

**Asignatura:** Procesamiento Digital de Imágenes

**Profesor:** Dr.Sc. Gerardo García Gil

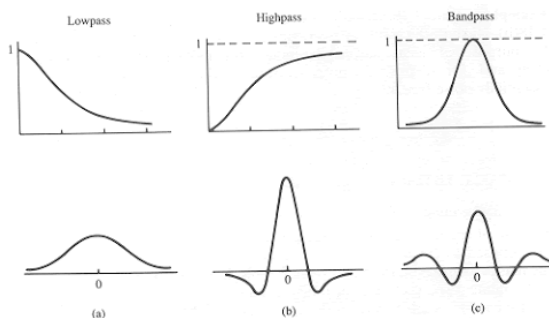
**Alumno:** René Francisco Coss y León Monterde

**Registro:** 17310066 **Semestre:** 2020-B  
**Ingeniería en Desarrollo de Software**

### INTRODUCCIÓN.

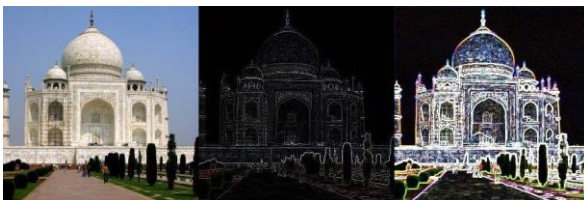
#### Filtrado (Imágenes Digitales)

Son técnicas dentro del preprocesamiento de imágenes para obtener, a partir de una imagen, otra que sea más adecuada para una aplicación específica, mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella.



#### Filtro (Imágenes Digitales)

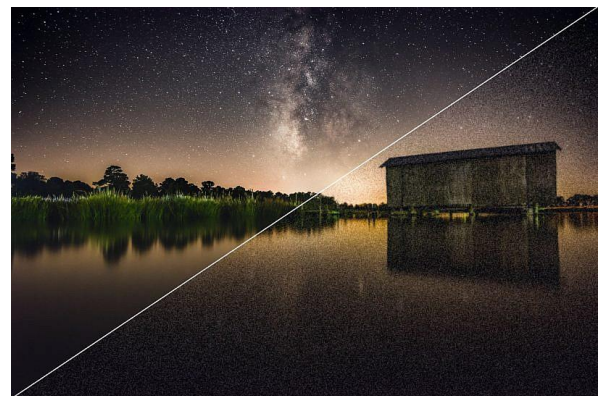
Es la aplicación de un filtrado de imagen para lograr el cambio en una imagen, el cual no depende únicamente el píxel original, sino de otros píxeles que están en una determinada vecindad en relación a este.



(Ejemplo: Filtro Laplaciano y Sobel)

### Ruido

Variación aleatoria (que no se corresponde con la realidad) del brillo o el color en las imágenes digitales, mayoritariamente producido por el dispositivo de entrada o captura de la imagen.



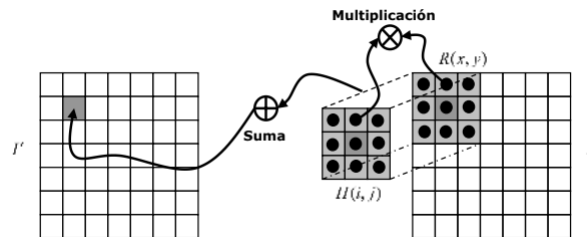
### Ruido Sal y Pimienta

Lo que define este tipo de ruido es que el píxel de ruido en cuestión no tiene relación alguna con los píxeles circundantes. Sobre la imagen hay puntos blancos sobre puntos negros o puntos negros sobre puntos blancos, de ahí el término sal y pimienta.

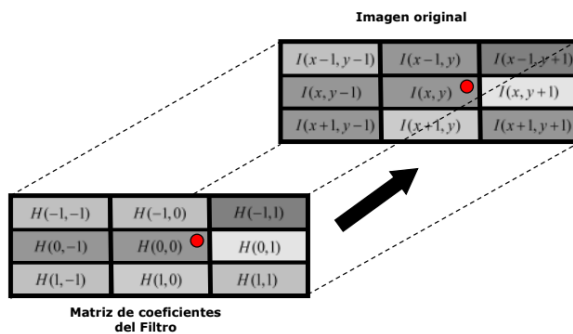


## Suavizado 1/9

Filtro en el cuál la intensidad del píxel cambia en función del promedio de la sumatoria de sus vecinos, considerando la vecindad 8-vecinos.



