Div. Ingeniería de Sistemas y Automática

Universidad Miguel Hernández

VISIÓN POR COMPUTADOR

Tema 9. Compresión de imágenes

GRUPO DE TECNOLOGÍA INDUSTRIAL

Tabla de Contenidos

Introducción (i)

- Compresión de imágenes:
  - Minimizar el número de bits requeridos para representar una imagen, mientras se retiene la información necesaria
- Razón de compresión:
  - Razón entre la dimensión de la imagen no comprimida y la dimensión de la imagen comprimida (D_{NC}/D_C)
- Clave de la compresión:
  - Retención de la información necesaria
  - Diferencia entre datos e información (valor del nivel de gris de cada píxel/interpretación de los datos)
- Los algoritmos de compresión se desarrollan teniendo en cuenta la redundancia que existe en los datos.

Introducción (ii)

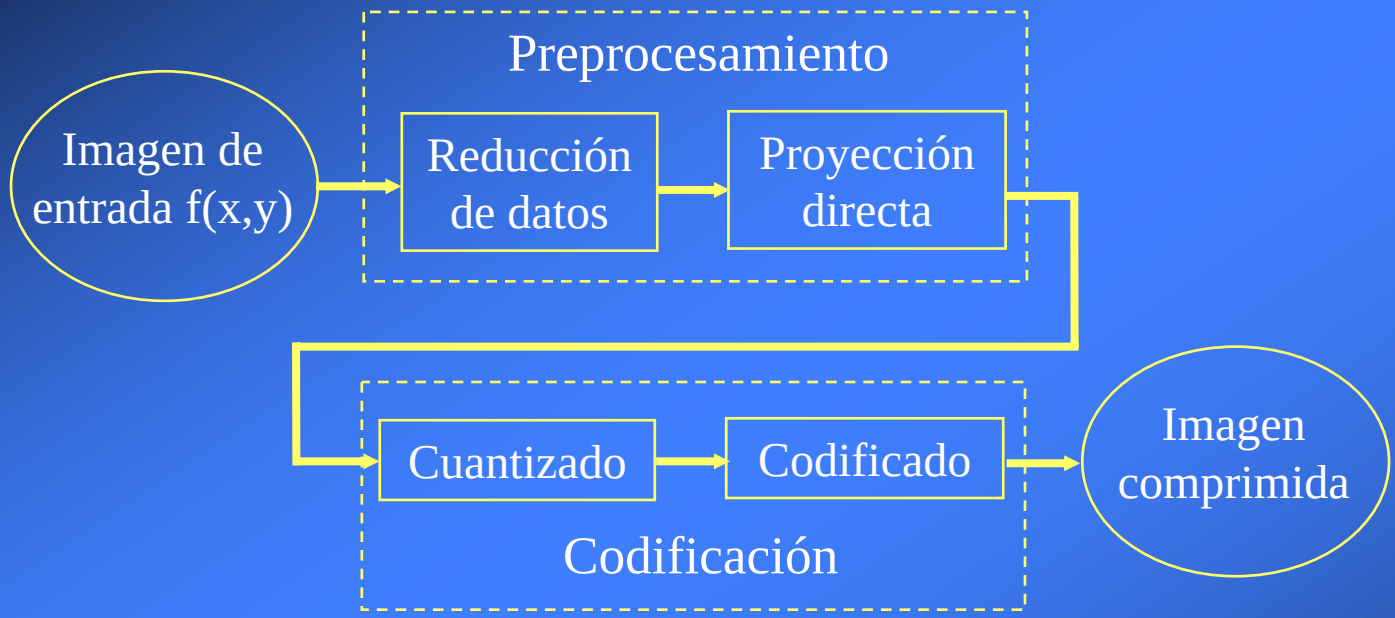
- Tipos de redundancia en las imágenes:
  - Codificación: Los datos usados para representar una imagen no se usan de forma óptima.
    - Ejemplo: Almacenar una imagen que tiene únicamente 16 niveles de gris en un formato de 8 bits por pixel (sólo serían necesario 4 bits).
  - Interpixel: Los pixels adyacentes tienden a estar altamente correlados.
    - En muchas imágenes los niveles de gris no cambian rápidamente, sino gradualmente.
  - Psicovisual: Cierto tipo de información es más importante para el sistema de visión humano que otro tipo de información.
    - Ejemplo: El sistema de visión humano sólo puede percibir frecuencias espaciales por debajo de 50 ciclos (cualquier información de mayor frecuencia carece de interés).
- Algoritmos de compresión de imágenes:
  - Determinan los mínimos datos requeridos para mantener la información necesaria, teniendo en cuenta la redundancia que existe en la imagen.

Introducción (iii)

- Métodos de compresión:
  - Compresión sin pérdidas (lossless)
    - Se preservan los datos.
    - La imagen se recupera de forma exacta a partir de los datos comprimidos.
  - Compresión con pérdidas (lossy)
    - Permiten pérdida de datos.
    - La imagen original no puede recuperarse de forma exacta a partir de la comprimida.
    - Estos métodos de compresión han de establecer un compromiso entre la calidad de la imagen y el grado de compresión.
    - Con alguno de los métodos más avanzados las imágenes se pueden comprimir entre 10 y 20 veces sin pérdida de información visible.

