Situación problema del curso de Métodos Matemáticos para la Física

Luis Santiago Vargas Ochoa, Facundo Bautista Barbera

9 de junio de 2025

Resumen

La compresión de imágenes es una técnica para reducir el tamaño de los archivos gráficos sin perder información crítica, facilitando su almacenamiento y transmisión. En este artículo se explora el uso de la serie (transformada) de Fourier como herramienta de compresión de imágenes. Se presenta brevemente la teoría de la Transformada de Fourier y cómo suprime componentes de alta frecuencia para lograr compresión con pérdida controlada de calidad. Además, se incluye y analiza un código en *Python* que implementa la compresión de una imagen mediante la transformación de Fourier, explicando de forma clara cada paso. Los resultados demuestran que es posible reducir significativamente la cantidad de datos de una imagen manteniendo la mayor parte de su información visual relevante.

1. Introducción

Hoy en día, enormes volúmenes de imágenes son generados y compartidos a diario, desde fotografías personales hasta imágenes satelitales o médicas. La **compresión de imágenes** es muy util para disminuir el espacio de almacenamiento y el ancho de banda necesario al manejar estos datos (Sayood, 2018). Un archivo de imagen sin comprimir (por ejemplo, formato `BMP`) almacena información píxel a píxel y suele ser de gran tamaño; en cambio, utilizando métodos de compresión adecuados es posible reducir drásticamente el tamaño del archivo manteniendo una calidad visual aceptable. Por ejemplo, el formato con pérdida JPEG logra tasas de compresión cercanas al 90 % en comparación con BMP, sin diferencias apreciables a la vista en niveles de compresión moderados (Benítez López, 2016). Esto se logra aprovechando la *redundancia* de la información visual y las limitaciones de percepción del ojo humano (Shannon, 1948).

Existen dos tipos generales de compresión: *sin pérdida*, donde la imagen original se puede recuperar exactamente, y *con pérdida*, donde se permite una leve degradación a cambio de mayores tasas de compresión (Sayood, 2018). En compresión con pérdida, una estrategia muy utilizada es la **codificación por transformación**: consiste en transformar la imagen a un dominio donde su información esté mejor organizada para la compresión (Gonzalez & Woods, 2018). En particular, las transformadas de la familia de Fourier han sido fundamentales en métodos de compresión con pérdida (Wallace, 1992). La transformada de coseno discreta (DCT), estrechamente relacionada con la transformada de Fourier, es el núcleo del estándar JPEG (Wallace, 1992; Benítez López, 2016; Ahmed et al., 1974). En este artículo nos centraremos en la idea básica de la *compresión mediante series de Fourier*, mostrando cómo representar y reducir una imagen en el dominio frecuencial para obtener compresión.

2. Fundamento teórico

La **Transformada de Fourier** descompone una señal (en este caso, una imagen) en una suma de funciones seno y coseno de distintas frecuencias (Bracewell, 1999). De forma intuitiva, una imagen en escala de grises puede considerarse como una función $f(x, y)$ que asigna a cada coordenada espacial (x, y) un valor de intensidad. Mediante la transformada de Fourier discreta (DFT) bidimensional es posible obtener un conjunto de *coeficientes* $F(u, v)$ que representan la contribución de las distintas frecuencias (ondeletas sinusoidales) presentes en la imagen (Gonzalez & Woods, 2018). La DFT para una imagen digital de tamaño $N \times M$ viene dada por la expresión:

$$F(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y)\, e^{-j\,2\pi\left(\frac{u\,x}{N} + \frac{v\,y}{M}\right)}, \tag{1}$$

donde $0 \leq u < N$ y $0 \leq v < M$. Cada coeficiente complejo $F(u, v)$ indica la amplitud (y fase) de la componente sinusoidal de frecuencia (u, v) en la imagen. El término $F(0, 0)$ corresponde a la *componente DC* (frecuencia cero), que refleja el brillo promedio de la imagen, mientras que coeficientes con valores de u, v más grandes corresponden a *frecuencias más altas* que representan variaciones rápidas o detalles finos en la imagen (Oppenheim & Schafer, 1989). La transformada inversa (IDFT) permite reconstruir $f(x, y)$ a partir de todos los coeficientes $F(u, v)$, garantizando que no haya pérdida de información si se utilizan todos los componentes.

