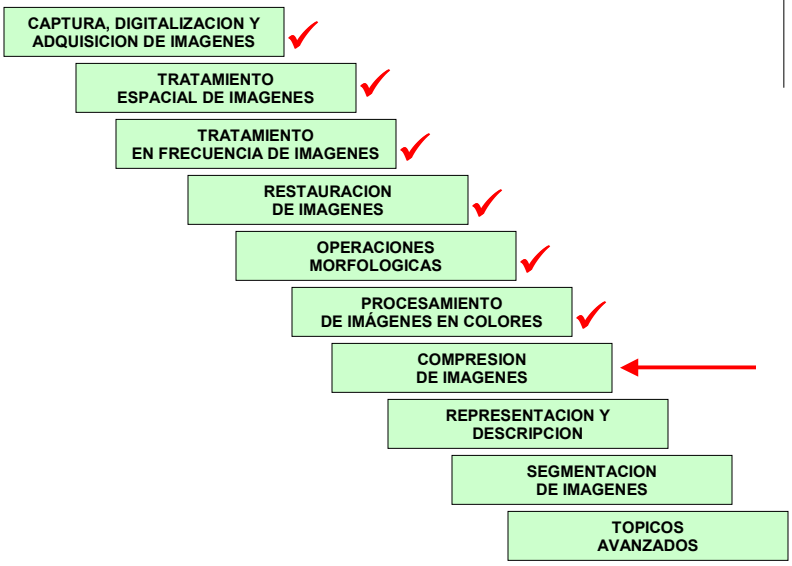
Procesamiento Digital de Imágenes

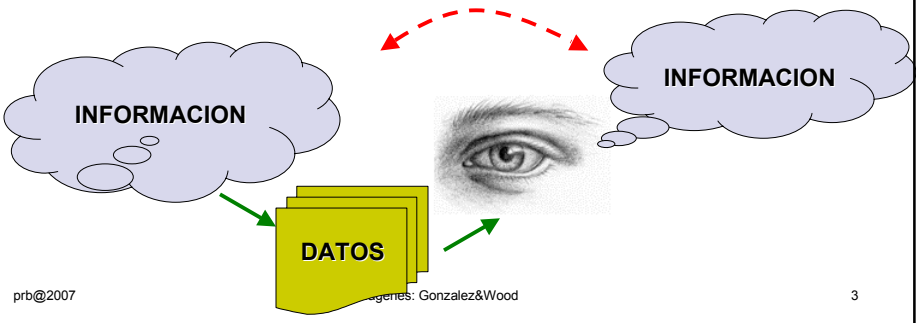
Pablo Roncagliolo B.
Nº 15

Orden de las clases...



Fundamentos de Compresión de Imágenes

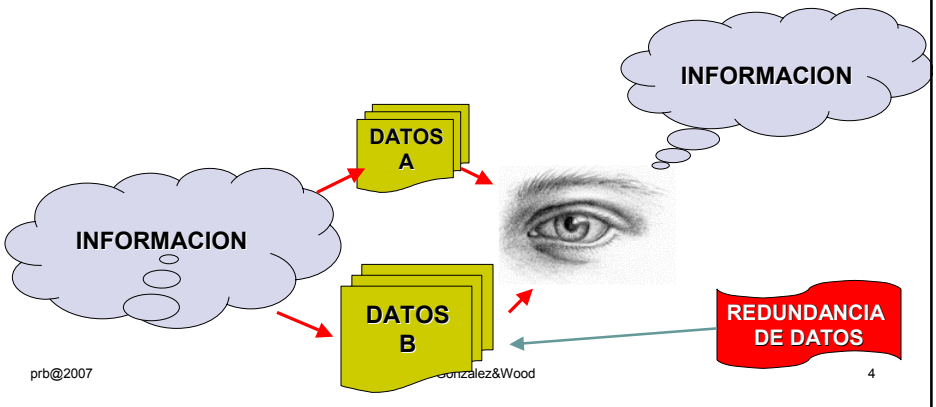
- Hay diferencia entre información y datos.
- En muchas ocasiones se utilizan como sinónimos y no lo son.
- Los datos son una forma representar la información.



Fundamentos de Compresión de Imágenes

Una misma información puede ser representada por distintas cantidades de datos.

Por tanto, algunas representaciones de la misma información contienen datos redundantes.



Fundamentos de Compresión de Imágenes

La compresión de datos se define como: *el proceso de reducir la cantidad de datos necesarios para representar eficazmente una información*, es decir, la eliminación de datos redundantes.

En el caso de las imágenes, existen tres maneras de reducir el número de datos redundantes:

- **eliminar código redundante**
- **eliminar píxeles redundantes**
- **eliminar redundancia visual**

Fundamentos de Compresión de Imágenes

Código redundante

El código de una imagen representa el cuerpo de la información mediante un conjunto de símbolos.

La eliminación del código redundante consiste en utilizar el menor número de símbolos para representar la información.

Las técnicas de compresión por codificación de Huffman y codificación aritmética utilizan cálculos estadísticos para lograr eliminar este tipo de redundancia y reducir la ocupación original de los datos.

Fundamentos de Compresión de Imágenes

Píxeles redundante

La mayoría de las imágenes presentan semejanzas o correlaciones entre sus píxeles.

Estas correlaciones se deben a la existencia de estructuras similares en las imágenes, puesto que no son completamente aleatorias.

De esta manera, el valor de un píxel puede emplearse para predecir el de sus vecinos.

Las técnicas de compresión Lempel-Ziv implementan algoritmos basados en sustituciones para lograr la eliminación de esta redundancia.

Fundamentos de Compresión de Imágenes

Redundancia Visual

El ojo humano responde con diferente sensibilidad a la información visual que recibe.

La información a la que es menos sensible se puede descartar sin afectar a la percepción de la imagen.

Se suprime así lo que se conoce como redundancia visual

La eliminación de la redundancia esta relacionada con la cuantización de la información, lo que conlleva una pérdida de información irreversible. Técnicas de compresión como JPEG, EZW o SPIHT hacen uso de variaciones en la cuantización.