Modelo del sistema de compresión (i)

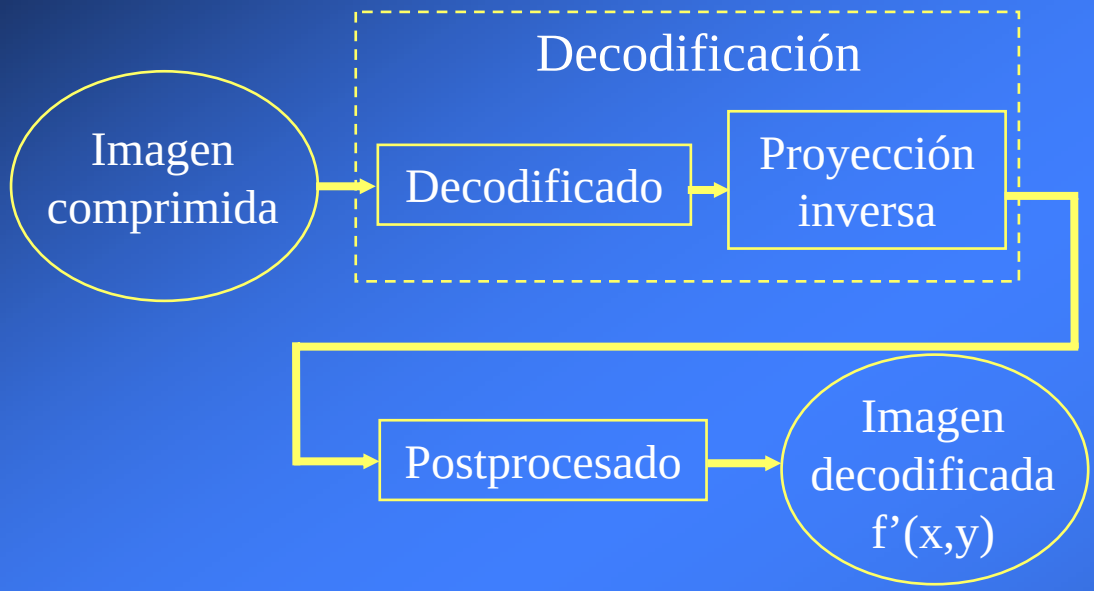
Compresor



- Reducción de datos: cuantización del nivel de gris, realzado o eliminación de ruido
- Proyección directa: se convierten los datos a otro espacio matemático
- Cuantizado: discretiza los valores continuos del estado anterior
- Codificado: convierte los datos discretos del cuantizador a un código de forma óptima

Modelo del sistema de compresión (ii)

Descompresor



- Decodificado: invierte el codificado original proyectando los códigos a los valores cuantizados originales
- Proyección inversa: se invierte el proceso de proyección original
- Postprocesado: realza el aspecto final de la imagen para aminorar cualquier efecto no deseado de la compresión

Modelo del sistema de compresión (iii)

- Preprocesado
  - Preparar la imagen para la codificación eliminando información no relevante
  - Ejemplo

