

CAPÍTULO II

MARCO TEÓRICO

2.1 Introducción.

En este capítulo se revisan los fundamentos teóricos, que originan los métodos propuestos en la compresión de imágenes fijas, se inicia con la revisión de conceptos básicos para tener la idea de la reducción del almacenamiento de una imagen y continuando con las transformaciones como método para cambiar la imagen a otra dimensión que puede ser mejor controlada por la concentración de energía en unos cuantos coeficientes.

2.2 Representación de una imagen

Una imagen monocroma o simplemente imagen puede variar en la apreciación que pueda tener un observador sobre ésta, pero en otro sentido esta referida a una función bidimensional de intensidad de luz $f(x,y)$, donde x e y representan las coordenadas espaciales y el valor de f en un punto cualquiera (x,y) es proporcional al brillo (o nivel de gris) de la imagen en ese punto [1]. La asignación de los ejes a una imagen puede cambiar dependiendo de las conveniencias para quien la presenta, en este caso la ubicación de los ejes será como se indica en la figura 2.1.



Figura 2.1 Utilización de ejes en una imagen

Por otra lado, una imagen digital es aquella que se ha discretizado tanto en las coordenada x, y como en los niveles de gris. Una imagen digital puede ser representada por una matriz, donde la posición de los elementos representan un punto de la imagen y el valor que tiene cada elemento de la matriz representa el nivel de gris en ese punto. A estos elementos también se les llama píxels o pels (derivado de la abreviatura del inglés picture elements).

2.3 Muestreo y cuantificación de una imagen

En una imagen digital, a la digitalización de las coordenadas espaciales (x, y) se les denomina muestreo de la imagen y la digitalización de la amplitud es la cuantificación del nivel de gris.

De lo anterior se puede establecer que una imagen continua, puede ser representada de forma aproximada, por una serie de muestras igualmente espaciadas, esto es representable por una matriz de $M \times N$, como se muestra en la ecuación 2.1, donde el término de la derecha será la imagen digital.

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & f(0,2) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,2) & \dots & f(1,N-1) \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ f(M-1,0) & f(M-1,1) & f(M-1,2) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

2.4 Almacenamiento de una imagen digital.

En el proceso de digitalización los valores de M, N y los niveles de gris permitidos son los que determinan la cantidad de bits requeridos para almacenar una imagen, con las características de la parte derecha de la ecuación 2.1. Es practica común que estas cantidades sean potencias de 2. Por esto,

$$N = 2^n \quad (2.2)$$

$$M = 2^K \quad (2.3)$$

$$G = 2^m \quad (2.4)$$

en donde m , n y k son valores enteros y G representa el número de niveles de grises. Bien, la cantidad de bits requeridos para almacenar una imagen digitalizada se puede determinar con :

$$b = MxNxm \quad (2.5)$$

y si $M=N$, entonces

$$b = N^2 m \quad (2.6)$$

Por ejemplo, una imagen de 256×256 puntos con 128 niveles de gris necesita 458,752 bits de memoria. Si el número de pixels y niveles de grises aumenta, la memoria requerida es mayor, unos ejemplos para imágenes cuadradas, están resumidos en la tabla 2.1.

m \ N	1	2	3	4	5	6	7	8
32	1024	2048	3072	4096	5120	6144	7168	8192
64	4096	8192	12288	16384	20480	24576	28672	32768
128	16384	32768	49152	65536	81920	98304	114688	131072
256	65536	131072	196608	262144	327680	393216	458752	524288
512	262144	524288	786432	1048576	1310720	1572864	1835008	2097152

Tabla 2.1 Número de bits requeridos para almacenar una imagen de valores de N y m .

2.5. Resolución de una imagen digital.

La resolución de una imagen digital depende estrechamente del muestreo y cuantificación del nivel de gris. Entre mas se incrementan estos parámetros, mas se aproxima una imagen digitalizada a una imagen original. Aunque el aumentar estos parámetros, trae como consecuencia el aumento de bits requeridos para almacenar la imagen (de acuerdo con la ecuación 2.6) y mayores requerimientos en el procesamiento.

La calidad de una imagen es subjetiva y depende mucho de las necesidades de una aplicación determinada y no es fácil definirla como ‘una buena imagen’. Bien, si la resolución espacial de una imagen digital se reduce, como una forma de reducir el numero de bits requeridos para almacenarla, tiene los efectos que se muestran en la figura 2.2. La imagen original esta en la figura 2.2 (a), que muestra una imagen digital de 512 x 512 pixels y 256 niveles de gris. Mientras en las figuras 2.2(b) a la 2.2(d) se muestran los resultados de reducir la resolución espacial de $N = 512$ a $N = 256, 128$ y 64 respectivamente. En todos los casos, el número de niveles de gris no se redujo. Todas las imágenes representan la misma área de 512×512 puntos, por lo que en las imágenes con menor resolución los pixels han sido duplicados para cubrir toda el área. Esta acción es la que produce que en las imágenes de menor resolución, la imagen tiende a parecer un tablero de ajedrez.

