

Practica 4: Filtros Espaciales

Aguilera Palacios Luis Ernesto, leapvader.1998@hotmail.com

Padilla Castillo Aaron Samir, samir.castill@gmail.com,

Facultad de Ingeniería, Universidad Nacional Autónoma de México

Resumen—En esta practica aprenderemos sobre los filtros espaciales en el procesamiento digital de imágenes, desde operaciones de suavizado, hasta mejoramiento de nitidez con unsharp masking, pasando por operaciones como detección de bordes o aproximaciones de gradientes y laplacianos.

Abstract—In this school work we'll learn about spacial filtering in digital image processing, from smoothing to sharpening techniques with unsharp masking.

I. INTRODUCCIÓN

Los filtros espaciales tienen como objetivo modificar la contribución de determinados rangos de frecuencias de una imagen. El término espacial se requiere a que el filtro se aplica directamente a la imagen y no a una transformada de la misma, es decir, el nivel de gris de un pixel se obtiene directamente en función del valor de sus vecinos. La convolución es la operación con la cual se hace filtrado espacial.

Los filtros espaciales pueden clasificarse basándose en su linealidad en filtros lineales y en filtros no lineales. A su vez los filtros lineales pueden ser clasificados según las frecuencias que dejen pasar: los filtros paso bajo atenúan o eliminan las componentes de alta frecuencia a la vez que dejan inalteradas las bajas frecuencias; los filtros paso altas atenúan o eliminan las componentes de baja frecuencia con lo que agudizan las componentes de alta frecuencia; los filtros paso banda eliminan regiones elegidas de frecuencias intermedias. A continuación se describe el uso de los diferentes filtros:

3x3 Window		
Input		
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9
16	12	20
13	9	15
2	7	12

Output		
		12

Fig. 1. Ejemplo de filtro espacial

Filtros paso bajas: son utilizados en la reducción de ruido; suavizan y aplanan un poco las imágenes y como consecuencia se reduce o se pierde la nitidez. En inglés son conocidos como Smoothing Spatial Filters.

Filtros paso altas: estos filtros son utilizados para detectar

cambios de luminosidad. Son utilizados en la detección de patrones como bordes o para resaltar detalles finos de una imagen. En inglés son conocidos como Sharpening Spatial Filters. Los filtros unsharp masking son filtros paso altas usados en el mejoramiento de la nitidez o de la calidad visual de una imagen.

Filtros paso banda: son utilizados para detectar patrones de ruido. Ya que un filtro paso banda generalmente elimina demasiado contenido de una imagen casi no son usados, sin embargo, los filtros paso banda son útiles para aislar los efectos de ciertas bandas de frecuencias seleccionadas sobre una imagen. De esta manera, estos filtros ayudan a simplificar el análisis de ruido, razonablemente independiente del contenido de la imagen.

II. DESARROLLO

- Para todos los puntos siguientes, utilizar una imagen sin ruido y otra imagen con ruido. La imagen con ruido se puede generar a partir de la imagen sin ruido usando el siguiente comando de MATLAB: `J = IMNOISE(I,TIPO...)`, donde TIPO es una cadena que puede tomar valores 'gaussian' , 'localvar', etc.

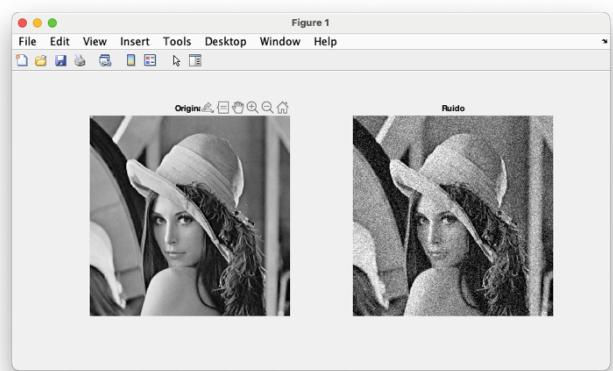


Fig. 2. A la izquierda la imagen original, a la derecha con ruido

- Aplicar los filtros paso bajas de bloque a la imagen sin ruido y a la imagen con ruido usando filtros de orden 3x3, 5x5, 7x7 y 11x11.

