IMAGE PROCESSING LAB-5

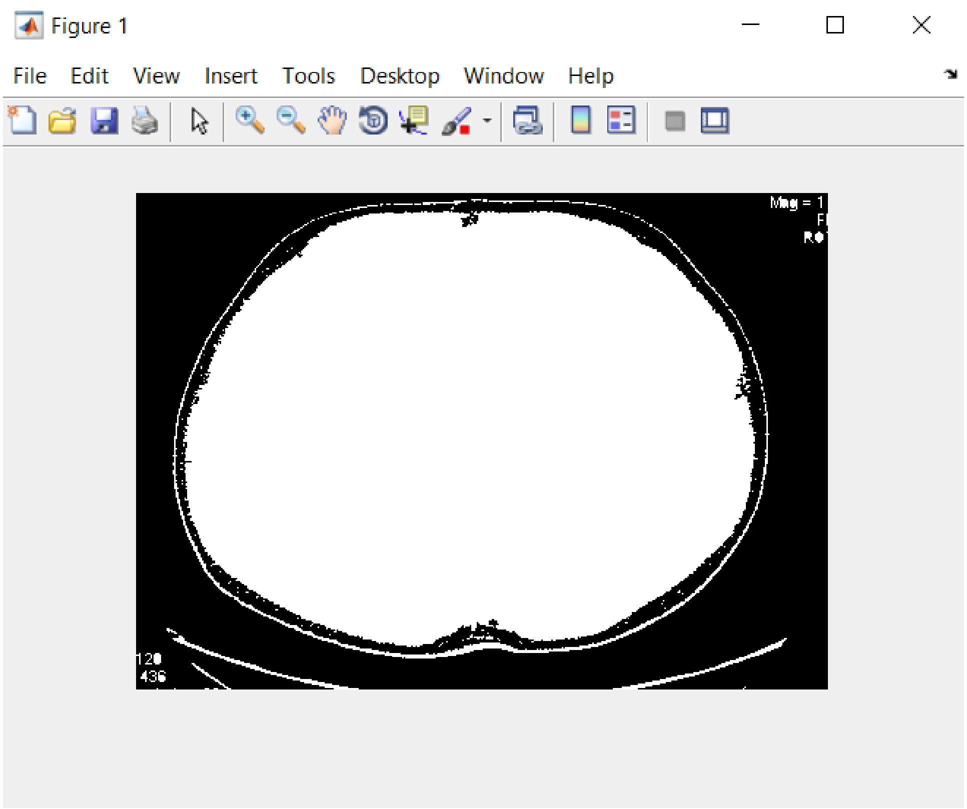
INTERNATIONAL BIOMETRICS (M1).

PRESENTE BY,

MOUNIKA REDDY PATIMIDHI.

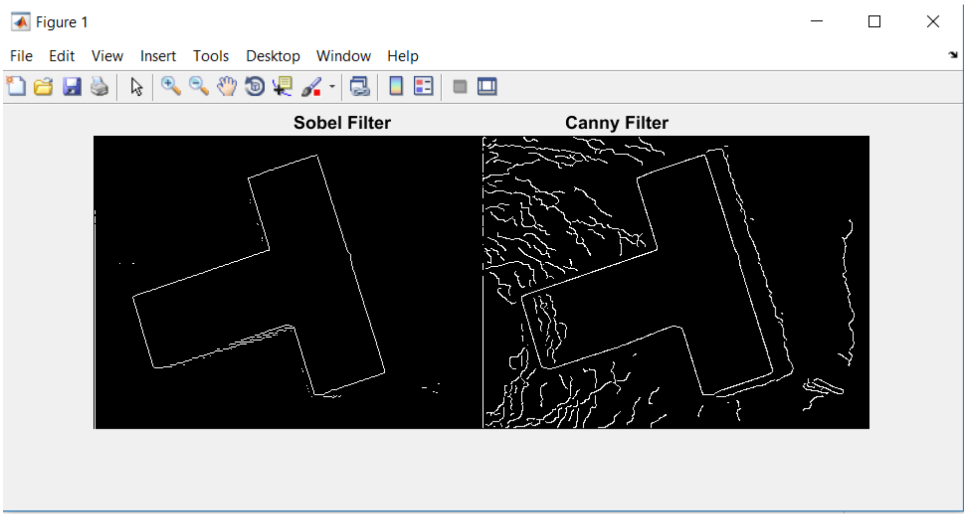
Segmentation by thresholding

```
clear all;

clc;

%%%excersise 1

%%%Read a grayscale image into the workspace.

I = imread('se000.jpg');

Imshow=('se000.jpg')

%%%Calculate a threshold using graythresh. The threshold is normalized to the range [0, 1].

level = graythresh(I)

BW = im2bw(I,level);

%%%Convert the image into a binary image using the threshold.

BW = imbinarize(I,level);

%%%%the original image next to the binary image.

imshowpair(I,BW,'montage');

%%%%%%Imfill function

BW2 = imfill(BW,'holes')

imshow(BW2);

se = strel('disk',5);
```



