UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

COMPLEJIDAD COMPUTACIONAL (2024-1)

PRÁCTICA 04

- Vargas Bravo Paola 318074755

FECHA LIMITE : 06/11/23

Contents

1.

1 Reglas generales para el desarrollo de las Prácticas de Laboratorio

- Deberás respetar la estructura general de este documento, i.e, entregar tu práctica con las secciones: objetivos, introducción, desarrollo,código, conclusiones y referencias.
- El desarrollo de la práctica deberá ser auténtico. Aquellas personas que presenten los mismo cálculos, código fuente, etc , serán sancionados.
- El día de entrega establecido deberá ser respetado por todos. La hora límite de entrega será establecida en su momento y no se reciben trabajo posteriormente.
- Deberás entregar el documento impreso así como el código a Miguel Angel Veloz Lucas via Goolge Classroom https://classroom. google.com/c/NjE4MjMxNzU1MDE2

2 Objetivos

- 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.

3 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 MxN 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 anteriores 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:

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$$

$$\phi(u,v) = tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

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 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 área 2D ocupada por la DFT.

- Filtrado en frecuencia
  Generalmente es imposible hacer relaciones directas entre los compo- nentes 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:
  - Multiplicar la imagen de entrada por $(1)^{x+y}$ para centrar la transformación.
  - Calcular, $F(u,v)$, la TDF de la imagen resultado de 1.
  - Multiplicar $F(u,v)$ por la función filtro $H(u,v)$.
  - Calcular la TDF inversa del resultado de 3.
  - Obtener la parte real del resultado de 4.
  - 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 de salida después de aplicar el filtro está dada por:

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

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

4 Desarrollo

- **Ejercicio 1** : Calcular la transformada discreta de Fourier a la imagen del Sr. Fourier y generar el ejemplo visto del libro Castleman, 1996.
  (a) La versión sin corrimiento de la amplitud y la fase de la imagen de entrada I.
  (b) La versión con corrimiento de la amplitud y la fase de la imagen de entrada I.
  (c) Regresar con la transformada inversa completa a la imagen original.
  (d) Regresar con la transformada inversa solo con amplitud.
  (e) Regresar con la transformada inversa solo con la fase.

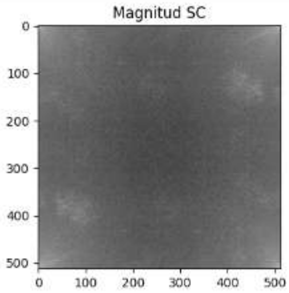
`Función : apply_TFF:`

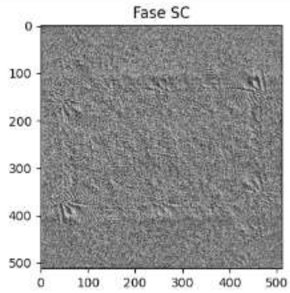
Parámetros : imagen

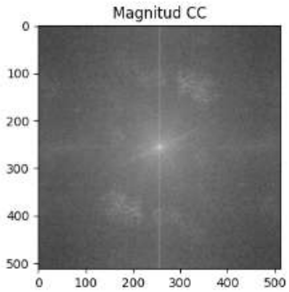
- Abre la imagen original.
- Convertimos una imagen PIL en una matriz NumPy llamada **image_array**.
- Para explicar lo que hacemos vamos a dividir el código en dos partes :
  * Con corrimiento .
    (a) Para ello multiplicamos primero la imagen por $(-1)^{x+y}$ a la imagen original.
    (b) Obtenemos la transformada de fourier (TF).
    (c) Obtenemos la fase y la magnitud con corrimiento (CC) a partir de TF.
    (d) Para regresar la imagen original; A la TF le calculamos la inversa, sacamos su parte real y luego volvemos a multiplicar por $(-1)^{x+y}$.
    (e) En el caso de la Magnitud y de la fase para regresarlas se hace algo diferente :
      · Amplitud : Sacamos la inversa de la amplitud y luego aplicamos el valor absoluto.
      · Fase : Sacamos la inversa pero no solo de la fase sino de :

$$e^{\text{fase}*1j}$$

, luego aplicamos el valor absoluto.
  * Sin corrimiento .
    Hacemos los mismos pasos solo sin multiplicar por $(-1)^{x+y}$ a la imagen original ,de ahí obtenemos la amplitud y la magnitud SC (sin corrimiento).


Magnitud SC


Fase SC


Magnitud CC

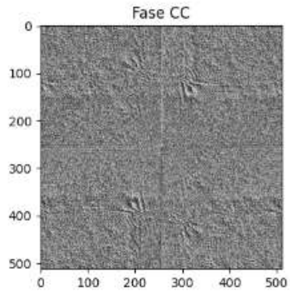

Fase CC


Imagen Regresada

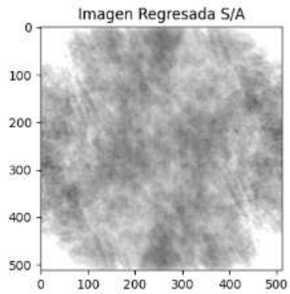


Imagen Regresada S/A

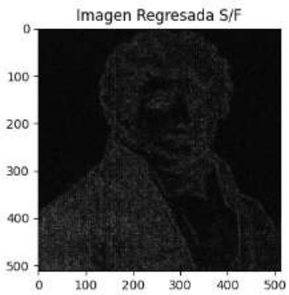


Imagen Regresada S/F

- **Ejercicio 2** : Aplicar a una imagen sin ruido y la misma imagen con ruido sal y pimienta filtros de suavizamiento y realce. La imagen con ruido se puede generar a partir de la imagen sin ruido usando el siguiente comando de MATLAB:J=imnoise(I,TIPO,...), donde TIPO es una cadena que puede tomar valores 'gaussian', 'salt & peper', etc. Ver comando help imnoise.
  (a) Aplicar el filtro paso bajas Butterworth, para eliminación de ruido. Prueba para varios valores de frecuencia de corte D0 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.
  (b) Aplicar el filtro paso altas Butterworth para el realce de bordes. Prueba para varios valores de frecuencia de corte D0 y para varios valores de orden n y establece cuales valores son mejores para la imagen en cuestión.

`Función : apply_FB _pb`

Parámetros : i mage, D0, n , tipoFiltro

- Abre la imagen original.
- Convertimos una imagen PIL en una matriz NumPy llamada **image_array**.

- Centramos la imagen multiplicando por $(-1)^{x+y}$.
- Calculamos la TF de la imagen.
- Calculamos el filtro butterwoth con una función auxiliar en el cual se aplica a la formula de :

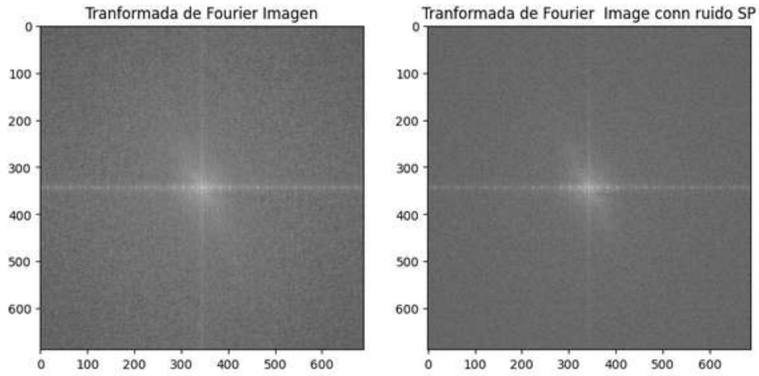
$$H(u,v) = \frac{1}{1 + (D(u,v)/D0)^{2n}}$$

  a las coordenadas de la frecuencia (u, v).
- Luego si el filtro es 'pa' (paso altas) calculamos el complemento de el filtro anterior :

$$H = H - 1$$

  En otro caso se queda igual.
- Luego multiplicamos nuestro filtro por TF, calculamos la inversa, obtenemos la parte real y multiplicamos por $(-1)^{u+v}$ la cual será nuestra imagen objetivo. Regresamos así la imagen objetivo y el filtro H.

(a) Imágenes Transformada de fourier de la imagen con y sin ruido S&P.

