

Práctica 4 ”Filtros Espaciales”

Carrillo Sánchez R. * Martínez Soto M. **
Sánchez Majarrez A. *** Valenzuela García de León F. R. ****

* DIE, Facultad de Ingeniería, CDMX, México
(r.carrillosanchez@comunidad.unam.mx)

** DIE, Facultad de Ingeniería, CDMX, México
(marjo_soto@comunidad.unam.mx)

*** DIE, Facultad de Ingeniería, CDMX, México
(andrew.manjarrez@comunidad.unam.mx)

**** DIE, Facultad de Ingeniería, CDMX, México
(fer_rodri-val@hotmail.com)

Abstract: El presente documento recopila ejercicios de modelado de imágenes con el uso de distintos filtros espaciales y técnicas de propósito general para lograr componentes y efectos de realce, abstracción, suavizado, mejora de nitidez, filtrado y mejora en el análisis de estudio de patrones. **Keywords:** Filtros espaciales, Filtros Paso-Altas, Filtros Paso-Bajas, Blurring, Filtro Binomial, Unsharp Masking

Keywords: Filtros espaciales, Filtros Paso-Altas, Filtros Paso-Bajas, Blurring, Filtro Binomial, Unsharp Masking

1. OBJETIVOS

- Realizar operaciones de suavizado y de reducción de ruido en imágenes utilizando filtros espaciales de bloque y binomiales.
- Realizar operaciones de detección de bordes en imágenes, tanto limpias como ruidosas, utilizando filtros basados en aproximaciones de gradientes y laplacianos, así como derivadas de primer y segundo orden de funciones Gausianas (binomiales).
- Mejorar la nitidez de las imágenes sin ruido y con ruido usando los filtros unsharp masking

Imagen original

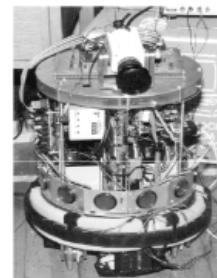


Fig. 1. Ejemplo de Imagen

Filtro gaussiano con $\sigma=2.0$



Fig. 2. Aplicación de filtro Gaussiano

3. DESARROLLO

3.1 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 .

Los filtros digitales es una forma de poder manejar las diferentes imágenes digitales existentes, donde sirven para diferentes fines, teniendo en común que dependen de los pixeles de entorno. Algunos de los objetivos puede ser el suavizar una imagen para reducir las diferentes variaciones de intensidad entre pixeles vecinos. Eliminar ruido para modificar la intensidad de los pixeles cuyos difieren mucho de sus vecinos. Realizar una imagen para aumentar las variaciones de intensidad donde se producen y detección de bordes para detectar aquellos pixeles con cambios bruscos en su intensidad.

Una de los filtros más conocidos es el filtro gaussiano, el cual ayuda para emborrifar imágenes y eliminar ruido de esa forma. De esta forma, se puede realizar convoluciones bidimensionales al igual que es uniforme.

Por ejemplo:



Fig. 3. Imagen original



Fig. 6. Im. con ruido filtro PB de bloque 7x7



Fig. 4. Imagen con ruido



Fig. 7. Im. con ruido filtro PB de bloque 9x9



Fig. 5. Im. con ruido filtro PB de bloque 3x3



Fig. 8. Im. con ruido filtro PB de bloque 11x11

3.2 Aplicar los filtros paso bajas de bloque a la imagen sin ruido y a la imagen con ruido usando filtros de orden 3x3, 7x7, 9x9 y 11x11

3.3 Aplicar los filtros paso bajas binomiales a la imagen sin ruido y a la imagen con ruido usando filtros de orden 3x3, 7x7, 9x9 y 11x11.