Se toma en cuenta la matriz de coeficientes del filtro para determinar las posiciones, tomando en cuenta que no todos los píxeles van a ser tratados de esta forma, pues los bordes no cuentan con 8 vecinos y esto generaría una indeterminación.



## Matemáticamente (Suavizado 1/9)

Se toma  $p_0$  como el píxel original, y se hace un promedio de la sumatoria con sus 8 vecinos.

$$I'(x, y) \leftarrow \frac{p_0 + p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8}{9}$$

Cada píxel tiene una coordenada, al ser una matriz de 3x3 se puede definir:

$$I'(x, y) \leftarrow \frac{1}{9} \cdot \begin{bmatrix} I(x-1, y-1) + I(x, y-1) + I(x+1, y-1) + \\ I(x-1, y) + I(x, y) + I(x+1, y) + \\ I(x-1, y+1) + I(x, y+1) + I(x+1, y+1) \end{bmatrix}$$

Generalizando la fórmula queda:

$$I'(x, y) \leftarrow \frac{1}{9} \cdot \sum_{j=-1}^1 \sum_{i=-1}^1 I(x+i, y+j)$$

## Pseudocódigo (Suavizado 1/9)

Matriz suave, gris  
Entero filas, columnas

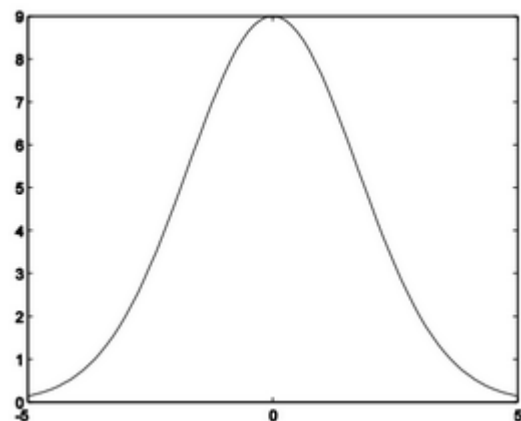
```
gris = ObtenerImagenGris()
filas = ObtenerFilasImagen(gris)
columnas = ObtenerColumnasImagen(gris)
Para x = 2 hasta filas-1
  Para y = 2 hasta columnas-1
    suave(x,y) = 1/9*
      (gris(x-1,y-1)+gris(x-1,y)+gris(x-1,y+1)
      +gris(x,y-1)+gris(x,y)+gris(x,y+1)
      +gris(x+1,y-1)+gris(x+1,y)+gris(x+1,y+1))
  Fin Para
Fin Para
ImprimirMatriz(suave)
```

## Suavizado Gauss

Filtro lineal el cual su matriz de coeficientes se obtiene a partir de la función bidimensional de Gauss, realizando un efecto de suavizado sobre una imagen.

0	1	2	1	0
1	3	5	3	1
2	5	9	5	2
1	3	5	3	1
0	1	2	1	0

Dicha matriz determina las posiciones, al igual que con la 1/9 pero se tendrían que definir los límites -2 debido a los bordes y al hecho de que la matriz crece de 3x3 a 5x5. Produciendo un efecto como el siguiente.



### Matemáticamente (Suavizado Gauss)

Corresponde una función discreta que dice:

$$G_{\sigma}(r) = e^{-\frac{r^2}{2\sigma^2}} \quad \text{O} \quad G_{\sigma}(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Donde  $\sigma$  representa el radio de cobertura de la función.

