

**Asignatura:** Procesamiento Digital de Imágenes

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

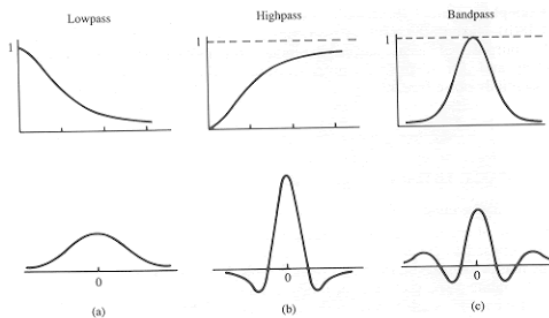
**Alumno:** René Francisco Coss y León Monterde

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

## INTRODUCCIÓN.

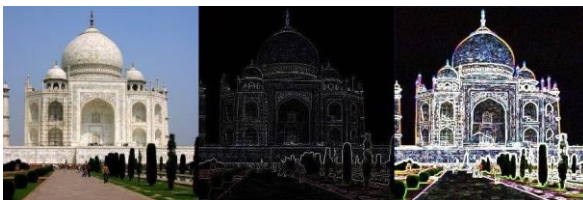
### Filtrado (Imágenes Digitales)

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)

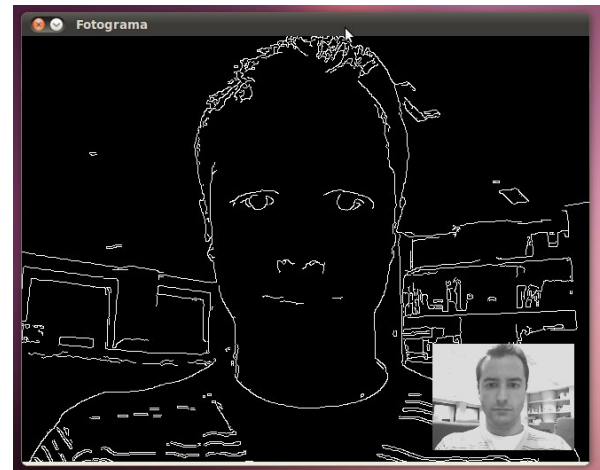
Es la aplicación de un filtrado de imagen para lograr el cambio en una imagen, el cual no depende únicamente el píxel original, sino de otros píxeles que están en una determinada vecindad en relación a este.



(Ejemplo: Filtro Laplaciano y Sobel)

## Detección de bordes

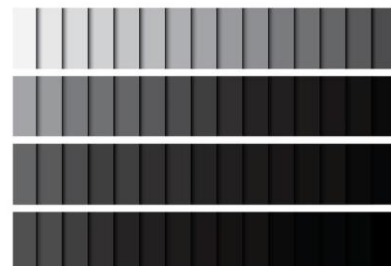
Los bordes de una imagen digital se definen como transiciones entre dos regiones de niveles de gris significativamente distintos.



Permiten reconocer las fronteras de los objetos y segmentar la imagen, así como detectar patrones para reconocer objetos.

## Gradiente

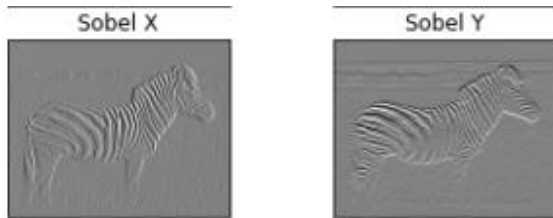
El gradiente de una imagen mide cómo esta cambia en términos de color o intensidad.



La magnitud del gradiente indica la rapidez con la que la imagen cambia, mientras que la dirección del gradiente indica la dirección en la que está cambiando.

### Filtro Sobel

Filtro utilizado para la detección de bordes y formas, donde el cálculo del gradiente local corresponde a la matriz de la segunda derivada con base al resultado de Prewitt.



### Matemáticamente

La matriz de coeficientes para este operador es definida como:

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{y} \quad H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Producen estimaciones del gradiente local para todos los píxeles de la imagen en sus dos diferentes direcciones, manteniendo la siguiente relación:

$$\nabla I(x,y) \approx \frac{1}{6} \begin{bmatrix} H_x^P \cdot I \\ H_y^P \cdot I \end{bmatrix} \quad \text{y} \quad \nabla I(x,y) \approx \frac{1}{8} \begin{bmatrix} H_x^S \cdot I \\ H_y^S \cdot I \end{bmatrix}$$

### Pseudocódigo

```
[m n]=size(imgr)
suave=double(imgr)
Gx=zeros(size(suave))
Gy=zeros(size(suave))
Para r=2:m-1
    Para c=2:n-1
        Gx(r,c)=-1*suave(r-1,c-1) -2*suave(r-1,c)
        -1*suave(r-1,c+1) +suave(r+1,c-1)
        +2*suave(r+1,c) +suave(r+1,c+1);
        Gy(r,c)= -1*suave(r-1,c-1) -2*suave(r,c-1)
        -1*suave(r+1,c-1) +suave(r-1,c+1)
        +2*suave(r,c+1) +suave(r+1,c+1)
    Fin Para
Fin Para
Gt=sqrt(Gx.^2+Gy.^2)
VmaxGt=max(max(Gt))
GtN=(Gt/VmaxGt)*255
GtN=uint8(GtN)
B=GtN>100
VminGx=min(min(Gx))
VminGy=min(min(Gy))
GradOffx=Gx-VminGx
GradOffy=Gy-VminGy
VmaxGx=max(max(GradOffx))
VmaxGy=max(max(GradOffy))
GxN=(GradOffx/VmaxGx)*255
GyN=(GradOffy/VmaxGy)*255
GxN=uint8(GxN)
GyN=uint8(GyN)

MostrarImagen(GxN)
MostrarImagen(GyN)
MostrarImagen(GtN)
```

