
Table of Contents

Q1	1
A	1
Image: object1.bmp	1
Image: object2.bmp	4
Image: object3.bmp	6
Used function q3b2g:	9

Q1

```
clear
close all
clc

addpath(genpath('./materials'))
```

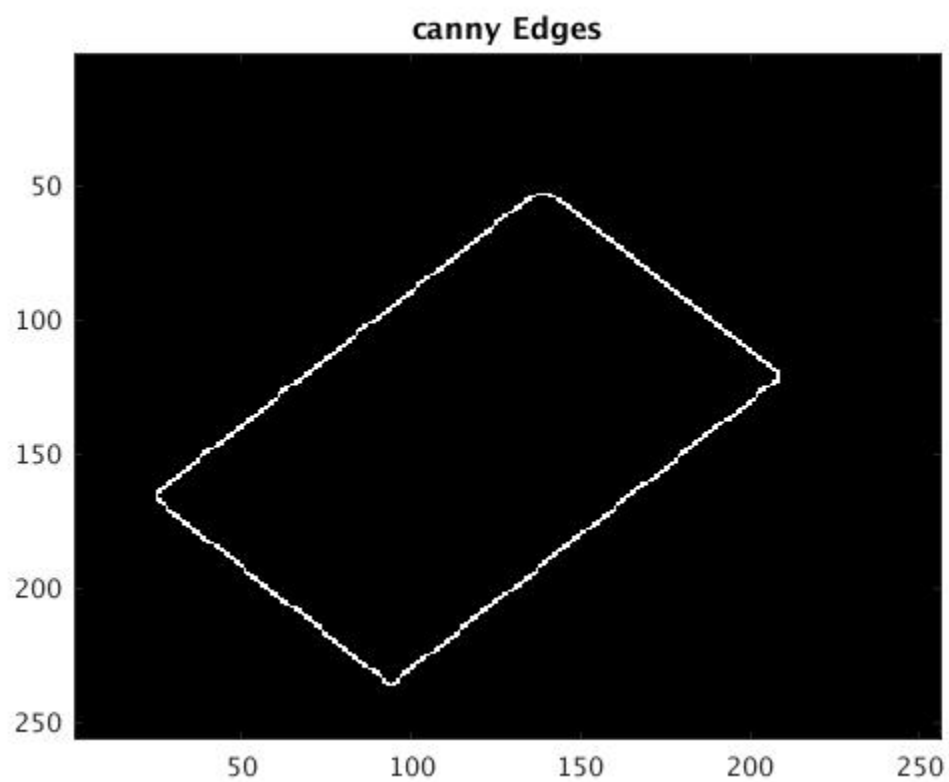
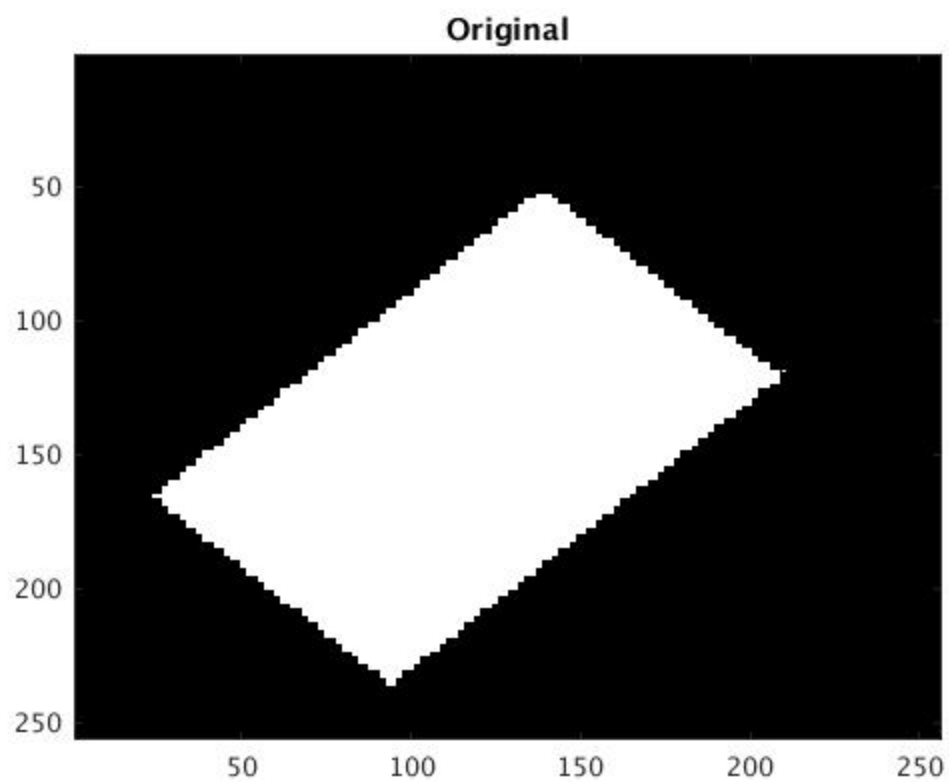
A

```
OB1 = imread('object1.bmp');
R2 = imread('object2.bmp');
C3 = imread('object3.bmp');
```

