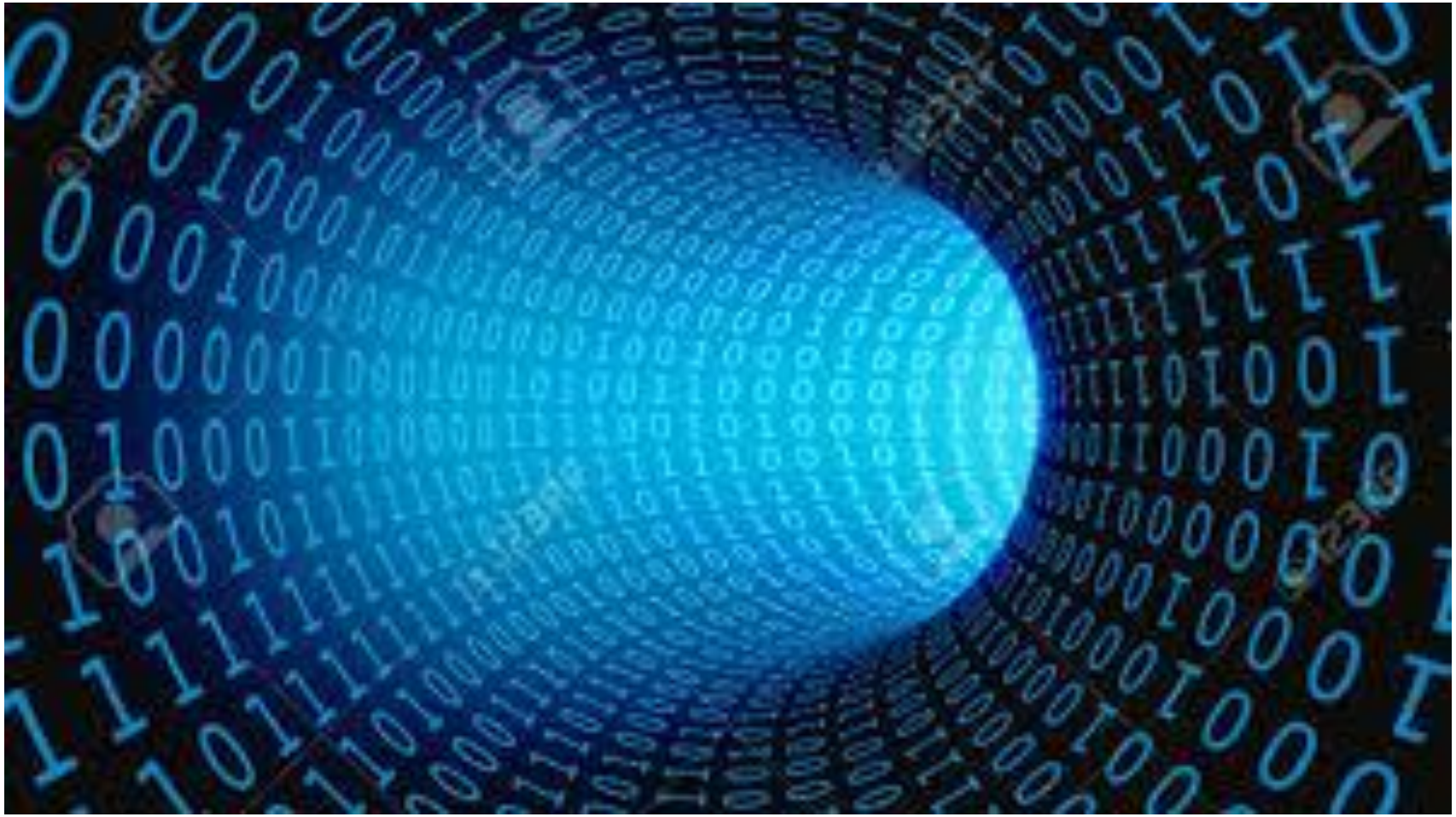
Compresión de imágenes

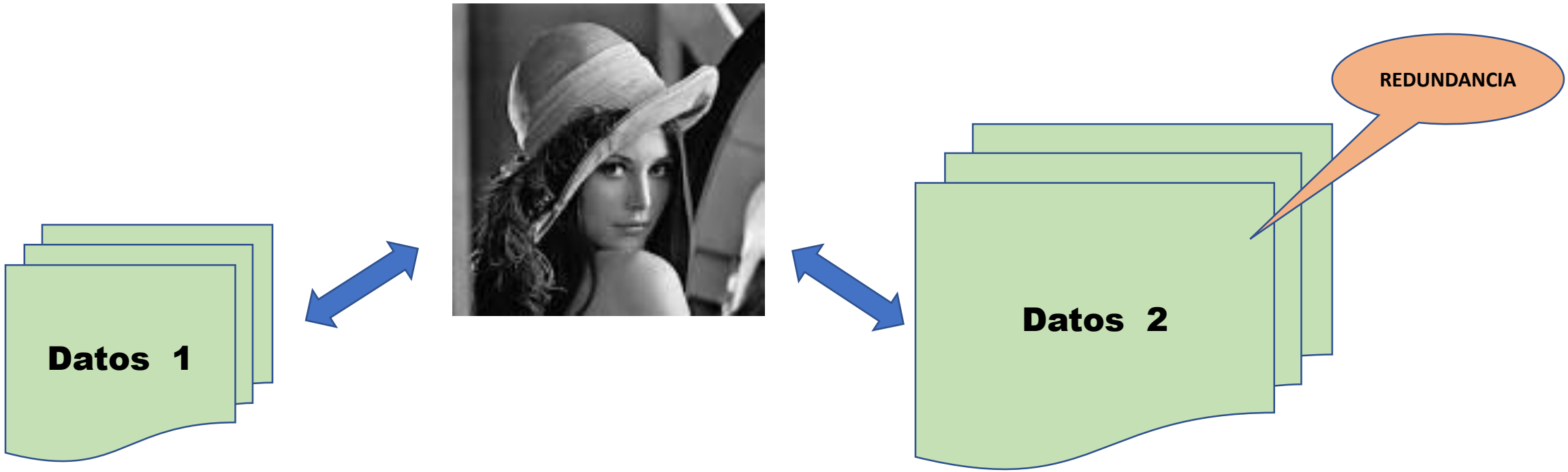


Introducción

La compresión de imágenes comprende un conjunto de técnicas que se aplican a las imágenes para almacenarlas o transmitirlas de manera eficiente.

Introducción

Según González y Woods, la compresión de imágenes es el proceso de reducción del volumen de datos para representar una determinada cantidad de información.



