

Trabajo práctico 3

Filtrado espacial

Ejercicio 1:

Utilizando la función `noise()`, agregar ruido aditivo a las imágenes `cameraman.tif`, `curves.gif`, `kit.tif`, `imprueba.jpg` y `circuito.tif` con las siguientes características:

- ruido gaussiano: media 0 y desvío 10, 30, 50 y 100 para toda la imagen
- ruido uniforme: ruido máximo 10%, 20% y 30% del rango de intensidades, para el 10%, 50% y 100% de la imagen
- ruido sal, pimienta y sal y pimienta: 1%, 5%, 10%, 40% de la imagen utilizando la función `imnoise()` del toolbox de imágenes

Ejercicio 2:

Implementar una función Matlab que evalúe la convolución entre una imagen y un kernel dado de $m \times n$ elementos. Usar esta función para filtrar la imagen `cameraman.tif` con el kernel $k = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 9$. Comparar los resultados obtenidos con la aplicación de las funciones `conv2()`, `imfilter()` y `fspecial()` provistas por Matlab.

Ejercicio 3:

Para las imágenes usadas en el ejercicio 1, aplicar los siguientes filtros usando las funciones `imfilter()` y `fspecial()`, para kernels de 3x3, 5x5, 7x7 y 9x9:

- filtro promedio: rectangular y circular
- filtro gaussiano: con desvíos 0.3, 1, 3.32, 5

Comparar los resultados obtenidos respecto a la efectividad en la remoción de ruido y la conservación de detalles originales de la imagen.

Ejercicio 4:

Para las imágenes usadas en el ejercicio 1, aplicar los siguientes filtros pasa altos usando la función `imfilter()`, para kernels de 3x3, 5x5, 7x7 y 9x9:

- filtro para agudizar detalles: elemento central $w=n-1$ siendo n la cantidad de elementos
- filtro para enfatizar altas frecuencias: elemento central $w=nA-1$, $A=2, \dots, 10$

Comparar los resultados obtenidos respecto a la efectividad para resaltar detalles y la influencia del ruido presente en la imagen.

Ejercicio 5:

Para las imágenes usadas en el ejercicio 1, aplicar los siguientes filtros derivativos usando las funciones `imfilter()` y `fspecial()`:

- filtro de derivada
- operadores alternativos de Roberts, Prewitt, Sobel
- operador laplaciano con parámetro de forma 0 (eps), 0.1, 0.5, 0.8 y 1.

Comparar los resultados obtenidos respecto a la efectividad para resaltar detalles y la remoción del ruido presente en la imagen.

Ejercicio 6:

Implementar los siguientes filtros no lineales usando Matlab:

- filtro de mediana: usar la función `medfilt2()`

- filtro promedio alfa recortado: usar la función `sort()`

- filtro promedio geométrico:

$$g(x, y) = [\prod_{i, j \in W} f(x + i, y + j)]^{1/N} \text{ con } N \text{ la cantidad de pixels del kernel } W$$

- filtro promedio potencial de orden p :

$$g(x, y) = \left[\frac{\sum_{i, j \in W} f(x + i, y + j)^p}{N} \right]^{1/p} \text{ con } N \text{ la cantidad de pixels del kernel } W$$

- filtro promedio armónico:

$$g(x, y) = \frac{N}{\sum_{i, j \in W} \frac{1}{f(x + i, y + j)}} \text{ si } f(x + i, y + j) \neq 0 \text{ para todos los pixels de la}$$

imagen dentro de la máscara de filtrado o 0 en otro caso

- filtro contraarmónico de orden p :

$$g(x, y) = \frac{\sum_{i, j \in W} f(x + i, y + j)^{p+1}}{\sum_{i, j \in W} f(x + i, y + j)^p} \text{ si } f(x + i, y + j) \neq 0 \text{ para algún píxel de la}$$

imagen dentro de la máscara de filtrado o 0 en otro caso

- filtros de máximo y mínimo

$$g_{\max}(x, y) = \max_{i, j \in W} [f(x + i, y + j)] \text{ y } g_{\min}(x, y) = \min_{i, j \in W} [f(x + i, y + j)]$$

Aplicar los filtros desarrollados a las imágenes anteriores y analizar su desempeño, considerando kernels W de 3x3, 5x5, 7x7 y 9x9 elementos.

Ejercicio 7:

Implementar el filtro adaptativo que minimiza el error cuadrático medio y analizar su desempeño en el filtrado de la imagen cameraman.tif contaminada con ruido aditivo gaussiano de media 0 y desvíos 10, 30, 50 y 100. Considerar el caso de kernels de 3x3, 5x5, 7x7 y 9x9 elementos. Aplicar este filtro con una de las imágenes anteriores estimando el contenido de ruido.