

PRACTICA 2: FILTRADO

ÍNDICE

1-INTRODUCCIÓN(2)

2-RESPUESTA A LAS PREGUNTAS (2-5)

3-COMENTARIOS SOBRE ALGUNOS RESULTADOS Y PROBLEMAS(5)



1-INTRODUCCIÓN

El objetivo de la práctica es experimentar más con filtros e implementar manualmente la función imgaussfilt de la práctica anterior.

2-RESPUESTA A LAS PREGUNTAS

1-¿Ves alguna diferencia entre usar randn e imnoise? ¿Por qué se produce? Prueba con diferentes valores de sigma y de densidad y compara los resultados.

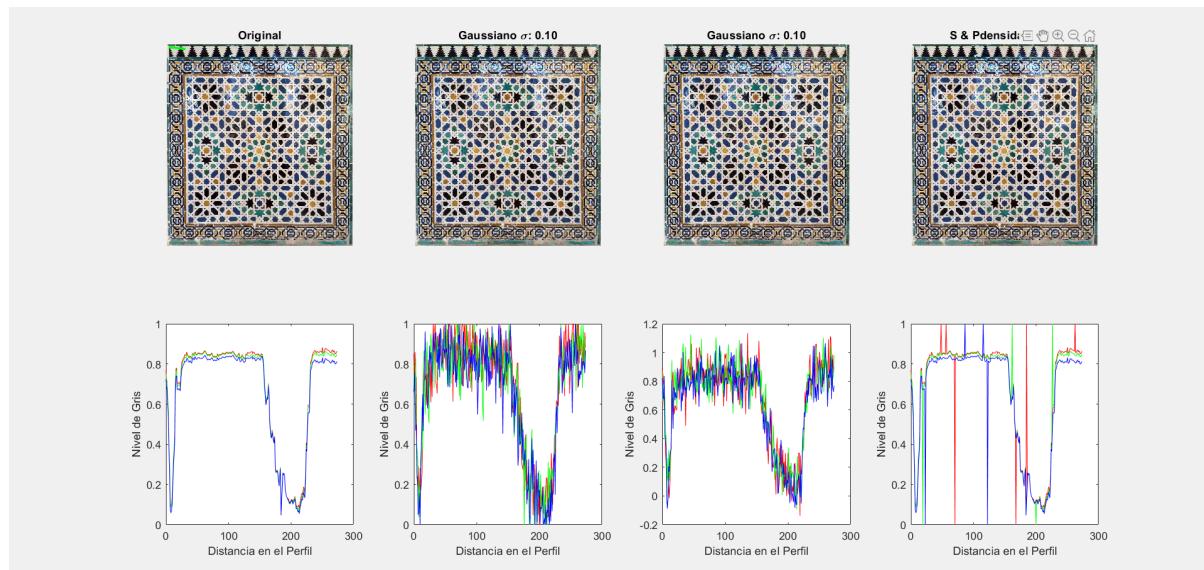


Fig1-Rand Imnoise

Para valor de sigma 0,1 y densidad 0,02 no se aprecia diferencia.

