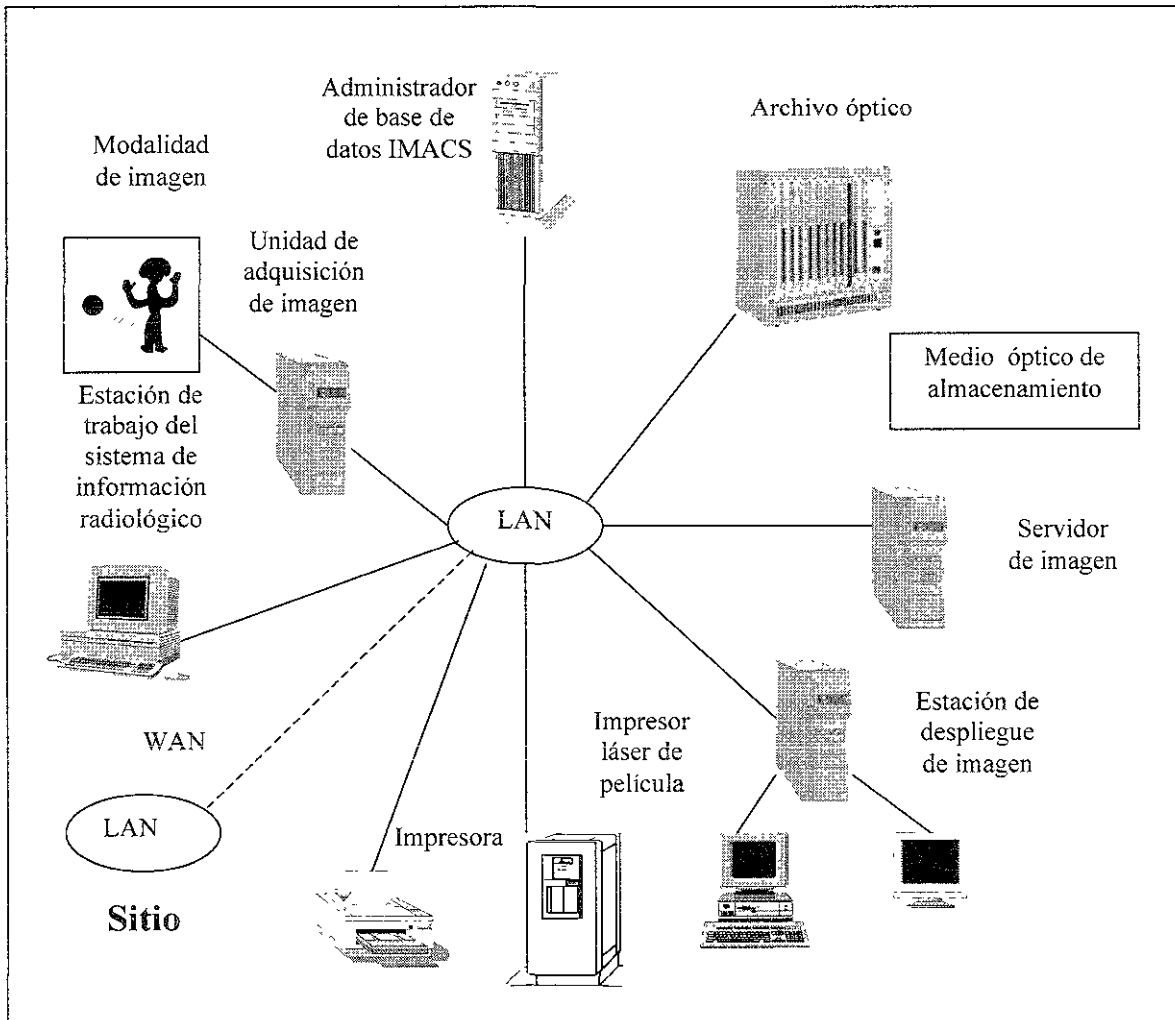


Figura 5.46. Sistema de administración de imágenes y comunicaciones (HOFFMAN, Roy. Data compression in digital systems).

La compresión de datos es un factor crítico para el éxito de los IMACS en su intención de sustituir el uso de las imágenes médicas que utilizan película. Debido a que las imágenes comprimidas utilizan menos bits, son de menos costo para su almacenamiento y pueden transmitirse de forma más rápida. Esto incrementa la funcionalidad de los IMACS ya que tienen costos de almacenamiento y los tiempos de transmisión se reducen. Sin embargo, Muchos prototipos PACS operan con éxito sin utilizar compresión, debido esencialmente a que sólo manejan imágenes pequeñas producidas por modalidades de imagen pseudodigitales como las CT y MRI (que comprimen pero sólo en una parte de las posibilidades). Esta situación cambia cuando los IMACS deben manejar imágenes médicas

de gran tamaño. Para establecer la comparación, la tabla 5.16 muestra los tamaños y tiempos de transmisión típicos de todos los tipos de imágenes médicas digitalizadas sin comprimir.

Modalidad de Imagen	Resolución Espacial y Profundidad de Píxeles de Imagen	Imágenes por Examen	Almacenamiento para Resultados de Exámenes (MB)	Tiempo de Transmisión Mínimo para Resultados de Exámenes		
				LAN 10 Mbps	WAN 1.5 Mbps	ISDN 128 Kbps
CT	512x512x12	30	15.7	12.6 seg.	83.9 seg.	16.4 min.
MRI	256x256x12	50	6.6	5.2 seg.	35 seg.	6.8 min.
US	512x512x6	36	9.4	7.5 seg.	50.3 seg.	9.8 min.
DSA	1024x1024x8	20	21	16.8 seg.	111.8 seg.	21.8 min.
DF	1024x1024x8	15	15.7	12.6 seg.	83.9 seg.	16.4 min.
SPECT	128x128x6	50	1.6	1.3 seg.	8.7seg.	1.7 min.
PET	128x128x6	62	2	1.6 seg.	10.8 seg.	2.1 min.
CR	2048x2560x10	4	41.9	33.6 seg.	223.7 seg.	43.7 min.
Rayos X Digitalizados	4096x5120x12	4	167.8	134.2 seg.	894.8 seg.	174.8 min.