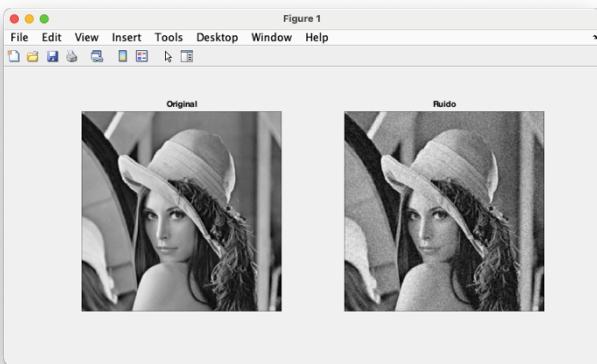


Fig. 3. Filtro pasa bajas de bloque 3x3

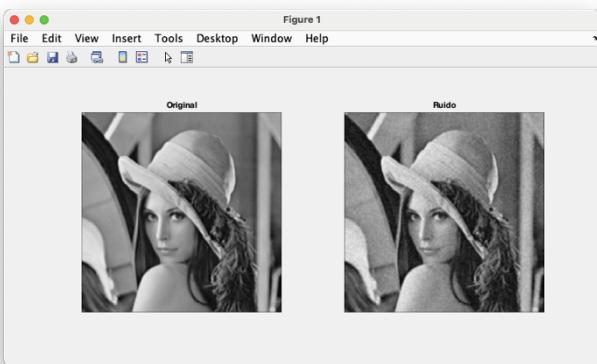


Fig. 4. Filtro pasa bajas de bloque 5x5

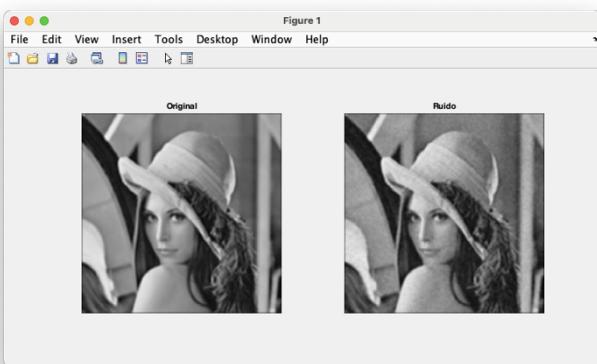


Fig. 5. Filtro pasa bajas de bloque 7x7

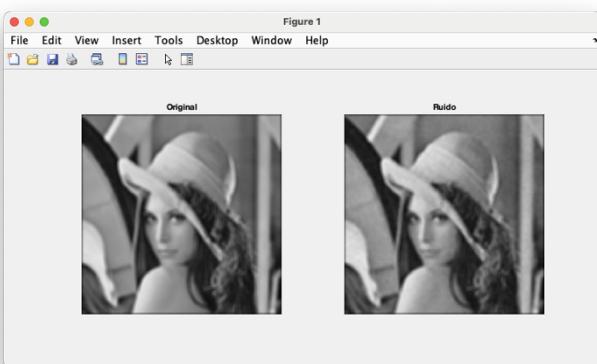


Fig. 6. Filtro pasa bajas de bloque 11x11

En las imágenes con filtro de bloque se puede observar como en algunas zonas aparecen artefactos en la imagen y que el ruido favorece la aparición de estos. Aquí entre más grande es el filtro perdemos detalle rápidamente y el ruido nos hace perder calidad.

3. Aplicar los filtros paso bajas binomiales a la imagen sin ruido y a la imagen con ruido usando filtros de orden 3x3, 5x5, 7x7 y 11x11.

