



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

COMPLEJIDAD COMPUTACIONAL (2024-1)

REPORTE



• Vargas Bravo Paola 318074755

FECHA LÍMITE : 11/11/23

Contents

1	Reglas generales para el desarrollo de las Prácticas de Laboratorio	3
2	Objetivos	3
3	Introducción	3
4	Desarrollo	6
5	Código	15
6	Conclusiones	15
7	Referencias	15

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

Como en el realce de imágenes, la meta final de las técnicas de restauración es mejorar la imagen en un sentido predeterminado. A pesar de que existen áreas de sobreexposición, el realce de una imagen es un proceso altamente subjetivo, mientras que la restauración de una imagen es parte de un proceso objetivo.

La restauración intenta reconstruir o recobrar una imagen que ha sido degradada utilizando conocimiento a priori del modelo de degradación y aplicando el proceso inverso a este para poder recobrar así la imagen original.

Si H es lineal y es un proceso invariante a la posición, entonces la imagen degradada está dada, en el dominio espacial por:

$$g(x, y) = h(x, y)f(x, y) + \eta(x, y)$$

donde $h(x, y)$ es la representación espacial de la función de degradación, el símbolo “ $*$ ” indica convolución y $\eta(x, y)$ es ruido aditivo.

- Restauración en presencia de ruido (filtros espaciales).

Las fuentes principales de ruido en las imágenes digitales son durante la adquisición (digitalización) y/o durante la transmisión. El desempeño de los sensores de imágenes es afectado por una variedad de factores, como son las condiciones ambientales durante la adquisición de la imagen y por la calidad de los elementos de sensor.

Cuando la única fuente de degradación en una imagen es ruido, la ecuación anterior se reescribe:

$$g(x, y) = f(x, y) + \eta(x, y)$$

Existen diversos filtros útiles para eliminar el ruido:

- (a) El filtro promedio aritmético calcula el valor promedio de la imagen corrupta $g(x, y)$ en el área S_{xy} . El valor de la imagen restaurada en el punto (x, y) es simplemente el promedio aritmético calculado en esa vecindad:

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

- (b) Una imagen restaurada utilizando un filtro promedio geométrico está dada por la expresión:

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

En este caso, cada píxel restaurado está dado por el producto de los píxeles en la subimagen (ventana), elevado a la potencia $1/mn$.

- (c) Filtro adaptativo cuyo comportamiento cambia según las características de la imagen dentro de la región del filtro definida por una ventana rectangular S_{xy} de tamaño $m \times n$.

Una expresión adaptativa para obtener $f(x, y)$ basada en los supuestos anteriores, puede escribirse como:

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_n^2}{\sigma_L^2} [g(x, y) - m_L]$$

donde σ_n^2 , es la varianza del ruido que corrompe a $f(x, y)$ para formar $g(x, y)$, y m_L , σ_L^2 son la media y la varianza locales de los píxeles en la vecindad S_{xy} , respectivamente. La única cantidad que se necesita conocer o estimar es la varianza del ruido general, σ_n^2 . Los otros parámetros se calculan de los píxeles en S_{xy} en cada posición (x, y) en donde el filtro está centrado.

- (d) El filtro mediana adaptativo también trabaja en una vecindad S_{xy} . Sin embargo, a diferencia de los demás filtros, el filtro mediana adaptativo cambia (incrementa) el tamaño de S_{xy} durante su operación, dependiendo en ciertas condiciones que veremos más adelante.

