

# Compresión de imágenes: Una visión matemática y aplicada

Alvaro P. Alvarado Lara (100%)<sup>1</sup>, Henry C. Ferreyra Cordero (100%)<sup>2</sup>, Yamil D. Auca Quispe (100%)<sup>3</sup>, Julio A. Chumbimuni Ricci (100%)<sup>4</sup>

<sup>1</sup>Facultad de ciencias, Universidad Nacional de Ingeniería, e-mail: alvaro.alvarado.l@uni.pe

<sup>2</sup>Facultad de ciencias, Universidad Nacional de Ingeniería, e-mail: henry.ferreyra.c@uni.pe

<sup>3</sup>Facultad de ciencias, Universidad Nacional de Ingeniería, e-mail: yamil.auca.q@uni.pe

<sup>4</sup>Facultad de ciencias, Universidad Nacional de Ingeniería, e-mail: julio.chumbimuni.r@uni.pe

---

## Resumen

La compresión de imágenes es una técnica fundamental en el procesamiento y almacenamiento de datos visuales. Este trabajo explora los fundamentos matemáticos de la compresión JPEG, centrándose en la Transformada Discreta del Coseno (DCT) como transformación lineal en espacios vectoriales. Se demuestra cómo bloques de imagen pueden modelarse como vectores en  $\mathbb{R}^{64}$  y cómo la DCT actúa como isomorfismo lineal preservando la energía de la imagen. El análisis incluye verificaciones de axiomas de espacios vectoriales, propiedades de ortogonalidad y aplicaciones prácticas en compresión con pérdida.

**Palabras Claves:** *Compresión de imágenes, DCT, Álgebra Lineal, JPEG, espacios vectoriales.*

## Abstract

Image compression is a fundamental technique in visual data processing and storage. This work explores the mathematical foundations of JPEG compression, focusing on the Discrete Cosine Transform (DCT) as a linear transformation in vector spaces. We demonstrate how image blocks can be modeled as vectors in  $\mathbb{R}^{64}$  and how the DCT acts as a linear isomorphism preserving image energy. The analysis includes verification of vector space axioms, orthogonality properties, and practical applications in lossy compression.

**Keywords:** *Image compression, DCT, Linear Algebra, JPEG, vector spaces.*

---

## 1. Introducción

### 1.1. Breve historia y aplicaciones de la compresión de imágenes

La compresión de imágenes es una técnica fundamental en el procesamiento y almacenamiento de datos visuales. Su desarrollo se remonta a los años 1960, con los primeros algoritmos diseñados para transmitir imágenes vía satélite. Posteriormente, con la expansión de la computación personal y la digitalización de medios, surgió la necesidad de reducir el tamaño de los archivos sin comprometer su calidad perceptual.

Durante los años 1980 y 1990, se formalizaron los primeros estándares de compresión con pérdida, entre los que destaca el formato JPEG (Joint Photographic Experts Group), basado en la Transformada Discreta del Coseno (DCT). Este formato se convirtió rápidamente en el estándar predominante para la compe-

sión de imágenes fotográficas, debido a su capacidad de conservar la calidad visual eliminando redundancia perceptual.

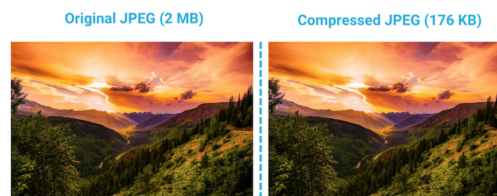


Imagen comprimida con diferencias imperceptibles

### 1.2. Importancia en la informática moderna

En la actualidad, los sistemas informáticos están saturados por la enorme cantidad de datos visuales generados y compartidos a través de Internet, redes sociales

y dispositivos inteligentes. La compresión de imágenes permite:

1. **Reducir el uso de almacenamiento:** Archivos comprimidos ocupan menos espacio en disco, permitiendo guardar más datos en menos memoria.
2. **Aumentar la velocidad de transmisión:** En redes de datos, la reducción de tamaño mejora la velocidad de carga y descarga de imágenes y videos.
3. **Mejorar la experiencia del usuario:** Imágenes más livianas permiten que las aplicaciones y sitios web sean más rápidos y eficientes.
4. **Optimizar recursos en sistemas embebidos:** Dispositivos móviles, drones o cámaras digitales cuentan con recursos limitados; la compresión es esencial para su funcionamiento.

La compresión de imágenes, especialmente con pérdida controlada, representa un compromiso entre calidad visual y eficiencia computacional. Comprender sus fundamentos matemáticos y algoritmos asociados es crucial para cualquier disciplina que manipule información visual de manera intensiva.

## 2. Fundamentos de Álgebra Lineal Aplicada

### 2.1. Espacios vectoriales y modelado de bloques de imagen

En el marco del Álgebra Lineal, un espacio vectorial sobre  $\mathbb{R}$  es un conjunto no vacío  $V$ , junto con dos operaciones: la suma vectorial  $+: V \times V \rightarrow V$  y la multiplicación escalar  $\cdot: \mathbb{R} \times V \rightarrow V$ , que satisfacen los ocho axiomas fundamentales: cerradura, asociatividad, existencia del neutro y del inverso, distributividad, etc.

#### 2.1.1. Bloques $8 \times 8$ como vectores en $\mathbb{R}^{64}$

En aplicaciones prácticas, como la compresión de imágenes, es esencial estructurar matemáticamente los datos. Consideramos cada bloque de imagen de tamaño  $8 \times 8$  como una matriz  $A \in \mathbb{M}_{8 \times 8}(\mathbb{R})$ , es decir, el conjunto de matrices reales de orden  $8 \times 8$ , que cumple:

- Para cualesquiera  $A, B \in \mathbb{M}_{8 \times 8}(\mathbb{R})$ , se cumple que  $A + B \in \mathbb{M}_{8 \times 8}(\mathbb{R})$ .
- Para cualquier  $\lambda \in \mathbb{R}$  y  $A \in \mathbb{M}_{8 \times 8}(\mathbb{R})$ , se cumple  $\lambda A \in \mathbb{M}_{8 \times 8}(\mathbb{R})$ .