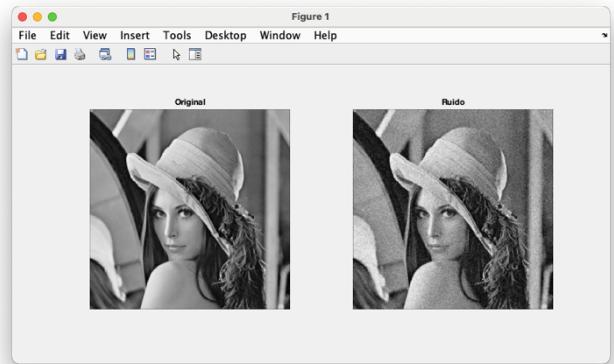


Fig. 7. Filtro pasa bajas binomial 3x3

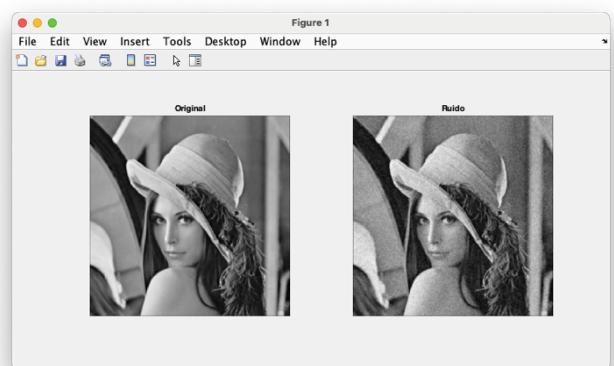


Fig. 8. Filtro pasa bajas binomial 5x5

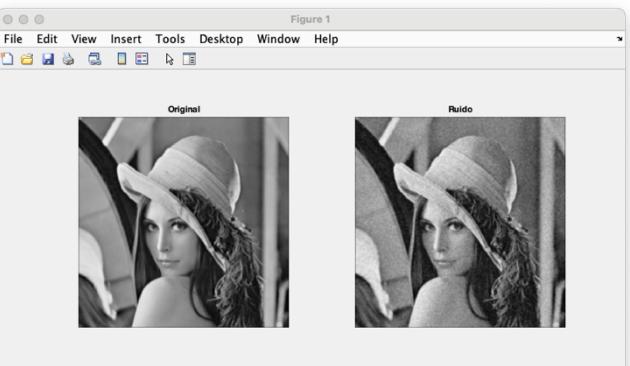


Fig. 9. Filtro pasa bajas binomial 7x7

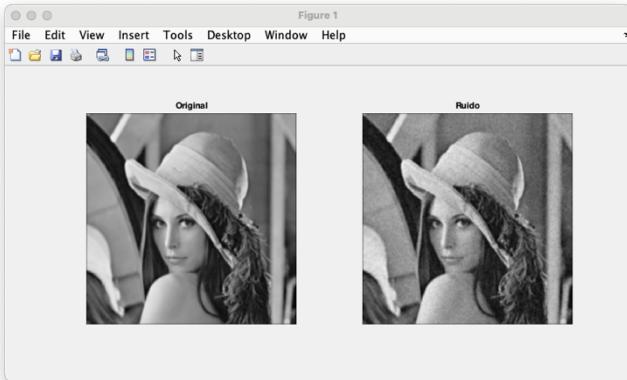


Fig. 10. Filtro pasa bajas binomial 11x11

En las imágenes con filtro binomial podemos ver que el ruido a pesar de que afecta la imagen a la hora de filtrarlo no nos genera artefactos y además sigue manteniendo bastante bien la nitidez de la imagen. En este caso entre más grande es el filtro menos ruido tenemos en la imagen final sin perder tanto detalle como en el de bloque.

4. Aplicar a la imagen sin ruido y con ruido los filtros basados en la primera derivada de gaussiana o detectores de borde siguientes:

a) De bloque [1 -1].

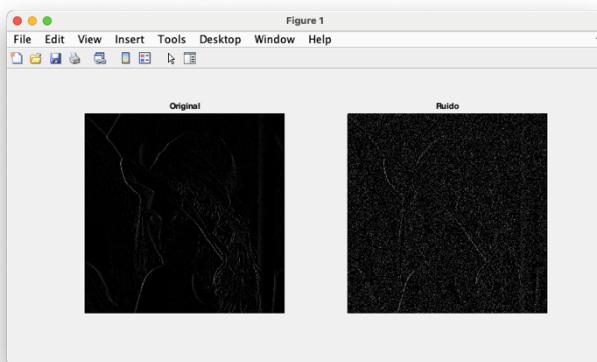


Fig. 11. Filtro basado en bloque [1 -1]

Detector de borde básico que se basa en los mínimos y máximos locales por lo que solo vemos bordes tenues.

b) Prewitt en la dirección X y en la dirección Y.