Fig. 9. Img. sin ruido con FP Gaussiano 3x3



Fig. 12. Img. con ruido con FP Gaussiano 7x7



Fig. 10. Img. con ruido con FP Gaussiano 3x3



Fig. 13. Img. sin ruido con FP Gaussiano 9x9



Fig. 11. Img. sin ruido con FP Gaussiano 7x7



Fig. 14. Img. con ruido con FP Gaussiano 9x9



Fig. 15. Img. sin ruido con FP Gaussiano 11x11

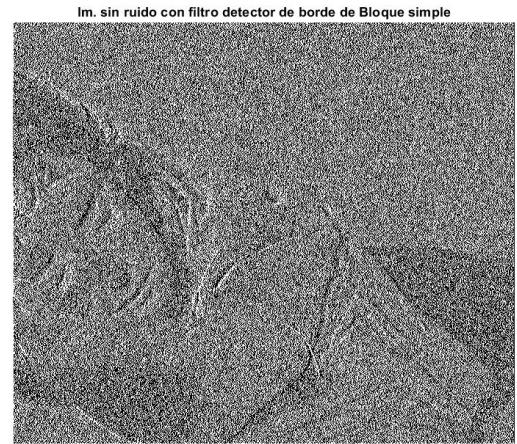


Fig. 17. Img. de bloque [1 -1]



Fig. 16. Img. con ruido con FP Gaussiano 11x11



Fig. 18. Img. Con Ruido Prewitt en X

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

De bloque [1 -1].



Fig. 19. Img. Con Ruido Prewitt en Y

Fig. 22. Img. Con Ruido Sobel en X



Fig. 20. Img. Sin Ruido Prewitt en X

Fig. 23. Img. Con Ruido Sobel en X



Fig. 21. Img. Sin Ruido Prewitt en Y
Sobel en la dirección X y en la dirección Y.

