

**Asignatura:** Procesamiento Digital de Imágenes

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

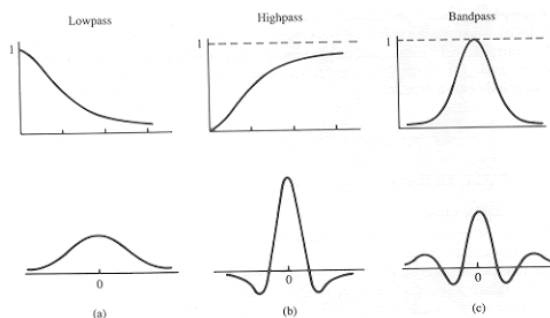
**Alumno:** René Francisco Coss y León Monterde

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

### INTRODUCCIÓN.

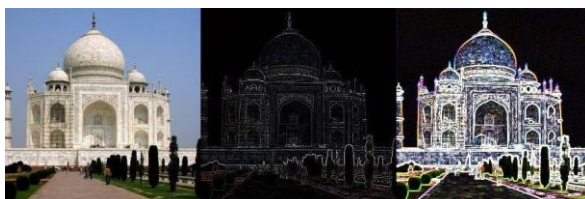
#### Filtrado (Imágenes Digitales)

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)

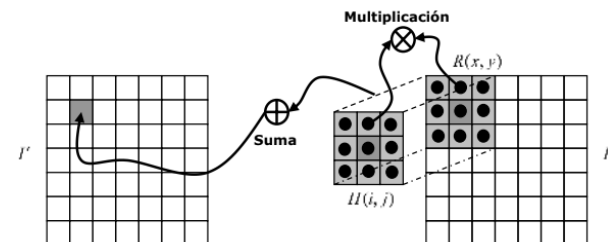
Es la aplicación de un filtrado de imagen para lograr el cambio en una imagen, el cual no depende únicamente el píxel original, sino de otros píxeles que están en una determinada vecindad en relación a este.



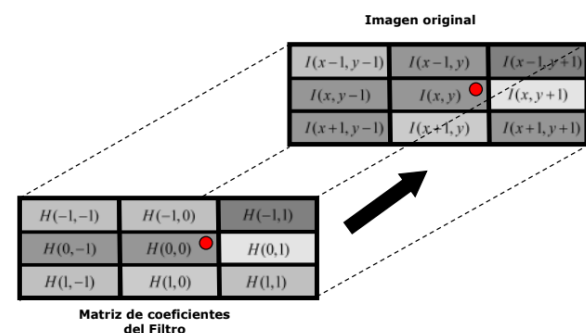
(Ejemplo: Filtro Laplaciano y Sobel)

### 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

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:**

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)

## 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 aplica la traducción del pseudocódigo a codificación de Matlab, siendo muy directo con la transformación de matrices. Este código nos va a dar como resultado una función que permite filtrar la imagen completamente, debido a que el proceso suele ser con variaciones muy pequeñas, es necesario repetirlo para obtener resultados más notorios, esto se hace con fines prácticos de comparación y no es una regla para todas las imágenes.

puede notar que es un resultado directo de aplicar la fórmula en la imagen.



Fig1. Comparación del proceso de filtrado, aplicado varias veces para demostrar la diferencia.

Al ser una función (en el programa) nosotros podemos llamarla cuantas veces sea necesaria para obtener mayores diferencias, aunque se

## Código

```
clear all; clc;

A = imread('lena.jpg');

gr = escalagris(A);
subplot(2,2,1);
imshow(gr),title("Grises Original");

gr = double(gr);
gr1 = suavizar_noveno(gr);
subplot(2,2,2);
imshow(gr1),title("Suavizado (1 vez)");

gr2 = suavizar_noveno(gr);
for i=1:3
    gr2 = double(gr2);
    gr2 = suavizar_noveno(gr2);
end
subplot(2,2,3);
imshow(gr2),title("Suavizado (3 veces)");

gr3 = suavizar_noveno(gr);
for i=1:5
    gr3 = double(gr3);
    gr3 = suavizar_noveno(gr3);
end
subplot(2,2,4);
imshow(gr3),title("Suavizado (5 veces)");

function suave =
suavizar_noveno(img)
    [x y] = size(img);
    suave = img;
    for r=2:x-1
        for c=2:y-1
```

```
            suave(r,c) = 1/9*(img(r-
1,c-1)+img(r-1,c)+img(r-
1,c+1)+img(r,c-
1)+img(r,c)+img(r,c+1)+img(r+1,c-
1)+img(r+1,c)+img(r-1,c+1));
        end
    end

    suave = uint8(suave);
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
```

## CONCLUSIONES

El efecto del linealizado  $1/9$  es más notorio conforme se hagan más repeticiones de este, esto no significa que sea mejor, pero nos deja observar que en algunos filtros la cantidad de iteraciones que se hagan del proceso van a brindar un resultado diferente. La principal limitante que se puede notar es que el suavizado no puede aplicarse a los bordes, además de que se vuelve notorio que el filtro aumenta su eficiencia si se limita las áreas donde se aplica, pero esto vuelve más difícil su automatización, pues se tendría que pasar la imagen por otro proceso anterior al filtrado para determinar en que zona se debe aplicar. En resumen, el filtrado brinda los resultados esperados, se puede mejorar delimitando la parte que afecta y se vuelve más notorio conforme más repeticiones se hagan.

## **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>

Apuntes tomados en clase.