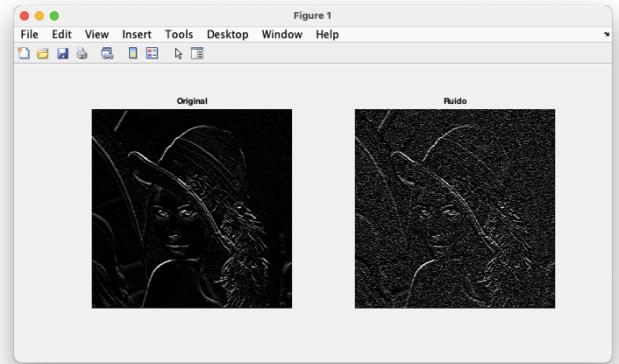


Fig. 12. Prewitt en direccion X

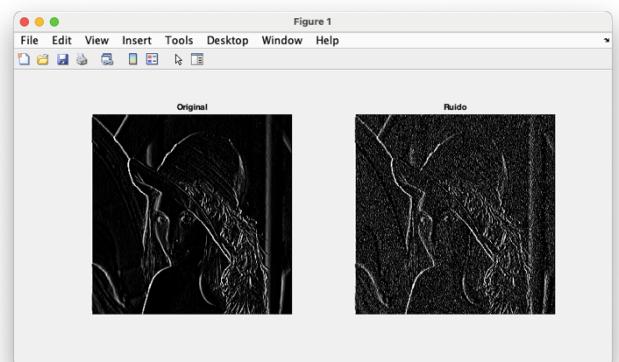


Fig. 13. Prewitt en direccion Y

Detectores de borde con cruce en 0, por lo que podemos sortear mejor los mínimos y máximos locales de mejor manera, aunque solo obtenemos los bordes muy evidentes.

c) Sobel en la dirección X y en la dirección Y.

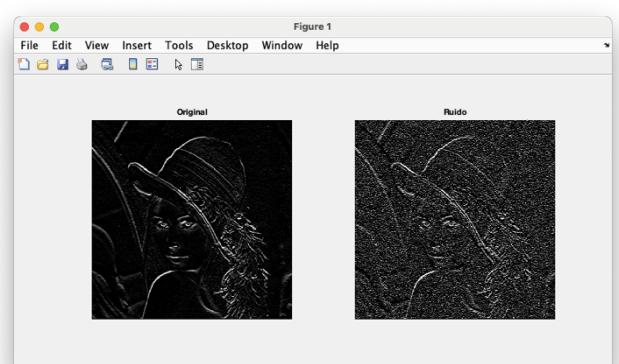


Fig. 14. Sobel en direccion X

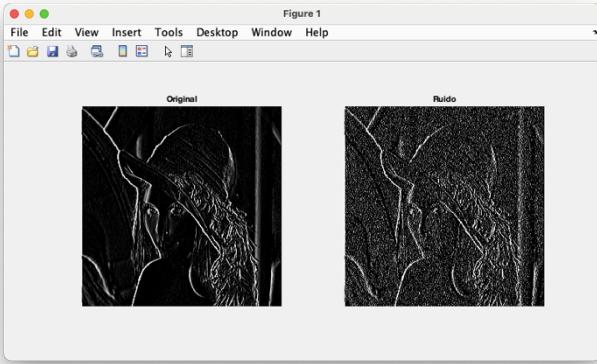


Fig. 15. Sobel en direccion Y

Sobel nos permite detectar los mismos bordes pero dando un suavizado en los cambios lo que nos ayuda a distinguir un poco mejor los mismos.

d) Basados en la primera derivada de Gaussiana de orden 5x5, 7x7 y 11x11.

