LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.07**

**A.1 Aim:**

Write a program to detect edges in the image using Robert, Prewitt and Sobel operators.

**A.2 Prerequisite:**

1 Python programming syntax (Refer the documentation of library).

2. Knowledge of fundamentals of image segmentation and edge detection.

2. Availability of Soft copy of solo photograph, medical image, and Natural Image.

**A.3 Outcome:**

**After successful completion of this experiment students will be able to**

1. Apply Robert, Prewitt and Sobel operators on given image.
2. Differentiate the outputs of different edge detection operators.
3. Identify applications of operators studied.

**A.4 Theory:**

**A.4.1. Edge Detection**

Edge detection is one of the most frequently used techniques in digital image processing. The boundaries of object surfaces in a scene often lead to oriented localized changes in intensity of an image, called edges. This observation combined with a commonly held belief that edge detection is the first step in image segmentation, has fueled a long search for a good edge detection algorithm to use in Image processing. Edge is nothing but a boundary between two regions having distinct intensity levels. The goal of edge detection is to select the pixels in a digital image at which the intensity level changes sharply. For image processing system to interpret an image, it must be able to detect the edges of each object in the image. Edge representation drastically reduces the amount of data to be processed by retaining the important information in an image such as the shape of objects. This description of an image is easy to integrate into a large number of object recognition algorithms. Edge detection generates an edge map that contains vital information of the image.

Image segmentation is an essential step in image analysis. The objective of segmentation is to simplify and/or change the representation of an image in to something that is more meaningful and easier to analyze. It divides (segments) an image into its constituent regions or objects. Generally, it is used to locate objects and boundaries in image. Image Segmentation is used when we need to automate a particular activity. Image segmentation methods are categorized on the basis of two properties discontinuity and similarity. The choice of image segmentation technique is depends on the nature of the problem under consideration. Edge detection is a part of image segmentation. The effectiveness of image segmentation depends on the perfection of detecting meaningful edges.

**EDGE DETECTION TECHNIQUES**

Edge detection techniques try to locate points with abrupt changes in an image. Edge is nothing but boundary between two regions having distinct intensity levels.

1. Robert Edge Detection

It is very simple computation technique, introduced by Lawrence Roberts. Here high frequency spatial frequency region is corresponds to an edge. 2-D mask for Robert edge detection is as shown in Fig.1.

Fig.1. Roberts Edge Operator

In this technique the output represents pixels of every point which estimated complete magnitude of spatial gradient of the image at that point.

2. Sobel Edge Detection

The Sobel edge detection method is introduced by Sobel in 1970. This method of edge detection for image segmentation finds edges using the Sobel approximation to the derivative. The Sobel masks are as shown in Fig. 2. The first mask is responsible for computing horizontal edges and the other one is responsible for computing vertical edges. One mask is simply the other rotated by 90o.

Fig.2. Sobel Operator

3. Prewitt Edge Detection

This edge detection technique was introduced by J.M.S. Prewitt in 1970. The Prewitt operator assigns similar weights to all the neighbors of the candidate pixel whose edge strength is being calculated. The Prewitt operator is as shown in Fig. 3.

Fig.3. Prewitt Operator

Similar to Sobel operator, The first mask is responsible for computing horizontal edges and the other one is responsible for computing vertical edges and one mask is simply the other rotated by 90o.

**A.5 Procedure/Algorithm:**

**A.5.1:**

**TASK 1:**

1. Read the i/p image

2. Apply Roberts, Sobel and Prewitt operator to the image as per following and

obtain the 3 outputs separately.

1. X gradient
2. Y gradient
3. Combined of both X and Y gradient.

3. Display the original and the output image.

4. Observe the output and complete PART B of lab manual.

5. Save and close the file and name it as **EX7\_Task1\_your Roll no.doc**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)***

|  |  |
| --- | --- |
| Roll No.: C026 | Name: Anirbaan Ghatak |
| Class : B | Batch : B1 |
| Date of Experiment: 27/09/2023 | Date of Submission 28/09/2023 |
| Grade : | Time of Submission: |
| Date of Grading: |  |

**B.1 Software Code written by student:**

# Name: Anirbaan Ghatak

# Roll no.: C026

# Aim: Write a program to detect edges in the image using Robert, Prewitt and Sobel operators.

import cv2

import numpy as np

import matplotlib.pyplot as plt

# Load the image

img = cv2.resize(cv2.imread('IMG\_2458\_grey\_CLOSEUP.jpg',

                 cv2.IMREAD\_GRAYSCALE), (0, 0), fx=0.5, fy=0.5)

nat = cv2.resize(cv2.imread('nature.jpg', cv2.IMREAD\_GRAYSCALE),

                 (0, 0), fx=0.5, fy=0.5)

medical = cv2.resize(cv2.imread(

    'medical.jpg', cv2.IMREAD\_GRAYSCALE), (0, 0), fx=0.5, fy=0.5)

def edges(img):

    # Apply the Roberts operator

    edges\_roberts = cv2.Sobel(

        img, cv2.CV\_8U, 1, 0, ksize=3) + cv2.Sobel(img, cv2.CV\_8U, 0, 1, ksize=3)

    # Apply the Sobel operator

    edges\_sobel = cv2.Sobel(img, cv2.CV\_8U, 1, 0, ksize=3) + \

        cv2.Sobel(img, cv2.CV\_8U, 0, 1, ksize=3)

