

Mejoramiento de Imágenes con Transformada de Fourier

Image Enhancement Using Fourier Transform

 Luis Santiago Laime Chamby^{1,*}, Miguel Angel Torrico Silvestre², and Lian Pablo Saravia Irahola³

¹Luis Santiago Laime Chamby

²Miguel Angel Torrico Silvestre

³Lian Saravia

Resumen

El proyecto aborda la mejora de imágenes deterioradas con interferencias y errores mediante el uso de transformadas de Fourier (FFT). Se aplicaron distintos filtros para reducir las interferencias más comunes, esto se hizo en un programa de python donde se comprobó la eficacia de estos.

Palabras clave: Transformada de Fourier, filtros, interferencia, errores, patrones, detalle visual.

Abstract

The project focuses on enhancing degraded images affected by interference and errors using Fourier Transforms (FFT). Various filters were applied to reduce the most common interferences, this was done in a Python program, where their effectiveness was tested.

Keywords: Fourier Transform, filters, interference, errors, patterns, visual detail

Introducción

Las series viejas muestran distintos problemas con la calidad de imágenes debido a las limitaciones tecnológicas de ese entonces. Entre los mayores problemas están la estática, interferencias y ruido, lo que provocaba distorsiones en la transmisión de imágenes. Estos defectos pueden hacerse menores con un tratamiento digital de imágenes usando la transformada de Fourier, que permite analizar y mover información en el rango de frecuencia. Este trabajo estudia el uso de la FFT para sacar errores en fotos antiguas de TV. Se usan diferentes filtros para hacer que la imagen se vea mejor, pero cuidando los detalles importantes de la foto.

Materiales y Métodos

Serie de Fourier 2D

Para poder tratar imágenes tendremos que utilizar la serie de Fourier en dos dimensiones, que tiene la siguiente forma:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cdot e^{-j2\pi(ux+vy)} dx dy \quad (1)$$

Nuestras imágenes se tomarán como funciones bidimensionales discretas, por lo que tendremos que usar la transformada discreta que usa sumatorias

$$F(u, v) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (2)$$

con m y n siendo el número de columnas y filas de píxeles en la imagen. Para reducir el tiempo de cálculo, se suele utilizar la transformada rápida de Fourier. También es importante calcular el espectro, pero para visualizarlo tendremos que usar la función

$$D(u, v) = c \cdot \log [|F(u, v)|] \quad (3)$$

Siendo c una constante de escalado. El espectro nos servirá para analizar la imagen y los posibles filtros que se podrán utilizar.

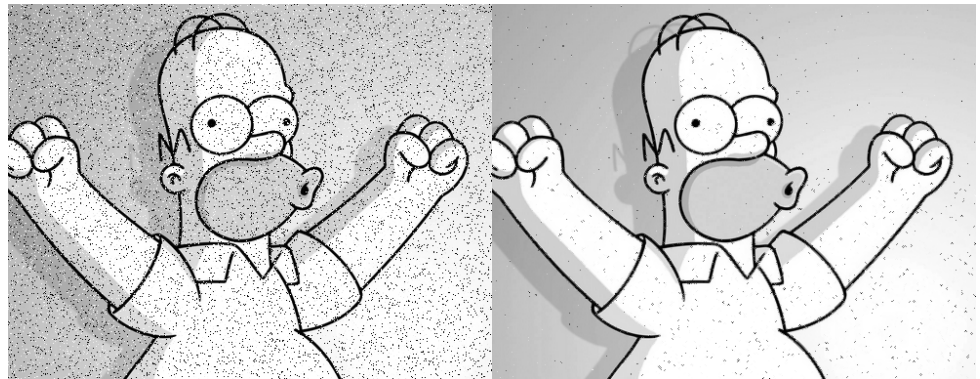
Uso de filtros

La forma básica en la que se manejarán las imágenes será que primero se aplicará la transformada de Fourier, luego se aplicará un filtro dependiendo del tipo de problema o distorsión que tiene la imagen, y luego se utilizará la transformada inversa para obtener la imagen mejorada. Existen diferentes filtros predeterminados para utilizar, pero en varias ocasiones se tendrá que crear un filtro personalizado para distorsiones como las de barras en la imagen. Entre los principales filtros están:

- Filtro Gaussiano: Es un filtro de suavizado que utiliza una función gaussiana para eliminar el ruido de una imagen
- Filtro Negativo: Invierte los colores de una imagen, transformando cada valor de píxel en su complemento
- Filtro Mediana: Filtro no lineal que reemplaza cada píxel de la imagen con el valor de la mediana de los píxeles cercanos. Es eficaz para eliminar el ruido "sal y pimienta")
- Filtro Media: Este filtro reemplaza el valor de cada píxel con el valor promedio de los píxeles cercanos
- Filtro Máximo: En este filtro, cada píxel de la imagen se reemplaza por el valor máximo de los píxeles vecinos
- Filtro Mínimo: Cada píxel es reemplazado por el valor mínimo de los píxeles en su vecindad

Tipos de distorsiones

- Ruido de sal y pimienta: Presenta pequeños puntos blancos y negros distribuidos al azar, lo que da una apariencia granulada.



- Rayas horizontales o verticales: Líneas que aparecen en toda la longitud de la imagen, pueden ser oscuras o claras.



- Deterioro de contraste: Pérdida de contraste en ciertas áreas, lo que hace que partes de la imagen se vean más apagadas o descoloridas.



- Desalineación de imagen: La imagen está desplazada o cortada, lo que puede resultar en la pérdida de información importante en los bordes.
- Pérdida de detalles: Zonas donde los detalles son difíciles de distinguir, lo que puede hacer que la imagen se vea borrosa o poco clara.