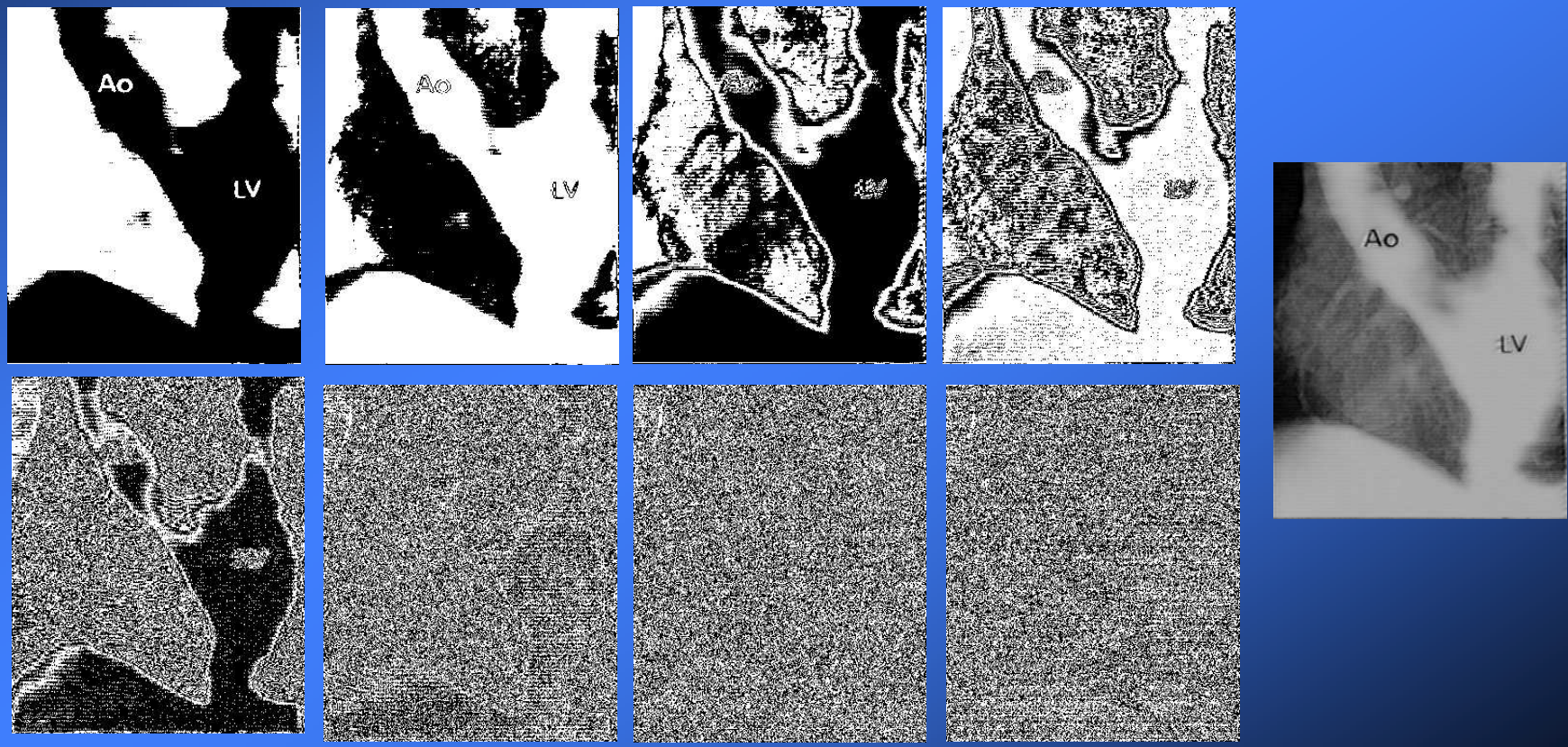


Tabla de Contenidos

Criterios de Fidelidad (i)

- Permiten determinar la bondad de los métodos de compresión con pérdidas al evaluar la similitud entre la imagen original y la recuperada.
- Criterios de fidelidad objetivos:
  - Proveen ecuaciones para hallar la cantidad de error en la imagen descomprimida.
  - Las medidas más usadas son:

Raíz cuadrada del error al cuadrado medio

$$e_{RMS} = \sqrt{\frac{1}{N^2}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}\left[\hat{I}(r,c) - I(r,c)\right]^2}$$

Raíz cuadrada entre la media de los cuadrados de la señal y el ruido

$$SNR_{RMS} = \sqrt{\frac{\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}\left[\hat{I}(r,c)\right]^2}{\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}\left[\hat{I}(r,c) - I(r,c)\right]^2}}$$

Razón de pico señal ruido

$$SNR_P = 10\log_{10}\frac{(L-1)^2}{\frac{1}{N^2}\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}\left[\hat{I}(r,c) - I(r,c)\right]^2}$$

Criterios de Fidelidad (ii)

- Criterios de fidelidad subjetivos:
  - Requieren la definición de una escala cualitativa para medir la bondad de la imagen comprimida
  - Mejor método para comparar los algoritmos si el objetivo es lograr imágenes de alta calidad según nuestra percepción visual
  - Las medidas subjetivas pueden clasificarse en:
    - Pruebas de daño: se clasifican las imágenes según lo malas que son
    - Pruebas de calidad: se clasifican las imágenes en función de lo buenas que son
    - Pruebas de comparación: se evalúan las imágenes de acuerdo a una referencia

Tabla de Contenidos

Métodos de compresión sin pérdidas

- Proporcionan una reducción relativamente pequeña en el tamaño de las imágenes:
  - 10% - 50%
- Técnicas:
  - Codificación de Huffman
  - Codificación en planos de bit
  - Run-Length Coding (Código de longitud de cadenas)
  - Codificación del contorno
  - Algoritmo de Lempel-Ziv-Welch

Codificación de Huffman (i)

- Características
  - Sirve cuando los pixels cuantificados no se encuentran uniformemente distribuidos.
  - Se genera un código de longitud variable para cada bloque donde los bloques muy probables se representan por códigos de longitud pequeña.
- Ejemplo:

Símbolo	Código	Probabilidad	Código Huffman
s_0	000	0.25	00
s_1	001	0.21	10
s_2	010	0.15	010
s_3	011	0.14	011
s_4	100	0.0625	1100
s_5	101	0.0625	1101
s_6	110	0.0625	1110
s_7	111	0.0625	1111

000101101011 → $s_0\ s_2\ s_5\ s_3$

Codificación de Huffman (ii)