Notas: CT = tomografía computacional, MRI = imagen de resonancia magnética, US = ultrasonografía, DSA=angiografía substractiva digital, DF = fluorografía digital, SPECT = tomografía computarizada de emisión única de fotones, PET = tomografía de emisión de positrones, CR = radiografía computacional.

Tabla 5.16. Tamaños de imágenes radiológicas sin comprimir y sus tiempos de transmisión (HOFFMAN, Roy. Data compression in digital systems).

El almacenamiento requerido para imágenes grandes es una cuestión crítica, debido a que, por ejemplo, las imágenes de rayos X comprimen alrededor del 70 al 75 por ciento de la compresión potencial. El tiempo de transmisión para imágenes grandes es un asunto aún más crítico, ya que son tiempos demasiado largos como para proporcionar un alto desempeño en la visualización interactiva de las imágenes, ya sea a través de una LAN o de una WAN. Para aplicaciones donde las imágenes deben enviarse a través de líneas telefónicas como en la teleradiología en línea, el tiempo de transmisión para imágenes de cualquier tamaño amplio es demasiado grande. La figura 5.47 muestra el flujo de trabajo de las imágenes médicas dentro de los IMACS.

La forma más efectiva para utilizar la compresión de datos en esta aplicación es almacenar las imágenes solamente en forma comprimida, y siempre que sea posible, eliminar las imágenes sin comprimir del tráfico de las redes IMACS. Esto requiere que la compresión se efectúe cerca de las fuentes de las imágenes y la descompresión cerca de los despliegues y las impresoras. El resultado es que el almacenamiento se reduce en un factor igual al de la proporción de compresión, así como su tráfico de red. Efectuando la compresión y la descompresión en las posiciones sugeridas en la figura 5.47 se logra este objetivo. Se asume que las estaciones de despliegue tienen suficiente poder como para descomprimir

los datos. Las impresoras no pueden aceptar imágenes comprimidas, por lo que la descompresión debe efectuarse en el servidor de imágenes.

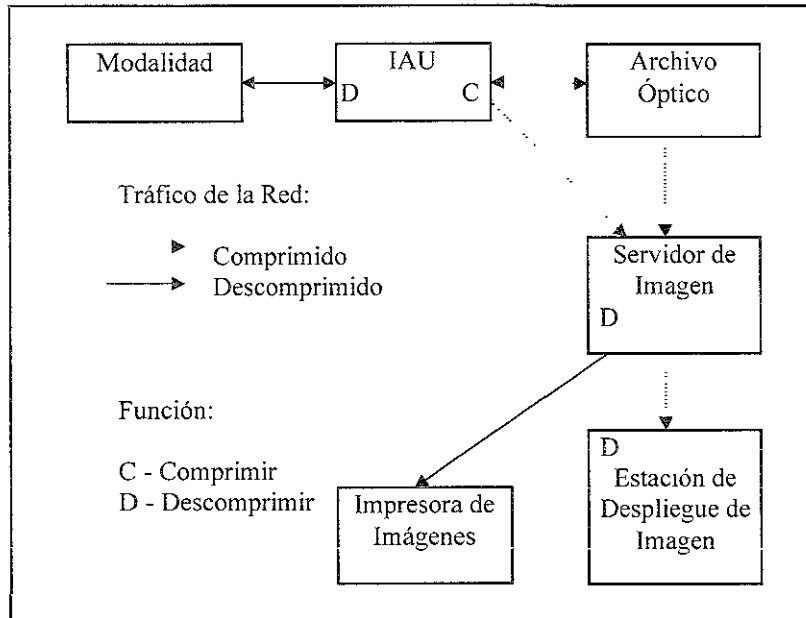


Figura 5.47. Flujo de trabajo de los IMACS (HOFFMAN, Roy. *Data compression in digital systems*).

5.12. RESUMEN DE ESTÁNDARES COMERCIALES Y ÁREAS DE APLICACIÓN DE LA COMPRESIÓN.

Los estándares descritos en este capítulo aplican de manera importante técnicas de reducción de la redundancia de codificación e interpixel como las revisadas en las secciones 1.5.1 y 1.5.2. Este hecho se hace más evidente para los estándares de compresión para FAX como el T.4 y el T.6, lo que implica que las técnicas de compresión que utilizan no son muy sofisticadas y por lo tanto su proporción de compresión es baja, debido a que su objetivo principal es conservar todos los datos que sean transmitidos. Esta situación es diferente para el caso de los estándares que no están dirigidos a la compresión de documentos e imágenes sino exclusivamente de imágenes, como el JBIG y el JPEG. Si bien el JBIG también es un estándar sin pérdida de información, aplica técnicas de selección de valores más sofisticadas y dirigidas especialmente hacia la información de tipo visual. El caso del estándar JPEG es especial, ya que es el estándar de compresión más popular a todos los niveles del ambiente gráfico digital. Sobre su estructura básica, se han creado y se siguen creando nuevos formatos de imagen digitales, hardware de compresión y aceleración de gráficos, herramientas de compresión para aplicaciones de diseño gráfico además de ser la técnica de compresión bajo la cual, con las correspondientes

modificaciones de los proveedores del formato, se procesan la mayoría de las imágenes que se encuentran en Internet.

Con respecto de las aplicaciones de compresión digital de imágenes en la vida corriente puede establecerse que, cualquier actividad que involucre el manejo de información gráfica es susceptible de utilizar la compresión como un medio fundamental para mejorar la eficiencia de los recursos y el aprovechamiento del tiempo invertidos. En la tabla 5.17 se muestra un cuadro comparativo acerca del uso, la factibilidad, o la conveniencia de utilizar cierto grupo de técnicas de compresión de las expuestas en este trabajo, como parte del funcionamiento de las aplicaciones descritas en el capítulo, dependiendo de factores como velocidad a la que deben comprimirse las imágenes, velocidad a la que deben desplegarse las imágenes, velocidad a la que deben transmitirse las imágenes, observador final al cual se dirigen las imágenes resultantes, procesos intermedios a los cuales se someten las imágenes y tipo de dispositivo que efectuará la compresión.

