Experiment 5: Edge Detection using Canny and Sobel Algorithms

Aim

To detect the edges of a given input image using the following edge detection algorithms:

- Sobel Edge Detection
- Canny Edge Detection

Software Required

- Python
- OpenCV
- NumPy
- Matplotlib

Theory

Edge detection identifies points in an image where intensity changes abruptly. These points correspond to object boundaries and are important for image analysis, segmentation, and feature extraction.

—

Sobel Edge Detection

Theory

Sobel edge detection uses first-order derivatives to compute image gradients. Two convolution masks are applied to detect horizontal and vertical edges.

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

Gradient magnitude is given by:

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

Algorithm

1. Read the grayscale image.
2. Apply Sobel operator in horizontal and vertical directions.
3. Compute gradient magnitude.
4. Display the edge detected image.

Pseudo Code (Python Style)

1. Read image into `img`
2. Compute horizontal gradient `Gx`
3. Compute vertical gradient `Gy`
4. Compute gradient magnitude
5. Display Sobel edge image

Example

Consider a 3×3 image region:

$$\begin{bmatrix} 10 & 10 & 10 \\ 10 & 50 & 50 \\ 10 & 50 & 50 \end{bmatrix}$$

Applying the Sobel operator produces high gradient values at the boundary between low and high intensity regions, indicating the presence of an edge.

—

Canny Edge Detection

Theory

Canny edge detection is a multi-stage algorithm designed to produce thin and continuous edges. It reduces noise and minimizes false edge detection.

The steps involved are:

- Gaussian smoothing
- Gradient calculation
- Non-maximum suppression
- Double thresholding and hysteresis

Algorithm

1. Read the grayscale image.
2. Apply Gaussian filter to reduce noise.
3. Compute gradient magnitude and direction.
4. Perform non-maximum suppression.
5. Apply double thresholding.
6. Track edges using hysteresis.

Pseudo Code (Python Style)

1. Read image into `img`
2. Apply Gaussian blur
3. Compute gradient magnitude and direction
4. Perform non-maximum suppression
5. Apply double threshold
6. Track edges using hysteresis
7. Display Canny edge image

Example

Consider an image with a sharp boundary between dark and bright regions. After Gaussian smoothing and gradient calculation, Canny edge detection identifies strong edges and suppresses weak responses, resulting in thin and continuous edge boundaries.

—

Result

Thus, edges of the given input image were successfully detected using Sobel and Canny edge detection algorithms.