- Algoritmo:
  1. Obtener el histograma de la imagen y, a partir de él, las probabilidades de los niveles de gris
  2. Ordenar las probabilidades de menor a mayor
  3. Combinar mediante adición los dos menores
  4. Ir a 2 hasta que sólo queden dos probabilidades
  5. Retrocediendo en el árbol, generar códigos alternando las asignaciones de 0 y 1
- Ejemplo:
  - Imagen con 2 bits/píxel (4 niveles de gris)
  - 20 filas x 20 columnas

Codificación en Planos de Bit (1)

- Objetivo:
  - Descomponer una imagen multinivel (monocromo o color) en una serie de imágenes binarias y comprimir estas imágenes binarias.
- Problemas:
  - Pequeños cambios en los niveles de gris pueden tener un impacto significativo en la complejidad de los planos de bit (127 a 128).
  - Se puede evitar este problema al representar la imagen por un código Gray.
    - Un Código Gray es aquel que al aumentar un nivel su intensidad, su valor codificado sólo cambia en un bit.

Código decimal	Código binario	Código de Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

Codificación en Planos de Bit (2)

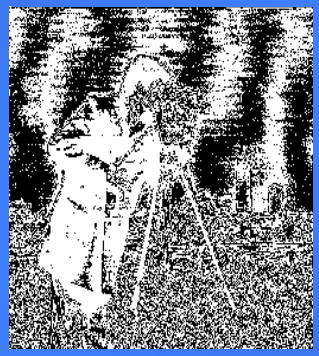
Imagen con 256 Tonos de Gris

Bit 7

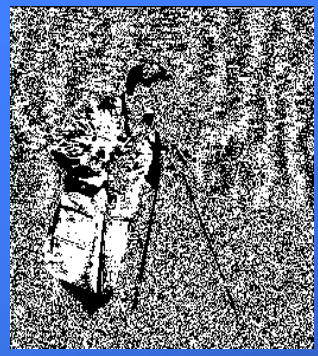
Bit 6

Bit 5

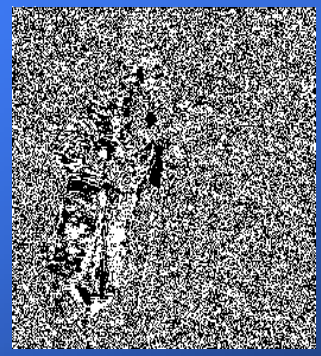
Bit 4



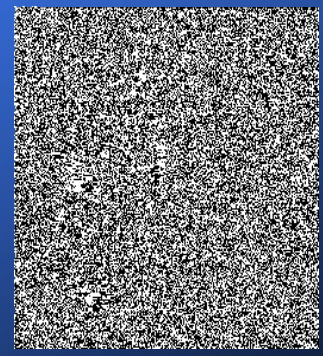
Bit 3



Bit 2



Bit 1



Bit 0

Run-Length Coding (RLC-1)

- Características:
  - En imágenes binarias, se puede codificar el número de ceros entre dos unos consecutivos.
  - Es útil cuando se esperan grandes cadenas de ceros de forma continua.
    - ⇦Documentos impresos
    - ⇦Gráficos, etc...
  - Para codificar el número de ceros entre dos unos se realiza una codificación de longitud fija.
  - Se ha convertido en el estándar para la compresión vía FAX.
  - Ejemplo:

000010010000000001000000000000001

0100001010101111

Run-Length Coding (RLC-2)

- Otra posibilidad consiste en almacenar el número de 0's y de 1's en la imagen que se desea comprimir:
  - Es necesario definir que se representa en primer lugar
  - Ejemplo:

```
0 0 0 0 0 0 0 0   1000 (8)
1 1 1 1 0 0 0 0   0000 0100 0100 (0, 4, 4)
0 1 1 0 0 0 0 0   0001 0010 0101 (1, 2, 5)
0 1 1 1 1 1 0 0   0001 0101 0010 (1, 5, 2)
0 1 1 1 0 0 1 0   0001 0011 0010 0001 0001 (1, 3, 2, 1, 1)
0 0 1 0 0 1 1 0   0010 0001 0010 0010 0001 (2, 1, 2, 2, 1)
1 1 1 1 0 1 0 0   0000 0100 0001 0001 0010 (0,4, 1, 1, 2)
0 0 0 0 0 0 0 0   1000 (8)
```

  TOTAL: 64 bits

- El método puede extenderse a imágenes con niveles de gris al utilizar la técnica de codificación en planos de bit.