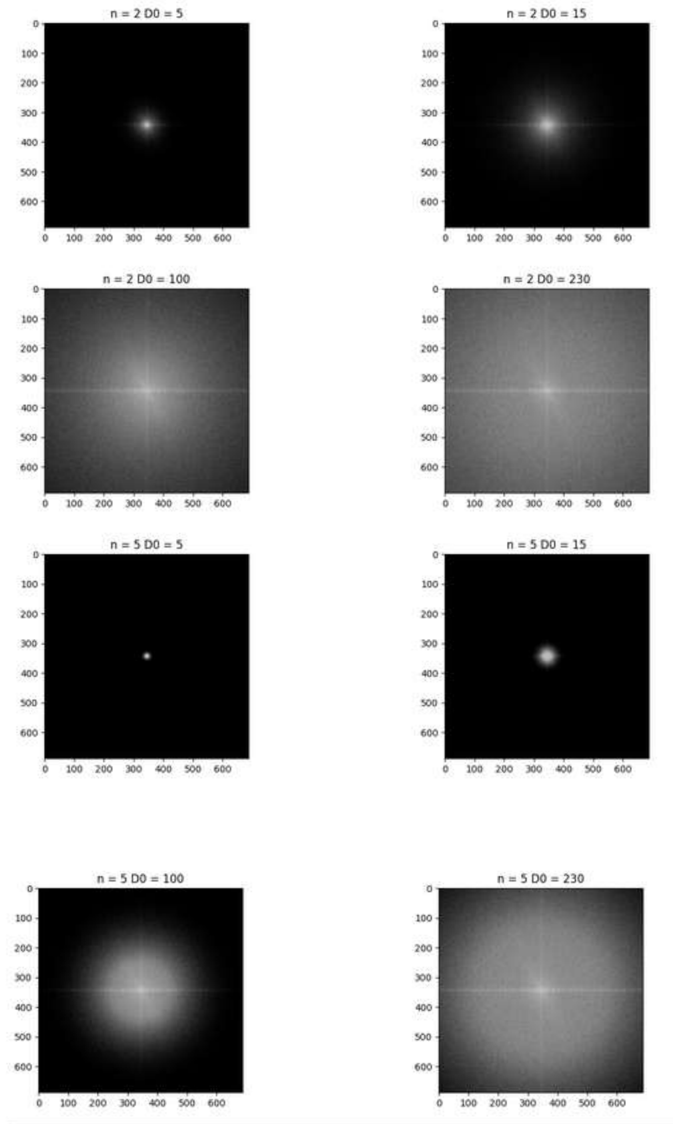


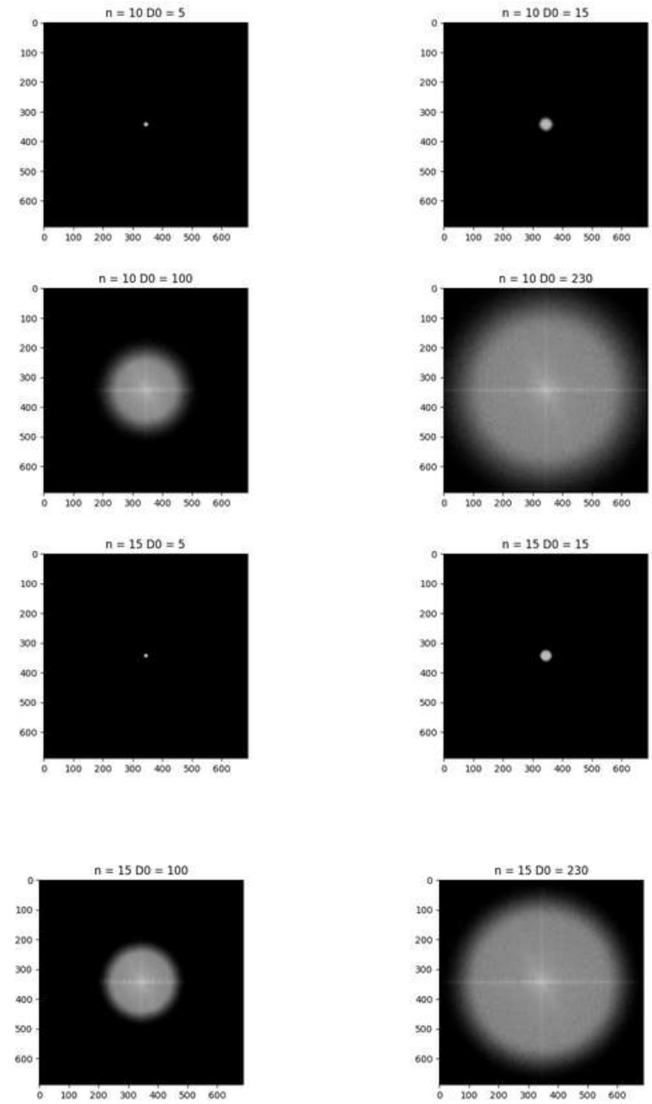
(b) Paso Bajas

i.

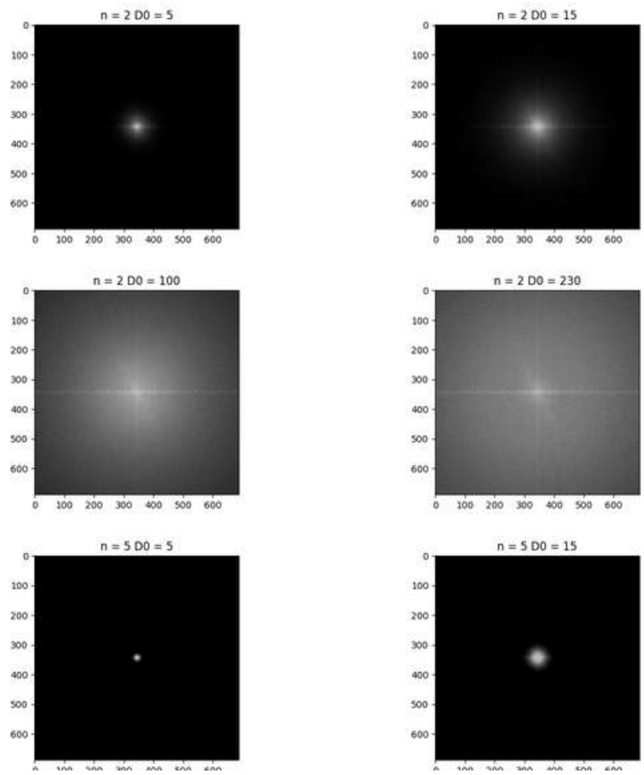
$$G(u,v) = F(u,v)H(u,v)$$

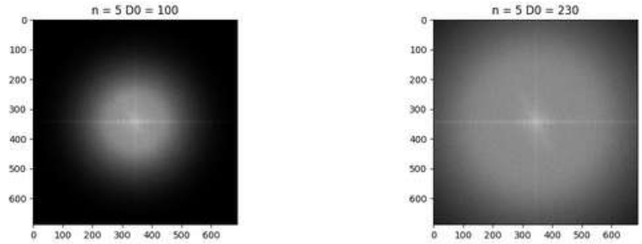
– Sin ruido



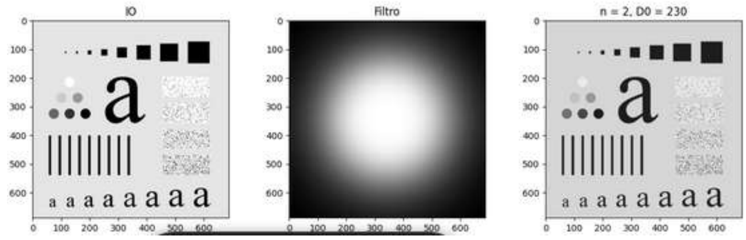

n = 10 D0 = 5
n = 10 D0 = 15
n = 10 D0 = 100
n = 10 D0 = 230
n = 15 D0 = 5
n = 15 D0 = 15
n = 15 D0 = 100
n = 15 D0 = 230