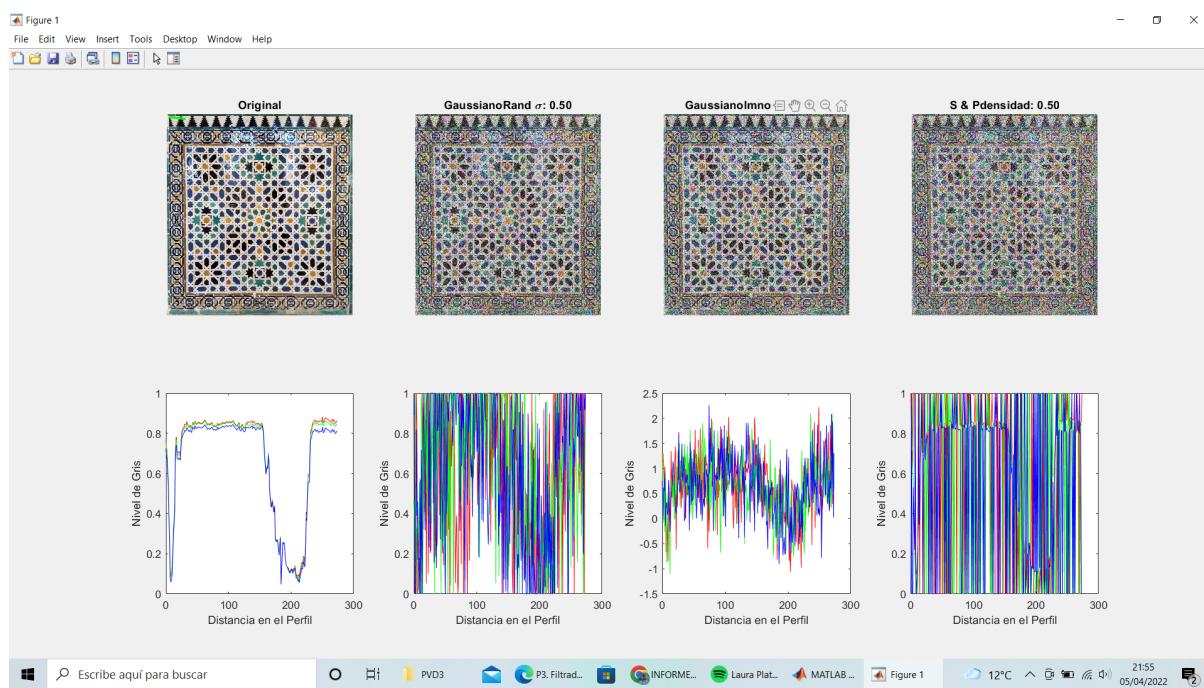


Fig2- Rand Imnoise 2

Para valor de sigma =0,5 y densidad 0,5 ya si existe una gran diferencia. En el caso de la utilización de rand se observa que el filtro introduce gran cantidad de ruido en la imagen, sin embargo, al utilizar imnoise este es mucho menor. En el caso del filtro salt and pepper, existe tanto ruido en la imagen que su distancia en el perfil prácticamente no tiene parecido a la de la imagen original.

2-¿Por qué las imágenes filtradas con los filtros de mediana y gaussiano tienen un borde negro?

Son filtros que preservan los bordes, ya que la ventana del filtro cae fuera de la imagen, para ello se añade relleno y si este relleno es a 0, predeterminado, aparece borde.

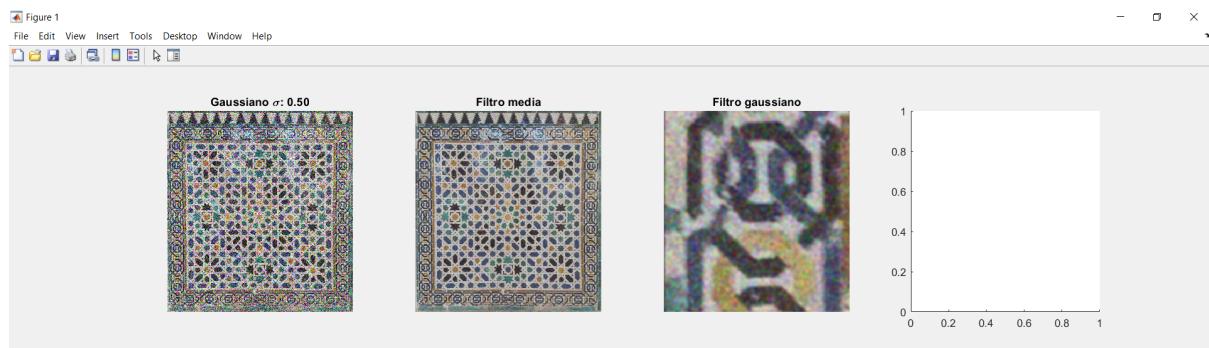


Fig3-Filtros para eliminar el ruido

Se observa un resultado muy satisfactorio teniendo en cuenta las imágenes de las que se parte (Figura 2)

3-La siguiente imagen de salida, ¿corresponde a una convolución o a una correlación?