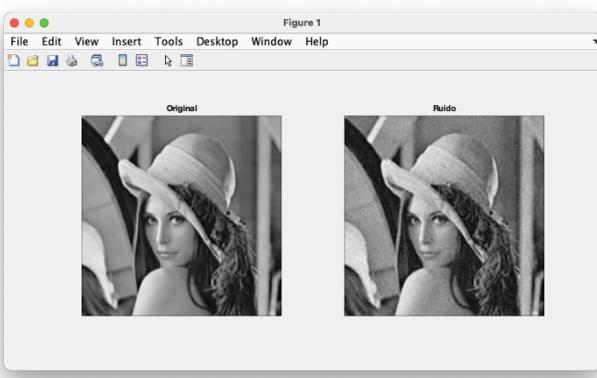


Fig. 16. Basado en la primera derivada de Gaussiana de orden 5x5

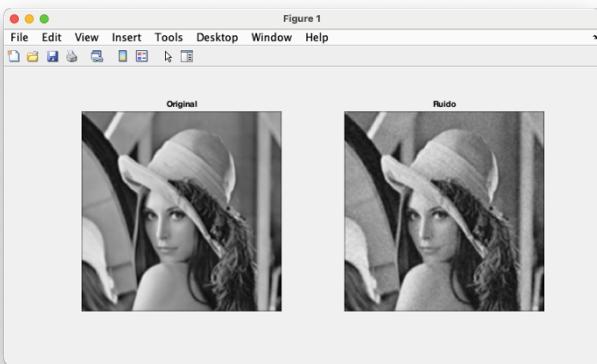


Fig. 17. Basados en la primera derivada de Gaussiana de orden 7x7

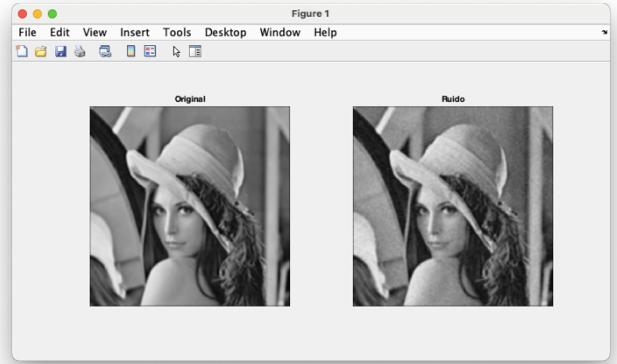


Fig. 18. Basados en la primera derivada de Gaussiana de orden 11x11

Primera derivada de gaussiana que nos deja ver un comportamiento similar en los distintos tamaños.

5. De igual manera, aplicar a la imagen sin ruido y a la imagen con ruido los filtros basados en la segunda derivada de gausiana siguientes:

a) Laplaciano

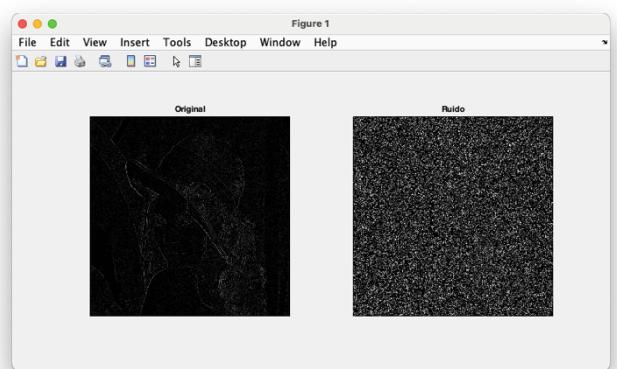


Fig. 19. Laplaciano

El filtro laplaciano encuentra los cruces por cero por lo que en un lado vemos solo ciertos bordes y en el otro vemos más bien el ruido que se introdujo en la imagen.

b) Basados en la segunda derivada de Gaussiana de orden 5x5, 7x7 y 11x11.

