LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.05**

**A.1 Aim:**

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

**A.2 Prerequisite:**

1 Matlab programming syntax (Refer the Matlab manual).

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

2. Availability of Soft copy of medical 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 edge detection 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.

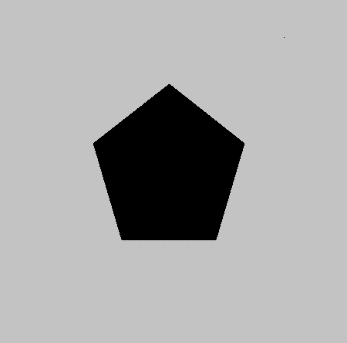
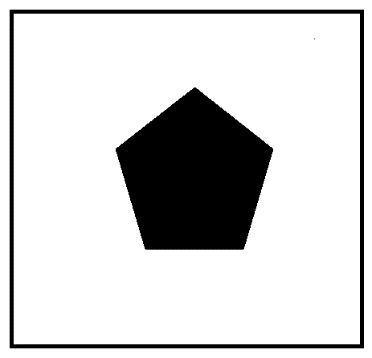
3. Display the original and the output image.

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

5. Save and close the file and name it as **EX8\_Task1\_your Roll no.m**

**TASK 2:**

1. **Apply Global thresholding and local thresholding approaches to segment the following input images.**

** **

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

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: B032 | Name: Naman Garg |
| Class :BTech CS B | Batch : B2 |
| Date of Experiment: 8-8-20 | Date of Submission |
| Grade : | Time of Submission: |
| Date of Grading: |  |

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

***(Paste your Matlab code completed during the 2 hours of practical in the lab here)***

a=imread('gg.jpg');

img1 = imresize(rgb2gray(a),[256,256]);

b=imread('rr.jpg');

img2 = imresize(rgb2gray(b),[256,256]);

%robert

[imgX,imgY, img] = robert(double(img1));

subplot(2,4,1),imshow(img1);

title('input image');

subplot(2,4,2),imshow(uint8(imgX));

title('X gradient');

subplot(2,4,3),imshow(uint8(imgY));

title('Y gradient');

subplot(2,4,4),imshow(uint8(img));

title('X & Y combined');

[imgmX,imgmY, imgm] = robert(double(img2));

subplot(2,4,5),imshow(img2);

title('Input image');

subplot(2,4,6),imshow(uint8(imgmX));

title('X gradient');

subplot(2,4,7),imshow(uint8(imgmY));

title('Y gradient');

subplot(2,4,8),imshow(uint8(imgm));

title('X & Y combined');

sgtitle('Edge Detection: Robert Operator');

function [imgRx, imgRy, imgR] = robert(img)

imgR = double(img);

robx=[-1 0; 0 1];

roby=[0 -1; 1 0];

for i=1:255

for j=1:255

imgtemp=img(i:i+1,j:j+1);

x = sum(sum(robx.\*imgtemp));

y = sum(sum(roby.\*imgtemp));

pixvalue = sqrt(x^2 + y^2);

imgRx(i,j) = x;

imgRy(i,j) = y;

imgR(i,j) = pixvalue;

end

end

end

%sobel

[imgX,imgY, img] = sobel(double(img1));

subplot(2,4,1),imshow(img1);

title('input image');

subplot(2,4,2),imshow(uint8(imgX));

title('X gradient');

subplot(2,4,3),imshow(uint8(imgY));

title('Y gradient');

subplot(2,4,4),imshow(uint8(img));

title('X & Y combined');

[imgmX,imgmY, imgm] = sobel(double(img2));

subplot(2,4,5),imshow(img2);

title('Input image');

subplot(2,4,6),imshow(uint8(imgmX));

title('X gradient');

subplot(2,4,7),imshow(uint8(imgmY));

title('Y gradient');

subplot(2,4,8),imshow(uint8(imgm));

title('X & Y combined');

sgtitle('Edge Detection: Sobel Operator');

function [imgSx, imgSy, imgS] = sobel(img)

imgS=double(img);

sobx=[-1 -2 -1; 0 0 0; 1 2 1];

soby=[-1 0 1; -2 0 2; -1 0 1];

for i=2:255

for j=2:255

imgtemp = img(i-1:i+1,j-1:j+1);

x = sum(sum(sobx.\*imgtemp));

y = sum(sum(soby.\*imgtemp));

pixvalue = sqrt(x^2 + y^2);

imgSx(i,j) = x;

imgSy(i,j) = y;

imgS(i,j) = pixvalue;

end

end

end

%prewitt

[imgX,imgY, img] = prewitt(double(img1));

subplot(2,4,1),imshow(img1);

title('input image');

subplot(2,4,2),imshow(uint8(imgX));

title('X gradient');

subplot(2,4,3),imshow(uint8(imgY));

title('Y gradient');

subplot(2,4,4),imshow(uint8(img));

title('X & Y combined');