Código en Python

Respecto al código de Python desarrollado para la mejora de la calidad de las imágenes, se necesitó la importación de librerías tales como:

- **Numpy** (`numpy as np`): cálculos numéricos.
- **OpenCV** (`cv2`): procesamiento de imágenes en escala de grises.
- **Matplotlib** (`matplotlib.pyplot`): visualización de imágenes y resultados.
- **SciPy** (`scipy.signal`): convoluciones en 2D para el filtrado.
- **Etcétera.**

En el caso del ruido de sal y pimienta (granulado), primero se transforman aleatoriamente píxeles de la imagen original en blanco (sal) y negro (pimienta), obteniendo como resultado una imagen con la respectiva interferencia. Todo este proceso se realiza después de una pequeña verificación sobre la ubicación de la imagen.

El código desarrollado genera la imagen en escala de grises y añade un 5% de ruido de este tipo mediante la función antes mencionada.

Posterior a esto, se crea un **Kernel** (matriz pequeña) que actúa como un suavizador de la imagen mediante una convolución en 2D.

Finalmente, en el código se emplea el **filtro de Wiener** (4) para mejorar la imagen afectada, complementándolo con un **filtro de mediana de 7x7** para reducir los residuos de ruido restantes.

$$\hat{I} = I_f + \frac{\sigma_r^2}{\sigma_r^2 + \sigma_n^2} (I - I_f) \quad (4)$$

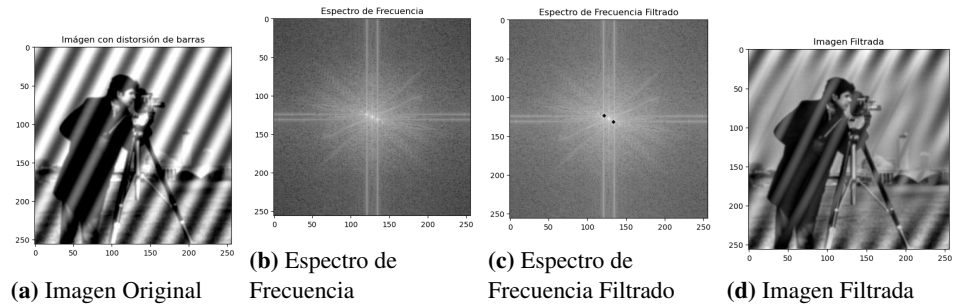
Donde:

- \hat{I} es la imagen filtrada.
- I_f es la imagen suavizada (resultado de la convolución con el kernel).
- σ_r^2 es la varianza del ruido en la imagen.
- σ_n^2 es una estimación de la varianza del ruido agregada manualmente.
- I es la imagen original con ruido.

Resultados y Discusión

Se obtuvo buenos resultados al aplicar filtros a dos imágenes con diferentes distorsiones. Aunque no se logró restaurarlas por completo, sí hubo una diferencia notoria en el antes y el después.

Para la imagen con barras, luego de realizar la transformada de fourier y analizar el espectro de frecuencia se utilizó un filtro personalizado. Este consiste en dos círculos que irán en los puntos de luz que se identificaron.



En el caso de la imagen de Sal y Pimienta, obtenemos dos imágenes:

Imagen con Ruido de Sal y Pimienta



Imagen Filtrada con Wiener + Mediana (7x7)



La primera imagen nos muestra una gran cantidad de píxeles afectados por el ruido, lo cual dificulta la visibilidad de la imagen. tras el uso de los filtros, se observa una correcta aplicación y mejoramiento para la erradicación del ruido en nuestra imagen.

realizando una comparación de ambas imágenes, se logró una restauración de la imagen sin perder de manera significativa los detalles importantes

Conclusiones

En este proyecto, se implementó el uso de las transformadas de Fourier para el mejoramiento de imágenes, usando Python e implementando técnicas de filtrado con el objetivo de reducir los errores de "Sal y Pimienta" y "Rayas horizontales y verticales". A través del dominio de frecuencia logramos disminuir la presencia de estos errores, mejorando en medida la calidad de las imágenes. Aunque la mejora no fue perfecta, los resultados mostraron una notable reducción del ruido, por lo que confirmamos que el método de filtrado funciona correctamente. Usando los filtros correctos, se nos permitió eliminar patrones indeseados en la imagen sin modificar los detalles de la imagen original.

Bibliografía

- DSpace. (s. f.). Recuperado de <https://riunet.upv.es/entities/publication/05981df9-7649-4246-8e94-fff953b953da>
- Del Carmen Tutora, Q. V. L. (2020, 9 octubre). Repositorio Institucional, UNAN-León: Procesamiento y análisis de forma de imágenes digitales. Recuperado de <http://riul.unanleon.edu.ni:8080/jspui/handle/123456789/7723>
- Turmero, P. (2021, 12 marzo). Realzado en el dominio de la frecuencia (página 2). Monografias.com. Recuperado de <https://www.monografias.com/trabajos106/realzado-dominio-frecuencia/realzado-dominio-frecuencia2>
- Jorge Miranda Redes Neuronales. (2019, 24 octubre). TRANSFORMADA DE FOURIER EN 2D DE IMÁGENES DIGITALES EN Python & OpenCV [Vídeo]. YouTube. Recuperado de <https://www.youtube.com/watch?v=mLFHsi0Zjuk>
- Mekler, G. C. M., & Director, D.-. (2024, 25 abril). Análisis no lineal de series de tiempo e imágenes basado en fases de Fourier. Recuperado de <http://riaa.uaem>

.mx/handle/20.500.12055/4535

- DSpace. (s. f.-b). Recuperado de <https://riunet.upv.es/handle/10251/68301>