Fundamentos de Compresión de Imágenes

Clasificación de los métodos de compresión:

Compresión sin pérdida de información (*Lossless*)
Compresión con pérdida de información (*Lossy*).

No confundir el concepto de pérdida de "datos" o "paquetes" en redes. Por . Ej.:

Se puede transmitir una imagen utilizando compresión sin pérdida de información sobre un protocolo de transmisión con pérdida, como UDP.

Por el contrario se puede transmitir una imagen comprimida con pérdida de información sobre un protocolo sin pérdida de datos como TCP.

Fundamentos de Compresión de Imágenes

Compresión sin pérdida de información:

Los métodos de compresión sin pérdida de información se caracterizan porque la tasa de compresión que proporcionan está limitada por la entropía (magnitud de la información) de la señal original.

Entre estas técnicas destacan las que emplean métodos estadísticos, basados en la teoría de Shannon, que permite la compresión sin pérdida. Por ejemplo: codificación de Huffman, codificación aritmética y Lempel-Ziv.

Son métodos idóneos para la compresión dura de archivos.

Fundamentos de Compresión de Imágenes

Compresión con pérdida de información:

Los métodos de compresión con pérdida de información logran alcanzar unas tasas de compresión más elevadas a costa de sufrir una pérdida de información sobre la imagen original.

Por ejemplo: JPEG, compresión fractal, EZW, SPIHT, etc. Para la compresión de imágenes se emplean métodos con pérdida, ya que se busca alcanzar una tasa de compresión considerable, pero que se adapte a la calidad deseada que la aplicación exige.

Compresión de Imágenes

Compresión sin Pérdida

Se distingue entre:

- **no adaptativos,**
- **semiadaptativos,**
- **sistemas adaptativos,**

según tengan en cuenta o no las características del archivo a comprimir.

Compresión de Imágenes

Compresión sin Pérdida

Los no adaptativos (código Huffman) establecen a priori una tabla de códigos con las combinaciones de bits que más se repiten estadísticamente.

A estas secuencias se asignan códigos cortos, y a otras menos probables claves más largas.

El problema que presentan es que un diccionario de claves único tiene resultados muy diferentes en distintos originales.

Compresión de Imágenes

Compresión sin Pérdida: Huffman

Un código de tipo Huffman se puede aplicar de modo semiadaptativo, si se analiza primero la cadena de datos a comprimir y se crea una tabla a medida.

Se logra mayor compresión, pero introduce dos inconvenientes:

- la pérdida de velocidad al tener que leer el original dos veces

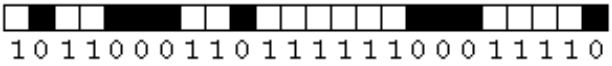
- la necesidad de anexar en el archivo, el índice de claves

Compresión de Imágenes

Compresión sin Pérdida: Huffman

Los compresores de uso general más populares utilizan métodos como éste, por eso tardan más en empaquetar los datos que en descomprimirlos.

El número de entradas de la tabla puede ser configurable.



A=10 C=000 B=11 D=111	ABCB0DDCDA	9 simb. + 1 bit = 18+1 = 19 bits
A=101 C=011 E=110 B=100 D=111 F=000	ABCCDBCE	8 simb. = 3*8 = 24 bits

Compresión de Imágenes

Compresión sin Pérdida: Huffman

El código del ejemplo es de longitud variable, pero no se requiere usar ningún tipo de separador entre los valores.

La razón es que siempre puede reconocer el final de una palabra porque ninguna otra palabra es el principio de otra dada.

Un código con esta propiedad se denomina código prefijo. El código Huffman es el código prefijo que requiere el mínimo número medio de bits por símbolo.

Compresión de Imágenes

Compresión sin Pérdida: Huffman

- Contar cuantas veces aparece cada carácter.
- Crear una lista enlazada con la información de caracteres y frecuencias.
- Ordenar la lista de menor a mayor en función de la frecuencia.
- Convertir cada elemento de la lista en un árbol.
- Fusionar todos estos árboles en uno único, para hacerlo se sigue el siguiente proceso, mientras la lista de árboles contenga más de un elemento:
  - Con los dos primeros árboles formar un nuevo árbol, cada uno de los árboles originales en una rama.
  - Sumar las frecuencias de cada rama en el nuevo elemento árbol.
  - Insertar el nuevo árbol en el lugar adecuado de la lista según la suma de frecuencias obtenida.
- Para asignar el nuevo código binario de cada carácter sólo hay que seguir el camino adecuado a través del árbol. Si se toma una rama cero, se añade un cero al código, si se toma una rama uno, se añade un uno.
- (Cero a la izquierda y Uno a la derecha)

Compresión de Imágenes

Compresión sin Pérdida: Huffman. Ej.

- "PREGUNTA DE PRUEBA"

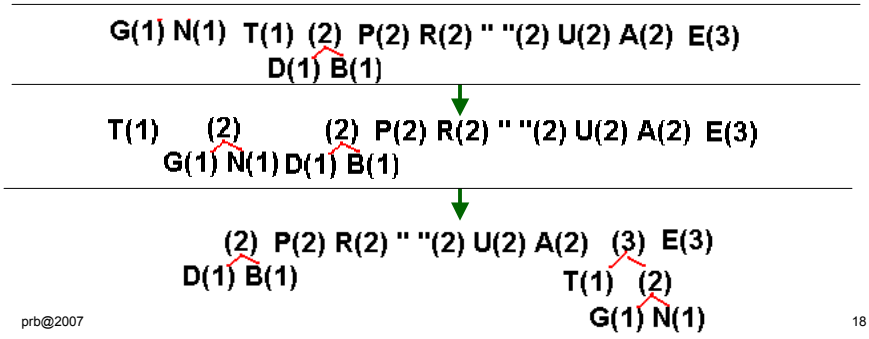
- P(2) R(2) E(3) G(1) U(2) N(1) T(1) A(2) " "(2) D(1) B(1)

Menor a mayor:

- D(1) B(1) G(1) N(1) T(1) P(2) R(2) " "(2) U(2) A(2) E(3)