### Pseudocódigo (Suavizado Gauss)

Matriz suave, gris

Entero filas, columnas

imgr = ObtenerImagenGris()

filas = ObtenerFilasImagen(imgr)

columnas = ObtenerColumnasImagen(imgr)

Para x = 3 hasta filas-2

Para y = 3 hasta columnas-2

```
suave(x,y) = 1/25 * ((0*imgr(x-2,y-2)) +  
(1*imgr(x-2,y-1)) + (2*imgr(x-2,y)) +  
(1*imgr(x-2,y+1)) + (0*imgr(x-2,y+2)) +  
(1*imgr(x-1,y-2)) + (3*imgr(x-1,y-1)) +  
(5*imgr(x-1,y)) + (3*imgr(x-1,y+1)) +  
(1*imgr(x-1,y+2)) + (2*imgr(x,y-2)) +  
(5*imgr(x,y-1)) + (9*imgr(x,y)) +  
(5*imgr(x,y+1)) + (2*imgr(x,y+2)) +  
(1*imgr(x+1,y-2)) + (3*imgr(x+1,y-1)) +  
(5*imgr(x+1,y)) + (3*imgr(x+1,y+1)) +  
(1*imgr(x+1,y+2)) + (0*imgr(x+2,y-2)) +  
(1*imgr(x+2,y-1)) + (2*imgr(x+2,y)) +  
(1*imgr(x+2,y+1)) + (0*imgr(x+2,y+2))  
)
```

Fin Para

Fin Para

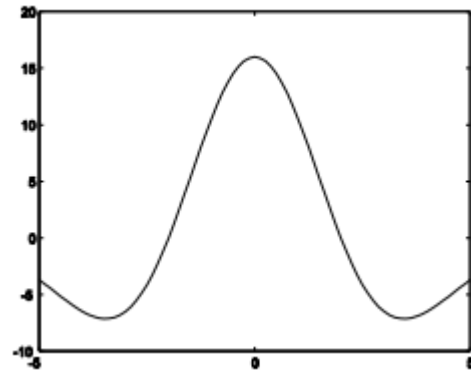
ImprimirMatriz(suave)

### Suavizado Laplace

Filtro lineal el cual su matriz de coeficientes se obtiene a partir de la función bidimensional de Laplace, realizando un efecto de suavizado sobre una imagen.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Dicha matriz determina las posiciones, al igual que con la 1/9 pero se tendrían que definir los límites -2 debido a los bordes y al hecho de que la matriz crece de 3x3 a 5x5. Produciendo un efecto como el siguiente (también conocido como sombrero mexicano).



### Matemáticamente (Suavizado Laplace)

Corresponde una función discreta que dice:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Aunque también se puede realizar una aplicación del filtro gaussiano al principio para obtener un mejor resultado, lo que daría: (Laplace of Gauss)

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Aunque se utiliza una máscara en este ejemplo que se basa directamente de la primera fórmula, sin filtro previo.

### Pseudocódigo (Suavizado Laplace)

Matriz suave, gris

Entero filas, columnas

imgr = ObtenerImagenGris()

filas = ObtenerFilasImagen(imgr)

columnas = ObtenerColumnasImagen(imgr)

Para x = 3 hasta filas-2

Para y = 3 hasta columnas-2

```
suave(x,y) = (0*imgr(x-2,y-2)) +  
(0*imgr(x-2,y-1)) + ((-1)*imgr(x-2,y)) +  
(0*imgr(x-2,y+1)) + (0*imgr(x-2,y+2)) +  
(0*imgr(x-1,y-2)) + ((-1)*imgr(x-1,y-1)) +  
+((-2)*imgr(x-1,y)) + ((-1)*imgr(x-1,y+1))
```

```

+(0*imgr(x-1,y+2)) +((-1)*imgr(x,y-2))
+((-2)*imgr(x,y-1)) +(16*imgr(x,y))
+((-2)*imgr(x,y+1)) +((-1)*imgr(x,y+2))
+(0*imgr(x+1,y-2)) +((-1)*imgr(x+1,y-1))
+((-2)*imgr(x+1,y)) +((-1)*imgr(x+1,y+1))
+(0*imgr(x+1,y+2)) +(0*imgr(x+2,y-2))
+(0*imgr(x+2,y-1)) +((-1)*imgr(x+2,y))
+(0*imgr(x+2,y+1)) +(0*imgr(x+2,y+2));

```

Fin

Fin

ImprimirMatriz(suave)

Para

Para

### Suavizado Mediana

Filtro lineal el cual su matriz de coeficientes se obtiene a partir de la ordenación ascendente de la matriz a 8 vecinos del píxel central, tomando el valor de la mediana para reemplazar al píxel actual y obtener un suavizado.