Clasificación De Técnicas de Compresión	Computadoras Multimedia	Sistemas de Imágenes de Documentos	Servidores Multimedia	Facsímile	Fotografía Digital	Cámaras Digitales	Photo CD	Publicaciones Tradicionales	Publicaciones Electrónicas	Sistemas de Imágenes Médicas
Forma de Onda	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí
Segunda Generación	No	No	Sí	No	Sí	Sí	Sí	Sí	Sí	Sí
Fractal	No	No	No	No	No	No	No	No	Sí	No

Tabla 5.17. Cuadro comparativo de las clasificaciones de técnicas de compresión de imagen fija con respecto a las aplicaciones expuestas.

CONCLUSIONES

El conocimiento obtenido a partir de la realización de esta investigación abarca varios aspectos. Al exponer el funcionamiento de las técnicas de alta compresión para imágenes digitales, se muestra que son métodos de procesamiento de información digital que aprovechan la estructura y las características particulares de los datos que las componen así como el objetivo al que se orientan dichas imágenes.

Por una parte, las técnicas de compresión ven a las imágenes como un arreglo matricial de muestras con valores similares en un cierto rango. Este hecho permite aplicar desde algoritmos de predicción de valores siguientes a partir de un valor inicial dado en su forma más simple, hasta métodos de evaluación y discriminación de muestras basándose en la representatividad de su valor para el resto de la región examinada. Dicha región varía en su tamaño y forma, ya sea que consista en el arreglo de muestras completo, en subregiones cuadrículares de tamaño fijo o en subregiones de distinto tamaño y forma, dependiendo del método seleccionado.

Por otra parte, las técnicas de alta compresión también aprovechan el hecho de que las imágenes digitales, en la mayoría de sus aplicaciones, serán observadas por el ser humano. Esta circunstancia les otorga la posibilidad de eliminar los datos que no son indispensables para brindar una representación reducida dirigida al observador final, ya que éste construirá la imagen completa por medio de las funciones neuronales relacionadas con el sentido de la vista. La representación reducida consiste en conservar solo la información más significativa para el cerebro.

Además de los aspectos anteriores, las técnicas de compresión toman en cuenta como parte de su diseño la forma en que se construye la representación final de la imagen comprimida. Estas formas pueden consistir en secuencias de datos que identifiquen niveles de color para las muestras a reconstruir en la imagen descomprimida, hasta ecuaciones polinomiales que identifiquen figuras básicas contenidas en la imagen original. El uso de una u otra forma de representación en una aplicación depende de varios factores como la calidad en la apariencia final de la imagen descomprimida, la velocidad a la que deba desplegarse en un dispositivo de visualización o los costos implicados en su almacenamiento y transmisión. A manera de ilustración sobre este aspecto, en la tabla I se muestra un comparativo sobre las clasificaciones de técnicas de compresión estudiadas en esta investigación. Esta relación plantea la conveniencia o inconveniencia de su uso para las aplicaciones expuestas

en las secciones 5.7, 5.8, 5.9, 5.10 y 5.11, dependiendo del proceso al que someten a la imagen original y de los criterios utilizados en los cuadros comparativos de cada capítulo.

Las técnicas que reducen en alta proporción la cantidad de datos necesarios para representar a las imágenes digitales, son el resultado de combinar disciplinas como los métodos matemáticos, la estadística y las ciencias de la computación. Pero la trascendencia de su uso alcanza ámbitos más allá del científico, como el comercial y el industrial. A pesar de que la capacidad de procesamiento de los equipos de cómputo permite realizar millones de operaciones por segundo, la situación del mercado del procesamiento de información digital ha postergado la utilización a gran escala de muchas de las técnicas expuestas en este trabajo, por circunstancias que están fuera del alcance de la investigación, pero que permiten establecer que el área de la compresión de imágenes es aún campo de estudio floreciente y una herramienta de gran valor actual y potencial.

Clasificación de Técnicas de Compresión	Computadoras Multimedia	Sistemas de Imágenes de Documentos	Servidores Multimedia	Facsimile	Fotografía Digital	Cámaras Digitales	Photo CD	Publicaciones Tradicionales	Publicaciones Electrónicas	Sistemas de Imágenes Médicas
Forma de Onda	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí
Segunda Generación	No	No	Sí	No	Sí	Sí	Sí	Sí	Sí	Sí
Fractal	No	No	No	No	No	No	No	No	Sí	No

Tabla I. Cuadro comparativo de las clasificaciones de técnicas de compresión de imagen fija con respecto a las aplicaciones expuestas.

RECOMENDACIONES

Para que un nuevo conocimiento pueda comprenderse y valorarse en toda su amplitud, debe acompañarse de resultados que certifiquen su veracidad. En varias secciones de esta investigación se incluyeron, en la medida de la disponibilidad, imágenes originales y las correspondientes imágenes resultantes que sirvieran para ejemplificar el efecto de la aplicación de las técnicas de compresión tratadas en cada sección, aunque en su mayoría el criterio de comparación puede remitirse únicamente al ámbito de la fidelidad subjetiva de la imagen resultante como se explica en la sección 1.7.3.2. Una circunstancia similar ocurre para con los documentos sobre este tema disponibles a los estudiantes universitarios en los centros de investigación citados en la introducción de este trabajo.

Dichas investigaciones en su mayoría hacen referencia a resultados experimentales realizados por otras instituciones o por los mismos autores y cuya disponibilidad está fuera del alcance de los documentos publicados y por consiguiente de esta investigación. Los resultados experimentales corresponden a pruebas de laboratorio realizadas en centros de investigación internacionales, orientadas a la evaluación del desempeño de la compresión de imágenes digitales utilizando como criterios las medidas expuestas en la sección 1.7 y algunas más sofisticadas. La falta de disponibilidad de dichas evaluaciones a nivel alumnado obedece a que son materiales con derechos de autor, los cuales pertenecen a centros como el Comité Consultivo de Telefonía y Telegrafía (CCITT), ahora Unión Internacional de Telecomunicaciones (ITU), el Grupo de Expertos Fotográficos Unidos (JPEG), la Organización de Estándares Internacionales (ISO) y el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE).

Mi recomendación para futuras investigaciones consiste, por lo tanto, en orientar dichos esfuerzos hacia el estudio profundo de alguna de las técnicas aquí expuestas, y especialmente hacia la técnica de compresión basada en transformadas (ver sección 2.3). Esta recomendación se debe a que es la técnica que dispone de más información documentada y a que el conocimiento detallado de la misma, acompañado de resultados comparativos concretos, facilitará la comprensión del resto de las técnicas de compresión, por lo que valdrá la pena cualquier inversión realizada en este sentido.