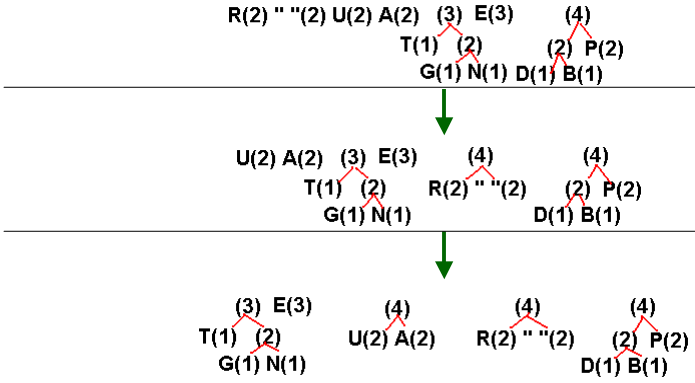
Árbol:

- D(1)→B(1)→G(1) →N(1)→T(1)→P(2)→R(2)→" "(2)→U(2)→A(2) →E(3)



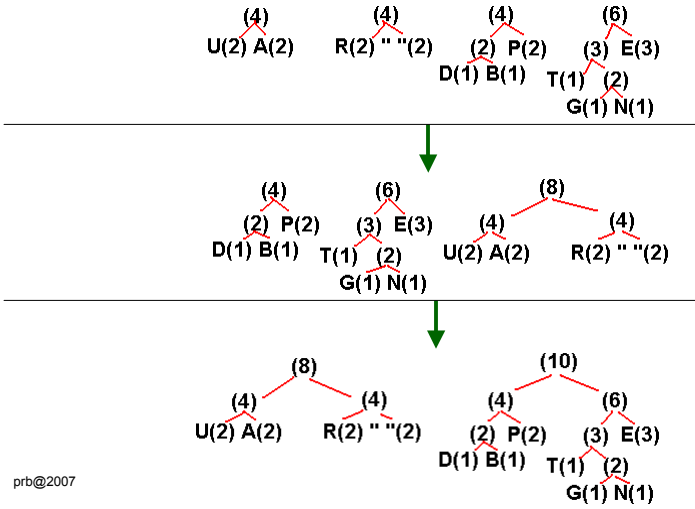
Compresión de Imágenes

Compresión sin Pérdida: Huffman. Ej.



Compresión de Imágenes

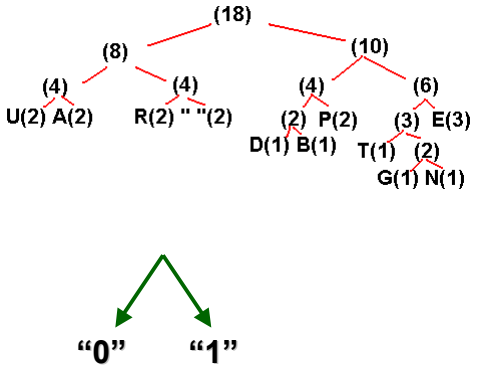
Compresión sin Pérdida: Huffman. Ej.



Compresión de Imágenes

Compresión sin Pérdida: Huffman. Ej.

U = 000
A = 001
R = 010
" "= 011
B = 1001
D = 1000
P = 101
E = 111
T = 1100
G = 11010
N = 11011

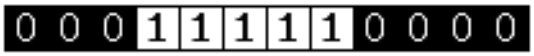


"0" "1"

Compresión de Imágenes

Compresión sin Pérdida: RLE

Entre los métodos adaptativos, el más simple es el RLE (Run Length Encode), que consiste en sustituir series de valores repetidos por una clave con indicador numérico.

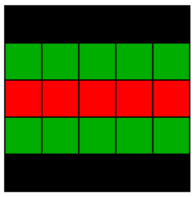


3*0 5*1 4*0

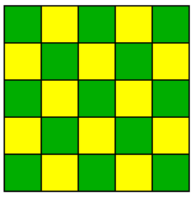
Compresión de Imágenes

Compresión sin Pérdida: RLE

Su eficacia depende del tipo de imagen. Los dos ejemplos siguientes tienen 25 valores, pero mientras que el primero se queda en 10 datos, el segundo no reduce su tamaño, sino que lo duplica.



5A5V5R5V5A



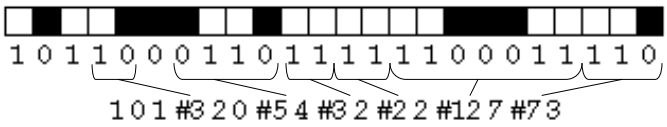
1V1N1V1N1V1N1V1N1V
1V1N1V1N1V1N1V1N1V
1V1N1V1N1V1N1V1N1V

La anotación de píxeles por series es adecuada en imágenes con zonas amplias de colores uniformes, pero no en otras con cambios frecuentes de valor o de texturas

Compresión de Imágenes

Compresión sin Pérdida: LZ

El sistema adaptativo LZ (Lempel-Ziv), del que deriva el LZW (Lempel-Ziv-Welch), es más ingenioso y consigue, en una lectura única, codificar repeticiones sin crear una tabla de códigos.

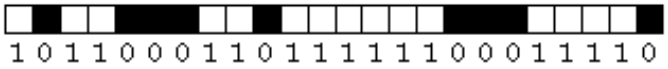


Cuando se localiza una secuencia similar a otra anterior, se sustituye por una clave de dos valores: los correspondientes a cuántos pasos se retrocede y cuántos datos se repiten.

Compresión de Imágenes

Compresión sin Pérdida: LZ

Rápido y fiable, se utiliza en formatos universales como el GIF o el TIFF. Aunque no logra relaciones de compresión muy altas, normalmente ahorra un tercio del archivo.



1 0 1 #3 2 0 #5 4 #3 2 #2 2 #12 7 #7 3

 # 3 2 significa retroceder tres píxeles y repetir dos
 # 12 7 significa retroceder 12 píxeles y repetir siete

Compresión de Imágenes

Cuantización Vectorial

Consiste en subdividir la imagen en pequeños bloques.

En términos ideales cada bloque posee un índice, el cual se transmite. Si existen varios bloques iguales, se logran tasas de compresión.