## 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 realiza la traducción del pseudocódigo a codificación de Matlab, siendo muy directo con la transformación de matrices. Este código ocupa de una traducción muy bruta, puesto que las posiciones se obtienen fijas para cada píxel, lo que vuelve al código menos dinámico y más difícil de adaptar en caso de requerir una matriz más grande.

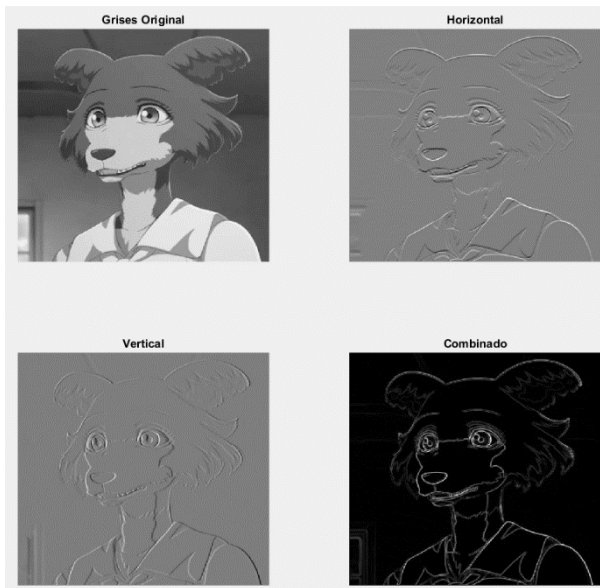


Fig1. Comparación de la imagen original (superior izquierda) con la aplicación de Sobel en cada eje y combinados.

Contrario a lo hecho en las practicas anteriores, la función de este filtro no se guarda en una variable, por lo que se vuelve un proceso en sí mismo y la reutilización del código a futuro es muy ineficiente. Es decir, que a nivel código, esta función no es modular, pero esto no tiene impacto en el efecto del filtro.

### Código

```
clear all; clc;

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

```
gr = escalagris(A);
figure('Name','Filtro Sobel','NumberTitle','off')
subplot(2,2,1);
imshow(gr),title("Grises Original");

filtrar_sobel(gr);

function filtrar_sobel(imgr)
    [m n]=size(imgr);
    suave = double(imgr);
    Gx=zeros(size(suave));
    Gy=zeros(size(suave));
    for r=2:m-1
        for c=2:n-1
            Gx(r,c)=-1*suave(r-1,c-1)-
2*suave(r-1,c)-1*suave(r-
1,c+1)+suave(r+1,c-
1)+2*suave(r+1,c)+suave(r+1,c+1);
            Gy(r,c)=-1*suave(r-1,c-1)-
2*suave(r,c-1)-1*suave(r+1,c-
1)+suave(r-
1,c+1)+2*suave(r,c+1)+suave(r+1,c+1)
;

            end
        end

        Gt=sqrt(Gx.^2+Gy.^2);
        VmaxGt=max(max(Gt));
        GtN=(Gt/VmaxGt)*255;
        GtN=uint8(GtN);

        B=GtN>100;
        VminGx=min(min(Gx));
        VminGy=min(min(Gy));
        GradOffx=Gx-VminGx;
        GradOffy=Gy-VminGy;
        VmaxGx=max(max(GradOffx));
        VmaxGy=max(max(GradOffy));
        GxN=(GradOffx/VmaxGx)*255;
        GyN=(GradOffy/VmaxGy)*255;

        GxN=uint8(GxN);
        GyN=uint8(GyN);

        subplot(2,2,2);
        imshow(GxN),title("Horizontal");
        subplot(2,2,3);
        imshow(GyN),title("Vertical");
        subplot(2,2,4);
        imshow(GtN),title("Combinado");
```

```

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 filtro de Sobel nos permite visualizar la detección de los bordes en ambos ejes, esto

significa que hace un escaneo en horizontal (x) y otro escaneo en vertical (y) para determinar los cambios de intensidad en ambas direcciones, la combinación de sus resultados es la que nos permite determinar donde se encuentran dichos bordes y así definir la forma de las figuras. Esto último se hace por lo regular binarizado para poder visualizar el efecto.

Si se combina con algún algoritmo de detección de figuras y formas irregulares, nosotros podríamos automatizar la detección de cosas y objetos específicos (tales como rostros, ropas, personas, vehículos, etc.), así pues, con los resultados obtenidos ya tenemos la base deseada para futuras operaciones, por lo que la práctica fue un éxito.

## **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>

Sosa-Costa, A. (2019, 30 agosto). Gradiente de Imágenes. ▷ Cursos de Programación de 0 a Experto © Garantizados. <https://unipython.com/gradiente-de-imagenes/>

Apuntes tomados en clase.