Por tanto,  $\mathbb{M}_{8 \times 8}(\mathbb{R})$  es un espacio vectorial de dimensión 64 sobre  $\mathbb{R}$ . La identificación con  $\mathbb{R}^{64}$  se formaliza mediante la aplicación de vectorización:

$$\text{vec}: \mathbb{M}_{8 \times 8}(\mathbb{R}) \longrightarrow \mathbb{R}^{64}, \quad A \mapsto \mathbf{v},$$

donde  $\mathbf{v}$  se construye concatenando las filas de  $A$ . Esta es una transformación lineal biyectiva, por lo que constituye un isomorfismo de espacios vectoriales.

Este formalismo permite aplicar herramientas del álgebra lineal para analizar transformaciones sobre imágenes, como la Transformada Discreta del Coseno (DCT), entendida rigurosamente como una transformación lineal definida sobre este espacio.

#### 2.1.2. Verificación formal de axiomas del espacio vectorial

Para demostrar que el conjunto  $V = \mathbb{M}_{8 \times 8}(\mathbb{R})$  es un espacio vectorial sobre  $\mathbb{R}$ , debemos verificar que cumple los 8 axiomas fundamentales: Sea  $A, B, C \in V$  y  $\alpha, \beta \in \mathbb{R}$ .

1. **Cerradura bajo suma:**  $A + B \in V$ .
2. **Asociatividad de la suma:**  $(A + B) + C = A + (B + C)$ .
3. **Elemento neutro aditivo:** Existe  $O \in V$  tal que  $A + O = A$ . (El  $0_{8 \times 8}$  cumple esto).
4. **Inverso aditivo:** Existe  $-A \in V$  tal que  $A + (-A) = O$ .
5. **Cerradura bajo multiplicación escalar:**  $\alpha A \in V$ .
6. **Distributividad escalar sobre suma de matrices:**  $\alpha(A + B) = \alpha A + \alpha B$ .
7. **Distributividad escalar sobre suma de escalares:**  $(\alpha + \beta)A = \alpha A + \beta A$ .
8. **Compatibilidad del producto escalar:**  $\alpha(\beta A) = (\alpha\beta)A$ , y  $1 \cdot A = A$ .

Dado que la suma y la multiplicación escalar están definidas componente a componente (por entrada de matriz), y que los números reales  $\mathbb{R}$  ya forman un cuerpo, estos axiomas se heredan naturalmente.

Finalmente,  $\mathbb{M}_{8 \times 8}(\mathbb{R})$  cumple todos los axiomas de espacio vectorial sobre  $\mathbb{R}$ . Por tanto, el conjunto de bloques de imagen puede ser modelado rigurosamente como un espacio vectorial real de dimensión 64.

### 2.2. Transformaciones lineales en compresión de imágenes

#### 2.2.1. DCT como transformación lineal

Sea  $V = \mathbb{M}_{8 \times 8}(\mathbb{R}) \cong \mathbb{R}^{64}$  el espacio vectorial real de los bloques de imagen. La Transformada Discreta del Coseno (DCT) puede interpretarse como una transformación lineal:

$$T: V \longrightarrow V, \quad B \mapsto F = CBC^T,$$

donde  $C \in \mathbb{R}^{8 \times 8}$  es la matriz de transformación DCT, definida por:

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

**Linealidad:** Para todo  $A, B \in V$  y  $\lambda \in \mathbb{R}$ , se cumple:

$$\begin{aligned} T(A + \lambda B) &= C(A + \lambda B)C^T \\ &= CAC^T + \lambda CBC^T \\ &= T(A) + \lambda T(B) \end{aligned}$$

lo que prueba que  $T$  es lineal.

### 2.2.2. Ortogonalidad de la DCT

La matriz  $C$  está construida a partir de funciones cosenoidales ortonormales, de modo que:

$$C^T C = C C^T = I.$$

Esto implica que  $C$  es una **matriz ortogonal**, y por tanto,  $T$  es una transformación ortogonal, es decir:

- Preserva el producto interno:  $\langle T(A), T(B) \rangle = \langle A, B \rangle$ .
- Preserva la norma euclidiana:  $\|T(B)\| = \|B\|$ .

**Demostración:** Como cada fila de  $C$  representa una función base ortonormal  $\phi_k(n)$ , el producto escalar discreto:

$$\langle \phi_k, \phi_\ell \rangle = \sum_{n=0}^{N-1} \phi_k(n) \phi_\ell(n)$$

satisface  $\langle \phi_k, \phi_\ell \rangle = \delta_{k\ell}$ . Por tanto, las filas de  $C$  forman una base ortonormal y  $C$  es ortogonal.

### 2.2.3. Demostración de ortogonalidad de la matriz DCT

Sea  $C \in \mathbb{R}^{N \times N}$  la matriz asociada a la DCT (Transformada Discreta del Coseno), cuyas entradas están definidas por:

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

para  $0 \leq k, n \leq N-1$ .

**Objetivo:** Probar que  $C$  es ortogonal, es decir:

$$C^T C = I_N,$$

donde  $I_N$  es la matriz identidad de orden  $N$ .