Recuerde que la salida del filtro es un sólo valor que se utiliza para reemplazar el valor del píxel en la posición (x, y) , del punto central particular de la vecindad S_{xy} . El algoritmo del filtro mediana adaptativo trabaja en dos niveles, denotados nivel A y nivel B como sigue:

Nivel A: $A1 = z_{med} - z_{min}$

$A2 = z_{med} - z_{max}$

Si $A1 > 0$ AND $A2 < 0$, ve al nivel B

si no incrementa el tamaño de la ventana

Si el tamaño de la ventana $\leq S_{max}$ repita el nivel A
 si no salida = z_{xy}

Nivel B: $B1 = z_{xy} - z_{min}$

$$B2 = z_{xy} - z_{max}$$

Si $B1 > 0$ AND $B2 < 0$, salida = z_{xy}

si no salida = z_{med}

donde z_{min} , z_{max} y z_{med} es valor mínimo, máximo y mediana de los niveles de gris de S_{xy} , respectivamente. z_{xy} es el nivel de gris en las coordenadas (x, y) y S_{max} es el valor de tamaño máximo permitido para la ventana S_{xy} .

- (e) Filtro Wiener. Un método que incorpora ambos, la función de degradación y las características estadísticas del ruido, en el proceso de restauración es el llamado filtro Wiener. El método consiste en considerar imagen y ruido como un proceso aleatorio, y el objetivo es encontrar un estimador f de la imagen no-corrupta f de tal manera que el error promedio al cuadrado entre ellas sea mínimo. Este error está dado por:

$$e^2 = E\{(f - \hat{f})^2\}$$

donde E es el valor esperado (la esperanza) del argumento. Se asume que el ruido y la imagen no están correlacionados, que una o la otra tienen media igual a cero, y que los niveles de gris en la estimación son una función lineal de los niveles de la imagen degradada.

Basados en estas condiciones, la función de error mínima está dada en el dominio de la frecuencia por:

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)} \right] G(u, v)$$

donde $H(u, v)$ = función de degradación, $H^*(u, v)$ = conjugado complejo de $H(u, v)$. $|H(u, v)|^2 = H^*(u, v)H(u, v)$. $S_\eta(u, v) = |N(u, v)|^2$ = espectro de potencia del ruido. $S_f(u, v) = |F(u, v)|^2$ = espectro de potencia de la imagen no degradada.

El filtro que consiste en los términos dentro de los corchetes, también se conoce como filtro de error promedio mínimo al cuadrado, o filtro de error de mínimos cuadrados.

4 Desarrollo

- **Ejercicio 1 :** Utiliza la imagen del circuito, genera ruido gaussiano aditivo con media cero y desviación estándar de 0.04. Filtra la imagen primero con un filtro promedio aritmético de tamaño 3×3 , y luego filtrala con un filtro promedio geométrico del mismo tamaño Comparalos (genera así la figura 5.7 del libro de Gonzalez et. al.)

Función : filtro_aritmetico

Parámetros : orden,image

- La técnica que usamos para el tratamiento de bordes es la técnica de **reducción de la imagen**. La explicaremos en esta función un poco más detalle pero a lo largo de este reporte se asumirá esto.

La aplicación se divide en dos partes :

- * Resta de orden : La resta en *ancho x largo* de la imagen original para la imagen objetivo.
- * Suma orden : La suma en las coordenadas dentro de la imagen objetivo para obtener su valor correspondiente en la imagen original.

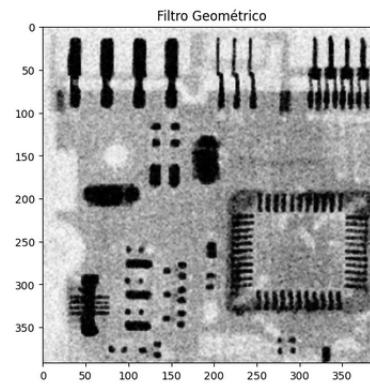
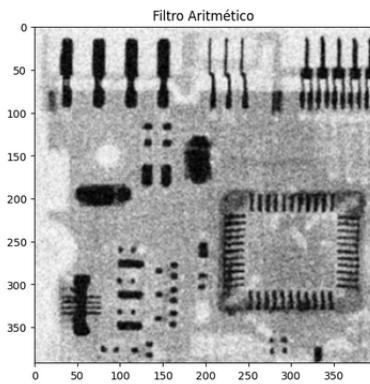
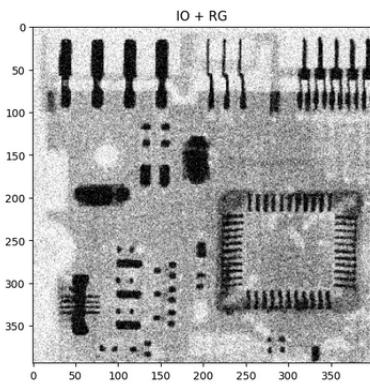
La suma de orden y resta depende del orden $n \times n$ del filtro .