Por otra parte, puede también optarse por la disminución de los niveles de gris requeridos para representar a la imagen, pero produce lo que se denomina *falso contorno*, originado por un número insuficiente de niveles de gris en las áreas mas suaves de una imagen digital[1].

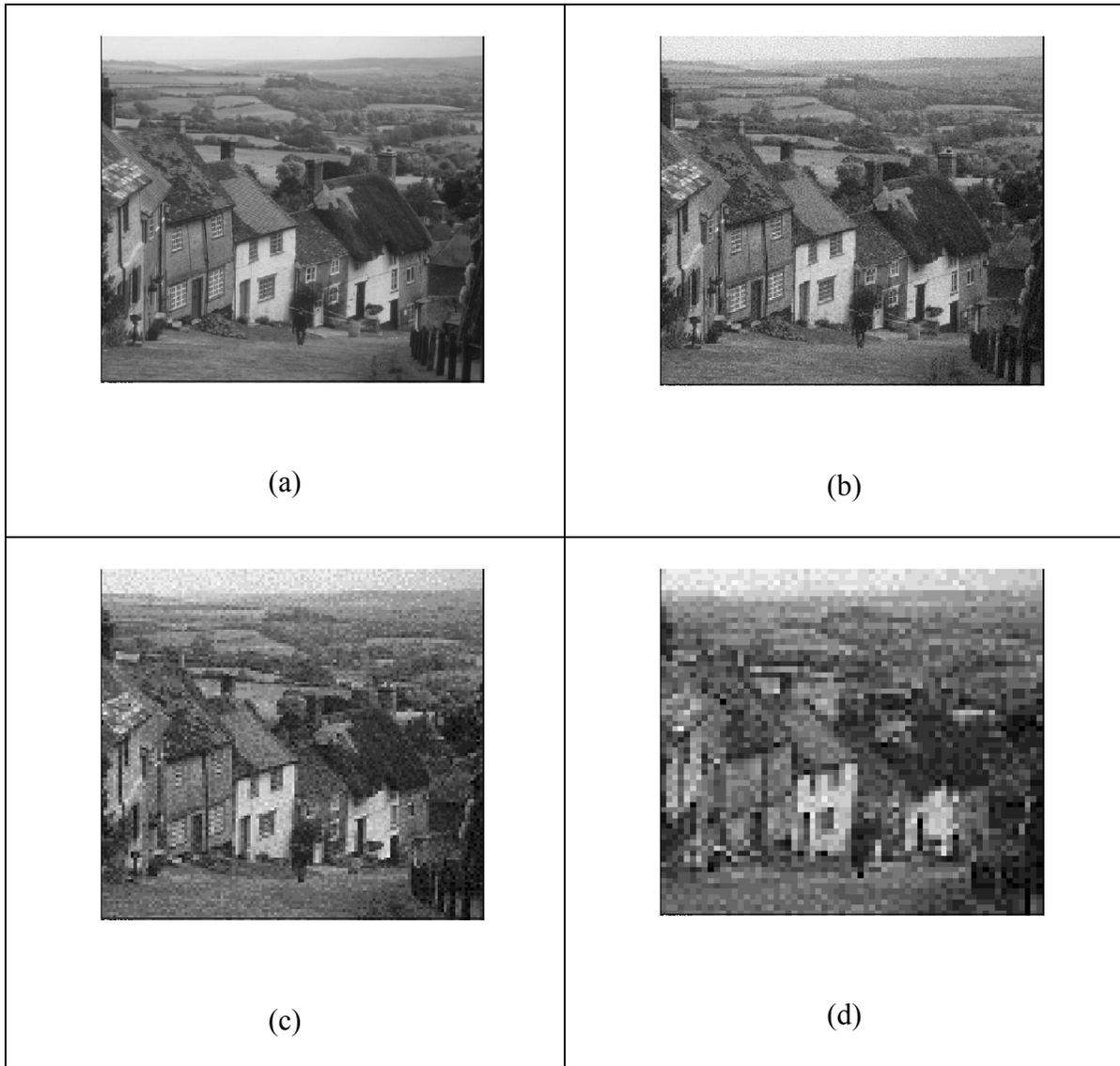


Figura 2.2 Efectos que se producen al reducir la resolución espacial.

2.6 Compresión de imágenes.

Como se analizó en las secciones anteriores, el muestreo y cuantificación de una imagen para crear una imagen digital, produce una gran cantidad de datos. Estos datos pueden llegar a ser tan grandes que su almacenamiento, procesamiento y comunicación pueden llegar a ser excesivos para una aplicación práctica.