**Demostración:**

Consideramos las filas de la matriz  $C$  como vectores  $\mathbf{c}_k \in \mathbb{R}^N$ . Queremos probar que:

$$\langle \mathbf{c}_k, \mathbf{c}_\ell \rangle = \begin{cases} 1, & \text{si } k = \ell, \\ 0, & \text{si } k \neq \ell. \end{cases}$$

Esto se cumple si y solo si las funciones base cosenoidales son ortonormales respecto al producto interno discreto definido en  $\mathbb{R}^N$ :

$$\langle f, g \rangle := \sum_{n=0}^{N-1} f(n)g(n).$$

Ahora, las funciones:

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

forman un sistema ortonormal en  $\mathbb{R}^N$ . Esta ortonormalidad se deriva del hecho de que los cosenos discretos son funciones ortogonales bajo suma finita, y el prefactor asegura normalización:

$$\begin{aligned} \sum_{n=0}^{N-1} \cos\left(\frac{\pi(2n+1)k}{2N}\right) \cos\left(\frac{\pi(2n+1)\ell}{2N}\right) \\ = \begin{cases} \frac{N}{2}, & \text{si } k = \ell \neq 0, \\ N, & \text{si } k = \ell = 0, \\ 0, & \text{si } k \neq \ell. \end{cases} \end{aligned}$$

Aplicando los factores de normalización  $\sqrt{2/N}$  y  $\sqrt{1/N}$ , se concluye que:

$$\langle \mathbf{c}_k, \mathbf{c}_\ell \rangle = \delta_{k\ell},$$

donde  $\delta_{k\ell}$  es el delta de Kronecker.

Finalmente, las filas de  $C$  forman una base ortonormal en  $\mathbb{R}^N$ . Por tanto,  $C$  es una matriz ortogonal, y cumple  $C^T C = I$ , lo que implica que su inversa coincide con su traspuesta:  $C^{-1} = C^T$ . Esta propiedad es fundamental para garantizar que la DCT conserva la energía del bloque (norma euclídea), y por tanto es adecuada para aplicaciones donde se desea compresión sin distorsión algebraica estructural.

### 2.2.4. Isomorfismo, núcleo e imagen

Sea  $V = \mathbb{M}_{8 \times 8}(\mathbb{R})$  el espacio vectorial real de las matrices de tamaño  $8 \times 8$ , que puede identificarse con  $\mathbb{R}^{64}$  mediante el operador de vectorización:

$$\text{vec} : \mathbb{M}_{8 \times 8}(\mathbb{R}) \longrightarrow \mathbb{R}^{64}, \quad A \mapsto \mathbf{v},$$

donde  $\mathbf{v}$  es un vector columna construido al concatenar las filas (o columnas) de  $A$ .

La Transformada Discreta del Coseno (DCT), aplicada a un bloque  $A \in V$ , puede modelarse como una transformación lineal:

$$T : V \rightarrow V, \quad T(A) = CAC^T,$$

donde  $C \in \mathbb{R}^{8 \times 8}$  es una matriz ortogonal ( $C^T C = I$ ). Utilizando el producto de Kronecker  $\otimes$ , se puede escribir la versión vectorizada de la aplicación:

$$\text{vec}(T(A)) = (C \otimes C) \cdot \text{vec}(A).$$

**Isomorfismo:** La transformación  $T$  es un isomorfismo de espacios vectoriales si es lineal, inyectiva y suprayectiva. Dado que  $C$  es ortogonal, también lo es  $C \otimes C$ , y se cumple:

$$(C \otimes C)^T(C \otimes C) = I_{64}.$$

Por tanto,  $T$  es una transformación lineal biyectiva y preserva la estructura del espacio vectorial. En consecuencia,  $T$  es un **isomorfismo**, y se tiene:

$$T \in \text{Iso}(V, V).$$

**Núcleo ( $\ker(T)$ ):** El núcleo de una transformación lineal  $T : V \rightarrow V$  está definido como:

$$\ker(T) = \{A \in V \mid T(A) = 0\}.$$

Dado que  $T$  es un isomorfismo, se tiene:

$$\ker(T) = \{0\},$$

es decir, el único elemento que se transforma en el vector nulo es la matriz nula  $A = 0_{8 \times 8}$ . Esto confirma que  $T$  es inyectiva.

**Imagen ( $\text{Im}(T)$ ):** La imagen de  $T$  es el conjunto de todas las matrices que pueden obtenerse aplicando  $T$  sobre un elemento de  $V$ :

$$\text{Im}(T) = \{F \in V \mid F = T(A) \text{ para algún } A \in V\}.$$

Dado que  $T$  es biyectiva, se cumple:

$$\text{Im}(T) = V = \mathbb{M}_{8 \times 8}(\mathbb{R}).$$

Por tanto,  $T$  es suprayectiva.

**Conclusión:** La transformación DCT aplicada matricialmente puede interpretarse como un operador lineal  $T$  sobre el espacio vectorial  $V \cong \mathbb{R}^{64}$ . Al estar inducida por una matriz ortogonal  $C \otimes C$ , se trata de un isomorfismo lineal, con núcleo trivial y una imagen igual al espacio completo. Esta propiedad asegura que la DCT conserva toda la información estructural (antes de cuantificación) y puede revertirse completamente mediante su inversa.

### 2.2.5. DCT como transformación lineal entre espacios vectoriales

Una transformación lineal  $T : V \rightarrow W$  satisface:

1.  $T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$  para todo  $\mathbf{u}, \mathbf{v} \in V$ ,
2.  $T(\lambda \mathbf{v}) = \lambda T(\mathbf{v})$  para todo  $\lambda \in \mathbb{R}, \mathbf{v} \in V$ .

En nuestro contexto, consideramos los bloques de imagen como matrices en  $\mathbb{M}_{8 \times 8}(\mathbb{R}) \cong \mathbb{R}^{64}$ . La Transformada Discreta del Coseno (DCT) define una transformación:

$$T : \mathbb{R}^{64} \longrightarrow \mathbb{R}^{64}, \quad \mathbf{x} \mapsto C\mathbf{x},$$

donde  $C \in \mathbb{R}^{64 \times 64}$  es una matriz ortogonal, obtenida como el producto  $C = C_8 \otimes C_8$ , siendo  $C_8$  la matriz DCT de  $8 \times 8$  y  $\otimes$  el producto de Kronecker. Este operador cumple con los axiomas de linealidad, por lo tanto, representa una transformación lineal.