    # Apply the Prewitt operator

    kernelx = np.array([[1, 1, 1], [0, 0, 0], [-1, -1, -1]])

    kernely = np.array([[-1, 0, 1], [-1, 0, 1], [-1, 0, 1]])

    edges\_prewitt = cv2.filter2D(

        img, -1, kernelx) + cv2.filter2D(img, -1, kernely)

    plt.figure(figsize=(10, 8))

    # Plot the first image in the top left position

    plt.subplot(2, 2, 1)

    plt.imshow(img, cmap='gray')

    plt.title('Original')

    plt.axis('off')

    # Plot the second image in the top right position

    plt.subplot(2, 2, 2)

    plt.imshow(edges\_roberts, cmap='gray')

    plt.title('Roberts Edges')

    plt.axis('off')

    # Plot the third image in the bottom left position

    plt.subplot(2, 2, 3)

    plt.imshow(edges\_sobel, cmap='gray')

    plt.title('Soble Edges')

    plt.axis('off')

    # Plot the fourth image in the bottom right position

    plt.subplot(2, 2, 4)

    plt.imshow(edges\_prewitt, cmap='gray')

    plt.title('Prewitt Edges')

    plt.axis('off')

    plt.tight\_layout()

    plt.show()

edges(img)

edges(nat)

edges(medical)

**B.2 Input and Output:**

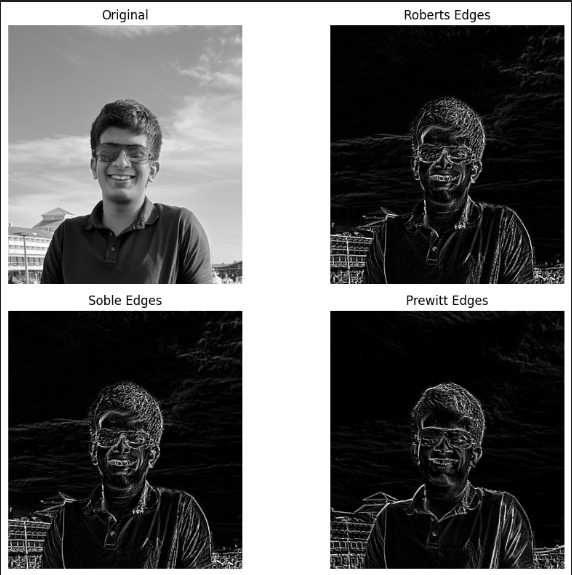
***(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)***

**Input Images:**

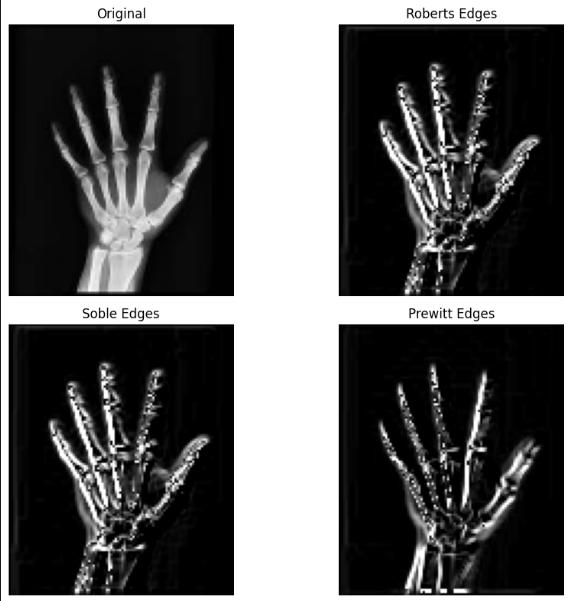
****

**Output Images:**

1. **For each edge detection operatoras per the procedure discussed in section A.5.**

****

****

****

**B.3 Observations and learning:**

***Applied Robert Prewitt and Sobel operators. Robert operator a basic edge detection method using simple 2x2 convolution kernels to approx. gradients by computing diagonal differences, Sobel operator also an edge detection technique using 3x3 convolution kernels to calculate image gradients in both horizontal and vertical directions, providing a more accurate gradient estimation and Prewitt operator another edge detection method using 3x3 convolution kernels to compute image gradients, but with a slightly different weight distribution. Observed the processed outputs to see the difference in different algorithms***

**B.4 Conclusion:**

***Performed Robert, Prewitt and Sobel operators on the selected image, medical and nature image, and found the edges of the selected images and observed the differences between the images which used the different algorithms***

**B.5 Question of Curiosity**

**1. Compare the performance of Sobel & Prewitt operators.**

Both the Sobel and Prewitt operators are commonly used for edge detection in image processing. Here's a short comparison of their performance:

Sobel Operator:

Pros: Provides slightly better edge detection results due to the specific weight distribution in its convolution kernels.

Cons: Can be computationally more intensive due to the 3x3 convolution kernels.

Prewitt Operator:

Pros: Easier to compute compared to Sobel due to the simpler convolution kernels with a more uniform weight distribution.

Cons: May not provide as sharp edge detection results as Sobel, especially for certain types of edges.

**2. What result you will get if all these (Roberts, Sobel and Prewitt) operators are applied to binary image?**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*