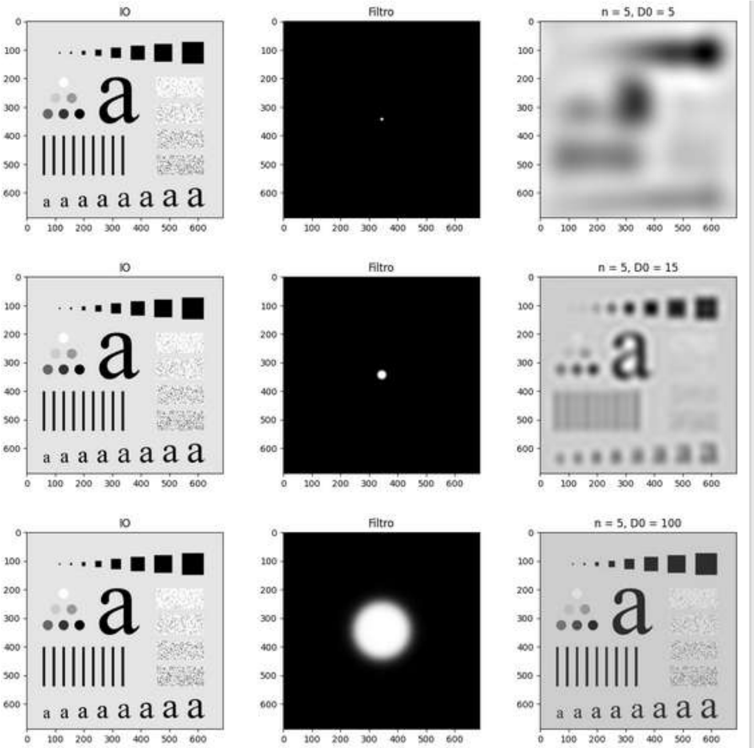
- Con ruido

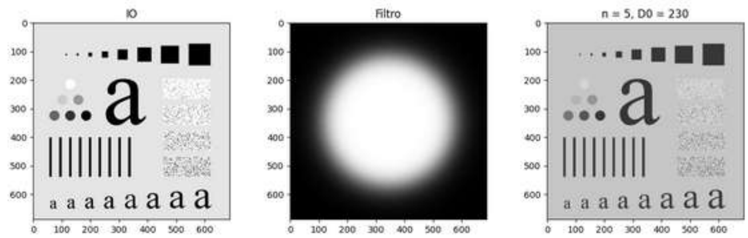

n = 2 D0 = 5
n = 2 D0 = 15
n = 2 D0 = 100
n = 2 D0 = 230
n = 5 D0 = 5
n = 5 D0 = 15

