Universidad Nacional Autónoma de México

Facultad de Ciencias

Proceso Digital de Imágenes

Práctica 04
”Transformada discreta de Fourier en 2D”
y
”Filtrado en el espacio de la frecuencia”.

Autor:

Delgado Díaz Hermes Alberto
319258613

5 de noviembre de 2025

1. Objetivo

- Calcular la transformada discreta directa e inversa de Fourier de una imagen manipulando sus componentes.
- Realizar operaciones de suavizado y de reducción de ruido en imágenes utilizando filtros en frecuencia.
- Realizar operaciones de detección de bordes en imágenes, tanto limpias como ruidosas, utilizando filtros en frecuencia.

2. Introducción

La contribución Joseph Fourier(1822) establece que cualquier función que se repite de manera periódica puede ser expresada como la suma de senos y/o cosenos de diferentes frecuencias cada una multiplicada por un coeficiente diferente. A lo anterior se le conoce como "series de Fourier".

- Transformada discreta de Fourier en 2D.

  Las funciones pares que no son periódicas(pero cuya área bajo la curva es finita) pueden ser expresadas como la integral de senos y/o cosenos multiplicados por una función ponderada(pesos). A esta formulación se la conoce como "transformada de Fourier".
  La transformada de Fourier discreta de una función(imagen) $f(x, y)$ de tamaño $M \times N$ está dada por la ecuación:

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

  para $u = 0, 1, 2, ..., M - 1$ y $v = 0, 1, 2, ..., N - 1$

  De manera similar, dada $F(u, v)$, obtenemos $f(x, y)$ via la transformada inversa discreta de Fourier:

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

  para $x = 0, 1, 2, ..., M - 1$ y $y = 0, 1, 2, ..., N - 1$

  Las ecuaciones anterires comprenden al par de transformadas discretas de Fourier bidimensionales (DFT). Las variables u y v son las variables de la transformada o variables de frecuencia, mientras que x y y son las variables espaciales o variables de la imagen.

  Se definen al espectro de Fourier, al ángulo de fase y al espectro de potencia de la siguiente manera, respectivamente:

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$
$$\phi(u,v) = \tan^{-1}[\frac{I(u,v)}{R(u,v)}]$$
$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

donde $R(u,v)$ e $I(u,v)$ son las partes real e imaginaria de $F(u,v)$ respectivamente.

Es de práctica común multiplicar la función de entrada $f(x,y)$ por $(-1)^{x+y}$ antes de calcular la transformada de Fourier. De las propiedades de los exponentes tenemos:

$$\Im[f(x,y)(-1)^{x+y}] = F(u - M/2, v - N/2)$$

donde $\Im[\bullet]$ denota la transformada de Fourier del argumento. Esta ecuación establece que el origen de la transformada de Fourier de $f(x,y)(-1)^{x+y}$ [que es $F(0,0)$] está localizada en $u = M - 1$y $v = N - 1$. En otras palabras multiplicando $f(x,y)$ por $(-1)^{x+y}$ hace que $F(u,v)$ se recorra a las coordenadas de frecuencia $(M/2, N/2)$ que es el centro del area $2D$ ocupada por la DFT.

- Filtrado en frecuencia.

  Generalmente es imposible hacer relaciones directas entre los componentes de los dominios del espacio de la imagen y de la frecuencia. Sin embargo, se pueden encontrar algunas relaciones entre los componentes de la frecuencia y algunas características de la imagen.

  Por ejemplo, se pueden asociar las frecuencias de la transformada de Fourier con patrones de variación de las intensidades de la imagen. La frecuencia más baja ($u = v = 0$) corresponde al promedio de los valores de gris de la imagen. Mientras nos alejamos del origen, las frecuencias corresponden a variaciones suaves en los tonos de gris. Conforme nos alejamos más las frecuencias altas empiezan a corresponder a cambios rápidos o abruptos en los tonos de gris como son por ejemplo los bordes de los objetos y/o el ruido.

  Filtrar en el dominio de la frecuencia consiste en los siguientes pasos:

  1. Multiplicar la imagen de entrada por $(-1)^{x+y}$ para centrar la transformación.
  2. Calcular, $F(u,v)$, la TDF de la imagen resultado de 1.
  3. Multiplicar $F(u,v)$ por la función filtro $H(u,v)$.
  4. Calcular la TDF inversa del resultado de 3.
  5. Obtener la parte real del resultado de 4.
  6. Multiplicar el resultado de 5 por $(-1)^{x+y}$.

