Extracción de bordes

Ivan Cruz Aceves
ivan.cruz@cimat.mx

Centro de Investigación en Matemáticas, A.C. (CIMAT)

Enero del 2019
Cubo- I304, Ext. 4506

Contenido

Conceptos

Bordes: Son píxeles en los que la imagen presenta una brusca variación en los niveles de gris.

Conceptos

Bordes: Son píxeles en los que la imagen presenta una brusca variación en los niveles de gris.

Objetivo: Localizar los bordes generados por elementos de la escena contenida en la imagen y no por ruido.

Conceptos

Bordes: Son píxeles en los que la imagen presenta una brusca variación en los niveles de gris.

Objetivo: Localizar los bordes generados por elementos de la escena contenida en la imagen y no por ruido.

Edgels: Los píxeles o puntos de borde, también suelen denominarse *edgels*, del término *edge elements*.

Conceptos

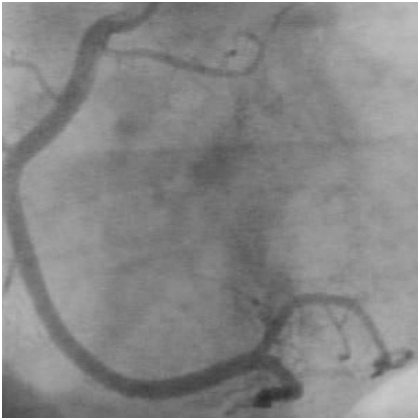
Bordes: Son píxeles en los que la imagen presenta una brusca variación en los niveles de gris.

Objetivo: Localizar los bordes generados por elementos de la escena contenida en la imagen y no por ruido.

Edgels: Los píxeles o puntos de borde, también suelen denominarse *edgels*, del término *edge elements*.

Representaciones: Contornos de objetos, sombras, marcas en objetos que pueden ser utilizados para tareas de reconocimiento, movimiento o calibración.

Seleccionando una ROI.



Seleccionando una ROI.

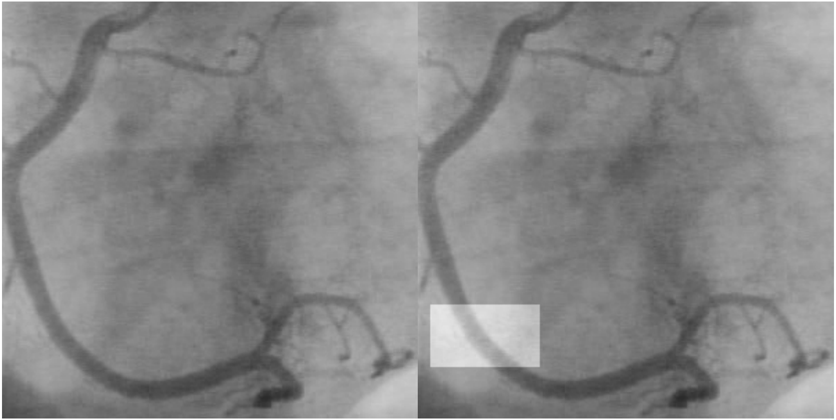


Figura: (a) Angiograma original, (b) ROI (resaltada).

ROI de 79×45 pixeles

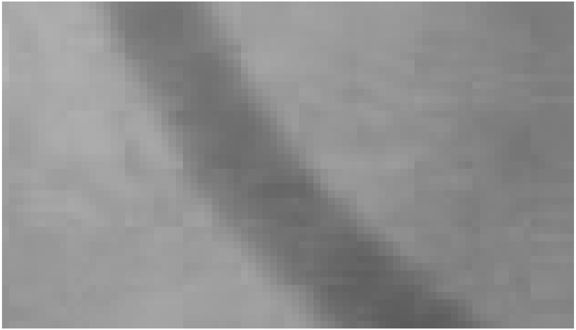


Figura: Región de interés original.