La propuesta que derivo de la realización de este trabajo consiste en incluir al tema de la compresión de imágenes digitales como uno de los proyectos de los centros de investigación en computación ya sea de una institución como la UNAM, el IPN, o en

colaboración con otras instituciones gubernamentales o privadas como el Instituto Tecnológico de Estudios Superiores de Monterrey. Estas tres instituciones cuentan con centros de investigación sobre computación, y dentro de ellos con un área específica para el estudio del procesamiento de imágenes digitales. Sin embargo no han considerado aún a la compresión de imágenes dentro de sus proyectos principales, pasando por alto el beneficio potencial de este campo de la computación.

BIBLIOGRAFÍA

ANSON, Louisa F. "Fractal image compression" en BYTE. Estados Unidos, octubre 1993, pp. 195-200.

BARNSLEY, Michael F., SLOAN, Alan D. "A better way to compress images" en BYTE. Estados Unidos, enero de 1988, pp. 215-223.

BOSS, R. D., FISHER, Yuval, JACOBS, E. W., "Image compression: A study of the iterated transform method" en Signal processing. Vol. 29, n.3, Estados Unidos, diciembre 1992, pp. 251-263.

CHELLAPPA, Rama. Digital image processing. Estados Unidos, IEEE Society Press.

EBRAHIMI, Touradj, KUNT, Murat. "Visual data compression for multimedia applications" en Proceedings of the IEEE. Vol. 86, n. 6, Estados Unidos, junio 1998, pp. 1109-1125.

FISHER, Yuval. Fractal image compression. Estados Unidos, Springer-Verlag.

FURHT, Bolko, SMOLIAR, Stephen. Video and image processing in multimedia systems. 2ª ed., Estados Unidos, ed. Kluwer Academic Publishers, 1996.

GHARAVI, Hamid, TABATABAI, Ali. "Sub-band coding of monochrome and color images" en IEEE transactions on circuits and systems. Vol. 35, n.2, Estados Unidos, febrero 1988, pp. 207-214.

HELD, Gilbert. Data and image compression. Estados Unidos, ed. John Wiley, 1996.

HOFFMAN, Roy. Data compression in digital systems. Estados Unidos, ed. Chapman & Hall, 1997.

IKONOMOPOULOS, Athanassios, KOCHER, Michel, KUNT, Murat. "Second-generation image-coding techniques" en Proceedings of the IEEE. Vol. 73, n.4, Estados Unidos, abril 1985, pp. 549-574.

- JAIN, Anil K. Fundamentals of image processing. Estados Unidos, Ed. Prentice-Hall.
- KAY, David C., LEVINE, John R. Graphics file formats. Estados Unidos, Ed. Windcrest/McGraw-Hill, 1992.
- KESPRET, Irene. PKZIP LHARC & Co. The ultimate data compression book. 2ª ed., Estados Unidos, Ed. Abacus, 1994.
- KOU, Weidong. Digital image compression. Algorithms and standards. Estados Unidos, Kluwer Academic Publishers, 1995.
- MITCHELL, PENNEBAKER. JPEG still image data compression standard. Van Nostrand Reinhold.
- MORRISON, Mike. The magic of image processing. Estados Unidos, Sams Publishing.
- NELSON, Mark. The data compression book. Estados Unidos, Ed. M & T Books, 1992.
- SCHALKOFF, Robert. Digital image processing and computer vision. Estados Unidos, Wiley.
- VETTERLI, Martin. "Multi-dimensional sub-band coding: Some theory and algorithms" en Signal processing. Vol. 6, n.2, Estados Unidos, abril 1984, pp. 97-112.
- VETTERLI, Martin. Wavelets and subband coding. Estados Unidos, Prentice-Hall, 1995.
- WALLACE, Gregory. "The JPEG still picture compression standard" en Communications of the ACM. Vol. 34, n.4, Estados Unidos, abril 1991, pp. 31-44.
-

ANEXO A:

TÉCNICAS DE COMPRESIÓN SIN PÉRDIDA DE INFORMACIÓN.

A.1. CODIFICACIÓN RUN-LENGTH.

Debido a la correlación entre píxeles colindantes, se forman secuencias de píxeles repetidos en una imagen digital. Este hecho es particularmente cierto para las imágenes binivel. Una idea simple es utilizar un número contador para reemplazar una secuencia de píxeles repetidos. Un desarrollo más amplio de esta idea lleva a un método de compresión llamado Codificación run-length (codificación de longitud de corrida). En la codificación run-length, un píxel inicial representado por un símbolo fuente específico se considera como el comienzo de una corrida (una secuencia de píxeles idénticos consecutivos), y la longitud de la corrida se usa para reemplazar a la corrida. En la figura A.1 se muestran dos ejemplos de imágenes digitales y los resultados de la codificación run-length correspondiente.

Para separar las secuencias de diferentes corridas, existen dos posibles soluciones:

1. Usar un número de longitud de corrida y un símbolo fuente si el símbolo fuente no es un dígito o un símbolo especial para indicar cada símbolo fuente de dígito específico.
2. Diseñar palabras de código para cada longitud de corrida y para cada símbolo fuente asociado.

El método de codificación run-length es particularmente útil para codificar imágenes binivel. Por ejemplo, los estándares de compresión CCITT T.4 y T.6 usan codificación run-length.

A.2. CODIFICACIÓN SHANNON-FANO.

Para una fuente de información dada, de la teoría de la información de Shannon se sabe que la mejor tasa de compresión que puede lograrse es la entropía fuente. La siguiente cuestión es cómo conseguirla. A finales de los 40's, Shannon y Fano descubrieron un método de compresión que depende de las probabilidades tentativas de los símbolos fuente en un mensaje el cual es una secuencia de símbolos fuente. Su método se conoce como la codificación Shannon-Fano. La idea básica detrás de la codificación Shannon-Fano es utilizar una longitud variable de los bits para codificar los símbolos fuente de acuerdo a sus probabilidades. Entre más frecuentemente aparezca un símbolo, se usará un longitud de código menor para codificarlo.