Introducción

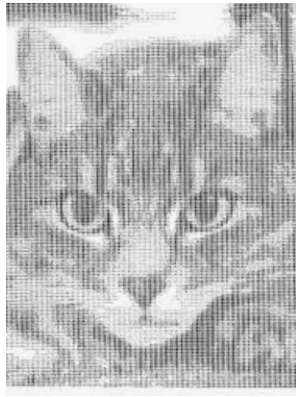
En el caso de las imágenes, existen tres maneras de reducir el número de datos redundantes:

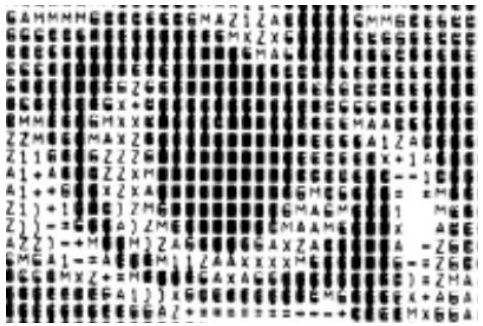
- Eliminar código redundante
- Eliminar píxeles redundantes
- Eliminar redundancia visual

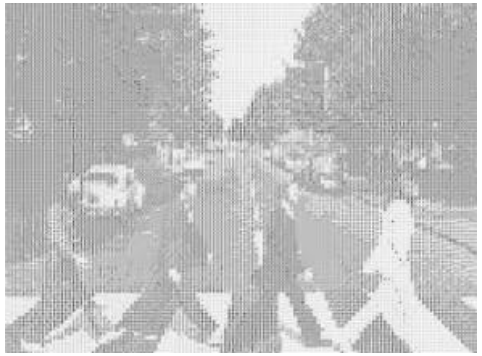
Código redundante

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.