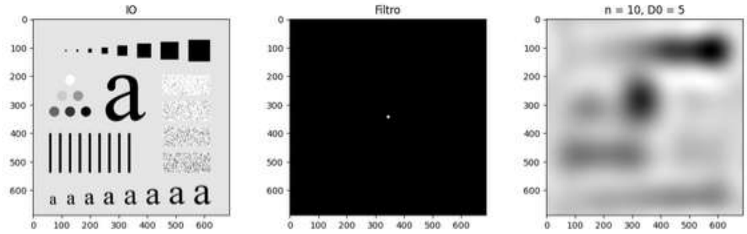


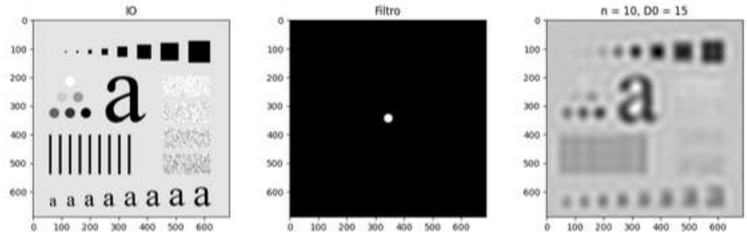
ii. Imagen + Filtro + Resultado

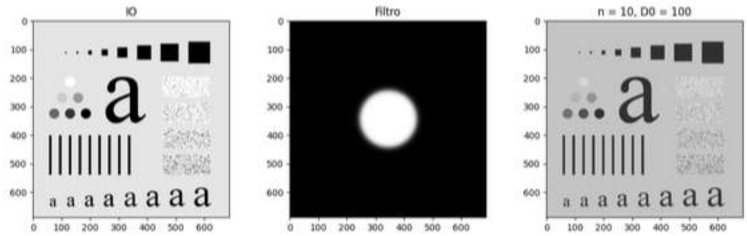


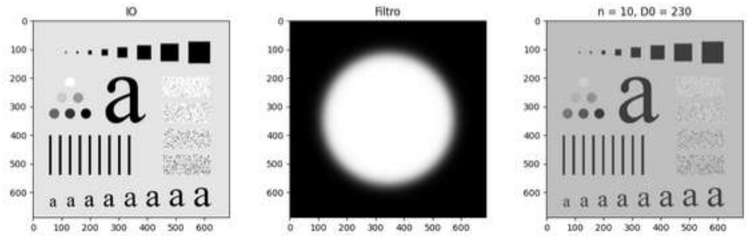


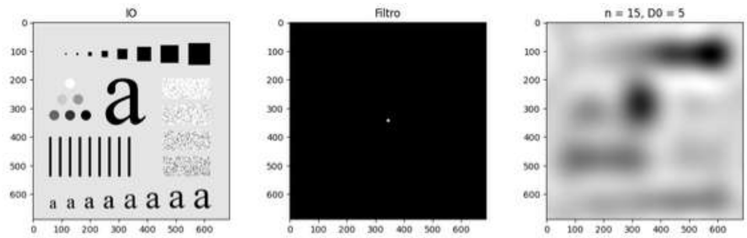


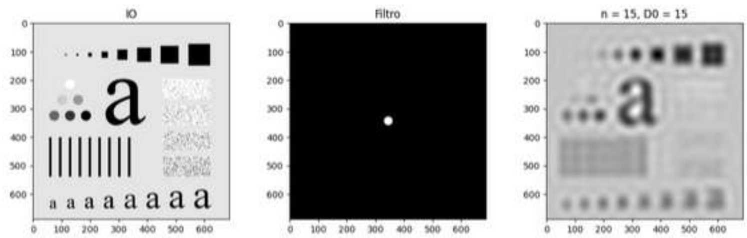

IO
Filtro
n = 10, D0 = 5

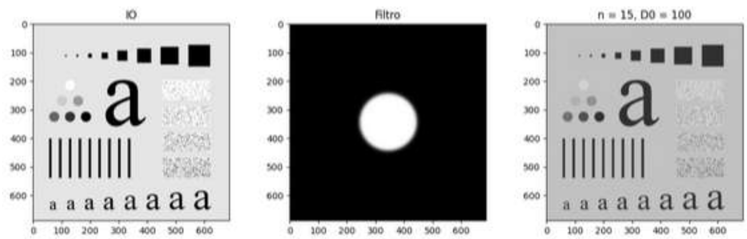

IO
Filtro
n = 10, D0 = 15

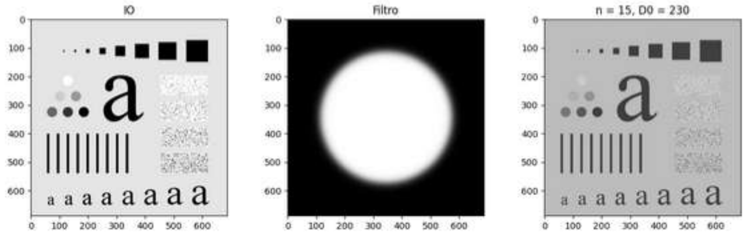

IO
Filtro
n = 10, D0 = 100

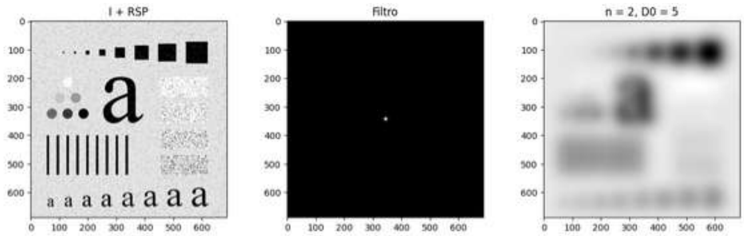

IO
Filtro
n = 10, D0 = 230

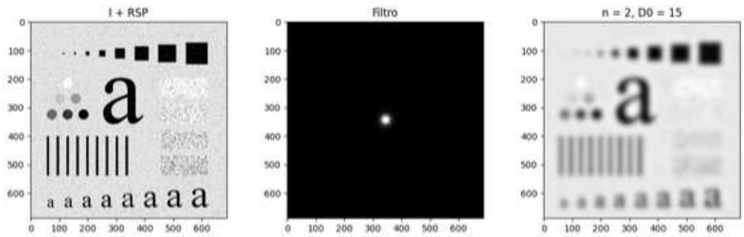

IO
Filtro
n = 15, D0 = 5

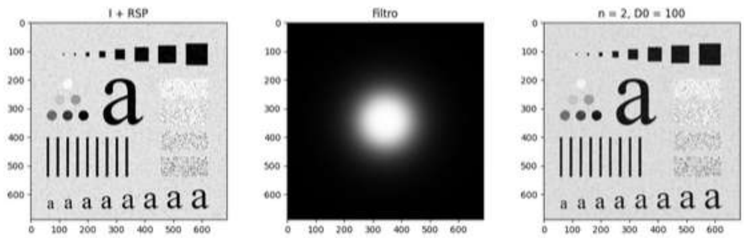

IO
Filtro
n = 15, D0 = 15

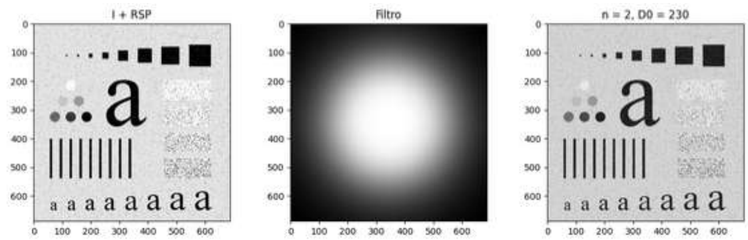

IO
Filtro
n = 15, D0 = 100

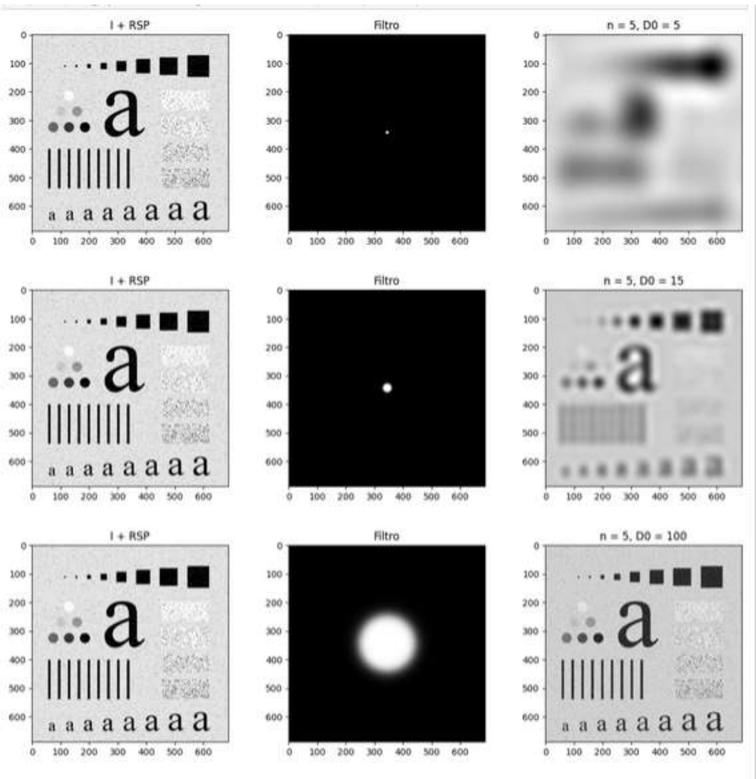

IO
Filtro
n = 15, D0 = 230

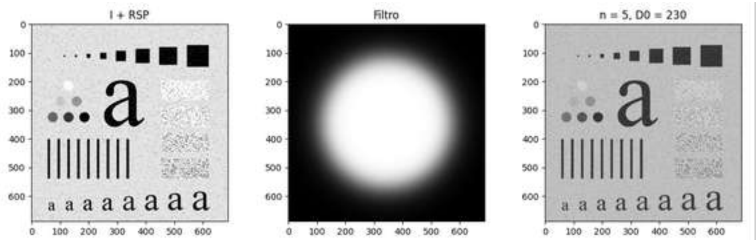