IMAGEN ORIGINAL

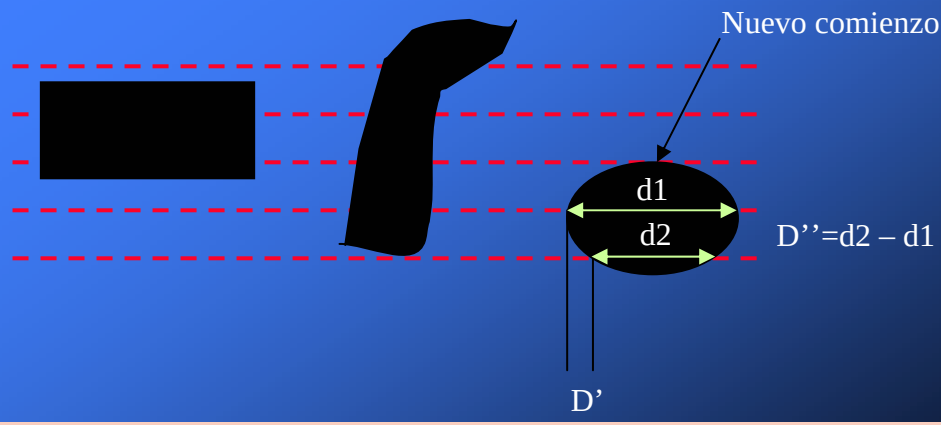
Planos de Bit 7 y 6

Planos de Bit 7,6,5,4

Planos de Bit 7,6,5,4,3,2

Codificación del Contorno

- Objetivo:
  - Representar cada contorno por un conjunto de puntos de borde o por un único punto de borde y un conjunto de direcciones.
- Codificación:
  - Para cada imagen binaria se toman dos medidas:
    - D': Distancia entre las coordenadas de comienzo de dos líneas consecutivas.
    - D'': Diferencia entre el tamaño de los contornos en dos líneas consecutivas.
    - Mensajes especiales de fin de contorno y de inicio de contorno.
      - Se codifican las coordenadas de las filas y columnas de cada inicio y fin de contorno



Algoritmo de Lempel-Ziv-Welch (LZW)

- Se creó inicialmente para codificar cadenas de datos y no para comprimir imágenes.
- Funciona al crear una tabla que contiene secuencias de pixels y sus correspondientes códigos asociados.
- La codificación LZW usa códigos con más bits que los datos originales.
  - Ejemplo:
    - Para imágenes de 8 bits (256 niveles de gris) podrían usarse códigos de 12 bits.
    - De esta forma es posible tener una tabla con $2^{12} = 4096$ entradas distintas:
      - Los códigos del 0 a 255 de estas entradas son los posibles valores de cada uno de los pixels
      - Del 256 al 4095 se utilizan para representar cadenas de bytes
        - Ejemplo: código 523 para la secuencia de bytes 121 200 45
- Los formatos GIF y TIFF presentan un algoritmo LZW específico patentado por Unisys Corporation.
- Existen versiones similares de este algoritmo que se usan en la función 'compress' de UNIX, y en la función 'gzip'.

Métodos de compresión con pérdidas

- Exigen un compromiso entre el grado de compresión y la calidad de la imagen.
  - Existen parámetros ajustables para permitir al usuario seleccionar la razón de compresión y la fidelidad de la imagen deseada.
- Se consiguen razones de compresión de:
  - 10 a 20 veces sin pérdida de información
  - 30 a 50 veces con una degradación imperceptible
- Técnicas:
  - En el dominio espacial:
    - Run-Length-Coding a Nivel de Gris
    - Block Truncation Coding
    - Codificación Predictiva Diferencial
  - En el dominio transformado:
    - Codificación de la Transformada
    - Transformada Discreta del Coseno
    - Algoritmo JPEG
    - Compresión Wavelet

Run-Length-Coding a Nivel de Gris

- Objetivo: Reducir el número de niveles de gris en una imagen y entonces usar las técnicas RLC.
- Para que este método sea efectivo es necesario que los datos de la imagen reducida se mapeen en un código para cada nivel de gris, de forma que los números adyacentes difieran en un único bit (Código Gray).
- Un algoritmo más sofisticado es el RLC basado en una ventana dinámica:
  - No es necesario que las cadenas tengan el mismo valor
  - Se mueve una ventana dinámica a lo largo de la imagen
  - Esta ventana se hace más grande o más pequeña en función de los valores de sus pixels. Todos estos valores han de encontrarse dentro de un rango dinámico definido.
  - Se codifica el tamaño de la ventana y el valor promedio de todos ellos.
  - Ejemplo: 65 67 66 64 63 64 70, con un rango dinámico de 5
    - ⇦ Primer valor de referencia: R = 65
    - ⇦ Rango: Min = R –(5-1), Max = R + (5-1)