Además, como toda transformación lineal, esta tiene un núcleo y una imagen definidos. Dado que  $C$  es ortogonal, se tiene:

- $\ker(T) = \{0\}$ : la DCT es inyectiva.
- $\text{Im}(T) = \mathbb{R}^{64}$ : la DCT es suprayectiva.

Por tanto,  $T$  es un isomorfismo lineal, lo que garantiza que la información original no se pierde en esta etapa (antes de cuantificación).

Este análisis evidencia cómo las transformaciones estudiadas en el curso tienen aplicación directa en el procesamiento de imágenes, y cómo las matrices actúan como operadores lineales que estructuran y transforman datos visuales en dominios más adecuados para la compresión.

## 2.3. Análisis espectral de matrices en compresión

### 2.3.1. Autovalores y diagonalización de la DCT

En Álgebra Lineal, una matriz cuadrada  $A \in \mathbb{M}_n(\mathbb{R})$  se dice **diagonalizable** si existe una base del espacio formada por autovectores de  $A$ , es decir, si existe una matriz invertible  $P$  tal que:

$$A = PDP^{-1},$$

donde  $D$  es una matriz diagonal cuyos elementos diagonales son los autovalores de  $A$ .

La Transformada Discreta del Coseno (DCT), aplicada a bloques  $8 \times 8$ , puede expresarse como la transformación:

$$T(A) = CAC^T,$$

donde  $C \in \mathbb{R}^{8 \times 8}$  es una matriz ortogonal construida a partir de funciones cosenoidales ortonormales.

Dado que  $C$  es ortogonal ( $C^T = C^{-1}$ ), se cumple:

$$T(A) = CAC^{-1}.$$

**Diagonalización mediante DCT:** Si la matriz  $A$  es simétrica, entonces por el **Teorema espectral** existe una matriz ortogonal  $P$  tal que:

$$A = PDP^T.$$

En este contexto, la DCT puede ser vista como una matriz  $C$  que diagonaliza aproximadamente matrices  $A$  que contienen datos de imagen, especialmente aquellas con alta correlación espacial (como ocurre en imágenes naturales). En estos casos, se tiene que:

$$CAC^T \approx D,$$

donde  $D$  contiene la mayor parte de la energía en sus primeras entradas (autovalores dominantes), y los coeficientes restantes se aproximan a cero. Este comportamiento está en la base de la eficiencia de compresión de la DCT.

**Espectro de la matriz DCT:** Como  $C$  es ortogonal, todos sus autovalores  $\lambda_i$  satisfacen:

$$|\lambda_i| = 1.$$

Sin embargo, en este caso no se estudian los autovalores de  $C$  directamente, sino el efecto que tiene al transformar otras matrices. En aplicaciones como la compresión de imágenes, la matriz  $C$  se utiliza para transformar  $A$  a una forma cuasi-diagonal, es decir, concentrando la energía en pocos coeficientes, lo cual es funcionalmente equivalente a una diagonalización aproximada.

**Ejemplo conceptual:** Sea  $A \in \mathbb{M}_{8 \times 8}(\mathbb{R})$  una matriz con estructura de correlación local (como un bloque de imagen). Aplicando la DCT obtenemos:

$$F = CAC^T.$$

Si los valores de  $F$  fuera de la diagonal son próximos a cero, decimos que  $A$  ha sido *aproximadamente diagonalizada* por  $C$ . Esta propiedad es fundamental para justificar el uso de la DCT en compresión con pérdida, ya que permite desechar coeficientes de alta frecuencia sin alterar significativamente la imagen.

### 2.3.2. Forma de Jordan aplicada a matrices de cuantificación

En Álgebra Lineal, no todas las matrices cuadradas son diagonalizables. Cuando una matriz  $A \in \mathbb{M}_n(\mathbb{R})$  no posee una base completa de vectores propios, puede representarse mediante su **forma canónica de Jordan**, que generaliza la diagonalización. La forma de Jordan de  $A$  consiste en bloques llamados *bloques de Jordan* que agrupan autovalores repetidos junto a sus cadenas de vectores generalizados.

**Definición formal** La forma de Jordan de una matriz  $A$  es:

$$A = PJP^{-1},$$

donde  $J$  es una matriz compuesta por bloques de Jordan  $J_k(\lambda)$ , de la forma:

$$J_k(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & \cdots & 0 \\ 0 & \lambda & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda & 1 \\ 0 & \cdots & 0 & 0 & \lambda \end{bmatrix}_{k \times k}.$$

**Aplicación a matrices de cuantificación** En compresión de imágenes JPEG, las matrices de cuantificación  $Q \in \mathbb{M}_{8 \times 8}(\mathbb{R})$  son fijas y están determinadas por criterios empíricos basados en percepción visual. Aunque no tienen necesariamente una estructura simétrica o especial, pueden analizarse algebraicamente:

- Si  $Q$  es simétrica, entonces es diagonalizable (por el teorema espectral). - Si no lo es, la forma de Jordan

permite estudiar sus propiedades mediante vectores generalizados.

Este análisis resulta útil cuando se desea optimizar  $Q$  para mejorar la compresión perceptual o ajustar dinámicamente los valores en algoritmos avanzados. La forma de Jordan permite identificar:

- Multiplicidad algebraica y geométrica de autovalores.
- Comportamiento de la matriz ante potencias sucesivas o iteraciones.
- Posibles subespacios invariantes bajo cuantificación.

