

# VISIÓN POR COMPUTADORA

## Reporte - Tarea 6: Gradient operator

Jaime Rangel Ojeda  
*zS22000513@estudiantes.uv.mx*

Maestría en Inteligencia Artificial

IIIA Instituto de Investigaciones en Inteligencia Artificial  
Universidad Veracruzana  
*Campus Sur, Calle Paseo Lote II, Sección 2a, No 112*  
*Nuevo Xalapa, Xalapa, Ver., México 91097*

9 de mayo de 2023

### 1. Objetivo

La detección de bordes reduce significativamente la cantidad de datos y filtra la información innecesaria mientras se preserva las propiedades importantes de la imagen.

Sobel filter es usado para la detección de bordes, funciona calculando el gradiente de la imagen en cada pixel, el borde de la imagen se localiza en donde el brillo cambia significativamente. Este operador es sensible únicamente en las direcciones verticales y horizontales

## 2. Metodología

La plantilla de convolución del operador está dada por  $I$  = imagen y el kernel:

**Cambios Horizontales:** Haciendo la convolución  $I$  con el kernel  $G_x$

$$G_x = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} * I$$

**Cambios Verticales:** Haciendo la convolución  $I$  con el kernel  $G_y$

$$G_y = \begin{array}{|c|c|c|} \hline 1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} * I$$

Para el cálculo de gradiente está dado por  $G_x$  y  $G_y$  usando la siguiente formula:

$$G = \sqrt{G_x^2 + G_y^2}$$

La dirección del gradiente está dada por la formula:

$$\theta = \arctan\left(\frac{G_x}{G_y}\right)$$

### 3. Resultados

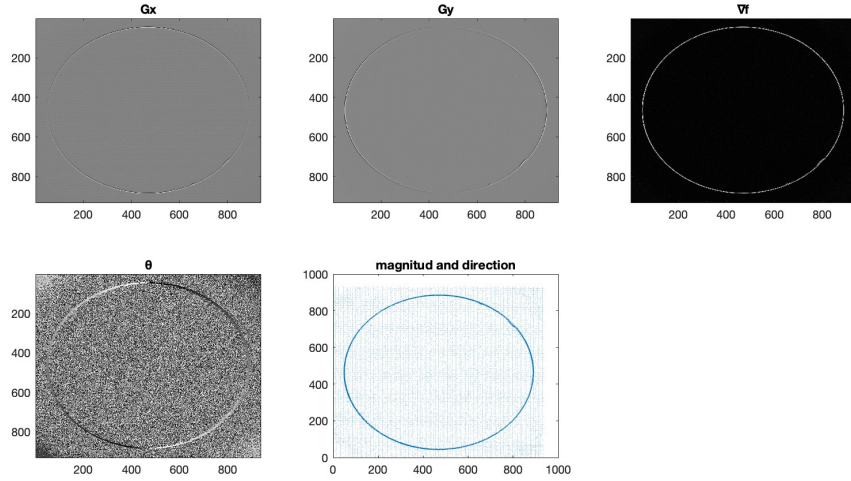


Figura 1: Resultados del operador gradiente

### 4. Conclusiones

Este método funciona con derivadas de primer orden, calcula por separado las derivadas de los ejes X y Y. Las derivadas son solo aproximaciones, para hacer la aproximación los kernels son usados para la convolución.

## 5. Código

```
1  clear all
2  close all
3  I = imread("Fig3.45(a).jpg");
4  I = double(I);
5
6  gx = [-1,-2,-1;0,0,0;1,2,1];
7  gy = [-1,0,1;-2,0,2;-1,0,1];
8
9  [rows,cols] = size(I);
10 g1 = zeros(rows,cols);
11 g2 = zeros(rows,cols);
12
13
14 for i=2:rows-1
15     for j=2:cols-1
16         i2=I(i-1:i+1,j-1:j+1);
17         % Se suman los extremos hasta i+1 y j+1
18         g1(i,j)=sum(sum(gx.*i2));
19         g2(i,j)=sum(sum(gy.*i2));
20     end
21 end
22
23 output = g1 + g2;
24 magnitud = sqrt(g1.^2+g2.^2);
25 direction = atan(g1 ./ g2);
26 deg = rad2deg(direction);
27
28 subplot(2,3,1)
29 imagesc(g1)
30 title('Gx')
31 colormap("gray");
32 subplot(2,3,2)
33 imagesc(g2)
34 title('Gy')
35 colormap("gray");
36 subplot(2,3,3)
37 imagesc(magnitud)
38 title('f')
```

```

39 colormap("gray");
40 subplot(2,3,4)
41 imagesc(deg)
42 title('')
43 colormap("gray");
44 subplot(2,3,5)
45 quiver(g1,g2)
46 title('magnitud and direction')
47 colormap("gray");
48
49

```

[Woo02] [Cha16]

## Referencias

- [Cha16] Jian-Dong Fang Chao-Chao Zhang. “Edge Detection Based on Improved Sobel Operator”. En: *Advances in Computer Science Research, (ACSR)*, 52.1 (2016), págs. 129-132.
- [Woo02] Rafael C. Gonzales Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2002.