Edge function

```
I = imread('T.png');

BW1 = edge(I,'sobel');

BW2 = edge(I,'canny');

figure;

imshowpair(BW1,BW2,'montage')

title('Sobel Filter                         Canny Filter');

%%%%%BRICKS IMAGE

% I=imread('bricks.jpg');

% BW1 = edge(I,'sobel');

% BW2 = edge(I,'canny');

% figure;

% imshowpair(BW1,BW2,'montage')
```

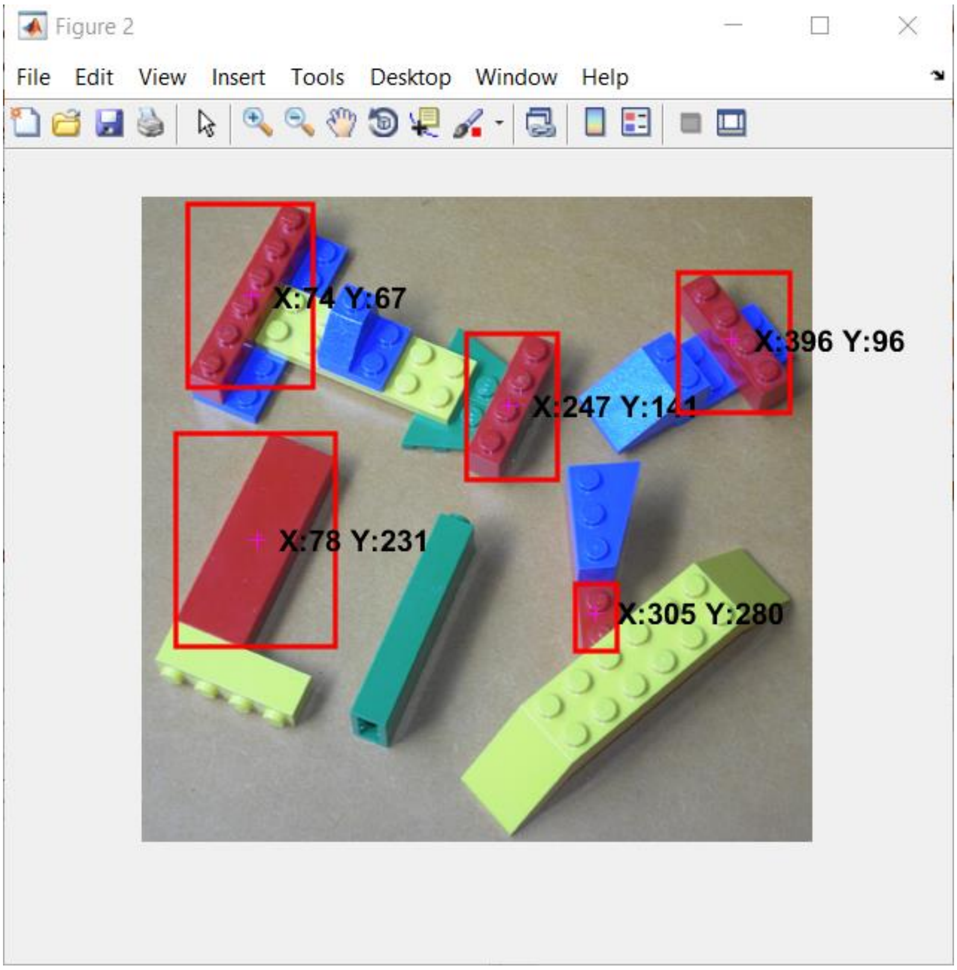


Download the image of toy building blocks bricks

```
data = imread('bricks.jpg');

   diff_im = imsubtract(data(:,:,2), rgb2gray(data));

   %Use a median filter to filter out noise

   diff_im = medfilt2(diff_im, [3 3]);

   diff_im = im2bw(diff_im,0.18);


   diff_im = bwareaopen(diff_im,300);


   bw = bwlabel(diff_im, 8);


   stats = regionprops(bw, 'BoundingBox', 'Centroid');


   % Display the image

   imshow(data)


   hold on


   for object = 1:length(stats)
```

```
        bb = stats(object).BoundingBox;

        bc = stats(object).Centroid;

        rectangle('Position',bb,'EdgeColor','r','LineWidth',2)

        plot(bc(1),bc(2), '-m+')

        a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), ' Y: ',
num2str(round(bc(2)))));

        set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color',
'black');

    end



clc;

clear;

close all;

figure, im = imread('bricks.jpg'); imshow(im)


%get color edges and normalize magnitude

C = coloredges(im);

C = C / max(C(:));


%get grayscale edges and normalize magnitude

G_image = single(rgb2gray(im)) / 255;

G = sqrt(imfilter(G_image, fspecial('sobel')').^2 + imfilter(G_image,
fspecial('sobel')).^2);

G = G / max(G(:));


%show comparison

figure, imshow(uint8(255 * cat(3, C, G, G)))
```

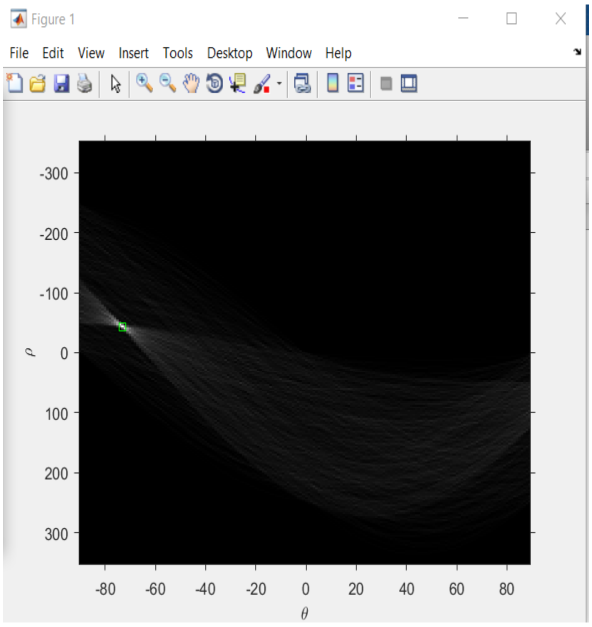
```
data = imread('bricks.jpg');
   diff_im = imsubtract(data(:,:,1), rgb2gray(data));
   %Use a median filter to filter out noise
   diff_im = medfilt2(diff_im, [3 3]);
   diff_im = im2bw(diff_im,0.18);

   diff_im = bwareaopen(diff_im,300);

   bw = bwlabel(diff_im, 8);

   stats = regionprops(bw, 'BoundingBox', 'Centroid');

   % Display the image
   imshow(data)

   hold on

   for object = 1:length(stats)
      bb = stats(object).BoundingBox;
      bc = stats(object).Centroid;
      rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
      plot(bc(1),bc(2), '-m+')
      a=text(bc(1)+15,bc(2), strcat('X: ', num2str(round(bc(1))), ' Y: ',
num2str(round(bc(2)))));
      set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color',
'black');
   End
```

