

# PRACTICA NO. 4

## FILTROS PASO ALTAS

### Objetivo

Implementar distintas técnicas para detección de bordes basadas en la primer y segunda derivada y analizar su comportamiento con el fin de identificar que técnica utilizar cuando se requiere extraer el contorno, líneas u puntos aislados en una imagen que puede estar en diversas situaciones como: bajo o alto contraste, brillante u oscura, con presencia de ruido o espuria, al resolver un problema real.

Análisis de Imágenes  
Profesora  
María Elena Cruz Meza



# Detección de bordes en una imagen

## **OBJETIVO**

Implementar algunas de las técnicas para la detección de bordes basadas en la primer y segunda derivada y analizar su comportamiento, con el fin de identificar que técnica utilizar cuando se requiere extraer el contorno, líneas u puntos aislados en una imagen que puede estar en diversas situaciones como: bajo o alto contraste, brillante u oscura, con presencia de ruido o espuria, al resolver un problema real.

## **Tareas particulares**

- a) Retomar la práctica Número 3 y ajustar la estructura para agregar los filtros detectores de bordes.
- b) Programar las técnicas basadas en la primera derivada o basadas en el gradiente:
  - a. Filtro Sobel
  - b. Filtro Prewitt
  - c. Filtro Robert
- c) Programar una de las técnicas basadas en la segunda derivada o Filtros Laplacianos
- d) Probar el funcionamiento de las técnicas con el conjunto de imágenes indicadas
- e) Mostrar los resultados y sus conclusiones mediante un informe establecido con la tarea #7 correspondiente al proyecto.

## **Instrucciones**

1. Conocimientos o requisitos previos
  - a) Haber completado la práctica No. 3
2. Sistema operativo y lenguaje de programación
  - a) Desarrollar tu práctica en el sistema operativo y lenguaje de programación que has elegido desde la práctica No. 1
3. Entrega de evidencia
  - a) Medio para su entrega: Entregar la tarea en la plataforma Microsoft Teams una vez terminada toda la práctica.
  - b) Esta práctica es fundamental para elaborar la tarea No.7 (Proyecto-Incluye el informe del desarrollo de esta práctica con los resultados obtenidos y las conclusiones conforme la rúbrica indicada para la tarea No 7. Consultar rúbrica).
  - c) Elaborar un informe con una nomenclatura específica y con el siguiente contenido:
    - Nombre al archivo como: InformePracticaFPA.FOR, donde “InformePracticaFPA” indica el número de práctica reportada que en este caso corresponde a la práctica 4, y “.FOR” es el formato correspondiente al archivo.
    - Portada: Es importante agregar algunos datos: identidad politécnica (Nombre o logos de la unidad académica: IPN y Escom), Nombre de la unidad de aprendizaje, Número de la

práctica, Objetivo de la práctica, Nombre los integrantes del equipo o integrante, es opcional mostrar el Nombre del(a) facilitador(A) (profesora) y Fecha de entrega.

- En el informe, la sección correspondiente al desarrollo de la práctica presente el análisis al efectuar la detección de bordes de las imágenes asignadas para la práctica (discusión con las conclusiones), anexe los segmentos de código correspondientes a la implementación de los filtros o bien anexe un archivo en formato PDF o TXT, con este código para su revisión y ponderación respectiva (el nombre de este archivo deberá nombrarse como “códigoFPA.txt”). Incluya en sus impresiones o conclusiones globales respecto al reuso del código de la practica anterior (filtros paso bajas).
- Programa y pruebas: Durante la presentación de los resultados del proyecto, se deberá mostrar el funcionamiento de esta práctica.
- En caso de no terminar todos o bien crea que aun tenga fallas en la programación, presentar los que se hayan realizado.
- Créditos: Agregar las referencias consultadas para el desarrollo de la actividad al final del documento.

# Fundamentos

## 1 DETECCIÓN DE BORDES.

Los bordes de una imagen digital se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos. Suministran una valiosa información sobre las fronteras de los objetos y puede ser utilizada para segmentar la imagen, reconocer objetos, etc.

La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen.

### 1.1. Operadores basados en la primera derivada (Gradiente).

La derivada de una señal continua proporciona las variaciones locales con respecto a la variable, de forma que el valor de la derivada es mayor cuanto más rápidas son estas variaciones.