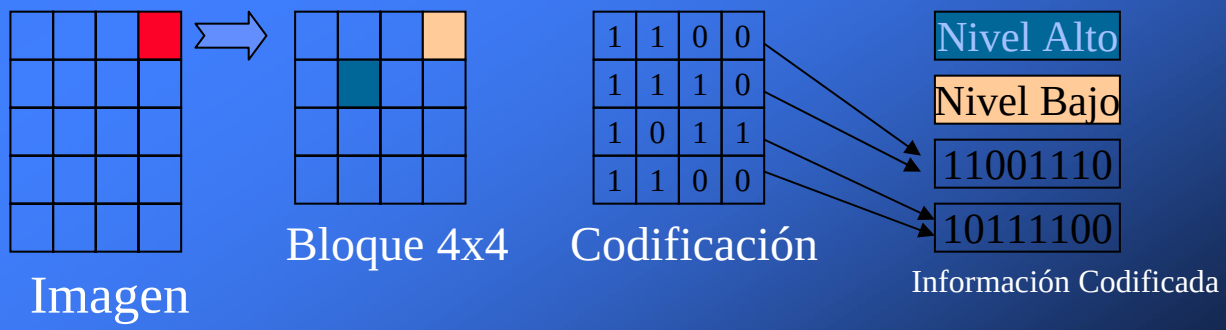
65	67	66	64	63	64	70
61-69	63-71	62-70	60-68	61-67	60-68	
	63-69	63-69	63-68	63-67	63-67	

Se codifica: Run-Length = 6

Valor = (65+67+66+64+63+64)/6 =65

Block Truncation Coding (BTC-1)

- Objetivo: Dividir la imagen en bloques reduciendo el número de niveles de gris de cada bloque.
- Algoritmo BTC básico:
  - Se divide la imagen en bloques de 4x4 pixels.
  - Se selecciona el nivel alto y el bajo en cada bloque.
  - Cada valor de pixel en el bloque se compara con la media del bloque. Se asigna un 1 si es mayor y un 0 si es menor.
  - Cada bloque 4x4 se codifica mediante 4 bytes: 2 bytes para almacenar los niveles, y 2 bytes para almacenar la cadena de ceros y unos.
    - ⇦ Razón de compresión: 16:4 → 4:1
  - Ejemplo:



Block Truncation Coding (BTC-2)