He utilizado la siguiente parte de código para obtener el siguiente resultado:

```
%%
original = im2double(imread('eight.tif'));
%f=zeros(5); f(3,3)=1%datos

%h = [1,2,3; 4,5,6; 7,8,9];%filtro no simétrico
h=zeros(7); h(6,4)=1
corre=imfilter(original, h, 'corr');%correlacion
convol=imfilter(original, h, 'conv');%convolucion
figure(1)
imshow(corre);title('correlada');
figure(2)
imshow(convol);title('convolucionada');
```

Fig4-Código convolución correlación



Fig5- Imagen filtrada con el filtro

La respuesta es que la imagen es la convolucionada, ya que aparece el borde negro arriba, esto es porque

$h = [00000000]$ En la convolución h se rota 180 grados, quedando:

[00000000]

[00000000]

[00000000]

[00000000]

[00010000]

[00000000]

$h180 = [00000000]$

[00001000]

[00000000]

[00000000]

[00000000]

[00000000]

y al realizarse la multiplicación y la agregación con la imagen producen esos bordes en distinto lugar

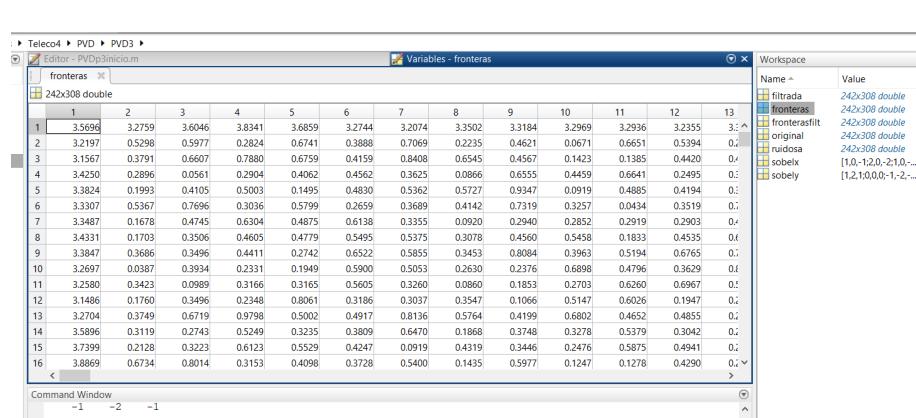
4-Observa el borde superior de la imagen con relleno circular. ¿Por qué ocurre ese artificio? Para hacer el tratamiento de los bordes con conv2 y filter2 se puede usar la función padarray, que rellena los bordes de la misma forma que imfilter ¿Qué tamaño de relleno tenemos que usar?

El tamaño del relleno tiene que ser el doble del tamaño del filtro para que cuando se haga la correlación o la convolución, se pueda obtener un resultado válido y luego eliminar la parte afectada por el relleno, quedando con el resultado correcto de la operación. El relleno a 0 produce un borde, en las imágenes con otro tipo de relleno no se puede apreciar.



Matriz de datos X, matriz plantilla(núcleo de convolución) el filtro H. H filtro, f datos en los apuntes.

5- Observa que usamos imshow(fronteras,[]) para mostrar las imágenes de fronteras. ¿Por qué?



imshow(I, []) displays the grayscale image I , scaling the display based on the range of pixel values in I . imshow uses $[\min(I(:)) \max(I(:))]$ as the display range. imshow displays the minimum value in I as black and the maximum value as white. For more information, see the DisplayRange argument.

Fig8- Imshow(I, [])

Imshow(I, []) muestra la imagen en escala de grises escalando la forma de mostrarla basada en el rango de valores de los píxeles en I, es decir, los valores se encuentran entre el mínimo de I como negro y el máximo de I como blanco. Si no se coloca la orden de esta manera se muestra lo siguiente.