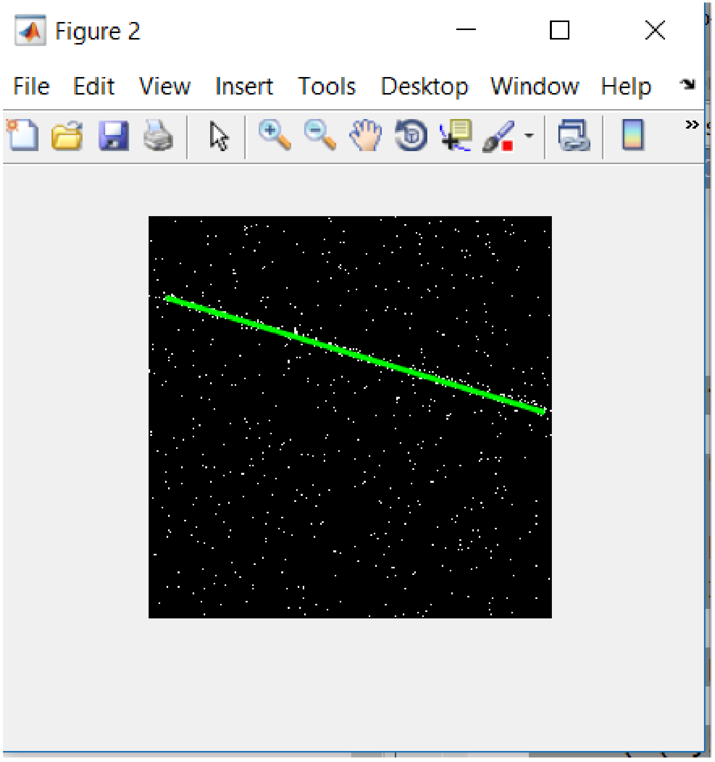


Hough transform

```
close all


clear all

clc


I=imread('first.png');


[H,T,R] = hough(I,'RhoResolution',pi,'ThetaResolution',1);

imshow(H,[],'XData',T,'YData',R,'InitialMagnification','fit');

xlabel('\theta'), ylabel('\rho');
```

```
axis on, axis normal, hold on;

P  = houghpeaks(H,1,'threshold',ceil(max(H(:))));

x = T(P(:,2)); y = R(P(:,1));
plot(x,y,'s','color','green');
% Find lines and plot them

lines = houghlines(I,T,R,P,'FillGap',5,'MinLength',3);
figure, imshow(I), hold on
```

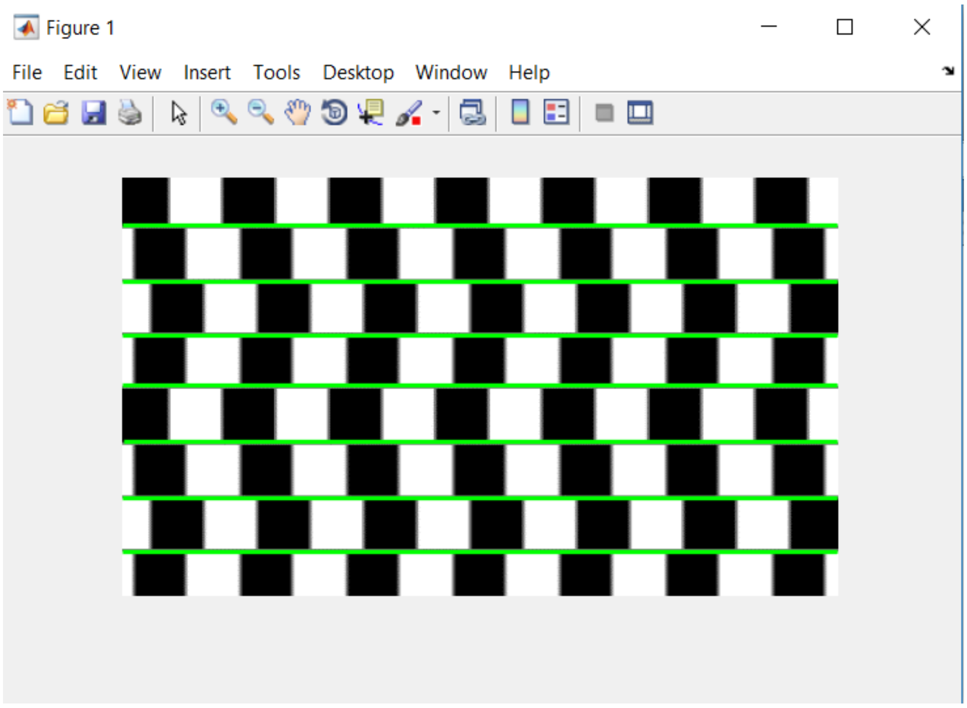