ROI en Negativo

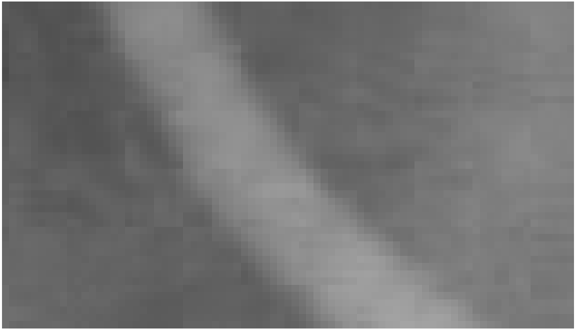


Figura: ROI en Negativo. $g(x, y) = 255 - f(x, y)$

ROI en Negativo

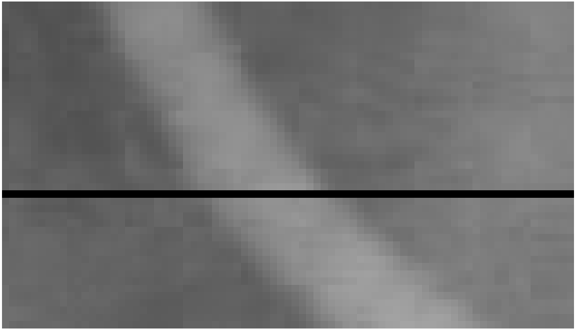
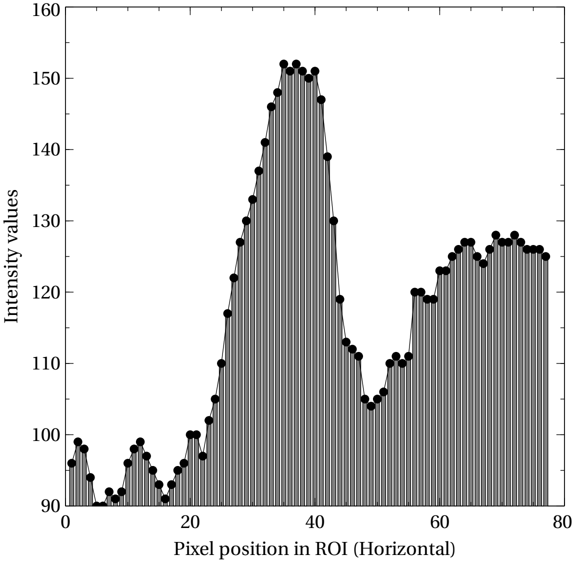


Figura: ROI en Negativo. $g(x, y) = 255 - f(x, y)$

Intensidad vs Posición (Transición oscuro-claro, viceversa)



Cálculo simple del gradiente

- En PI, no existe una forma universal de realizar tareas, en muchas ocasiones es tratar, ver resultados, y analizar cuál funciona mejor.

Cálculo simple del gradiente

- En PI, no existe una forma universal de realizar tareas, en muchas ocasiones es tratar, ver resultados, y analizar cuál funciona mejor.
- Existe varios operadores para calcular el gradiente de una imagen, todo depende de la aplicación.

Cálculo simple del gradiente

- En PI, no existe una forma universal de realizar tareas, en muchas ocasiones es tratar, ver resultados, y analizar cuál funciona mejor.
- Existe varios operadores para calcular el gradiente de una imagen, todo depende de la aplicación.

Gradiente: Es el cambio en niveles de gris en alguna dirección. Puede ser calculado mediante la diferencia de intensidades de pixeles vecinos.

Cálculo simple del gradiente

- En PI, no existe una forma universal de realizar tareas, en muchas ocasiones es tratar, ver resultados, y analizar cuál funciona mejor.
- Existe varios operadores para calcular el gradiente de una imagen, todo depende de la aplicación.

Gradiente: Es el cambio en niveles de gris en alguna dirección. Puede ser calculado mediante la diferencia de intensidades de pixeles vecinos.

