



Casa abierta al tiempo

**UNIVERSIDAD AUTÓNOMA METROPOLITANA**  
**Unidad Cuajimalpa**

**INGENIERIA EN COMPUTACION.**

**PROCESAMIENTO DE IMÁGENES**

**ALUMNO:**

GARCIA TORRES DANIEL  
ALEJANDRO

**PROFESOR:**

Oscar Sanchez Cortez

## INTRODUCCIÓN:

Para esta práctica se llevó a cabo la implementación de un histograma por medio de la librería JFreeChart, de igual manera se llevó a cabo la implementación de algunas funciones de la librería opencv.

## ANTECEDENTES:

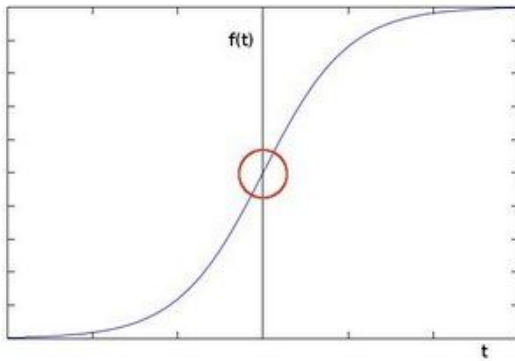
**Histograma.-** Un histograma es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados.

**Derivada de Sobel.-** El operador Sobel es un algoritmos de detección de bordes. Técnicamente es un operador diferencial discreto que calcula una aproximación al gradiente de la función de intensidad de una imagen.

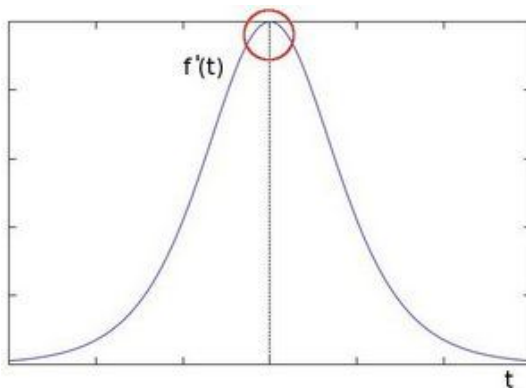


Puede notar fácilmente que, en un *borde* , la intensidad de píxel *cambia* de manera notoria. Una buena manera de expresar *cambios* es usando *derivados* . Un cambio alto en el gradiente indica un cambio importante en la imagen.

1. Supongamos que tenemos una imagen 1D. Un borde se muestra por la intensidad de "salto" en la siguiente gráfica:



2. El "salto" de borde se puede ver más fácilmente si tomamos la primera derivada (en realidad, aquí aparece como un máximo)



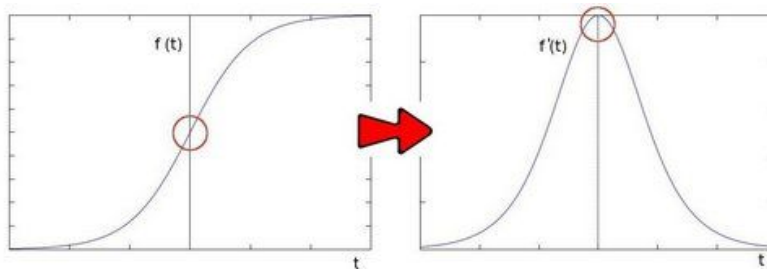
3. Por lo tanto, a partir de la explicación anterior, podemos deducir que un método para detectar bordes en una imagen puede realizarse ubicando las ubicaciones de los píxeles donde el gradiente es más alto que sus vecinos (o para generalizar, más alto que un umbral).

Ejemplo:

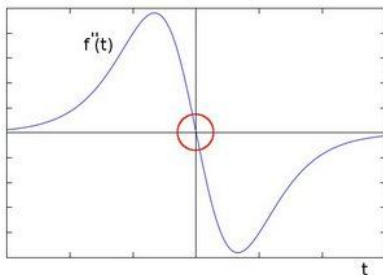


**Transformada de Laplace.-** La Transformada de Laplace es una técnica Matemática que forma parte de ciertas transformadas integrales como la transformada de Fourier, la transformada de Hilbert, y la transformada de Mellin entre otras. Estas transformadas están definidas por medio de una integral impropia y cambian una función en una variable de entrada en otra función en otra variable. Está al igual que la de Sobel sirve para detección de bordes.

