

# 22.05 - Análisis de Señales y Sistemas Digitales

## Compresión de imágenes JPEG

Matías Bergerman  
Pablo González Leiro  
Milagros Moutin

Instituto Tecnológico de Buenos Aires

10 de diciembre de 2021

# Introducción

JPEG es un popular algoritmo de compresión con pérdida (*lossy*) desarrollado en 1992. El algoritmo aprovecha:

- Las características de la percepción visual humana.
- Propiedades estadísticas de las imágenes (como el espectro espacial).
- Herramientas matemáticas como la DCT y la codificación Huffman.



Fotografía de un gato montés cuya calidad incrementa de izquierda a derecha.

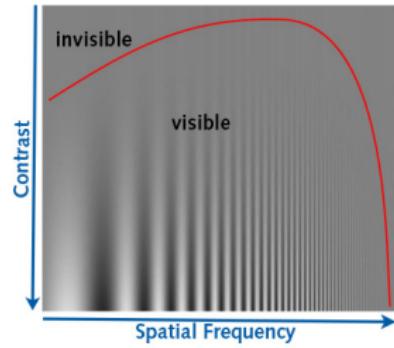
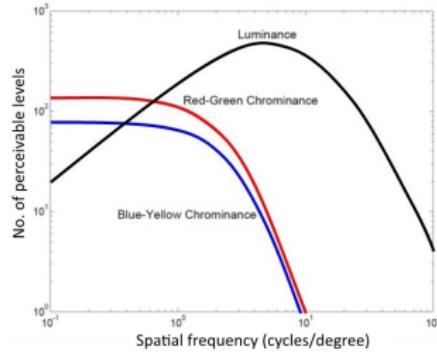
# Introducción: Percepción visual

## El ojo humano:

- Es más sensible a cambios en luminosidad o brillo que a cambios en tonalidad de color.
- No es sensible ante frecuencias espaciales de color altas.

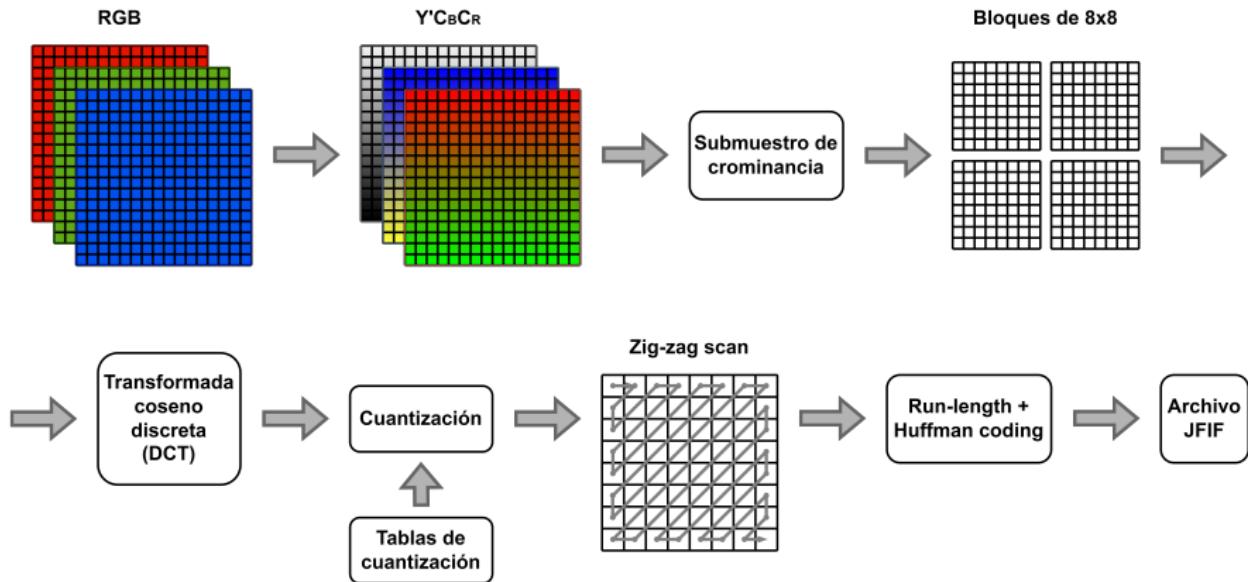
Estas deficiencias permiten reducir el espacio requerido para almacenar una imagen, a la vez que se preserva la forma en que es percibida.

Sensibilidad a luminancia y crominancia en función de frecuencia espacial:



# Introducción: JFIF

El modo de codificación más usual de imágenes JPEG es JFIF. Es por esto que el alcance de este trabajo se limita al estudio e implementación de este método de codificación. Los pasos que sigue el codificador son:

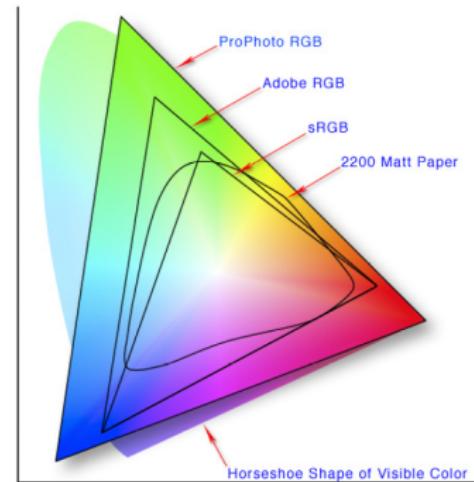


# Espacios y modelos de color

Un espacio de color consiste en un sistema para identificar colores que usualmente depende de un “modelo de color”.