Derivada: $f'(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$

Aproximaciones numéricas:

Diferencias hacia adelante: $f'(x) \approx \frac{f(x+h)-f(x)}{h}$

Diferencias hacia atrás: $f'(x) \approx \frac{f(x)-f(x-h)}{h}$

Cálculo simple del gradiente

Para calcular el *gradiente*, se tiene a dx y dy como las distancias (número de píxeles entre dos puntos) en las direcciones x e y, respectivamente.

$$G_x = \frac{f(x + d_x, y) - f(x, y)}{dx}$$

$$G_y = \frac{f(x, y + d_y) - f(x, y)}{dy}$$

Cálculo simple del gradiente

Para calcular el *gradiente*, se tiene a dx y dy como las distancias (número de píxeles entre dos puntos) en las direcciones x e y, respectivamente.

$$G_x = \frac{f(x + d_x, y) - f(x, y)}{dx}$$

$$G_y = \frac{f(x, y + d_y) - f(x, y)}{dy}$$

Calculando la magnitud y dirección mediante:

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

$$\Theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Cálculo simple del gradiente

- En imágenes, puede considerarse a dx y dy como la distancia entre dos puntos, en términos de píxeles.
- Por lo cual, si $dx = dy = 1$, podemos calcular el cambio en el gradiente para el punto (i, j) como:

$$G_x = f(i+1, j) - f(i, j)$$

$$G_y = f(i, j+1) - f(i, j)$$

Cálculo simple del gradiente

Considerando una muestra de 7 píxeles del perfil de intensidad previo, al cual denotamos como I, construiremos un nuevo vector I_G que contenga los valores del gradiente horizontal de I.

$I =$

150	153	147	139	130	119	113

Cálculo simple del gradiente

Considerando una muestra de 7 píxeles del perfil de intensidad previo, al cual denotamos como I, construiremos un nuevo vector I_G que contenga los valores del gradiente horizontal de I.

$$I = \boxed{150 \mid 153 \mid 147 \mid 139 \mid 130 \mid 119 \mid 113}$$

De la ecuación:

$$G_y = f(i, j+1) - f(i, j)$$

Podemos denotarla como:

$$I_G(i, j) = I(i, j+1) - I(i, j)$$

Cálculo simple del gradiente

Para obtener I_G sobre I calculamos:

$$I = \boxed{150 \mid 153 \mid 147 \mid 139 \mid 130 \mid 119 \mid 113}$$

$$I_G(i,j) = I(i,j+1) - I(i,j)$$

Cálculo simple del gradiente

Para obtener I_G sobre I calculamos:

$$I = \begin{array}{|c|c|c|c|c|c|c|} \hline 150 & 153 & 147 & 139 & 130 & 119 & 113 \\ \hline \end{array}$$

$$I_G(i,j) = I(i,j+1) - I(i,j)$$

$$I_G = \begin{array}{c} \hline I_G(0,0) = I(0,1) - I(0,0) = 153 - 150 = 3 \\ I_G(0,1) = I(0,2) - I(0,1) = 147 - 153 = -6 \\ I_G(0,2) = I(0,3) - I(0,2) = 139 - 147 = -8 \\ I_G(0,3) = I(0,4) - I(0,3) = 130 - 139 = -9 \\ I_G(0,4) = I(0,5) - I(0,4) = 119 - 130 = -11 \\ I_G(0,5) = I(0,6) - I(0,5) = 113 - 119 = -6 \\ \hline \end{array}$$

Cálculo simple del gradiente

Para obtener I_G sobre I calculamos:

$I =$

150	153	147	139	130	119	113

$$I_G(i,j) = I(i,j+1) - I(i,j)$$

$$I_G = \begin{array}{c}
\hline
I_G(0,0) = I(0,1) - I(0,0) = 153 - 150 = 3 \\
I_G(0,1) = I(0,2) - I(0,1) = 147 - 153 = -6 \\
I_G(0,2) = I(0,3) - I(0,2) = 139 - 147 = -8 \\
I_G(0,3) = I(0,4) - I(0,3) = 130 - 139 = -9 \\
I_G(0,4) = I(0,5) - I(0,4) = 119 - 130 = -11 \\
I_G(0,5) = I(0,6) - I(0,5) = 113 - 119 = -6 \\
\hline
\end{array}$$

$I_G =$

3	-6	-8	-9	-11	-6

Cálculo simple del gradiente

Aplicar el procedimiento anterior es igual que aplicar la convolución con el operador $A = [-1, 1]$.

$$I = \boxed{150 \mid 153 \mid 147 \mid 139 \mid 130 \mid 119 \mid 113}$$

Cálculo simple del gradiente

Aplicar el procedimiento anterior es igual que aplicar la convolución con el operador $A = [-1, 1]$.

