



UNIVERSIDAD DE JAÉN

Reducción del ruido en una imagen digital.

Departamento de Ingeniería electrónica, Telecomunicación y Automática.
Área de Ingeniería de Sistemas y Automática

OBJETIVOS:

- Estudio de distintas máscaras para reducción del ruido en una imagen digital.

RESUMEN:

Se detallará las distintas funciones de Matlab a aplicar para reducir el ruido en una imagen digital y el efecto que sobre ella producen.

1 OPERACIONES DE SUAVIZADO.

Tienen por objeto reducir el ruido y/o efectos espurios que pueden presentarse en una imagen a consecuencia del proceso de captura, digitalización y transmisión. Su utilización es normalmente necesaria antes de la aplicación de un detector de bordes.

Existen distintos tipos de algoritmos que permiten la reducción del ruido. En esta práctica veremos algunos filtros lineales (convolución de una imagen con una máscara predefinida) y no lineales (operación no lineal con los píxeles del entorno de vecindad).

2. FILTROS LINEALES.

Como se ha comentado anteriormente en este tipo de filtros se realiza una operación de convolución entre la imagen a ser filtrada y una máscara. El principal inconveniente de estas técnicas es el enturbiamiento que se produce en la imagen, provocando el difuminado de los bordes, por lo que tienen que ser utilizados con cierta precaución. A continuación veremos algunos filtros de este tipo.

2.1 Filtro de la media.

Dada una imagen $f(i,j)$, el procedimiento consiste en generar una nueva imagen $g(i,j)$ cuya intensidad para cada píxel se obtiene promediando los valores de intensidad de los píxeles $f(i,j)$ incluidos en un entorno de vecindad predefinido.

En Matlab la función que me permite realizar un filtro de la media será: **imfilter** (ver ayuda de Matlab).

Esta función tiene la siguiente estructura:

$$B = \text{imfilter}(A,H,\text{option1},\text{option2},\dots)$$

Esta función filtra el array A con el filtro multidimensional H, el resultado se almacena en B. Los parámetros option1, option2,..., son opciones de frontera, de tamaño del array de salida y opciones de correlación o convolución.

La función de Matlab que me permite generar el filtro H será **fspecial**.

La función `fspecial` crea filtros bidimensionales del tipo especificado por `type` (para más información ver ayuda de Matlab).

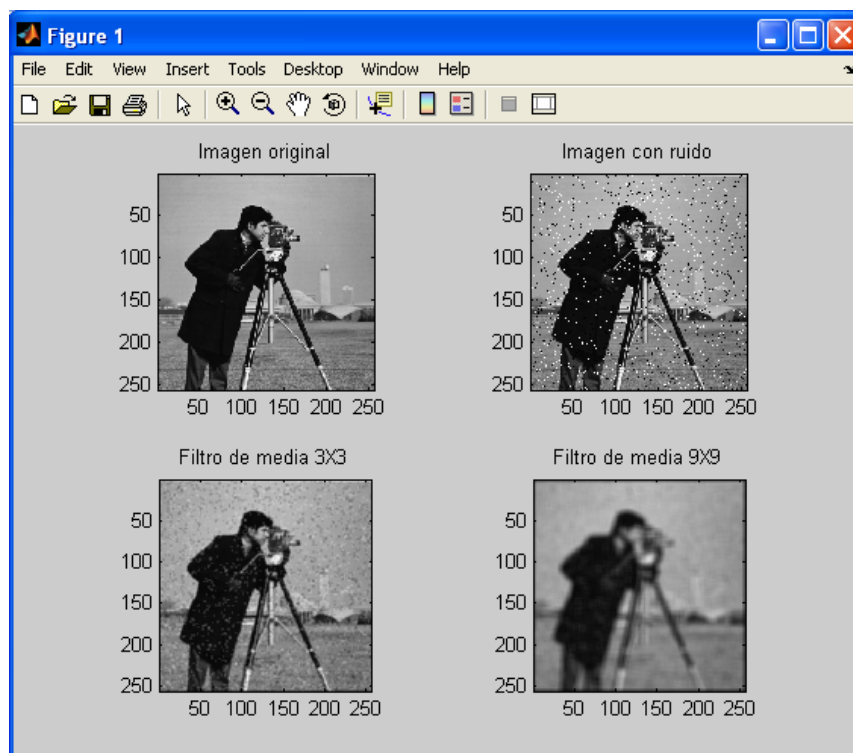
$$h = \text{fspecial}(\text{type}, \text{parameters})$$

El valor de `type` puede ser un filtro pasa baja gaussiano, un detector de bordes de sobel o prewitt, un operador laplaciano, un filtro de la media, etc.