Cada color en un espacio basado en un modelo se puede interpretar como un punto dentro de un sistema de coordenadas, de forma tal que conociendo los valores de estas coordenadas es posible reconstruir el color en cuestión.

En algunos casos como el espacio *Pantone* no se utiliza un modelo.



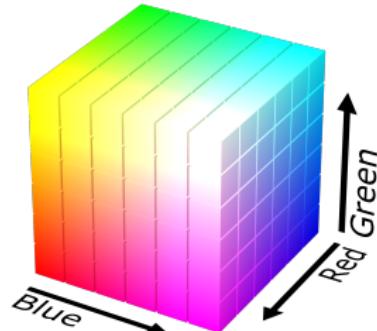
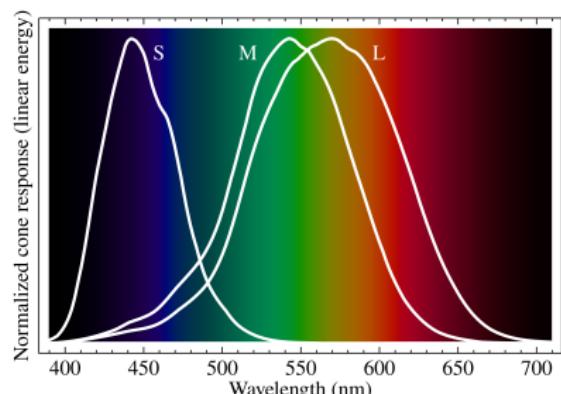
Diferentes espacios de color basados en el modelo RGB.

# Espacio RGB

El ojo humano detecta colores mediante tres tipos de foto-receptores (*conos*) que tienen una distinta respuesta en frecuencia a la luz.

Los máximos de estas respuestas se alinean aproximadamente con los colores azul, verde y rojo (los colores primarios de la luz).

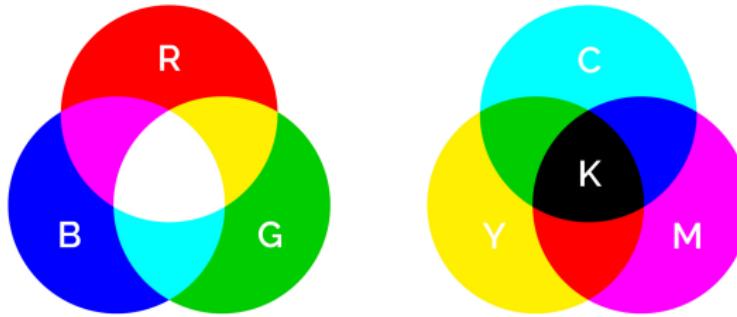
La mayoría de las pantallas digitales aprovechan este fenómeno para emular diferentes colores al estimular los foto-receptores del ojo humano de la misma forma en que serían estimulados por el color deseado.



# Otros espacios de color

En función de las necesidades de la aplicación, existen diversos otros espacios de color. Por ejemplo:

- CMYK (*Cyan, Magenta, Yellow, Key*) refleja las características del proceso de impresión, al indicar la cantidad de cada tinta que se debe aplicar. Representa una síntesis sustractiva de color.
- HSB/HSL (*Hue, Saturation, Brightness*) se utiliza por artistas dado que es más natural considerar al color en términos de matiz y saturación que como una síntesis aditiva o sustractiva.



## Transformación al espacio $YC_B C_R$

En la compresión JPEG el espacio de color utilizado es el  $YC_B C_R$  que consiste de un componente de *luminancia* ( $Y$ ) que representa el brillo, y dos componentes de *crominancia* o *chroma* ( $C_B$  y  $C_R$ ) que representan color:

- Permite aislar la luminosidad del color para que sean tratados por separado.
- No se trata de un espacio de color absoluto, sino que es una forma de codificar información RGB.
- Para poder representar un color  $YC_B C_R$  es necesario conocer el espacio de color RGB específico a utilizar (por ejemplo *sRGB*).
- La transformación entre el espacio RGB y  $YC_B C_R$  corresponde a una transformación lineal.
- Es posible introducir alinealidades mediante una *corrección gamma*.

# Transformación al espacio $YC_B C_R$

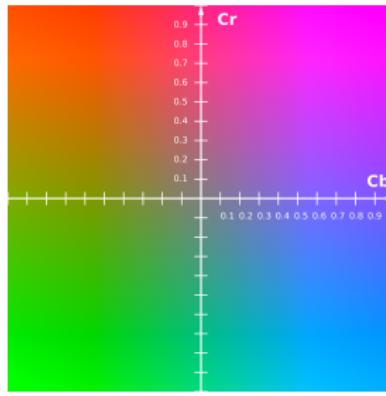
Las ecuaciones correspondientes a esta transformación para valores no-signados de 8 bits son:

$$Y = 0 + (0,299 \cdot R_D) + (0,587 \cdot G_D) + (0,114 \cdot B_D)$$

$$C_B = 128 - (0,168736 \cdot R_D) - (0,331264 \cdot G_D) + (0,5 \cdot B_D)$$

$$C_R = 128 + (0,5 \cdot R_D) - (0,418688 \cdot G_D) - (0,081312 \cdot B_D)$$

Plano  $C_B C_R$  para un valor de luminancia constante de un medio:

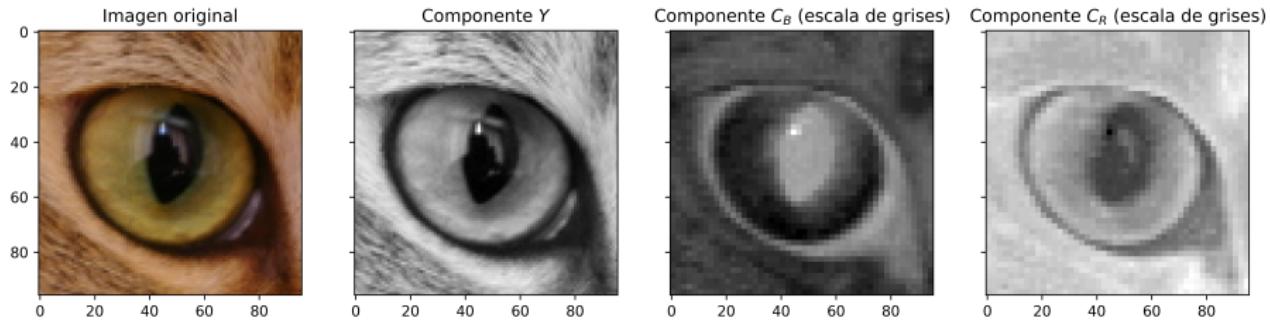


# Submuestreo de crominancia

El próximo paso es reducir la resolución de los componentes  $C_B$  y  $C_R$ .

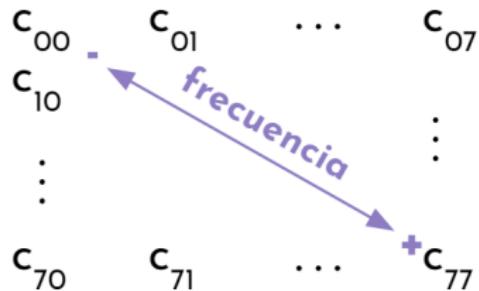
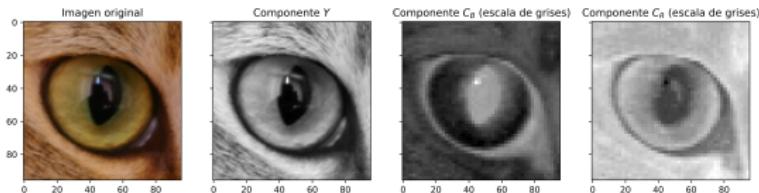
Este es el primer punto en el que se pierde información de la imagen, aunque sin afectar significativamente su percepción.

El *patrón de submuestreo* elegido, denominado 4:2:0, es uno de los más utilizados y corresponde a una reducción a la mitad en la resolución horizontal y vertical.



# Discrete Cosine Transform

El objetivo de esta próxima sección es descomponer esta información en distintas **bandas de frecuencia**. Para esto tomamos bloques de  $8 \times 8$  píxeles de cada una de las componentes  $Y$ ,  $C_b$  y  $C_R$ .



# Transformadas

$x(n)$  entre  $0 \leq n \leq N - 1$  se puede escribir como:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} A(k) \phi_k(n)$$

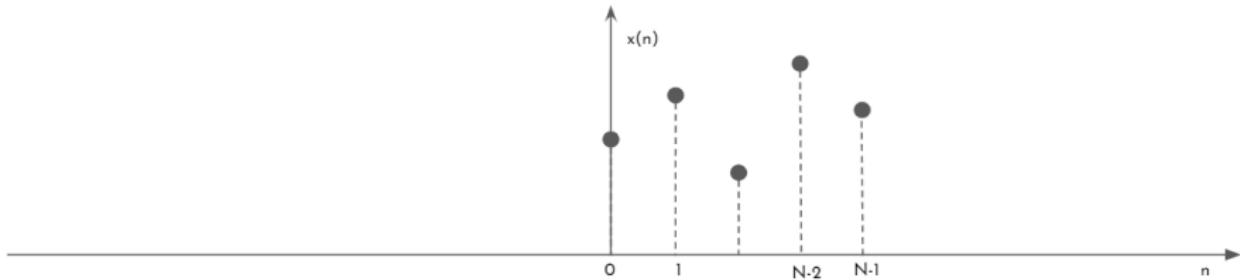
Donde a  $A(k)$  llamamos a **Transformada de  $x(n)$**  entre  $0 \leq n \leq N - 1$ :