Mediante la compresión de imágenes se puede enfrentar esta situación, reduciendo la cantidad de datos necesarios para representar una imagen digital, teniendo como base para la reducción de los datos, la eliminación de los datos redundantes. Las técnicas de compresión de imágenes se pueden clasificar en dos categorías: compresión sin pérdidas y compresión con pérdidas.

Algunos conceptos importantes que se utilizan en la compresión de una imagen digital, se presentan a continuación.

2.6.1 Compresión de los datos

La compresión de datos se refiere al proceso de reducción del volumen de datos (los datos son los medios a través de los que se conduce la información) necesarios para representar una determinada cantidad de información.

2.6.2 Redundancia de los datos.

Es cuando se contienen datos que contiene información sin relevancia o bien vuelve a decirse lo que ya se sabía.

2.6.3 Redundancia relativa de los datos.

Si n_1 y n_2 representan el número de unidades de información de dos conjuntos de datos que representan la misma información, la redundancia relativa de los datos R_D del primer conjunto de datos (el caracterizado por n_1) se puede definir como [1]:

$$R_D = 1 - \frac{1}{C_R} \quad (2.7)$$

donde C_R , es la relación de compresión, y esta definida por:

$$C_R = \frac{n_1}{n_2} \quad (2.8)$$

2.6.4 Redundancia de codificación.

Sea r_k una variable aleatoria discreta del intervalo de $[0,1]$, que representa los niveles de gris de una imagen y que cada r_k sucede con una probabilidad $p_r(r_k)$, entonces

$$p_r(r_k) = \frac{n_k}{n} \quad k=0,1,2,3 \dots, L-1 \quad (2.9)$$

donde L es el número de nivel de gris, n_k el número de veces que aparece en la imagen el k -ésimo nivel de gris, n el número de pixels de la imagen.

El número de bits empleados para representar cada valor de r_k es $l(r_k)$, *el promedio de bits necesarios para representar cada píxel* es:

$$L_{med} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k) \quad (2.10)$$

también llamada la *longitud media de las palabras de código* asignadas a los valores de los diversos niveles de gris. De lo anterior, el número de bits necesarios para codificar una imagen de $N \times M$ es $N \times M L_{med}$.

Si los niveles de gris de una imagen están codificados de forma que se emplean mas símbolos de los estrictamente necesarios para representar cada uno de ellos(es decir, el código no cumple con la ecuación 2.10), entonces se dice que la imagen resultante tiene *redundancia de código*[1].

2.6.5 Entropía

Al definir entropía se emplea el termino fuente, por lo se inicia definiendo este concepto. Una fuente es un par ordenado $F = (S, P)$, donde $S = s_1, s_2, s_3, \dots, s_n$ es un conjunto finito de mensajes o símbolos denominado alfabeto de la fuente y P es la distribución de probabilidad de S . La probabilidad de un mensaje s_i se denota como $P(s_i)$ o bien como p_i .

Ahora, considere una función que mapea fuentes en medidas de información, usando la notación $h(p_1, p_2, p_3, \dots, p_n)$ para denotar dicha función, a esta medida se le llama entropía de la fuente, la cual se define a continuación:

La Entropía de una fuente $F = (S, P)$, denotada como $h_b(p_1, p_2, p_3, \dots, p_n)$ es definida como:

$$h_b(p_1, p_2, p_3, \dots, p_n) = -\sum_{i=1}^n p_i \log_b p_i = \sum_{i=1}^n p_i \log_b \frac{1}{p_i} \quad (2.11)$$

La unidad de información está determinada por la base b del logaritmo. Si $b = 2$, entonces la entropía se mide en bits; y si $b = 10$ entonces la entropía se mide en *hartleys*.

La propiedad más importante de la entropía es que es exactamente la mínima longitud promedio de código posible para una fuente. Por ejemplo, si la entropía de una fuente es de 4 bits, entonces no existe otra posibilidad de codificar los mensajes de esa fuente, de modo que, en promedio, posea una longitud de código menor a 4 bits.

Veamos este ejemplo, sea F , una fuente con probabilidades $\left(\frac{3}{8} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8}\right)$. La entropía para este caso, usando la base 2 es:

$$h_2\left(\frac{3}{8}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}\right) = \frac{3}{8} \cdot \log_2 \frac{8}{3} + \frac{1}{4} \cdot \log_2 4 + \frac{1}{4} \cdot \log_2 4 + \frac{1}{8} \cdot \log_2 8 = 1.9056 \text{ bits}$$

Generalizando, cuando se construye un compresor utilizando un codificador por entropía, se desea que este último se acerque lo mas posible a la entropía. De este modo, esta medida puede ser utilizada para estimar inferiormente los bits por símbolo que tendrá la información comprimida. Al determinar la entropía de una imagen, se está definiendo implícitamente una fuente $F = (S, P)$. Por ejemplo, si es una imagen en escala de grises con valores entre $[0,255]$, a S lo integran los números enteros entre 0 y 255, mientras que a P lo integran las frecuencias de los números enteros en la imagen.