Los parámetros dependen del tipo de filtro.

Pongamos un ejemplo de aplicar el filtro de la mediana sobre una imagen “cameraman.jpeg” a la que se le ha introducido ruido aleatorio del tipo sal y pimienta, con una máscara de 3X3 y otra de 9X9.

```
im=imread('C:\MATLAB7\work\imagenes\cameraman.gif');
fn=imnoise(im,'salt & pepper',0.05);
h1=fspecial('average');
h2=fspecial('average',[9,9]);
media1=imfilter(fn,h1);
media2=imfilter(fn,h2);
%Representaciones de las imágenes
subplot(2,2,1),subimage(im),title('Imagen original');
subplot(2,2,2),subimage(fn),title('Imagen con ruido');
subplot(2,2,3),subimage(media1),title('Filtro de media 3X3');
subplot(2,2,4),subimage(media2),title('Filtro de media 9X9');
```



En esta imagen puede apreciarse la importancia de definir el tamaño de la máscara. Cuanto mayor sea ésta se consigue una mayor reducción del ruido, pero a cambio se produce una mayor difuminación de los bordes.

En la imagen anterior se le ha añadido ruido con la función **imnoise**.

$$J = \text{imnoise}(I, \text{type}, \text{parameters})$$

El ruido que se le puede introducir es variado. Podría introducirse ruido gaussiano, ruido multiplicativo, etc... (para más información ver ayuda de Matlab).

2.2 Filtro gaussiano.

El valor de cada punto es el resultado de promediar con distintos pesos los valores vecinos a ambos lados de dicho punto. Este tipo del filtro también tiene el problema del difuminado de los bordes, pero no es tan acusado como el caso de la media simple.

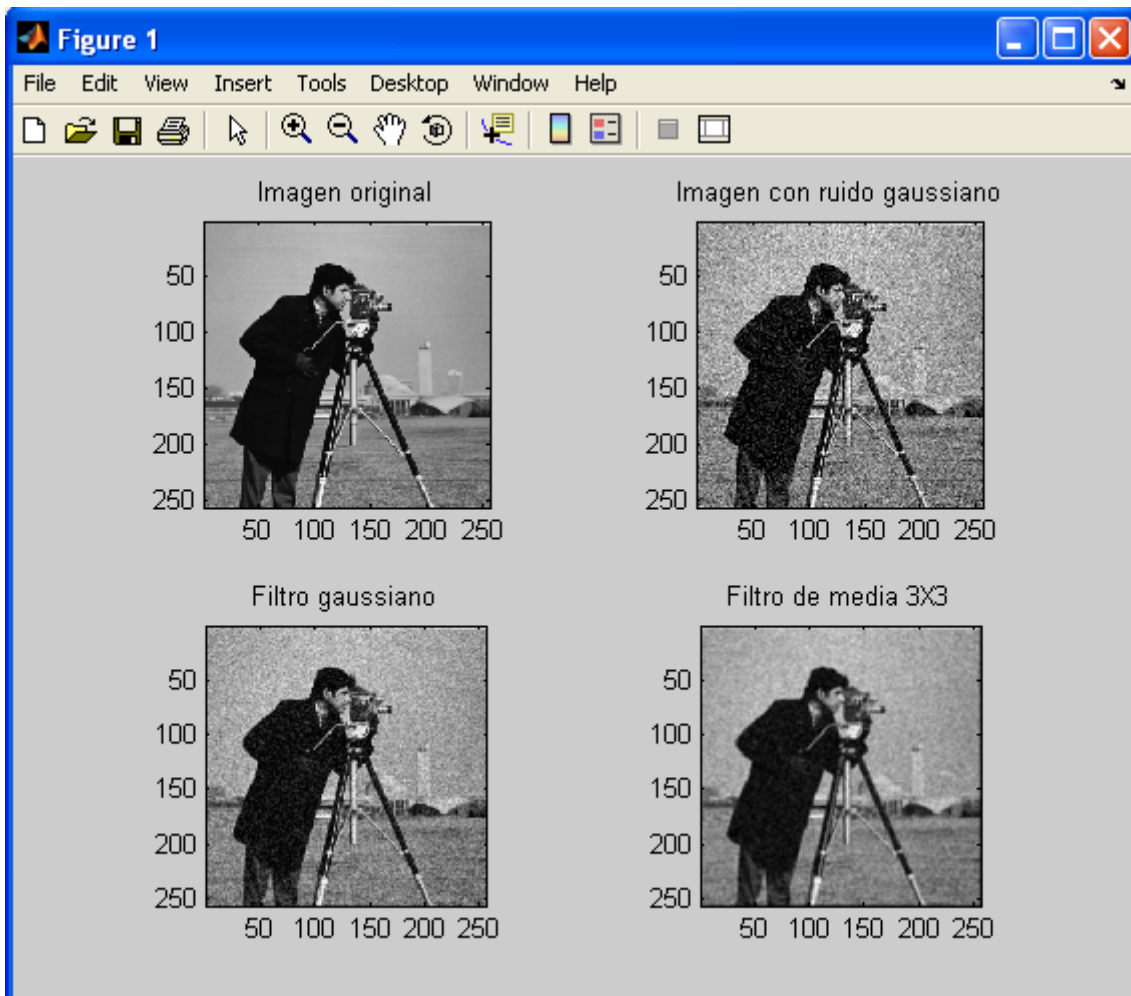
Este tipo de filtro reduce especialmente el ruido tipo gaussiano.