[imgmX,imgmY, imgm] = prewitt(double(img2));

subplot(2,4,5),imshow(img2);

title('Input image');

subplot(2,4,6),imshow(uint8(imgmX));

title('X gradient');

subplot(2,4,7),imshow(uint8(imgmY));

title('Y gradient');

subplot(2,4,8),imshow(uint8(imgm));

title('X & Y combined');

sgtitle('Edge Detection: Prewitt Operator');

function [imgPx, imgPy, imgP] = prewitt(img)

imgS=double(img);

prewx=[-1 -1 -1; 0 0 0; 1 1 1];

prewy=[-1 0 1; -1 0 1; -1 0 1];

for i=2:255

for j=2:255

imgtemp = img(i-1:i+1,j-1:j+1);

x = sum(sum(prewx.\*imgtemp));

y = sum(sum(prewy.\*imgtemp));

pixvalue = sqrt(x^2+ y^2);

imgPx(i,j) = x;

imgPy(i,j) = y;

imgP(i,j) = pixvalue;

end

end

end

**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.)***

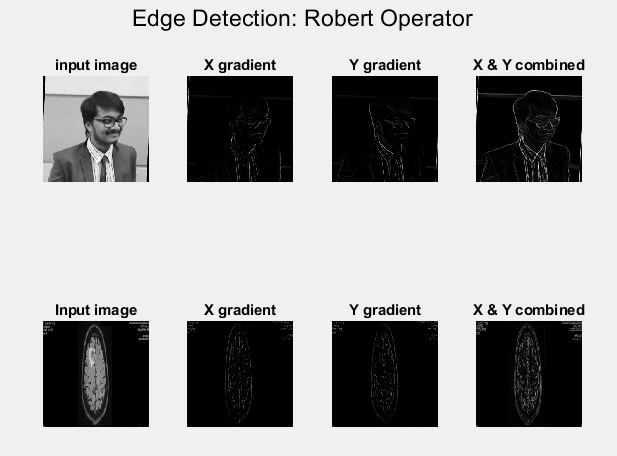
**Input Images:**

1. **Your own Photo**
2. **Medical image [MRI of Brain Tumor]**

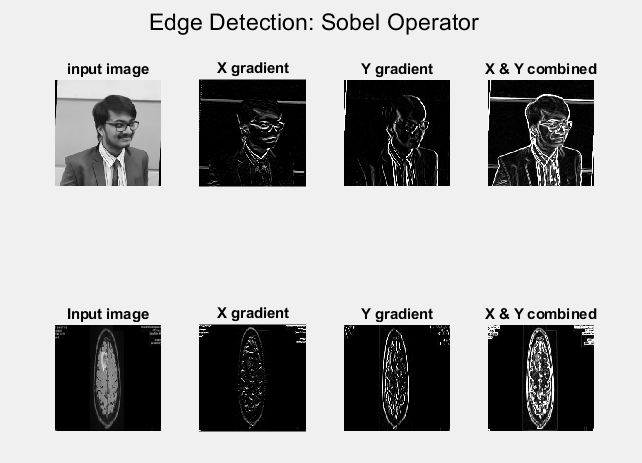
****

**Output Images:**

1. **For each edge detection operators per the procedure discussed in section A.5.**
2. **Robert:**

