CAPITULO 2

ETAPAS DE COMPRESIÓN DE IMAGENES

2.1 Transformada Discreta Coseno.

La transformada discreta del coseno (DCT) también denominada transformada del coseno es la más ampliamente utilizada en compresión de imágenes. Esta transformada cuenta con una buena propiedad de compactación de energía.

La decorrelación de coeficientes es muy importante para compresión, ya que, el posterior tratamiento de cada coeficiente se puede realizar de forma independiente, sin pérdida de eficiencia de compresión. Otro aspecto importante de la DCT es la capacidad de cuantificar los coeficientes utilizando valores de cuantificación que se eligen de forma visual.

Esta transformada ha tenido una gran aceptación dentro el tratamiento digital de imagen, debido al hecho de que, para los datos de una imagen convencional, que tienen una alta correlación entre elementos.

La DCT está bastante relacionada con la DFT , con la diferencia de que es una transformada real, debido a que los vectores base se componen exclusivamente de funciones coseno muestreadas.

Además la DCT minimiza algunos de los problemas que surgen con la aplicación de la DFT a series de datos.

El proceso de descomponer un conjunto de muestras en un conjunto ponderado de funciones base cosinusoidales se denomina Transformada Discreta del Coseno Directa (DCT), y al método de reconstruir el conjunto de muestras a partir del conjunto ponderado de funciones base cosinusoidales se le conoce como Transformada Discreta del Coseno Inversa (IDCT). Si la secuencia muestreada es mayor de ocho muestras, puede dividirse en grupos de ocho muestras y la DCT puede calcularse independientemente para cada grupo. Debido a que las funciones base cosinusoidales siempre tienen el mismo conjunto de valores en cada uno de los puntos de muestreo discretos, solamente cambian los valores de los coeficientes.

La DCT provee coeficientes de frecuencias correspondientes a una señal discreta variable en el tiempo (luminancia y cromancia).

Dado que aplica la DFT en el sentido horizontal y vertical, los valores muestreados en el tiempo de cada pixel, tendrán coeficientes muy precisos de frecuencias vertical y horizontal.

En si misma, la DCT no representa una reducción en el *bit-rate* sino mas bien lo opuesto. Esto es debido al hecho que la transformada es equivalente a una multiplicación matemática resultando en coeficientes con una palabra más larga que los valores de los pixel. Típicamente una bloque de 8 bits por pixel puede resultar en un bloque de 11bits por pixel. La diferencia en que los coeficientes sí pueden ser comprimidos o aun eliminados con técnicas se explican mas adelante.

Estos coeficientes de frecuencia son los que se transmiten luego de la aplicación de algunas de las técnicas de compresión que se describirán a lo largo de la tesis.

El concepto de la DCT se basa sintéticamente en tomar cada píxel de un bloque de 8 por 8 pixel.

Ese pixel es una "Muestra" (*sample*) de una señal variable en el tiempo, proporcional a la luminancia y de otra señal variable en el tiempo, proporcional a la crominancia.

Estas dos señales son las que se pasarán, separadamente, al dominio de las frecuencias , tal como se hacía con la DFT , obteniéndose los coeficientes en frecuencia y transmitiéndolos , en lugar de transmitir las funciones del tiempo. Como la señal de entrada es discreta, la transformada debe ser también discreta, de ahí el nombre de DCT.

Ya que hemos realizado el DCT ahora para decodificación utilizamos la transformada inversa (IDCT) que se utiliza para la obtención de un píxel aplicando la antitransformada DCT a un bloque de coeficientes. La IDCT no es otra cosa que un proceso de mezcla de distintas configuraciones de píxel donde las respectivas amplitudes y polaridades (fases) están dados por los coeficientes.

2.1.1 Concepto Matemático de la DCT y de la IDCT

A la entrada del codificador, la imagen original de agrupa en bloques de 8 x 8, se traslada de enteros sin signo en el rango $[0, 2^p\text{-}1]$ a enteros con signo con rango $[-2^{p-1}, 2^{p-1}-1]$ y se le da como la entrada de la transformada DCT. En la salida del codificador, la transformada DCT inversa (IDCT) devuelve los bloques 8 x 8 para formar la imagen reconstruida.