Imagen Digital	Resultados de Codificación Run-Length
\$	40\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$*\$	19\$2*19\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$*^*\$	17\$2*2^2*17\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$*^*^*\$	15\$2*6^2*15\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$*^*^*^*\$	15\$2*6^2*15\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$*^*^*^*^*\$	15\$2*6^2*15\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$*^*^*^*^*^*\$	15\$2*6^2*15\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$*^*^*^*^*^*^*\$	17\$2*2^2*17\$
\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$*\$	19\$2*19\$
\$	40\$

(a) Imagen digital representada por tres símbolos fuente { \$, *, ^ } y los resultados de la codificación run-length.

Imagen Digital	Resultados de Codificación run-length
000001111100000000000000000000	5w5b20w
000000111110000000000000000000	6w5b19w
000000011111000000000000000000	7w5b18w
000000001111100000000000000000	8w5b17w
000000000111110000000000000000	9w5b16w
000000000011111000000000000000	10w5b15w
000000000001111100000000000000	11w5b14w
000000000000111110000000000000	12w5b13w

(b) Imagen digital blanco y negro representada por dos símbolos fuente { 0, 1 } y los resultados de la codificación run-length con los símbolos especiales { w, b } (donde w: blanco y b: negro).

Figura A.1. Dos ejemplos de codificación run-length de imágenes digitales (KOU, Weidong, Digital image compression. Algorithms and standards).

El algoritmo de Shannon-Fano es el siguiente:

- Paso 1. Ordenar los símbolos fuente con sus probabilidades en un orden de probabilidades descendente.
- Paso 2. Dividir el conjunto completo de símbolos en dos partes de forma tal que cada parte tenga una probabilidad igual o aproximadamente igual.
- Paso 3. Codificar los símbolos en la primera parte con el bit 0 y los símbolos en la segunda parte con el bit 1.
- Paso 4. Regresar al paso 2, continuar el proceso recursivamente para cada dos particiones hasta que cada subdivisión contenga solamente un símbolo.

Ejemplo A.2.1.

Asúmase que una imagen digital tiene ocho símbolos fuente: $\{ S_0, S_1, S_2, S_3, S_4, S_5, S_6, S_7 \}$, con las siguientes probabilidades: $\{0.1, 0.19, 0.21, 0.3, 0.05, 0.05, 0.07, 0.03\}$. El procedimiento para obtener el código Shannon-Fano se muestra en la figura A.2.

Símbolos Ordenados	Procedimiento de Diseño de Código				Código		
S_3	0.3	0.51	0.3	0	00		
S_2	0.21	0	0.21	1	01		
S_1	0.19		0.19	0	100		
S_0	0.1		0.1	1	101		
S_6	0.07		0.07	0	1100		
S_5	0.05		0.12	0	0.05	1	1101
S_4	0.05		0.05	0	1110		
S_7	0.03		0.08	0.03	1	1111	

Figura A.2. Ejemplo de la codificación Shannon-Fano (KOU, Weidong. *Digital image compression. Algorithms and standards*).

Los símbolos fuente se ordenan de acuerdo a sus probabilidades, en la columna de la izquierda. Se realiza el agrupamiento y división de las probabilidades junto con la asignación de su código correspondiente. Las palabras de código resultantes se listan en la columna de la derecha.

La codificación Shannon-Fano es el primer método de compresión exitoso acerca del problema del procedimiento óptimo de codificación de la fuente. La longitud de código promedio C_L de la codificación Shannon-Fano se encuentra en el rango de $H \leq C_L < H + 1$, donde H es la entropía de la fuente. Para el ejemplo anterior, la longitud de código promedio es 2.69, y la entropía de la fuente es 2.63.

A.3. CODIFICACIÓN HUFFMAN.

De la teoría de la información de Shannon se sabe que para una fuente dada, existe una técnica de codificación tal que la fuente puede codificarse con una longitud de código promedio de o cercana a la entropía de la fuente. La técnica Shannon-Fano es eficiente, produciendo una longitud de código promedio C_L limitada por $H + 1$. Sin embargo, el método Shannon-Fano no es óptimo ya que algunas veces asigna un código más largo a un símbolo fuente de mayor probabilidad en lugar de a uno de menor probabilidad, produciendo una longitud de código promedio mayor. Puede probarse que una técnica de codificación de longitud variable óptima asigna un código más largo a un símbolo fuente menos probable, y que los dos símbolos menos probables tienen palabras de código iguales en longitud y difieren sólo en el bit final. En 1952, D.A. Huffman desarrolló una técnica de codificación que produce la longitud de código promedio más corta posible dados un conjunto de símbolos fuente y sus probabilidades asociadas. La codificación Huffman logra la longitud de código promedio más corta, con tal que todas las probabilidades de los símbolos fuente sean potencias exactas de $1/2$. Generalmente, puede mostrarse que la longitud de código Huffman promedio está limitada por $H + p + \log[2(\log e)/e] = H + p + 0.086$, donde p es la probabilidad del símbolo fuente más probable, y H es la entropía fuente.

El algoritmo Huffman es el siguiente:

Paso 1. Listar las probabilidades de los símbolos fuente, y producir un conjunto de nodos tomando a estas probabilidades como las hojas de un árbol binario.

- Paso 2. Tomar los dos nodos con las dos probabilidades más pequeñas del conjunto de nodos, y generar una nueva probabilidad la cual es la suma de estas dos probabilidades.
- Paso 3. Producir un nodo padre con la nueva probabilidad, y marcar la rama de su nodo hijo de la parte superior (o izquierda) con 1 y la rama de su nodo hijo de la parte inferior (o derecha) con 0, respectivamente.
- Paso 4. Actualizar el conjunto de nodos reemplazando los dos nodos con las dos probabilidades más bajas con los nodos recién producidos. Si el conjunto de nodos contiene solamente un nodo, salir. De otra forma, regresar al paso 2.

Ejemplo A.3.1.

Asúmase que se toman los mismos símbolos fuente y las probabilidades asociadas que en el ejemplo A.2.1. Siguiendo el algoritmo Huffman, se tiene el procedimiento de diseño de código como el que se muestra en la figura A.3.