En el caso de funciones bidimensionales  $f(x,y)$ , la derivada es un vector que apunta en la dirección de la máxima variación de  $f(x,y)$  y cuyo módulo es proporcional a dicha variación. Este vector se denomina gradiente y se define:

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

$$\text{Mag}[\nabla f(x,y)] = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

$$\theta = \arctan \frac{\frac{\partial f(x,y)}{\partial x}}{\frac{\partial f(x,y)}{\partial y}}$$

En el caso bidimensional discreto, las distintas aproximaciones del operador gradiente se basan en diferencias entre los niveles de grises de la imagen. La derivada parcial  $f_x(x,y)$  (gradiente de fila  $G_F(i,j)$ ) puede aproximarse por la diferencia de píxeles adyacentes de la misma fila.

$$\frac{\partial f(x,y)}{\partial x} \approx \nabla_x f(x,y) = f(x,y) - f(x-1,y)$$

-1	1
----	---

La discretización del vector gradiente en el eje Y ( $G_C(i,j)$ ), será:

$$\frac{\partial f(x,y)}{\partial y} \approx \nabla_y f(x,y) = f(x,y) - f(x,y-1)$$

-1
1

El gradiente de la fila  $G_F$  y de columna  $G_C$  en cada punto se obtienen mediante la convolución de la imagen con las máscaras  $H_F$  y  $H_C$ , esto es:

$$G_F(i, j) = F(i, j) \otimes H_F(i, j)$$

$$G_C(i, j) = F(i, j) \otimes H_C(i, j)$$

La magnitud y orientación del vector gradiente suele aproximarse por la expresión:

$$|G(i, j)| = \sqrt{G_F^2 + G_C^2} \approx |G_F(i, j)| + |G_C(i, j)|$$

Los operadores más utilizados son los de Roberts, Prewitt y Sobel y Frei-chen.

### Operador de Roberts.

Las máscaras utilizadas en este operador son:

Gradiente fila

0	0	0
0	0	1
0	-1	0

Gradiente columna

-1	0	0
0	1	0
0	0	0

Obtiene buena respuesta ante bordes diagonales. Ofrece buenas prestaciones en cuanto a localización. El gran inconveniente de este operador es su extremada sensibilidad al ruido y por tanto tiene pobres cualidades de detección.

### Operadores de Prewitt, Sobel y Frei-Chen.

Los tres operadores pueden formularse de forma conjunta con las siguientes máscaras de convolución mostradas a continuación.

Gradiente fila

$1/(2+k)$	1	0	-1
	K	0	-K
	1	0	-1

Gradiente columna

$1/(2+k)$	-1	-K	-1
	0	1	0
	1	K	1

En el caso del operador Prewitt ( $K=1$ ) se involucran a los vecinos de filas/columnas adyacentes para proporcionar mayor inmunidad al ruido.

El operador Sobel ( $K=2$ ), se supone que es más sensible a los bordes diagonales que el de Prewitt aunque en la práctica hay poca diferencia entre ellos.

Frei-Chen ( $K=\sqrt{2}$ ), el gradiente es el mismo para bordes verticales, horizontales y diagonales.

A estos detectores de bordes se les conoce como detectores de primer orden, debido a que estas operaciones utilizan la primera derivada digital para determinar los bordes. En la práctica (programación), se debe llevar a cabo completamente el desarrollo del filtro u operadores, es decir, debe efectuarse la multiplicación del factor por el kernel o máscara.

Para apreciar el efecto que producen los filtros basados en el gradiente, al aplicarlos en la figura 1, notemos que, como se muestra en la figura 2, en a) al aplicar una diferencia de píxeles, en una sola dirección, el resultado es una imagen oscura y los bordes tienen una intensidad uniforme. Al considerar otras referencias (direcciones del píxel, vecinos), en b) notemos que los bordes tienen una intensidad variable y son más claros. Sin embargo, en c), al efectuar un aproximado, la intensidad es alta, pero los bordes son muy gruesos.

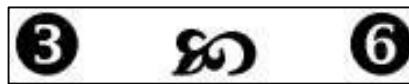


Fig. 1 Imagen original



Fig. 2 Operadores gradiente aplicados a la imagen 1, con sus distintos efectos.

## 1.2. Operadores basados en la segunda derivada (Laplaciano).

A las operaciones diferenciales digitales basadas en la segunda derivada les llamaremos operaciones de segundo orden:

Se basa en crear una máscara que destaque los píxeles (a través del aumento de su nivel de gris) cuya variación, con respecto a su vecindad, es significativa.

Se le denomina sharpening, que puede traducirse como enfoque de la imagen

- Destaca los detalles finos, bordes y otras discontinuidades de la misma a través de la aplicación de un filtro paso alto.