Fig. 24. Img. Sin Ruido Sobel en X



Fig. 25. Img. Sin Ruido Sobel en Y

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



Fig. 27. Img. Sin Ruido PDG 5x5

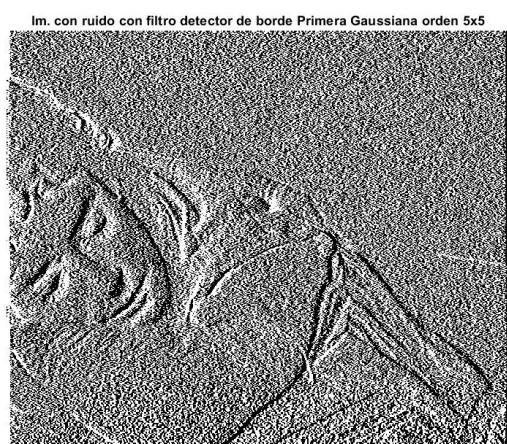


Fig. 26. Img. Con Ruido PDG 5x5



Fig. 28. Img. Con Ruido PDG 7X7

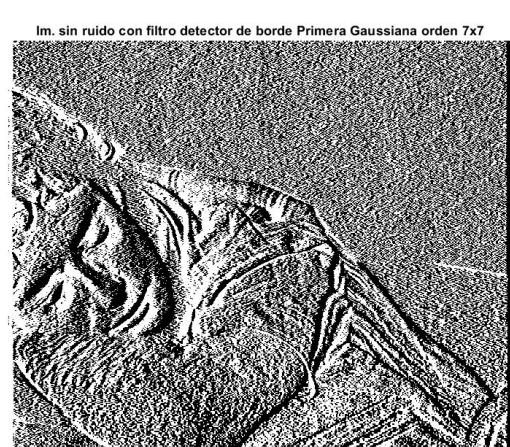


Fig. 29. Img. Sin Ruido PDG 7x7

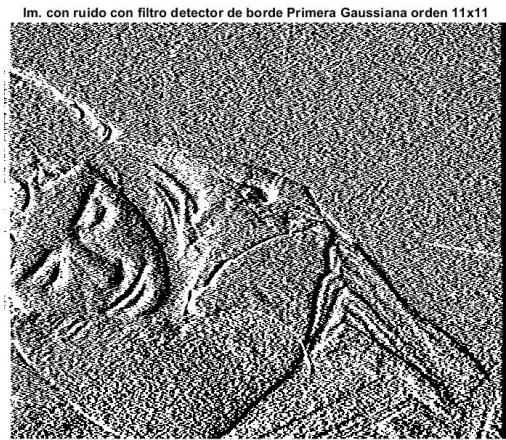


Fig. 30. Img. Con Ruido PDG 11x11

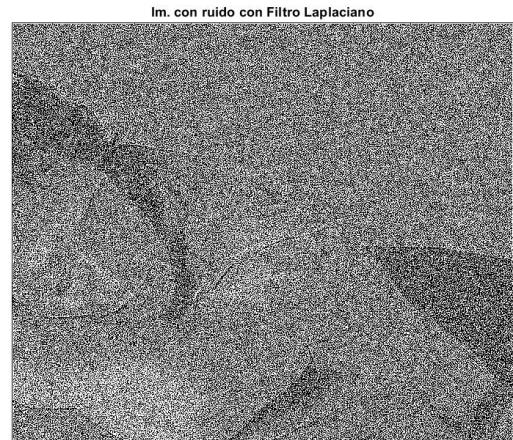


Fig. 32. Img. Con Ruido Laplaciano 3x3

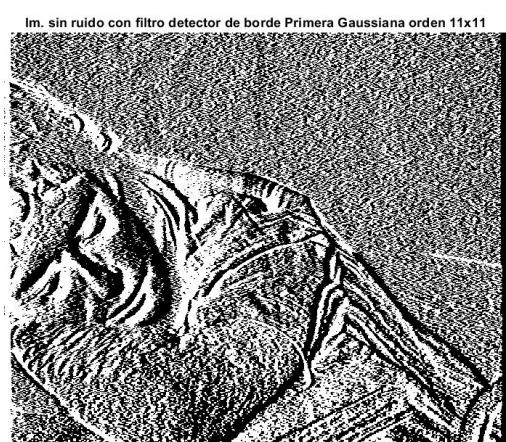


Fig. 31. Img. Sin Ruido PDG 11x11

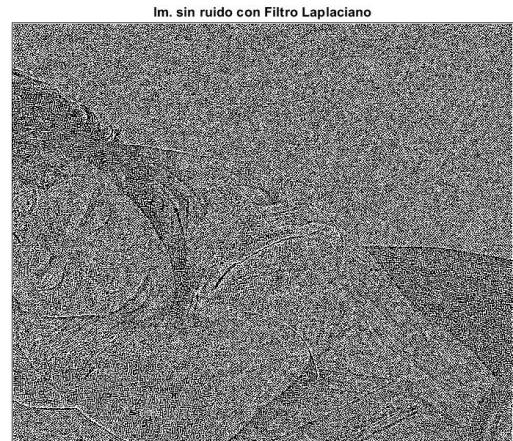
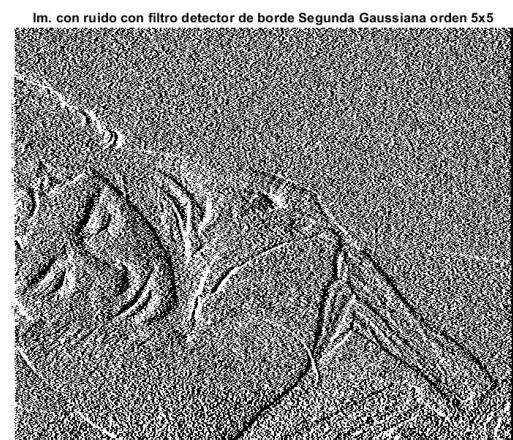


Fig. 33. Img. Sin Ruido Laplaciano 3x3

Laplacianos basados en la segunda derivada de Gaussiana de orden 5x5, 7x7 y 11x11, constrúyalos con el método visto en clase.



3.5 De igual manera, aplicar a la imagen sin ruido y a la imagen con ruido los filtros basados en la segunda derivada de gaussiana siguientes

El Laplaciano 3x3 que encuentre en la literatura, por ejemplo, el filtro con 8 al centro y rodeado de -1's