Hough transform.1

```
close all

clear all

clc


I=imread('lines.png');
```

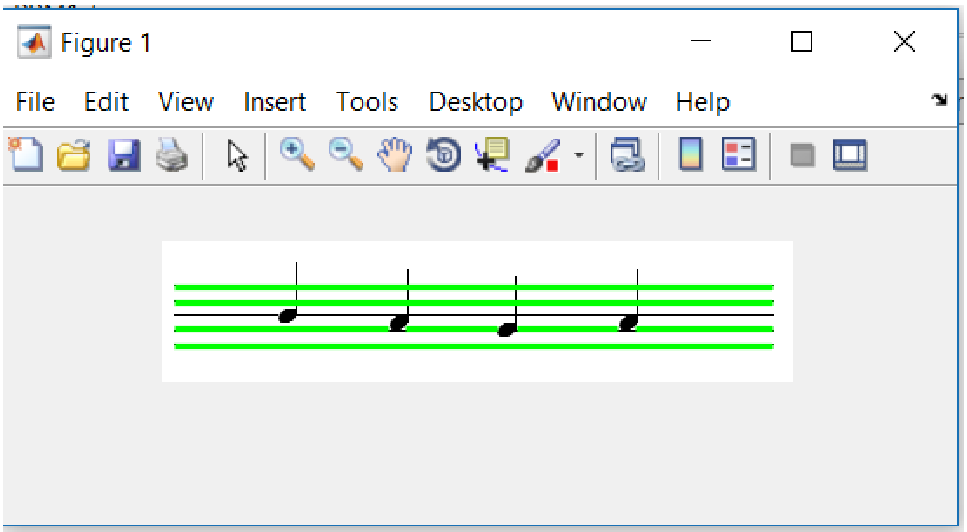
```
BW = edge(I,'sobel');

[H,T,R] = hough(BW,'RhoResolution',1,'ThetaResolution',1);
imshow(H,[],'XData',T,'YData',R,'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');

P  = houghpeaks(H,10,'threshold',ceil(0.3*max(H(:))));

x = T(P(:,2)); y = R(P(:,1));
plot(x,y,'s','color','white');
% Find lines and plot them
lines = houghlines(BW,T,R,P,'FillGap',5,'MinLength',3);
imshow(I), hold on

for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
End
```