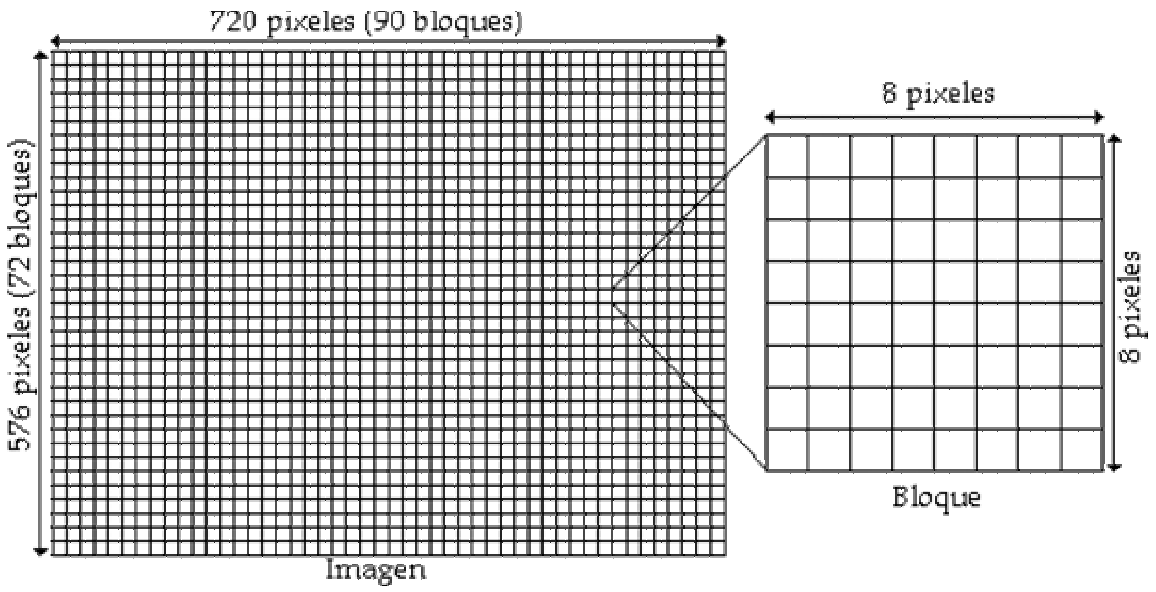

Figura 2.1 División en bloques o subimágenes de 8x8 píxeles.

Las siguientes ecuaciones son las definiciones matemáticas idealizadas de la DCT 8x8 (1.1) y de la IDCT 8x8 (1.2)

$$\mathrm{F}(u,v) = \frac{C(u)}{2}\frac{C(v)}{2}\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right] \quad (1.1)$$

$$f(x,y) = \sum_{u=0}^{7}\frac{C(u)}{2}\sum_{V=0}^{7}\frac{C(v)}{2}F(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \quad (1.2)$$

1 :F es el valor del coeficiente de la línea *"u"* y columna *"v"* (dominio de la frecuencia), recordemos que *u* y v varían de 0 a 7.

2 : Estas constantes valen 0.707 ($1/\sqrt{2}$) para cuando *"u"* y/o "*v"* valen 0 , y valen 1 en cualquier otro lugar del tablero.

3 : Sumatoria de todos los términos de *"x"* entre 0 y 7.

4 : Sumatoria de todos los términos de *"y"* entre 0 y 7.

5 : Valor del píxel en una dada columna y línea (dominio del tiempo).

6 : Generador de funciones vertical ,produce la forma de onda cosenoidal vertical: *x* es el numero de la columna de los pixels y *u* es la frecuencia. La unidad es en radianes

7 : Generador de funciones horizontal ,produce la forma de onda cosenoidal horizontal : "y" es el numero de la línea de los pixels y "v" es la frecuencia.