  La razón por la cual $H(u,v)$ se llama filtro (también se llama filtro de función de transferencia) es porque suprime ciertas frecuencias en la transformada y deja otras sin cambio.

  Sea $f(x,y)$ la imagen de entrada y $F(u,v)$ su transformada discreta de Fourier. La transformada de Fourier de la imagen salida depués de aplicar el filtro está dada por:

  $$G(u,v) = H(u,v)F(o,v)$$

  donde la multiplicación de H por F involucra funciones bidimensionales y está definida elemento a elemento.

3. Desarrollo

Resuelve los problemas de la lista siguiente y describe tu solución en cada inciso. Los incisos en donde únicamente tengas que desplegar imágenes no requieren de ninguna descripción.

1. Calcular la transformada discreta de Fourier a la imagen del Sr.Fourier y generar el ejemplo visto del libro Castlemam, 1996, figura 10.10.

   a) La versión sin corrimiento de la amplitud y la fase de la imagen de entrada I.

      Se implementó la función `fourier(imagen)` que calcula la transformada de Fourier y retorna el espectro de magnitud y fase sin aplicar corrimiento.

      Proceso:

      1. Se aplica `np.fft.fft2(imagen)` para obtener la transformada de Fourier de la imagen de entrada.
      2. Se calcula la magnitud como `abs(tf1)` y la fase como `np.angle(tf1)`.
      3. Para la visualización, se aplica escala logarítmica a la magnitud `np.log(1+magnitud)`, lo que comprime el rango dinámico y permite observar componentes de baja intensidad.

   b) La versión con corrimiento de la amplitud y la fase de la imagen de entrada I.

      Se implemento la función `fourier_corrimiento(imagen)`, que centra el espectro de frecuencias:

      Proceso:

      1. Se calcula la transformada de Fourier `np.fft.fft2(imagen)`.
      2. Se aplica `np.fft.fftshift()` para desplazar la componente de frecuencia cero al centro de la matriz.
      3. Se calcula magnitud y fase del espectro centrado.
      4. Se aplica escala logarítmica a la magnitud para visualización.

      Este corrimiento corresponde a multiplicar la imagen original por $(-1)^{x+y}$ antes de la transformada.

   c) Regresar con la transformada inversa completa a la imagen original.

      Se implemento la función `fourier_inv_comp(imagen)` que reconstruye la imagen utilizando magnitud y fase originales:

      Proceso:

      1. Se calcula la transformada de Fourier de la imagen.
      2. Se extraen magnitud y fase.
      3. Se reconstruye la señar compleja `tf1_magnitud * np.exp(1j * tf1_fase)`.
      4. Se aplica la transformada inversa `np.fft.ifft2()`.
      5. Se obtiene la magnitud del resultado `abs(itf1_completa)`

d) Regresar con la transformada inversa sólo con amplitud.

Se implementó la función `fourier_inv_amp(imagen)` que reconstruye usando únicamente la información de magnitud, asumiendo fase = 0.

Proceso:

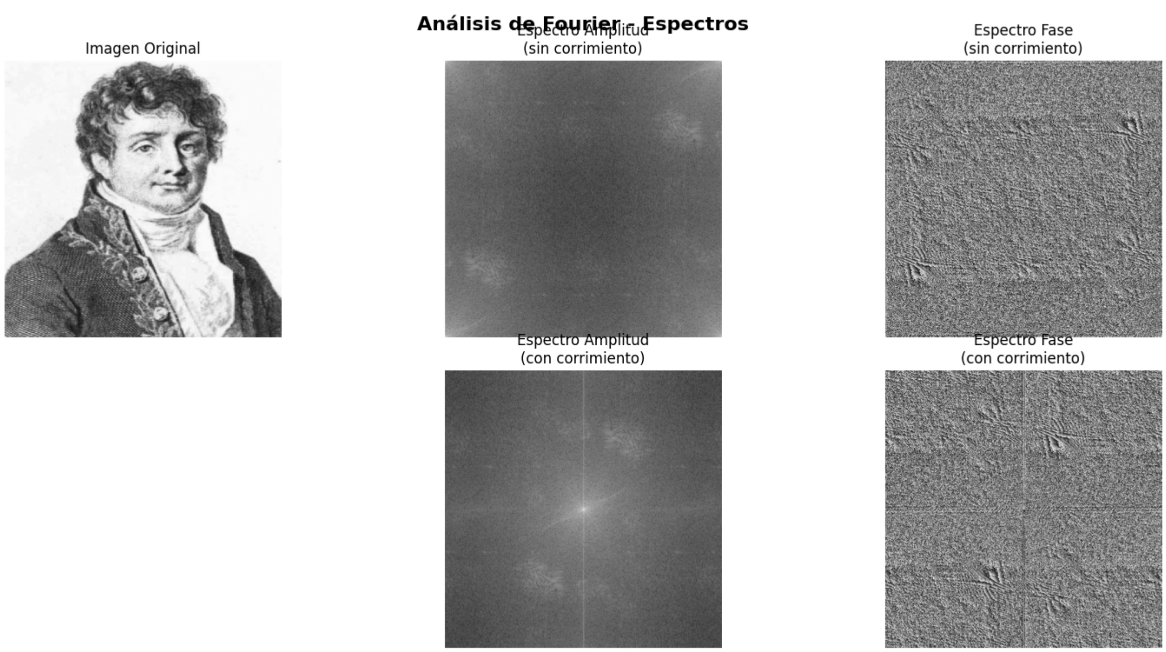
1. Se calcula la transformada de Fourier y se extrae solo la magnitud.
2. Se aplica la transformada invertida usando solo magnitud(sin fase) `np.fft.ifft2(tf1_magnitud)`.
3. Se obtiene la magnitud del resultado complejo.
4. Se aplica escala logarítmica y normalización para visualización.

e) Regresar con la transformada inversa sólo con la fase.

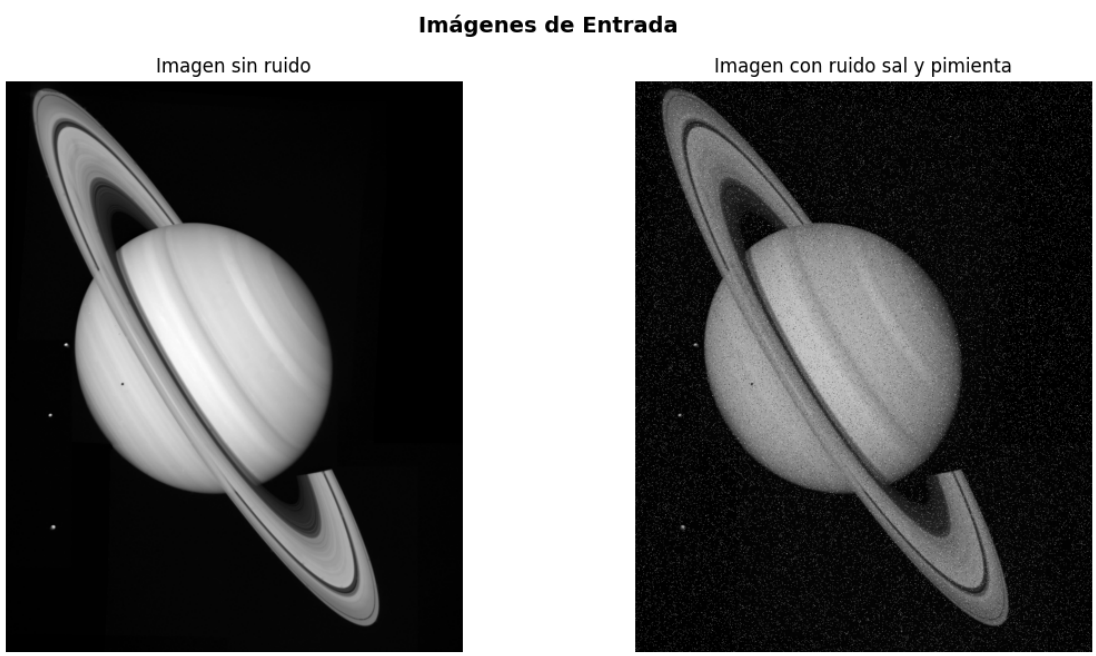
Se implementó la función `fourier_inv_fase(imagen)` que reconstruye usando únicamente la fase, con magnitud unitaria:

Proceso:

1. Se calcula la transformada de Fourier con corrimiento `np.fft.fftshift(np.fft.fft2(imagen))`.
2. Se extrae solo la fase `tfc1_fase = np.angle(tfc1)`.
3. Se reconstruye con magnitud = 1. `np.exp(1j * tfc1_fase)`.
4. Se aplica la transformada inversa `np.fft.ifft2()`.
5. Se obtiene la magnitud del resultado.