Image: object1.bmp

```
q3b2g(OB1);

Found line: r = -47, theta = 45
Found line: r = 52, theta = 45
Found line: r = -100, theta = 135
Found line: r = 62, theta = 135
There are Vertical lines!!
There are 2 parrellel lines with angle: 45
There are 2 parrellel lines with angle: 135
```



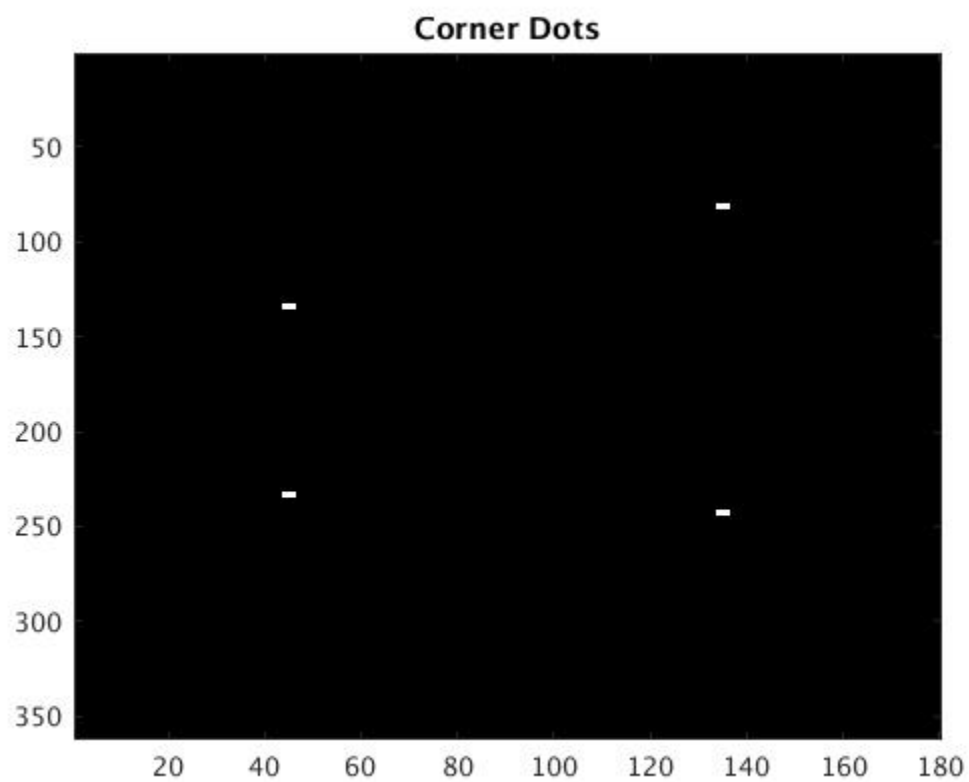
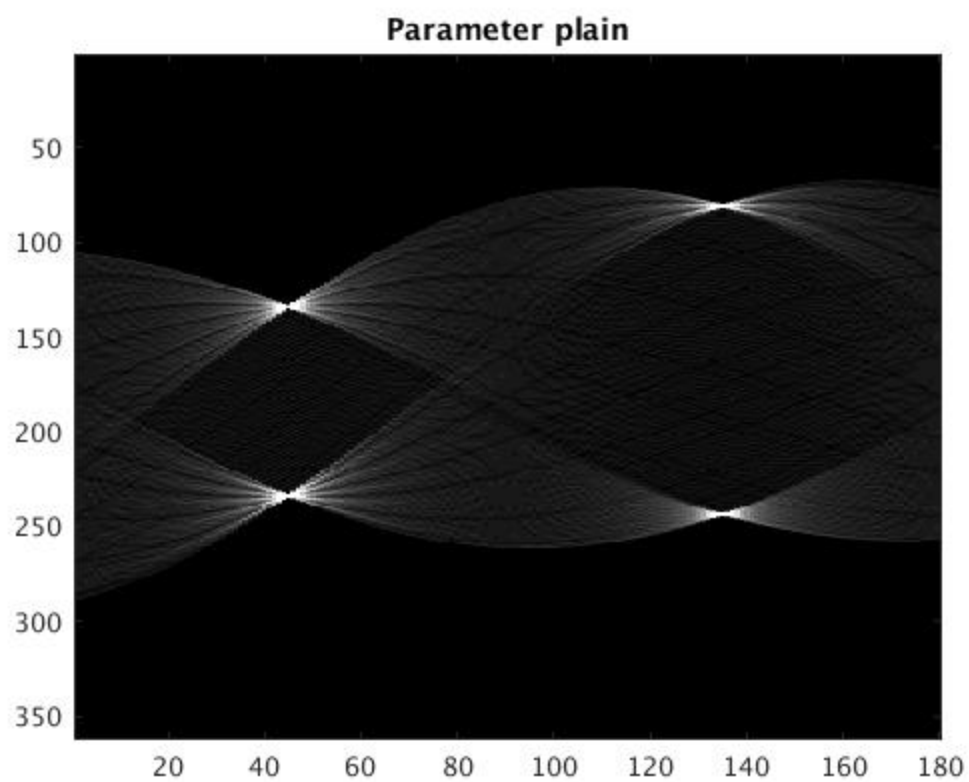