2.6.6 Medidas de la compresión.

Existen varias formas de expresar la compresión obtenida en un proceso específico. Se han seleccionando algunas que tienen relación con los sistemas que se desarrollan en los últimos capítulos.

En las definiciones que se utilizan en esta sección, se usan las siguientes nomenclaturas, con el significado que se indica:

O: longitud del archivo original (en bytes)

C: longitud del archivo comprimido (en bytes)

Longitud media por símbolo(LMPS).- Es la relación entre la longitud del archivo comprimido y la longitud del archivo sin comprimir, ambos expresado en bytes.

$$LMPS = \frac{C}{O} \cdot 8 \text{ bps (bits por símbolo)} \quad (2.12)$$

Es necesario aclarar que anteriormente se trato este mismo concepto, pero fue definido en función de la probabilidad de cada ocurrencia de los símbolos.

Factor de compresión (FC).- Es la relación entre la longitud del archivo original y del archivo comprimido.

$$FC = \frac{O}{C} \quad (2.13)$$

Compresión relativa(CR).- Representa lo que ha disminuido el archivo respecto del tamaño original, mediante la siguiente expresión.

$$CR = \frac{O - C}{O} \quad (2.14)$$

2.7 Compresión sin errores.

La compresión sin errores en diversas aplicaciones, es deseable y a veces la única manera que se acepta al reducir los datos. Por ejemplo, en evidencias del ámbito legal o bien una radiografía digital. En todos los casos la compresión sin errores, es motivada por la futura utilización de las imágenes en proceso.

Hay varias estrategias de compresión sin errores. Entre la principales están : codificación de longitud variable y la codificación de planos de bits. Dentro de las principales técnicas de codificación de longitud variables están: codificación de Huffman y codificación aritmética. En la codificación de planos de bits podemos encontrar como principales: descomposición de planos de bits y codificación por zonas constantes.

Nuestro interés se centra, en la codificación de longitud variable y como caso especial la codificación Huffman, que son los que de alguna manera se relacionan con los sistemas desarrollados en los capítulos finales.

2.7.1 Codificación de longitud variable.

Consiste en reducir la redundancia de codificación, como se vio antes, esta redundancia se puede eliminar codificando los niveles de gris de modo que se minimice la ecuación (2.10).

Lo anterior se logra, construyendo un código de longitud variable que asigne las palabras de código mas pequeña a los niveles de gris mas probables.

Al emplear un código de longitud variable, es posible comprimir los datos de una fuente dada, cuando estos cumplen ciertas características.

Ya que, la compresión de las imágenes, es representada por una matriz, la cual se subdivide en matrices mas pequeñas con fines de procesamiento; entonces, con la finalidad de explicar el uso de esta codificación de longitud variable, se usará la matriz de 4x4 de la figura 2.3.

$$\begin{bmatrix} 2 & 2 & 2 & 2 \\ 4 & 2 & 3 & 2 \\ 8 & 8 & 8 & 8 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

Figura 2.3 Matriz 4x4, como datos de la fuente

Si queremos codificar los elementos de esta matriz, podemos emplear 4bits para cada elemento. Por ello quedaría codificada a 4 bits por símbolo. Sin embargo, al analizar la fuente, se observan varias repeticiones de los datos, por lo que determinamos su frecuencia y probabilidad, esto se resume en la tabla 2.2.

Elemento/símbolo	Frecuencia	Probabilidad
2	6	6/16
3	5	5/16
4	1	1/16
8	4	4/16

Tabla 2.2. Frecuencia y probabilidad de los datos de la fuente.

Con los datos de la tabla 2.2, se puede calcular la entropía de la fuente, para inferir la mínima longitud promedio de bits por símbolo posible. Lo que se determina enseguida.

$$h_2\left(\frac{6}{16}, \frac{5}{16}, \frac{1}{16}, \frac{4}{16}\right) = \frac{6}{16} \cdot \log_2 \frac{16}{6} + \frac{5}{16} \cdot \log_2 \frac{16}{5} + \frac{1}{16} \cdot \log_2 16 + \frac{4}{16} \cdot \log_2 \frac{16}{4} = 1.805 \text{ bps}$$

Con lo anterior, se nota que es posible, mediante el empleo de un código de longitud variable que asigne menos bits a los más probables, mejorar la representación de 4 bps (bits por símbolo) con lo que se representa inicialmente la matriz. Esta representación se verá con el empleo del código Huffman.

2.7.2 Codificación Huffman.

Es código de longitud variable que permite la eliminación de la redundancia de codificación. Con este código se consigue el número más pequeño posible de símbolos de código por símbolo de la fuente.

Como ejemplo, se va a usar la matriz de la figura 2.3, para aplicar el código Huffman.

Primero hay que realizar una serie de reducciones de la fuente de manera que las probabilidades de símbolos considerados, estén ordenados. El resultado se muestra en la tabla 2.3.