$I =$

150	153	147	139	130	119	113

$$I_G = \begin{array}{c} \hline I_G(0,0) = -1 * I(0,0) + I(0,1) = -150 + 153 = 3 \\ I_G(0,1) = -1 * I(0,1) + I(0,2) = -153 + 147 = -6 \\ I_G(0,2) = -1 * I(0,2) + I(0,3) = -147 + 139 = -8 \\ I_G(0,3) = -1 * I(0,3) + I(0,4) = -139 + 130 = -9 \\ I_G(0,4) = -1 * I(0,4) + I(0,5) = -130 + 119 = -11 \\ I_G(0,5) = -1 * I(0,5) + I(0,6) = -119 + 113 = -6 \\ \hline \end{array}$$

$I_G =$

3	-6	-8	-9	-11	-6

Cálculo simple del gradiente

El problema de este operador ($A = [-1, 1]$) es que calcula el gradiente entre dos posiciones y no sólo de una.

Por lo cual, es imposible localizar el resultado centrado en el pixel empleado.

Debido a ello, se prefiere utilizar la siguiente ecuación:

$$I_G(i,j) = I(i,j+1) - I(i,j-1),$$

La cual puede ser representada por el siguiente vector:

$$I_G = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline\end{array}$$

Cálculo simple del gradiente

Aplicando $I_G(i,j) = I(i,j+1) - I(i,j-1)$ al problema anterior:

$I =$

150	153	147	139	130	119	113

Cálculo simple del gradiente

Aplicando $I_G(i,j) = I(i,j+1) - I(i,j-1)$ al problema anterior:

$$I = \boxed{150 \mid 153 \mid 147 \mid 139 \mid 130 \mid 119 \mid 113}$$

$I =$	150	153	147	139	130	119	113

$$I_G = \begin{array}{c} \hline I_G(0,1) = I(0,2) - I(0,0) = 147 - 150 = -3 \\ I_G(0,2) = I(0,3) - I(0,1) = 139 - 153 = -14 \\ I_G(0,3) = I(0,4) - I(0,2) = 130 - 147 = -17 \\ I_G(0,4) = I(0,5) - I(0,3) = 119 - 139 = -20 \\ I_G(0,5) = I(0,6) - I(0,4) = 113 - 130 = -17 \\ \hline \end{array}$$

$I_G =$	-3	-14	-17	-20	-17

Cálculo simple del gradiente

Aplicando $I_G(i,j) = I(i,j+1) - I(i,j-1)$ al problema anterior:

$I =$

150	153	147	139	130	119	113

$$I_G = \begin{array}{c} \hline I_G(0,1) = I(0,2) - I(0,0) = 147 - 150 = -3 \\ I_G(0,2) = I(0,3) - I(0,1) = 139 - 153 = -14 \\ I_G(0,3) = I(0,4) - I(0,2) = 130 - 147 = -17 \\ I_G(0,4) = I(0,5) - I(0,3) = 119 - 139 = -20 \\ I_G(0,5) = I(0,6) - I(0,4) = 113 - 130 = -17 \\ \hline \end{array}$$

$I_G =$

-3	-14	-17	-20	-17

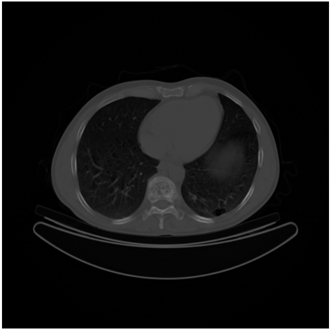
Es igual si aplicaramos la convolución con $A = [-1, 0, 1]$:

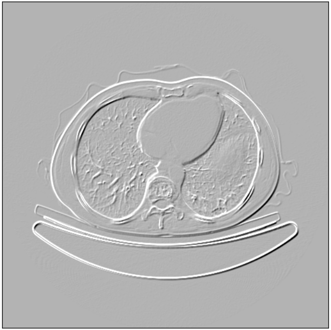
$$I_G(0,3) = -1 * I(0,2) + 0 * A(0,3) + A(0,4) = -147 + 130 = \mathbf{-17}$$

Similar es en el caso de la derivada vertical.