Image: object2.bmp

```
q3b2g(R2);
```

```
Found line: r = -63.5, theta = 90
```

```
Found line: r = 63.5, theta = 90
```

```
Found line: r = 58.5, theta = 28
```

```
Found line: r = -58.5, theta = 152
```

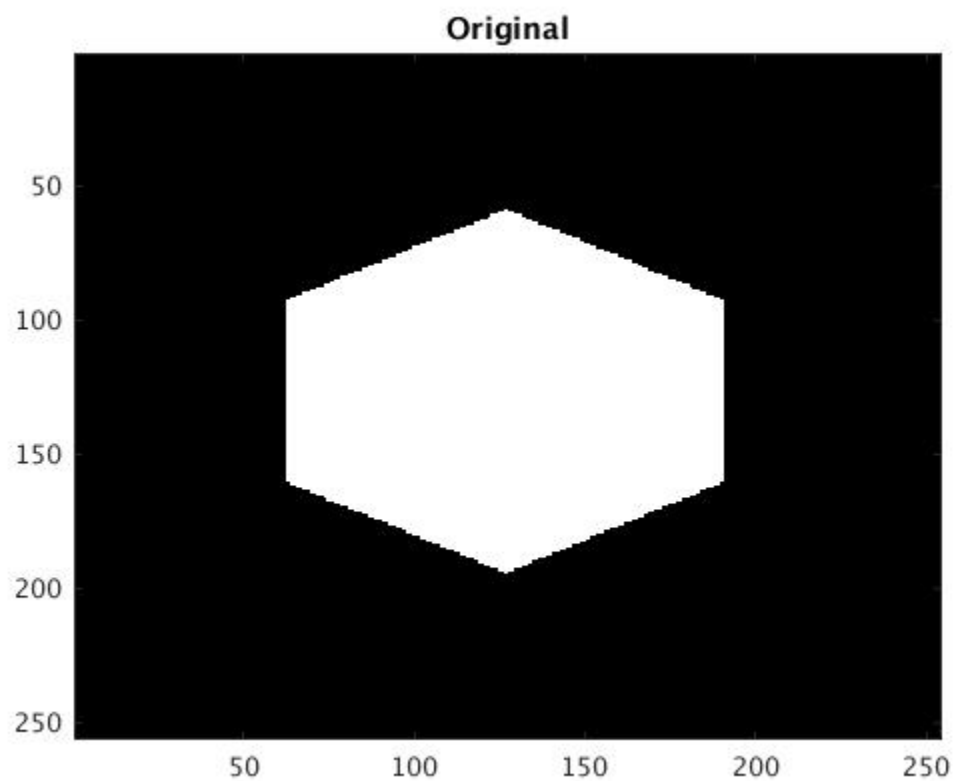
```
Found line: r = -61.5, theta = 28
```