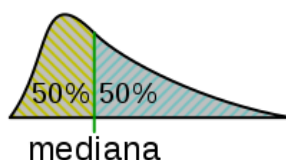
5	8	10
1	2	2
7	9	3

1	2	2
3	5	7
8	9	10

Dicha matriz primero obtiene el valor en cada posición, realiza un ordenamiento de menor a mayor, donde el menor valor ocupa el espacio (1,1) y el mayor ocupa la posición (3,3), determinando el valor de la posición central.

### Matemáticamente (Suavizado Mediana)

Corresponde a la definición de la mediana:



El acomodo de las frecuencias se hace en una distribución ascendente (menor a mayor) y se toma el valor en la posición a la mitad del acomodo. Si el acomodo es par, se toma el promedio de las 2 posiciones en la mitad.

$x_i$	$f_i$	$N_i$
$[x_{11}-x_{12}]$	$f_1$	$N_1$
.	.	.
.	.	.
.	.	$N_{(i-2)}$
$[x_{(i-1)1}-x_{(i-1)2}]$	$f_{(i-1)}$	$f_{(i-1)}+N_{(i-2)}=N_{i-1}$
$[x_{i1}-x_{i2}]$	$f_i$	$f_i+N_{i-1}=N_i$
$[x_{(i+1)1}-x_{(i+1)2}]$	$f_{(i+1)}$	$f_{(i+1)}+N_i=N_{(i+1)}$
.	.	.
.	.	.
.	.	.
$[x_{M1}-x_{M2}]$	$f_M$	$f_M+N_{(M-1)}=N_M$

### Pseudocódigo (Suavizado Mediana)

Matriz suave, gris, aux

Entero filas, columnas

imgr = ObtenerImagenGris()

filas = ObtenerFilasImagen(imgr)

columnas = ObtenerColumnasImagen(imgr)

aux = zeros(1,9);

Para x = 3 hasta filas-2

Para y = 3 hasta columnas-2

aux(1,1) = imgr(x-1,y-1);

aux(1,2) = imgr(x-1,y);

aux(1,3) = imgr(x-1,y+1);

aux(1,4) = imgr(x,y-1);

aux(1,5) = imgr(x,y);

aux(1,6) = imgr(x,y+1);

aux(1,7) = imgr(x+1,y-1);

aux(1,8) = imgr(x+1,y);

aux(1,9) = imgr(x+1,y+1);

aux = sort(aux);

suave(x,y) = aux(1,5);

Fin Para

Fin Para

ImprimirMatriz(suave)

## DESARROLLO.

Se obtiene la imagen a escala de grises con las funciones ya vistas, el proceso de lectura de un archivo de imagen es el mismo que se ha utilizado hasta el momento.

Se harán 2 comparaciones, una de los filtros aplicados a una imagen regular, y otra aplicados a una imagen con ruido, empezando con la primera respectivamente se obtiene la siguiente comparación directa de los 4 filtros vistos hasta ahora: Box 1/9, Gaussiano, Laplaciano y Mediana.

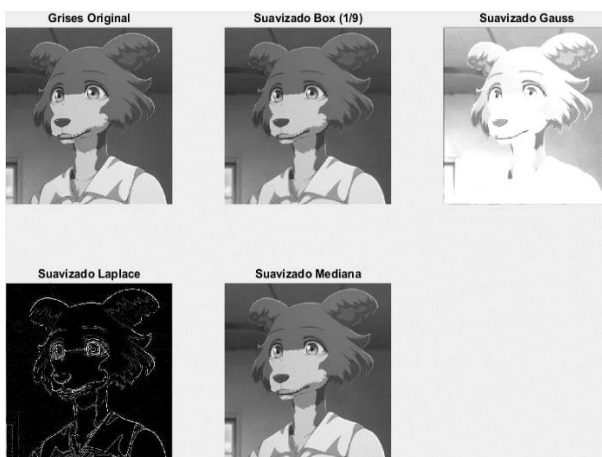


Fig1. Comparación de la imagen original con el efecto que cada filtro tiene sobre esta.

Después se aplica una máscara de ruido tipo sal y pimienta. Utilizando una función propia, que acepta un parámetro de imagen a grises, para aplicar dicho efecto en la imagen.



Fig2. Comparación de la imagen original (izquierda) con la aplicación del ruido sal y pimienta (derecha).