Fig. 34. Img. Con Ruido Segunda Derivada 5x5



Fig. 35. Img. Con Ruido Segunda Derivada 7x7

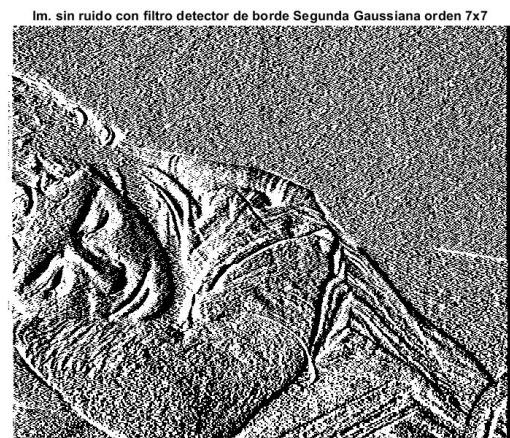


Fig. 38. Img. Sin Ruido Segunda Derivada 7x7

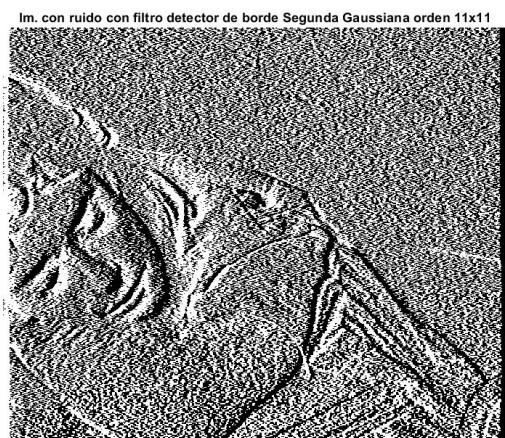


Fig. 36. Img. Con Ruido Segunda Derivada 11x11

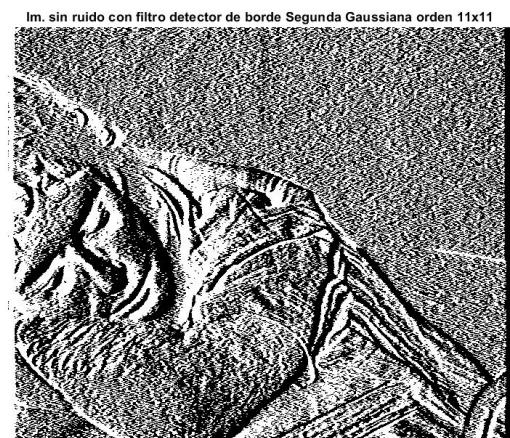


Fig. 39. Img. Sin Ruido Segunda Derivada 11x11



Fig. 37. Img. Sin Ruido Segunda Derivada 5x5

3.6 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

Filtro paso bajas de bloque de orden 3x3 y 7x7.