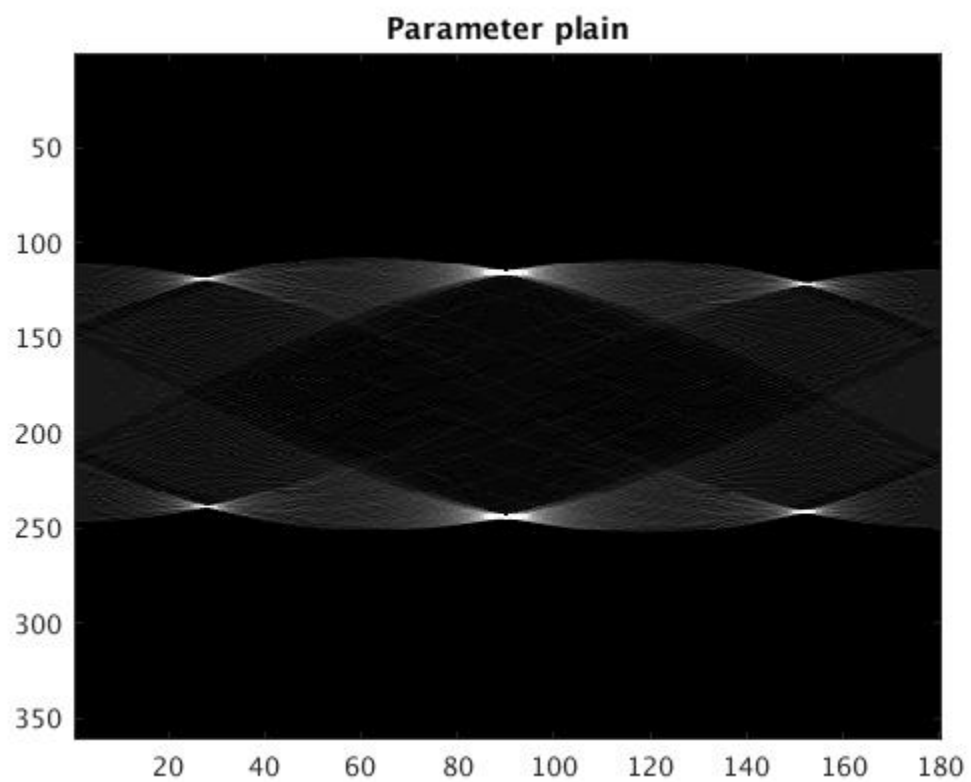
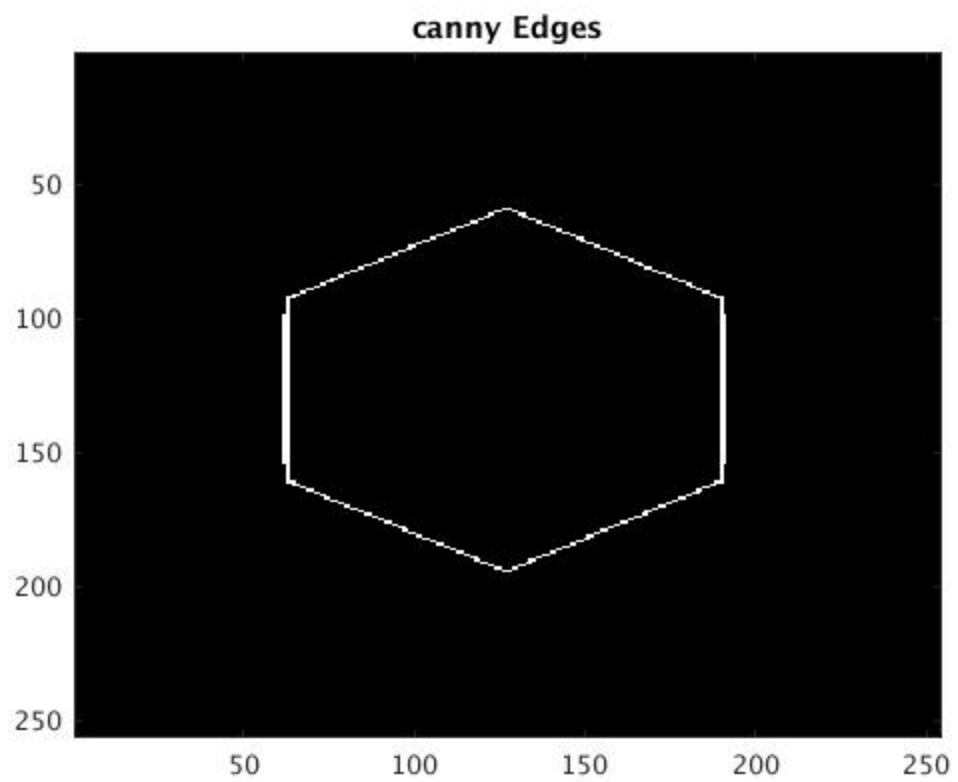
```
Found line: r = 61.5, theta = 151
```

```
There are 2 parrellel lines with angle: 28
```

```
There are 2 parrellel lines with angle: 90
```

```
There are 2 parrellel lines with angle: 151
```