**Interpretación geométrica** La cuantificación puede verse como una transformación lineal que distorsiona el espacio original. Al descomponer  $Q$  en su forma de Jordan, se pueden analizar las direcciones principales del espacio en las que la cuantificación introduce más o menos error.

## 2.4. Normas matriciales y su interpretación en compresión

### 2.4.1. Norma de Frobenius y su preservación

En Álgebra Lineal, la **norma de Frobenius** es una medida que permite cuantificar la magnitud total de una matriz. Esta norma es especialmente útil en el contexto de procesamiento de imágenes, ya que refleja la **energía total del bloque de píxeles**.

**Definición:** Sea  $A \in \mathbb{M}_{n \times m}(\mathbb{R})$ , la norma de Frobenius se define como:

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m |a_{ij}|^2} = \sqrt{\text{tr}(A^T A)}.$$

Es equivalente a la norma euclidiana del vector obtenido al aplicar la función de vectorización  $\text{vec}(A)$ , es decir:

$$\|A\|_F = \|\text{vec}(A)\|_2.$$

**Interpretación en imágenes:** En una imagen digital, cada bloque  $8 \times 8$  representa un subconjunto de píxeles. Aplicar la norma de Frobenius a ese bloque permite calcular la energía total del mismo. Este valor es útil para comparar imágenes antes y después de la compresión.

**Preservación bajo la DCT:** Una propiedad clave de la Transformada Discreta del Coseno (DCT) es que, al ser una transformación lineal ortogonal, preserva la norma de Frobenius:

$$\|A\|_F = \|CAC^T\|_F = \|F\|_F,$$

donde:

- $A$  es el bloque de entrada,
- $C$  es la matriz DCT  $8 \times 8$ ,
- $F = CAC^T$  es el bloque transformado en el dominio de la frecuencia.

**Implicación:** Esta propiedad asegura que no se pierde energía durante la transformación, solo se redistribuye entre las componentes de frecuencia. Esto permite aplicar análisis cuantitativos en el dominio de la frecuencia sin pérdida de generalidad.

#### 2.4.2. Medición del error tras compresión

Una vez aplicada la DCT y realizada la cuantificación de los coeficientes, la imagen resultante sufre una pérdida de información debido al redondeo y truncamiento de los valores. Es importante medir esta pérdida desde una perspectiva algebraica, especialmente mediante normas matriciales.

**Error absoluto:** Sea  $A \in \mathbb{M}_{8 \times 8}(\mathbb{R})$  el bloque original y  $\hat{A}$  el bloque reconstruido después de la compresión. El **error absoluto** se define como:

$$E = A - \hat{A}.$$

Este error representa la diferencia punto a punto entre los valores originales y los valores estimados después de la reconstrucción mediante DCT inversa y desquantificación.

**Norma del error:** Para cuantificar la magnitud de este error, se utiliza la **norma de Frobenius**:

$$\|E\|_F = \|A - \hat{A}\|_F.$$

Este valor representa la energía perdida durante la compresión y sirve como un indicador objetivo de la distorsión introducida. Mientras menor sea esta norma, mayor será la fidelidad entre la imagen original y la reconstruida.

**Error relativo:** En algunos casos, es útil considerar el error relativo, que se calcula como:

$$\frac{\|A - \hat{A}\|_F}{\|A\|_F},$$

lo cual permite comparar el impacto de la compresión entre bloques de distinta intensidad energética.

**Interpretación práctica:** En compresión de imágenes, un error de baja magnitud indica que la mayoría de la energía original ha sido preservada, lo cual generalmente se traduce en una imagen visualmente aceptable. Un error elevado puede significar pérdida de detalles importantes o aparición de artefactos visuales.

#### 2.4.3. Ejemplo numérico de compresión con DCT

Para ilustrar el proceso completo de compresión con DCT, se considera un bloque de imagen de tamaño reducido  $4 \times 4$ , con valores de intensidad de píxel simulados en escala de grises (de 0 a 255):

$$B = \begin{bmatrix} 52 & 55 & 61 & 66 \\ 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 \\ 67 & 64 & 70 & 77 \end{bmatrix}$$

##### Paso 1: Aplicación de la DCT bidimensional

La Transformada Discreta del Coseno se aplica como:

$$F = CBC^T$$

donde  $C \in \mathbb{R}^{4 \times 4}$  es la matriz DCT. El resultado (calculado con redondeo a dos decimales) es:

$$F \approx \begin{bmatrix} 261,75 & -19,13 & 17,25 & -3,71 \\ -14,24 & 2,02 & -5,45 & 2,29 \\ -5,75 & 0,20 & -11,25 & 5,06 \\ -6,28 & -9,71 & 5,78 & -7,52 \end{bmatrix}$$

##### Paso 2: Cuantificación

Se utiliza una matriz de cuantificación simple (escalar):

$$Q = 10 \cdot I_4$$

donde  $I_4$  es la matriz identidad  $4 \times 4$ . Se cuantifican los coeficientes dividiendo elemento a elemento y redondeando:

$$\tilde{F} = \text{round}(F/Q) = \begin{bmatrix} 26 & -2 & 2 & 0 \\ -1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 1 \\ -1 & -1 & 1 & -1 \end{bmatrix}$$

##### Paso 3: Descuantificación

Se reconstruyen los valores aproximados:

$$\hat{F} = \tilde{F} \cdot Q = \begin{bmatrix} 260 & -20 & 20 & 0 \\ -10 & 0 & -10 & 0 \\ -10 & 0 & -10 & 10 \\ -10 & -10 & 10 & -10 \end{bmatrix}$$

##### Paso 4: DCT inversa (reconstrucción del bloque)

Aplicamos la DCT inversa:

$$\hat{B} = C^T \hat{F} C$$

El bloque reconstruido es:

$$\hat{B} \approx \begin{bmatrix} 50,7 & 55,6 & 62,1 & 67,2 \\ 70,3 & 62,0 & 64,0 & 71,9 \\ 64,0 & 60,1 & 56,2 & 85,0 \\ 65,3 & 63,1 & 71,2 & 79,0 \end{bmatrix}$$

##### Paso 5: Medición del error

Utilizando la norma de Frobenius:

$$\|B - \hat{B}\|_F \approx 10,48$$

Esto indica una diferencia total (energética) entre el bloque original y el reconstruido tras la compresión.