Ruido gaussiano: produce pequeñas variaciones en la imagen. Tiene su origen en diferencias de ganancias del sensor, ruido en la digitalización, etc.

El código de Matlab y las imágenes obtenidas de aplicar este tipo de filtro, comparándolo con uno de la media son:

```
im=imread('C:\MATLAB7\work\imagenes\cameraman.gif');
fn = imnoise(im, 'gaussian');
h1=fspecial('gaussian');
h2=fspecial('average');
g1=imfilter(fn,h1);
media2=imfilter(fn,h2);
%Representaciones de las imágenes
subplot(2,2,1),subimage(im),title('Imagen original');
subplot(2,2,2),subimage(fn),title('Imagen con ruido gaussiano');
subplot(2,2,3),subimage(g1),title('Filtro gaussiano');
subplot(2,2,4),subimage(media2),title('Filtro de media 3X3');
```

Puede apreciarse cómo el filtro gaussiano elimina el ruido mejor y además emborrona menos los bordes.



3. Filtros no lineales. Filtro de la mediana.

Los píxeles de la nueva imagen se generan calculando la mediana del conjunto de píxeles del entorno de vecindad del píxel correspondiente a la imagen origen. De esta forma se homogeneizan los píxeles de intensidad muy diferente con respecto a la de los vecinos. Este tipo de filtro es bastante indicado cuando se tiene ruido aleatorio.

En Matlab la instrucción empleada para realizar el filtro de la mediana es **medfilt2**.