$$A(k) = \sum_{n=0}^{N-1} x(n) \phi_k^*(n)$$

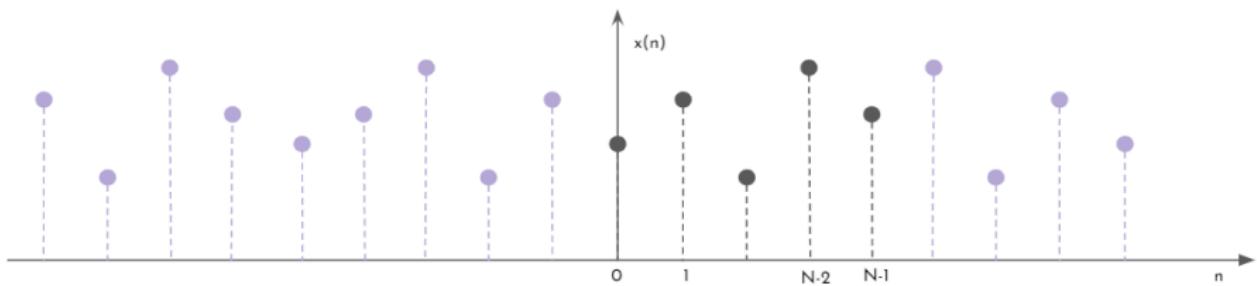
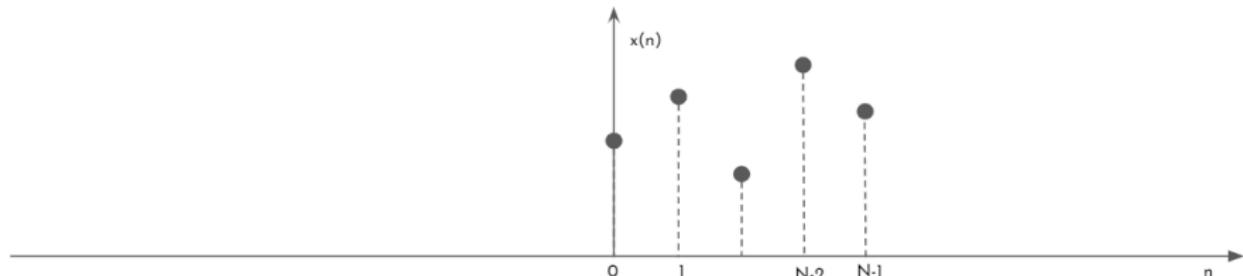
$\phi_k(n)$  son conjuntos de funciones ortonormales.

# Extensión de función $x(n)$

La función  $x(n)$  tiene que ser una función **par** si queremos escribirla como combinación lineal de funciones pares.

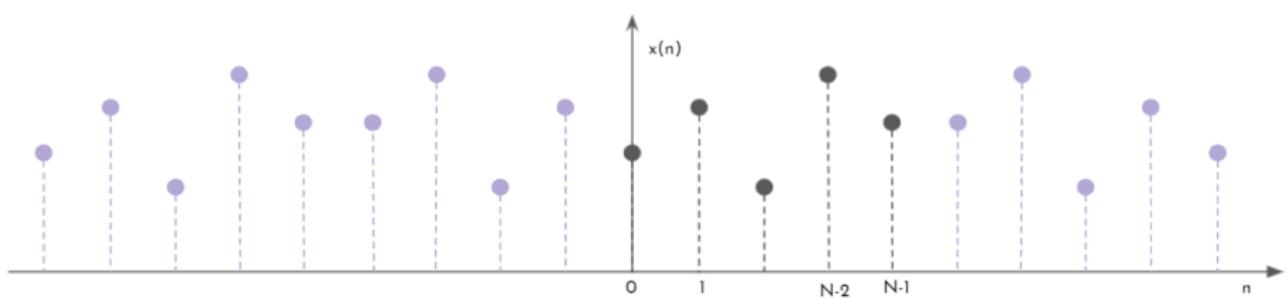
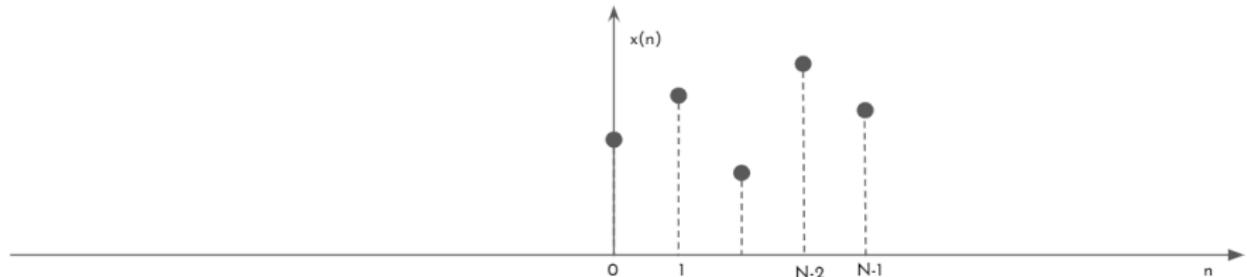


# Extensión de función $x(n)$ - DCT-I



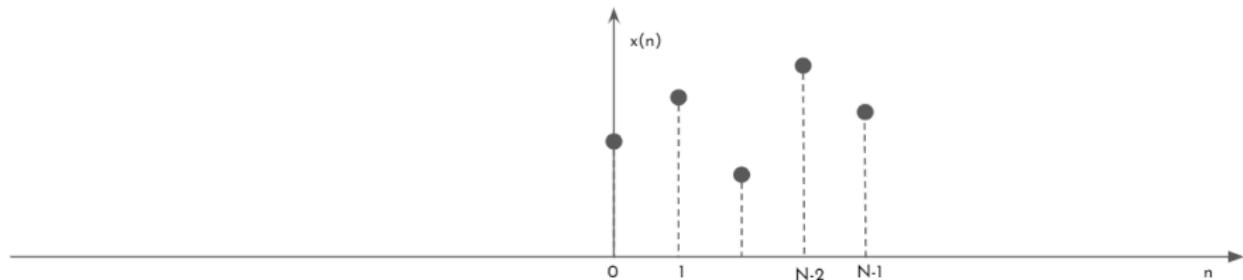
Paridad respecto de  $n = 0$  y  $n = N - 1$ .