Se debe transmitir también la tabla de bloques (codebook)

Para compresión con pérdida se elige un subconjunto de bloques que sea representativo.

Compresión de Imágenes

Cuantización Vectorial

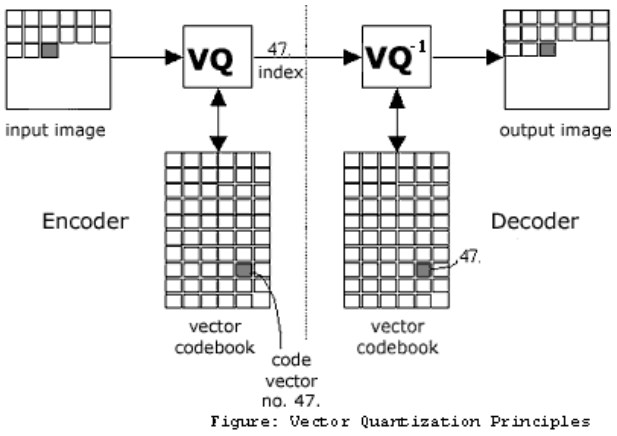
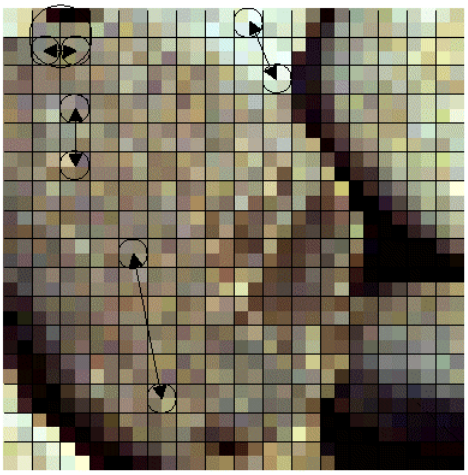


Figure: Vector Quantization Principles

Compresión de Imágenes

Cuantización Vectorial



Compresión de Imágenes

Cuantización Vectorial

Por Ej.: para una imagen de 512x512

Con bloques de 4x4 ➔ 16384 bloques

Algoritmo:

1. **Eliminar los bloques repetidos**
2. **Definir un umbral de diferencia mínima y eliminar así los bloques similares.**

¿Cómo de terminar la tabla de bloques adecuada?

➔ cada bloque es un vector ➔ clustering

Compresión de Imágenes

Compresión JPEG

El JPEG (Joint Photographic Experts Group) es el método de compresión más utilizado actualmente para la compresión de imágenes con pérdida.

Este método utiliza la transformada discreta del coseno (DCT), que se calcula empleando números enteros (mayor velocidad de cómputo).

El JPEG consigue una compresión "ajustable".

Compresión de Imágenes

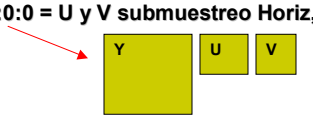

Esquema del modelo RGB


Esquema del modelo YUV

ETAPAS Compresión JPEG

1. **Transformación de espacio de color.**
   Se convierte la imagen RGB a YUV (similar a PAL Y NTSC)
   Y:luminancia, U y V: información de croma

2. **Submuestreo (opcional)**
   El visión del humano tiene menor sensibilidad al croma que a la luminancia, por lo tanto se puede muestrear con menor frecuencia los canales de croma.
   4:4:4 = los tres canales sin submuestreo
   4:2:2 = los canales U y V submuestreo horiz. (cada 2 píxeles)
   4:2:0 = U submuestreo horiz. , y V subm. Horiz y vert. cada 2
   4:0:0 = U y V submuestreo Horiz, y Vert. cada 2 (1/4 tamaño!)



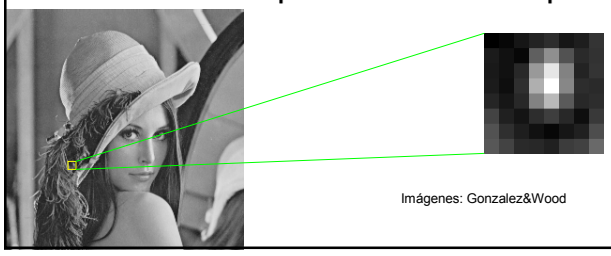
Compresión de Imágenes

ETAPAS Compresión JPEG

3. **Subdivisión imagen en bloques.**
   La imagen de entrada es dividida en bloques de NxN píxeles. El tamaño del bloque se escoge considerando los requisitos de compresión y la calidad de la imagen.

   En general, a medida que el tamaño del bloque es mayor, la relación de compresión también resulta mayor. Esto se debe a que se utilizan más píxeles para eliminar las redundancias. Pero al aumentar demasiado el tamaño del bloque la suposición de que las características de la imagen se conservan constantes no se cumple, y ocurren algunas degradaciones de la imagen, como bordes sin definir.

   Un tamaño del bloque conveniente es de 8x8 píxeles.



Compresión de Imágenes

ETAPAS Compresión JPEG

4. Transformada discreta de coseno (DCT).

Se procesa cada bloque de 8x8 de manera independiente. A cada bloque se le resta 128 para obtener valores entre –128 y 127 (byte) y luego se aplica la DCT.

Un ejemplo de uno de esos pequeños bloques de 8×8 inicial es este:

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

El siguiente proceso es restarles 128 para que queden números entorno al 0, entre -128 y 127.

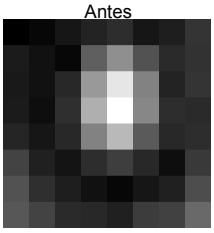
$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Se procede a la transformación por DCT de la matriz, y el redondeo de cada elemento al número entero más cercano.

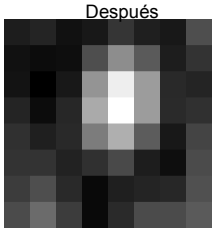
$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

Nótese que el elemento más grande de toda la matriz aparece en la esquina superior izquierda, este es el coecifiente DC.

Antes



Después



Compresión de Imágenes

ETAPAS Compresión JPEG

5. Cuantización (o cuantificación):

Los coeficientes de la transformada son cuantificados en base a un nivel de umbral para obtener el mayor número de ceros posibles.