I + RSP
Filtro
n = 2, D0 = 5

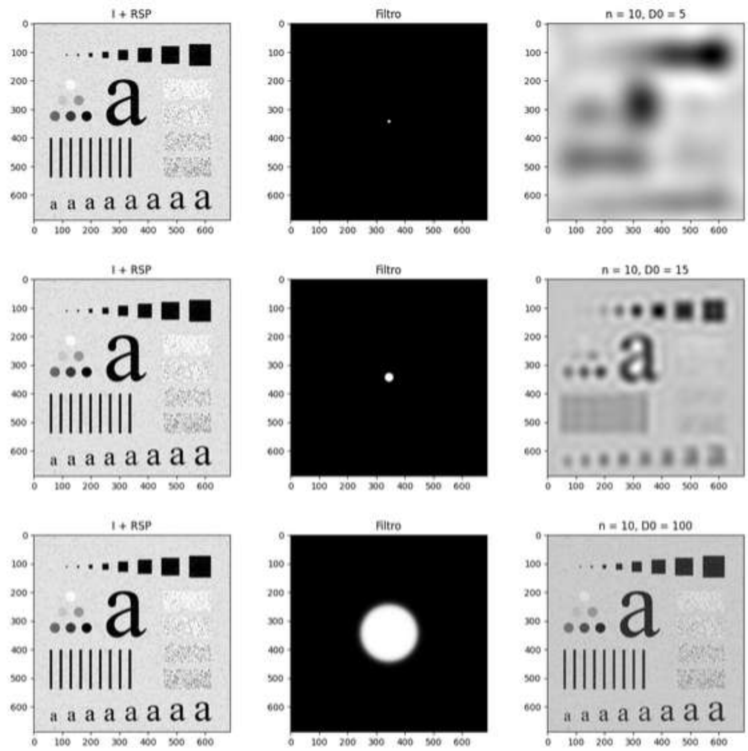

I + RSP
Filtro
n = 2, D0 = 15

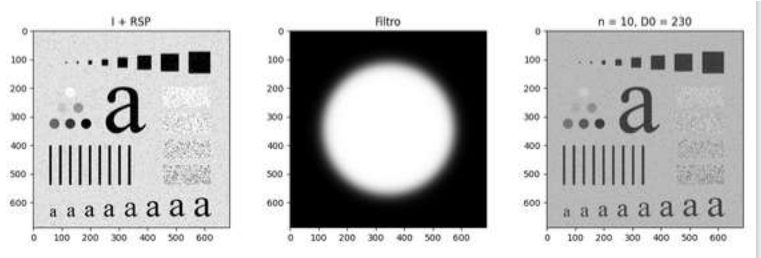

I + RSP
Filtro
n = 2, D0 = 100

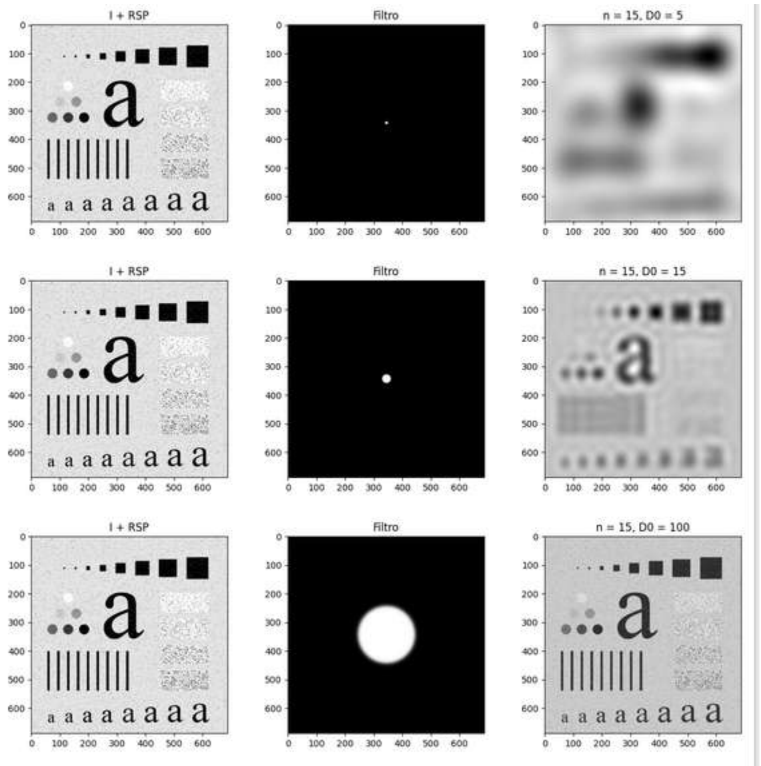

I + RSP
Filtro
n = 2, D0 = 230

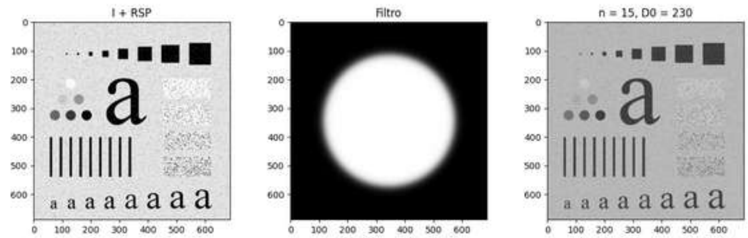

I + RSP
Filtro
n = 5, D0 = 5
I + RSP
Filtro
n = 5, D0 = 15
I + RSP
Filtro
n = 5, D0 = 100


I + RSP
Filtro
n = 5, D0 = 230


I + RSP
Filtro
n = 10, D0 = 5
I + RSP
Filtro
n = 10, D0 = 15
I + RSP
Filtro
n = 10, D0 = 100



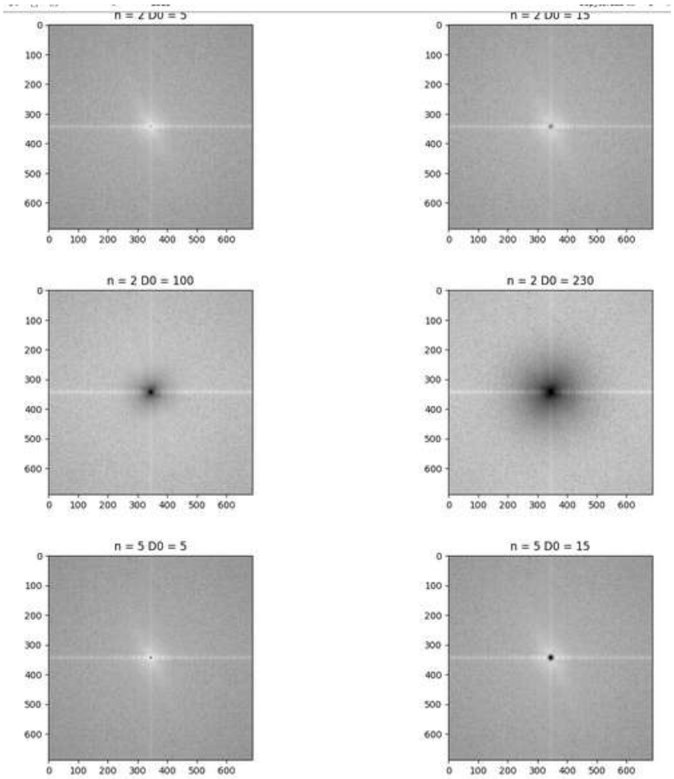




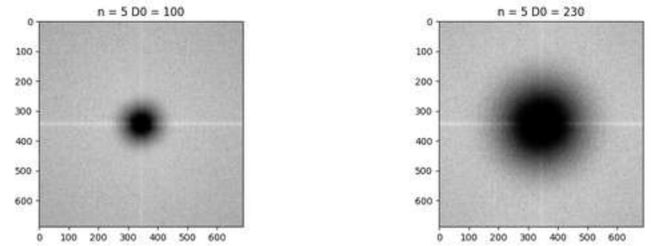
(c) Paso Altas

i.

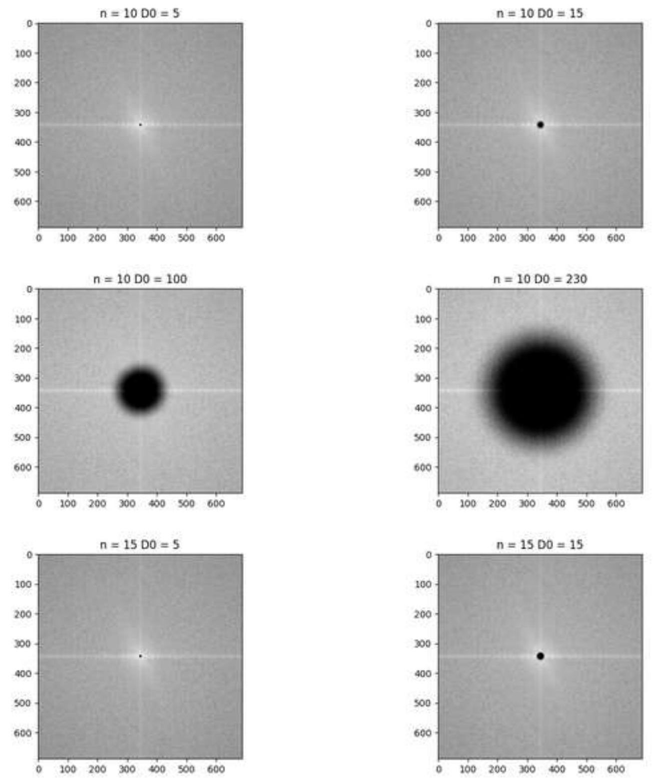
$$G(u,v) = F(u,v)H(u,v)$$



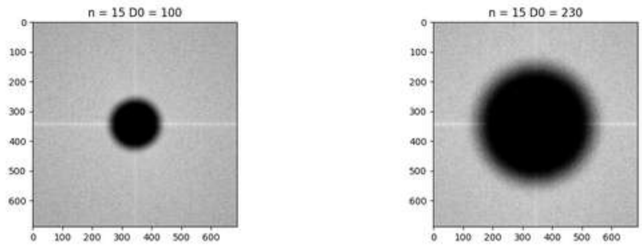
n = 2 D0 = 5
n = 2 D0 = 15
n = 2 D0 = 100
n = 2 D0 = 230
n = 5 D0 = 5
n = 5 D0 = 15