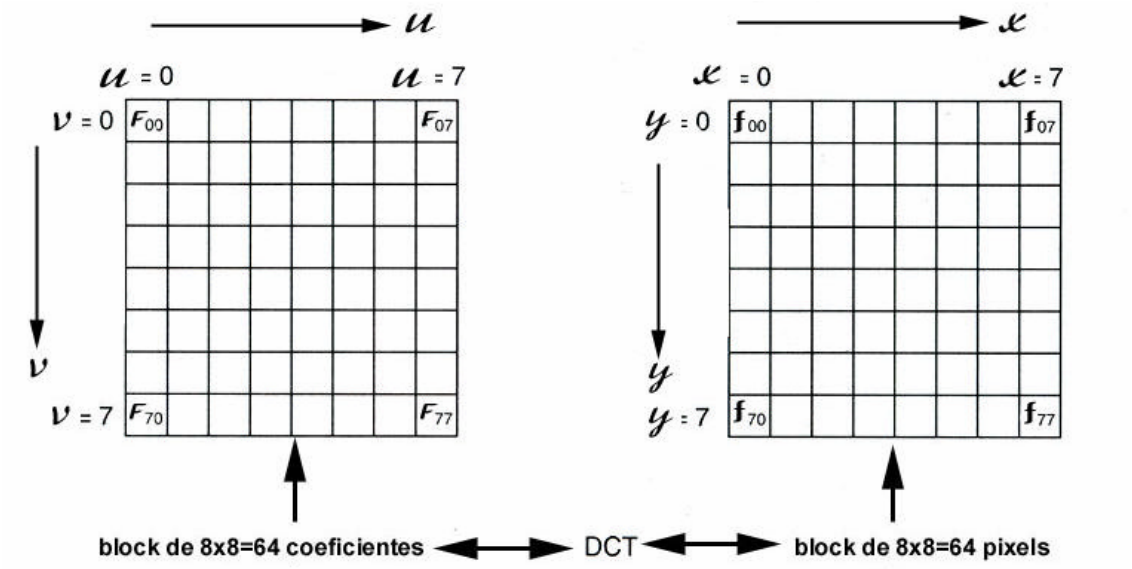

Figura 2.2 Representación Grafica.

Un block de pixel es transformado a un block de coeficientes de frecuencias

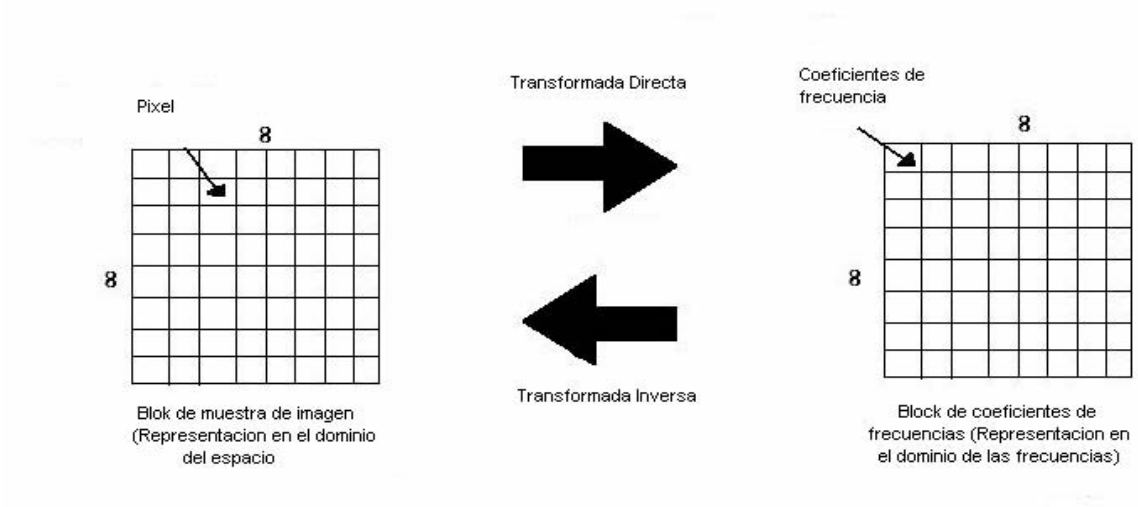

Figura 2.3 DCT y IDCT

La DCT toma cada bloque 8x8 y lo descompone en 64 señales de una base ortogonal. Cada una de estas señales contiene una de las 64 frecuencias espaciales únicas que comprenden el espectro de la señal de entrada. La salida de la DCT es un conjunto de 64 amplitudes de señal de la base o coeficientes cuyos valores son únicamente determinados por una señal de 64 puntos de entrada en particular. Al coeficiente con frecuencias ceros en las dos dimensiones se le llama coeficientes DC a los otros 63 coeficientes, coeficientes AC. Estos nombres vienen del uso habitual de la DCT en electrónica para analizar circuitos donde hay simultáneamente términos de corriente continua (DC) y de corriente alterna (AC).

Ya que los valores de la imagen varían lentamente entre pixel consecutivos, el proceso de la DCT realiza la compresión de los datos concentrados la mayor parte de la señal en las frecuencias espaciales bajas. Para un bloque típico 8 x 8 de una imagen típica, la mayoría de las frecuencias espaciales tienen una amplitud cero o casi cero y no serán codificadas.

