Detect Edges in Images

This example shows how to detect edges in an image using both the Canny edge detector and the Sobel edge detector.

Table of Contents

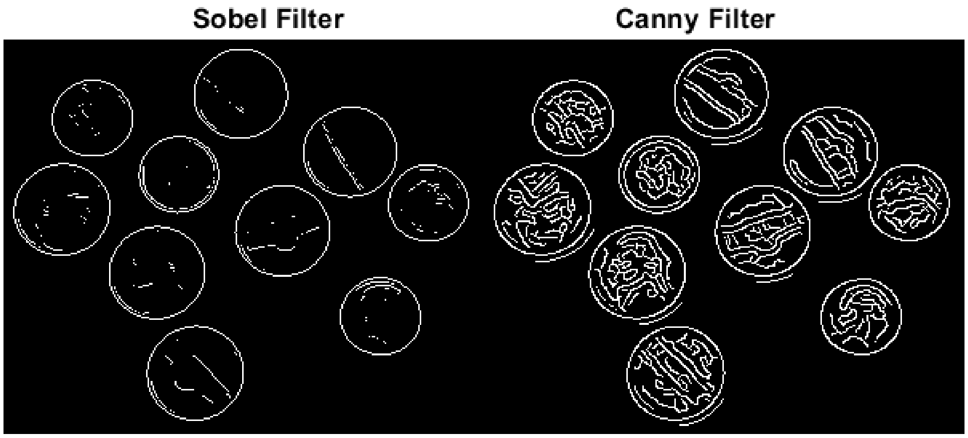
Read Image

```
I = imread('coins.png');
imshow(I)
```

Detect edges using two methods

Apply both the Sobel and Canny edge detectors to the image and display them for comparison.

```
BW1 = edge(I,'sobel');
BW2 = edge(I,'canny');
figure;
imshowpair(BW1,BW2,'montage')
title('Sobel Filter                                   Canny Filter');
```



How does this work?

There are many algorithms (such as Sobel and Canny), but the basic idea is quite simple.

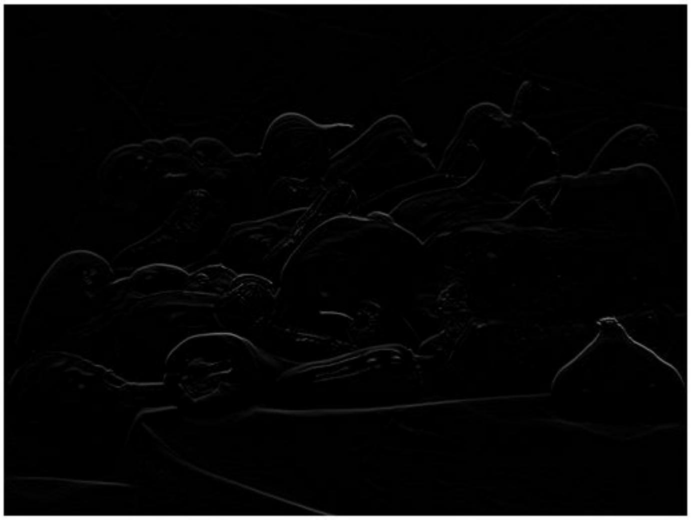
```
I = imread('peppers.png');
I(1:15, 1:15, 1)
```

```
ans = 15×15 uint8 matrix
   62   63   63   65   66   63   61   63   63   64   64   61   65   64   61
   63   61   59   64   63   60   61   64   64   63   63   65   65   64   63
   65   63   63   66   66   62   60   66   64   64   64   67   68   64   61
   63   67   67   63   64   62   63   68   67   67   66   64   65   68   63
   63   62   64   65   66   63   65   66   67   66   64   66   69   68   67
   63   57   59   64   65   64   66   62   62   65   63   64   66   64   63
   62   61   62   65   64   60   60   61   66   66   63   67   66   63   63
   65   66   67   65   65   66   65   63   62   62   60   63   66   66   64
   62   64   63   63   64   64   63   61   62   60   62   61   61   63   64
   61   62   62   64   66   65   65   63   61   63   64   64   61   60   62
      ⋮
```

```
imshow(I);
```

Since an image is just a series of numbers, we can differentiate them to find out the **gradient**. Now for a smooth region of the Image, the gradient is nearly constant. However, for a sharp edge, this will have a large value.

```
Igray = rgb2gray(I);
imshow(diff(Igray))
```