Elemento/símbolo	Frecuencia	Probabilidad
2	6	6/16
3	5	5/16
8	4	4/16
4	1	1/16

Tabla 2.3. Probabilidad de los datos de la fuente ordenados.

El segundo paso es reducir el número de las probabilidades, iniciando con la suma de las dos más pequeñas, la cual genera una nueva. Estas probabilidades se ordenan si es necesario y luego se vuelven a sumar las dos más pequeñas para generar la nueva probabilidad, que para el caso del ejemplo es 10/16. Este proceso se continua, hasta que queden solo dos probabilidades.

El tercer paso es asignar un cero a una de las dos probabilidades resultantes y un uno a la otra.

Finalmente, con la asignación del uno y cero del paso anterior, se regresa asignando en el mismo orden, un cero y uno a las probabilidades reducidas, hasta llegar a las probabilidades originales. El resultado final se muestra en la figura 2.4

Fuente original		Reducción de la fuente				
Elemento/ Símbolo	Probabilidad	Código final	Reducción 1	CM	Reducción 2	CI
2	6/16	1	6/16	1	10/16	0
3	5/16	00	5/16	00	6/16	1
8	4/16	010	5/16	01		
4	1/16	011				

Figura 2.4 Asignación de código con el Procedimiento de Huffman.

La longitud media por símbolo se determina empleando la ecuación 2.10 y es:

$$L_{med} = 1 \cdot \frac{6}{16} + 2 \cdot \frac{5}{16} + 3 \cdot \frac{4}{16} + 3 \cdot \frac{1}{16} = 1.9375 \text{ bps}$$

Como se puede observar, este resultado se aproxima a la entropía de la fuente, calculada anteriormente y que es de 1.805 bps. Al emplear este código, el número de bits requeridos para codificar toda la matriz es de 31 bits (ver figura 2.5), mientras que si se usaran los 4 bits planteados inicialmente se requeriría de 64 bits. De aquí que el código de Huffman sea una buena alternativa para la reducción de los datos de la fuente y por lo mismo se logre una compresión de los datos.

Elemento/ símbolo	Frecuencia	Longitud de código	Longitud en bits
2	6	1	6
3	5	2	10
4	1	3	3
8	4	3	12
Total en bits			31

Figura 2.5 Determinación del número de bits requeridos para la matriz fuente.

2.8 Compresión con pérdidas.

La codificación con pérdidas esta basada en comprometer la precisión de la imagen reconstruida que permita lograr una mayor compresión, si es posible tolerar la distorsión resultante. Muchas técnicas de compresión con pérdidas pueden lograr reproducir imágenes monocromas reconocibles con un factor de compresión de 30:1. De manera particular, la atención estará en la codificación por transformación, ya que es en lo que se basan los sistemas desarrollados de los últimos capítulos.

2.8.1 ¿ Que son las transformadas?

De forma general, una transformada representa un cambio hacia un dominio diferente, que debido a sus propiedades, facilita de alguna forma la realización de tareas determinadas. Este cambio puede caracterizarse mediante un operador, que genéricamente se denota como $T(\cdot)$. Al aplicar este operador a una señal f , se puede representar de la siguiente manera:

$$\begin{array}{ccc} f & \xrightarrow{T(\cdot)} & T(f) \\ \\ T(f) & \xrightarrow{T^{-1}(\cdot)} & f \end{array}$$

donde $T^{-1}(\cdot)$ es la transformación inversa

2.8.2 Aplicación de la transformada en el procesamiento de una imagen.

El procesamiento de una imagen puede llevarse a cabo en el dominio espacial (amplitud en función de la posición) o en el dominio transformado. La figura 2.6, muestra la configuración general del procesamiento digital de imagen basado en las transformadas.

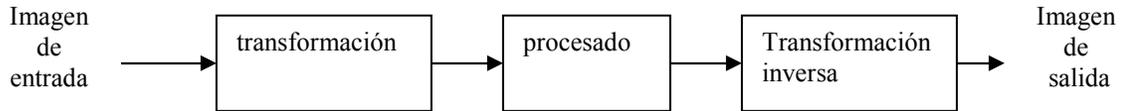


Figura 2.6 Tratamiento en el dominio transformado.

La imagen digitalizada es previamente transformada antes de efectuar el procesado. Después de que la imagen es procesada, se somete a la correspondiente transformación inversa complementaria, para conseguir el retorno de la imagen al dominio original. Este tipo de análisis permite que las imágenes sean tratadas con métodos totalmente diferentes de aquellos utilizados sobre los datos originales.

El procesado, basado en el dominio transformado consigue codificar imágenes reduciendo los requerimientos de almacenamiento y ancho de banda para la transmisión. Esto se realiza mediante el descarte o cuantificación de los coeficientes transformados de muy baja magnitud.

