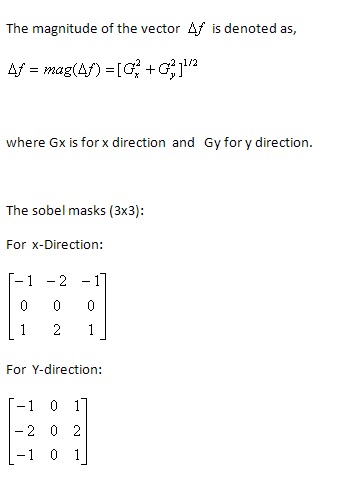
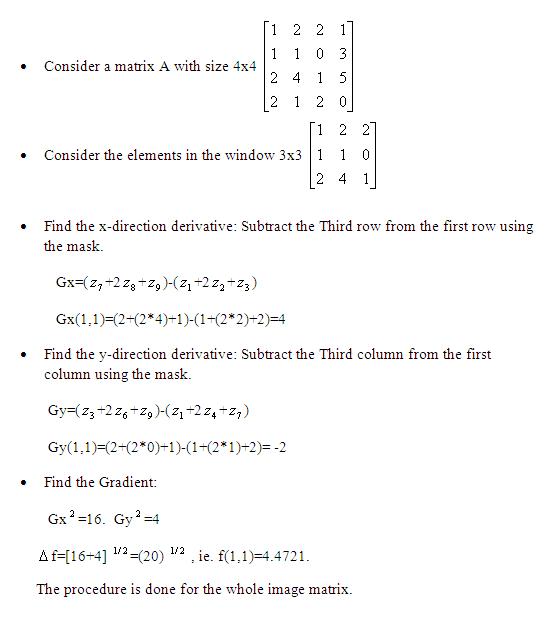
**Sobel edge detection**

         The gradient of the image is calculated for each pixel position in the image.

[](http://2.bp.blogspot.com/-y9KGeTKBP5Q/TvHC2WaqotI/AAAAAAAAAio/UGtwxQm07Is/s1600/sobel_formula.bmp)

**The procedure and the MATLAB  code for sobel edge detection without using MATLAB built-in function**[:](http://www.google.com/)

[](http://1.bp.blogspot.com/-U8zLYuVuRqk/TvHCalWjWqI/AAAAAAAAAiA/t1FfN4dbiew/s1600/sobel_procedure.JPG)

**MATLAB CODE:**  
  
  
A=imread('peppers.png');  
B=rgb2gray(A);  
  
C=double(B);  
  
  
for i=1:size(C,1)-2  
    for j=1:size(C,2)-2  
        %Sobel mask for x-direction:  
        Gx=((2\*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-(2\*C(i,j+1)+C(i,j)+C(i,j+2)));  
        %Sobel mask for y-direction:  
        Gy=((2\*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-(2\*C(i+1,j)+C(i,j)+C(i+2,j)));  
        
        %The gradient of the image  
        %B(i,j)=abs(Gx)+abs(Gy);  
        B(i,j)=sqrt(Gx.^2+Gy.^2);  
        
    end  
end  
figure,imshow(B); title('Sobel gradient');

|  |
| --- |
| <http://3.bp.blogspot.com/-7qgVYfKDFAY/TvHCYLh4mdI/AAAAAAAAAh4/9fu_caSLh0U/s1600/sobel_gradient.jpg> |
| Sobel Gradient |

%Define a threshold value  
Thresh=100;  
B=max(B,Thresh);  
B(B==round(Thresh))=0;  
  
B=uint8(B);  
figure,imshow(~B);title('Edge detected Image');

|  |
| --- |
| <http://3.bp.blogspot.com/-bzcduukEnjU/TvHCcmoNY5I/AAAAAAAAAiI/OQTOxYNNkus/s1600/sobel_thresh100.jpg> |
| Edge detected Image |

|  |
| --- |
| [http://3.bp.blogspot.com/-O7n4DTX-64w/TvHCeqkRUZI/AAAAAAAAAiQ/eb9vGpbRyK4/s400/sobel_thresh35.jpg](http://3.bp.blogspot.com/-O7n4DTX-64w/TvHCeqkRUZI/AAAAAAAAAiQ/eb9vGpbRyK4/s1600/sobel_thresh35.jpg) |
| Edge detected Image(Threshold value:35) |

The edge detected image can be obtained from the sobel gradient by  
using a threshold value.

* If the sobel gradient values are lesser than the threshold value then replace it with the threshold value.  
  if f < threshold value then  
  f = threshold value.

To avoid complex computation, the gradient can also be computed using the formula:

[http://3.bp.blogspot.com/-R5bgP8xBUxo/TvHCUHMVfTI/AAAAAAAAAhw/dz7l8RuvJ64/s1600/sobel_formula2.bmp](http://3.bp.blogspot.com/-R5bgP8xBUxo/TvHCUHMVfTI/AAAAAAAAAhw/dz7l8RuvJ64/s1600/sobel_formula2.bmp)

The Image obtained from computing X-direction derivative:

[](http://3.bp.blogspot.com/-sFVr_jc5DTA/TvHCgNDHGvI/AAAAAAAAAiY/_rhUju88PWs/s1600/sobel_xaxis.jpg)

The Image obtained from computing Y-direction derivative:

|  |
| --- |
| <http://2.bp.blogspot.com/-XHsQceU8J7A/TvHChUmRXVI/AAAAAAAAAig/5-ERyP-lBU8/s1600/sobel_yaxis.jpg> |
| **Check  [Sobel Edge Detection - Part 2](http://angeljohnsy.blogspot.com/2013/07/sobel-edge-detection-part-2.html)** |

### Sobel Edge Detection - Part 2

In [Edge Detection- fundamentals](http://angeljohnsy.blogspot.com/2013/07/edge-detection-fundamentals.html), we have seen how the first and second order derivatives are used in finding the edge strength. Now lets see another version of sobel edge detection.