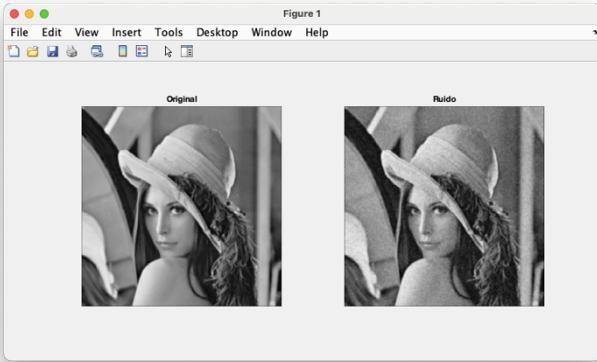


Fig. 20. Basados en la segunda derivada de Gaussiana de orden 5x5

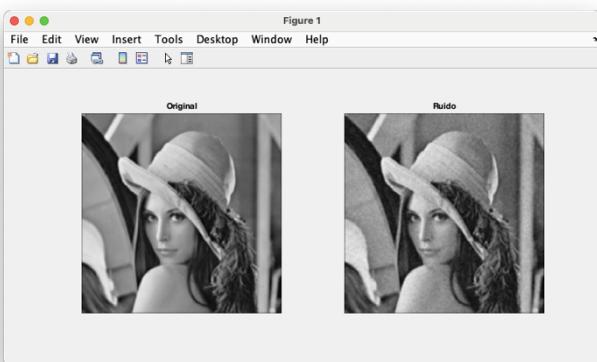


Fig. 21. Basados en la segunda derivada de Gaussiana de orden 7x7

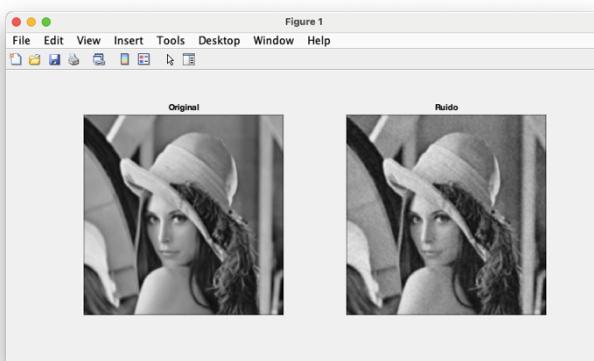


Fig. 22. Basados en la segunda derivada de Gaussiana de orden 11x11
En este caso la primera y la segunda derivada nos dieron lo mismo.

6. Difuminar las imágenes sin ruido y con ruido usando un filtro paso bajas de orden 5x5, de tal manera que se obtenga una imagen sin ruido y con pérdida de nitidez y otra imagen con ruido y perdida de nitidez. Para cada uno de los siguientes incisos, filtrar las imágenes utilizando el filtro unsharp masking encontrado con los siguientes tipos de filtro paso bajas:

a) Filtro paso bajas de orden 3x3 y 7x7.

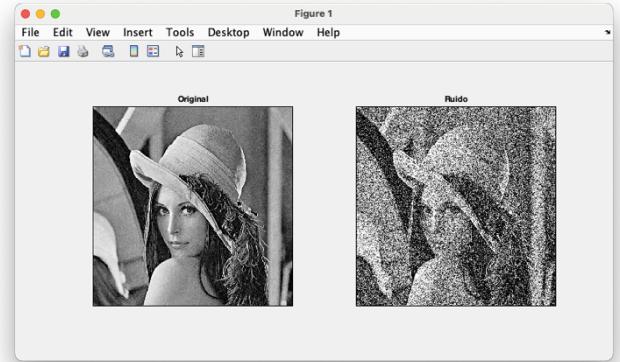


Fig. 23. UM con filtro pasa bajas de bloque 3x3

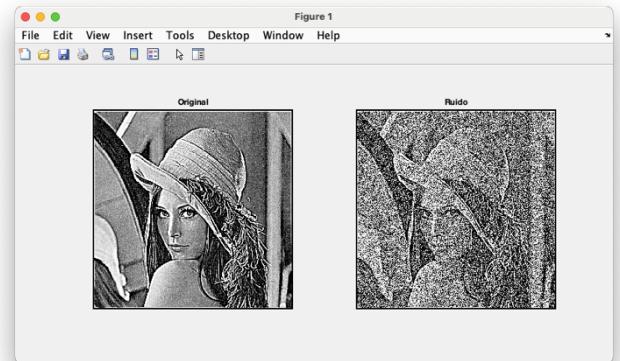


Fig. 24. UM con filtro pasa bajas de bloque 7x7
El filtro de bloque nos quita nitidez aunque el ruido no nos quita tanto detalle.

b) Filtro paso bajas binomial de orden 3x3 y 7x7.