Imagen Original

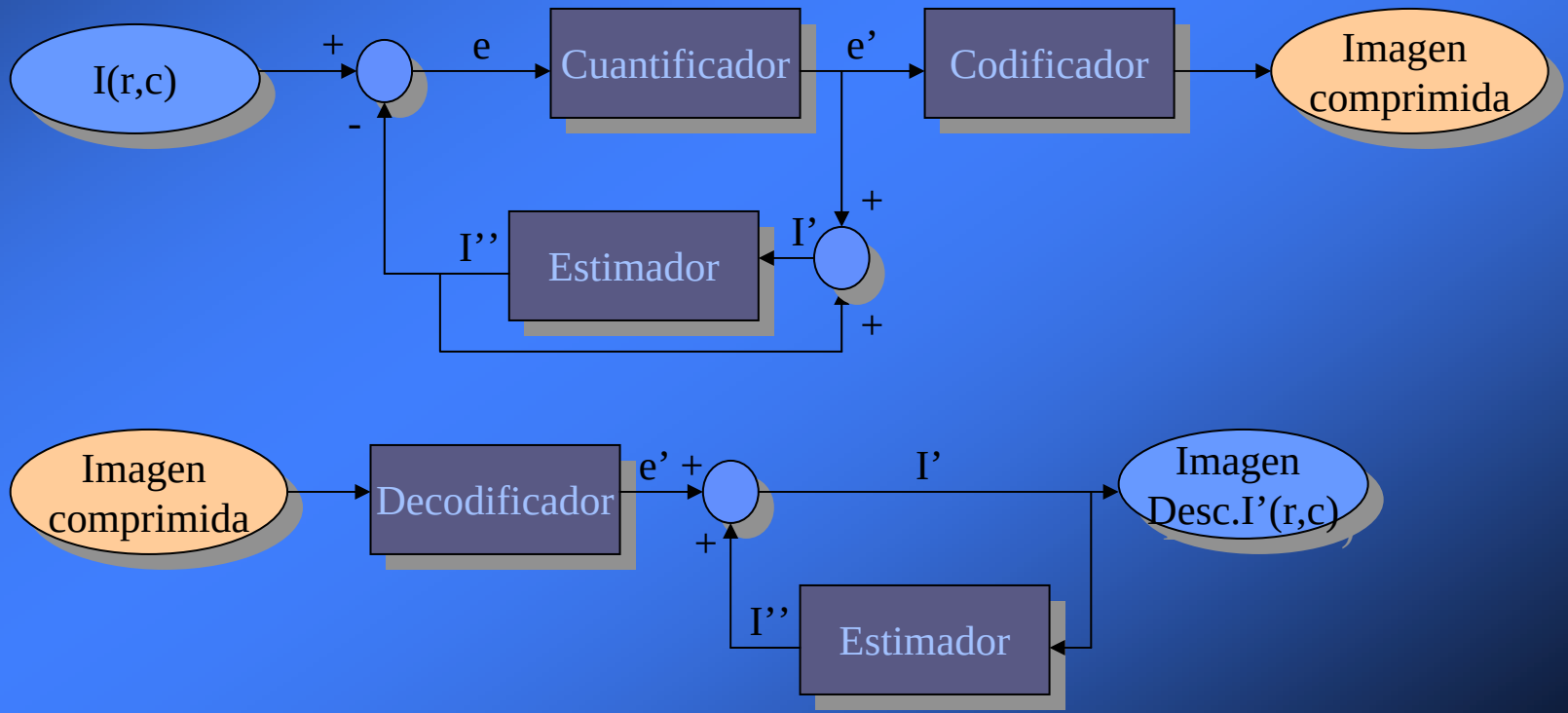
BTC 4x4

BTC 8x8

BTC 32x32

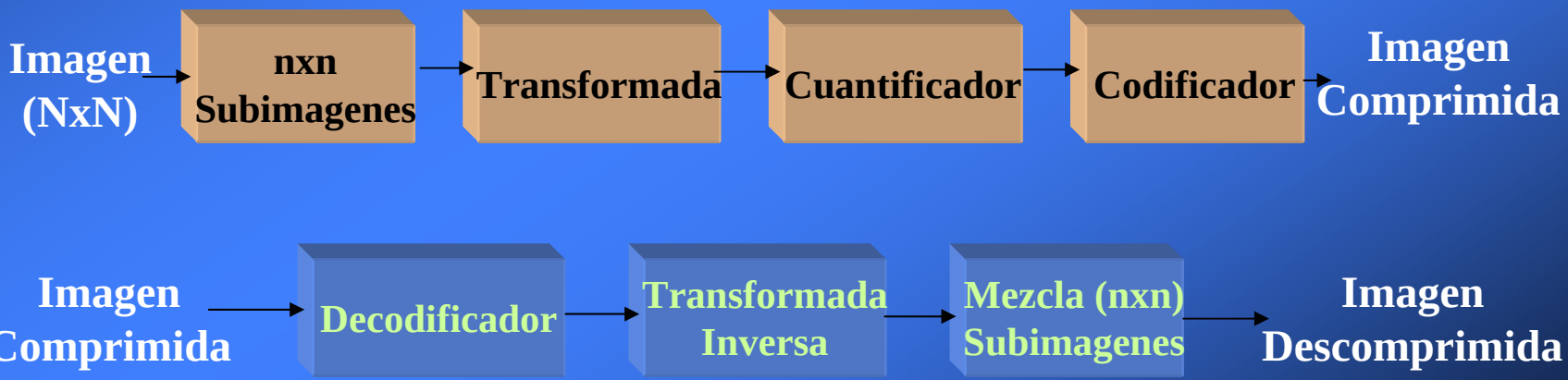
Codificación Predictiva Diferencial

- Objetivo: Predecir el nivel de gris del pixel siguiente en función de los valores previos, codificando la diferencia entre el valor estimado y el valor real.
- Justificación: Los pixels adyacentes en una imagen generalmente se encuentran altamente correlados.



Codificación de la Transformada

- En lugar de operar directamente con la información espacial, se opera con el resultado de realizar una transformación sobre la imagen.
- Se mapea la imagen en un conjunto de coeficientes de la transformada que son cuantificados y codificados.



Transformada Discreta del Coseno

- La imagen se divide en bloques sobre los que se calcula la transformada.
- Son los coeficientes de la transformada los que se codifican y se cuantifican.
- La mayor parte de la información se centra en unos cuantos coeficientes de forma que los de frecuencias altas pueden ser eliminados.
- Cuanto mayor sea el número de coeficientes almacenados mejor será la imagen comprimida obtenida, pero menor la razón de compresión.

$$c(u,v)=\alpha(u)\alpha(v)\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}I(r,c)\cos\left[\frac{(2r+1)u\pi}{2N}\right]\cos\left[\frac{(2r+1)v\pi}{2N}\right]$$

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

Algoritmo JPEG (i)

- En 1987 la Joint Photographic Experts Group formalizaron el estándar JPEG.
- Características:
  - Se usa la DCT en bloques de 8x8 pixels
  - Antes de realizar la DCT, los valores de los pixels se desplazan para centrarse en 0.
  - Los coeficientes DCT se cuantifican y se reordenan utilizando un patrón en zig-zag.
    - Se obtienen los coeficientes en orden creciente respecto de la frecuencia espacial
  - El proceso de codificación se diseña para aprovechar la ventaja sobre las cadenas de cero que resultan de la ordenación
  - Codificación:
    - Los coeficientes de frecuencia no cero se codifican utilizando un código de longitud variable.
    - El coeficiente de frecuencia cero se codifica de forma diferencial con respecto al bloque previo.