# Extensión de función $x(n)$ - DCT-II

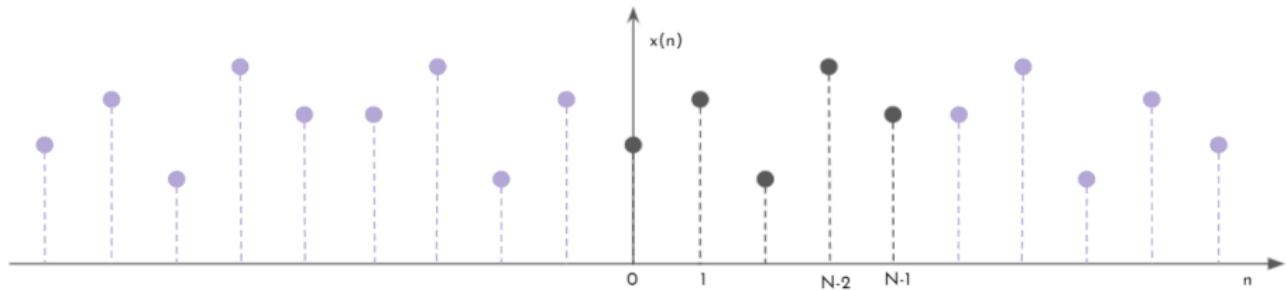


Paridad respecto de  $n = 0$ .

# Extensión de función $x(n)$ - DCT-IV



DCT-II



El período de la función  $x_2(n)$  es  $2N$ , y podemos expresar la función  $x_2(n)$  como función de  $x(n)$  de la siguiente forma:

$$x_2(n) = x(n)_{2N} + x(-n)_{2N}$$

## DCT-II

La DCT-II se define como:

$$X^{C2}(k) = 2 \sum_{n=0}^{N-1} x(n) \cos \left( \frac{\pi k(2n+1)}{2N} \right)$$

La base ortogonal se define como:

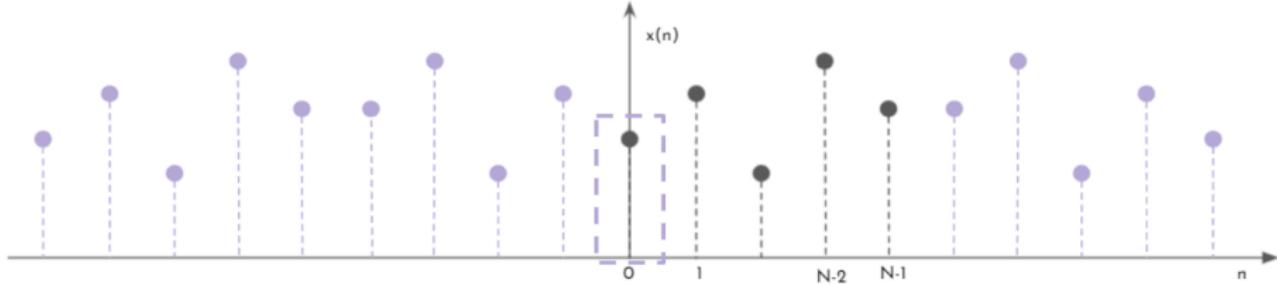
$$\left\{ \cos \left( \frac{\pi k(2n+1)}{2N} \right) \right\} \text{ con } k = 0, \dots, N-1$$

# DCT-II

La  $x(n)$  se puede escribir entre  $0 \leq n \leq N$  como:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \beta(k) X^{C2}(k) \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

$$\beta(k) = \begin{cases} \frac{1}{2} & \text{si } k = 0 \\ 1 & 1 \leq k \leq N - 1 \end{cases}$$



## DCT-II - Normalizada

Se redefine la transformada para que cumpla siguiente propiedad:

$$\sum_{n=0}^{N-1} x(n)^2 = \sum_{k=0}^{N-1} X^{C2}(k)^2$$

Se redistribuyen los factores entre la transformada y la inversa:

$$X^{C2} = \sqrt{\frac{2}{N}} \tilde{\beta}(k) \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

$$x(n) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \tilde{\beta}(k) X^{C2} \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

$$\tilde{\beta}(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } k = 0 \\ 1 & k = 1, 2, \dots, N-1 \end{cases}$$

# Definición Multidimensional

Para una función  $x(n, m)$  consiste en hacer la transformada primero de una variable y luego de la otra de la siguiente forma:

$$X_{ij} = \frac{\tilde{\beta}(i)\tilde{\beta}(j)}{\sqrt{NM}} \sum_{n=0}^{N-1} \left[ \sum_{m=0}^{M-1} x(n, m) \cos\left(\frac{\pi i(2m+1)}{2M}\right) \right] \cos\left(\frac{\pi j(2n+1)}{2N}\right)$$

# Representación Matricial

La transformada  $X$  se puede escribir como:

$$X = C_N x$$

donde  $x$  es la secuencia de valores a transformar.

Siendo que la transformada tiene una base ortogonal:

$$C_N^{-1} = C_N^T$$

Entonces la antitransformada resulta:

$$x = C_N^T X$$

Para calcular transformada bidireccional de una entrada  $x(i,j)$  de dimensión  $N \times M$ :