Un sistema típico de codificación basado en transformación es presentado en la figura 2.7. En la figura 2.7(a), una imagen entrada de $N \times N$ se subdivide en subimágenes de tamaño $n \times n$. Enseguida se aplica la transformación cuyo objetivo es deshacer la correlación de los píxeles de cada subimagen, o empaquetar tanta información como resulte posible en el menor número de coeficientes de la transformada. Se continúa con la cuantificación, que elimina selectivamente o cuantifica con menor precisión, los coeficientes que llevan la menor cantidad de información. Estos coeficientes influyen poco en la subimagen reconstruida. La etapa de codificación finaliza codificando los coeficientes cuantificados. El proceso inverso, para obtener la imagen reconstruida, se muestra en la figura 2.27(b).

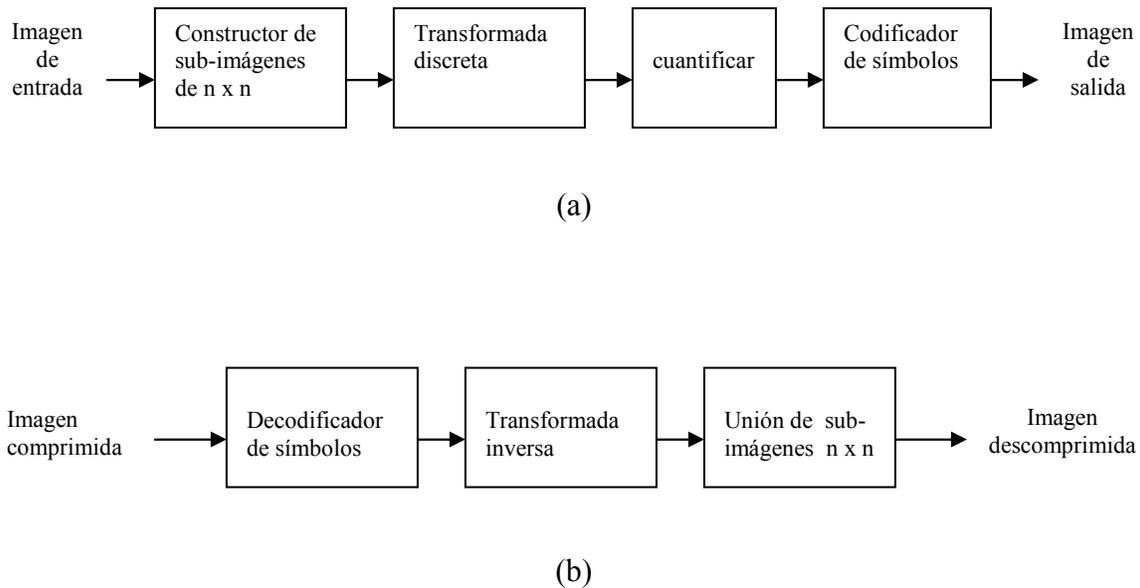


Figura 2.7 Sistema típico de codificación por transformación:
(a) codificador, (b) decodificador

2.8.3 Transformada en una dimensión.

En el caso general, una transformada se realiza mediante una multiplicación sucesiva de los elementos de un vector dato, por un conjunto de vectores “*base*”, cada uno de los cuales puede considerarse como un conjunto de factores de ponderación; la posterior suma de los valores obtenidos de esta forma, y la posible multiplicación por una constante, complementan las operaciones necesarias que dan lugar a los coeficientes transformados. Matemáticamente es útil representar la información de una imagen en forma matricial/vectorial como sigue:

$$C = [B] d \quad (2.15)$$

donde C y d son vectores columnas de $N \times 1$ que contienen los coeficientes y los datos de la transformada, respectivamente; mientras que $[B]$ es una “matriz base”. Para el caso de $N=4$, se tiene:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} \quad (2.16)$$

La implementación práctica de esta operación requiere, para el cálculo de N coeficientes transformados, N^2 multiplicaciones y $N(N-1)$ sumas, para un conjunto de datos numerosos, la carga computacional es elevada y son complejos, mucho mayor. Por esto, se han desarrollado diversos algoritmos de los denominados rápidos (“fast”), por ejemplo la transformada rápida de Fourier (FFT).

2.8.4 La transformada en dos dimensiones.

Como se ha tratado anteriormente, la transformada discreta básica consiste en la multiplicación término a término de los elementos de un vector dato, por uno de los elementos de un conjunto de vectores base y sumando los valores obtenidos.

Una transformada unidimensional solamente tendrá en cuenta la correlación entre los elementos individuales de un vector de dato, y no entre los vectores adyacentes. Por lo que, para el caso de dos dimensiones es de suponer que se tendrá que aplicar nuevamente la transformada unidimensional, la forma de realizar esto, es mediante la ecuación (2.17).

$$[C] = [B] [D] [B]^T \quad (2.17)$$

Donde $[D]$ es la matriz de datos, $[B]^T$ es la matriz transpuesta de la base. Con la ecuación (2.17), se consigue reducir considerablemente la correlación del conjunto de datos, tanto en la dirección vertical y como en la horizontal.

