



Pontificia Universidad
JAVERIANA
Bogotá

Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Imágenes Médicas
Ejercicios: Filtrado espacial, 2023-30

Objetivo

Utilizar MATLAB para interactuar de forma práctica con conceptos de ruido y filtrado espacial de imágenes.

Desarrollo del ejercicio

El desarrollo del ejercicio práctico consistirá en generar un informe escrito (en formato PDF), el cual debe enviarse a través de la respectiva asignación de BrightSpace. En el informe debe incluirse, como respuesta a cada ejercicio, el código fuente solicitado e impresiones de pantalla del resultado obtenido con ese código.

1. Generación de imágenes con ruido

El siguiente código (adaptado de <https://www.geeksforgeeks.org/what-are-different-types-of-denoising-filters-in-matlab/>) presenta diferentes opciones para agregar ruido a una imagen en escala de grises en MATLAB:

```
% cargar la imagen
img=imread("BrainMR.png");
% convertir a escala de grises
grayimg=im2gray(img);
% agregar ruido Gaussiano a la imagen
gauss_img=imnoise(grayimg,'gaussian',0,0.02);
% agregar ruido de Poisson a la imagen
poiss_img=imnoise(grayimg,'poisson');
% agregar ruido de sal y pimienta a la imagen
salpe_img=imnoise(grayimg,'salt & pepper', 0.05);

% visualizar la imagen original y sus versiones con ruido
figure(1)
subplot(2,2,1)
imshow(grayimg);
subplot(2,2,2)
imshow(gauss_img);
subplot(2,2,3)
imshow(poiss_img);
subplot(2,2,4)
imshow(salpe_img);

% guardar las imagenes con ruido
imwrite(gauss_img,"BrainMR_gauss.png");
imwrite(poiss_img,"BrainMR_poiss.png");
```

```
imwrite(salpe_img,"BrainMR_salpe.png");
```

- **Ejercicio 1**

Ejecute el código para obtener las diferentes versiones con ruido de la imagen BrainMR.png (adjunta a este enunciado), e incorpore los resultados obtenidos a su reporte.

- **Ejercicio 2**

Ejecute el código anterior con otra imagen (médica) diferente, para obtener sus versiones con ruido diferente, e incorpore la imagen utilizada y los resultados obtenidos a su reporte.

2. Remoción del ruido en las imágenes

El siguiente código (adaptado de <https://www.geeksforgeeks.org/what-are-different-types-of-denoising-filters-in-matlab/>) presenta diferentes opciones para suavizar una imagen en escala de grises con ruido en MATLAB, comparando los resultados obtenidos con respecto a la imagen original (sin ruido):

```
% cargar la imagen con ruido
img=imread("BrainMR_gauss.png");
% convertir a escala de grises
grayimg=im2gray(img);

% crear un kernel para filtro promedio 3x3
h = [1/9 1/9 1/9; 1/9 1/9 1/9; 1/9 1/9 1/9];
% filtrar la imagen con filtro promedio
prom_filt = uint8(conv2(grayimg,h,'same'));

% filtrar la imagen con filtro mediana
med_filt = medfilt2(grayimg);

% filtrar la imagen con filtro Gaussiano, sigma = 1
gauss_filt_1 = imgaussfilt(grayimg,1);

% visualizar la imagen con ruido y sus versiones filtradas
figure(1)
subplot(2,2,1)
imshow(grayimg);
subplot(2,2,2)
imshow(prom_filt);
subplot(2,2,3)
imshow(med_filt);
subplot(2,2,4)
imshow(gauss_filt_1);

% guardar las imagenes filtradas
imwrite(prom_filt,"BrainMR_gauss_prom.png");
imwrite(med_filt,"BrainMR_gauss_med.png");
imwrite(gauss_filt_1,"BrainMR_gauss_filt_1.png");

% cargar la imagen original
img2=imread("BrainMR.png");
% convertir a escala de grises
grayimg2=im2gray(img2);
% calcular las diferencias con respecto a la imagen original
```

```

diff1 = grayimg2-grayimg;
num_diff1 = sum(diff1(:)==0) / size(diff1(:),1)
diff2 = grayimg2-prom_filt;
num_diff2 = sum(diff2(:)==0) / size(diff2(:),1)
diff3 = grayimg2-med_filt;
num_diff3 = sum(diff3(:)==0) / size(diff3(:),1)
diff4 = grayimg2-gauss_filt_1;
num_diff4 = sum(diff4(:)==0) / size(diff4(:),1)
% visualizar las diferencias con respecto a la imagen original
figure(2)
subplot(2,2,1)
imshow(diff1);
subplot(2,2,2)
imshow(diff2);
subplot(2,2,3)
imshow(diff3);
subplot(2,2,4)
imshow(diff4);