Una propiedad importante en las imágenes naturales es que la mayor parte de la energía suele concentrarse en los coeficientes de baja frecuencia, mientras que muchos coeficientes de alta frecuencia son de magnitud muy pequeña (Gonzalez & Woods, 2018; Bracewell, 1999). En otras palabras, las variaciones lentas o globales (baja frecuencia) contienen la información visual más significativa (formas y tonalidades generales), mientras que las variaciones rápidas (alta frecuencia) aportan detalles finos y ruido. De hecho, al analizar el espectro de Fourier de una imagen típica se observa que la magnitud de los coeficientes tiende a decrecer a medida que aumenta la frecuencia (es decir, conforme uno se aleja del centro del espectro) (Gonzalez & Woods, 2018). Esta característica abre la puerta a la compresión: si muchos coeficientes de alta frecuencia son casi cero o poco significativos, podemos **descartar** (poner a cero) esos coeficientes con mínimo impacto visible en la imagen reconstruida.

El método de compresión por transformada de Fourier consiste precisamente en lo anterior: transformar la imagen al dominio frecuencial, eliminar (o cuantizar fuertemente) los coeficientes de alta frecuencia considerados prescindibles, y luego aplicar la transformada inversa para obtener una imagen aproximada de la original. Al eliminar información de alta frecuencia, la imagen resultante pierde algo de nitidez en los detalles finos, pero mantiene su estructura básica y contenido visual esencial. Este principio es usado en JPEG mediante la

DCT, donde tras la transformación se descartan o cuantizan los coeficientes menos importantes antes de la codificación entrópica (Wallace, 1992; Ahmed et al., 1974). En resumen, la transformada de Fourier proporciona un *dominio de representación* en el cual la compresión es posible al *omitir las componentes de frecuencia a las que el ojo humano es menos sensible* (Gonzalez & Woods, 2018).

Cabe destacar que el cálculo directo de la DFT bidimensional según la Ecuación (1) es computacionalmente costoso $\mathcal{O}(N^2)$, pero existe el algoritmo de *Transformada Rápida de Fourier* (FFT) que reduce la complejidad a $\mathcal{O}(N \log N)$ (Cooley & Tukey, 1965). Gracias a la FFT, es práctico aplicar transformadas de Fourier incluso a imágenes grandes, lo cual es crucial para los algoritmos de compresión basados en transformada.

3. Análisis del código

Para ilustrar de manera práctica la compresión de imágenes mediante series de Fourier, a continuación se presenta un código escrito en Python. Este programa lee una imagen en escala de grises, calcula su transformada de Fourier 2D, elimina las componentes de alta frecuencia (conservando solo un rango de frecuencias bajas), y luego reconstruye la imagen usando la transformada inversa. Finalmente se comparan la imagen original y la comprimida. Cada parte del código está comentada para facilitar su comprensión:

En el código anterior, los pasos 1 y 2 realizan la lectura de la imagen y el cálculo de su espectro de Fourier usando `NumPy`. La función `fftshift` se utiliza para centrar el espectro (llevando la frecuencia cero al centro de la matriz F), lo cual es una conveniencia para definir máscaras de filtrado en coordenadas centradas. En el paso 3, se define un parámetro `half_size` que controla el tamaño del bloque de frecuencias bajas que se conservarán. Todas las componentes fuera de ese cuadrado central en F_shift se eliminan asignándoles cero en la matriz `F_low`. Por ejemplo, si `imagen` tiene tamaño 512×512 píxeles y `half_size = 30`, entonces se preservan únicamente las frecuencias en un cuadro de 60×60 alrededor del centro (frecuencias bajas), lo que corresponde a 3600 coeficientes de un total de 262144, es decir, aproximadamente un 1,37 % de los coeficientes. Este drástico filtrado produce una compresión significativa de datos, a costa de perder detalles de alta frecuencia (bordes, texturas finas).