Entonces se realiza la comparación de el efecto que provoca cada filtro en una imagen con ruido, debido a su naturaleza y a lo visto en prácticas anteriores, se sabe que los filtros pueden requerir de iteraciones para mostrar una mejora en su eficacia, por ende se iteraron 5 veces para lograr ver el cambio.



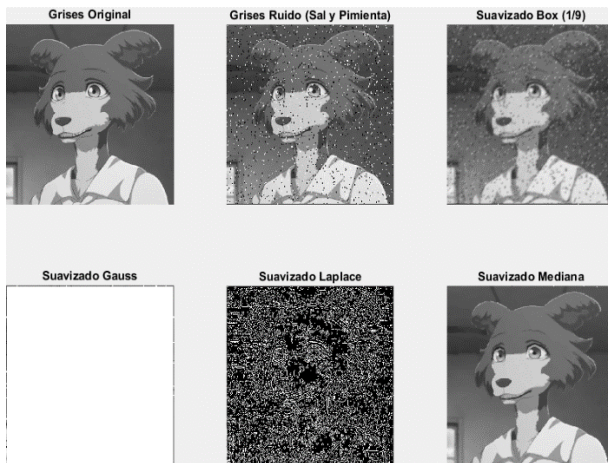


Fig3. Comparación de la imagen con ruido con el efecto que cada filtro tiene sobre esta

Por último, se comparan ambas comparaciones previamente vistas, es decir que veremos las diferencias una a lado de otra de cómo funcionan los filtros cuando la imagen cuenta con ruido y cuando no.

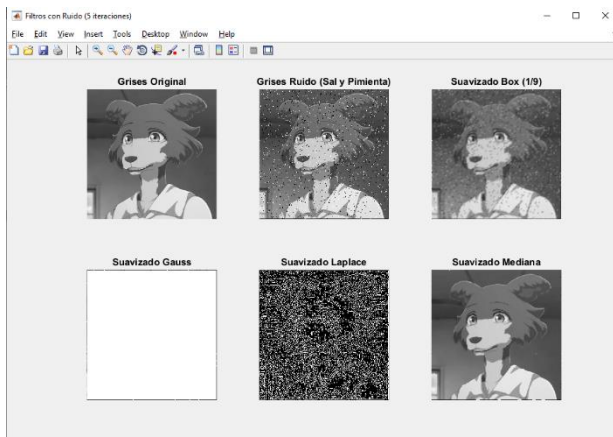


Fig4. Comparación final de todo el filtrado, imagen con ruido (izquierda) e imagen sin ruido (derecha)

Con esto la práctica esta por terminaba, podemos ver las diferencias lado a lado de absolutamente todos los filtros aplicados a diferentes objetivos que se han tomado en clase, abarcando desde el filtro box 1/9 hasta el filtro de mediana.