DCT

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

Ej. Matriz de Cuantificación

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

DCT cuantificada

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Ejemplo para coeficiente DC: $\text{round}\left(\frac{-415}{16}\right) = \text{round}(-25.9375) = -26$

Compresión de Imágenes

ETAPAS Compresión JPEG

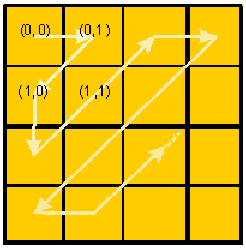
5. Cuantización (o cuantificación):

Para la cuantización se utiliza una matriz de normalización estándar, y se redondean los resultados a números enteros.

El ojo humano es muy bueno detectando pequeños cambios de brillo en áreas relativamente grandes, pero no cuando el brillo cambia rápidamente en pequeñas áreas (variación de alta frecuencia), esto permite eliminar las altas frecuencias, sin perder excesiva calidad visual.

Este es el proceso donde se produce la pérdida de información.

El paso siguiente consiste en reordenar en zig-zag la matriz de coeficientes cuantizados.



Compresión de Imágenes

ETAPAS Compresión JPEG

6. Codificación

Codificando con longitud variable los coeficientes, la imagen se puede comprimir aún más.

El codificador más utilizado es el algoritmo de Huffman, que se encarga de transmitir los coeficientes ordenados. Una razón para utilizar el codificador de Huffman es que es fácil de implementar.

Para comprimir los símbolos de los datos, el codificador de Huffman crea códigos más cortos para símbolos que se repiten frecuentemente y códigos mas largos para símbolos que ocurren con menor frecuencia.

Compresión de Imágenes

Transformada de Coseno

Es un caso particular de la Transformada de Fourier.

Utiliza solo la componente real de la transformada.

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)\pi u}{2N}\right]\cos\left[\frac{(2y+1)\pi v}{2N}\right]$$

$$\alpha() = \begin{cases} \sqrt{\frac{1}{N}} & para \;\; u,v = 0 \\ \sqrt{\frac{2}{N}} & para \;\; u,v = 1,2,...N-1 \end{cases}$$

Compresión de Imágenes

Transformada de Coseno

La transformada de Coseno Inversa:

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2x+1)\pi u}{2N}\right]\cos\left[\frac{(2y+1)\pi v}{2N}\right]$$

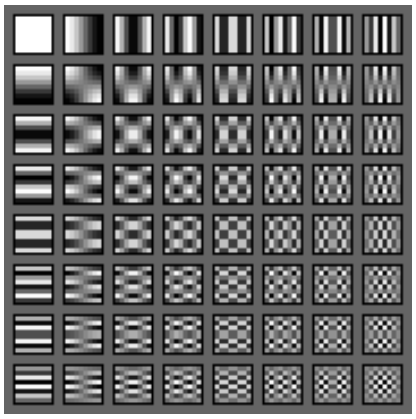
$$\alpha() = \begin{cases} \sqrt{\frac{1}{N}} & para \;\; u,v = 0 \\ \sqrt{\frac{2}{N}} & para \;\; u,v = 1,2,...N-1 \end{cases}$$

Compresión de Imágenes

Transformada de Coseno

La transformada de coseno se puede interpretar como la proyección de la imagen sobre cada una de las imágenes base. Los coeficientes representan la presencia de dichas componentes o imágenes base.

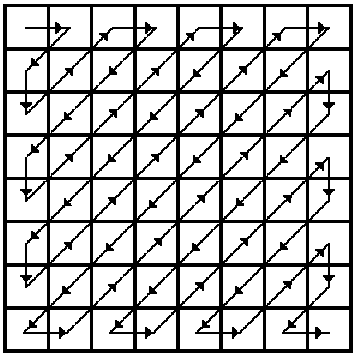
Para una imagen de 8x8 las imágenes base corresponden a las siguientes:



Compresión de Imágenes

Transformada de Coseno

Los coeficientes representan las frecuencias en la imagen. Su distribución desde bajas a altas frecuencias es la siguiente:



Utilizando un muestreo de este "recorrido" es una posible forma de compresión.

Compresión de Imágenes

Transformada de Coseno

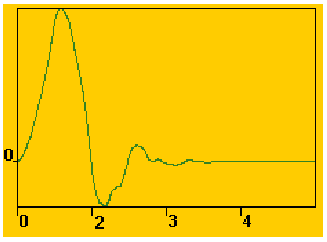
Algoritmo básico:

```
N=32;
C=zeros(N);
for u=1:N
   for v=1:N
     if (u==1) & (v==1)
       for x=1:N
         for y=1:N
            C(v,u)=C(v,u)+
     (1/N)*A(y,x)*cos((2*(x-1)+1)*pi*(u-1)/(2*N))*cos((2*(y-1)+1)*pi*(v-1)/(2*N));
         end;
       end;
     else
        for x=1:N
         for y=1:N
            C(v,u)=C(v,u)+
   (2/N)*A(y,x)*cos( (2*(x-1)+1)*pi*(u-1)/(2*N))*cos( (2*(y-1)+1)*pi*(v-1)/(2*N));
         end;
       end;
     end;
   end;
end;
```


Compresión de Imágenes

Wavelets

Las wavelets son funciones definidas sobre un intervalo finito y con valor medio cero.



La idea básica de la transformada wavelet es representar arbitrariamente una función como superposición de un conjunto de wavelets o funciones básicas.

Estas wavelets se obtienen a partir de una wavelet prototipo denominada wavelet madre, mediante dilataciones, escalados y traslaciones.

Compresión de Imágenes

Wavelets

La transformada wavelet discreta (DWT) se emplea para obtener una nueva representación de la imagen, más apropiada para el proceso de compresión.

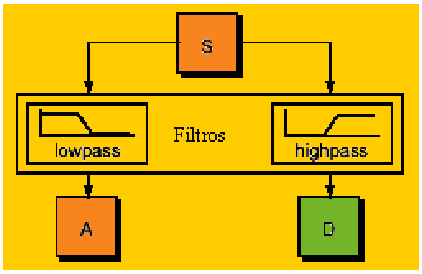
Para muchas imágenes la información más importante se encuentra en las frecuencias bajas, mientras que en las altas frecuencias se encuentran los detalles o matices de la señal.

