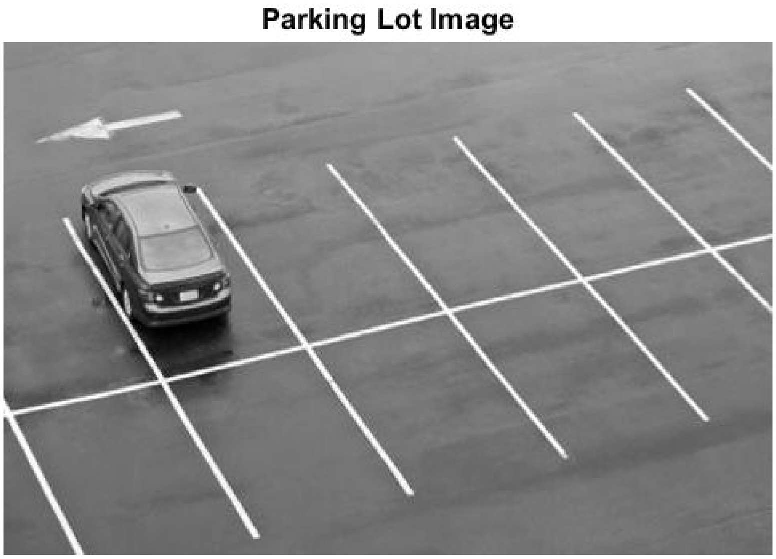
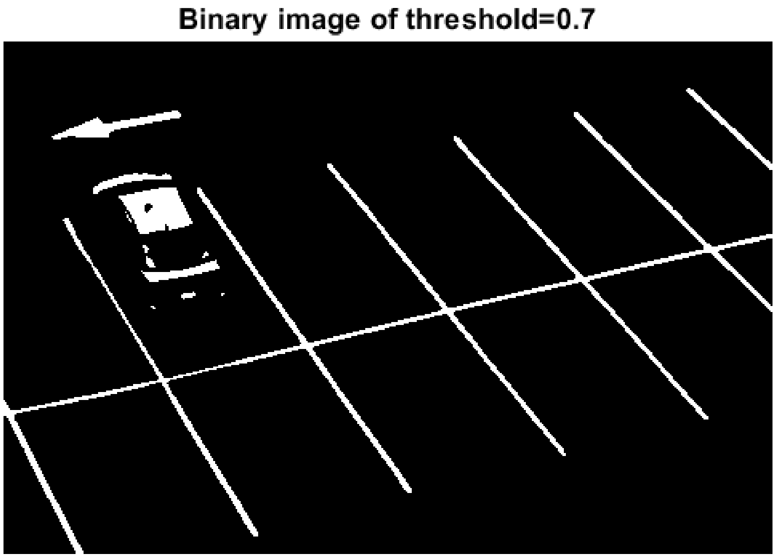
```
clear all
close all

%% Q(3.1)
image = imread("ParkingLot.jpg");
figure(1)
imshow(image);
title('Parking Lot Image')
```

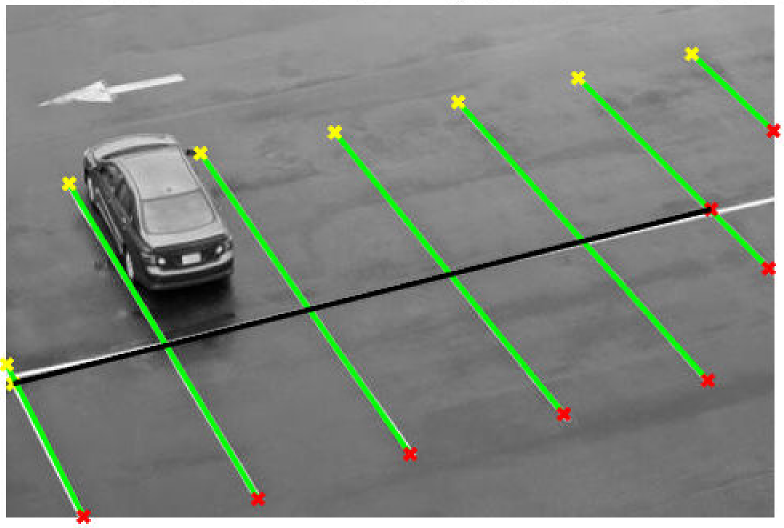


```
bin_image = imbinarize(imgaussfilt(image,1),0.7);
figure(2)
imshow(bin_image);
title('Binary image of threshold=0.7');
```