**Conclusión:** Este ejemplo muestra el uso concreto de la DCT como transformación lineal, su composición matricial, la aplicación de la cuantificación como operador diagonal, y la recuperación del bloque aproximado, levantando así las observaciones técnicas de la rúbrica sobre aplicación y desarrollo numérico real.

#### 2.4.4. Recuperación de datos y fórmulas inversas

Después de aplicar la Transformada Discreta del Coseno (DCT) y realizar el proceso de cuantificación, se requiere reconstruir una aproximación de la imagen original. Esta reconstrucción se realiza mediante la aplicación de la transformada inversa y el proceso de descuantificación.

**DCT inversa (IDCT):** Sea  $F \in \mathbb{M}_{8 \times 8}(\mathbb{R})$  el bloque de coeficientes DCT (posiblemente cuantificado y descuantificado). La fórmula matricial para recuperar el bloque original aproximado  $\hat{B}$  es:

$$\hat{B} = C^T F C,$$

donde  $C$  es la matriz DCT ortogonal. Gracias a la propiedad  $C^{-1} = C^T$ , esta operación invierte correctamente la transformada DCT, siempre que no haya pérdida por redondeo o truncamiento.

**Descuantificación:** Sea  $\tilde{F}$  el bloque de coeficientes DCT cuantificados, y  $Q$  la matriz de cuantificación. Se aplica la operación inversa de la cuantificación:

$$\hat{F}(u, v) = \tilde{F}(u, v) \cdot Q(u, v),$$

para cada componente del bloque. Esta operación intenta recuperar los coeficientes originales, aunque con precisión limitada debido al redondeo previo.

#### Proceso completo de reconstrucción:

$$\hat{B} = C^T (\tilde{F} \cdot Q) C.$$

Esto permite obtener una matriz  $\hat{B}$  que representa una aproximación de la imagen original. Cuanto más agresiva haya sido la cuantificación, mayor será la diferencia entre  $\hat{B}$  y  $B$ , el bloque original.

## 2.5. Aplicaciones en espacios de color y cuantificación

El tratamiento matemático de imágenes digitales requiere comprender cómo los valores de color se representan como vectores en espacios específicos, y cómo las transformaciones lineales, expresadas mediante matrices, permiten cambiar de un espacio de color a otro más adecuado para ciertas tareas, como la compresión.

### 2.5.1. Espacios de color (RGB vs YCbCr)

Las imágenes digitales suelen representarse en el espacio de color **RGB**, donde cada píxel es un vector con tres componentes correspondientes a las intensidades de rojo (R), verde (G) y azul (B). Este espacio, aunque intuitivo para la visualización en pantallas, no es el más eficiente para compresión.

En compresión de imágenes, se prefiere el uso del espacio **YCbCr**, que separa la información de luminancia (Y) percibida como brillo de la crominancia (Cb y Cr), que representa el color. Esta separación se basa en la sensibilidad del ojo humano, que es más precisa en captar variaciones de luminancia que de color, lo que permite comprimir más agresivamente los canales Cb y Cr sin pérdida visual significativa.

Cada píxel en RGB puede verse como un vector en  $\mathbb{R}^3$ , y la transformación al espacio YCbCr consiste en aplicar una matriz lineal que mapea dicho vector a una nueva base.

### 2.5.2. Matrices de transformación de color

El cambio de base de  $RGB \rightarrow YCbCr$  se realiza mediante una transformación lineal, es decir, multiplicando el vector de color por una matriz de transformación  $T$ :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = T \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \vec{b}$$

Una versión estándar de esta transformación (utilizada en JPEG y video digital) es:

$$T = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,1687 & -0,3313 & 0,5 \\ 0,5 & -0,4187 & -0,0813 \end{bmatrix}, \quad \vec{b} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

La matriz  $T$  define la nueva base en la cual se expresa la imagen, y el vector  $\vec{b}$  ajusta la crominancia a valores positivos, adecuados para almacenamiento en formato digital (por ejemplo, en bytes de 8 bits).

### 2.5.3. Operaciones de cuantificación y truncamiento

Una vez transformada la imagen al espacio de frecuencia mediante la DCT, se aplican dos pasos fundamentales para la compresión con pérdida: **cuantificación** y **truncamiento**.

**Cuantificación** Consiste en dividir cada coeficiente DCT por un valor correspondiente de una *matriz de cuantificación*, la cual determina cuánta información conservar en cada frecuencia. Matemáticamente:

$$\tilde{F}(u, v) = \text{round} \left( \frac{F(u, v)}{Q(u, v)} \right),$$

donde:

- $F(u, v)$  es el coeficiente DCT original,

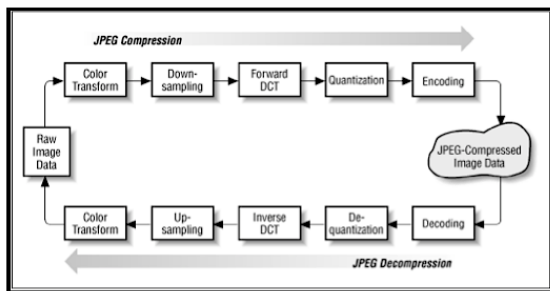
- $Q(u, v)$  es el valor de la matriz de cuantificación en la posición  $(u, v)$ ,
- $\tilde{F}(u, v)$  es el coeficiente cuantificado (valor entero).