En el decodificador la IDTC realiza el proceso inverso. Toma 64 coeficientes DCT (que para entonces ya han sido cuantificados) y reconstruye una imagen de 64 puntos de salida sumando las señales base. Si la DCT y la IDCT pudieran ser calculadas con precisión absoluta y los coeficientes de la DCT no se cuantificaran se podría reconstruir perfectamente la imagen original, simplemente la transforma a un dominio en el que puede ser codificada mas eficientemente.

Como resumen podemos decir que:

* Después de la DFT, la DCT es la transformada más ampliamente utilizada en el tratamiento digital de imagen.

* la DCT tiene unas características óptimas en cuanto a compactación de energía y se aproxima a la transformada más adecuada estadísticamente para la decorrelación de una señal.

* La DCT es una transformada real y ortogonal.

*Esta transformada puede implementarse a través de algoritmos rápidos, y puede estar basada en el correspondiente a la FFT.

* La aplicación más importante de la DCT es la compresión de imagen.

*Otras aplicaciones menos usuales de la DCT son el filtrado, la transmisión progresiva de imágenes y el reconocimiento de patrones.

2.2 Cuantificación.

Después de salir de la DCT, cada uno de los 64 coeficientes se cuantifica uniformemente en conjunción con una tabla de cuantificación de 64 elementos, que debe ser especificado por la aplicación o el usuario como entrada al codificador. Cada elemento de la tabla de cuantificación puede ser cualquier valor entre 1 y 255, que especifica el tamaño del paso del cuantificador para su coeficiente DCT correspondiente. El propósito de la cuantificación es conseguir aun mas compresión representando los coeficientes DCT sin mas precisión que la que es necesaria para conseguir la calidad de la imagen deseada.

Visto de otra forma, lo que se pretende con este proceso es descartar la información que no es visualmente relevante. La cuantificación es una aplicación muchos a uno y, por tanto, es fundamentalmente con perdidas de hecho es la principal fuente de perdidas en los codificadores basados en la DCT.

La cuantificación se define como una división de cada coeficiente DCT por su coeficiente de cuantificación, es decir, el tamaño del paso de la cuantificación correspondiente redondeado a entero mas cercano.

$$F^*(u,v) = \mathrm{Re}\,dondeo\left(\frac{F(u,v)}{Q(u,v)}\right) \qquad (1.3)$$

El valor de salida esta normalizado por el tamaño del paso de cuantificación. La decuantificacion es la función inversa que, en este caso, significa simplemente que la normalización se eliminan multiplicando por el tamaño del paso, lo que devuelve el resultado a la representación apropiada para la entrada de la IDCT

$$F^{'}(u,v) = F^{*}(u,v) * Q(u,v) \qquad (1.4)$$

Cuando se quiere comprimir una imagen tanto como sea posible sin artificios visibles hace falta escoger cada tamaño de paso de cuantificación como el umbral perceptual para la contribución de su función base coseno correspondiente. Estos umbrales son también funciones de las características de la imagen original, de las características de visualizador y de la distancia de visión.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Figura 2.4 Tabla de cuantificación propuesta por ISO

La tabla de la fig. 2.4 fue aprobada por los miembros de JPEG y apareció en el estándar ISO como informativa, no como un requerimiento para cuantificar la luminancia de una imagen.

La matriz de cuantificación es una matriz de 8x8 pixels formada por elementos denominados cuantos, esta matriz posee un elemento por cada coeficiente de la DCT. Esta matriz por término medio es siempre simétrica. Esta matriz tendrá la característica de que los elementos mas arriba y a la izquierda tendrán un menor valor, mientras que aquellos elementos que estén mas abajo a la izquierda tendrán un valor mas alto. El cuantizador divide los elementos de la DCT por el correspondiente valor del cuanto de la matriz de cuantización.

En esta fase de la codificación de JPEG es donde se va a producir en la imagen original las pérdidas de calidad típicas de JPEG y que estas serán aprovechadas para permitir grandes ratios de Compresión de las imágenes.

En el proceso de cuantización de la DCT, los coeficientes de alta frecuencia son divididos por un valor n>l y el resultado se redondea al entero más cercano. El valor de n varía con la posición del coeficiente en el bloque, con frecuencias más altas se asignan valores más altos. Como resultado, coeficientes que representan frecuencias espaciales bajas son cuantificados con pasos relativamente bajos y tienen una S/N alta, Los coeficientes que representan las frecuencias espaciales más altas son cuantificados con pasos grandes y padecen distorsión y S/N baja. El proceso de cuantificación es controlado por tablas específicas y al decodificador se le proporciona información acerca del modelo de cuantificación lo que permite la decodificación correcta.