El análisis wavelet permite descomponer la señal en aproximaciones y detalles, a éste proceso se le conoce con el nombre de *análisis*. Este filtrado nos proporciona el doble de datos de los que son necesarios, este problema se soluciona con la operación de downsampling.

Compresión de Imágenes

Wavelets

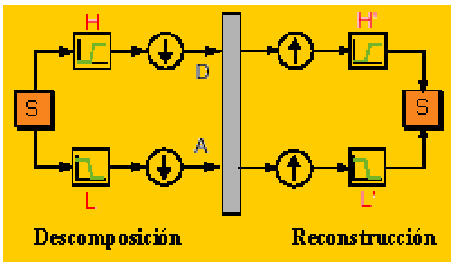
Proceso de descomposición (análisis).



Compresión de Imágenes

Wavelets

El proceso de *reconstrucción*, también denominado *síntesis*, se encarga de la obtención de la señal a partir de los detalles y aproximaciones. Éste proceso se lleva a cabo con la transformada wavelet discreta inversa.



La elección de los filtros (wavelets) influye notablemente en los resultados finales.

Compresión de Imágenes

Transformada Wavelets Discreta

La DWT aplicada a imágenes proporciona una matriz de coeficientes, conocidos como coeficientes wavelet.

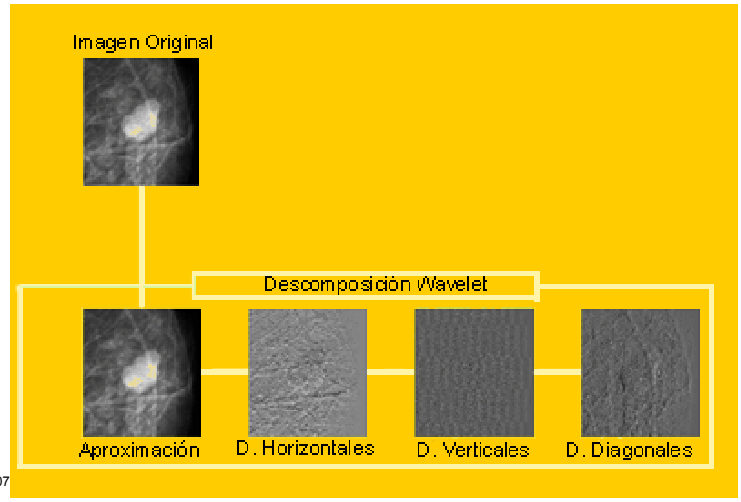
Se obtienen cuatro tipos de coeficientes:

- **aproximaciones**
- **detalles horizontales**
- **detalles verticales**
- **detalles diagonales**

La *aproximación* contiene la mayor parte de la energía de la imagen, es decir, la información más importante, mientras que los detalles tienen valores próximos a cero.

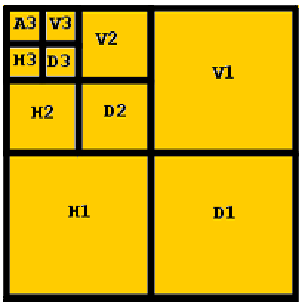
Compresión de Imágenes

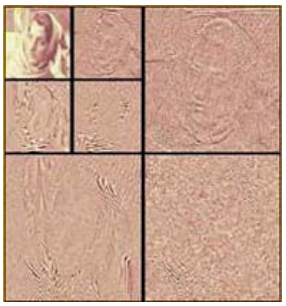
Transformada Wavelets Discreta



Compresión de Imágenes

Organización de los Coeficientes:





Compresión de Imágenes

Transformada Wavelets

Es una de las transformadas más utilizadas en procesamiento de imágenes. El algoritmo JPEG2000 se basa en ella..

Figura 2: imágenes de 512x512 píxeles. a) JPG b) JPEG2000. Ambas imágenes pesan 2,7Kb, sin embargo la imagen b) presentan distorsiones suaves, que son más naturales.

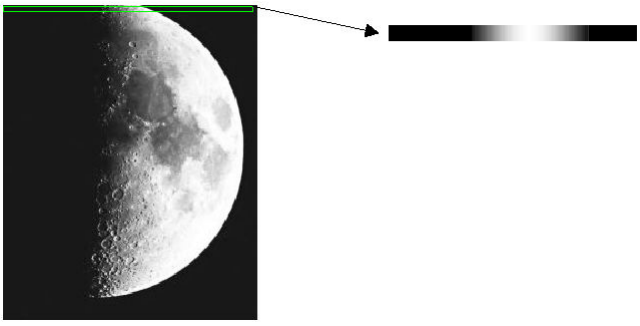
Compresión de Imágenes

Transformada Wavelets

Una explicación muy simple de wavelets en imágenes...

Supongamos la primera línea de la siguiente imagen:

0	0	20	40	120	200	240	240	240	240	200	120	40	20	0	0

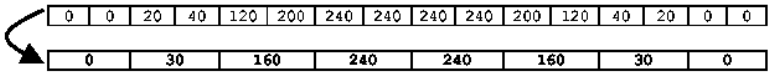


Compresión de Imágenes

Transformada Wavelets

Filtro Pasa Bajos (PB):

Se filtra las "bajas frecuencias" de la imagen: esto se puede lograr promediando los datos cada dos puntos adyacentes, logrando el siguiente vector:

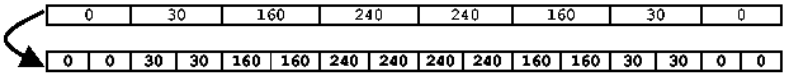


Compresión de Imágenes

Transformada Wavelets

Ajuste de resolución...