$$X = C_N x C_M = C_N x C_N^T$$

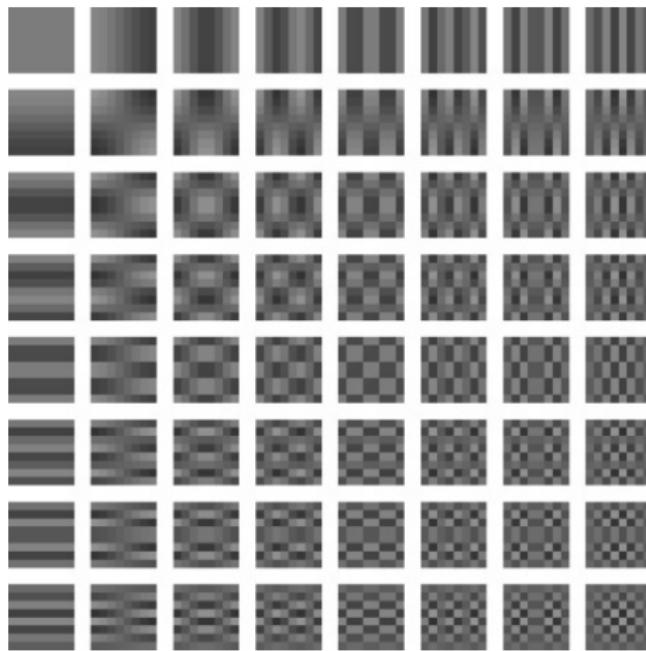
$$x = C_N^T X C_T$$

## DCT-II Unitaria - Matricial

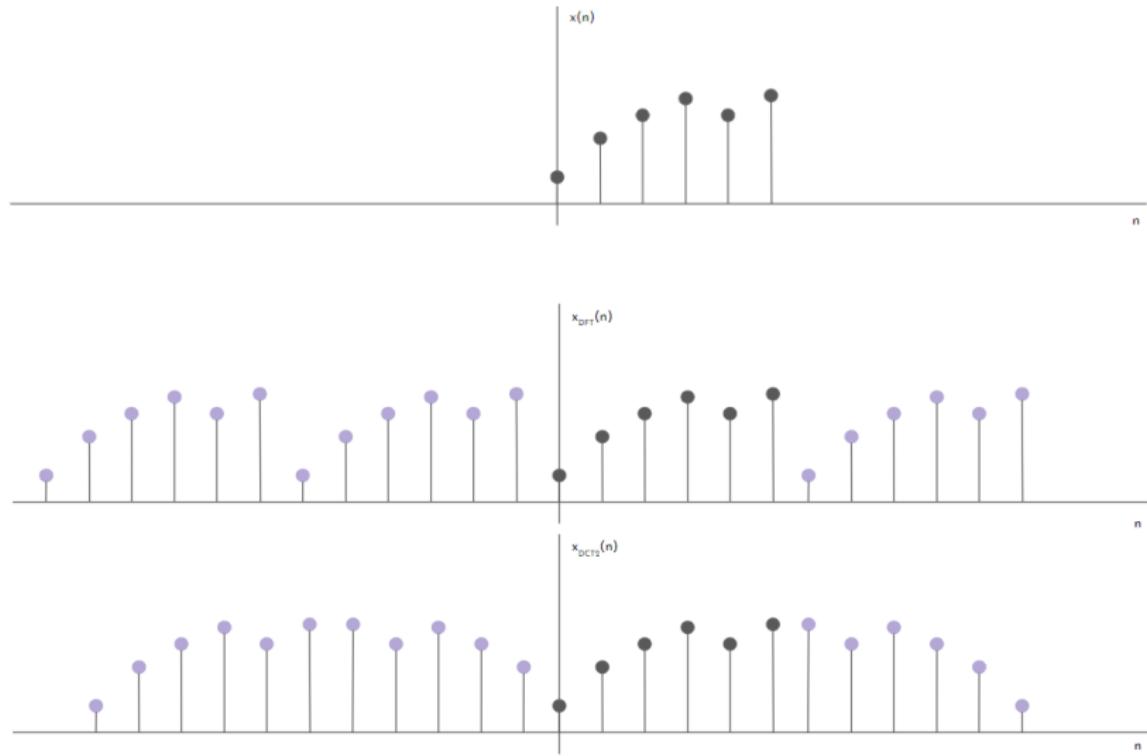
$$C_{k,n} = \begin{cases} \frac{1}{\sqrt{N}} & \text{si } k = 0, \quad 0 \leq k \leq N-1 \\ \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi(2n+1)k}{2N}\right) & \text{si } 1 \leq k \leq N, \quad 0 \leq n \leq N-1 \end{cases}$$

0,354	0,354	0,354	0,354	0,354	0,354	0,354	0,354	0,354
0,49	0,416	0,278	0,098	-0,098	-0,278	-0,416	-0,49	
0,462	0,191	-0,191	-0,462	-0,462	-0,191	0,191	0,462	
0,416	-0,098	-0,49	-0,278	0,278	0,49	0,098	-0,416	
0,354	-0,354	-0,354	0,354	0,354	-0,354	-0,354	0,354	
0,278	-0,49	0,098	0,416	-0,416	-0,098	0,49	-0,278	
0,191	-0,462	0,462	-0,191	-0,191	0,462	-0,462	0,191	
0,098	-0,278	0,416	-0,49	0,49	-0,416	0,278	-0,098	

# Matriz Base DCT

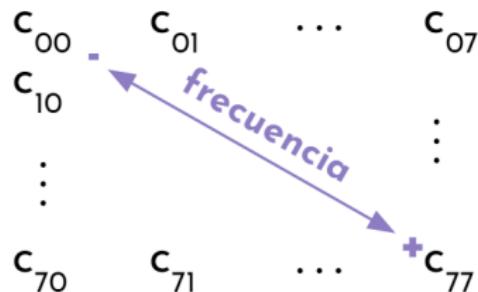


# Comportamiento en Alta Frecuencia



# Comportamiento en Alta Frecuencia

Si usamos DCT para descomponer en frecuencia la entrada  $x(i,j)$ , vemos que se cumple que los coeficientes que representan a las frecuencias más chicas tienen valores bastante más altos que aquellos de alta frecuencia.



- Los coeficientes de alta frecuencia tienen valores chicos, lo cual nos permite descartarlos.
- A esta propiedad de la DCT se la denomina *Energy Compaction*.