Una vez que tenemos el vector de elementos de la DCT ordenado mediante el método de Zig-Zag, debemos aplicar la cuantización a cada uno de los 64 elementos que forman el vector que tenemos (valores de la DCT); la cuantización por tanto consistirá en dividir cada valor de este vector por un número correspondiente a una tabla de cuantización prediseñada que permitirá redondear cada elemento de la DCT al entero mas cercano.

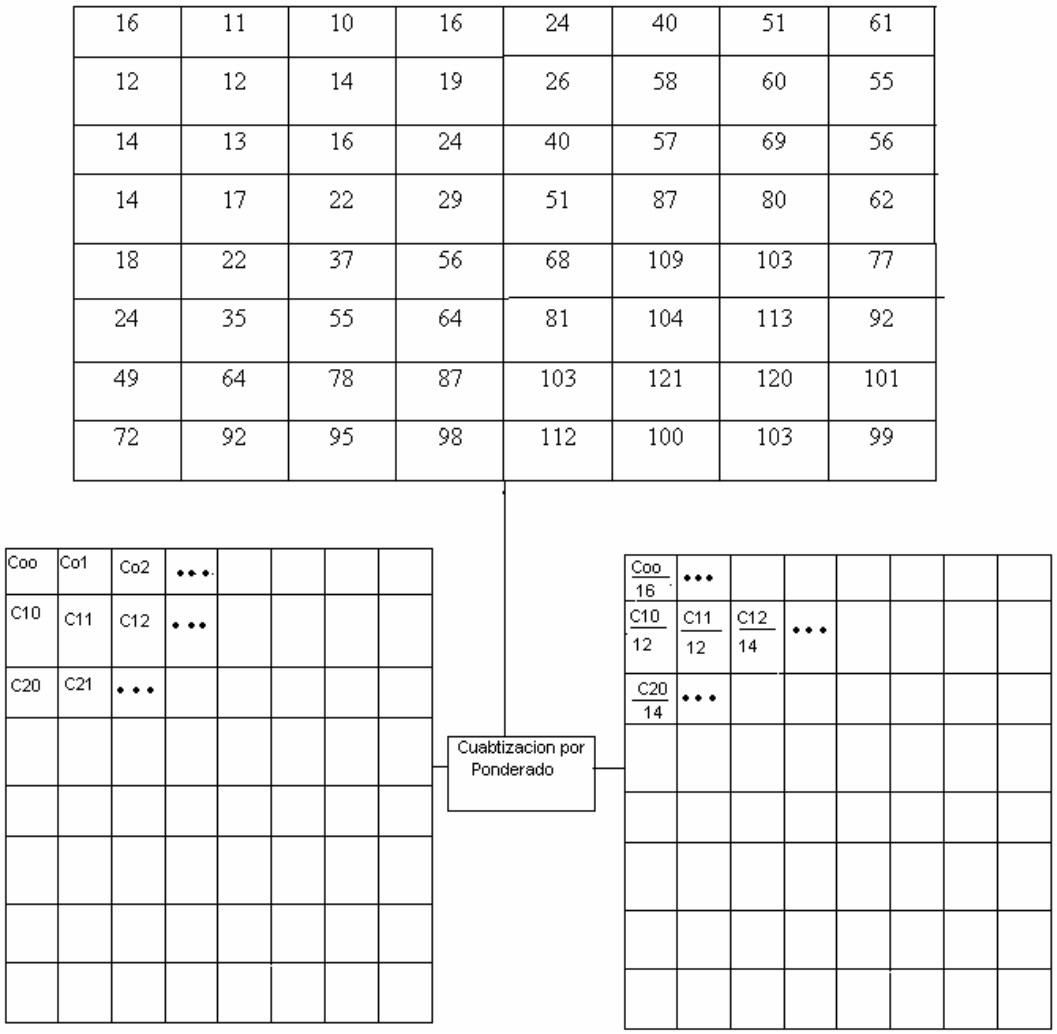

Figura 2.5 Tabla de cuantificación ya están normalizada en función de la respuesta del ojo humano

Tras las cuantificación, el coeficiente DC se trata de formar separadamente de los 63 coeficientes AC. El coeficientes DC es una medida del Valor medio de los 64 píxeles de la imagen. Ya que normalmente hay un fuerte correlación entre los coeficientes DC de dos bloques 8 x 8 adyacentes, los coeficientes DC se codifican como la diferencia entre el termino DC del bloque previo en el orden de codificación. Este tratamiento especial merece la pena que los coeficientes DC contienen una fracción significativa de la energía total de la imagen.