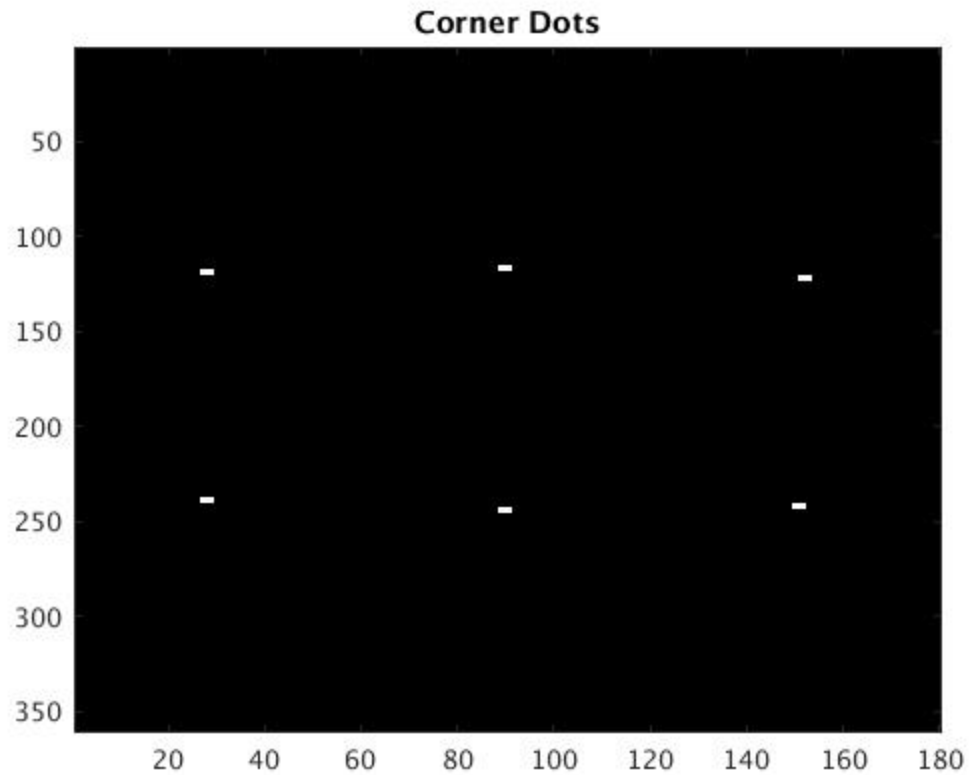


Image: object3.bmp

```
q3b2g(C3);
```

```
Found line: r = -21.5, theta = 90
```

```
Found line: r = 105.5, theta = 90
```

```
Found line: r = 114.5, theta = 28
```

```
Found line: r = -5.5, theta = 28
```

