PRÁCTICA 4

"Simulación y Filtrado de Ruidos en Imágenes"

Objetivo: simulación y filtrado de ruidos en imágenes; evaluación de distintos tipos y opciones de filtrado.

Instrucciones Matlab:

- `randn:` obtiene una matriz cuyos valores son generados por una distribución gaussiana de media nula y varianza 1 ($N(0,1)$).

  Utilizando esta instrucción se puede obtener una matriz cuyos valores se ajusten a una distribución gaussiana de media μ y varianza σ^2 ($N(\mu,\sigma^2)$):

$$N(0,1) = \frac{N(\mu,\sigma^2) - \mu}{\sigma} \quad \rightarrow \quad N(\mu,\sigma^2) = \mu + \sigma N(0,1)$$

- `rand`: permite obtiene una matriz cuyos valores son generados por una distribución uniforme en el rango (0,1).

- `fspecial:` para crear máscaras de aplicación en filtrado lineal.

- `medfilt2:` para filtrar una imagen digital mediante filtro de la mediana.

- `ordfilt2`: para filtrar una imagen digital mediante cualquier filtro de orden.

- `stdfilt:` para obtener la desviación estándar en cada píxel de una matriz, calculada en una vencidad especificada por parámetro.

- `mean` ; `median; std`: para calcular la media, mediana y desviación estándar.

La práctica se ha dividido en tres partes:

1. **Primera parte:** simulación de ruidos tipo gaussiano y *sal y pimienta*.
2. **Segunda parte:** implementación de filtros gaussiano, mediana y adaptativo.
3. **Tercera parte:** evaluación de eficiencia de filtros gaussiano, mediana y adaptativo.

PRIMERA PARTE: Simulación de ruidos.

1. Lee la imagen "P4.tif", que es una imagen en escala de gris.

2. Corrompe la imagen anterior con ruido de tipo de sal y pimienta y de tipo gaussiano para generar, tal y como se describe a continuación:

   a. Sal y pimienta, con $p = 0.9$ y $q = 0.95$ (ver ecuación). La imagen corrompida con este ruido (imagen A) se puede calcular de la siguiente manera:

   $$A(i,j) = \begin{cases} I(i,j) & x < p \\ 0 & p \le x < q \\ 255 & q \le x < 1 \end{cases}$$ siendo I la imagen original sin corromper, x una variable aleatoria uniforme de rango (0,1), q-p el porcentaje de píxeles con ruido de tipo pimienta y 1-q el porcentaje de píxeles con ruido de tipo sal.

   b. Gaussiano de media nula y desviación típica 10.

3. Visualiza las imágenes ruidosas A y B. Representa en un mismo gráfico la variación de los niveles de gris a lo largo de la línea horizontal central para la imagen original, para la imagen A con ruido de tipo sal y pimienta, y para la imagen B con ruido gaussiano. Observa las distintas distribuciones del ruido.

SEGUNDA PARTE: Implementación de filtros gaussiano, mediana y adaptativo.

Esta segunda parte de la práctica se debe realizar sobre la imagen corrompida con ruido sal y pimienta generada en la primera parte de la práctica

Aplica los siguientes filtros, visualiza las imágenes filtradas y discuta los resultados obtenidos:

a. Un filtro gaussiano con W = 5 σ, siendo σ la desviación típica del filtro y W el tamaño del filtro (considera un valor W=5). Para ello:

   - Crea la máscara del filtro gaussiano, implementada en Código Matlab (ver documentación teórica - Tema 3) y utilizando función `fspecial`. Comprueba que se obtienen los mismos resultados.
   - Utiliza la función `imfilter` para realizar la convolución.

b. Un filtro de la mediana considerando un entorno de vecindad 5x5. Para ello, implementa la función (ver observación):

```
Ifiltrada = Funcion_FiltroMediana (I, matriz_vecindad,
                    'opcion_padding')
```

   - `I:` imagen de entrada.
   - `matriz_vecindad`: matriz de 0's y 1's que define el entorno de vecindad. La posición de los 1's en la matriz indica los píxeles sobre los que se debe realizar la operación cuando la matriz se centra en el píxel en cuestión. Se debe asumir que esta matriz tiene un número impar de filas y columnas.
   - `'opcion_padding':` opciones válidas 'zeros', 'replicate' y 'symmetric'.
   - `Ifiltrada`: imagen filtrada de salida.

   Comprueba mediante la función "compara matrices" que se obtiene la misma imagen de salida que genera la función de Matlab `medfilt2`.

c. Un filtro adaptativo que actúe en un entorno de vecindad 7x7.

Para ello, implementa la función (ver observación):

```
Ifiltrada = Funcion_FiltAdapt (I, matriz_vecindad, VarRuido,
                               'opcion_padding')
```

- `VarRuido`: estimación de la varianza del ruido presente en la imagen.

Deben implementarse dos versiones de la función, una mediante programación píxel a píxel y otra mediante programación matricial. En este último caso, utilizar en la implementación las funciones `imfilter` y `stdfilt`.

Comprueba mediante la función "compara matrices" que ambas versiones de la función conducen a los mismos resultados. Analice y justifique el resultado obtenido.

TERCERA PARTE: Evaluación de eficiencia de filtros gaussiano, mediana y adaptativo.

Esta tercera parte de la práctica se debe realizar sobre la imagen "P4.tif"

1. A partir de la imagen "P4.tif", genera tres imágenes con ruido gaussiano de media nula y desviaciones típicas, por cada imagen generada, de 5, 10 y 35.

2. Filtra cada una de las imágenes ruidosas anteriores con filtros de tipo gaussiano, de tipo mediana y adaptativo, considerando tamaños 3x3 y 7x7 para cada filtro. Visualiza las distintas imágenes ruidosas y filtradas.

3. Evalúa la eficiencia de cada proceso de filtrado midiendo la relación señal-ruido (ISNR), definida como:

$$\mathrm{ISNR}_{\mathrm{dB}} = 10\log_{10}\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\mathbf{I}(i,j)-\mathbf{G}(i,j)\right)^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\mathbf{I}(i,j)-\mathbf{I_e}(i,j)\right)^2}$$

donde I es la imagen original sin ruido, G la imagen ruidosa, e I_e es la imagen filtrada.

4. Presenta los resultados de ISNR en una tabla con el siguiente formato:

SigmaRuido	TamagnoVentana	FiltroGauss	FiltroMediana	FiltroAdaptativo
5	3			
5	7			
10	3			
10	7			
35	3			
35	7			

5. A partir de los resultados obtenidos, realiza un informe de conclusiones.

Observación: realizar una programación eficiente que contemple vectores con los valores de desviaciones típicas de ruido y tamaño de ventanas de vecindad considerados y que permita generar la tabla de resultados de forma automática.

Aplicación filtro temporal

6. Genera, a partir de la imagen inicial, 10 imágenes ruidosas con ruido blanco gaussiano y desviación típica 35. Visualiza una de estas imágenes.

7. Aplica un promediado a estas imágenes y observa la imagen resultante.