****

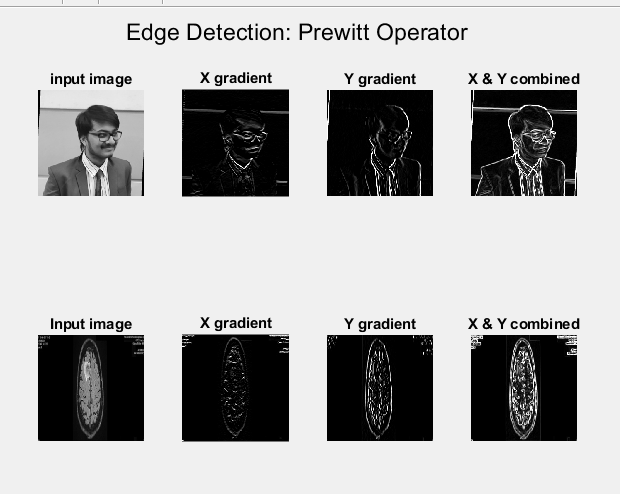
1. **Sobel**

****

1. **Prewitt**

**I. Your Photo [3 outputs : X, Y, Combined XY]**

**II. Medical Image [3 outputs: X, Y, Combined XY]**

****

**B.3 Observations and learning:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Image type** | **Gradient used** | **Impact on image (Observations)\* For more points you may use separate space below the table.** |
| **Robert** | Self Photo | X | the horizontal edges are sharpened and are more visible. |
| Y | the vertical edges are sharpened and are more visible. |
| Combined | all the edges present in the image get sharpened and become visible. |
| Medical Image | X | the horizontal edges are sharpened and are more visible. |
| Y | the vertical edges are sharpened and are more visible. |
| Combined | all the edges present in the image get sharpened and become visible. |
| **Sobel** | Self Photo | X | helps in better highlighting the horizontal edges, helps manage the noise present. |
| Y | helps in better highlighting the vertical edges, helps manage the noise present. |
| Combined | this helps in better highlighting all the edges that are present and also helps manage the noise present. |
| Medical Image | X | helps in better highlighting the horizontal edges, helps manage the noise present. |
| Y | helps in better highlighting the vertical edges, helps manage the noise present. |
| Combined | this helps in better highlighting all the edges that are present and also helps manage the noise present. |
| **Prewitt** | Self Photo | X | Upon applying the Prewitt operator, the noise in the background is blurred out and the horizontal edges become visible |
| Y | Upon applying the Prewitt operator, the noise in the background is blurred out and the vertical edges become visible |
| Combined | Upon applying the Prewitt operator, the noise in the background is blurred out and all the edges are sharply visible. |
| Medical Image | X | Upon applying the Prewitt operator, the noise in the background is blurred out and the horizontal edges become visible |
| Y | Upon applying the Prewitt operator, the noise in the background is blurred out and the vertical edges become visible |
| Combined | Upon applying the Prewitt operator, the noise in the background is blurred out and all the edges are sharply visible. |

***(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)***

X gradient works as a horizontal line filter and Y gradient works as a vertical line filter

Robert operator is the basic operator out of the three: only 4 pixels are examined in order to determine the value of the output pixel. However since it uses such a small kernel, it is more creates more noise and only when the edges are sharp is it able to detect as can be seen from the output images.

The Sobel and Prewitt operator are better as compared to the Robert operator in this aspect as they have larger kernels which end up smoothing the input and eliminating a lot of the noise. In addition to this, they are better at detecting and displaying the edges. Both their kernels are similar to each other but since the Sobel operator has more weights assigned, it is able to make more edges visible as compared to the Prewitt operator.

All of these operators however do not give continuous edges as can be seen in the medical images very clearly. Sobel and Prewitt do a better job than the Robert Operator at this and give a clearer depiction of the tumor but at the same time, the edges have different thickness and certain edges are a lot thicker as compared to the other.

**B.4 Conclusion:**

Upon completion of this experiment, I was able to apply the Robert, Prewitt and Sobel operators on given images and was able to differentiate between the outputs of the different edge detection operators. Furthermore, I was able to identify how these operators can be used for image segmentation and how by applying a proper threshold we may even be able to separate the background and all other objects.

**B.5 Question of Curiosity**

Q1. How segmentation of the image is achieved using Edge detectors?

Segmentation subdivides the image into its constituent regions or objects. Every object/ region is associated with a boundary which is characterized by an edge. And edge generally follows the path of a rapid change in image intensity. Thus edge detection can be used to characterize the intensity changes in terms of the regions that cause them and can hence be used to ‘segment’ the image or divide the regions from the background.

Q2. Explain Waterfall model of image segmentation?

The Watershed algorithm usually leads to a strong over-segmentation of an image. The Waterfall is a hierarchical approach that selects among all the contours of the Watershed those that are completely surrounded by more contrasted contours. By removing these contours, a simplified partition is obtained. The process may be iterated. At the end, a single region covering the whole image is obtained.

Q3. How does canny edge detector affect the input image [your photo and Medical image chosen] Show the output and comment on it.

a=imread('gg.jpg');

img1 = imresize(rgb2gray(a),[256,256]);

b=imread('rr.jpg');

img2 = imresize(rgb2gray(b),[256,256]);

c1=edge(img1,'canny');

c2=edge(img2,'canny');

subplot(2,2,1),imshow(img1);

title('Original image');

subplot(2,2,2),imshow(img2);

title('Medical image');

subplot(2,2,3),imshow(c1);

title('Output for own image')

subplot(2,2,4),imshow(c2);

title('Output for medical image');

sgtitle('Edge Detection: Canny Operator');

Canny edge operator is a more optimal edge detector since it has a multi stage process: it first smooths the image by gaussian convolution,

it calculates the gradient like the other operators.

Sobel and Prewitt end up giving thicker edges and this is eliminated by the canny operator and thins out the edges and then applies double thresholding to make sure all the edges have the same gray level and can easily be differentiated from the background.

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