Si existiese una imagen formada por una matriz de una única fila, se podría calcular la variación de un píxel con respecto a su vecino aplicando el diferencial discreto de primer orden

El Filtro Laplaciano tiene la forma:

$$\frac{\partial f(y)}{\partial y} = f(y+1) - f(y)$$

Si a esta ecuación se le aplica el diferencial de segundo orden, resulta:

$$\frac{\partial^2 f(y)}{\partial y^2} = (f(y+1) - f(y)) - (f(y) - f(y-1)) = f(y+1) + f(y-1) - 2f(y)$$

El Laplaciano se define como el diferencial de segundo orden con respecto a dos variables, esto es:

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

- De esta ecuación, los términos derivativos se calculan como una extensión de:

$$\frac{\partial^2 f(x, y)}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

- Encontrando una equivalencia con una máscara de la forma:

$w(-1, -1)$	$w(-1, 0)$	$w(-1, 1)$
$w(0, -1)$	$w(0, 0)$	$w(0, 1)$
$w(1, -1)$	$w(1, 0)$	$w(1, 1)$

Figura 1. Modelo espacial de una máscara

Donde podemos acomodar los elementos que definen  $\nabla^2 f$  en una cuadrícula según su posición relativa al pixel en (x,y), tendremos que:

$$\nabla^2 f = \begin{bmatrix} & -f[x, y-1] & \\ -f[x-1, y] & f[x, y] & -f[x+1, y] \\ & -f[x, y+1] & \end{bmatrix}$$

Figura 2. Modelo geométrico de la máscara laplaciana

En este modelo geométrico podemos ver como el Laplaciano cuantifica la diferencia entre el tono del píxel ubicado en la posición (x, y) y sus vecinos horizontales-verticales laterales. Si formamos una matriz con los pixeles vecinos del píxel centrado en (x, y), es decir, una ventana de la imagen  $f$  de 3x3 alrededor del píxel citado y por otro lado extraemos los coeficientes del arreglo bidimensional en la expresión (1.2-1), tendremos el kernel o filtro laplaciano básico:

0	1	0
1	-4	1
0	1	0

Figura 3. Filtro Laplaciano

Se ha introducido un índice (+) por la forma en que están ubicados los términos (-1) en la matriz.

Otra variante en las máscaras laplacianas son la inversión del signo de los elementos que conforman la misma (Figura 3-a), o bien, podemos optar por otras variantes como se muestran en la siguiente figura (Figura 3 en b, c y d).

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

Figura 3. Algunas variantes del filtro laplaciano

Es importante destacar que el filtro laplaciano, al basarse en el diferencial de los píxeles de la vecindad:

- Destaca las variaciones en las tonalidades de gris,
- Mientras que atenúa la semejanza entre estos niveles



Por otro lado, recordemos que del mismo modo que en los filtros basados en el gradiente o primera derivada, en la implementación de los filtros laplacianos también se debe:

1. Aplicar la función valor absoluto al resultado. *Eliminación directa de valores negativos.*
2. si  $z < 0 \Rightarrow z = 0$ , si  $z > 2^L - 1 \Rightarrow z = 2^L - 1$ . *Ajuste al intervalo válido de salida.*

La aplicación del Laplaciano debe considerar que los bordes de la imagen (superior, inferior, derecho e izquierdo) no será posible calcularlo, dado que estos carecen de alguno de los vecinos requeridos para la evaluación, por lo tanto si la imagen tiene dimensiones horizontal- vertical  **$n \times m$** , entonces el ciclo de aplicación se debe aumentar o reducir un pixel en cada borde dependiendo del caso, por lo tanto el algoritmo puede quedar de la siguiente manera.

```

for (i = 1 .. n-2)
  for (j = 1 .. m-2) {
    s ← 0
    for (α = -1 .. 1)
      for (β = -1 .. 1)
        s ← s + M[α, β] * I[x+ α, y+ β]
    I'[x,y] ← s
  }

```

Fig. 3. Algoritmo para aplicar una convolución de 3x3.

Donde se han modelado los índices de la matriz de convolución entre -1 y 1 con el objetivo de acceder de manera simple a los elementos de matriz de la ventana de la imagen correspondientes.

## FUENTES

- Manuel Martín Ortiz (2003). Libro “Tratamiento digital de imágenes”. Profesor de la Facultad de Ciencias de la Computación en la Universidad Autónoma de Puebla, México. Disponible en: <http://mmartin.cs.buap.mx/notas/PDI-MM-Rev.2013.pdf>
- Gonzalo Pajares Martinsanz & Jesús M. de la Cruz García. Visión por computadora: imágenes digitales y aplicaciones. Ed. Alfaomega Ra-Ma. 2002
- González, RC y Woods, RE Digital Image Processing. Addison-Wesley, USA, 1992.
- Forsyth y Ponce. Computer Vision: A modern approach. Prentice-Hall, New Jersey, 2003.
- J. Parker. Algorithms for image processing and computer vision. John Wiley & Sons ed. (1997), 116-149.