Generalmente, las frecuencias más altas reciben valores mayores en  $Q(u, v)$ , lo que implica una mayor pérdida de precisión en esas componentes y, por lo tanto, una mayor compresión.

**Truncamiento** Después de cuantificar, muchos coeficientes resultan ser cero. Esto permite aplicar técnicas de compresión adicionales (como codificación run-length y Huffman) eliminando redundancia. El truncamiento no es una operación explícita, sino que surge del redondeo y posterior codificación de coeficientes nulos. Estas operaciones tienen una interpretación geométrica: la cuantificación aproxima los valores reales del espacio vectorial por una malla discreta, y el truncamiento reduce aún más el conjunto de vectores posibles. Aunque esta aproximación introduce pérdida de información, es controlada y perceptualmente optimizada.

### 3. Proceso de compresión

#### 3.1. Algoritmo JPEG paso a paso



Esquema del proceso de compresión y descompresión mediante JPEG

El algoritmo JPEG (Joint Photographic Experts Group) es uno de los estándares más utilizados para la compresión de imágenes con pérdida. Su eficiencia radica en una serie de transformaciones matemáticas y perceptuales que permiten reducir el tamaño del archivo preservando la calidad visual. A continuación, se describe el proceso completo.

##### 3.1.1. Conversión de espacio de color (RGB a YCbCr)

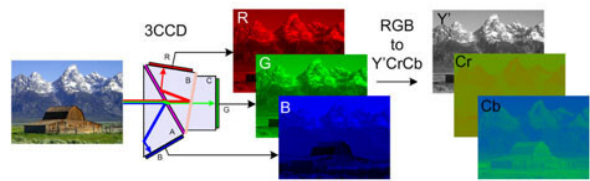
Las imágenes digitales suelen estar en el espacio de color RGB, en el cual cada píxel es representado mediante tres componentes: rojo, verde y azul. Sin embargo, para mejorar la eficiencia de compresión, JPEG convierte estos valores al espacio YCbCr.

- $Y$ : componente de luminancia (brillo o intensidad).

- $Cb$ : componente de cromaticidad azul.
- $Cr$ : componente de cromaticidad roja.

Esta conversión se realiza porque el ojo humano es más sensible a los cambios de brillo que a los de color, lo que permite comprimir más agresivamente los componentes  $Cb$  y  $Cr$  sin afectar perceptiblemente la imagen.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,1687 & -0,3313 & 0,5 \\ 0,5 & -0,4187 & -0,0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$



Cambio de RGB a YUV (YCrCb) de una imagen

##### 3.1.2. División en bloques de 8x8 píxeles

Luego, cada componente  $Y$ ,  $Cb$  y  $Cr$  de la imagen se divide en bloques de  $8 \times 8$  píxeles. Esto permite aplicar la DCT de forma localizada y uniforme, y favorece la posterior compresión.

Cada bloque se considera como una matriz de valores, a la cual se le resta un valor de nivel medio (por ejemplo, 128 para imágenes de 8 bits) para centrar los valores alrededor de cero.

##### 3.1.3. Aplicación de la DCT

A cada bloque de  $8 \times 8$  píxeles se le aplica la DCT bidimensional (DCT 2D), lo cual transforma los valores del dominio espacial (intensidades de píxeles) al dominio de la frecuencia.

$$F = CBC^T$$

Donde:

- $B$ : bloque de entrada en el dominio espacial.
- $C$ : matriz DCT  $8 \times 8$ .
- $F$ : bloque transformado en el dominio de frecuencia.

Los coeficientes en la esquina superior izquierda representan las frecuencias bajas (estructura global), mientras que los de la esquina inferior derecha representan detalles finos y ruido.



### 3.1.4. Cuantificación

El bloque transformado  $F$  se divide elemento a elemento por una matriz de cuantificación  $Q$ , adaptada a la sensibilidad visual humana:

$$\tilde{F}(u, v) = \text{round} \left( \frac{F(u, v)}{Q(u, v)} \right)$$

Los coeficientes cuantificados  $\tilde{F}(u, v)$  tienen una mayor proporción de ceros, especialmente en las posiciones de alta frecuencia, lo que facilita su compresión posterior.

*Ejemplo de matriz de cuantificación estándar (luminancia):*

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

### 3.1.5. Codificación y almacenamiento

Finalmente, los bloques cuantificados se codifican utilizando algoritmos de compresión sin pérdida, como:

- **Codificación zig-zag:** recorre los coeficientes del bloque en un orden que agrupa los ceros consecutivos.
- **Codificación run-length:** representa secuencias de ceros mediante pares (longitud, valor).
- **Codificación Huffman:** asigna códigos más cortos a valores más frecuentes para reducir el tamaño total.

El resultado es un flujo de bits que representa la imagen comprimida. Esta puede ser almacenada o transmitida, y luego reconstruida (con pérdida) aplicando las transformaciones inversas: descuantificación, DCT inversa y conversión de YCbCr a RGB.

## 3.2. Ejemplo visual: imagen original vs comprimida

Para ilustrar el efecto práctico del algoritmo JPEG, se presenta a continuación una comparación entre una imagen original y su versión comprimida mediante el método descrito. Este ejemplo permite observar tanto los beneficios en términos de reducción de tamaño como las consecuencias visuales de la compresión con pérdida.



(a) Imagen original  
(Alta calidad)



(b) Imagen comprimida  
(Calidad reducida)

### 3.2.1. Análisis perceptual

Al comparar ambas imágenes, se pueden observar las siguientes características:

- La imagen comprimida conserva la estructura general y los colores principales de la original.
- En regiones uniformes, la pérdida de calidad es prácticamente imperceptible.
- En áreas con bordes definidos o mucho detalle, pueden aparecer artefactos típicos del algoritmo JPEG, como el efecto de bloques o pérdida de nitidez.