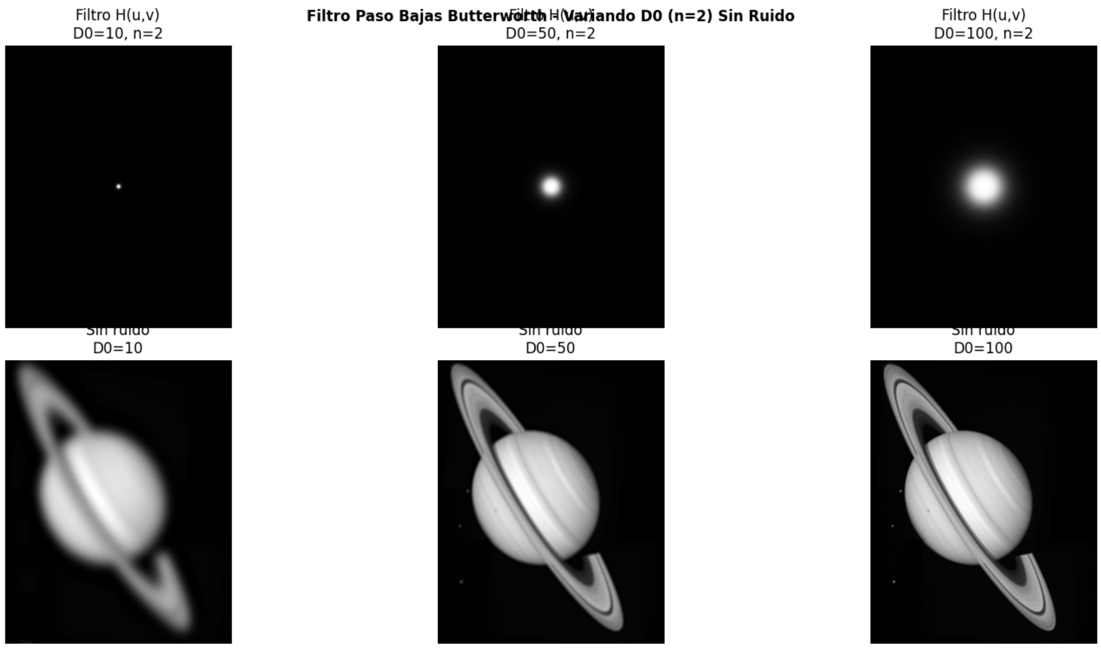


2. Aplicar a una imagen sin ruido y la misma imagen con ruido "sal y pimienta"filtros de suavizamiento y realce.



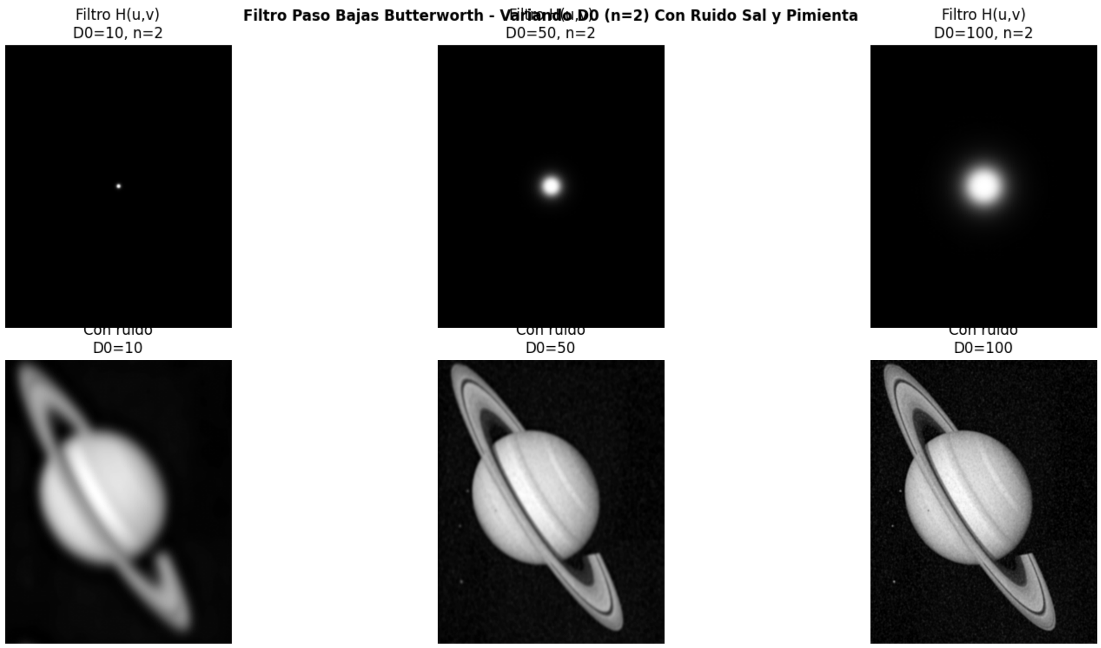
a) Aplicar el filtro de paso bajas Butterworth, para eliminación de ruido. Prueba para varios valores de frecuencia de corte D_0 y para varios valores de orden n y establece cuales valores son mejores para la imagen en cuestión. Realiza el ejercicio para la imagen con y sin ruido.

Resultados en Filtro de paso bajas Butterworth con varios $D_0 = [10, 50, 100]$ y $n = 2$, con y sin filtro