Operador de Sobel

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





Ejemplo de Convolución

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad I = \begin{bmatrix} 20 & 16 & 12 & 16 & 20 \\ 16 & {\color{blue}16} & 16 & 16 & 16 \\ 12 & 16 & 20 & 16 & 12 \\ 16 & 15 & 14 & 15 & 16 \\ 20 & 14 & 8 & 14 & 20 \end{bmatrix}$$

Ejemplo de Convolución

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad I = \begin{bmatrix} 20 & 16 & 12 & 16 & 20 \\ 16 & \color{blue}{16} & 16 & 16 & 16 \\ 12 & 16 & 20 & 16 & 12 \\ 16 & 15 & 14 & 15 & 16 \\ 20 & 14 & 8 & 14 & 20 \end{bmatrix}$$

$(-1 * 20) + (0 * 16) + (1 * 12) = -8$
$(-2 * 16) + (0 * 16) + (2 * 16) = 0$
$(-1 * 12) + (0 * 16) + (1 * 20) = 8$
Como resultado en $I_{G_x}(1, 1) = \color{blue}{0}$

Filtro de Sobel en C++ (parte 1)

```
void SobelBorder(unsigned char *data, int w, int h){
GrayScale(data,w,h);
//convert vector to matrix
float** img_mat = new float* [h];
for (int i=0; i < h; i++)
img_mat[i] = new float[w]; int con=0;
for( int y = 0 ; y < h ; y++ )
for( int x = 0 ; x < w ; x++ ){
img_mat[y][x] = data[con];
con = con+3; }
double Gx[9] = {-1,0,1,-2,0,2,-1,0,1};//Sobel Gx
double Gy[9] = {-1,-2,-1,0, 0, 0,1, 2, 1};//Sobel Gy
int gx1=0,gy1=0;
```

Filtro de Sobel en C++ (parte 2)

```
float** img_x = new float* [h];
for (int i=0; i < h; i++)
img_x[i] = new float[w];
float** img_y = new float* [h];
for (int i=0; i < h; i++)
img_y[i] = new float[w];
for(int x=0; x < h; x++)          //Sobel Gx
for(int y=0; y < w; y++){
if((x==0)||(y==0)||(y==(w-1))||(x==(h-1)))
gx1 = 0; else {
gx1=(((img_mat[x-1][y-1]*Gx[0])+(img_mat[x-1][y]*Gx[1])+
(img_mat[x-1][y+1]*Gx[2])+(img_mat[x][y-1]*Gx[3])+
(img_mat[x][y]*Gx[4])+(img_mat[x][y+1]*Gx[5])+
(img_mat[x+1][y-1]*Gx[6])+(img_mat[x+1][y]*Gx[7])+
(img_mat[x+1][y+1]*Gx[8]))/1.0)+127;
if(gx1 > 255)gx1=255; if(gx1 < 0)gx1=0;}
img_x[x][y] = gx1; }
```

Filtro de Sobel en C++ (parte 3)

```
//Sobel Gy
for(int x=0; x < h; x++)
for(int y=0; y < w; y++)
{
if((x==0)||(y==0)||(y==(w-1))||(x==(h-1)))
gy1 = 0;
else
{
gy1 = (((img_mat[x-1][y-1]*Gy[0])+
(img_mat[x-1][y]*Gy[1])+(img_mat[x-1][y+1]*Gy[2])+
(img_mat[x][y-1]*Gy[3])+(img_mat[x][y]*Gy[4])+
(img_mat[x][y+1]*Gy[5])+(img_mat[x+1][y-1]*Gy[6])+
(img_mat[x+1][y]*Gy[7])+
(img_mat[x+1][y+1]*Gy[8]))/1.0)+127;
if(gy1 > 255)gy1=255;
if(gy1 < 0)gy1=0; }
img_y[x][y] = gy1; }
```