**Prerequisite: [S](http://angeljohnsy.blogspot.com/2011/12/sobel-edge-detection.html)**[obel Edge Detection-Part 1](http://angeljohnsy.blogspot.com/2011/12/sobel-edge-detection.html)

#### Basic Steps followed in Sobel Edge Detection:

1.       Obtain the gradient of the image.

2.       Find the magnitude

3.       Threshold the gradient image.

#### SOBEL EDGE DETECTION USING ‘edge’ FUNCTION:

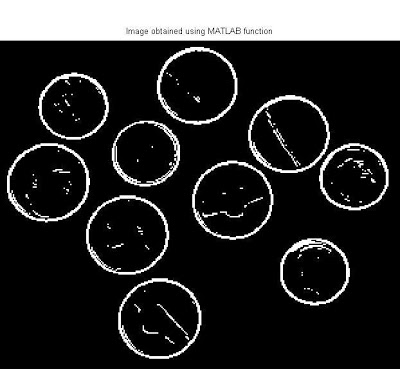
%Input Image

A=imread('coins.png');

%Image obtained using MATLAB function 'edge'

[E,th]=edge(A,'sobel','nothinning');

figure,imshow(E);title('Image obtained using MATLAB function')

[](http://4.bp.blogspot.com/-xWBWLC3pQoY/UfgA8fGMvdI/AAAAAAAAA9M/GWGmTgnz5ns/s1600/edge_sobel_01.jpg)

**Edge Detection without using the 'edge' function:**

#### MATLAB CODE:

%Input Image

A=imread('coins.png');

%Preallocate the matrices with zeros

I=zeros(size(A));

%Filter Masks

F1=[-1 0 1;-2 0 2; -1 0 1];

F2=[-1 -2 -1;0 0 0; 1 2 1];

A=double(A);

for i=1:size(A,1)-2

    for j=1:size(A,2)-2

        %Gradient operations

        Gx=sum(sum(F1.\*A(i:i+2,j:j+2)));

        Gy=sum(sum(F2.\*A(i:i+2,j:j+2)));

        %Magnitude of vector

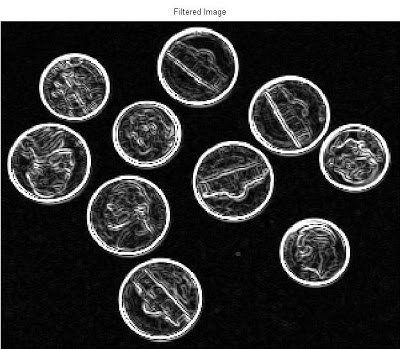
         I(i+1,j+1)=sqrt(Gx.^2+Gy.^2);

    end

end

I=uint8(I);

figure,imshow(I);title('Filtered Image');

[](http://1.bp.blogspot.com/-DYBnE3vjE-4/UfgBBKRVKhI/AAAAAAAAA9c/rXnTqarnnZE/s1600/edge_sobel_03.jpg)

%Define a threshold value

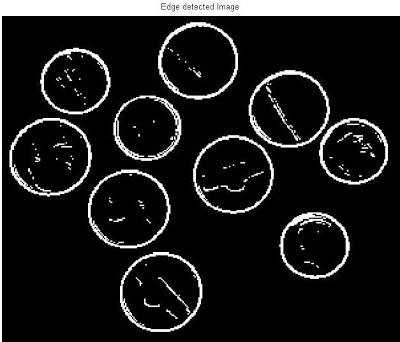
Thresh=210;

B=max(I,Thresh);

B(B==round(Thresh))=0;

B=im2bw(B);

figure,imshow(B);title('Edge detected Image');

[](http://1.bp.blogspot.com/-PalyMWv6yY4/UfgA_aeXDsI/AAAAAAAAA9Y/DXcuWoCKGWg/s1600/edge_sobel_02.jpg)

#### EXPLANATION:

1.       Read the image

2.       Convert the image to double

3.       Use the mask F1 for x direction and F2 for y direction and obtain the gradient of the image.

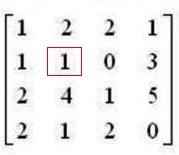
4.       Find the magnitude of the vector.

5.       Since we need 3x3 image pixels, the border pixels are not considered, and so starting from the pixel (2, 2) the edge detection process starts.  
       %Magnitude of vector

    I(i+1,j+1)=sqrt(Gx.^2+Gy.^2);

    When i=1 and j =1, then Image I pixel position will be I(2,2).Thus we are not considering the borders.

Example:

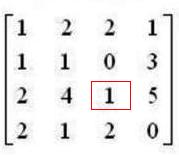
[](http://2.bp.blogspot.com/-lgXRhEDGjNY/UfgJhQzQHVI/AAAAAAAAA9s/jot-anO61-0/s1600/matrixA_01.JPG)

    In the for loop 2 is subtracted.

"

for i=1:size(A,1)-2

    for j=1:size(A,2)-2 "  
     The filter mask is 3x3, so the last position to be processed in our example is I(3,3).And normally it will be I(size(A,1)-2,size(A,2)-2). Thus the borders are left.  
  
Example :

[](http://2.bp.blogspot.com/-jZy63RBUkAg/UfgKfcgV_dI/AAAAAAAAA94/t1OcVE6fRCA/s1600/matrixA.jpg)

6.       Threshold the image

7.       Display the logical image

#### Advantage:

1. Sobel masks perform better noise suppression.

2. Image smoothing

#### Disadvantage:

1.       Diagonal direction points are not preserved always.