

Matlab Code for Edge Detection Robert, Prewitt, Sobel

```
%%%%% EDGE Detection

clc;
close all;
clear all;
a=imread('C:\Documents and Settings\student\Desktop\lenna1.bmp');

b=im2double(a);
[m,n]=size(a);

%ROBERT
L(1:m,1:n)=0
for i=1:m-2;
    for j=1:m-2;
        L(i,j)=-1*b(i,j)+0+0+1*b(i+1,j+1);
    end;
end;

M(1:m,1:n)=0
for i=1:m-2;
    for j=1:m-2;
        M(i,j)=0-1*b(i,j+1)+1*b(i+1,j)+0;
    end;
end;

figure;
subplot(2,2,1)
imshow(L)
title('Robert Gx');
subplot(2,2,2)
imshow(M)
title('Robert Gy');
N=M+L;
subplot(2,2,3)
imshow(N)
title('Robert Gx+Gy');
subplot(2,2,4)
imshow(b)
```

```
title('Original Image');
```

```
%PREWIT
```

```
N(1:m,1:n)=0
```

```
for i=1:m-2;
```

```
    for j=1:m-2;
```

```
        N(i,j)=-1*b(i,j)-1*b(i,j+1)-1*b(i,j+2)+0+0+0+1*b(i+2,j)+1*b(i+2,j+1)+1*b(i+2,j+2);
```

```
    end;
```

```
end;
```

```
O(1:m,1:n)=0
```

```
for i=1:m-2;
```

```
    for j=1:m-2;
```

```
        O(i,j)=-1*b(i,j)+0+1*b(i,j+2)-1*b(i+2,j)+0+1*b(i+1,j+2)-1*b(i+2,j)+0+1*b(i+2,j+2);
```

```
    end;
```

```
end;
```

```
figure;
```

```
subplot(2,2,1)
```

```
imshow(N)
```

```
title('Prewit Gx');
```

```
subplot(2,2,2)
```

```
imshow(O)
```

```
title('Prewit Gy');
```

```
Z=N+O;
```

```
subplot(2,2,3)
```

```
imshow(Z)
```

```
title('Prewit Gx+Gy');
```

```
subplot(2,2,4)
```

```
imshow(b)
```

```
title('Original Image');
```

```
%SOBEL
```

```
P(1:m,1:n)=0
```

```
for i=1:m-2;
```

```
    for j=1:m-2;
```

```
        P(i,j)=-1*b(i,j)-2*b(i,j+1)-1*b(i,j+2)+0+0+0+1*b(i+2,j)+2*b(i+2,j+1)+1*b(i+2,j+2);
```

```
    end;
```

```
end;
```

```
R(1:m,1:n)=0
```

```
for i=1:m-2;
```

```
    for j=1:m-2;
```

```
        R(i,j)=-1*b(i,j)+0+1*b(i,j+2)-2*b(i+1,j)+0+2*b(i+1,j+2)-1*b(i+2,j)+0+1*b(i+2,j+2);
```

```

    end;
end;

figure;
subplot(2,2,1)
imshow(P)
title('Sobel Gx');
subplot(2,2,2)
imshow(R)
title('Sobel Gy');

Y=P+R;
subplot(2,2,3)
imshow(Y)
title('Soble Gx+Gy');
subplot(2,2,4)
imshow(b)

title('Original Image');

```

Edge Detection with sobel operator



Edge Detection with sobel operator

shahnewaz ali

the sobel operator is widely used to detect edges of image , in the area of image and [video processing](#) ,[computer vision](#), robotics etc . here I am going to give a brief description of sobel operator and matlab implementation ([source code](#)) both using **matlab** built-in function for edge detection and manually sobel edge detection (function).

Basics :

the sobel operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction.

Mathematically, the gradient of a two-variable function (here the image intensity function) is at each [image point](#) a 2D vector with the components given by the derivatives in the horizontal and vertical directions.

sobel operator :

-1	0	1
-2	0	2

-1	0	1
----	---	---

Sx
and

1	2	1
0	0	0
-1	-2	-1

Sy

Convolution :

The *convolution operation* is a mathematical operation which takes two functions $f(x)$ and $g(x)$ and produces a third function $h(x)$. In image processing 2D convolution is used .

no more theory or mathematical definition if you want go to the following link :

more about theory : http://en.wikipedia.org/wiki/Sobel_operator

Source code :

I am going to detect the edges of image (flower) that is placed on the right side of this post , the threshold is determined empirically.

type 1 : matlab built in function :-

```
im=imread('flower.jpg');
img=rgb2gray(im);

sob_im = edge(img,'sobel');
figure(2);
imagesc(sob_im);
axis('square');
colormap('gray');

imshow(sob_im);
```

type 2:

edge_sobel is a M-file that contain the function :

the function describe below . in matlab file ->new->M file . a [new text editor](#) will be open then copy the following code and paste it , save the [file name](#) as edge_sobel.m

```
function [y]=edge_sobel(imT,i,j)

% sobel kernel
kx=[-1,-2,-1;0,0,0;1,2,1];
ky=[-1,0,1;-2,0,2;-1,0,1];
sx=0;
sy=0;
frac=0;
for xx=-1:1
for yy=-1:1
frac=double (imT(i+xx , j+yy))* kx(xx+2,yy+2);
```

```

%convolution
sx = double(sx+frac);

%kx(xx+2,yy+2)
%frac
%sx
end;
end;
frac=0;
for xx=-1:1
for yy=-1:1
frac=double(imT(i+xx , j+yy))* ky(xx+2,yy+2);
%ky(xx+2,yy+2)
sy=double(sy+frac);
%frac
%sy
end;
end;
%sx
%sy
z=uint8(sqrt(sx^2+sy^2))
%here threshold i defined threshold empirically . you can choose any method
%adaptive thresholding
if(z>=100)
y=255;
else
y=0;
end;
y;

```

then copy the following code and paste it in command window and enjoy it

```

im=imread('flower.jpg');
img=rgb2gray(im);
im3=img ;
[ht wd ch] = size(img);
im_out = imresize(img, [432 432], 'bilinear');
for i=2:431
for j=2:431
im3(i,j)=edge_sobel(im_out,i,j);%edge_sobel.m file
end;
end;
imshow(im3);

```

hope this post will help you to start learning image processing methodology