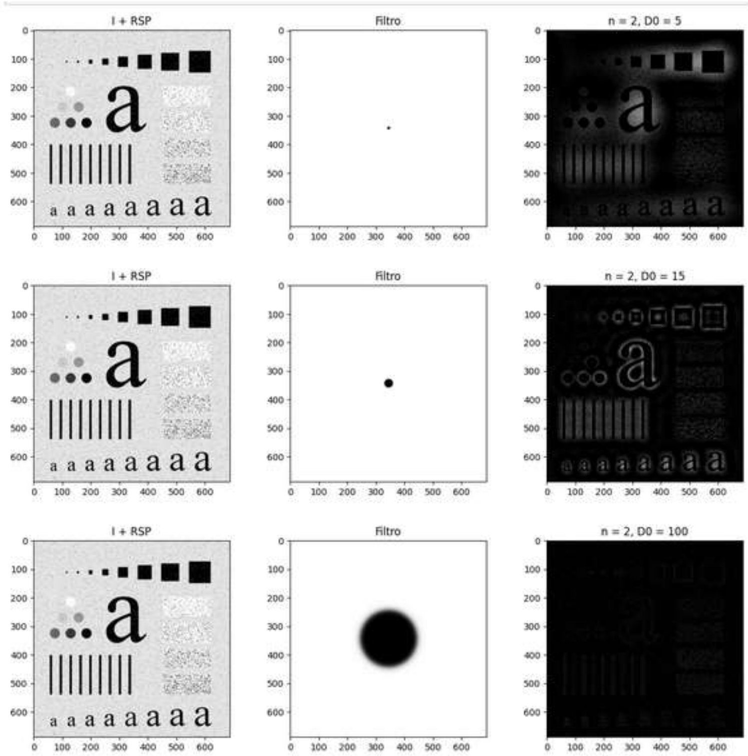
n = 5 D0 = 100
n = 5 D0 = 230

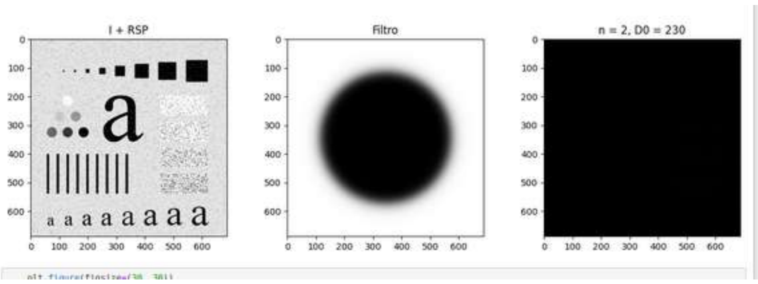


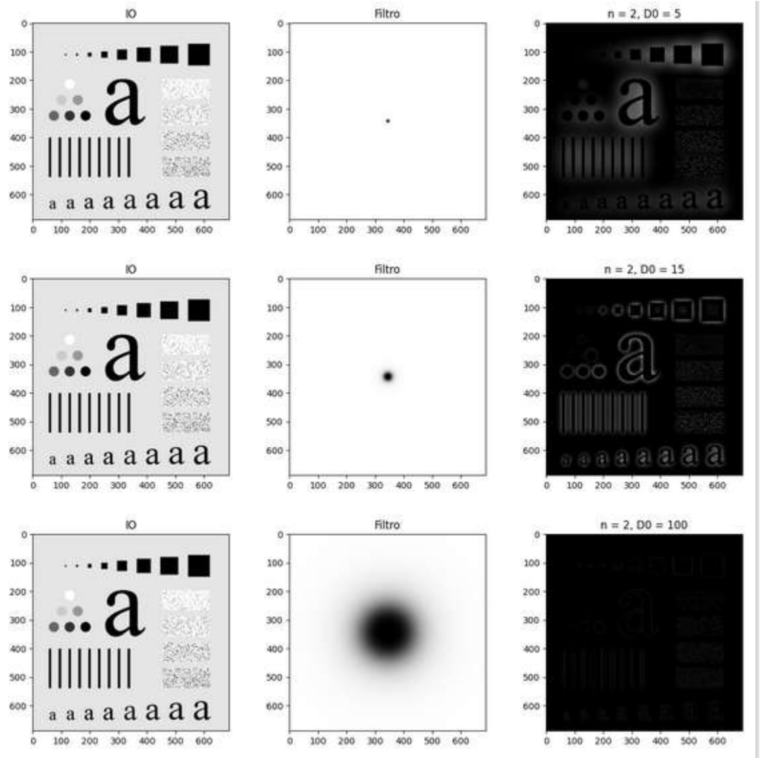
n = 10 D0 = 5
n = 10 D0 = 15
n = 10 D0 = 100
n = 10 D0 = 230
n = 15 D0 = 5
n = 15 D0 = 15