Algoritmo JPEG (ii)

Imagen Original

381 Kbytes

Imagen Comprimida JPEG

33 Kbytes

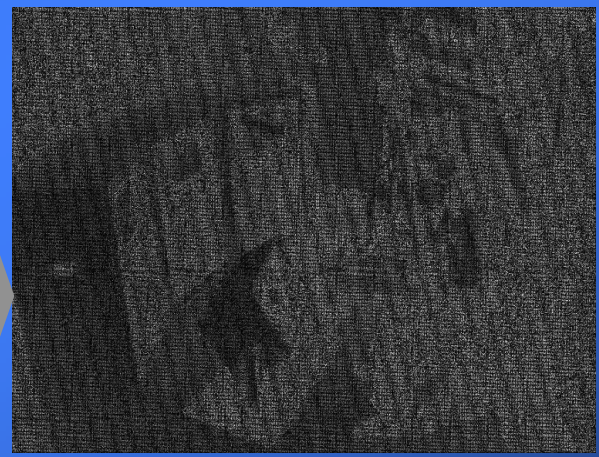
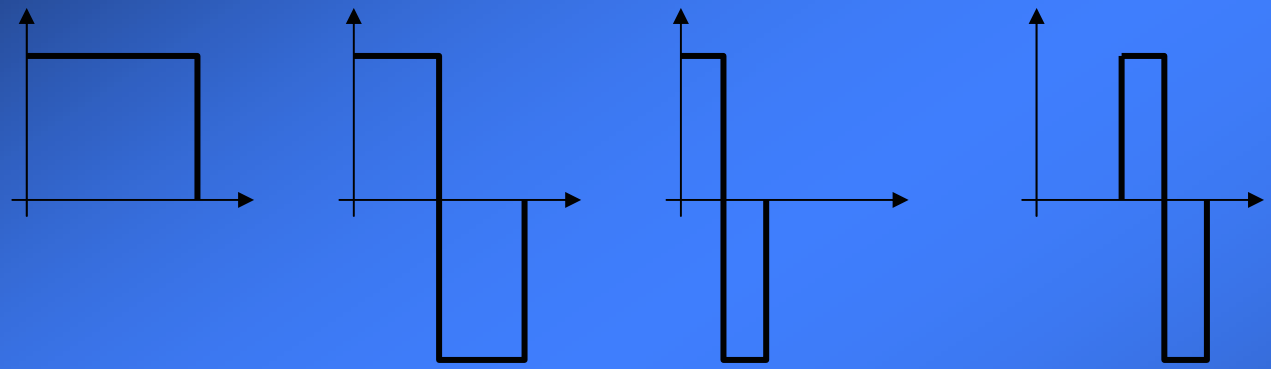


Imagen de Error Multiplicada por 10

$e_{RMS} = 5{,}3792$

Transformada Wavelet (1)

- Las funciones base de la transformada Wavelet son versiones desplazadas y expandidas de una misma. Por ejemplo las funciones de HAAR:



- Estos filtros deben ser perfectos, es decir, cualquier distorsión introducida en un sentido se debe cancelar con el inverso.
- Típicamente se usan filtros de convolución 1D, puesto que las funciones base de la transformada 2D son separables.:
  - Haar, Daubechies, Hadamard, etc.

Transformada Wavelet (2)

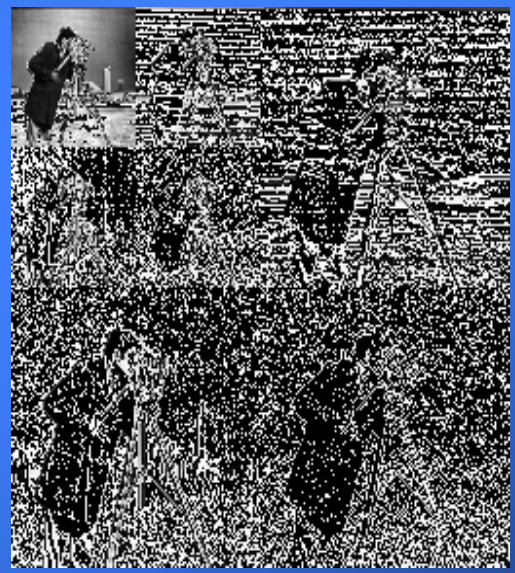
- La Transformada Wavelet particiona una imagen en cuatro subimagenes:
  - Una imagen que ha sido filtrada (paso alto) en sentido horizontal y vertical
  - Una imagen filtrada en vertical (paso alto) y en horizontal (paso bajo)
  - Una imagen filtrada en vertical (paso bajo) y en horizontal (paso alto)
  - Una imagen filtrada en paso bajo en ambos sentidos
- La transformada wavelet mantiene información espacial puesto que la imagen aún es visible en el dominio de la transformada.
- La convención usada para representar los resultados de la transformada wavelet es:

LOW/ LOW	**LOW/ HIGH**
HIGH/ LOW	**HIGH/ HIGH**

Transformada Wavelet (3)

Imagen Original

Tr. Wavelet en 4 bandas



Tr. Wavelet en 7 bandas

- Procedimiento general:
  - Realizar la transformada Wavelet con unas funciones base determinadas, utilizando sus máscaras de convolución.
  - Numerar las diferentes bandas wavelet desde 0 a N-1, donde 0 es la banda de baja frecuencia.
  - Cuantificar linealmente la banda 0 a 8 bits.
  - Cuantificar las diferentes bandas usando tamaños de bloque pequeños.
  - Eliminar las bandas de frecuencia más altas.
- Con este método de compresión se alcanzan rangos de 30:1