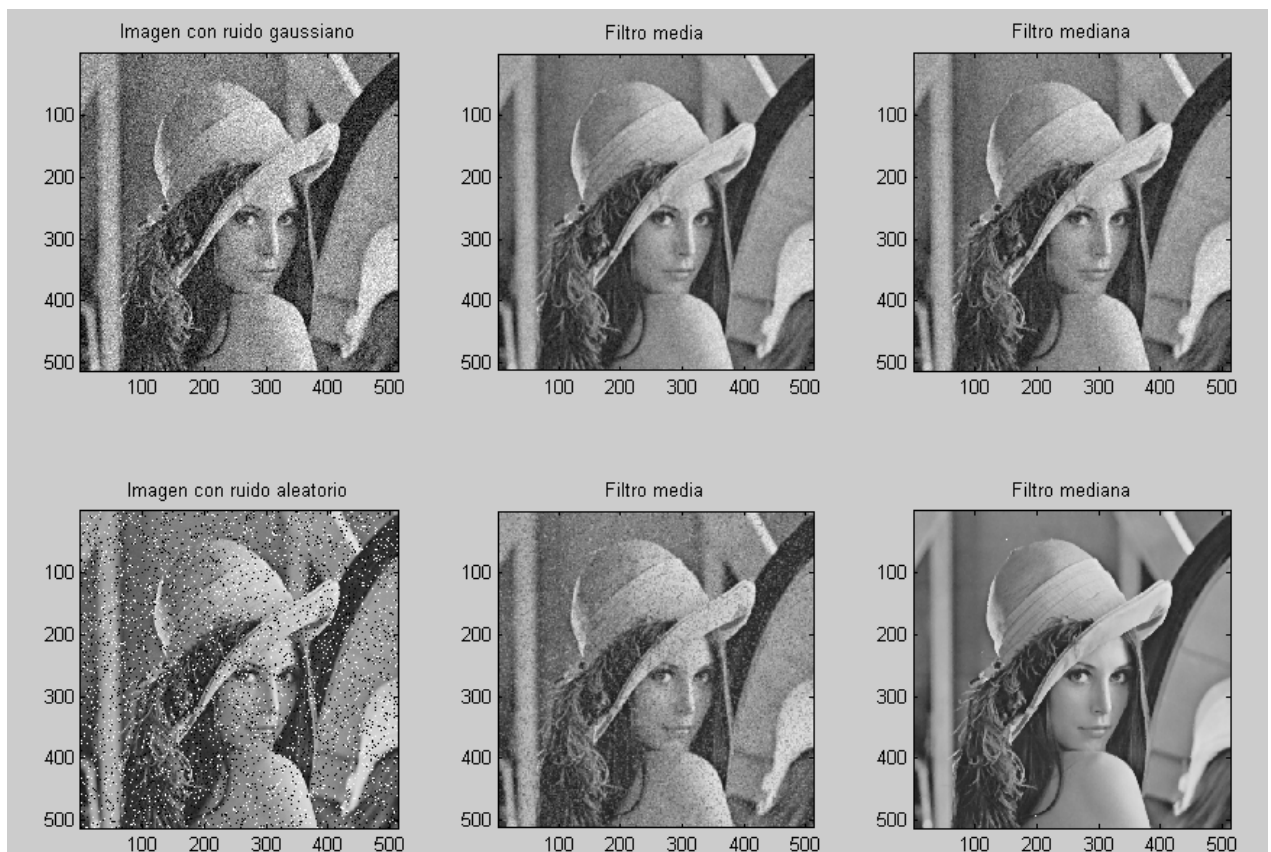
$$B = \text{medfilt2}(A)$$

Donde A es la matriz de entrada a la que se le aplica el filtro de la mediana utilizando por defecto una vecindad de 3X3.

A continuación se muestra un ejemplo de una imagen afectada por ruido gaussiano y aleatorio y filtrada por el filtro de la media y de la mediana.

```
imc=imread('C:\MATLAB7\work\imagenes\lena.jpg');
```

```
im=rgb2gray(imc);  
fg = imnoise(im,'gaussian');  
fs = imnoise(im,'salt & pepper',0.1)  
h1=fspecial('average');  
  
media1=imfilter(fg,h1);  
media2=imfilter(fs,h1);  
  
mediana1=medfilt2(fg);  
mediana2=medfilt2(fs);  
  
%Representaciones de las imágenes  
subplot(2,3,1),subimage(fg),title('Imagen con ruido gaussiano');  
subplot(2,3,4),subimage(fs),title('Imagen con ruido aleatorio');  
subplot(2,3,2),subimage(media1),title('Filtro media');  
subplot(2,3,5),subimage(media2),title('Filtro media');  
subplot(2,3,3),subimage(mediana1),title('Filtro mediana');  
subplot(2,3,6),subimage(mediana2),title('Filtro mediana');
```



Ejercicios.

1. Aplicar un filtro de la media a la imagen cuadraditos.jpg. Usar una matriz de 31X31 elementos como filtro, del siguiente estilo:

$$\frac{1}{31^2} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix}$$

Crear 3 imágenes filtradas, una con parámetro 'replicate', otra con parámetro 'symmetric' y por último una con parámetro 'circular', usando correlación y el resultado debe ser del mismo tamaño que la imagen original.

2. Cargar la imagen circuito.jpg. Introducir ruido gaussiano y ruido aleatorio en la imagen. Probar con distintos tipos de filtros hasta obtener una imagen de calidad para los ruidos anteriormente introducidos. ¿Qué tipo de filtro funciona mejor en cada uno de los casos anteriores?. Justifica tu respuesta.

NOTA: Representar en una pantalla la imagen y en otra la imagen afectada por los distintos tipos de ruido y las imágenes filtradas.