# Cuantización

- La DCT produce coeficientes de tipo float, estos deben ser cuantizados con cierta precisión.
- Se producirá una inevitable pérdida de información.
- Una tabla de cuantización define la precisión de cada coeficiente.
- Se realiza:

$$B_{ij} = \text{round} \left( \frac{C_{ij}}{Q_{ij}} \right)$$

Siendo  $B$  la matriz cuantizada,  $C$  la matriz DCT y  $Q$  la tabla o matriz de cuantización.

# Matrices de Cuantización

Tablas estándar de cuantización para luminancia y crominancia respectivamente:

16	11	10	16	24	40	51	61	17	18	24	47	99	99	99	99
12	12	14	19	26	58	60	55	18	21	26	66	99	99	99	99
14	13	16	24	40	57	69	56	24	26	56	99	99	99	99	99
14	17	22	29	51	87	80	62	47	66	99	99	99	99	99	99
18	22	37	56	68	109	103	77	99	99	99	99	99	99	99	99
24	35	55	64	81	104	113	92	99	99	99	99	99	99	99	99
49	64	78	87	103	121	120	101	99	99	99	99	99	99	99	99
72	92	95	98	112	100	103	99	99	99	99	99	99	99	99	99

# Cuantización

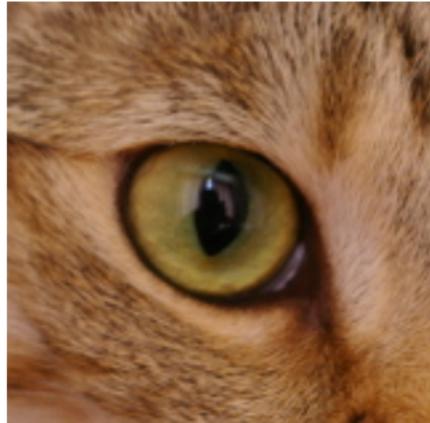


Imagen digital original.



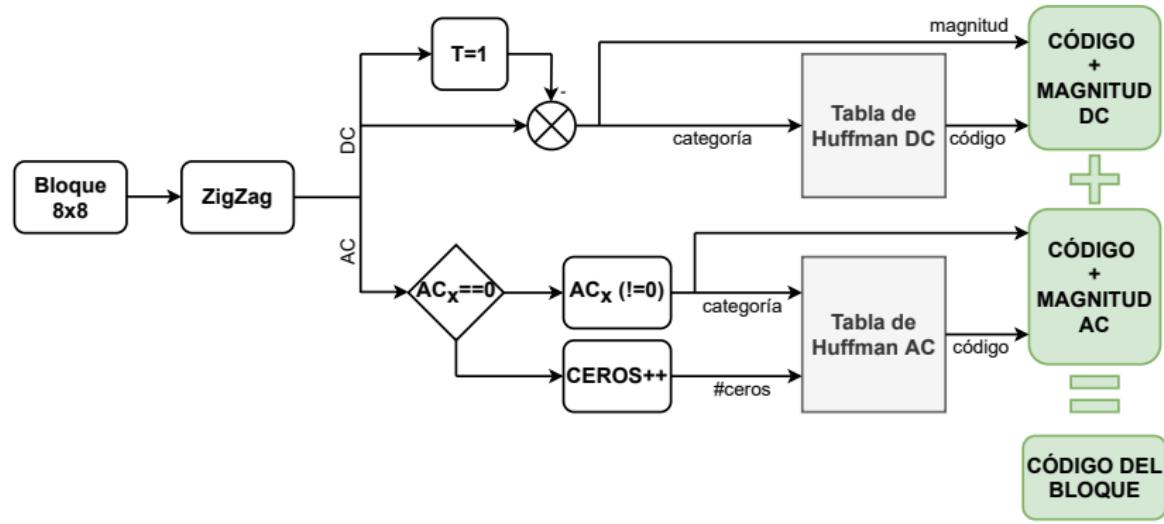
Imagen comprimida  
mediante JPEG con  
calidad 100 %.



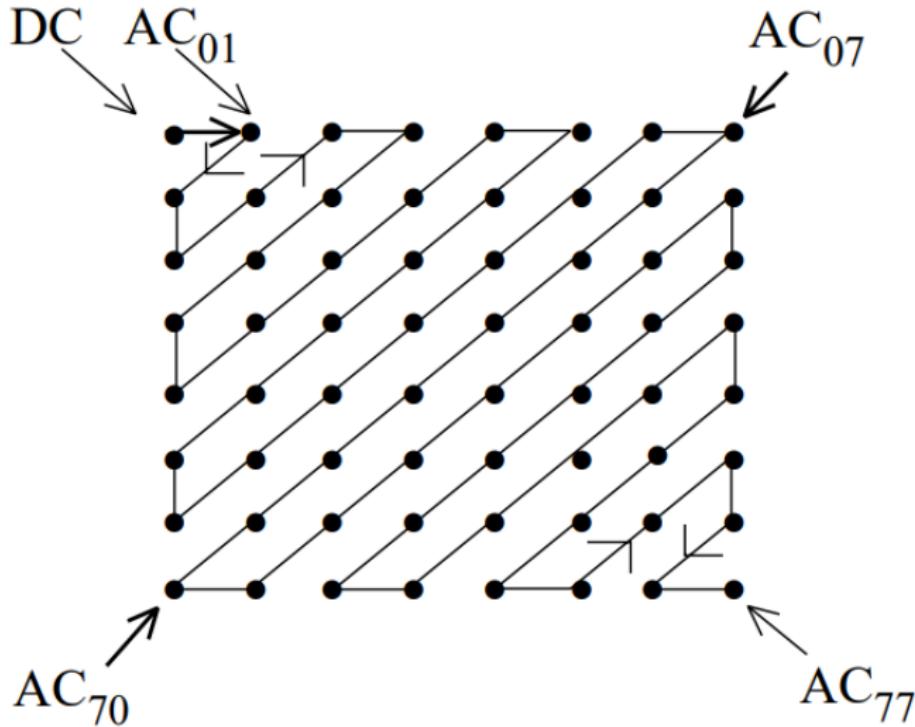
Imagen comprimida  
mediante JPEG con  
calidad 50 %.