Para visualizar la imagen, en el tamaño original, se puede hacer un proceso simple, repetir los números adyacentes, obteniendo el siguiente vector:



Compresión de Imágenes

Transformada Wavelets

Filtro Pasa Alto:

Para reconstruir la imagen original es necesario "guardar" la información de detalles. Esto se puede lograr con otro vector, que contenga los valores específicos tal que sumando y restando al vector de baja frecuencia, logren restituir el vector original. Este vector de "detalles" se observa en la tabla siguiente:

	0	0	30	30	160	160	240	240	240	240	160	160	30	30	0	0
±	0		-10		-40		0		0		40		10		0	

0	0	20	40	120	200	240	240	240	240	200	120	40	20	0	0

Compresión de Imágenes

Transformada Wavelets

Proceso recursivo:

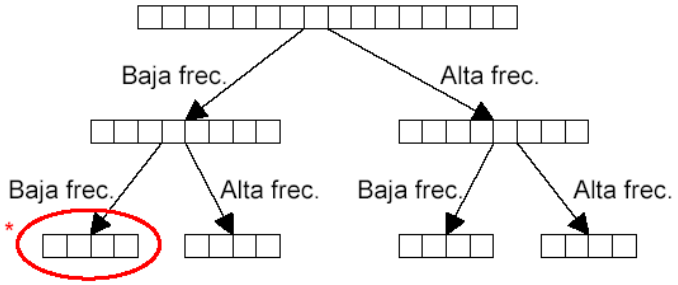


Fig. 4: Descomposición 1-D wavelets en 2 niveles.

Compresión de Imágenes

Transformada Wavelets

Multiresolución o descomposición Wavelets

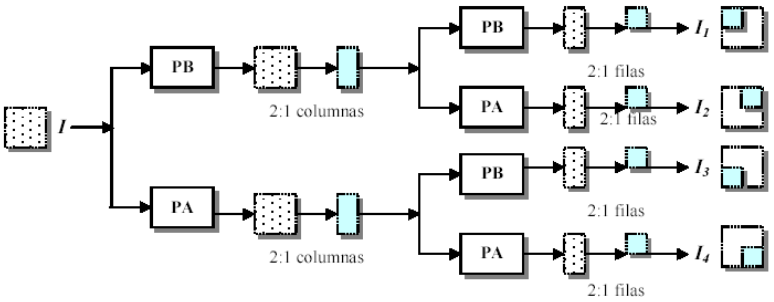


Figura 4: Multiresolución o descomposición *Wavelets*. Los filtros "pasa bajos" PB y "pasa bajos" PA se aplican sobre filas y columnas de la imagen de manera alternada.

Compresión de Imágenes

Transformada Wavelets

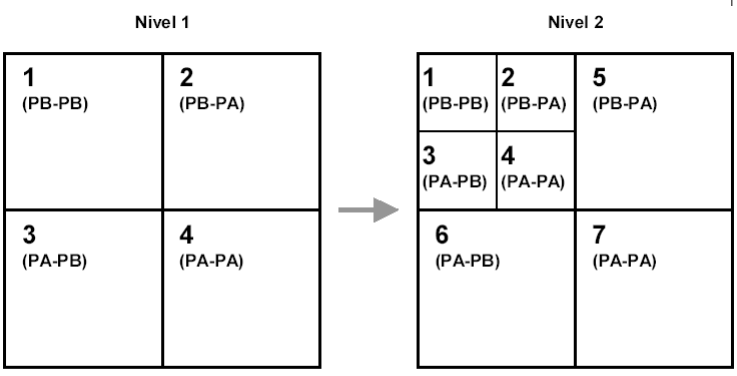


Figura 6: Aplicación de 2 niveles de descomposición *wavelet*. PB indica la aplicación de filtro pasa bajos; PA, filtro pasa altos

Compresión de Imágenes

Transformada Wavelets

Filtrados PA y PB según método de lifting.
Se aplica sobre todas las filas y sobre todas las columnas.

PA
$$y(2n+1) = x(2n+1) - floor\left(\frac{x(2n)+x(2n+2)}{2}\right)$$

PB
$$y(2n) = x(2n) + floor\left(\frac{y(2n-1)+y(2n+1)+2}{4}\right)$$

Compresión de Imágenes

Transformada Wavelets

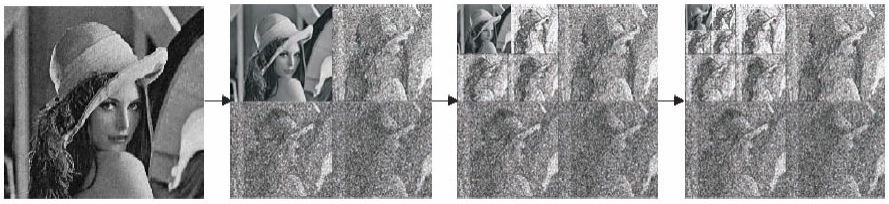


Figura 5: Descomposición *wavelets* en 3 niveles de la imagen clásica "lena".

Compresión de Imágenes

Transformada Wavelets

Figura 7: a) Escalabilidad en distorsión (SNR). b) Escalabilidad en resolución.

Compresión de Imágenes

Transformada Wavelets

Pseudo-Implementación:

```
for nivel=0:2
   n=2^nivel;
   for f=1:2:nf/n
     for c=1:2:nc/n
         R=A(f,c);
         H=A(f,c+1)-A(f,c);
         V=A(f+1,c)-A(f,c);
         D=A(f+1,c+1)-A(f,c);

         B((f+1)/2,(c+1)/2)=R;
         B((f+1)/2,(c+1)/2+nc/n/2)=H;
         B((f+1)/2+nf/n/2,(c+1)/2)=V;
         B((f+1)/2+nf/n/2,(c+1)/2+nc/n/2)=D;
     end;
   end;
   A=B;
 end;
```

Compresión de Imágenes

Transformada Wavelets

Pseudo-Implementación Inversa:

```
for nivel=2:-1:0
   n=2^nivel;
   for f=1:2:nf/n-1
    for c=1:2:nc/n-1
         R=A((f+1)/2,(c+1)/2);
         H=A((f+1)/2,(c+1)/2+nc/n/2);
         V=A((f+1)/2+nf/n/2,(c+1)/2);
         D=A((f+1)/2+nf/n/2,(c+1)/2+nc/n/2);

         B(f,c)=R;
         B(f,c+1)=R+H;
         B(f+1,c)=R+V;
         B(f+1,c+1)=R+D;
     end;
  end;
  A=B;
end;
```


Compresión de Imágenes

EZW

EZW (Embedded Zerotree Wavelet) (Shapiro,1993).