- Creamos la imagen objetivo de acuerdo a la resta de orden, en escala de grises
- De acuerdo al orden llamamos a nuestra función **vecinos_nxn_2**, la cual nos da el área de vecinos de un píxel de acuerdo al orden del filtro que queremos, en específico el nivel de gris de esos píxeles del área.
- Pasamos estos vecinos, a la imagen original, un valor $n = \text{orden} * \text{orden}$, a la función auxiliar de convolución , donde cada píxel dentro de los vecinos de ese punto obtenidos de la imagen original se suman y se dividen entre n , donde el resultado de ello será el nuevo valor de la coordenada (x, y) .
- Regresamos la imagen objetivo.

Función : filtro_geometrico

Parámetros : orden,image

- Creamos la imagen objetivo de acuerdo a la resta de orden en escala de grises.
 - De acuerdo al orden llamamos a nuestra función **vecinos_nxn_2**, la cual nos da el área de vecinos de un píxel de acuerdo al orden del filtro que queremos, en sí el valor de esos vecinos del área que queremos.
 - Pasamos estos vecinos a la función auxiliar de convolución , donde cada píxel dentro de los vecinos de ese punto obtenidos de la imagen original se multiplican entre ellos mismos y al final se eleva a la $\frac{1}{n}$, donde $n = \text{orden} * \text{orden}$, tomando en cuenta que excluimos valores de nivel de gris iguales a cero.
- El resultado será el nuevo valor del píxel en la coordenada (x, y) .
- Regresamos la imagen objetivo.



- **Ejercicio 2** : Utiliza nuevamente la imagen circuito, genera ruido gaussiano aditivo de media cero y desviación estándar de 0.04. Filtrala primero con un filtro promedio aritmético de tamaño 7×7 . Filtrala ahora con un filtro geométrico del mismo tamaño. Finalmente filtrala con un filtro adaptativo para reducción de ruido del mismo tamaño que los anteriores.

Comparalos (genera así la figura 5.13 del libro de Gonzalez et.al.).