Píxeles redundantes

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

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.

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.

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.

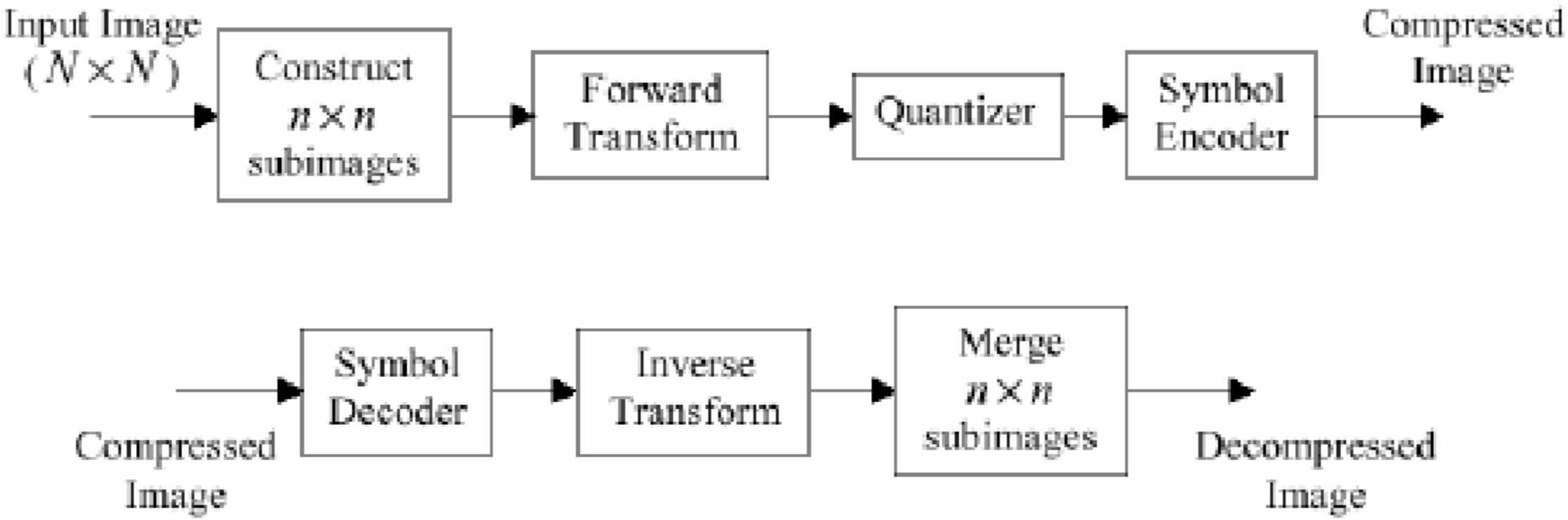
Compresión JPEG

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”.

Transform Coding



ETAPAS Compresión JPEG

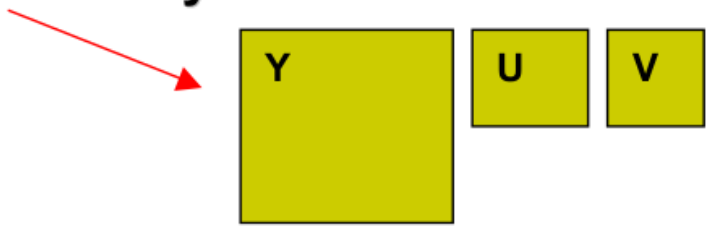
1. Transformación de espacio de color.

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.