El paso 4 aplica la transformada inversa para obtener la imagen comprimida en el dominio espacial. Se utiliza `ifftshift` para revertir el corrimiento del espectro y luego `ifft2` para la transformada inversa. Dado que la imagen original es real, tomamos la parte real del resultado (eventuales partes imaginarias surgen solo por errores numéricos de redondeo). También se convierten los valores a enteros de 8 bits (0-255) para poder manejar la imagen resultante como imagen estándar.

Finalmente, en el paso 5 se muestran lado a lado la imagen original y la imagen comprimida usando `matplotlib`. Al visualizar los resultados, se apreciará que la imagen comprimida sigue mostrando los elementos principales de la escena (formas, regiones de color/gris), pero los detalles finos (especialmente bordes nítidos o texturas) aparecen atenuados o borrosos. Esto concuerda con la expectativa: al descartar las altas frecuencias, esencialmente se ha aplicado un filtro pasa-bajas fuerte a la imagen, reteniendo solo las componentes de variación lenta.

4. Conclusión

En este trabajo se ha revisado y aplicado el uso de la transformada de Fourier como técnica de compresión de imágenes, integrando teoría y práctica. Mediante la representación de una imagen en el dominio de la frecuencia, es posible identificar y eliminar información redundante o de menor relevancia visual (particularmente las componentes de alta frecuencia), logrando una reducción significativa del tamaño de los datos. La implementación práctica con código demostró cómo al conservar solo una pequeña fracción de los coeficientes de Fourier de una imagen, ésta puede reconstruirse manteniendo su contenido esencial. Se observó que la compresión por series de Fourier tiende a suavizar la imagen resultante, pues elimina detalles finos, pero aun así mantiene las características globales (un efecto análogo al que aprovecha el formato JPEG al utilizar la transformada coseno).

En síntesis, la técnica de compresión mediante Fourier ilustra los principios fundamentales de la compresión con pérdida: aprovechar la redundancia y la poca importancia de ciertas componentes para reducir datos. Si bien en la práctica los estándares modernos emplean variantes optimizadas (como la DCT en JPEG o las ondículas en JPEG2000) en lugar de la DFT pura, el concepto subyacente es el mismo. Comprender la compresión por medio de series de Fourier proporciona una base sólida para adentrarse en métodos más avanzados de compresión de imágenes, y resalta el poderoso vínculo entre las matemáticas de las series de Fourier y las aplicaciones de procesamiento digital de imágenes.

Referencias

Ahmed, N., Natarajan, T., & Rao, K. R. (1974). *Discrete Cosine Transform. IEEE Transactions on Computers*, 23(1), 90-93.

Benítez López, J. (2016). *El sistema de compresión JPEG. Un pequeño paseo por la transformada discreta de Fourier y la transformada coseno. La Gaceta de la RSME*, 19(1), 25-45.

Bracewell, R. N. (1999). *The Fourier Transform and Its Applications* (3rd ed.). New York: McGraw-Hill.

Cooley, J. W., & Tukey, J. W. (1965). *An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation*, 19(90), 297-301.

Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson.

Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall.

Oppenheim, A. V., & Schafer, R. W. (1989). *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice Hall.

Sayood, K. (2018). *Introduction to Data Compression* (5th ed.). Morgan Kaufmann.

Shannon, C. E. (1948). *A mathematical theory of communication. Bell System Technical Journal*, 27(3), 379-423.

Wallace, G. K. (1992). *The JPEG still picture compression standard. Communications of the ACM*, 34(4), 30-44.