Finalmente, todos los coeficientes cuantificados se ordenan en una secuencia en zig-zag, como se muestra en la siguiente figura. Esta ordenación ayuda a facilitar la codificación por entropía colocando los coeficientes de las bajas frecuencias, que normalmente no son cero, antes de los coeficientes de las altas frecuencias.

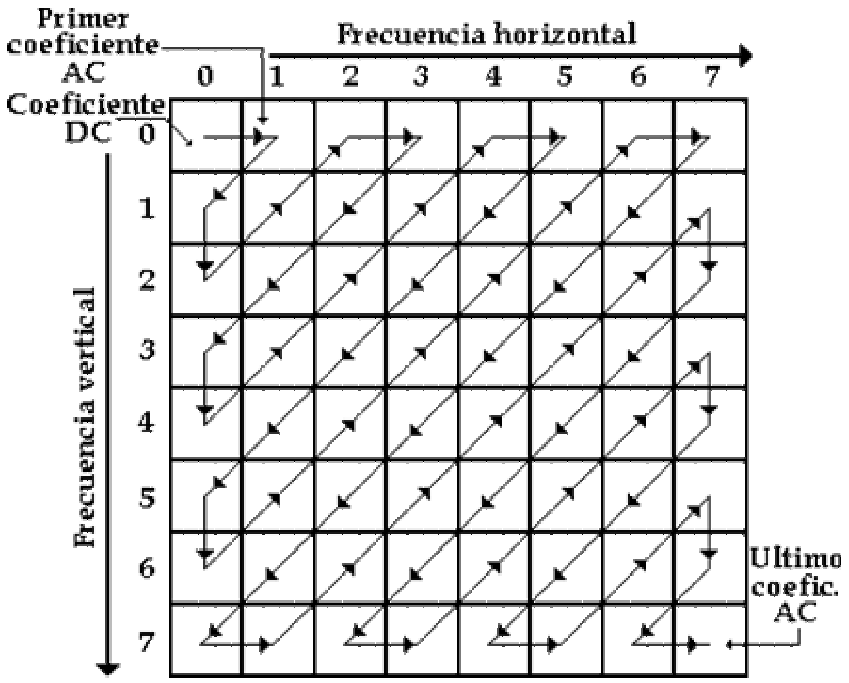

Figura 2.6 Barrido o exploración en zig-zag.

El coeficiente DC de cada bloque es codificado usando DPCM. Es decir, se codifica la diferencia entre coeficiente DC del presente bloque y el del bloque previamente codificado. Puesto que la cadena unidimensional reordenada según el barrido en zig-zag, está distribuida cualitativamente según una frecuencia espacial creciente, el procedimiento de codificación JPEG ha sido diseñado de modo que se beneficia de la existencia de largas series de ceros que se producen normalmente en la reordenación. En particular, los coeficientes AC no nulos se codifican utilizando un código de longitud variable que define el valor del coeficiente y el número de ceros precedentes.

La DCT de una imagen con barrido progresivo, resulta en coeficientes significativos localizados en el área superior izquierda del bloque. Leyendo los valores a la salida de la memoria en una diagonal en "zigzag" a 45° como se muestra en las Fig.2.6 resulta en que se envían primero los coeficientes que no son cero, seguidos de una larga serie de valores cero.

2.3 Codificadores de Entropía

El paso final de la codificación basada en la DCT es la codificación por entropía. Este paso consigue una compresión sin pérdidas adicional al codificar los coeficientes cuantificados de la DCT de una forma más compacta basándose en sus características estadísticas. La propuesta del JPEG especificaba dos métodos de codificación por entropía: Codificación de Huffman y codificación aritmética. El modelo base de codificador-decodificador usa códigos Huffman pero se especifican códigos con los dos métodos para todos los modos de operación.

Es útil considerar la codificación por entropía como un proceso de dos pasos. El primero convierte la secuencia de coeficientes cuantificados en zig-zag en una secuencia de símbolos intermedia mediante RLC. El segundo paso convierte los símbolos a una secuencia de datos en la que los símbolos no tienen ya fronteras externamente identificables. La forma y definición de los símbolos intermedios depende tanto del modo de operación como el método de codificación por entropía.

La codificación de Huffman requiere que la aplicación especifique uno o mas conjuntos de tablas de códigos Huffman. Estas deben ser las mismas para comprimir una imagen y para descomprimir. Las tablas de códigos Huffman pueden ser predefinidas por la aplicación o bien calculadas específicamente para una imagen.