Fig. 40. Im. sin ruido con filtro suavizador de 5x5

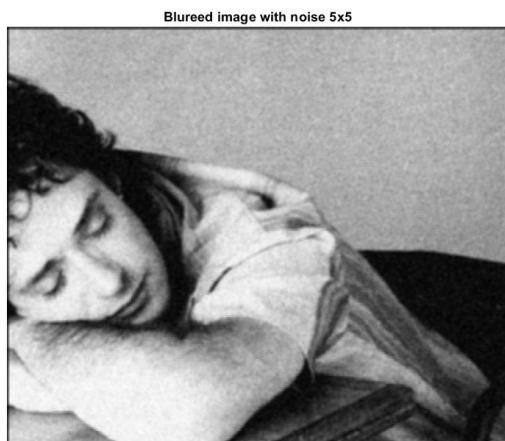


Fig. 41. Im. con ruido con filtro suavizador de 5x5

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



Fig. 42. Im. sin ruido: Realce filtro de bloque 3x3



Fig. 43. Im. con ruido: Realce filtro de bloque 3x3



Fig. 44. Im. sin ruido: Realce filtro de bloque 7x7

Im. con ruido: Realce filtro de bloque 7x7



Im. sin ruido: Realce filtro Binomial 7x7



Fig. 45. Im. con ruido: Realce filtro de bloque 7x7

Im. sin ruido: Realce filtro Binomial 3x3



Fig. 48. Im. sin ruido: Realce filtro binomial 7x7

Im. sin ruido: Realce filtro Binomial 7x7

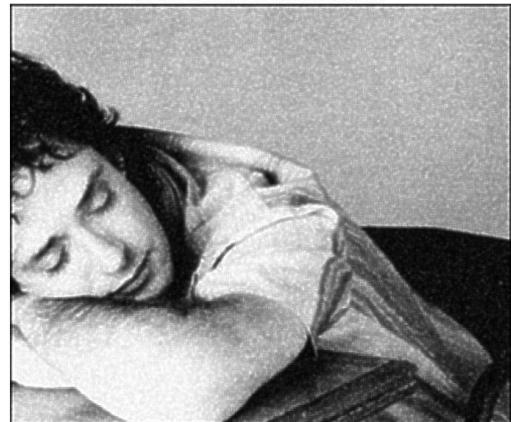


Fig. 46. Im. sin ruido: Realce filtro binomial 3x3

Im. sin ruido: Realce filtro Binomial 3x3



Fig. 49. Im. con ruido: Realce filtro binomial 7x7

Fig. 47. Im. con ruido: Realce filtro binomial 3x3

4. CÓDIGO

```
%Prctica 4: Filtros espaciales
%Task 1: Noise
img_c = imread('CERATI.jpg');
img = rgb2gray(img_c);
img_n = imnoise(img,'gaussian');

%Task 2: Blurred filter
b_filter_3 = ones(3)/9;
b_filter_7 = ones(7)/49;
b_filter_9 = ones(9)/81;
b_filter_11 = ones(11)/121;

%Also we can use this form img_b_3 =
    imfilter(img,b_filter_3);
img_b_3 = uint8(convn(img,b_filter_3));
img_b_7 = uint8(convn(img,b_filter_7));
img_b_9 = uint8(convn(img,b_filter_9));
img_b_11 = uint8(convn(img,b_filter_11));

img_nb_3 = imfilter(img_n,b_filter_3);
img_nb_7 = imfilter(img_n,b_filter_7);
img_nb_9 = imfilter(img_n,b_filter_9);
img_nb_11 = imfilter(img_n,b_filter_9);

figure(1);
%subplot(2, 1, 1);
imshow(img); title('Imagen original');
figure(2);
%subplot(2, 1, 2);
imshow(img_b_3); title('Im. sin ruido con filtro PB orden
    3x3');
figure(3);
%subplot(2, 1, 2);
imshow(img_b_7); title('Im. sin ruido con filtro PB orden
    7x7');
figure(4);
%subplot(2, 1, 2);
imshow(img_b_9); title('Im. sin ruido con filtro PB orden
    9x9');
figure(5);
%subplot(2, 1, 2);
imshow(img_b_11); title('Im. sin ruido con filtro PB orden
    11x11');

figure(6);
%subplot(2, 1, 2);
imshow(img_nb_3); title('Im. con ruido con filtro PB orden
    3x3');
figure(7);
%subplot(2, 1, 2);
imshow(img_nb_7); title('Im. con ruido con filtro PB orden
    7x7');
figure(8);
%subplot(2, 1, 2);
imshow(img_nb_9); title('Im. con ruido con filtro PB orden
    9x9');
figure(9);
%subplot(2, 1, 2);
imshow(img_nb_11); title('Im. con ruido con filtro PB
orden 11x11');

figure(10);
%subplot(2, 1, 2);
imshow(img_n); title('Im. con ruido');

%It is possible to change the size of the filter
img_c = imread('CERATI.jpg');
img = rgb2gray(img_c);
img_n = imnoise(img,'gaussian');

lp_img3 = imgaussfilt(img,2,'FilterSize',3); %Desviacion
estndar = 2
lp_img_n3 = imgaussfilt(img_n,2,'FilterSize',3);

lp_img7 = imgaussfilt(img,2,'FilterSize',7);
lp_img_n7 = imgaussfilt(img_n,2,'FilterSize',7);

lp_img9 = imgaussfilt(img,2,'FilterSize',9);
lp_img_n9 = imgaussfilt(img_n,2,'FilterSize',9);

lp_img11 = imgaussfilt(img,2,'FilterSize',11);
lp_img_n11 = imgaussfilt(img_n,2,'FilterSize',11);

figure(1);
imshow(lp_img3); title('Img. sin ruido con Filtro
    Gaussiano 3x3');
figure(2);
imshow(lp_img_n3); title('Img. con ruido con Filtro
    Gaussiano 3x3');

figure(3);
imshow(lp_img7); title('Img. sin ruido con Filtro
    Gaussiano 7x7');
figure(4);
imshow(lp_img_n7); title('Img. con ruido con Filtro
    Gaussiano 7x7');

figure(5);
imshow(lp_img9); title('Img. sin ruido con Filtro
    Gaussiano 9x9');
figure(6);
imshow(lp_img_n9); title('Img. con ruido con Filtro
    Gaussiano 9x9');

figure(7);
imshow(lp_img11); title('Img. sin ruido con Filtro
    Gaussiano 11x11');