Sin Ruido

- $D_0 = 10$: El suavizado es muy fuerte, lo que causa una pérdida significativa de detalles.
- $D_0 = 50$: Hay un balance óptimo.
- $D_0 = 100$: Suavizado suave.

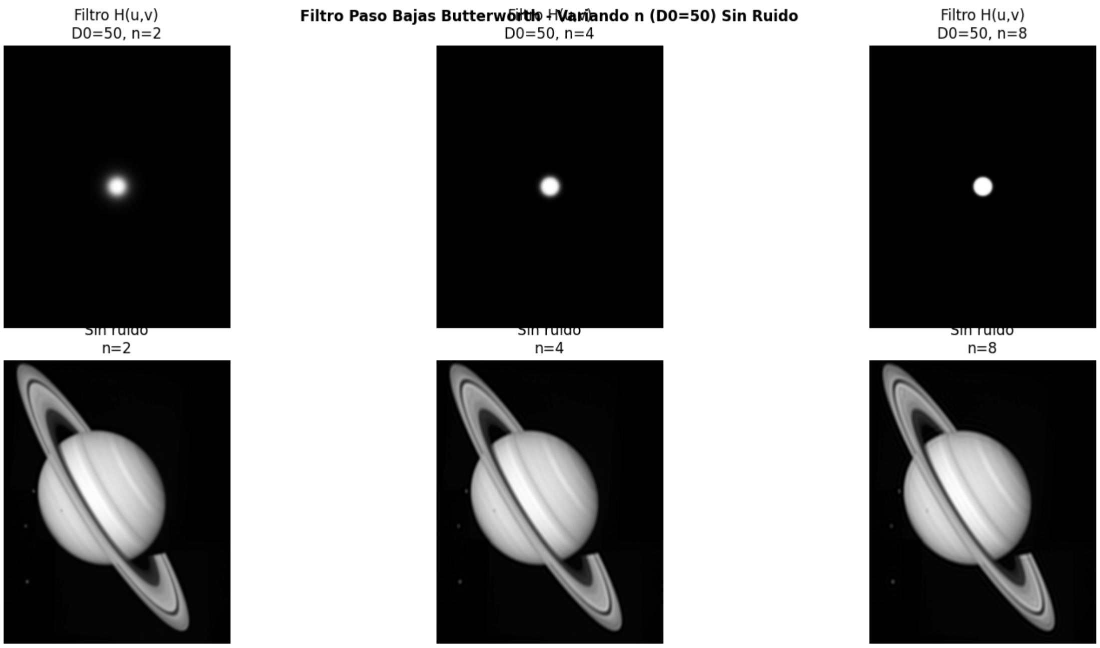


Con Ruido

- $D_0 = 10$: El suavizado es muy fuerte, elimina el ruido sal y pimienta pero causa pérdida significativa de detalles.
- $D_0 = 50$: Hay un balance óptimo. Reduce considerablemente el ruido manteniendo los bordes principales visibles.
- $D_0 = 100$: Suavizado suave. Pasa más frecuencias altas, preserva mejor los detalles pero elimina menos ruido.