El método de codificación aritmética propuesta por JPEG, por lo contrario, no requiere ninguna tabla externa por que es capaz de adaptarse a las características de la imagen conforme la va codificando. Este tipo de codificación produce entre 5% y 10% mas de compresión de los codigos Huffman para muchas de las imágenes de pruebas probadas por los miembros de JPEG pero es mas complejo de implementar que la codificación Huffman..

La DCT (al igual que las demás transformadas discretas) puede aplicarse a codificación de imagen, desde un punto de vista de reducción de ancho de banda o compresión de datos. El objetivo es conseguir que una imagen (dominio espacial) o secuencia de imágenes (dominios espacial-temporal), se traslade a un dominio transformado de tal forma que se reduzca el ancho de banda para la transmisión o los requerimientos para el almacenamiento; de tal forma que la subsiguiente recuperación de la imagen o secuencia de imágenes mediante la transformada inversa, no presente una distorsión perceptible.

La minimización de la distorsión es significante desde el punto de vista subjetivo, y a pesar de que se han definido muchas medidas cuantitativas (error cuadrático medio - MSE-, error absoluto medio - MAE-, coeficiente de correlación - CC-, etc..), la DCT cuenta con la ventaja de que sea el observador humano el último en juzgar la calidad de las imágenes procesadas. Mientras que los criterios cuantitativos y subjetivos (visuales) son los puntos de referencia para evaluar un algoritmo, son igualmente importantes otros factores tales como la complejidad de la implementación y las diversas características opcionales pertenecientes a cada aplicación específica.

2.4 Codificación Huffman

Codificación Huffman convierte los valores de brillo de los pixeles de la imagen original en nuevos códigos de longitud variable, basado en su frecuencia de ocurrencia en la imagen. De esta manera, a los valores de brillo que ocurren más frecuentemente se les asignan los códigos más cortos y a los valores de brillo que ocurren con menos frecuencia se les asignan los códigos más largos. El resultado es que la imagen comprimida requerirá de menos bits para describir la imagen original.

El esquema de compresión Huffman comienza mirando el histograma de brillo de una imagen. Con el histograma, la frecuencia de ocurrencia para cada brillo en la imagen está disponible. Ordenando los valores de brillo por sus frecuencias de ocurrencia, se obtiene una lista donde el primer valor se encuentra más a menudo en la imagen, y el último valor se encuentra menos a menudo en la imagen. Con esta lista, el codificador Huffman asigna nuevos códigos a cada valor de brillo. Los códigos asignados son de longitudes variables; los códigos más cortos son asignados a los primeros (más frecuentes) valores m de la lista y, eventualmente, los códigos más largos se asignan a los últimos (menos frecuentes) valores de la lista. Finalmente, la imagen comprimida es creada simplemente sustituyendo los nuevos códigos de valores de brillo de longitud variable por los códigos de valores de brillo originales de 1 byte. Por supuesto, la lista de códigos Huffman que acopla los valores de brillo originales a sus nuevos códigos Huffman variables se debe añadir a la imagen para el uso de la operación de descompresión Huffman, como se muestra en la siguiente figura.