Bien, para la recuperación de los datos originales, puede derivarse de la ecuación (2.17), mediante el empleo de operaciones matriciales elementales, lo que produce la siguiente ecuación:

$$[D] = [B]^T [C] [B] \quad (2.18)$$

2.8.5 Representación de una imagen en bases.

Partiendo de la ecuación (2.18), es posible notar como están representados los datos en una base. Con la finalidad de facilitar las operaciones, se selecciona la matriz base de 2×2 , por lo que la ecuación (2.18) es ahora:

$$[D] = \begin{bmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

la matriz de coeficientes se puede expandir de la siguiente manera:

$$[C] = \begin{bmatrix} c_{11} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & c_{12} \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ c_{21} & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & c_{22} \end{bmatrix}$$

con la expansión de los coeficientes se realiza las operaciones correspondientes, por lo que los datos quedarán representados como:

$$[D] = c_{11} \begin{bmatrix} b_{11}b_{11} & b_{11}b_{12} \\ b_{11}b_{12} & b_{12}b_{12} \end{bmatrix} + c_{12} \begin{bmatrix} b_{11}b_{21} & b_{11}b_{22} \\ b_{12}b_{21} & b_{12}b_{22} \end{bmatrix} + c_{21} \begin{bmatrix} b_{21}b_{11} & b_{21}b_{12} \\ b_{22}b_{11} & b_{22}b_{12} \end{bmatrix} + c_{22} \begin{bmatrix} b_{21}b_{21} & b_{21}b_{22} \\ b_{22}b_{21} & b_{22}b_{22} \end{bmatrix}$$

Cada una de la matrices del lado derecho de cada coeficiente, representa una base. Por lo tanto, la imagen puede reconstruirse como una suma ponderada de imágenes bases. Por ejemplo, si la matriz B es:

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Figura 2.8 Matriz B, propuesta

las imágenes base son:

$$\begin{array}{cccc} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} & \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\ \text{(a)} & \text{(b)} & \text{(c)} & \text{(d)} \end{array}$$

Figura 2.9 Imágenes base de la matriz B de la figura 2.8

La figura 2.9(a), representa la media de los cuatro elementos de imagen (DC), la figura 2.9(b) el detalle que se compone de un cruce por cero horizontal, pero sin cambios en la dirección vertical, la figura 2.9(c) es la complementaria de 2.9(b), y la 2.9(d) un cruce por cero tanto en la dirección vertical como en la horizontal.

De manera general, para una imagen de $N \times N$, habrá N^2 imágenes base que juntas con los adecuados valores de ponderación, pueden sumarse para reconstruir la imagen.

Ya para concluir esta sección, se afirma que:

una imagen se puede expresar como una combinación lineal de imágenes “base”, siendo los coeficientes de su transformada los que ponderan el peso de cada imagen base en la imagen original.

2.8.6 Distribución de la energía en una transformada.

La distribución de la energía en el resultado de una transformación, de manera general, tiende a concentrarse en pocos coeficientes. Sin embargo, la energía total de los datos se conserva en el dominio transformado, como se podrá ver en el siguiente ejemplo, que por

simplicidad de cálculo se usa una matriz 4x4 de una transformada, que se usará como la matriz B. Esta matriz, junto con el vector de datos son presentados a continuación, haciendo uso de la ecuación (2.15) para una transformación unidimensional, donde la expresión que resulta es:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$$

y el vector de coeficiente resultante es:

$$C^T = [13 \quad -2.0 \quad 0.0 \quad -1.0]$$

La energía total en el dominio de los datos es: $5^2 + 6^2 + 7^2 + 8^2 = 174$ y en el dominio transformado $(13)^2 + (-2.0)^2 + (0)^2 + (-1.0)^2 = 174$. De este modo, los cuatro elementos de los datos, han sido transformado de forma que uno de ellos tiene un valor grande y los demás son relativamente pequeños.

Lo interesante es que, al concentrar la energía en pocos coeficientes, se pueden guardar (almacenar) los mas dominantes, logrando con ello la compresión de datos, como se vera en los sistemas desarrollados de los último capítulos. Desde luego la recuperación de los datos es de forma aproximada. Por ejemplo, si se elimina el último coeficiente, tendremos dos que son ceros y solo almacenaríamos los dos más dominantes y el vector dato reconstruido (aproximado) que resulta es:

$$\tilde{d} = [5.5 \quad 5.5 \quad 7.5 \quad 7.5]$$

Para ver el comportamiento de la energía en el caso bidimensional se usará la misma matriz B, pero los datos ahora estarán en una matriz , esta es:

$$[D] = \begin{bmatrix} 5 & 6 & 8 & 9 \\ 6 & 6 & 5 & 7 \\ 4 & 5 & 4 & 3 \\ 8 & 6 & 5 & 7 \end{bmatrix}$$