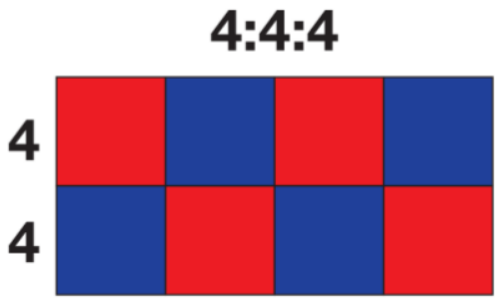
4:4:4 = los tres canales sin submuestreo
4:2:2 = los canales U y V submuestreo horiz. (cada 2 píxeles)
4:2:0 = U submuestreo horiz. , y V subm. Horiz y vert. cada 2
4:0:0 = U y V submuestreo Horiz, y Vert. cada 2 (1/4 tamaño!)

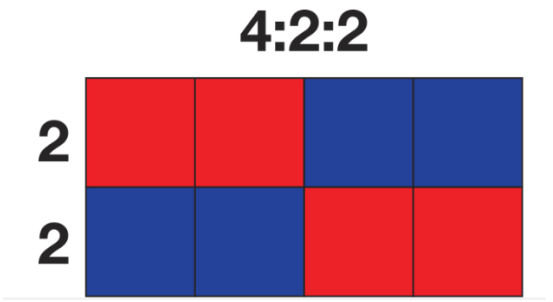


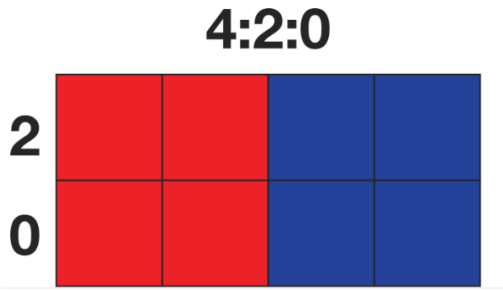
Chroma subsampling

S:H:V S=tamaño de la muestra H=Muestreo horizontal V=Muestreo Vertical

Ejemplo: 4:2:0 Muestra de 4 pixeles , replico cada 2 en sentido Horizontal, No replico en sentido Vertical





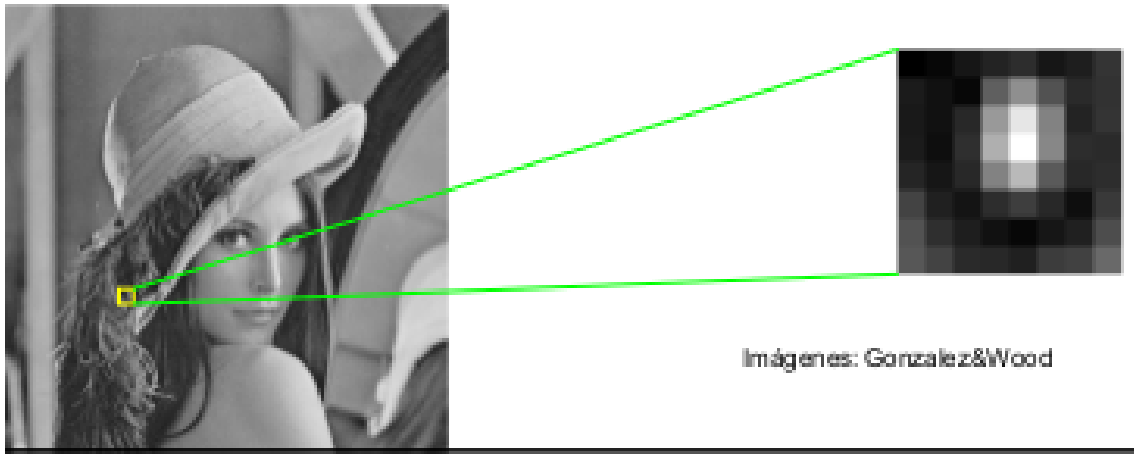


3. Subdivisión imagen en bloques.

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.