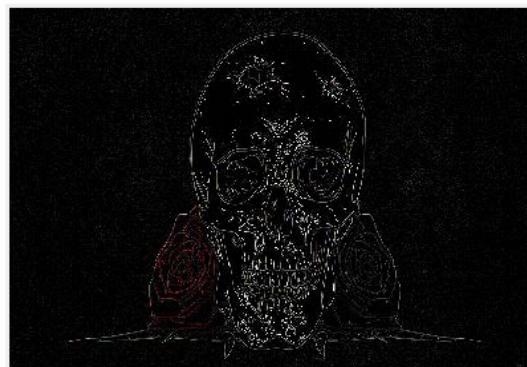
1. La intensidad de píxel muestra un "salto" o una variación alta de intensidad. Al obtener la primera derivada de la intensidad, observamos que un borde se caracteriza por un máximo, como se puede ver en la figura:



2. Y ... ¿qué pasa si tomamos el segundo derivado?



¡Puedes observar que la segunda derivada es cero! Por lo tanto, también podemos usar este criterio para intentar detectar bordes en una imagen.



**Transformada de Canny.-** También conocido por muchos como el detector óptimo , el algoritmo Canny pretende satisfacer tres criterios principales:

**Baja tasa de error:** lo que significa una buena detección de solo los bordes existentes.

**Buena localización:** la distancia entre los píxeles de borde detectados y los píxeles de borde real debe minimizarse.

**Respuesta mínima:** solo una respuesta del detector por borde.

1. Filtra cualquier ruido. El filtro gaussiano se utiliza para este propósito. A continuación se muestra un ejemplo de un núcleo gaussiano **size = 5** que podría usarse:

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

2. Encuentra el gradiente de intensidad de la imagen. Para esto, seguimos un procedimiento análogo a Sobel:

1. Aplique un par de máscaras de convolución (en **x** y **y** direcciones:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$
$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

2. Encuentra la fuerza y dirección del gradiente con:

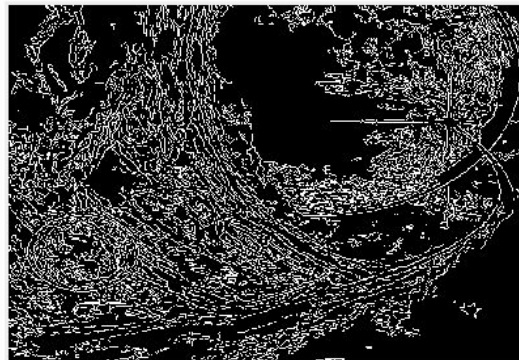
$$G = \sqrt{G_x^2 + G_y^2}$$
$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

La dirección se redondea a uno de los cuatro ángulos posibles (a saber, 0, 45, 90 o 135)

3. *No* se aplica supresión *máxima* . Esto elimina los píxeles que no se consideran parte de un borde. Por lo tanto, solo quedarán las líneas finas (bordes candidatos).
4. *Histéresis* : El paso final. Canny usa dos umbrales (superior e inferior):
  1. Si un gradiente de píxeles es más alto que el umbral *superior* , el píxel se acepta como un borde

2. Si un valor de gradiente de píxeles está por debajo del umbral *inferior* , se rechazará.
3. Si el gradiente de píxeles se encuentra entre los dos umbrales, solo se aceptará si está conectado a un píxel que esté por encima del umbral *superior* .

Ejemplo:



**Transformada discreta de Fourier.-** La transformada de Fourier se utiliza para analizar las características de frecuencia de varios filtros. Para las imágenes, se utiliza la Transformada Discreta de Fourier 2D (DFT) para encontrar el dominio de la frecuencia. Se usa un algoritmo rápido llamado Fast Fourier Transform (FFT) para el cálculo de DFT.



## **CONCLUSIÓN:**

Por medio de la práctica conocí más sobre opencv, a sí mismo como las diversas detecciones de bordes, y el las diversas implementaciones que se realizan en estas.