figure(8);
imshow(lp_img_n11); title('Img. con ruido con Filtro
    Gaussiano 11x11');

%Task 4:
img_c = imread('CERATI.jpg');
img = rgb2gray(img_c);
img_n = imnoise(img,'gaussian');
% a) De bloque [1 -1]
simple_block = [1 -1];
img_simple_block = convn(img,simple_block);
img_n_simple_block = convn(img_n,simple_block);
% b) Prewitt ambas direcciones
pw_x = [-1,0,1 ; -1,0,1; -1,0,1];
pw_y = [-1,-1,-1 ; 0,0,0; 1,1,1];
img_pw_x = imfilter(img,pw_x);
img_pw_nx = uint8(conv2(img_n,pw_x));
img_pw_y = imfilter(img,pw_y);
img_pw_ny = uint8(conv2(img_n,pw_y));
% c) Sobel en ambas direcciones
s_x = [-1,0,1 ; -2,0,2; -1,0,1];
s_y = [-1,-2,-1 ; 0,0,0; 1,2,1];
img_s_x = imfilter(img,s_x);
img_s_nx = uint8(conv2(img_n,s_x));
img_s_y = imfilter(img,s_y);
img_s_ny = uint8(conv2(img_n,s_y));
```

```

% d)Primera Derivada Gaussiana
N = 11; %5,7,11
r = 1;
vec = [];

% %-----caso base-----
if N-r >= 0
    vec(1) = factorial(N-r) / factorial(N-r);
else
    vec(1) = 0;
end
% %-----

for x = 1:N
    if N-r >=0 && (N-r)-x >= 0
        f1 = factorial(N-r) / (factorial(x) *
            factorial((N-r)-x));
    else
        f1 = 0;
    end
    if N-r >=0 && x-r>= 0 && (N-r)-(x-r) >= 0
        f2 = factorial(N-r) / (factorial(x-r) *
            factorial((N-r)-(x-r)));
    else
        f2 = 0;
    end
    vec(x+1) = f1-f2;
end

img_pdg = convn(img,vec);
img_pdg_n = convn(img_n,vec);
%a)
figure(1);
imshow(img_simple_block); title('Im. sin ruido con filtro
detector de borde de Bloque simple');

figure(1);
imshow(img_n_simple_block); title('Im. sin ruido con
filtro detector de borde de Bloque simple');

%b)
figure(2);
imshow(img_pw_x); title('Im. sin ruido con filtro Prewitt
direccin en x');

figure(3);
imshow(img_pw_nx); title('Im. con ruido con filtro Prewitt
direccin en x');

figure(4);
imshow(img_pw_y); title('Im. sin ruido con filtro Prewitt
direccin en y');

figure(5);
imshow(img_pw_ny); title('Im. con ruido con filtro Prewitt
direccin en y');

%c)
figure(6);
imshow(img_s_x); title('Im. sin ruido con filtro Sobel
direccin x');

figure(7);
imshow(img_s_nx); title('Im. con ruido con filtro Sobel
direccin x');

figure(8);
imshow(img_s_y); title('Im. sin ruido con filtro Sobel
direccin y');

figure(9);
imshow(img_s_ny); title('Im. con ruido con filtro Sobel
direccin y');

%d)
figure(10);
imshow(img_pdg); title('Im. sin ruido con filtro detector
de borde Primera Gaussiana orden 11x11');

figure(11);
imshow(img_pdg_n); title('Im. con ruido con filtro
detector de borde Primera Gaussiana orden 11x11');

img = rgb2gray(img_c);
img_n = imnoise(img,'gaussian');

% a) Laplaciano 3x3
delta = [0 0 0; 0 1 0; 0 0 0];
hp = ones(3)/9;
MxH = delta - hp;
img_l = conv2(img,MxH);
img_ln = conv2(img_n,MxH);
% b)
N = 7; %5,7,11
r = 2;
vec = [];

% %-----caso base-----
if N-r >= 0
    vec(1) = factorial(N-r) / factorial(N-r);
else
    vec(1) = 0;
end
% %-----

for x = 1:N
    if N-r >=0 && (N-r)-x >= 0
        f1 = factorial(N-r) / (factorial(x) *
            factorial((N-r)-x));
    else
        f1 = 0;
    end
    if N-r >=0 && x-r>= 0 && (N-r)-(x-r) >= 0
        f2 = factorial(N-r) / (factorial(x-r) *
            factorial((N-r)-(x-r)));
    else
        f2 = 0;
    end
    vec(x+1) = f1-f2;
end

img_pdg = convn(img,vec);
img_pdg_n = convn(img_n,vec);

figure(1);
imshow(img_l); title('Im. sin ruido con Filtro
Laplaciano');
figure(2);
imshow(img_ln); title('Im. con ruido con Filtro
Laplaciano');

figure(3);
imshow(img_pdg); title('Im. sin ruido con filtro detector
de borde Segunda Gaussiana orden 7x7');

figure(4);
imshow(img_pdg_n); title('Im. con ruido con filtro
detector de borde Segunda Gaussiana orden 7x7');