Filtro de Sobel en C++ (parte 4)

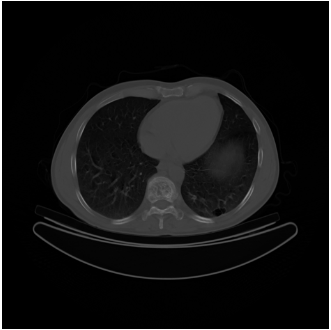
```
//Obtencion y asignacion de magnitud
int gval=0;
for(int x=0; x < h; x++)
for(int y=0; y < w; y++)
{
if((x==0)||(y==0)||(y==(w-1))||(x==(h-1)))
{
gval=0;
}
else
{
gval = (int) sqrt(pow(img_x[x][y],2)+
pow(img_y[x][y],2));
if(gval<0)gval=0;
if(gval>255)gval=255; }
img_mat[x][y] = gval; }
```

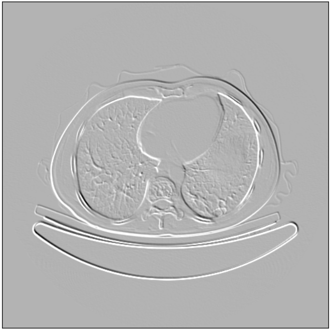
Filtro de Sobel en C++ (parte 5)

```
//convert from matrix to vector
con=0;
for( int y = 0 ; y < h; y++ )
for( int x = 0 ; x < w; x++ )
{
data[con + 0] = img_mat[y][x];
data[con + 1] = img_mat[y][x];
data[con + 2] = img_mat[y][x];
con = con+3;
}
}
```

Filtro de Prewitt

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





Máscaras de Kirsch

0,45,90,135,180,225,270,315 grados

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

Otros operadores

Diferencia de Pixeles:

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

Diferencia de Pixeles separados:

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

Otros operadores

Operador de Roberts:

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

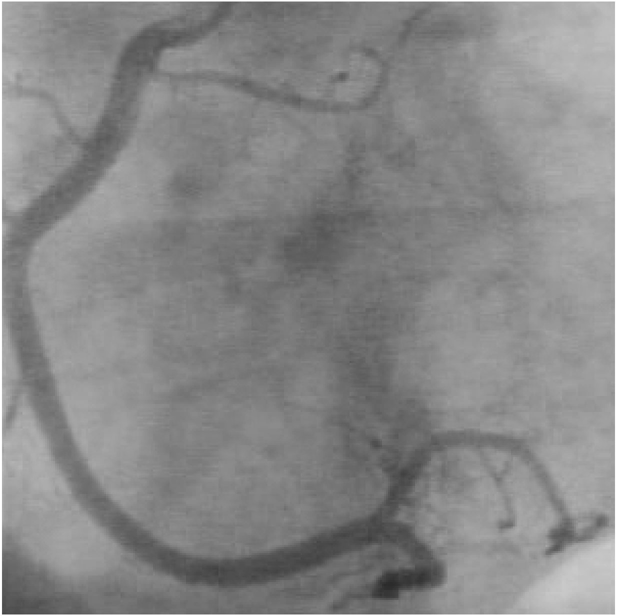
Máscaras de Frei-Chen:

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

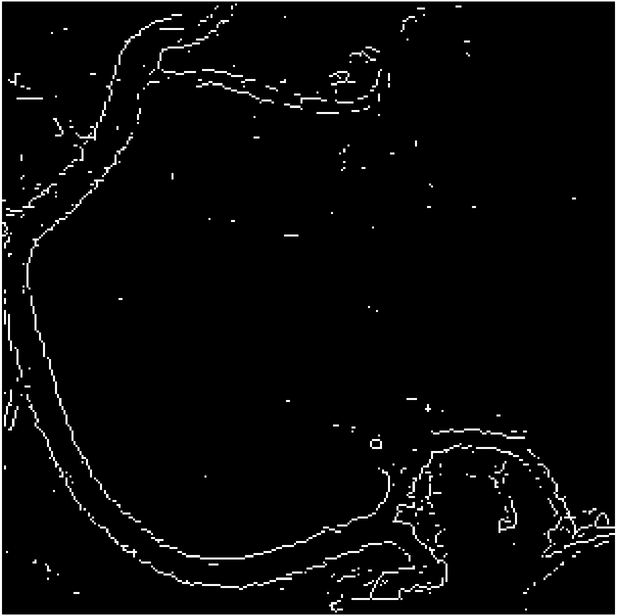
Más detectores de bordes

- Máscaras de Robinson
- Método de Nevatia-Babu
- Operador Laplaciano
- Algoritmo de Canny
  - Suavizado de imagen
  - Obtención del gradiente
  - Supresión de no máximos
  - Histeresis de umbral (2 umbrales)
  - Cierre de contornos
- Edge detection with fuzzy cellular automata transition function optimized by PSO.

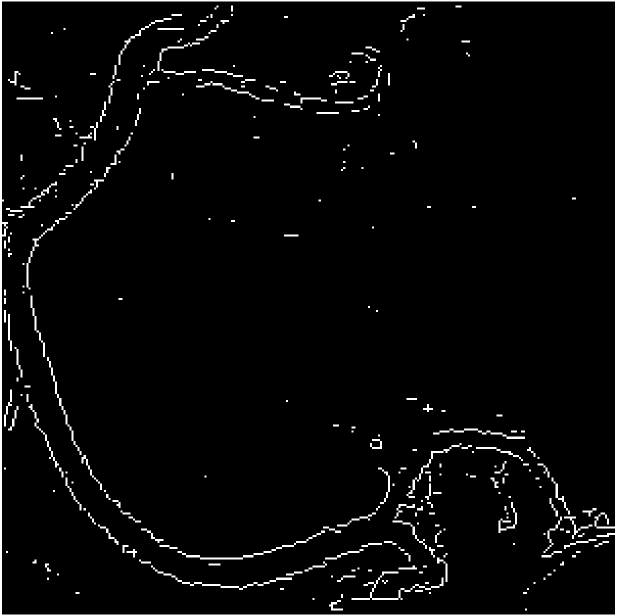
Imagen de Prueba



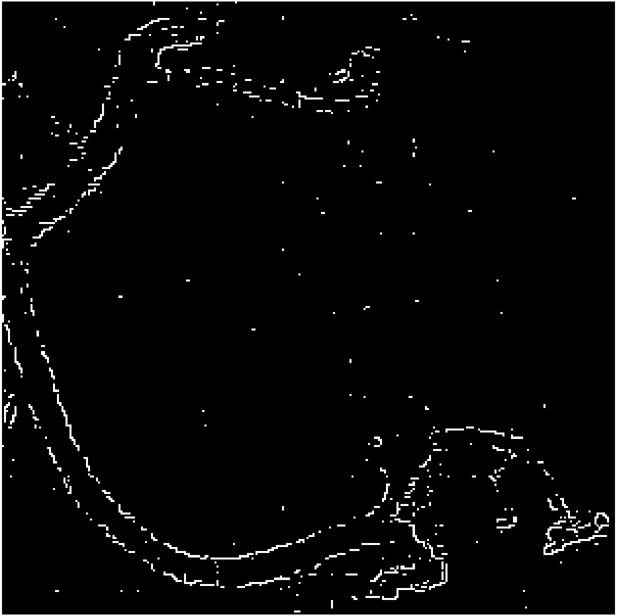
Aplicación de Sobel



Aplicación de Prewitt



Aplicación de Roberts



Aplicación de Canny

Tarea 1

- Analizar y reproducir por partes el método publicado en el paper: *Edge detection with fuzzy cellular automata transition function optimized by PSO*.
- De un método de detección de bordes, analizar y concluir sobre la definición de su operador de gradiente de acuerdo al autor.