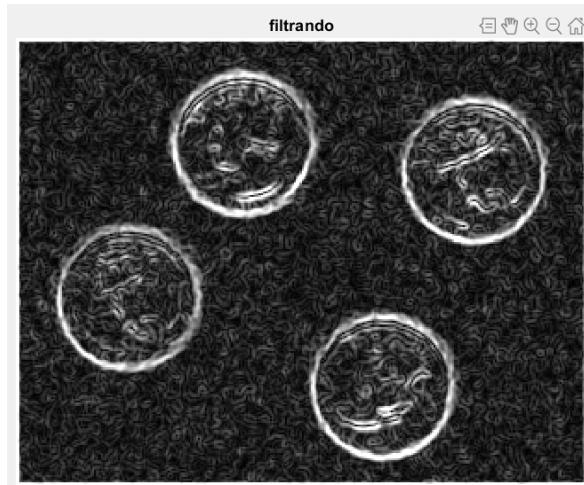


Fig9-Utilizando Imshow(I)

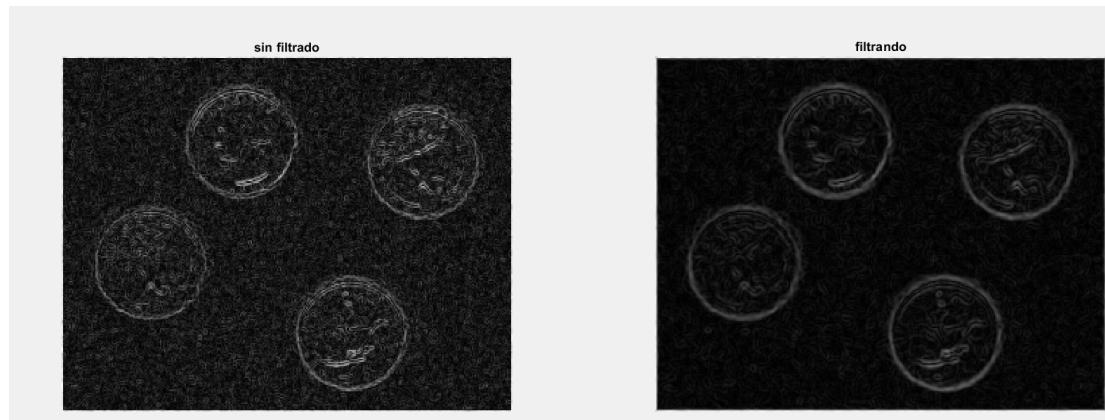


Fig10- Utilizando Imshow(I, [])

Se suaviza la imagen

6- ¿Merece la pena usar imgaussfilt?

```
filtrada = imfilter(ruidosa, fspecial('gaussian', 5, 1));
%alternativamente se puede usar: filtrada = imgaussfilt(ruidosa, 1);
fronterasfilt = sqrt(imfilter(filtrada, sobely, 'conv').^2 +
imfilter(filtrada, sobelx, 'conv').^2);
subplot(1,2,2); imshow(fronterasfilt, []); title('filtrando')
```

Fig11-Imgaufilt

Si, merece la pena ya que es mucho mejor en cuanto a economía de código y tiempo de procesamiento del programa aunque se tiene algo de menos poder de control sobre lo que se quiere hacer, pero siempre que imgaussfilt sirva para el cometido buscado es más sencillo y eficiente.

7-Implementación de imgaussfilt programada

```
%Se genera un filtro gaussiano general
%se convoluciona cada una de las bandas
%rellenas con replicación como hagan falta para hacer la convolución
%horizontales
%se coloca la imagen como un vector horizontal
%se hace la convolución 1D
%se vuelve al orden original|
%se quita el borde vertical
%ahora se repite de manera vertical

%el resultado es 1 banda de la imagen completa
%se hace con las tres bandas
```