Función : filtro_adaptativo

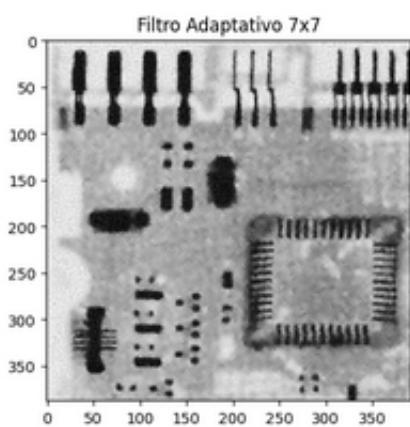
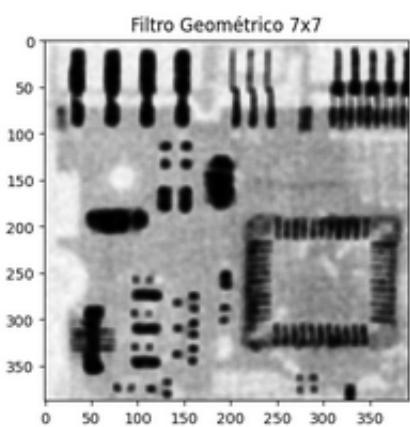
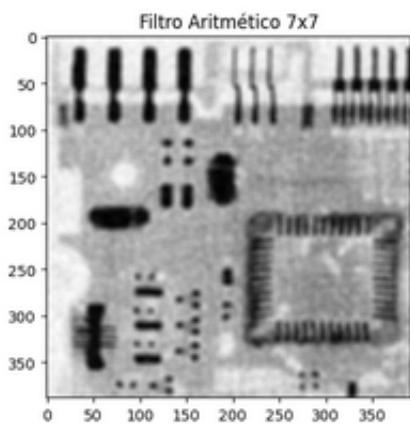
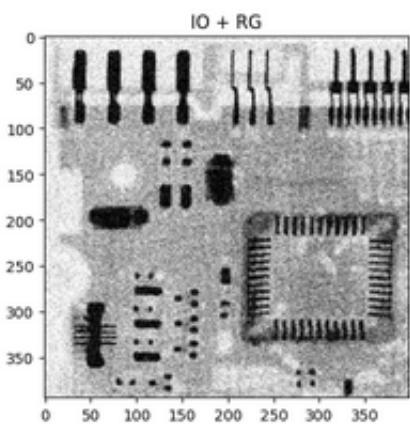
Parámetros : orden,image,varianzaN

- Creamos la imagen objetivo de acuerdo a la resta de orden en escala de grises.
- De acuerdo al orden llamamos a nuestra función `vecinos_nxn_2`, la cual nos da el área de vecinos de un píxel de acuerdo al orden del filtro que queremos, en sí el valor de esos vecinos del área que queremos.
- Pasamos estos vecinos a la funciones auxiliares para obtener :
 - * La media Local
 - * La varianza Local
 - * La expresión aritmética, la cual es :

$$f(x, y) = g(x, y) - \frac{\sigma_n^2}{\sigma_L^2} [g(x, y) - m_L]$$

El resultado será el nuevo valor del píxel en la coordenada (x, y) .

- Regresamos la imagen objetivo.



- **Ejercicio 3 :** Utiliza la imagen circuito, genera un ruido sal y pimienta aditivo con probabilidades $Pa = Pb = 0.25$. Filtrala primero con un filtro mediana de tamaño 7×7 . Filtrala ahora con un filtro mediana adaptativo con $Smax = 7$. Comparalos (genera así la figura 5.14 del libro de Gonzalez et. al.).

Función : filtro_mediana_adaptativo

Parámetros : image, orden

- Creamos la imagen objetivo de acuerdo a la resta de orden en escala de grises.
- Entramos en un ciclo por cada (x, y) de forma que de ser necesario solo podríamos salir del mismo solo si llegamos a un caso base o a un orden de filtro de 7. Entonces de acuerdo al orden llamamos a nuestra función `vecinos_nxn_2`, la cual nos da el área de vecinos de un píxel de acuerdo al orden del filtro que queremos, en sí el valor de gris de esos píxeles en la vecindad.
- Pasamos estos vecinos a las funciones auxiliares para obtener :
 - * zMin
 - * zMax
 - * zMed

Verificamos los siguientes casos :

* Nivel A :

- $A1 = zmed - zmin$
- $A2 = zmed - zmax$
- Si $A1 > 0$ AND $A2 < 0$, ve al nivel B si no incrementa el tamaño de la ventana Si el tamaño de la ventana $\leq Smax$ repita el nivel A si no salida = zxy