```

- **Ejercicio 3**

Ejecute el código para obtener las versiones filtradas de las imágenes BrainMR_gauss.png, BrainMR_pois.png y BrainMR_salpe.png, e incorpore los resultados obtenidos a su reporte. Utilice los valores calculados de las diferencias (valores en variables num_diff*) para identificar cuál es el filtro que mejor reduce cada uno de los tipos de ruido agregados en la primera parte.

- **Ejercicio 4**

Replique el ejercicio anterior (3), ahora con las imágenes con ruido generadas en el Ejercicio 2, e incorpore los resultados obtenidos en su reporte, así como los valores calculados de las diferencias para identificar de nuevo el filtro que mejor reduce el ruido en cada caso.

3. Realce de bordes en las imágenes

El siguiente código presenta un ejemplo simple de opciones para realzar los bordes y texturas de una imagen en escala de grises en MATLAB:

```

% cargar la imagen
img=imread("BrainMR.png");
% convertir a escala de grises
grayimg=im2gray(img);

% crear un kernel para filtro de realce 3x3
h1 = [0 -1 0; -1 5 -1; 0 -1 0];
% filtrar la imagen con filtro de realce
real_filt_1 = uint8(conv2(grayimg,h1,'same'));

% crear otro kernel para filtro de realce 3x3
h2 = [-1 -1 -1; -1 9 -1; -1 -1 -1];
% filtrar la imagen con nuevo filtro de realce
real_filt_2 = uint8(conv2(grayimg,h2,'same'));

% visualizar la imagen y sus versiones filtradas
figure(1)
subplot(2,2,1)

```

```

imshow(grayimg);
subplot(2,2,3)
imshow(real_filt_1);
subplot(2,2,4)
imshow(real_filt_2);

% guardar las imagenes filtradas
imwrite(real_filt_1,"BrainMR_real_1.png");
imwrite(real_filt_2,"BrainMR_real_2.png");

```

- **Ejercicio 5**

Ejecute el código para obtener las versiones filtradas de la imagen BrainMR.png (adjunta a este enunciado), e incorpore los resultados obtenidos a su reporte.

- **Ejercicio 6**

Ejecute el código anterior con otra imagen (médica) diferente, para obtener sus versiones con realce, e incorpore la imagen utilizada y los resultados obtenidos a su reporte.

Entrega del ejercicio

La entrega del ejercicio práctico consistirá únicamente en el archivo de reporte en formato PDF, nombrado con los apellidos de los integrantes del grupo. Este archivo deberá enviarse a través de la correspondiente asignación en BrightSpace antes de finalizar la sesión de clase (1:00pm) del martes 24 de octubre de 2023. El envío del archivo comprimido en otro formato diferente a los especificados resultará en una calificación de (0.0/5.0) para el taller.

La escala de evaluación es la siguiente, para cada ejercicio individual:

- **Excelente (5.0/5.0):** El estudiante propone un código que realiza todos los pasos sugeridos y presenta evidencias de los resultados obtenidos.
- **Bueno (3.5/5.0):** El estudiante propone un código que realiza la mayoría de los pasos sugeridos, y presenta al menos una evidencia de los resultados obtenidos.
- **Aceptable (2.5/5.0):** El estudiante propone un código que realiza solo algunos de los pasos sugeridos, y presenta evidencia(s) de los resultados obtenidos, no todos correctos.
- **Malo (1.0/5.0):** El código propuesto por el estudiante no es correcto en su sintaxis (no puede interpretarse en el entorno de ejecución adecuadamente).
- **No entregó (0.0/5.0):** El estudiante no entrega los resultados solicitados.