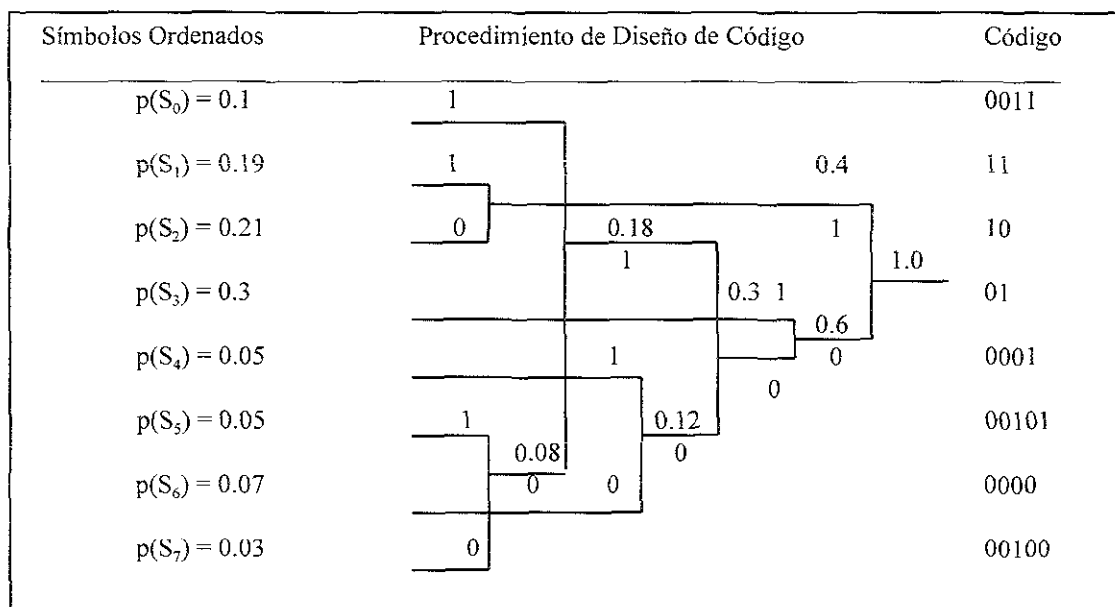


Figura A.3. Ejemplo de la codificación Huffman (KOU, Weidong, Digital image compression. Algorithms and standards).

La longitud de código promedio obtenida en este ejemplo es de 2.68, mejorando el rendimiento de la codificación Shannon-Fano donde se obtuvo un resultado de 2.69 con los mismos datos. Generalmente, si la fuente tiene muchos símbolos y no está distribuida

uniformemente, la codificación Huffman es mucho mejor que la Shannon-Fano en términos de eficiencia de compresión.

Si bien la codificación Huffman es eficiente, debe entenderse que la codificación Huffman es óptima solamente si se conoce de antemano la distribución de probabilidad de la fuente y si cada símbolo se codifica usando una cantidad integral de bits. En la práctica, la codificación Huffman se realiza sobre un conjunto de imágenes, pero después se aplica a muchas otras imágenes, cada una de las cuales tiene su propia distribución de probabilidad de sus símbolos. Por lo tanto, la codificación Huffman no necesariamente es óptima para cualquier tipo de imagen.

Sin embargo, la codificación es una de las técnicas de compresión más importantes y se aplica en varios estándares de compresión como el JPEG.

A.4. CODIFICACIÓN ZIV-LEMPEL.

La codificación basada en diccionario es una de las técnicas de compresión usadas más frecuentemente. Por medio de la construcción de un diccionario, de un mensaje, el cual es una secuencia de símbolos de entrada, puede codificarse después como una secuencia de entradas de referencia hacia el diccionario. Un ejemplo simple es buscar la palabra “terminología” en un diccionario. Si esta palabra aparece en la i -ésima palabra de la página x , entonces la palabra “terminología” puede representarse por medio de una entrada de referencia (x,y) . De forma similar, Una frase como “compresión es una terminología” puede codificarse como una secuencia de referencia: $(x_1,y_1) (x_2,y_2) (x_3,y_3) (x_4,y_4)$. La codificación basada en diccionario puede ser codificación de diccionario estático o codificación de diccionario dinámico. En la codificación de diccionario estático, el diccionario no se altera durante los procesos de codificación y decodificación ni en el lado emisor ni en el receptor. Sin embargo, la codificación de diccionario dinámico construye un diccionario dinámico basándose en el mismo mensaje que se ha codificado en el lado transmisor y decodificado en el lado receptor.

La codificación Ziv-Lempel es una técnica de codificación de diccionario dinámico que genera una familia de algoritmos de compresión. El trabajo original fue presentado por Ziv y Lempel en 1977 y 1978 respectivamente. Desde entonces Esta técnica de codificación ha sido desarrollada y ampliada por muchos otros. Un algoritmo conocido de la familia Ziv-Lempel es el algoritmo LZW (Lempel-Ziv-Welch), el cual se ha utilizado ampliamente en la industria de la computación. Es conocido como el comando *compress* en el sistema UNIX y en muchos otros.

En la industria de las imágenes, el formato TIFF (formato de archivo de imagen de punto) efectúa la compresión Ziv-Lempel. Esta técnica es una herramienta útil para comprimir imágenes binarias sombreadas, las cuales simulan ser imágenes de escala de gris por medio de una variación en la densidad de los puntos negros, y a menudo produce mejor eficiencia de compresión comparada con la compresión CCITT T.4 o la T.6.

La idea básica de la codificación Ziv-Lempel es que la parte actual de un mensaje puede reemplazarse por una referencia hacia su ocurrencia previa a través de la actualización dinámica del diccionario.

Para mostrar cómo trabaja la codificación Ziv-Lempel se tomará como ejemplo el denominado algoritmo de compresión LZW, el cual es el siguiente:

- Paso 1. Crear la primera entrada en el diccionario agregando una entrada vacía (entrada en blanco) al diccionario y configurando el índice de entrada del mensaje vacío en el diccionario como n .
- Paso 2. Agregar 256 palabras de código de 1 byte adicionales desde 0x00 hasta 0xFF (representación hexadecimal), asumiendo que la longitud de byte es 8.
- Paso 3. Leer un byte b a la vez desde un mensaje a codificar. Si la palabra $\langle n, b \rangle^*$ se encuentra en el diccionario, se actualiza n como el índice de entrada de referencia de la palabra $\langle n, b \rangle$. De otra forma, se transmite o se almacena n . Si el diccionario no está lleno, se agrega la palabra $\langle n, b \rangle$ al diccionario. Se actualiza n como el índice de entrada de referencia del byte b .
- Paso 4. Repetir el paso 3 hasta que todo el mensaje sea codificado y transmitido o almacenado.

