Edge Detection

Intro :-

Edge :- • Edge contains pixels of high gradient

↳ Boundary b/w 2 homogeneous regions.

↳ significant local change of intensity in a digital image

•

• Edge is typically extracted by computing the derivative of the image intensity funcⁿ.

Computation of derivatives

→ Magnitude (edge strength

→ Direction (edge orientation)

Gradient :-

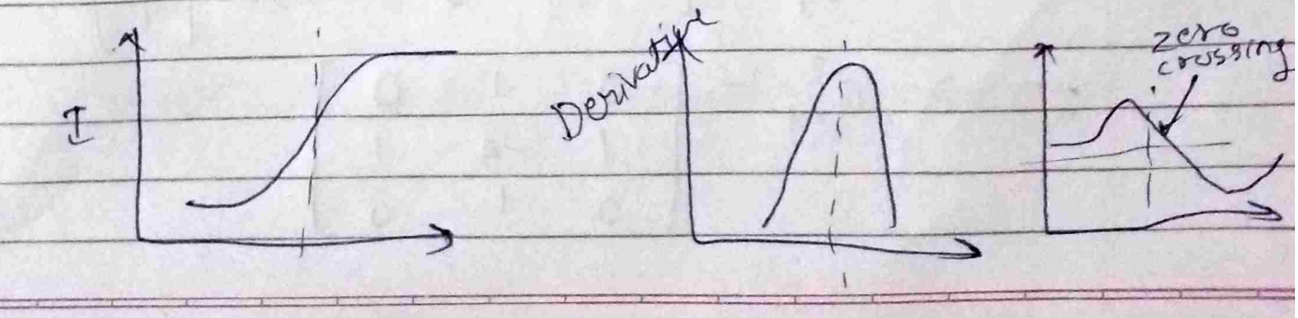
$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

The gradient points in the direction of most rapid change in intensity.

Edge strength $\Rightarrow |\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

Gradient direction $\Rightarrow \theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x} \right)$

$\frac{\partial f}{\partial x} = G_x \quad , \quad \frac{\partial f}{\partial y} = G_y$



PAGE No.
DATE / /

☆ Prewitt Operator:

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

☆ Robert Cross operator:

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

☆ Sobel Operator:-

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

Edge-Detection-Thresholding

→ performed to get the edge location.

Laplacian operator

∇^2 is laplacian operator:-

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

1D → $G_x^2 = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$ $\quad G_y^2 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$

2D → $G_{xy}^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

PAGE No.
DATE / /

~~14~~

→ Edge location is computed by locating the zero crossing

Input Image → Smoothing operation → Edge detection → Edge Map.

- Noise smooting is done using Gausian func^n, causes blurring or smearing of edge information or gradient values.

- Laplacian of Gausian func^n (LOG)
  - ↳ 2 operations:- → Smoothing
    - ↳ Applying Laplacian operator

- → These produced edges as zero crossings (ZC's) in output func^n.

Limitations :-
- ~~Ne~~ Does not give idea of
- → Neither, gradient magnitude or
- → orientation of the edges.

★ Canny Detection
- ↳ uses Gausian smoothing & derivative func^n together
- ↳ First derivative of Gausian func^n
- ↳ combines both the derivative and smoothing properties.
- → Hyteresis based thresholding strategy

PAGE No.
DATE / /

Canny Edge Detector :-

→ It is an optimal edge detector.

→ An optimal edge detector :-

1. Good detection
2. Good localisation.

~~Edge detection is done by :-~~

(1D) ⇒ $\bar{\nabla} g(x) = \frac{-x}{\sqrt{2\pi}\,\sigma^3} \exp\left(\frac{-x^2}{2\sigma^2}\right)$

LOG operator :-

$$\bar{\nabla}^2 g(x) = \frac{1}{\pi\sigma^4}\left(\frac{x^2}{2\sigma^2} - 1\right) \exp\left(\frac{-x^2}{2\sigma^2}\right)$$

Edge detection is done by :-

$\bar{\nabla}^2 G * f = 0$ &

$\bar{\nabla} G * f$ reaches a maximum

⇒ It detects :→ weak edges as well as strong edges.

It uses ~~∃~~ Two ~~two~~ threshold values to detect strong & weak edges

- Stages of ~~G~~ Canny algorithm :-

1. Noise Reduction → using gausian filter
2. Finding the intensity gradient of image
3. Non-maximum suppression.
4. Hysterisis thresholding

Algo :-

Original Img

↓

Smoothing by Gausian convolution

↓

Differential operator along x & y axis

↓

Non-maximum compression, finds peaks in the image gradient

↓

Hysterisis holding locates edge strings

↓

Edge map

Smooth

↓ ↓

h_x h_y

↓ ↓

Magnitude phase

↓

Non-Max. supression

↓

Threshold