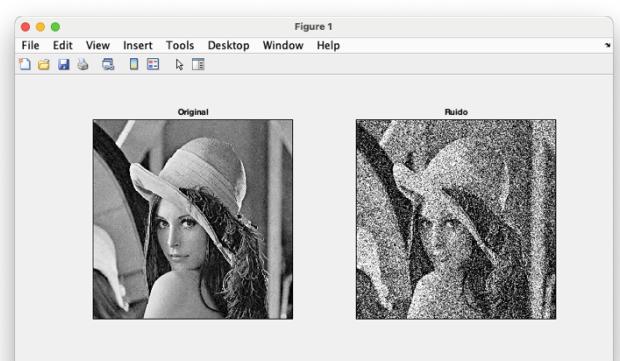


Fig. 25. UM con filtro pasa bajas binomial 3x3

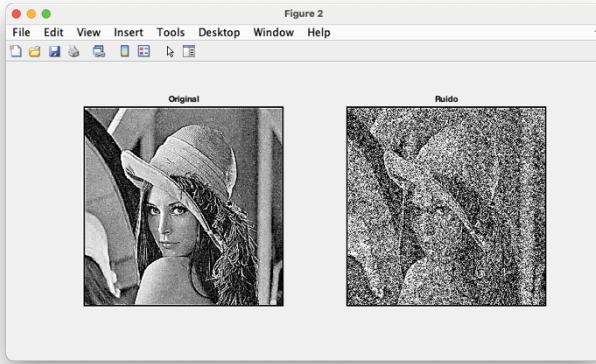


Fig. 26. UM con filtro pasa bajas binomial 7x7

Aquí se puede apreciar como los filtros afectan la nitidez de la imagen y tratan de distinta manera al ruido. En el binomial podemos ver como se conserva mejor la nitidez, aunque el ruido nos ha quitado algo de detalle.

III. CÓDIGO

```

clc
clear

%imprecision
figure('Renderer', 'painters', 'Position', [350 200 800 400])
subplot(1,2,1);
imshow(im_f/255);
title("Original");

subplot(1,2,2);
imshow(imn_f/255);
title("Ruido");

%punto1: generar im_noise
im = imread('lenna.jpeg');
%im = imread('cameraman.jpg');
imn = imnoise(im, "gaussian");

%punto2: 4 filtros (11x11, 7x7, 5x5, 3x3) paso bajas de bloque a
im e im_noise
N = 11; %5, 7, 11
f = ones(N) / (N*N);
im_f = conv2(im, f, 'same');
%im_f = filter2(f, im, 'same'); %correlacion
imn_f = conv2(imn, f, 'same');
%imn_f = filter2(f, imn, 'same'); %correlacion

%punto3: 4 filtros (11x11, 7x7, 5x5, 3x3) paso bajas binomiales a
im e im_noise
binom = [1 2 1] %3
binom = [1 4 6 4 1] %5
binom = [1 6 15 20 15 6 1] %7
binom = [1 8 28 56 70 56 28 8 1] %9
binom = [1 10 45 120 210 252 210 120 45 10 1] %11

filtroB = (binom.*binom')/(sum(binom)^2);
im_f = conv2(im, filtroB, 'same');
imn_f = conv2(imn, filtroB, 'same');

%punto4
%a
    %bloque derivada
    filtro = [1 -1]
    im_f = conv2(im, filtro, 'same');
    imn_f = conv2(imn, filtro, 'same');

%b