Imágenes: Gonzalez&Wood

The discrete cosine transform (DCT)

$$F(u,v) = c(u)c(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

where $c(u) = 1/\sqrt{N}$ if $u = 0$, $\sqrt{2/N}$ otherwise

La DCT es como la DFT pero con coeficientes REALES (usa cosenos en lugar de exponenciales Complejas)

The discrete cosine transform (DCT)

$$F(u,v) = c(u)c(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N}$$

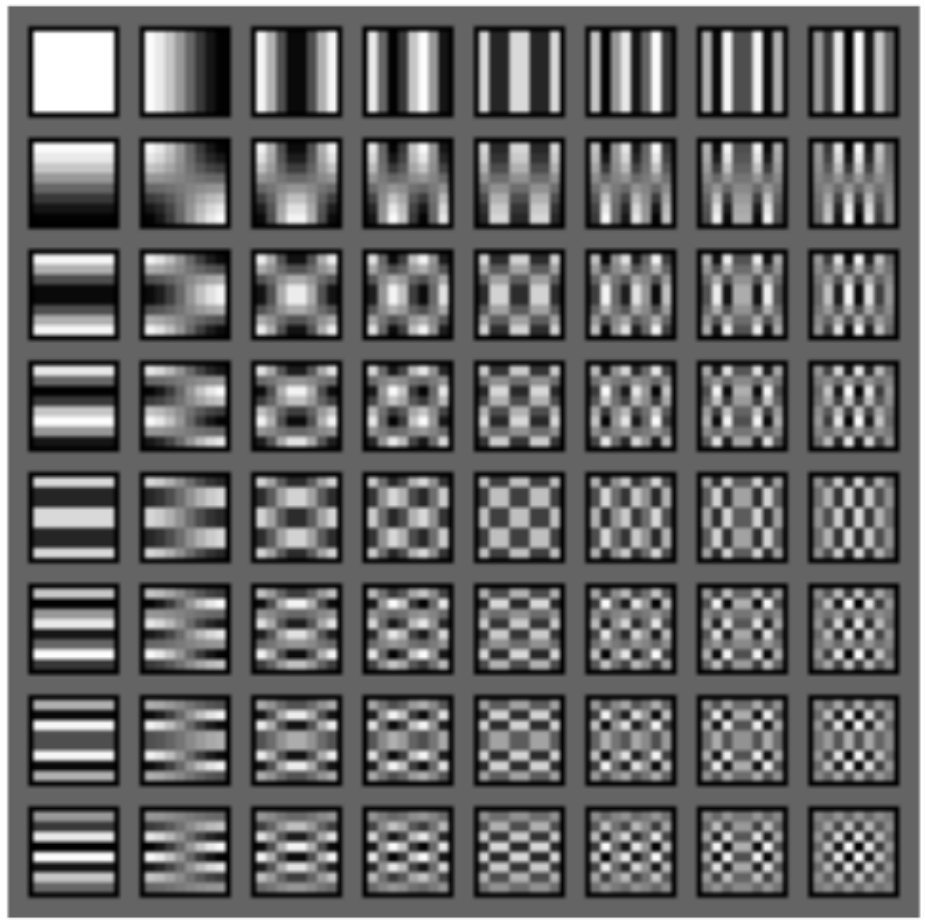
where $c(u) = 1/\sqrt{N}$ if $u = 0$, $\sqrt{2/N}$ otherwise

La DCT es como la DFT pero con coeficientes REALES (usa cosenos en lugar de exponenciales Complejas)