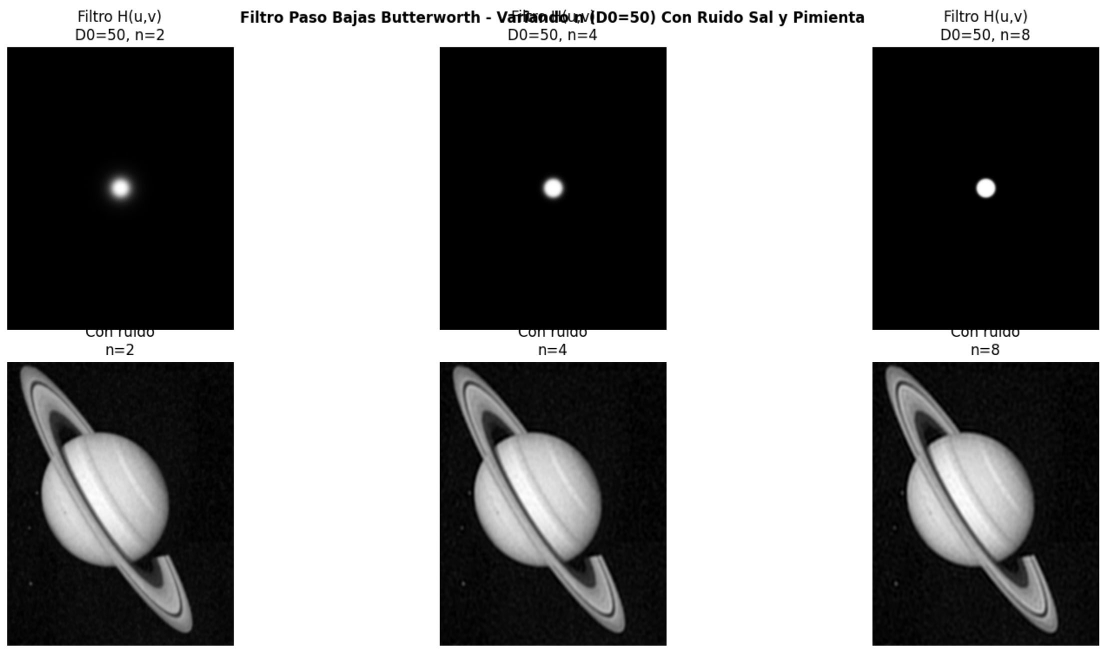
Podemos concluir que el mejor resultado tanto con y sin ruido es con $D_0 = 50, n = 2$

Resultados en Filtro de paso bajas Butterworth con varios $n = [2, 4, 8]$ y $D_0 = 50$, con y sin filtro



Sin Ruido

- $n = 2$: Transición suave, produce un resultado óptimo.
- $n = 4$: Transición más pronunciada, produce un resultado similar al anterior.
- $n = 8$: Transición muy abrupta, es un resultado ideal, sin embargo puede percibirce un poco de artefactos de anillos.



Con Ruido

- $n = 2$: Transición suave, produce resultados óptimos y reduce considerablemente el ruido.
- $n = 4$: Transición más pronunciada. Mejora la selectividad del filtro, separando mejor las frecuencias que pasan de las que se atenúan.
- $n = 8$: Transición muy abrupta, se pueden percibir artefactos de anillos.