Ejemplo A.4.1.

En este ejemplo primero se toman algunos bytes de una imagen y se codifican usando el algoritmo LZW:

00000000 00000011 10001111 11110000 00000000 11111111 11110000

* La palabra $\langle n, b \rangle$ es la n -ésima entrada en el diccionario seguida por el byte b .

Siguiendo el paso 1 y el paso 2 del algoritmo, se tendría el diccionario inicial con 257 entradas comenzando desde el índice de entrada 0. Como el primer byte 00000000 se encuentra en el diccionario con el índice de entrada 1, se reasigna $n=1$. Después, se lee el segundo byte 00000011. Como la palabra $\langle 1,00000011 \rangle = 00000000\ 00000011$ no está en el diccionario, se envía $n=1$, se agrega 00000000 00000011 al diccionario con el índice de entrada 257, y se reasigna $n=4$, el índice de entrada del segundo byte 00000011. Repitiendo este proceso una vez más, se envía $n=4$ y se agrega la palabra $\langle 4,10001111 \rangle = 00000011\ 10001111$ al diccionario en la siguiente ronda para codificar el siguiente byte. Este proceso continúa hasta que todos los bytes en la imagen se hayan codificado. El proceso de codificación para este ejemplo se muestra en la tabla A.1.

Índice de Entrada	Entrada a Diccionario	Entrada	N	Salida
0	Entrada Vacía			
1	00000000			
...	...			
256	11111111			
			0	
		00000000	1	
257	00000000 00000011	00000011	4	1
258	00000011 10001111	10001111	144	4
...

Tabla A.1. Proceso de codificación del ejemplo A.4.1 (KOU, Weidong, Digital image compression. Algorithms and standards).

El algoritmo descompresión LZW simplemente reemplaza el índice de entrada por la palabra de código de entrada misma. Debido a que no hay ninguna búsqueda involucrada, la descompresión es muy eficiente. Esta es una característica muy atractiva para aplicaciones asimétricas de imágenes que requieren descompresión rápida.

Una desventaja del algoritmo LZW es que cuando el diccionario se llena, se convierte en un diccionario estático y por la misma razón se reduce la eficiencia de la compresión.

Otros algoritmos de compresión de la familia Ziv-Lempel varían en sus decisiones de diseño, sin embargo, el principio de funcionamiento es el mismo que el del algoritmo LZW ya que todos se basan en codificación de diccionario dinámico.

A.5. CODIFICACIÓN ARITMÉTICA.

En la codificación basada en diccionario, un mensaje se representa como una secuencia de índices de diccionario. Un mensaje puede codificarse en muchas formas distintas. La codificación aritmética es otra forma para codificar un mensaje. En la codificación aritmética un mensaje se codifica como un número real en un intervalo de cero a uno. La codificación aritmética juega un papel muy importante en los estándares de imágenes como el JBIG y el JPEG.

Existen dos conceptos fundamentales en la codificación aritmética: la probabilidad de un símbolo y el rango de su intervalo de codificación. Las probabilidades de los símbolos fuente determinan la eficiencia de la compresión. También determinan los rangos de intervalo de los símbolos fuente para el proceso de codificación. Los rangos de intervalo se encuentran dentro del intervalo de cero a uno. Los rangos de intervalo para el proceso de codificación determinan la salida de la compresión.

El proceso de codificación de la codificación aritmética puede explicarse a través del ejemplo A.5.1.

Ejemplo A.5.1.

Asúmase que los símbolos fuente son $\{00, 01, 10, 11\}$ y las probabilidades de estos símbolos son $\{0.1, 0.4, 0.2, 0.3\}$, respectivamente. Entonces, basándose en estas probabilidades, el intervalo $[0, 1)$ puede dividirse en cuatro subintervalos: $[0, 0.1)$, $[0.1, 0.5)$, $[0.5, 0.7)$, $[0.7, 1)$, donde $[x, y)$ denota un intervalo medio abierto, el cual incluye x pero excluye y . La información anterior se resume en la tabla A.2.

Símbolos	00	01	10	11
Probabilidades	0.1	0.4	0.2	0.3
Intervalos de Codificación Iniciales	$[0, 0.1)$	$[0.1, 0.5)$	$[0.5, 0.7)$	$[0.7, 1)$

Tabla A.2. Símbolos fuentes, sus probabilidades e intervalos de codificación iniciales (KOU, Weidong. Digital image compression. Algorithms and standards).

Para codificar un mensaje de una secuencia binaria: 10 00 11 00 10 11 01, se toma el primer símbolo del mensaje (10) y se encuentra en la tabla A.2. que su rango de codificación es $[0.5, 0.7)$. Como el rango del segundo símbolo del mensaje (00) es $[0, 0.1)$ y su probabilidad es de 0.1, éste se codifica tomando la décima parte del intervalo $[0.5, 0.7)$

(0.02) y ubicando el nuevo intervalo en la posición correspondiente al intervalo del nuevo símbolo, obteniéndose como el nuevo intervalo $[0.5, 0.52)$. De forma similar, al codificar el tercer símbolo 11 se obtiene un nuevo intervalo $[0.514, 0.52)$, el cual corresponde al 30 por ciento final del intervalo anterior. Después de codificar el cuarto símbolo (00), el nuevo intervalo es $[0.514, 0.5146)$. El codificar el quinto símbolo (10) se produce el nuevo intervalo $[0.5143, 0.51442)$. El siguiente intervalo es $[0.514384, 0.51442)$ para el sexto símbolo (11). El último intervalo es $[0.51439, 0.5143948)$. La salida de la compresión de este mensaje puede ser cualquier número en el último intervalo, como puede ser el límite izquierdo, 0.51439. El proceso de codificación anterior se esquematiza en la figura A.4.