### Código

```
clear all; clc;
```

```
A = imread('junobest.jpg');
```

```
gr = escalagris(A);
imgaux = double(gr);
box = suavizar_box(imgaux);
gauss = suavizar_gauss(imgaux);
laplace = suavizar_laplace(imgaux);
mediana = suavizar_mediana(imgaux);
```

```
figure('Name','Filtros
Normales','NumberTitle','off')
subplot(2,3,1);
imshow(gr),title("Grises Original");
subplot(2,3,2);
imshow(box),title("Suavizado Box
(1/9)");
subplot(2,3,3);
imshow(gauss),title("Suavizado
Gauss");
subplot(2,3,4);
imshow(laplace),title("Suavizado
Laplace");
subplot(2,3,5);
imshow(mediana),title("Suavizado
Mediana");
```

```
ruido = ruido_salpimienta(gr);
imgaux = double(ruido);
```

```
box = suavizar_box(imgaux);
gauss = suavizar_gauss(imgaux);
laplace = suavizar_laplace(imgaux);
mediana = suavizar_mediana(imgaux);
```

```
for i=1:5
    box = double(box);
    gauss = double(gauss);
    laplace = double(laplace);
    mediana = double(mediana);

    box = suavizar_box(box);
    gauss = suavizar_gauss(gauss);
    laplace =
    suavizar_laplace(laplace);
    mediana =
    suavizar_mediana(mediana);
end
```

```
figure('Name','Filtros con Ruido (5
iteraciones)','NumberTitle','off')
subplot(2,3,1);
imshow(gr),title("Grises Original");
subplot(2,3,2);
imshow(ruido),title("Grises Ruido
(Sal y Pimienta)");
subplot(2,3,3);
imshow(box),title("Suavizado Box
(1/9)");
```

```

subplot(2,3,4);
imshow(gauss),title("Suavizado
Gauss");
subplot(2,3,5);
imshow(laplace),title("Suavizado
Laplace");
subplot(2,3,6);
imshow(mediana),title("Suavizado
Mediana");

function suave =
suavizar_mediana(imgr)
[x y] = size(imgr);
suave = imgr;
aux = zeros(1,9);
for r=2:x-1
    for c=2:y-1
        aux(1,1) = imgr(r-1,c-
1);
        aux(1,2) = imgr(r-1,c);
        aux(1,3) = imgr(r-
1,c+1);
        aux(1,4) = imgr(r,c-1);
        aux(1,5) = imgr(r,c);
        aux(1,6) = imgr(r,c+1);
        aux(1,7) = imgr(r+1,c-
1);
        aux(1,8) = imgr(r+1,c);
        aux(1,9) = imgr(r-
1,c+1);
        aux = sort(aux);
        suave(r,c) = aux(1,5);
    end
end

suave = uint8(suave);
end

function suave =
suavizar_laplace(imgr)
[x y] = size(imgr);
suave = imgr;
for r=3:x-2
    for c=3:y-2
        suave(r,c) = ((0*imgr(r-
2,c-2))+(0*imgr(r-2,c-1))+((-
1)*imgr(r-2,c))+(0*imgr(r-
2,c+1))+(0*imgr(r-2,c+2))...
        +(0*imgr(r-1,c-2))+((-
1)*imgr(r-1,c-1))+((-2)*imgr(r-
1,c))+((-1)*imgr(r-
1,c+1))+(0*imgr(r-1,c+2))...
        +((-1)*imgr(r,c-2))+((-
2)*imgr(r,c-1))+(16*imgr(r,c))+((-
2)*imgr(r,c+1))+((-
1)*imgr(r,c+2))...

```

```

        +(0*imgr(r+1,c-2))+((-
1)*imgr(r+1,c-1))+((-
2)*imgr(r+1,c))+((-
1)*imgr(r+1,c+1))+(0*imgr(r+1,c+2)).
    ..
        +(0*imgr(r+2,c-
2))+(0*imgr(r+2,c-1))+((-
1)*imgr(r+2,c))+(0*imgr(r+2,c+1))+(0
*imgr(r+2,c+2)));
    end
end

suave = uint8(suave);
end

function suave =
suavizar_gauss(imgr)
[x y] = size(imgr);
suave = imgr;
for r=3:x-2
    for c=3:y-2
        suave(r,c) =
1/25*((0*imgr(r-2,c-2))+(1*imgr(r-
2,c-1))+(2*imgr(r-2,c))+(1*imgr(r-
2,c+1))+(0*imgr(r-2,c+2))...
        +(1*imgr(r-1,c-
2))+(3*imgr(r-1,c-1))+(5*imgr(r-
1,c))+(3*imgr(r-1,c+1))+(1*imgr(r-
1,c+2))...
        +(2*imgr(r,c-
2))+(5*imgr(r,c-
1))+(9*imgr(r,c))+(5*imgr(r,c+1))+(2
*imgr(r,c+2))...
        +(1*imgr(r+1,c-
2))+(3*imgr(r+1,c-
1))+(5*imgr(r+1,c))+(3*imgr(r+1,c+1)
)+(1*imgr(r+1,c+2))...
        +(0*imgr(r+2,c-
2))+(1*imgr(r+2,c-
1))+(2*imgr(r+2,c))+(1*imgr(r+2,c+1)
)+(0*imgr(r+2,c+2)));
    end
end

suave = uint8(suave);
end

function suave = suavizar_box(imgr)
[x y] = size(imgr);
suave = imgr;
for r=2:x-1
    for c=2:y-1
        suave(r,c) =
1/9*(imgr(r-1,c-1)+imgr(r-
1,c)+imgr(r-1,c+1)+imgr(r,c-