Figura12-Descripción del proceso seguido

```
function f = miimgaussfilt2(im,sigma)
    %Gaussiana filtro separable
    %2 filtrados 1 d
    % generar filtro gaussiano horizontal
    h = filtrogaussiano(sigma);
    % La salida tiene el mismo tamaño que la entrada
    f = zeros(size(im));
    %para cada banda de la imagen (por si es una imagen en color)
    %se convoluciona cada una de las bandas
    for b=1:size(im,3)
        % relleno horizontal
        tamanorelleno = floor(size(h,2)/2);%redondeo
        rellena = padarray(im(:,:,b), [0 tamanorelleno], 'replicate', 'both');
        %solo relleno en x
        % convertir a unidimensional por filas
        %se convierte en un vector fila para hacerlo 1d
        aux = reshape(rellena', [1,numel(rellena)]);
        % convolucionar en horizontal (devuelve el mismo tamaño)
        bandaf = micorr(aux,h);
        %correlación y convolución es igual porque el filtro es simétrico
        % reconvertir a bidimensional
        bandaf = reshape(bandaf,size(rellena'))';
        % eliminar el relleno
        bandaf = bandaf(:, tamanorelleno+1:end-tamanorelleno);
    %end es el ultimo pixel, size de banda f,2 (segunda dimensión)
```

```
% relleno vertical
rellena = padarray(bandaf, [tamanorelleno 0], 'replicate', 'both');
% convertir a unidimensional por columnas
aux = reshape(rellena, [numel(rellena),1]);
% convolucionar en vertical (devuelve el mismo tamaño)
bandaf = micorr(aux,h');
% reconvertir a bidimensional
bandaf = reshape(bandaf,size(rellena));
% eliminar el relleno
bandaf = bandaf(tamanorelleno+1:end-tamanorelleno,:);
% Ponemos la banda filtrada en la imagen de salida
f(:,:,:,b) = bandaf;
end
end
```

Figura13-Función que realiza el filtrado de la imagen, im con el filtro de sigma, sigma

```
function h = filtrogaussiano(sigma)
    % el radio 2*sigma contiene el 95,45% de la masa de la gaussiana
    radio = ceil(2*sigma);%redondeo
    % kernel gaussiano 1-D para filtrado separable
    X = (-radio:radio); % puntos en que se muestrea

    %%%% MODIFICAR AQUI %%%
    % calcular los valores del filtro gaussiano. Recuerda que
    % h(i) = exp(-1/2 * (x(i)^2)/(sigma^2)); % valor del filtro
    for(i=1:1:2*radio+1)%El filtro se supone impar así que el tamaño del filtro es
        %el valor del radio +1
        %h(i) = exp(-1/2 * (X(i)^2)/(sigma^2));
        h(i) = exp(-1/2 * (X(i)^2)/(sigma^2));
    end
    %%% HASTA AQUI %%%

    % Eliminar los valores demasiado pequeños
    h(h<eps*max(h(:))) = 0;
    %eps es 1 por 10 elevado a -6
    % Normalizar para que sume 1
    sumH = sum(h(:));
    if sumH ~=0 % comprobación de seguridad.
        h = h./sumH;
    end
end
```

Figura14-Función que crea el filtro de sigma, sigma

Cabe destacar que lo que se calcula en h(i) es el exponente de la gaussiana, la base se calcula a partir de la normalización.

```

function f = micorr(x,h)
    % tamaño del relleno: mitad del tamaño del filtro
    mf = floor(numel(h)/2);
    % La salida tiene el mismo tamaño que la entrada
    f = zeros(size(x));
    % Para la zona de la señal válida
    for i = mf+1:numel(x)-mf % Esto recorre desde K/2+1 a el tamaño de la imagen - K/2
        %que son las posiciones donde el filtrado tiene valores de interés

        %%%% MODIFICAR AQUI %%%
        % calcular el valor de la correlación en la posición i de x
        %x es un vector fila

        %K/2 es mf, k es el indice de la sumatoria

        %Sin embargo esta k, indice de la sumatoria no aparece, se ha hecho
        %una transformación. Respecto a la fórmula vista en clase que va de
        %k=-k a k=k se le suma k+1 a todo para que la sumatoria recorra
        %desde 1 hasta 2k+1. Una vez hecho esto uno debe ver que los
        %valores de i(pixeles) para la imagen de los que hay que realizar
        %la sumatoria multiplicada por el filtro son los que están en el
        %rango de i=i-la mitad del tamaño del filtro (mf) a i=i+ la mitad
        %del tamaño del filtro(mf), por ello la sumatoria que representa
        %el valor de la correlación está descrita por la fórmula siguiente
        f(i)=sum(h.*x(i-mf:i+mf));
        %%% HASTA AQUI %%%

    end