```

```

%Prewit X
filtro = fspecial('prewitt');
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%Prewit Y
filtro = fspecial('prewitt');
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%C
%Sobel X
filtro = fspecial('sobel');
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%Sobel Y
filtro = fspecial('sobel');
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%d
%primera derivada Gaussiana orden 5
filtro = fspecial('gaussian', [5 5], 3);
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%primera derivada Gaussiana orden 7
filtro = fspecial('gaussian', [7 7], 3);
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%primera derivada Gaussiana orden 11
filtro = fspecial('gaussian', [11 11], 3);
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%punto5
%a
%laplaciano
filtro = fspecial('laplacian',0);
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%b
%segunda derivada Gaussiana orden 5
filtro = fspecial('gaussian', [5 5], 2);
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%segunda derivada Gaussiana orden 7
filtro = fspecial('gaussian', [7 7], 2);
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%segunda derivada Gaussiana orden 11
filtro = fspecial('gaussian', [11 11], 2);
im_f = conv2(im, filtro, 'same');
imn_f = conv2(imn, filtro, 'same');

%punto6
%a
%unsharp masking UM 3x3
k3 = 3;
UMZ3 = zeros(3);
UMZ3(2,2) = (1+k3);
UMPB3 = (k3/(3^3)) * ones(3);
UMF3 = UMZ3-UMPB3;

UMS3 = conv2(im, UMF3, 'full');
UMS3_n = conv2(imn, UMF3, 'full');

figure('Renderer', 'painters', 'Position', [350 200
800 400])
subplot(1,2,1);

```

```

imshow(UMS3/255);
title("Original");

subplot(1,2,2);
imshow(UMS3_n/255);
title("Ruido");

%unsharp masking UM 7x7
k7 = 7;
UMZ7 = zeros(7);
UMZ7(4,4) = (1+k7);
UMPB7 = (k7/(7*7)) * ones(7);
UMF7 = UMZ7-UMPB7;

UMS7 = conv2(im, UMF7, 'full');
UMS7_n = conv2(imn, UMF7, 'full');

figure('Renderer', 'painters', 'Position', [350 200
800 400])
subplot(1,2,1);
imshow(UMS7/255);
title("Original");

subplot(1,2,2);
imshow(UMS7_n/255);
title("Ruido");

%b

%unsharp masking UM 3x3 con Binomial
binom = [1 2 1] %3

k3 = 3;
UMZ3 = zeros(3);
UMZ3(2,2) = (1+k3);
filtroB = k3*(binom.*binom')/(sum(binom)^2);
UMF3 = UMZ3-filtroB;

UMS3 = conv2(im, UMF3, 'full');
UMS3_n = conv2(imn, UMF3, 'full');

figure('Renderer', 'painters', 'Position', [350 200
800 400])
subplot(1,2,1);
imshow(UMS3/255);
title("Original");

subplot(1,2,2);
imshow(UMS3_n/255);
title("Ruido");

%unsharp masking UM 7x7 con Binomial
binom = [1 6 15 20 15 6 1] %7

k7 = 7;
UMZ7 = zeros(7);
UMZ7(4,4) = (1+k7);
filtroB = k7*(binom.*binom')/(sum(binom)^2);
UMF7 = UMZ7-filtroB;

UMS7 = conv2(im, UMF7, 'full');
UMS7_n = conv2(imn, UMF7, 'full');

figure('Renderer', 'painters', 'Position', [350 200
800 400])
subplot(1,2,1);
imshow(UMS7/255);
title("Original");

subplot(1,2,2);
imshow(UMS7_n/255);
title("Ruido");

```

IV. CONCLUSIONES

En esta practica hemos analizado cada uno de los filtros espaciales vistos en clase, tanto en una imagen con ruido como otra sin el. Este tipo de filtros nos ayudan a obtener informacion especifica de nuestra matriz/imagen, la cual podremos utilizar a nuestra conveniencia dependiendo de cual sea el objetivo. Conocer los tipos de filtros espaciales nos puede ser de ayuda para construir redes neuronales convolucionales, o para filtrar los datos que nos interesan de cierta imagen.

REFERENCIAS

- [1] Omar Sanchez. (2008). Imagen Filtrado Espacial, 28/11/2020, de Slideshare Sitio web: <https://es.slideshare.net/omarspp/imagen-filtrado-espacial>