

Asignatura: Procesamiento Digital de Imágenes

Profesor: Dr.Sc. Gerardo García Gil

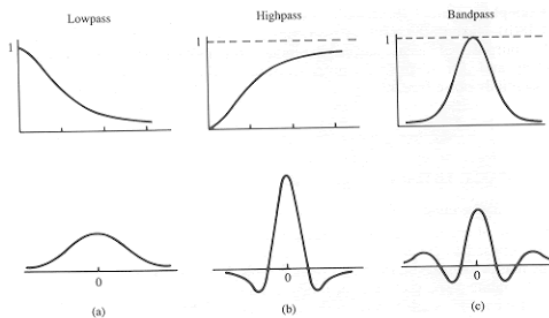
Alumno: René Francisco Coss y León Monterde

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

INTRODUCCIÓN.

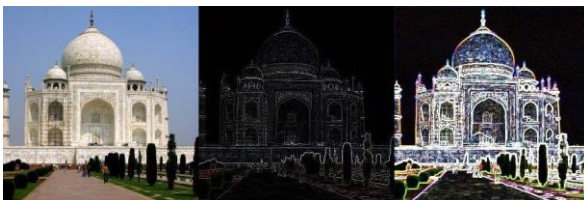
Filtrado (Imágenes Digitales)

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.



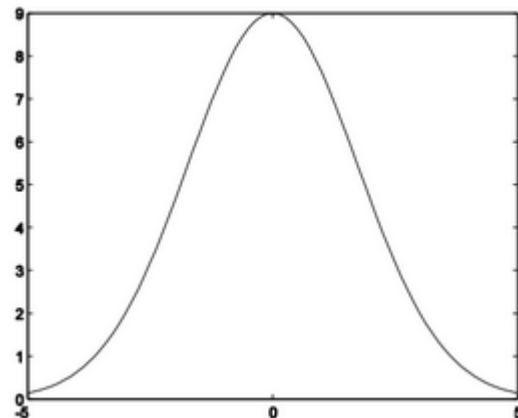
(Ejemplo: Filtro Laplaciano y Sobel)

Suavizado 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

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 σ representa el radio de cobertura de la función.

Pseudocódigo:

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)

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, en caso contrario al filtro $1/9$ este proceso es notorio desde la primera aplicación, por lo que su recursión no es necesaria para notar los cambios.



Fig1. Comparación del proceso de filtrado, aplicado una única vez a toda la imagen.

Al ser una función (en el programa) nosotros podemos llamarla cuantas veces sea necesaria para obtener mayores diferencias, aunque se puede notar que es un resultado directo de aplicar la fórmula en la imagen.

Código

```
clear all; clc;

A = imread('junobest.jpg');

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

gr = double(gr);

gr1 = suavizar_gauss(gr);
subplot(1,2,2);
imshow(gr1),title("Suavizado Gauss");

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 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 suavizado Gaussiano tiene un mejor efecto cuando se realiza por una única vez, aunque cabe denotar que los valores que se obtienen son muy grandes, por eso es necesario hacer la división con 25. Para efectos prácticos de este caso se promedió al dividir todo entre la cantidad total de pixeles que se tomaron en cuenta, es decir todo se multiplicó por el factor de $1/25$. Siguiendo el concepto del filtro lineal que tiene como

fundamento los suavizados. Podemos decir que el contraste de la imagen es mejor, o incluso que la imagen se ve más brillante, esto no significa que sea mejor, pero nos deja observar que el tipo de filtro varió mucho del anterior, y que al ser aplicado a toda la imagen brinda mejores resultados por lo que su automatización será más fácil. Para terminar, el resultado fue satisfactorio y se tomó en cuenta el factor de $1/25$ que no se había tomado en cuenta teóricamente.

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

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

Apuntes tomados en clase.