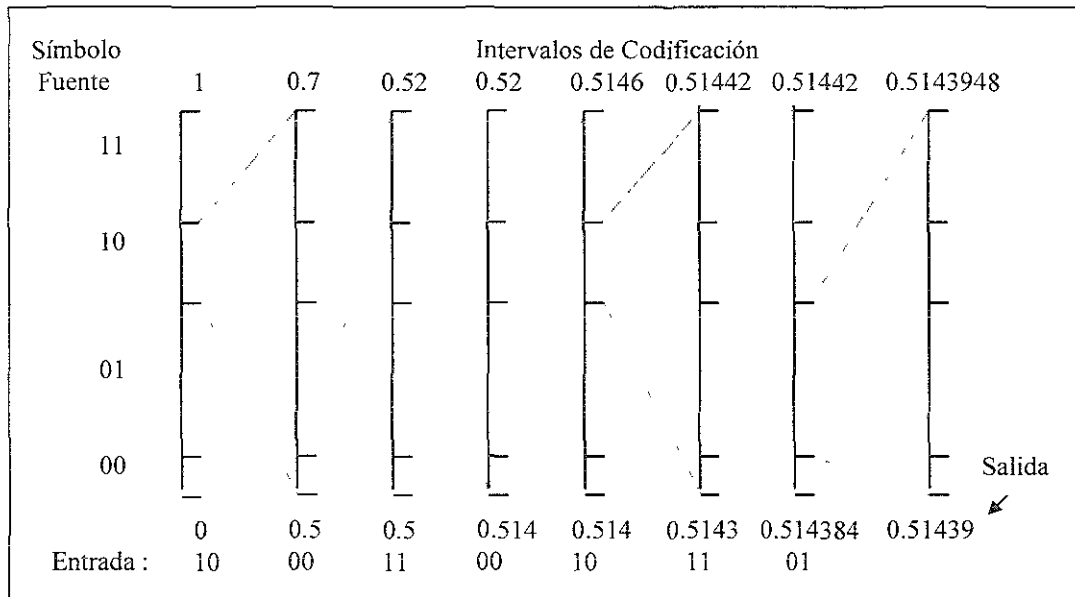


Figura A.4 Esquema del proceso de codificación del ejemplo A.5.1 (KOU, Weidong, Digital image compression. Algorithms and standards).

Si se envía el número de salida 0.51439 al receptor, el decodificador aritmético puede descomprimirlo fácilmente en el mensaje original examinando a que intervalo pertenece el número.

Los procesos de codificación y decodificación se resumen en las tablas A.3 y A.4.

Paso	Símbolo de Entrada	Intervalo de Codificación	Decisión de Codificación
1	10	[0.5, 0.7)	El rango del símbolo es [0.5, 0.7)
2	00	[0.5, 0.52)	La 1ª décima parte de [0.5, 0.7)
3	11	[0.514, 0.52)	Las tres últimas décimas partes de [0.5, 0.52)
4	00	[0.514, 0.5146)	La 1ª décima parte de [0.514, 0.52)
5	10	[0.5143, 0.51442)	En la 5ª décima parte, dos décimas partes de [0.514, 0.5146)
6	11	[0.514384, 0.51442)	Las tres últimas décimas partes de [0.5143, 0.51442)
7	01	[0.51439, 0.5143948)	En la 2ª décima parte, cuatro décimas partes de [0.514384, 0.51442)
8	Seleccionar un número del intervalo [0.51439, 0.5143948) como la salida : 0.51439		

Tabla A.3. Proceso de codificación del ejemplo A.5.1 (KOU, Weidong. Digital image compression. Algorithms and standards).

Paso	Rango	Símbolo Decodificado	Decisión de Decodificación
1	[0.5, 0.7)	10	0.51439 está en [0.5, 0.7)
2	[0.5, 0.52)	00	0.51439 está en la 1ª décima parte de [0.5, 0.7)
3	[0.514, 0.52)	11	0.51439 está entre las tres últimas décimas partes de [0.5, 0.52)
4	[0.514, 0.5146)	00	0.51439 está en la 1ª décima parte de [0.514, 0.52)
5	[0.5143, 0.51442)	10	0.51439 está en la 5ª décima parte de [0.514, 0.5146)
6	[0.514384, 0.51442)	11	0.51439 está entre las tres últimas décimas partes de [0.5143, 0.51442)
7	[0.51439, 0.5143948)	01	0.51439 está en la 4ª décima parte de [0.514384, 0.51442)
8	El mensaje decodificado es 10 00 11 00 10 11 01		

Tabla A.4. Proceso de decodificación del ejemplo A.5.1 (KOU, Weidong. Digital image compression. Algorithms and standards).

En el ejemplo anterior, se asume que tanto el codificador como el decodificador conocen la longitud del mensaje por lo que el decodificador no continuaría el proceso de decodificación sin detenerse. En la práctica, es necesario incluir un símbolo de terminación especial para que cuando el decodificador encuentre el símbolo terminal, detenga el proceso de decodificación.

Hay algunos problemas relacionados con la codificación aritmética:

- Como no existe una máquina con precisión infinita, son obvios los problemas de sobreflujo (overflow) y subflujo (underflow) para las máquinas actuales, la mayoría con 16, 32 o 64 bits de precisión.
- Un codificador aritmético produce solamente una palabra de código, un número real en el intervalo de [0, 1) para codificar el mensaje completo a transmitir. No es posible efectuar ningún proceso de decodificación hasta que se hayan recibido todos los bits que representen dicho número.

- La codificación aritmética es un esquema de compresión sensible al error. Un error de un solo bit puede distorsionar el mensaje completo.

Estos problemas han sido investigados por muchos investigadores, y se han propuesto muchos esquemas. Por ejemplo, es posible usar un proceso de escalado para solucionar los problemas de sobreflujo y subflujo.

La codificación aritmética puede ser estática o adaptativa. En la codificación aritmética estática, se fijan las probabilidades de los símbolos fuente. En la codificación aritmética adaptativa, las probabilidades de los símbolos fuente se estiman dinámicamente basándose en la frecuencia de cambio en los símbolos que se han detectado en el mensaje a codificar. El proceso para estimar las probabilidades de los símbolos fuente a partir de una parte del mensaje detectado hasta cierto momento durante la codificación se conoce como “modelado”. Como usualmente las probabilidades de los símbolos fuente son desconocidas o inconvenientes para trabajar, no debe esperarse una eficiencia máxima al comprimir un mensaje. Lo mejor que puede hacerse es estimar las probabilidades sobre el desarrollo del proceso de codificación. Por lo tanto, el modelado dinámico se convierte en la clave para determinar la eficiencia de la compresión de un codificador aritmético.