# Codificación Huffman

- DC → Difference encoding
- AC → Running length encoding



# ZigZag scan



# Tablas de Huffman

La idea es usar códigos de longitud variable para maximizar la entropía de la información. Por ejemplo para términos de DC de la luminancia se usa la siguiente tabla:

Rango	Categoría	Código
0	0	00
-1,+1	1	010
-3,-2,+2,+3	2	011
-7,...,-4, +4,...,+7	3	100
-15,...,-8,+8,...,+15	4	101
-31,...,-16,+16,...,+31	5	110
-63,...,-32,+32,...,+63	6	1110
-127,...,-64,+64,...,+127	7	11110
-255,...,-128,+128,...,+255	8	111110
-511,...,-256,+256,...,+511	9	1111110
-1023,...,-512,+512,...,+1023	10	11111110
-2047,...,-1024,+1024,...,+2047	11	111111110

# Codificación Huffman: Ejemplo

Bloque cuantizado:

$$\begin{bmatrix} 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Vector: [11, 0, -1, -5, 0, 0, 0, 0, 0, 1, 2, -1, 0, 0, ..., 0]

Código: **100101 11000 100010 11110101 0110 000 11111111001**  
**11111111001 11111111001 1010**

- DC anterior = 6 → Codifico  $11 - 6 = 5$  (**100101**)
- Para la parte AC codifico según la cantidad de ceros y el valor no nulo más cercano. Por ejemplo 5 ceros seguidos de un 1 se codifica con **11110101**.
- 52 ceros al final → **11111111001** indica 16 ceros y **1010** indica el fin del bloque (es decir, completar con ceros lo que falta).

# Interleaving

- Data unit = Bloque 8x8
- Minimum Coded Unit = mínimo grupo de bloques intercalados

Ejemplo: para una imagen de 48x32 píxeles (6x4 bloques), se submuestrean las componentes de crominancia con un factor de 2. Por lo tanto, se deben tomar los bloques de luminancia en regiones de 2x2 y en orden izquierda-derecha y arriba-abajo. El primer MCU corresponde a:

$$[Y_1 \ Y_2 \ Y_3 \ Y_4 \ C_{B1} \ C_{R1}]$$

Componente Y					
1	2	5	6	9	10
3	4	7	8	11	12
13	14	17	18	21	22
15	16	19	20	23	24

Componente Cb		
1	2	3
4	5	6

Componente Cr		
1	2	3
4	5	6

# Problemas con la compresión JPEG

El resultado de la compresión JPEG es perceptiblemente similar a la imagen original cuando esta no tiene componentes de frecuencia espacial demasiado altos, por ejemplo cambios repentinos de color o luminancia.

Dicha condición se cumple para las fotografías del mundo físico, pero no para imágenes generadas por computadora que contienen bordes nítidos o texto rasterizado.

En este tipo de imágenes se producen “artefactos de compresión” que consisten generalmente de puntos o zonas de ruido al rededor del texto y los bordes nítidos, denominado “ruido de mosquito”.

# Problemas con la compresión JPEG: Ejemplo



**ITBA**

Imagen rasterizada del escudo del ITBA.



**ITBA**

Imagen del escudo del ITBA comprimida mediante JPEG con calidad 50 %.

## Problemas con la compresión JPEG: Ejemplo



Imagen rasterizada del escudo del ITBA (ampliada).



Imagen del escudo del ITBA comprimida mediante JPEG con calidad 50 % (ampliada).

# Bibliografía

- CCITT, "T.81 – Digital Compression and Coding of Continuous-Tone Still Images – Requirements and Guidelines", 1992. Disponible: <https://www.w3.org> [Accedido: 3/12/2021]
- A. V. Oppenheim, R. W. Schaffer and J. R. Buck, *Discrete-Time Signal Processing*, second ed. Upper Saddle River, New Jersey, US: Prentice Hall, 1999, pp 589-599.
- J. G. Proakis and D. G. Manolakis, *Signal Digital Processing*, fourth ed. Upper Saddle River, New Jersey: US: Prentice Hall, 2007, pp 496-500.
- H. M. Hang, *Handbook of Visual Communications*, first ed. New York: Academic Press, 1995.
- K. Sayood, *Introduction to Data Compression*, fifth ed. Cambridge, US: Morgan Kaufmann Publishers, 2018, pp 428-430
- K. Cabeen and P. Gent, "Image Compression and the Discrete Cosine Transform", [Online]. Disponible: <https://www.math.cuhk.edu.hk> [Accedido: 3/12/2021]
- M.R. Azimi, "Digital Image Processing Lectures 11 & 12", presentados Departamento de Ingeniería Eléctrica y Computacional, Colorado State University. [PowerPoint Slides]. Disponible: <https://www.engr.colostate.edu> [Accedido: 3/12/2021]