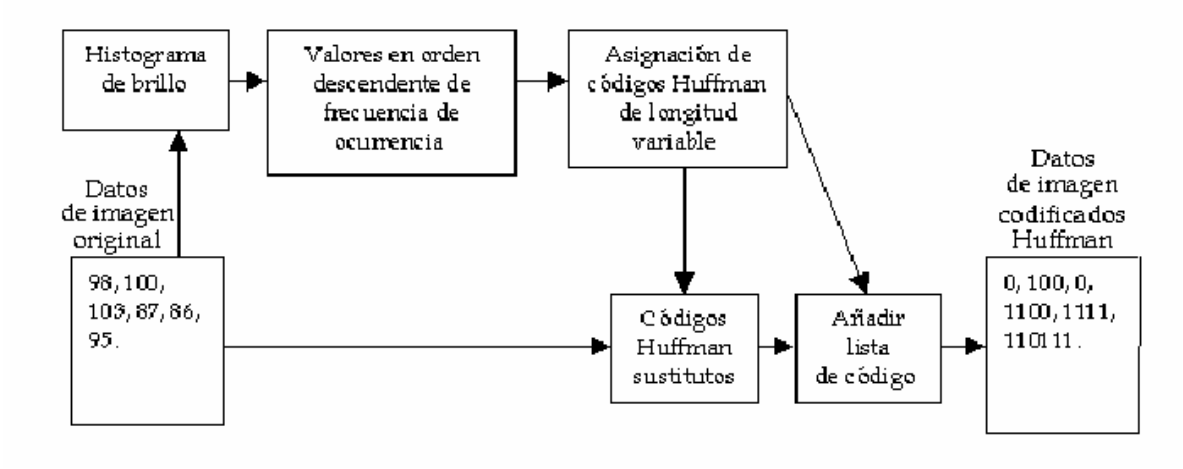

Figura 2.7 Diagrama de bloques de la codificación Huffman

Los códigos Huffman son asignados creando un árbol de Huffman que hace combinaciones con los valores de brillo basado en la suma de las frecuencias de ocurrencia. El árbol de Huffman asegura que los códigos más largos se asignen a los brillos menos frecuentes y los códigos más cortos se asignen a los brillos más frecuentes. Usando el brillo clasificado en orden de sus frecuencias de ocurrencia, los dos del final de la lista (menos frecuentes) se combinan y se etiquetan como 0 y 1. Los brillos combinados son representados por la suma de las frecuencias de ocurrencia. Entonces, se determinan y se combinan las próximas dos frecuencias de ocurrencia más bajas. De nuevo, el siguiente par se etiqueta 0 y 1, y es representado por la suma de las frecuencias de ocurrencia. Esto continúa hasta que todo el brillo se ha combinado. El resultado es un árbol que, cuando se sigue del final hasta el principio, indica el nuevo código Huffman binario para cada brillo en la imagen.

El algoritmo de Huffman se basa en la creación de un nuevo alfabeto y la traducción del alfabeto original , al nuevo. El nuevo alfabeto no tendrá longitud constante, (cada ocurrencia de este no ocupará un número constante de bits), y además en el algoritmo de Huffman, se asignan longitudes mas cortas a las ocurrencias con mayor probabilidad de aparición, y mas bits para las que menos se repiten, así de este modo se consigue la codificación y además la compresión.

Para la codificación Huffman, deberemos tener un determinado alfabeto de símbolos elegidos, en este caso en el que lo mostraremos con un ejemplo se se ha usado el siguiente alfabeto:

Alfabeto de símbolos {a,b,c,d,e,f,g};

Una vez que se ha elegido el conjunto de símbolos, se pasa a la obtención de la tabla de frecuencias de probabilidad de aparición de los símbolos, para ello, he asignado de manera fija una probabilidad de aparición de la siguiente forma:

Caracter	**Probabilidad**
a	**0.25**
b	**0.25**
c	**0.12**
d	**0.11**
e	**0.12**
f	**0.075**
g	**0.075**

Una vez que hemos fijado el conjunto de caracteres a usar y la probabilidad de aparición de cada uno de ellos, debemos construir el árbol de Huffman, para ello, debemos crear un árbol binario usando la tabla de Probabilidades de aparición.

Pondremos todos los caracteres como elementos terminales del árbol , con lo cual inicialmente no tendremos ningún nodo padre; ordenamos los caracteres de acuerdo con su probabilidad de aparición:

Caracter	Probablidad
a	0.25
b	0.25
c	0.12
e	0.12
d	0.11
f	0.075
g	0.075

Una vez ordenados todos los caracteres por su probabilidad, debemos realizar el siguiente proceso, se cogen aquellos dos caracteres que poseen la frecuencia más baja de todas las posibles, se suman sus probabilidades y se crea un nuevo nodo cuya frecuencia es la suma de los dos nodos anteriores, ahora volvemos a realizar la comprobación (se vuelven a elegir los dos nodos de menor frecuencia) y así sucesivamente hasta que únicamente nos quedemos con dos elementos.

Luego la tabla de codificación será la siguiente:

Caracter	Codigo Huffman
a	01
b	11
c	001
e	100
d	101
f	0001
g	0000

Pues teniendo esta tabla, ya podemos codificar todos aquellos caracteres que pertenezcan al lenguaje que hemos determinado anteriormente. Por ejemplo: Queremos codificar la siguiente palabra: { cgaef }, luego conociendo ya la tabla de codificación, únicamente, tendremos que sustituir cada carácter por su correspondiente código de Huffman; con lo cual se ha obtenido además de una buena codificación una buena compresión.

2.5 Error Cuadrático Medio

La cantidad de ruido se mide por la varianza del ruido, aunque es más interesante medirla con relación a la energía de la señal o la imagen.

$$e_{rms} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sqrt{(G(i,j)-I(i,j))^2}$$

$$PSNR_{db} = 10\log_{10}\frac{Media}{e}$$

e = error cuadrático medio

PSNR = error señal-ruido

Media = tamaño de la imagen