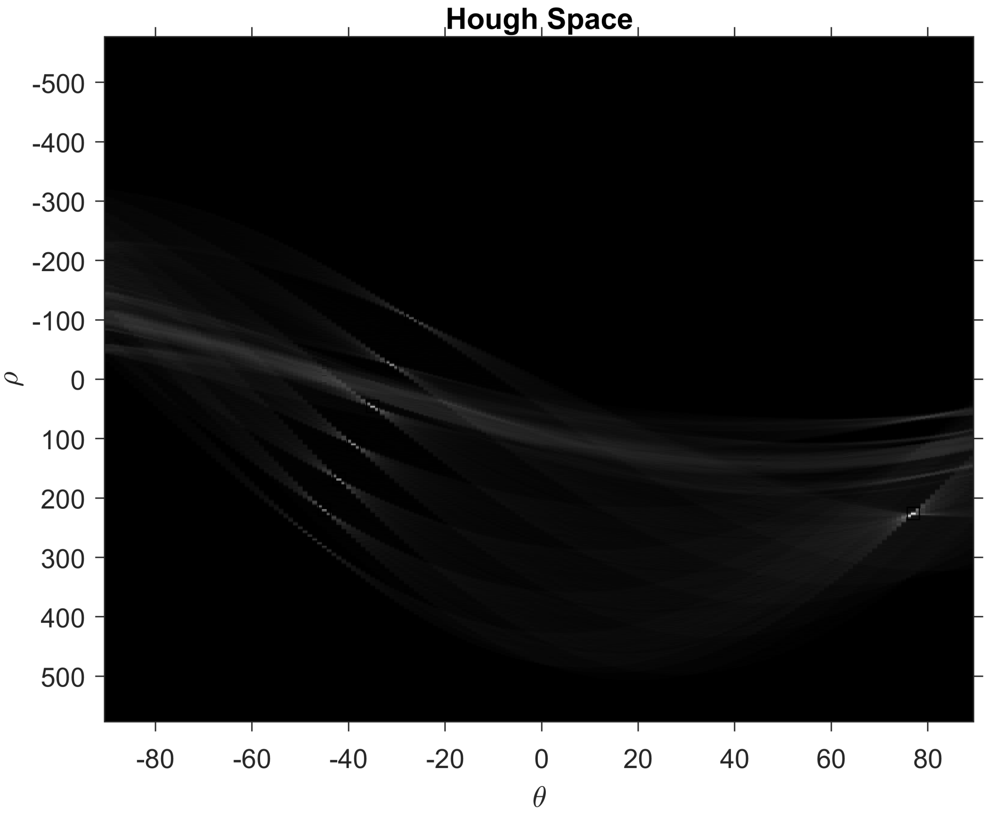


```
%% Q(3.2)
BW=im2bw(image,0.93);
BW=imfill(BW,'holes');
[H,theta,rho] = hough(BW);
P = houghpeaks(H,8,'threshold',ceil(0.3*max(H(:))));
x = theta(P(:,2));
y = rho(P(:,1));
lines = houghlines(BW,theta,rho,P,'FillGap',400,'MinLength',50);
figure(3)
imshow(image)
hold on
max_len = 0;
for k = 1:length(lines)
xy = [lines(k).point1; lines(k).point2];
plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
% Plot beginnings and ends of lines
plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
% To determine the endpoints of the longest line segment
len = norm(lines(k).point1 - lines(k).point2);
if ( len > max_len)
max_len = len;
xy_long = xy;
end
end
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','black');
title('The line detection between parking spaces with their points')
```

The line detection between parking spaces with their points



```
%% Q(3.4)
image_parking_edge = edge(bin_image,'approxcanny');
[H, T, R]= hough(bin_image,'RhoResolution',2);
P = houghpeaks(H,7,'threshold',ceil(0.3*max(H(:))),'NHoodSize', [19 19]);
lines = houghlines(bin_image,T,R,P,'FillGap',50,'MinLength',60);
figure(4)
imshow(H,[],'XData',T,'YData',R,'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');
title('Hough Space');
axis on, axis normal, hold on;
plot(T(P(:,2)),R(P(:,1)),'s','color','black');
hold off;
```



```
figure(5)
imshow(image);
hold on
for i = 1:length(lines)
    line = [lines(i).point1; lines(i).point2];
    hold on
    plot(line(:,1),line(:,2),'LineWidth',2,'Color','w');
end
title('Parking Space Detection')
hough_transform = [lines(1).point1;lines(1).point2];
pointx = [];
pointy = [];
pointz = [];
imshow(image);
hold on;
for i = 1:length(lines)
    line = [lines(i).point1; lines(i).point2];
    hold on
    plot(line(:,1),line(:,2),'LineWidth',3,'Color','b');
end
for i=2:length(lines)
    line_x = [lines(i).point1;lines(i).point2];
    X1 = line_x(1,2) -line_x(2,2);
    Y1 = line_x(2,1) -line_x(1,1);
    Z1 = line_x(1,1) *line_x(2,2) - line_x(2,1) * line_x(1,2);
    X2 = hough_transform(1,2) - hough_transform(2,2);
    Y2 = hough_transform(2,1) - hough_transform(1,1);
    Z2 = hough_transform(1,1) * hough_transform(2,2)- hough_transform(2,1) * hough_transform(1
    % To find the cross link between the lines for detection
    slope = X1*Y2 - X2*Y1;
    i_x = (Y1*Z2 - Y2*Z1)/slope;
    i_y = (X2*Z1 - X1*Z2)/slope;
    pointx(end+1,:) = [i_x,i_y];
    pointz(end+1,:) = lines(i).point2;
    pointy(end+1,:) = lines(i).point1;
    end
    [~,I] = sort(pointx(:,1));
    pointx = pointx(I,:);
    [~,I] = sort(pointz(:,1));
    pointz = pointz(I,:);
    [~,I] = sort(pointy(:,1));
    pointy = pointy(I,:);
    for i = 1:length(pointx)-1
    x = [pointz(i,1) pointx(i,1) pointx(i+1,1) pointz(i+1,1)];
    y = [pointz(i,2) pointx(i,2) pointx(i+1,2) pointz(i+1,2)];
    plot(x,y);
    fill_point_rect = fill(x,y,rand(1,3));
    set(fill_point_rect,'facealpha',0.4)
    x = [pointy(i,1) pointx(i,1) pointx(i+1,1) pointy(i+1,1)];
    y = [pointy(i,2) pointx(i,2) pointx(i+1,2) pointy(i+1,2)];
    fill_point_rect = fill(x,y,rand(1,3));
    set(fill_point_rect,'facealpha',0.4)
end
```