Este método explota las propiedades aportadas por la DWT para obtener resultados satisfactorios en la compresión:

- **un gran porcentaje de coeficientes wavelets próximos a cero**
- **y la agrupación de la energía de la imagen.**

Compresión de Imágenes

EZW

El EZW es sensible al grupo de bits transmitidos por orden de significancia, lo que le permite una compresión progresiva (embedded coding) de la imagen.

Cuantos más bits se añadan al resultado de la compresión, más detalles se estarán transmitiendo

Compresión de Imágenes

EZW

Los primeros coeficientes que se transmiten (o se almacenan) son los que superan un umbral predefinido.

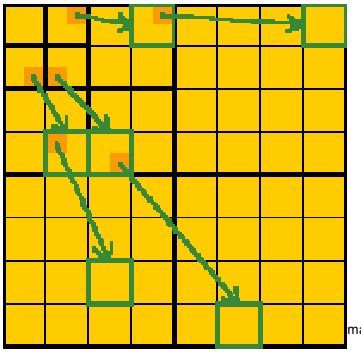
Luego que se han transmitido dichos coeficientes, se disminuye el umbral en potencias de 2 y se transmite el nuevo grupo de coeficientes.

Esto corresponde a una especie de "binarización por umbral" de los coeficientes.

Compresión de Imágenes

EZW

Se construye un mapa de significancias (zerotrees) basado en la búsqueda de los coeficientes mayores o iguales al umbral determinado en el paso anterior.

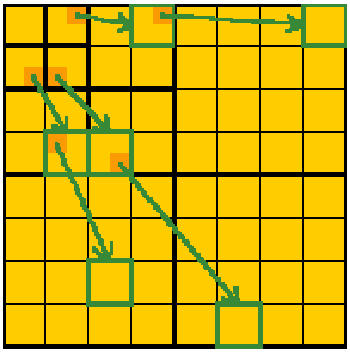


La estructura zerotree agrupa los coeficientes de cuatro en cuatro: cada coeficiente tiene cuatro hijos, cada uno los cuales tiene sus propios cuatro hijos y así sucesivamente.

Compresión de Imágenes

EZW

Por lo general, los hijos tienen unas magnitudes menores que las de sus padres.



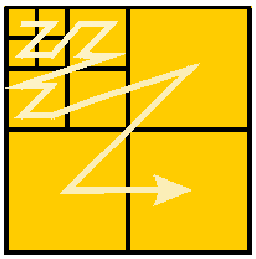
El EZW se aprovecha de esta organización basada en el hecho de que los coeficientes wavelet se decrementan a medida que aumenta la escala.

Así se puede garantizar que los coeficientes de un quadtree son más pequeños que el umbral de estudio, si su padre es más pequeño que el umbral antes mencionado

Compresión de Imágenes

EZW

El EZW transmite el valor de los coeficientes en orden decreciente. El recorrido de la matriz de coeficientes se realiza en zig-zag transmitiendo el n-ésimo bit más significativo en cada pasada.



Este pequeño detalle lo hace muy robusto a la pérdida de paquete en transmisiones, por ejemplo inalámbricas.

Compresión de Imágenes

SPIHT

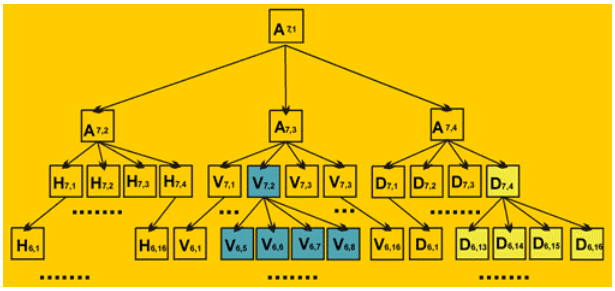
El algoritmo SPIHT (Set Partitioning In Hierarchical Trees) se basa en la clasificación de árboles jerárquicos de coeficientes de la transformada wavelet.

Al igual que el EZW permite la transmisión progresiva de la información por orden de bits más significativos, y también logra imágenes con una gran calidad y altas tasas de compresión.

Compresión de Imágenes

SPIHT

El SPIHT ofrece una nueva y mejor implementación del EZW basada en la utilización de conjuntos de datos organizados en árboles jerárquicos, es decir, el SPIHT tiene en cuenta la significancia de la descendencia del coeficiente que codifica



Compresión de Imágenes

Compresión Fractal

La compresión fractal es una técnica en desarrollo.

Pequeños bloques con similitudes se pueden codificar un mismo conjunto de datos

Cuantificación del Error entre dos imágenes:

A: Imagen Original
B: Imagen Comprimida

Cuantificación del "error" o "ruido" en imágenes

$$error = A(c,f) - B(c,f)$$

Error Cuadrático Medio, permite estimar la varianza del ruido de la imagen

$$ecm = \frac{1}{nc \cdot nf} \sum_{c=1}^{nc} \sum_{f=1}^{nf} (A(c,f) - B(c,f))^2$$

Cuantificación del “error” o “ruido” en imágenes

Relación Señal a Ruido :*La Relación Señal a Ruido Máxima o PSNR (del inglés Peak Signal-to-Noise Ratio) es un término utilizado en ingeniería para definir la relación entre la máxima energía posible de una señal y el ruido que afecta a su representación fidedigna. Debido a que muchas señales tienen un gran rango dinámico, el PSNR se expresa generalmente en escala logarítmica, utilizando como unidad el decibel.*

El uso más habitual del PSNR es como medida cuantitativa de la calidad de la reconstrucción en el ámbito de la compresión de imágenes. Se calcula en decibeles, el valor máximo de una imagen gris (255) dividido por la raíz del error cuadrático medio.

PSNR: Relación Señal a Ruido Máxima en decibeles

$$PSNR_{dB} = 20\log_{10}\left(\frac{255}{\sqrt{ecm^2}}\right)$$