Para este caso, se hace uso de la ecuación (2.17). Al realizar la primera multiplicación de esta ecuación; es decir, $[B] [D]$, que es como si se aplicará el caso unidimensional, pero ahora sobre cada uno de los vectores de forma separada, da como resultado:

$$[C'] = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 5 & 6 & 8 & 9 \\ 6 & 6 & 5 & 7 \\ 4 & 5 & 4 & 3 \\ 8 & 6 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 11.5 & 11.5 & 11.0 & 13.0 \\ -0.5 & 0.5 & 2.0 & 3.0 \\ 1.5 & 0.5 & 2.0 & 3.0 \\ -2.5 & -0.5 & 1.0 & -1.0 \end{bmatrix}$$

donde $[C']$ denota el resultado de la primera etapa de la transformada bidimensional. Hay que observar que la energía total en cada dominio es la misma, es decir 592 . También se puede ver que cuatro de los coeficientes en el dominio transformado contienen el 93.7 % de la energía total.

Es evidente que el conjunto de coeficientes obtenidos, aun contiene una estructura notable en la dirección horizontal. Aplicando la siguiente parte de la ecuación (2.17), es decir,

$[C'] [B]^T$, se obtiene:

$$[C]= \begin{bmatrix} 23.5 & -0.5 & 1.0 & -1.0 \\ 2.5 & -2.5 & 0.0 & -1.0 \\ 3.5 & -1.5 & 1.0 & 0.0 \\ -1.5 & -1.5 & -2.0 & 0.0 \end{bmatrix}$$

Ahora, solo un único coeficiente es dominante, la transformada bidimensional ha reducido considerablemente la correlación del conjunto de datos en la dirección vertical y horizontal. Este solo coeficiente, contiene el 93.3% de la energía existente en el conjunto de datos original .

Como antes, la eliminación de los coeficientes mas pequeños (los que sean necesarios, para un error tolerable) en este dominio, permite la recuperación aproximada de los datos originales, al efectuar la eliminación, se guardan pocos coeficientes, logrando finalmente la *compresión de datos por eliminación de coeficientes*.

2.8.7 Selección de la transformada

Hay diversos sistemas de codificación por transformación basados en transformadas discretas bidimensionales, los cuales han sido estudiados ampliamente. La aplicación de una determinada transformación en una aplicación dada, depende de la cantidad de error

que puede tolerarse y de los recursos de cálculo disponibles. Mas adelante, en los capítulos siguientes, se analizara y seleccionará una determinada transformada, para la construcción de un compresor de imágenes fijas.

2.8.8 Criterios de fidelidad

Debido a que la información de interés puede perderse por la eliminación de datos, se requiere de métodos que nos permitan evaluar estas pérdidas. Esta valoración puede hacerse utilizando: 1) los criterios de fidelidad objetiva y 2) los criterios de fidelidad subjetiva.

El primero esta referido, a cuando es posible expresar la pérdidas como una función de la imagen de entrada y de salida comprimida y que posteriormente se descomprime.

Supongamos que la entrada de una imagen es representada por $f(x,y)$ y $\tilde{f}(x,y)$ representa la aproximación de $f(x,y)$ que resulta de la compresión y posterior descompresión de la imagen de entrada. Luego, el error total entre las dos imágenes con tamaño $M \times N$ es:

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\tilde{f}(x,y) - f(x,y) \right] \quad (2.19)$$

y el error cuadrático es:

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\tilde{f}(x, y) - f(x, y) \right]^2 \right]^{1/2} \quad (2.20)$$

otro criterio de fidelidad objetiva es la media cuadrática de la relación señal a ruido de la imagen de salida, ésta es:

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \tilde{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\tilde{f}(x, y) - f(x, y) \right]^2} \quad (2.21)$$

El valor cuadrático medio (rms) de la relación señal a ruido, representado por SNR_{rms} , es obtenido tomando la raíz cuadrada de la ecuación (2.13).

Otro criterio usado en la comunidad de compresión de imágenes es el valor PSNR(Peak Signal to Noise Ratio), la cual queda definida como:

$$PSNR = 10 \cdot \log_{10} \left(\frac{255^2}{e_{rms}^2} \right) \quad (2.22)$$

Los criterios de fidelidad objetiva ofrecen mecanismos sencillos y convenientes para la evaluación de la pérdida de información; pero, finalmente las imágenes descomprimidas son vistas por seres humanos. Por ello es apropiado las evaluaciones basadas en un criterio de *fidelidad subjetiva*, lo que se puede conseguir, mostrando una imagen descomprimida a un conjunto adecuado de observadores y luego promediando sus evaluaciones con la

finalidad de medir la calidad de la imagen descomprimida. La calidad puede ser determinada por los observadores, basándose en criterios preestablecidos, como por ejemplo: buena, regular, mala y pésima.