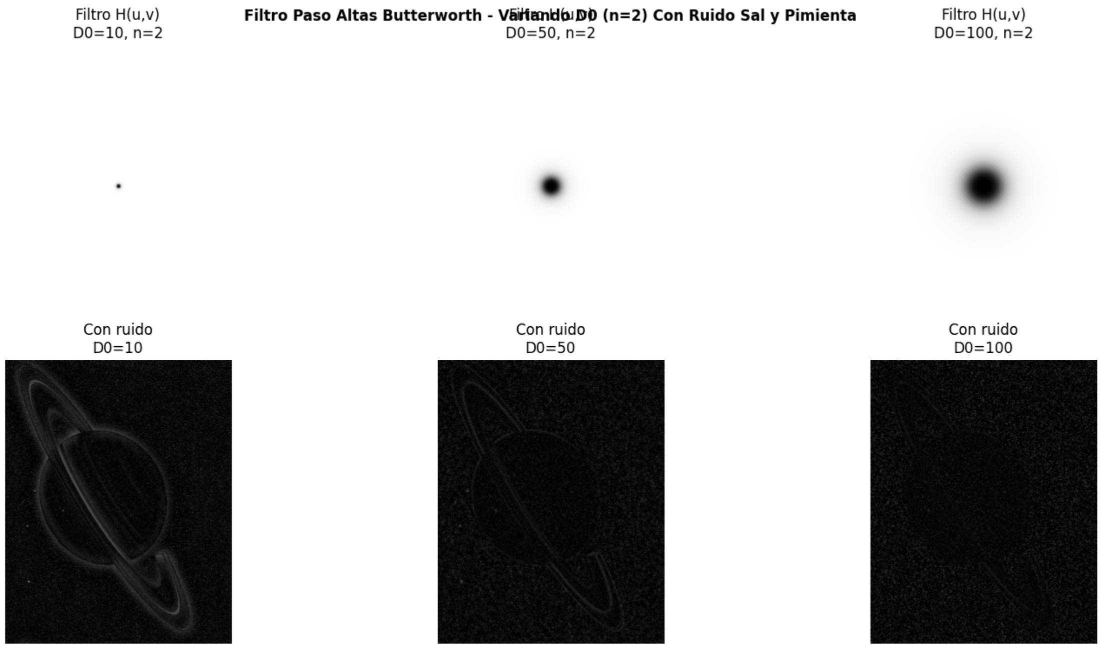
Podemos concluir que el mejor resultado tanto con y sin ruido es con $D_0 = 50, n = 4$

b) Aplicar el filtro paso altas Butterworth para el realce de bordes. Prueba para varios valores de frecuencia de corte D_0 y para varios valores de orden n y establece cuales valores son mejores para la imagen en cuestión.

Resultados en Filtro de paso altas Butterworth con varios $D_0 = [10, 50, 100]$ y $n = 2$, con y sin filtro

Sin Ruido

- $D_0 = 10$: Se pueden percibir los bordes de manera similar a la imagen original.
- $D_0 = 50$: Se pueden percibir bordes, sin embargo hay perdida de detalles.
- $D_0 = 100$: Pérdida significativa de bordes, pero todavía se alcanza a percibir la figura.



Con Ruido

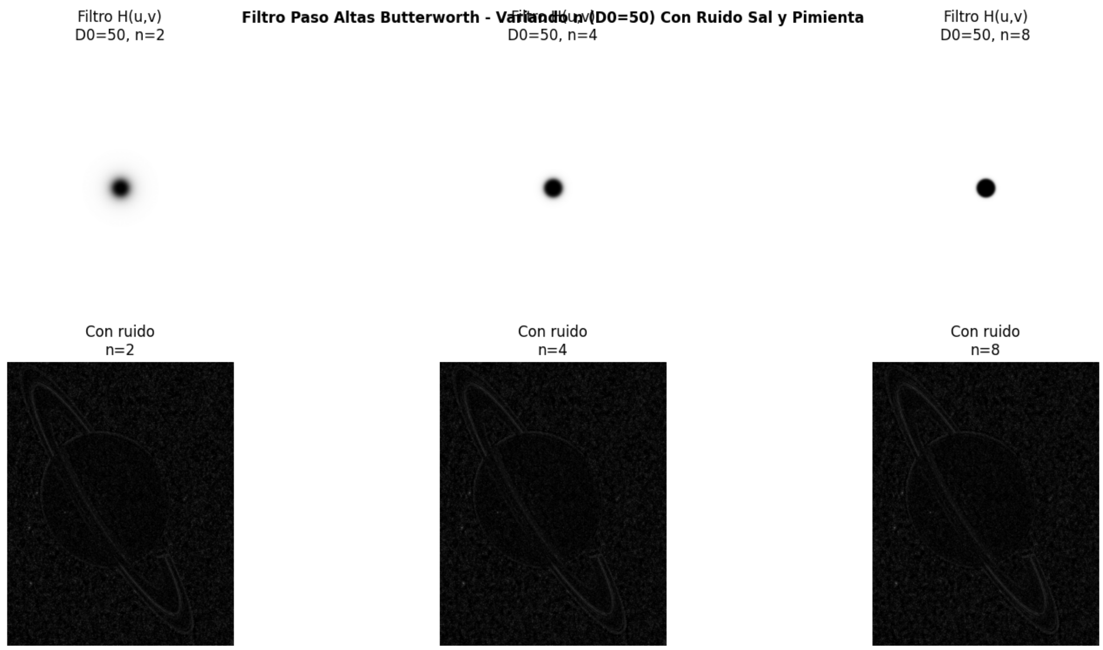
- $D_0 = 10$: Se pueden percibir los bordes de manera óptima, y poco ruido.
- $D_0 = 50$: Se perciben pocos bordes además de ruido.
- $D_0 = 100$: Se puede percibir la silueta, sin embargo hay pérdida significativa de bordes.