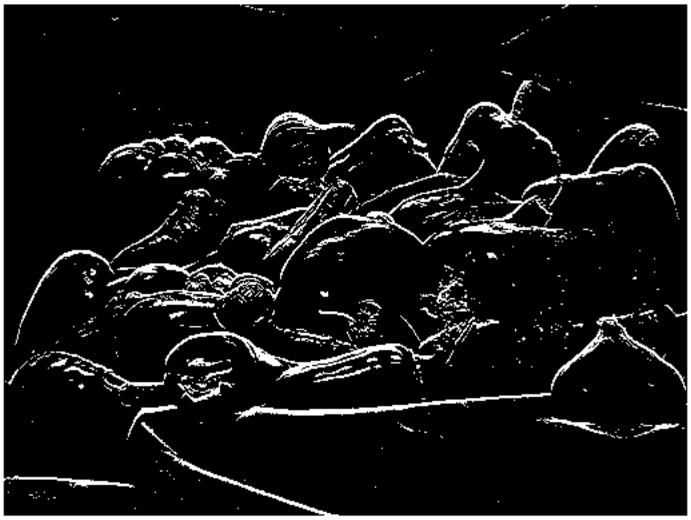
We can now simply segment the differentiated image to get the sharp edges. The threshold value can be adjusted to fine-tune the edge detection.

```
Iedge = diff(Igray);
Iedge = 383×512 uint8 matrix
     1     0     0     0     0     0     0     2     2     0     0     2     0     0     1     2     1     0     0     1···
     0     0     0     0     3     1     0     0     0     1     1     1     3     1     1     1     1     0     0     1
     0     0     1     0     0     0     2     1     1     2     1     0     0     2     2     2     0     1     1     1
     1     1     1     0     1     0     1     0     0     0     0     1     2     0     0     0     0     0     1     0
     1     0     0     1     0     0     0     0     0     1     0     0     0     0     0     0     0     1     0     1
     0     0     0     0     0     1     0     0     2     0     2     2     0     1     0     0     2     1     1     0
     1     1     1     0     0     3     1     0     0     0     0     0     0     0     0     0     0     1     0     0
     0     0     0     1     2     0     0     0     0     0     1     0     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     1     0     2     1     4     0     0     0     0     0     0     0     1
     0     0     0     1     1     3     0     0     1     1     1     0     1     0     0     0     0     1     0     0
       ⋮
```

```
threshold_value = 7
```

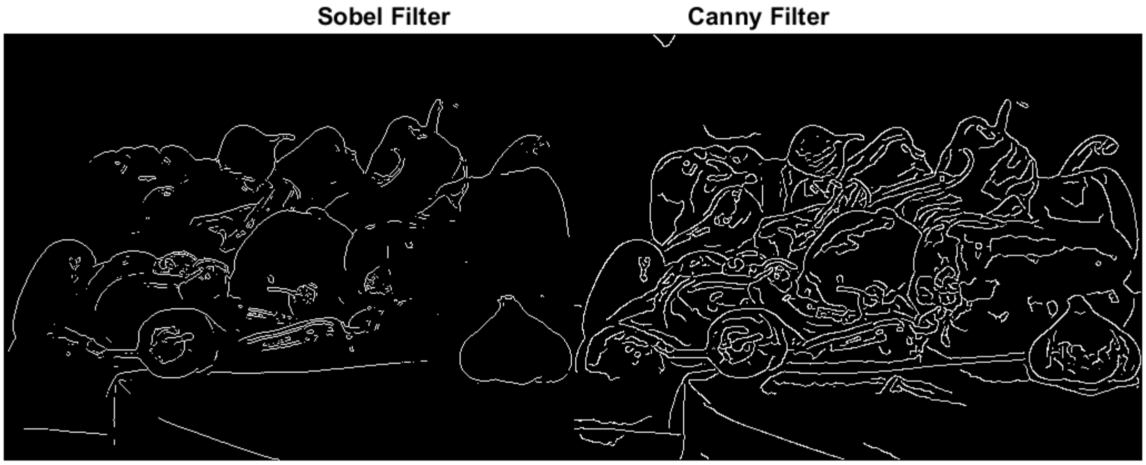
```
threshold_value = 7
```

```
Iedge(Iedge>=threshold_value) = 255;
Iedge(Iedge<threshold_value) = 0;
imshow(Iedge)
```



This is a rudimentary method and more advanced algorithms like Sobel or Canny apply various techniques to deliver sharper edges. These can include non-maximal suppression, computing gradients across a larger spatial region, and much more.

```
BW1 = edge(Igray,'sobel');
BW2 = edge(Igray,'canny');
figure;
imshowpair(BW1,BW2,'montage')
title('Sobel Filter                                               Canny Filter')
```



Sobel Filter

Canny Filter