Hough transform.2

```
close all

clear all

clc


I=imread('musical_score.png');

BW = edge(I,'sobel');


[H,T,R] = hough(BW,'RhoResolution',1,'ThetaResolution',1);

imshow(H,[],'XData',T,'YData',R,'InitialMagnification','fit');

xlabel('\theta'), ylabel('\rho');


P  = houghpeaks(H,10,'threshold',ceil(0.3*max(H(:))));


x = T(P(:,2)); y = R(P(:,1));
```

```
plot(x,y,'s','color','white');

% Find lines and plot them

lines = houghlines(BW,T,R,P,'FillGap',5,'MinLength',3);

imshow(I), hold on

for k = 1:length(lines)

  xy = [lines(k).point1; lines(k).point2];

  plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

End
```



Descriptors

```
clc;

close all;

clear;


%%%%%%%%%%%%Descriptors

% SP

SP = imread('card_spade.tif');

statsSP = regionprops(SP, 'all');   % parameters of binary image

areaSP = statsSP.Area;
```

```
perimeterSP = statsSP.Perimeter;
PSP = bwperim(SP);
circularitiesSP = perimeterSP .^ 2 ./ (4 * pi * areaSP);

figure; imshowpair(SP, PSP, 'montage'); title('Region and Perimeter of Spade');

hold on
for i = 1:length(statsSP)
   bb = statsSP(i).BoundingBox;
   bc = statsSP(i).Centroid;
   rectangle('Position', bb, 'EdgeColor', 'r', 'LineWidth', 1)
   plot(bc(1), bc(2), '-m+')
   a = text(bc(1) + 15, bc(2), strcat('X: ', num2str(round(bc(1))), ', Y: ', num2str(round(bc(2)))));
   set(a, 'FontName', 'Arial', 'FontWeight', 'bold', 'FontSize', 12, 'Color', 'green');
End
```

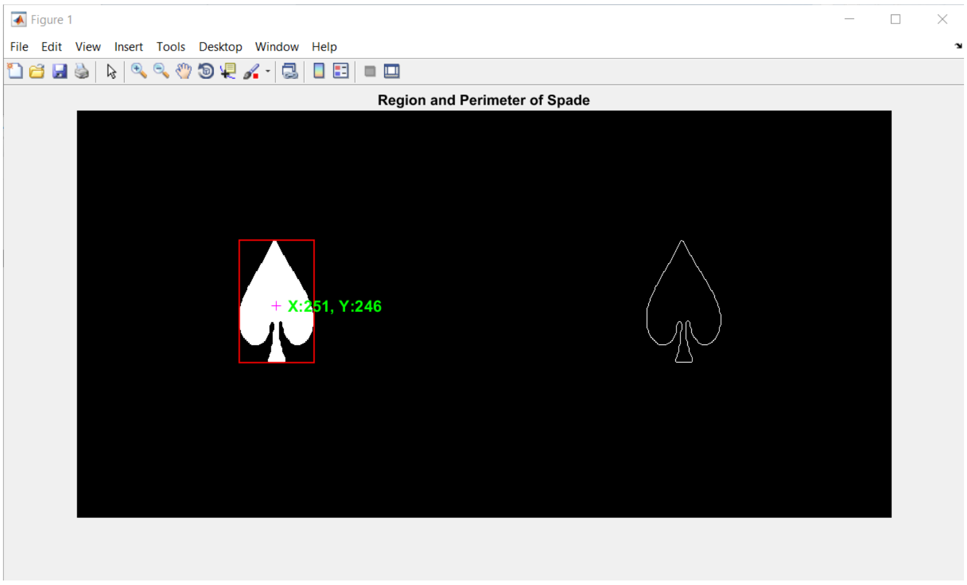


Figure 1
File Edit View Insert Tools Desktop Window Help
Region and Perimeter of Spade
X:251, Y:246