

Reporte Practica 5

Caballero Jimn ez Oscar Emilio

November 2023

1. Objetivos

- Implementar el filtro promedio aritm tico, geom trico y adaptativo. Comparar su desempe o en presencia de ruido.
- Implementar el filtro mediana adaptativo y comparar su desempe o comparado con el filtro mediana simple, en presencia de ruido.
- De acuerdo al modelo de degradaci n que sufre una serie de im genes, encontrar y aplicar el filtro de Wiener adecuado para la restauraci n  ptima de cada imagen.

2. Introducci n

Como en el realce de im genes, la meta final de las t cnicas de restauraci n es mejorar la imagen en un sentido predeterminado. A pesar de que existen  reas de solapamiento, el realce de una imagen es un proceso altamente subjetivo, mientras que la restauraci n de una imagen es parte de un proceso objetivo.

La restauraci n intenta reconstruir o recuperar una imagen que ha sido degradada utilizando conocimiento a priori del modelo de degradaci n y aplicando el proceso inverso a este para poder recuperar as  la imagen original. Si H es lineal y es un proceso invariante a la posici n, entonces la imagen degradada est  dada, en el dominio espacial por:

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

donde $h(x, y)$ es la representaci n espacial de la funci n de degradaci n, el s mbolo “*” indica convoluci n y $\eta(x, y)$ es ruido aditivo.

- Restauraci n en presencia de ruido (filtros espaciales).
Las fuentes principales de ruido en las im genes digitales son durante la adquisici n (digitalizaci n) y/o durante la transmisi n. El desempe o de los sensores de im genes es afectado por una variedad de factores, como son las condiciones ambientales durante la adquisici n 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:

1. 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:

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

2. 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 $1/mn$.

3. **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 $\hat{f}(x, y)$ basada en los supuestos anteriores, puede escribirse como:

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

donde σ_{η}^2 es la varianza del ruido que corrompe a $f(x, y)$ para formar $g(x, y)$, y m_L y 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_{η}^2 . Los otros parámetros se calculan de los píxeles en S_{xy} en cada posición (x, y) en donde el filtro está centrado.

- 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 solo 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: $A1 = z_{\text{med}} - z_{\text{min}}$

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

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

Si no, incrementa el tamaño de la ventana.

Si el tamaño de la ventana $\leq S_{\text{max}}$, repita el nivel A.

Si no, la salida es z_{xy} .

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

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

Si $B1 > 0$ AND $B2 < 0$, la salida es z_{xy} .

Si no, la salida es z_{med} .

donde z_{min} , z_{max} y z_{med} son el 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} .

- **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 \hat{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:

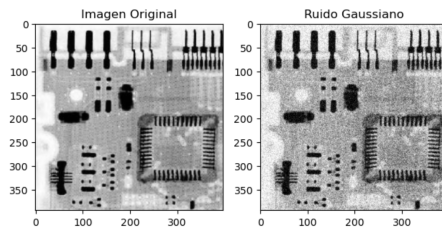
$$e^2 = E \left\{ (f - \hat{f})^2 \right\}$$

donde $E\{\cdot\}$ es el valor esperado (la esperanza) del argumento. Se asume que el ruido y la imagen no están correlacionados, que una u 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.

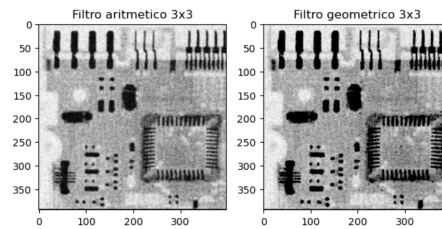
Basados en estas condiciones, la función de error mínima está dada en el dominio de la frecuencia por:

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

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



(a) Ejercicio 1 primera imagen



(b) Ejercicio 1 segunda imagen

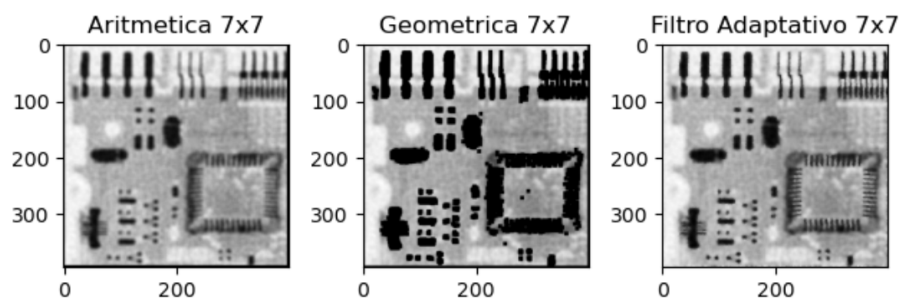
El filtro que consiste en los términos dentro de los corchetes también se conoce como **filtro de error promedio mínimo al cuadrado**, o **filtro de error de mínimos cuadrados**.