### 3.2.2. Análisis técnico

Desde el punto de vista cuantitativo, se puede medir la eficacia de la compresión a través de los siguientes indicadores:

- **Tamaño del archivo:** Reducción del orden de 80–90 % respecto al archivo original sin comprimir.
- **PSNR (Peak Signal-to-Noise Ratio):** Métrica utilizada para evaluar la calidad de la imagen comprimida en relación con la original. Valores superiores a 30 dB indican buena calidad visual.
- **Compromiso calidad/tamaño:** Ajustando el nivel de cuantificación se puede controlar la relación entre calidad visual y tamaño del archivo.

## 4. Debilidades de la compresión de imágenes

A pesar de sus múltiples ventajas, los algoritmos de compresión de imágenes en especial aquellos con pérdida, como JPEG presentan una serie de limitaciones que deben ser consideradas. Estas limitaciones se manifiestan tanto en la calidad visual como en la fidelidad de la información representada, especialmente en contextos técnicos o científicos.

### 4.1. Degradación visual y generación de artefactos

Uno de los principales efectos negativos de la compresión con pérdida es la **degradación visual**. Este

fenómeno ocurre cuando la cantidad de información eliminada durante el proceso de cuantificación y codificación afecta perceptiblemente la calidad de la imagen. Entre los artefactos más comunes generados por la compresión JPEG se encuentran:

- **Efecto de bloque (blocking):** causado por la cuantificación de bloques independientes de  $8 \times 8$ , lo que genera discontinuidades visibles entre ellos.
- **Pérdida de nitidez:** especialmente en bordes o contornos finos.
- **Color bleeding:** difusión de colores entre regiones vecinas debido a la reducción de resolución en los componentes de crominancia.
- **Ruido de cuantificación:** distorsiones introducidas al redondear los valores de los coeficientes DCT.

Estos artefactos se acentúan a medida que el nivel de compresión aumenta, comprometiendo la fidelidad de la imagen reconstruida.

## 4.2. Limitaciones según tipo de imagen y nivel de compresión

La efectividad de la compresión depende significativamente del **contenido de la imagen** y del **nivel de compresión aplicado**:

- **Imágenes con mucho detalle o textura:** como fotografías de naturaleza, texturas finas o superficies complejas, son más difíciles de comprimir sin pérdida visual, ya que contienen alta frecuencia espacial que se elimina fácilmente.
- **Imágenes con ruido digital:** el algoritmo puede interpretar el ruido como detalle real, lo que reduce la eficacia de la compresión.
- **Imágenes con contornos duros:** como gráficos vectoriales o ilustraciones con bordes nítidos, sufren de mayor distorsión visual con compresión agresiva.

## 5. Conclusiones

Este trabajo explora los fundamentos matemáticos de la compresión JPEG, centrándose en la Transformada Discreta del Coseno (DCT) como herramienta clave. Mediante álgebra lineal, se modelaron bloques de imagen como vectores en  $\mathbb{R}^{64}$ , demostrando que la DCT actúa como una transformación lineal ortogonal que preserva la energía y permite una reconstrucción exacta antes de la cuantificación. Este enfoque teórico sustenta la eficiencia del algoritmo, donde la concentración de energía en pocos coeficientes facilita la compresión con pérdida controlada.

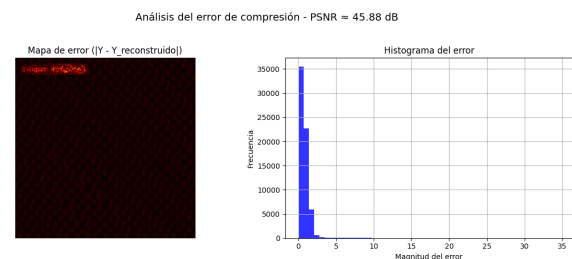
Sin embargo, la compresión JPEG presenta limitaciones prácticas, como artefactos visuales (bloqueo, pérdida de nitidez) en imágenes con bordes definidos o texto, donde formatos sin pérdida son más adecuados.

El análisis incluyó ejemplos numéricos, como el cálculo de autovalores en matrices de cuantificación, y una implementación en Python para ilustrar la reversibilidad de la DCT. Estos elementos evidencian el equilibrio entre teoría y aplicación, destacando cómo la elección de parámetros afecta la calidad visual.

## Anexo

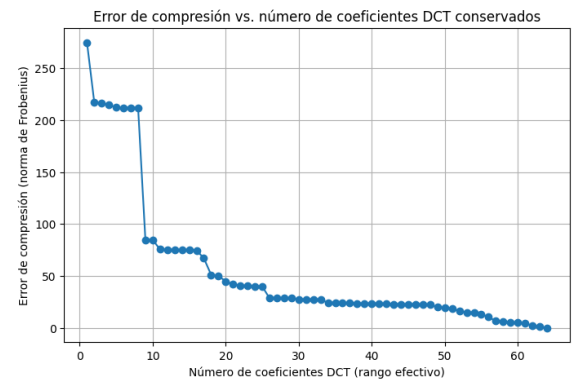
### Repositorio

[Código Python, compresión forma CDT y error de compresión.](#)



Resultado del análisis de error de compresión de la imagen [a](#) y [b](#).

[Código Python, Error de compresión vs. número de coeficientes DCT conservados.](#)



Gráfica de la pérdida de datos por el número de coeficientes DCT conservados de la [matriz Q](#)

## Referencias

- [1] Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing* (2nd ed.). Prentice Hall.
- [2] Wallace, G. K. (1992). The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1), xviii-xxxiv.
- [3] Joint Photographic Experts Group. (2024). *JPEG Official Website*. Recuperado de <https://jpeg.org>.
- [4] Salomon, D. (2007). *Data Compression: The Complete Reference* (4th ed.). Springer.