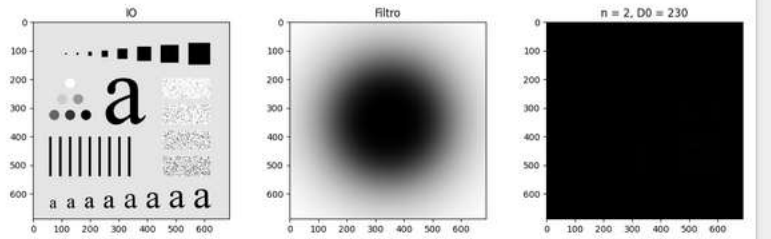


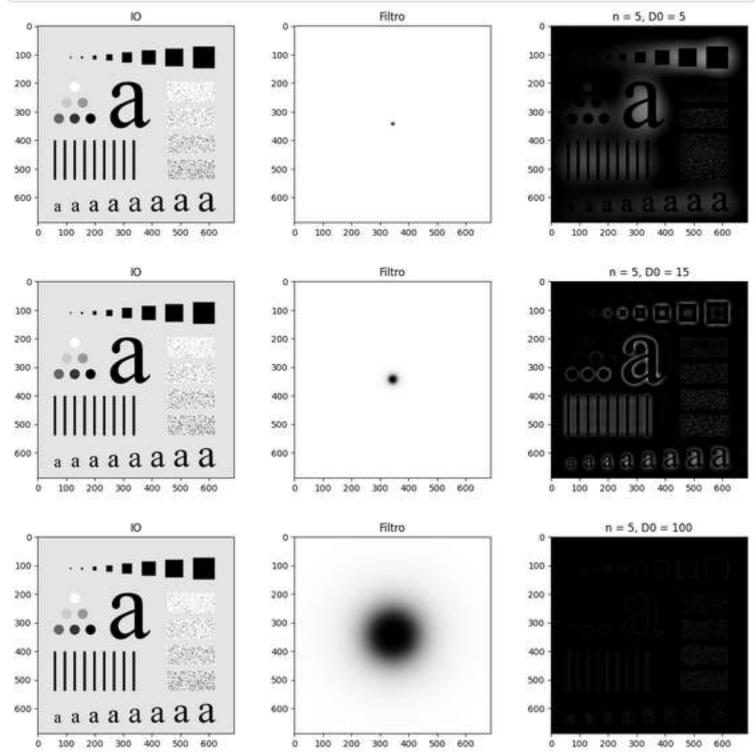
ii. Imagen + Filtro + Resultado

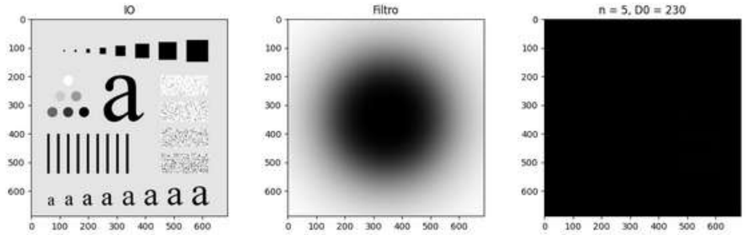


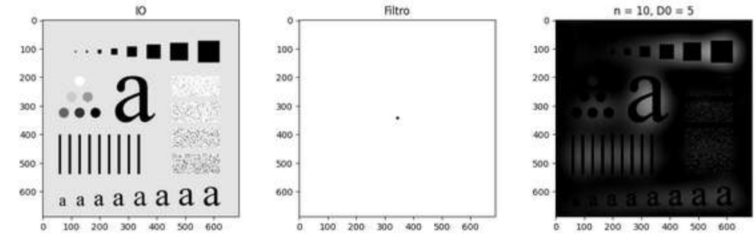


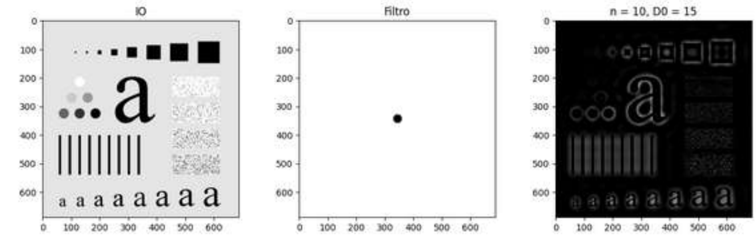

IO
Filtro
n = 2, D0 = 5
IO
Filtro
n = 2, D0 = 15
IO
Filtro
n = 2, D0 = 100

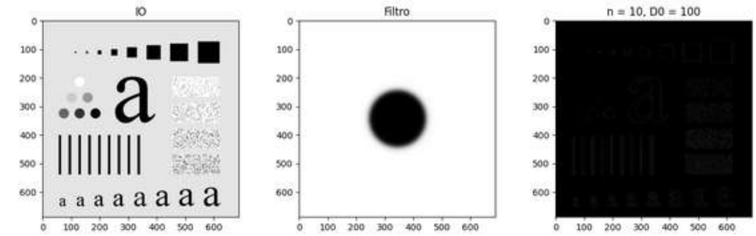

IO
Filtro
n = 2, D0 = 230

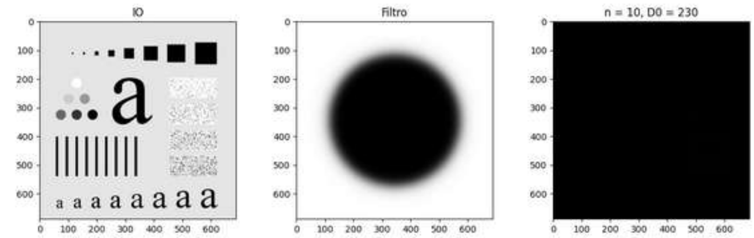

IO
Filtro
n = 5, D0 = 5
IO
Filtro
n = 5, D0 = 15
IO
Filtro
n = 5, D0 = 100

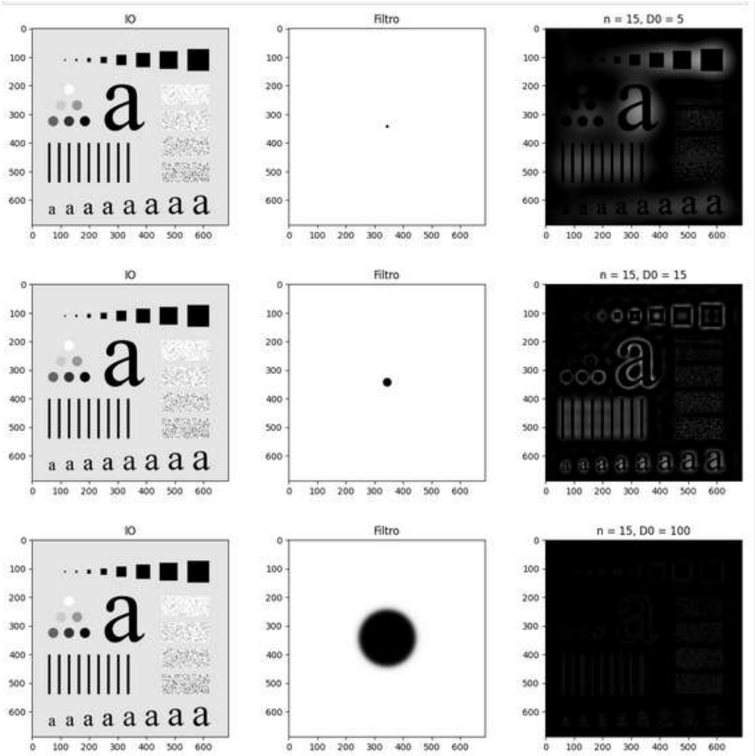

IO
Filtro
n = 5, D0 = 230

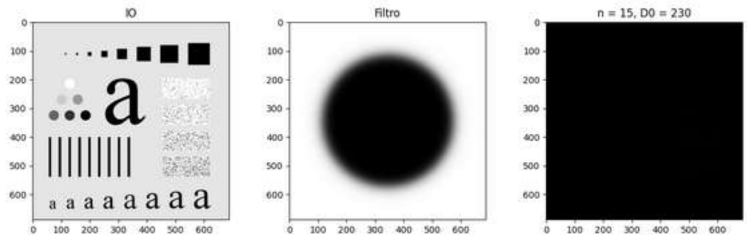

IO
Filtro
n = 10, D0 = 5

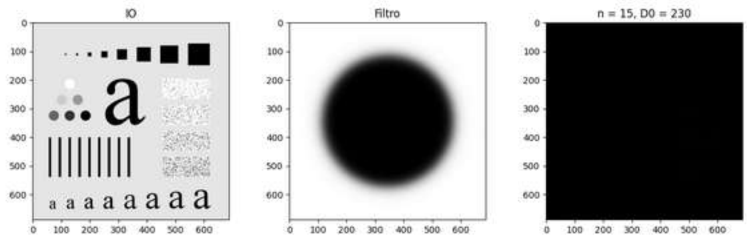

IO
Filtro
n = 10, D0 = 15

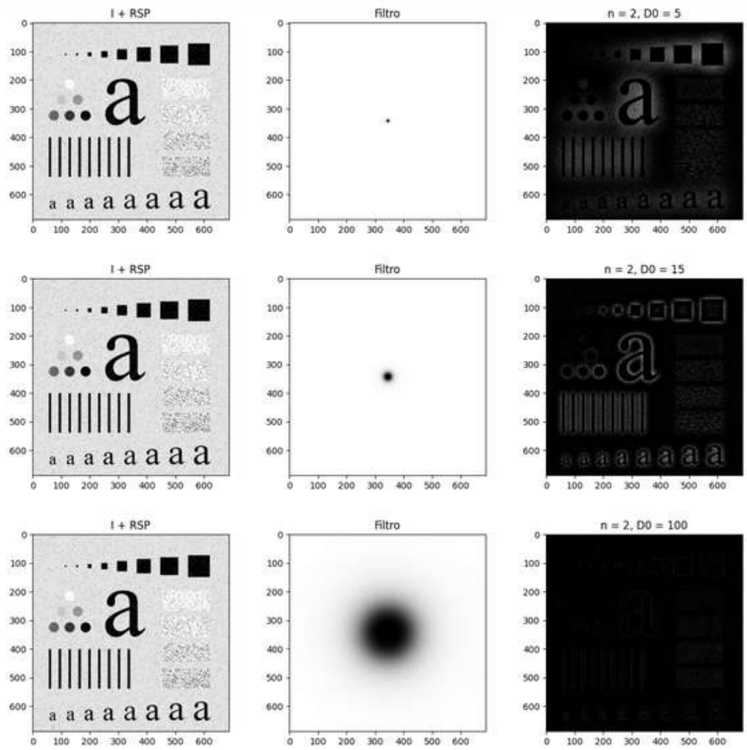

IO
Filtro
n = 10, D0 = 100

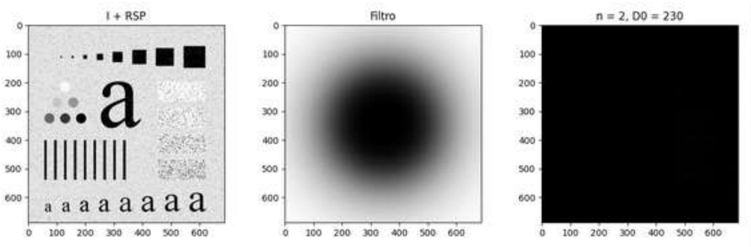

IO
Filtro
n = 10, D0 = 230

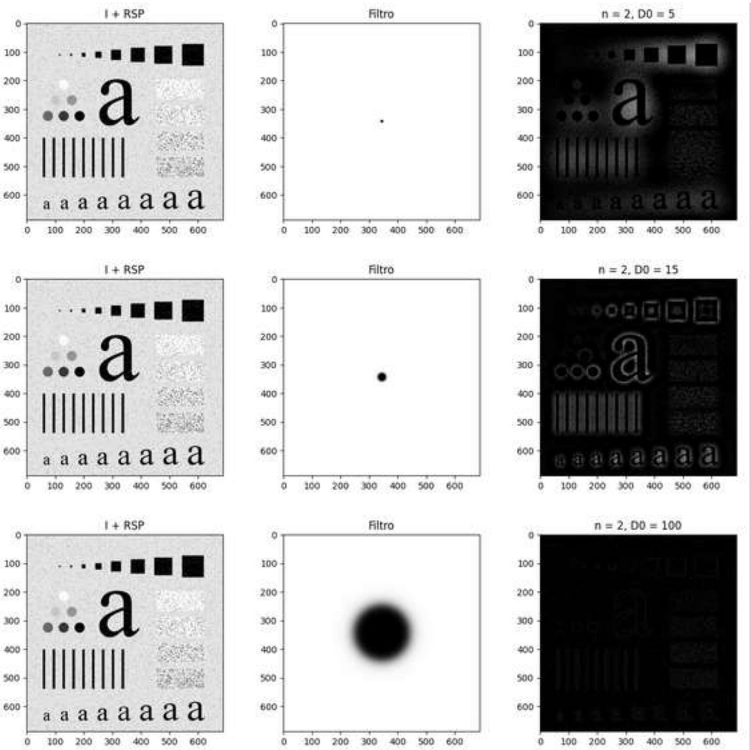

IO
Filtro
n = 15, D0 = 5
IO
Filtro
n = 15, D0 = 15
IO
Filtro
n = 15, D0 = 100

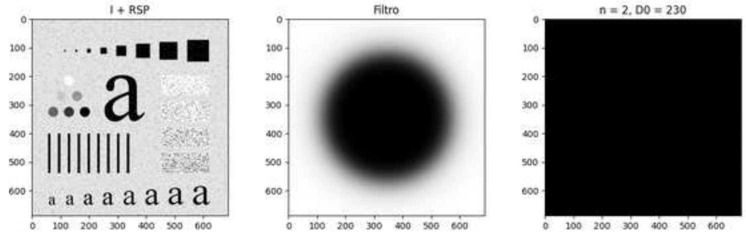

IO
Filtro
n = 15, D0 = 230

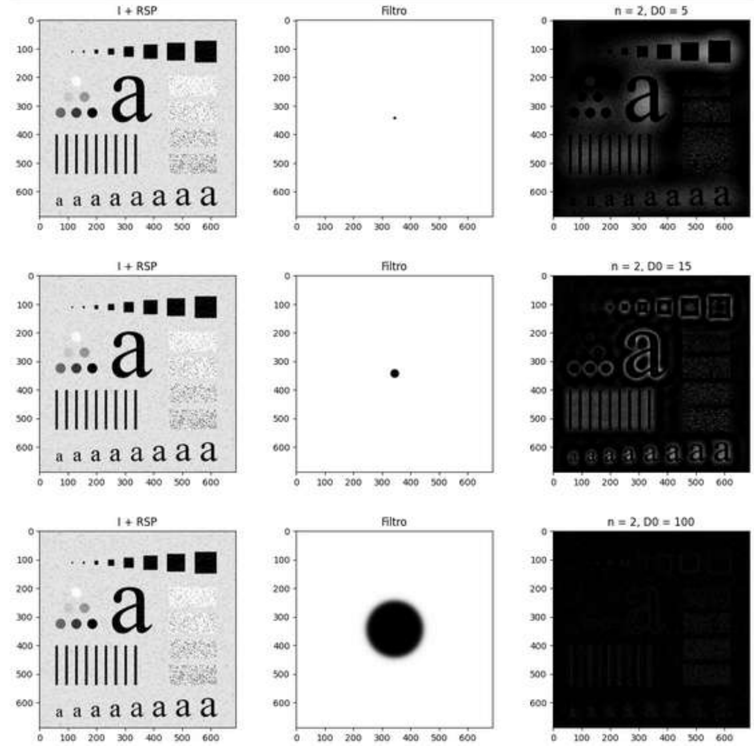

IO
Filtro
n = 15, D0 = 230

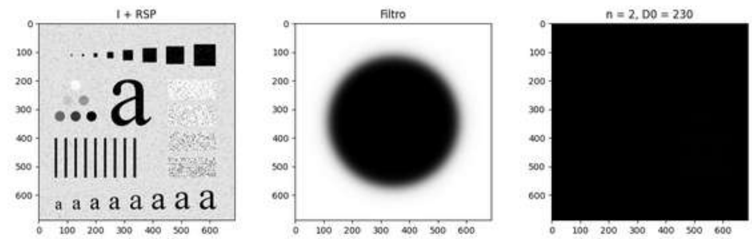

I + RSP
Filtro
n = 2, D0 = 5
I + RSP
Filtro
n = 2, D0 = 15
I + RSP
Filtro
n = 2, D0 = 100

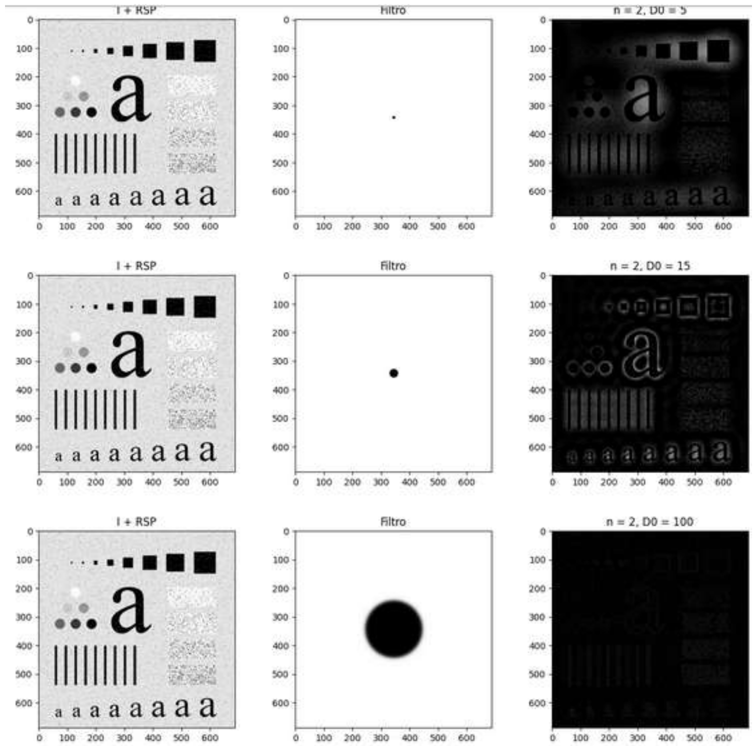

I + RSP
Filtro
n = 2, D0 = 230

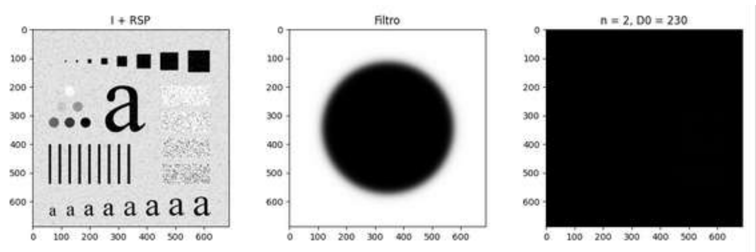

I + RSP
Filtro
n = 2, D0 = 5
I + RSP
Filtro
n = 2, D0 = 15
I + RSP
Filtro
n = 2, D0 = 100


I + RSP
Filtro
n = 2, D0 = 230


I + RSP
Filtro
n = 2, D0 = 5
I + RSP
Filtro
n = 2, D0 = 15
I + RSP
Filtro
n = 2, D0 = 100


I + RSP
Filtro
n = 2, D0 = 230


I + RSP
Filtro
n = 2, D0 = 5
I + RSP
Filtro
n = 2, D0 = 15
I + RSP
Filtro
n = 2, D0 = 100


I + RSP
Filtro
n = 2, D0 = 230

5 Código

Observación :en está práctica se usaron las imágenes de la carpeta de imágenes que proporciono la profesora.

- **NOTA**