```

```

1)+imgr(r,c)+imgr(r,c+1)+imgr(r+1,c-
1)+imgr(r+1,c)+imgr(r-1,c+1));
    end
end

```

```

    suave = uint8(suave);
end

```

```

function salpimienta =
ruido_salpimienta(imgr)
    salpimienta=imgr;
    [re co]=size(salpimienta);

```

```

for v=1:1000
    x=round(rand*re);
    y=round(rand*co);

```

```

    if x==0
        x=1;
    end

```

```

    if y==0
        y=1;
    end

```

```

end
if x==600
    x=598;
end

```

```

if y==800
    y=798;
end

```

```

    salpimienta(x,y)=255;
    salpimienta(x,y+1)=255;
    salpimienta(x+1,y)=255;
    salpimienta(x+1,y+1)=255;
    salpimienta(x+2,y)=255;
    salpimienta(x+2,y+1)=255;

```

```

end

```

```

for v=1:1000

```

```

    x=round(rand*re);
    y=round(rand*co);

```

```

    if x==0
        x=1;
    end

```

```

    if y==0
        y=1;
    end

```

```

    if x==600
        x=598;
    end

```

```

    if y==800
        y=798;
    end
end

```

```

    salpimienta(x,y)=0;
    salpimienta(x,y+1)=0;
    salpimienta(x+1,y)=0;
    salpimienta(x+1,y+1)=0;
    salpimienta(x+2,y)=0;
    salpimienta(x+2,y+1)=0;

```

```

end

```

```

end

```

```

function gris = escalagris(img)
    [r c z] = size(img);

```

```

    for i=1:r
        for j=1:c
            gris(i, j) = img(i,j,1)*0.2989
+ img(i,j,2)*0.5870 +
img(i,j,3)*0.1140;

```

```

        end
    end

```

```

end
end

```

## CONCLUSIONES

La comparación de filtros nos ayuda a visibilizar la utilidad de cada uno para diferentes casos, cabe denotar que la mayor diferencia en las comparaciones fue su utilidad para la eliminación de ruido o la mejora de la imagen y como se mencionó anteriormente cada filtro es diferente.

Claramente, el filtro de mediana es el mejor para la eliminación de ruido, seguido por el lineal de 1/9, el cual debido a la pérdida de definición se vuelve ineficiente, sin mencionar que tanto el Gaussiano como el Laplaciano fueron inútiles para este objetivo.

En cambio, la aplicación del filtro en una imagen sin ruido es donde el Laplaciano y Gaussiano destacan, puesto que estos cumplen su objetivo adecuadamente, con una sola aplicación el Laplaciano nos deja visualizar los bordes de la figura principal y Gaussiano tiene este aumento de brillo notorio, ambos bases para futuros procesos. En cambio, el 1/9 se vuelve situacional, pues este se aplica en toda la imagen en lugar de posiciones específicas que podrían aumentar su efectividad. Mientras el de Mediana simplemente no muestra ningún cambio relevante, incluso si se tuviera que aplicar en zonas determinadas como



el 1/9 su cambio no es tan efectivo y por ende el menos útil.

Es decir que depende del objetivo la efectividad, utilidad y uso de cada filtro cambia, volviéndose mejor o peor según se requiera, y eso es lo que esta comparación deja visualizar.

## BIBLIOGRAFÍA

colaboradores de Wikipedia. (2020, 5 agosto). Procesamiento digital de imágenes. Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Procesamiento\\_digital\\_de\\_im%C3%A1genes](https://es.wikipedia.org/wiki/Procesamiento_digital_de_im%C3%A1genes)

Filtrado espacial. (s. f.). uniovi. Recuperado 1 de octubre de 2020, de <http://www6.uniovi.es/vision/intro/node41.html>

colaboradores de Wikipedia. (2020a, agosto 14). Ruido en la fotografía digital. Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Ruido\\_en\\_la\\_fotograf%C3%ADa\\_digital](https://es.wikipedia.org/wiki/Ruido_en_la_fotograf%C3%ADa_digital)

colaboradores de Wikipedia. (2020b, septiembre 6). Mediana (estadística). Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Mediana\\_\(estad%C3%ADstica\)](https://es.wikipedia.org/wiki/Mediana_(estad%C3%ADstica))

Apuntes tomados en clase.