```

```

%Task 6
img_c = imread('CERATI.jpg');
img = rgb2gray(img_c);
img_n = imnoise(img,'gaussian');

b_filter_5 = ones(5)/25;
img_b_5 = uint8(convn(img,b_filter_5));
img_bn_5 = uint8(convn(img_n,b_filter_5));

delta = [0 0 0; 0 1 0; 0 0 0];

%Adjust same sizes
img_b_5 = imresize(img_b_5,size(img));
img_bn_5 = imresize(img_bn_5,size(img));

% a) LPF order 3x3 y 7x7 (Block)
% Unsharp Masking Filter(Hum) with LPF 3x3 (Block)
k = 3;
b_filter_3 = ones(3)/9;
Flpf = uint8(convn(img_b_5,b_filter_3));
Flpf = imresize(Flpf,size(img_b_5));%Resize tamao
img_um3 = img_b_5 + k*(img_b_5-Flpf);

Flpf_n = uint8(convn(img_bn_5,b_filter_3));
Flpf_n = imresize(Flpf_n,size(img_b_5));%Resize tamao
img_um3_n = img_bn_5 + k*(img_bn_5-Flpf_n);
figure(1);
imshow(img_b_5); title('Blureed image 5x5');
figure(2);
imshow(img_um3); title('Unsharp Masking 3x3');
figure(3);
imshow(img_bn_5); title('Blureed image with noise 5x5');
figure(4);
imshow(img_um3_n); title('Unsharp Masking 3x3 with noise');

%Unsharp Masking Filter(Hum) with LPF 7x7
b_filter_7 = ones(7)/49;
Flpf_7 = uint8(convn(img_b_5,b_filter_7));
Flpf_7 = imresize(Flpf_7,size(img));%Resize tamao
img_um7 = img_b_5 + k*(img_b_5-Flpf_7);

Flpf_7_n = uint8(convn(img_bn_5,b_filter_7));
Flpf_7_n = imresize(Flpf_7_n,size(img));%Resize tamao
img_um7_n = img_bn_5 + k*(img_bn_5-Flpf_7_n);

figure(5);
imshow(img_um7); title('Unsharp Masking 7x7');
figure(6);
imshow(img_um7_n); title('Unsharp Masking 7x7 with noise');

% b)LPF order 3x3 y 7x7 (binomial)
%Unsharp Masking Filter(Hum) with LPF 3x3 (Binomial)
L = 2;
num = factorial(L);
pascal = [];
%-----caso base-----
den = factorial(0) * factorial(L-0);
coc = num / den;
pascal(1) = coc;
%-----
for i = 1:L
    den = factorial(i) * factorial(L-i);
    coc = num / den;
    pascal(i+1) = coc;
end
pascal_t= transpose(pascal);
filter3 = conv2(pascal_t,pascal)/L^2;%Hacemos filtro cuadrado y normalizamos
Flpf_3b = uint8(convn(img_b_5,filter3));
Flpf_3b = imresize(Flpf_3b,size(img));%Resize tamao

img_um3b = img_b_5 + k*(img_b_5-Flpf_3b);

figure(10);
subplot(2,1,1);imshow(img_b_5); title('Blureed image 5x5 (Binomial)');
subplot(2,1,2);imshow(img_um3b); title('Unsharp Masking 3x3 (Binomial)');

%Unsharp Masking Filter(Hum) with LPF 7x7 (Binomial)
L = 6;
num7 = factorial(L);
pascal7 = [];
%-----caso base-----
den7 = factorial(0) * factorial(L-0);
coc7 = num7 / den7;
pascal7(1) = coc7;
%-----
for i = 1:L
    den7 = factorial(i) * factorial(L-i);
    coc7 = num7 / den7;
    pascal7(i+1) = coc7;
end
pascal7_t= transpose(pascal7);
filter7 = conv2(pascal7_t,pascal7)/L^2;

Flpf_7b = uint8(convn(img_b_5,filter7));
Flpf_7b = imresize(Flpf_7b,size(img));%Resize tamao
img_um7b = img_b_5 + k*(img_b_5-Flpf_7b);
% lpf_image7 = uint8(convn(img_b_5,lpf_b7));%Aplicamos filtro a la imagen
% image_f7 = imresize(lpf_image7,size(img));%Resize Imagen filtrada pasa-bajas
% hpf7_b = minus(img_b_5,image_f7);%Restamos imagen original menos filtrada pasa-bajas
% img_umb7 = img_b_5 + k*(hpf7_b);%Hacemos la operación de sharp

figure(11);
subplot(2,1,1);imshow(img_b_5); title('Blureed image 5x5 (Binomial)');
subplot(2,1,2);imshow(img_um7b); title('Unsharp Masking 7x7 (Binomial)');