Transformada de Coseno

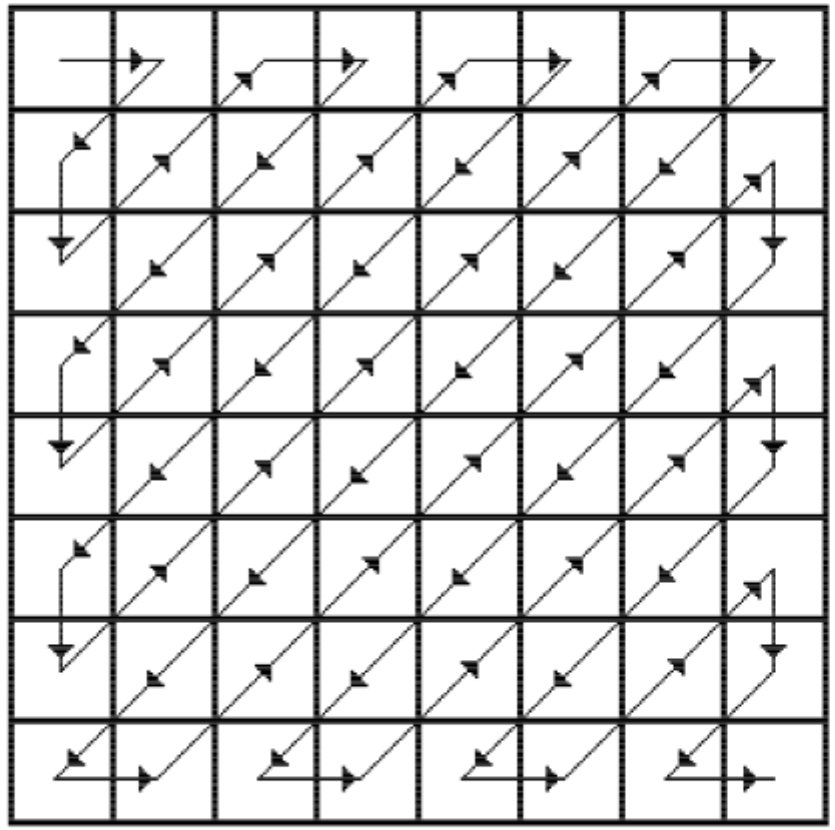
La transformada de coseno se puede interpretar como la proyección de la imagen sobre cada una de las imágenes base. Los coeficientes representan la presencia de dichas componentes o imágenes base.

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



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.

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

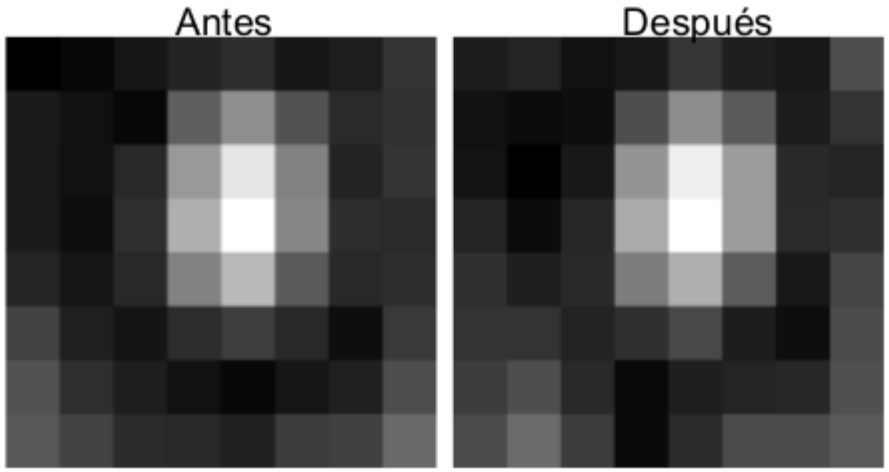
$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61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

-128

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

DCT & Round

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



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:

$$\text{round}\left(\frac{-415}{16}\right) = \text{round}\left(-25.9375\right) = -26$$

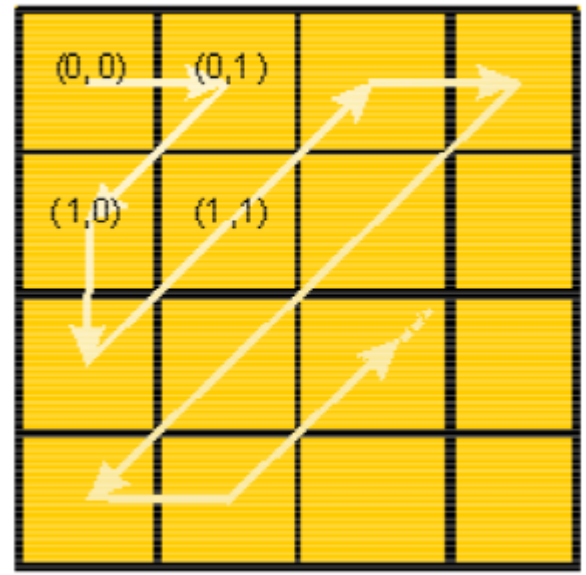
5. Cuantización (o cuantificación):

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.



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.