3. Desarrollo

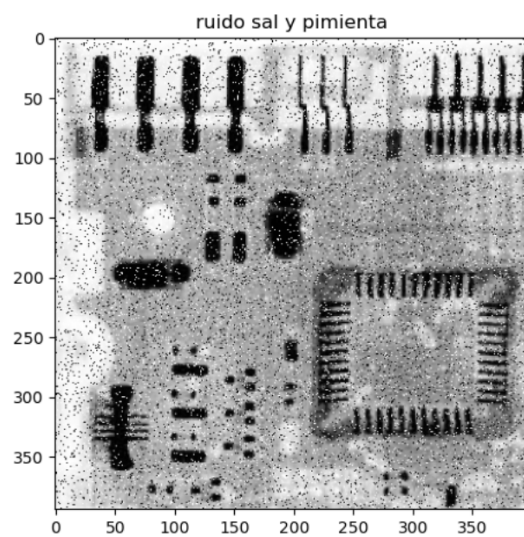
Resuelve los problemas de la lista siguiente y describe tu solución en cada inciso. Los incisos en donde únicamente tengas que desplegar imágenes no requieren de ninguna descripción.

1. Utiliza la imagen del circuito, génerele 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. Compáralos
2. Utiliza nuevamente la imagen circuito, génerele 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. Compáralos
3. Utiliza la imagen circuito, génerele un ruido sal y pimienta aditivo con probabilidades $P_a = P_b = 0.25$. Filtrala primero con un filtro mediana de tamaño 7×7 . Filtrala ahora con un filtro mediana adaptativo con $S_{max} = 7$. Compáralos.

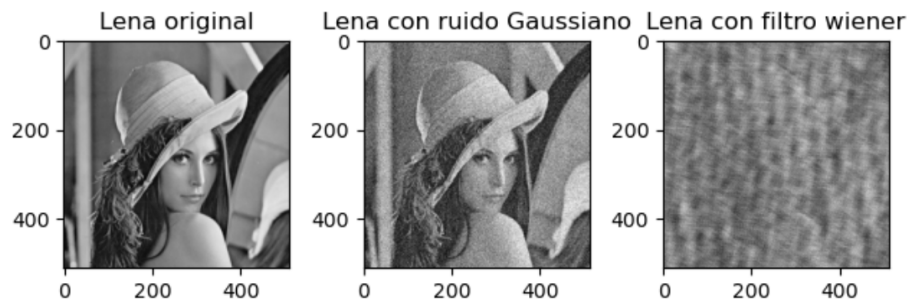
Este ejercicio no me salio, solo logre ponerle ruido sal y pimienta a la imagen



(a) Ejercicio 2



(a) Ejercicio 3



(a) Ejercicio 4



(a) Ejercicio 5

4. Para la imagen lena con ruido aditivo de tipo gaussiano, encontrar el filtro de Wiener y restaurar la imagen.

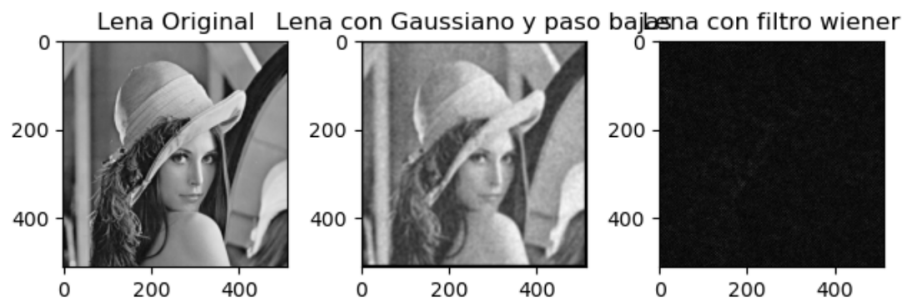
Por alguna razon, el filtro wiener no me salio como debia, a pesar de haber reajustado la imagen.

5. Encontrar el filtro de Wiener y restaurar una imagen lena que ha sido sometida a un proceso de pérdida de nitidez. La imagen con perdida de nitidez se obtiene filtrando una imagen nítida y libre de ruido con un filtro paso bajas de tamaño 9x9 normalizado (filtro promedio ponderado).

En el código se logra ver un poco mejor la imagen con filtro wiener

6. Para una imagen lena a la que se le ha agregado ruido de tipo gaussiano y posteriormente ha perdido nitidez, encontrar el filtro de Wiener y restaurarla.

Lo mismo, en el código se ve mejor el filtro wiener.



(a) Ejercicio 6

4. Código

El código se encuentra en la carpeta *src* o en mi repo de github, ahí también está público.

Usuario de GitHub: @DexenRoss

5. Conclusiones

Para esta práctica la verdad, a pesar de reciclar código, tuve que trabajar extra debido a que no hice bien la convolución en la práctica dos, a pesar de ello, creo que logré sacar adelante esta práctica.

Con respecto a los objetivos, en la parte práctica, creo no se lograron al 100 %, pero en la parte teórica creo que entiendo el concepto del filtro wiener, ya que este ayuda y/o sirve para restaurar imágenes que tienen ruido, haciéndolas más nítidas o limpiándolas del propio ruido.

6. Referencias

1. Gonzalez, R., Woods, R., Digital Image Processing, Prentice Hall, Second edition, 2002.
2. La ayuda del ayudante Miguel Angel Veloz Lucas.