Podemos concluir que el mejor resultado tanto con y sin ruido es con $D_0 = 10, n = 2$

Resultados en Filtro de paso altas Butterworth con varios $n = [2, 4, 8]$ y $D_0 = 50$, con y sin filtro

Sin Ruido

- $n = 2$: Se pueden observar de manera óptima los bordes.
- $n = 4$: Aún se observan los bordes y muy poca presencia de artefactos de anillo.
- $n = 8$: Se siguen observando los bordes y hay presencia de artefactos de anillo.



Con Ruido

- $n = 2$: Se pueden observar ciertos bordes y un ruido presente.
- $n = 4$: Aún se observan los bordes y muy poca presencia de artefactos de anillo y ruido.
- $n = 8$: Se siguen observando los bordes y hay presencia de artefactos de anillo y ruido.

Podemos concluir que el mejor resultado tanto con y sin ruido es con $D_0 = 50, n = 2$

4. Conclusión

La práctica demostró la **reversibilidad de la Transformada Discreta de Fourier (DFT) 2D** y el papel crítico de sus componentes. Se confirmó que la **fase es el componente estructural** esencial, ya que su reconstrucción en solitario preservó la forma visual de la imagen, mientras que la magnitud sola no lo logró. Los **Filtros Butterworth** resultaron ser superiores a los filtros ideales debido a sus transiciones suaves, las cuales previenen los **artefactos de *ringing*** (oscilaciones). Específicamente, el **Filtro Pasa Bajas** con $D_0 \approx 50$ y $n = 2$ fue óptimo para el **suavizado y eliminación de ruido** impulsivo. Por otro lado, el **Filtro Pasa Altas** con $D_0 = 50$ y $n = 2$ fue efectivo para el **realce de bordes** en imágenes limpias; sin embargo, amplifica el ruido en imágenes contaminadas, lo que subraya la necesidad de aplicar primero el suavizado. El control preciso sobre los parámetros D_0 y n, manteniendo órdenes bajos ($n \leq 4$), convierte a la transformada de Fourier en una herramienta **fundamental y eficiente** para el procesamiento de imágenes.

Referencias

[1] Abhishek Jain, "Salt and pepper noise and how to remove it in machine learning", https://medium.com/@abhishekjainindore24/salt-and-pepper-noise-and-how-to-remove-it-in-machine-learning-032d76b138a5

[2] Ms Somanna, "Guide to adding noise to your data using Python and Numpy", https://medium.com/@ms_somanna/guide-to-adding-noise-to-your-data-using-python-and-numpy-c8be815df524

[3] eastWillow, "Fourier Transform", https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_transforms/py_fourier_transform/py_fourier_transform.html

[4] Stephen Gruppetta, "How to Create Any Image Using Only Sine Functions | 2D Fourier Transform in Python", https://thepythoncodingbook.com/2021/08/30/2d-fourier-transform-in-python-and-fourier-synthesis-of-images/