```

Figura15-Función que realiza la correlación del filtro gaussiano con la imagen

La explicación de este proceso ha sido escrita en forma de comentario en el script, en la imagen se puede apreciar.

```

%% Comprobación Imgaußfilt vs Miimgaußfilt2
%figure(1) y figure(2) deben ser iguales

%% Borrado de variables
clear all
clc
%% Lectura de imágenes
imagen1 = im2double(imread('MosaicoAlhambra.jpg'));
sigma=30;
%gaußfilt1
imagenfilt1 = imgaussfilt(imagen1, sigma);
%gaußfilt270
imagenfilt2 = miimgaussfilt270(imagen1, sigma);

figure(1)
imshow(imagenfilt1);
figure(2)
imshow(imagenfilt2);

```

Figura16-Script de comprobación

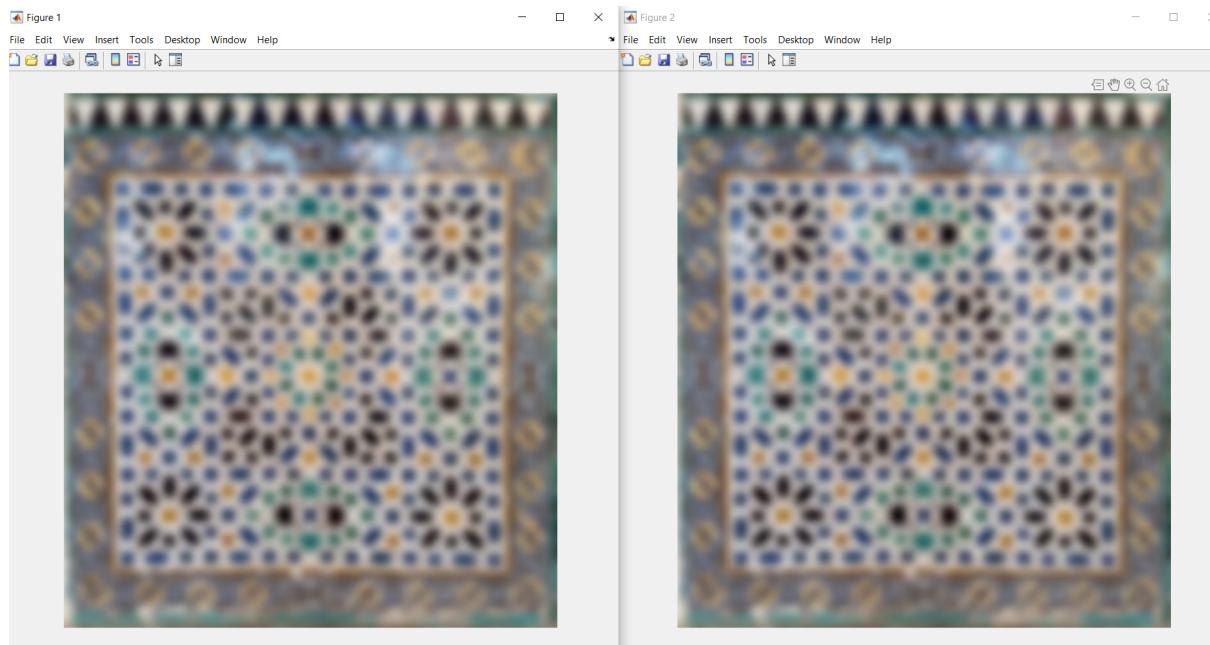


Figura17-Resultados de la comprobación para sigma=30

Visualmente comprobamos que el resultado es igual, así que la función creada funciona de manera correcta

3-COMENTARIOS SOBRE ALGUNOS RESULTADOS Y PROBLEMAS

Tuve problemas con la comprensión de cómo calcular la correlación a partir de los datos de entrada, ya que no sabía que se podía introducir un barrido en un rango dentro de un bucle sin realizar otro bucle. Sin embargo este problema pudo solucionarse gracias a una explicación del profesor el día anterior a la entrega.