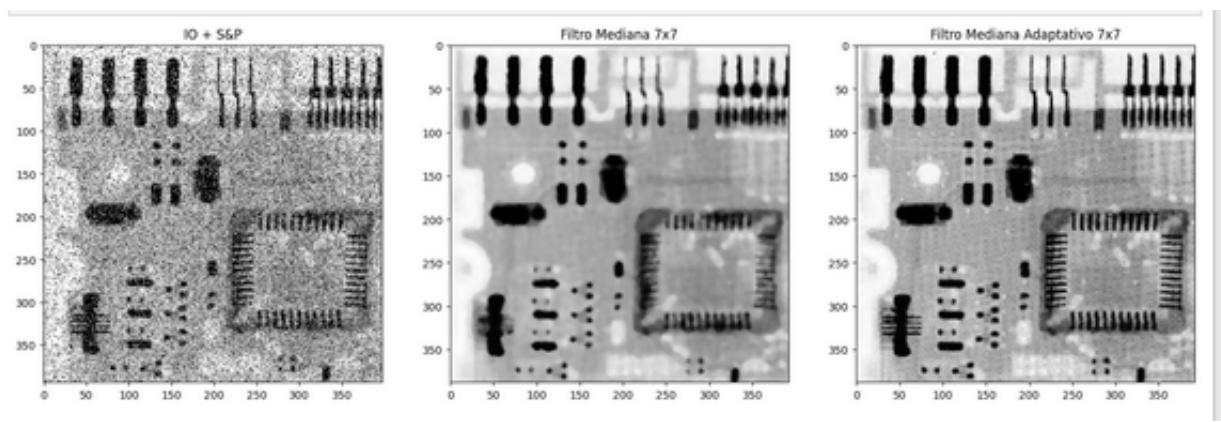
* Nivel B :

- $B1 = z_{xy} - zmin$
- $B2 = z_{xy} - zmax$
- Si $B1 > 0$ AND $B2 < 0$, salida = zxy si no salida = $zmed$

Donde de acuerdo a lo anterior y dependiendo de cada píxel su nuevo valor está en dos posibilidades :

- z_{xy}
- $zMed$

- Regresamos la imagen objetivo



- **Ejercicio 4,5,6 :** Resumimos estos cuatro ejercicios en uno solo, dado que es la programación del filtro Wiener, en sus cuatro casos.

Función : filtro_wiener_general

Parámetros : orden,image,caso

- De forma general, definimos r como la resta de la imagen con ruido/pérdida de nitidez o ambas digamos imagen modificada, con la imagen original, donde a r, la imagen original y la imagen modificada, se le saca la transformada de fourier, y a su vez su parte real.
- Tenemos dentro del mismo cuatro casos :

* CASO 1 :

$$W(u, v) = \frac{S_{ff}(u, v)}{S_{ff}(u, v) + S_{nn}(u, v)}$$

* CASO 2 :