```

5. CONCLUSIONES

5.1 Carrillo Sánchez Ricardo

Esta práctica resultó de mucha utilidad para comprender conceptos básicos para la restauración de imágenes, en realidad el proceso de ecualización de las imágenes fue bastante sencillo y aunque consideramos que aun podemos percibir la unión entre las dos imágenes esta presenta mucha calidad. Considero que pudimos hacer un mejor realce en las zonas de grises ya que la parte izquierda se percibe con tonos más oscuros, nuestra propuesta y acercamiento fue la aplicación de un filtro de suavizado para promediar los tonos de grises en la unión. Eso mejoró mucho la imagen, pero el detalle se perjudicó un poco. Por lo anterior, tomando en cuenta los problemas presentados consideramos que se cumplieron los objetivos generales de la práctica.

5.2 Martínez Soto Mariana

Para esta práctica pudimos realizar parcialmente los objetivos, ecualizamos el histograma de una imagen usando

distintos niveles de cuantización. Sin embargo tuvimos dificultades a la hora de eliminar todos los trazos de esta unión, aunque la atenuación se ve drasticamente mejorada en los resultados finales para los dos imágenes satelitales contiguas al momento de procurar mostrar imperceptible esta unión (aunque se logra visualizar bastante mitigada respecto a la unión original). Por lo que consideramos un buen resultado final.

5.3 Sánchez Manjarrez Andrew

Para esta práctica coseguimos que a partir de dos imágenes satelitales, pudieramos generar un mosaico casi imperceptible en la unión transversal, aunque aún pueden observarse ligeramente trazas de la intersección, consideramos que no es un mal resultado. Comprendimos además técnicas de ecualización para los histogramas utilizando los diferentes niveles de cuantización, estas técnicas demuestran ser bastante efectivas en múltiples áreas como la restauración de imágenes.

5.4 Valenzuela García de León Fernando Rodrigo

Para la sumarización de esta práctica pudimos observar y analizar a detalle el proceso de ecualización de imágenes. Se puede apreciar como los filtros en diferentes extensiones en matrices de 3x3, 5x5 o 11x11, las cuales se puede apreciar como los filtros modifican una imagen, a la par de que se puede apreciar como se ven los filtros en comparación de imágenes con y sin ruido, viendo la capacidad grande que tienen los algoritmos para modificar imágenes.

REFERENCES

- [1] Overleaf Documentation (2021)
- [2] Matplotlib Documentation (2021)
[#introductory](https://matplotlib.org/tutorials/index.html)
- [3] Python Documentation (2021)
[#numbers](https://docs.python.org/3/tutorial/introduction.html)