En caso de requerir ver o correr de nuevo el código de cada una de las funciones mencionadas en este reporte, el mismo se proporcionará con el nombre **Practica04_PDI.ipynb** junto con las imágenes usadas en la misma y mencionadas en este reporte.

6 Conclusiones

De nuevo puedo decir que la práctica me ayudo demasiado a poder entender este capitulo 4, en general puedo decir que la misma de verdad me ayudo para el examen.
En general pude entender explícitamente el como se aplica la transformada de fourier, sacar la amplitud, fase, los corrimientos; Además de la importancia de tener tanto la fase como la magnitud por que una representa las coordenada en sí y otra las frecuencias, entonces ambas son útiles.
Para el caso del filtro butterwoth, lo vimos demasiado en comparación con el filtro ideal, las cosas que pude notar fueron que :

- En efecto cuando es paso bajas el filtro tiene un hueco blanco en el centro al filtrar solo las frecuencias bajas y cuando es paso altas tiene un hueco negro al filtrar solo las frecuencias altas.
- Entre más crezca D0 aumentará el tamaño de la circunferencia del hueco en el centro y mientras se aumente n entonces se harás mas definido y comportará como un ideal por o que tendrá ciertos anillos alrededor de la imagen, cosa que sabemos pasará dado que Butterwoth es mejor pero cuando n no es tan grande con respecto al filtro Ideal.
- En general cuando :

$$F(u,v)H(u,v)$$

  En el caso de la imagen con y sin ruido no cambia casi nada, es por ello que no seguimos sacando G con la imagen con ruido, dado que es casi igual al que no tiene ruido.
- Nos es bueno para quitar el ruido, en sí podría suavizarlo con el filtro paso bajas por filtrar solo las frecuencias bajas, pero con las altas no ayuda en nada.

Todo lo anterior fueron cosas que puede más notar y que me quedaron más claro el por que es cierto lo visto en teoría.

7 Referencias

- **Gonzalez, R., Woods, R., Digital Image Processing, Prentice Hall, 2008.**