$$W(u, v) = \frac{1}{H(u, v)}$$

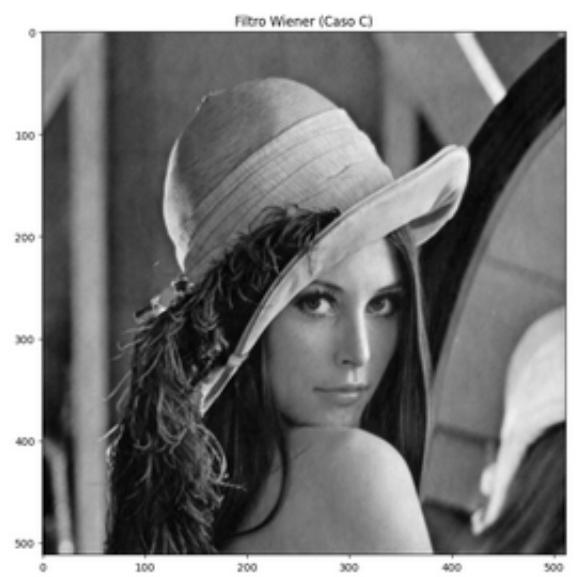
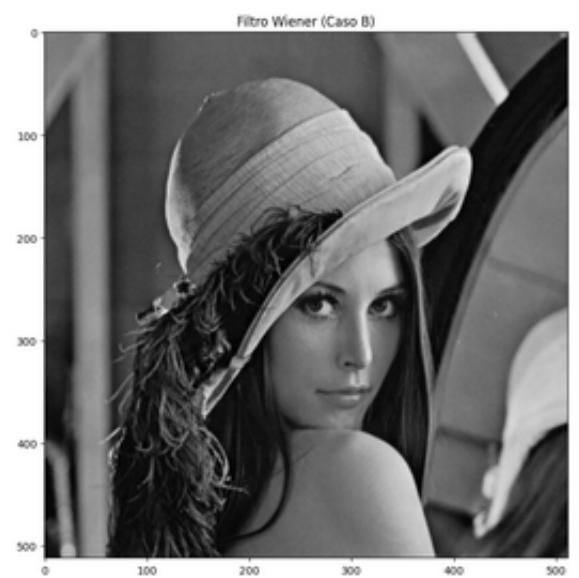
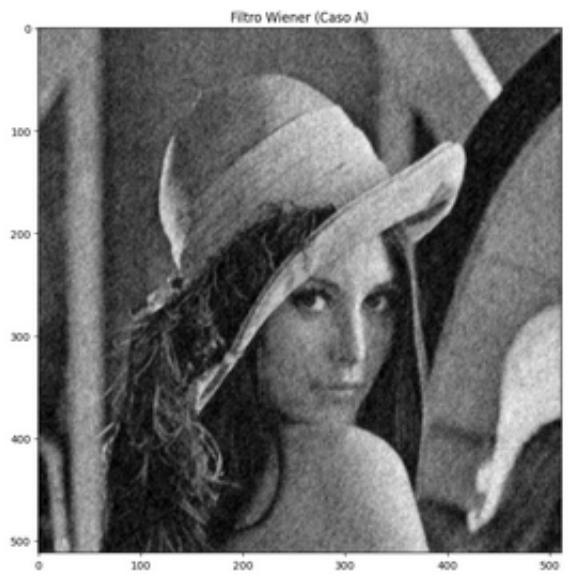
* CASO 3 :

$$W(u, v) = \frac{S_{ff}(u, v)}{H(u, v)(S_{ff}(u, v) + S_{nn}(u, v))}$$

* CASO 4 :

$$W(u, v) = \frac{S_{ff}(u, v)H^*(u, v)}{H(u, v)S_{ff}(u, v)H^*(u, v) + S_{nn}(u, v)}$$

- W lo multiplicamos por la transformada inversa de la imagen con ruido, que llamamos G, a la misma se le calcula la transformada inversa , su parte real, y un clip de 0 a 255.
- Regresamos G.



IO + RG + PN



Filtro Wiener (Caso D)



5 Código

Observación :en esta 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 **Practica05_PDI.ipynb** junto con las imágenes usadas en la misma y mencionadas en este reporte.

6 Conclusiones

La práctica termino de hacerme comprender los filtros para la restauración de la imagen, en particular pude notar los siguientes puntos :

- El filtro geométrico marca más los bordes y es mejor contra el aritmético.
- El filtro adaptativo hace un efecto de acuarela a la imagen que hace que se vea mejor con respecto a las imágenes con filtro geométrico y aritmético.
- El filtro mediana adaptativo conserva detalles y elimina mejor el ruido sal y pimienta que el filtro mediana.
- Parece ser que cuando una imagen tiene perdida de nitidez, perdida de nitidez + ruido gaussiano, hace que para estos casos funciones mejor el filtro wiener que para el caso 1 y caso 3.

Me hizo entender más este concepto de restauración y el como en efecto en la práctica 02 solo se hacía un emborronamiento, pero en este caso hacemos una gran obtención de la imagen original y es por ello que filtros que aunque lo pudieran quitar como el mediana en la práctica 02 ,realmente hay mejores filtros que siguen conservando detalles.

7 Referencias

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