

## الگوریتم Edge detection

الگوریتم های پردازش تصویر :

تصویر اصلی یک ماتریس  $m*n$  است که  $m$  طول تصویر و  $n$  عرض تصویر را مشخص می کند. هر عضو از این ماتریس بیانگر شدت نور تابیده شده به دوربین است و ماتریس edge معادل این ماتریس با درایه های اولیه ۰ است

الگوریتم لبه یابی edge detection :

اگر یک پیکسل با پیکسل مجاورش اختلافی بیش از یک مقدار آستانه داشته باشد، به آن مقدار آستانه ، آستانه ی لبه یابی و عنصر کرانه لبه می نامیم.

تعریف ماتریس edge :

یک ماتریس  $m*n$  است که مقادیر آن شامل صفر و یک می باشد. که صفر نشان دهنده ی نداشتن لبه و یک نشان دهنده ی داشتن لبه است.

نکته: در این الگوریتم، مقدار آستانه ، هیورستیکی بدست می آید.

ماتریس هسته ی کرنل :

یک ماتریس  $k*k$  است که شعاع لبه یابی را مشخص می کند. این ماتریس یک ماتریس فرد است. (عددی فرد  $k=$ ).

$$\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & +1 \\ 0 & +1 & +1 \end{bmatrix}$$

Vertical & Horizontal

$$\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}$$

Vertical

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

Horizontal

تمام درایه های ماتریس کرنل را در عنصر معادل در تصویر ضرب می کنیم و در نهایت همه ی نتایج را با هم جمع می کنیم. اگر مقدار به دست آمده از مقدار آستانه بیشتر باشد، عدد متناظر در ماتریس edge را یک می گذاریم.

الگوریتم edge detection برای حالت  $k=3$  :

```
for i = 1: Image_size
    for j = 1: Image_size
        result = Image[i-1,j-1]*kernel[0,0] + Image[i,j-1]*kernel[1,0] +
        Image[i+1,j-1]*kernel[2,0]
        + Image [i-1,j]*kernel[0,1] + Image[i,j]*kernel[1,1] +
        Image[i+1,j]*kernel[2,1]
        + Image[i-1,j+1]*kernel[0,2] + Image[i,j+1]*kernel[1,2]
        + Image[i+1,j+1]*kernel[2,2]
        if (result )> threshold
            edge[i,j]= 1
        else
            edge[i,j]= 0
        end
    end
end
end
```

## Edge Detector Types

Introduction

### Gradient Based Edge Detection

Gaussian Operators used

Roberts Operator

$$D_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad D_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Sobel Operator

$$D_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad D_y = \begin{bmatrix} -1 & -2 & +1 \\ 0 & 0 & 0 \\ -1 & +2 & +1 \end{bmatrix}$$

Prewitt Operator

$$D_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad D_y = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