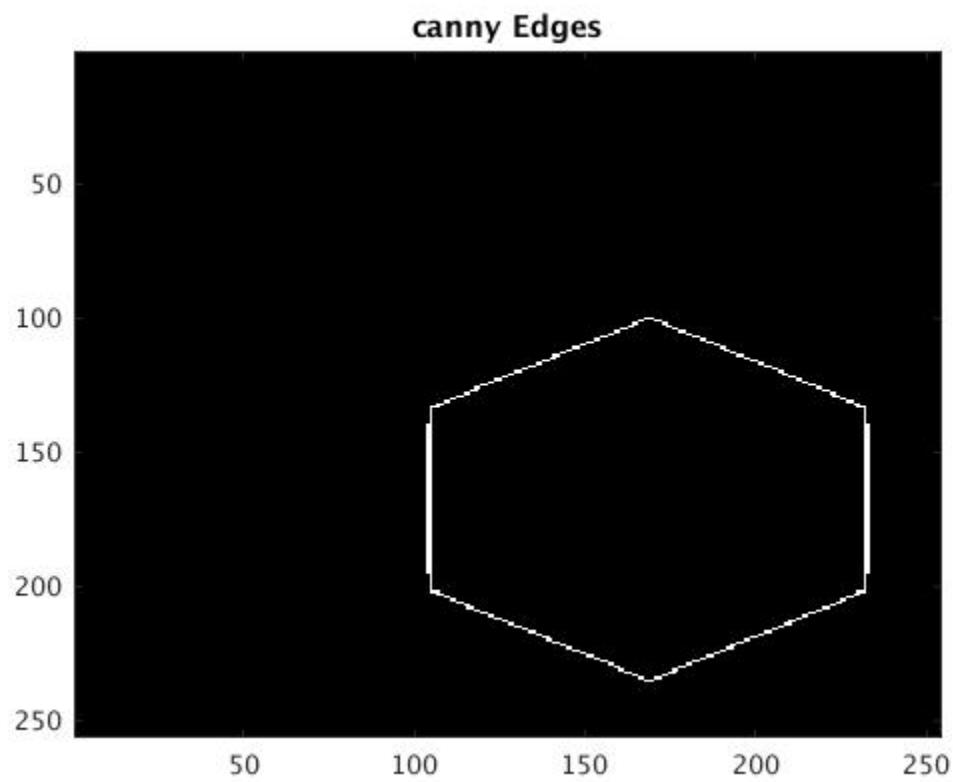
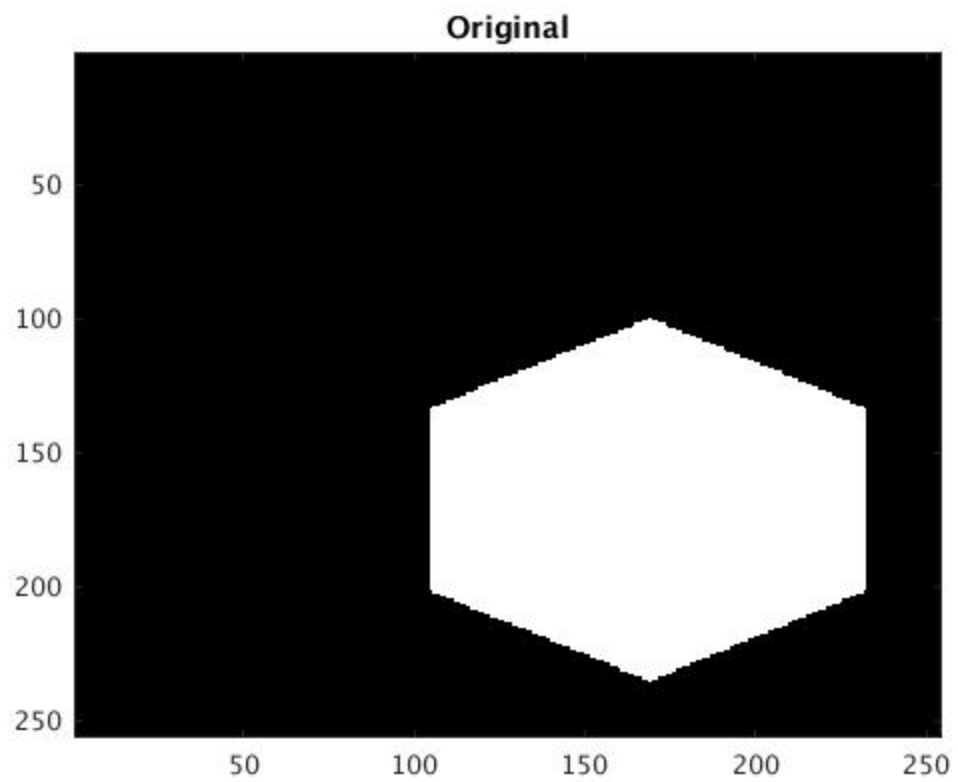
```
Found line: r = 44.5, theta = 152
```

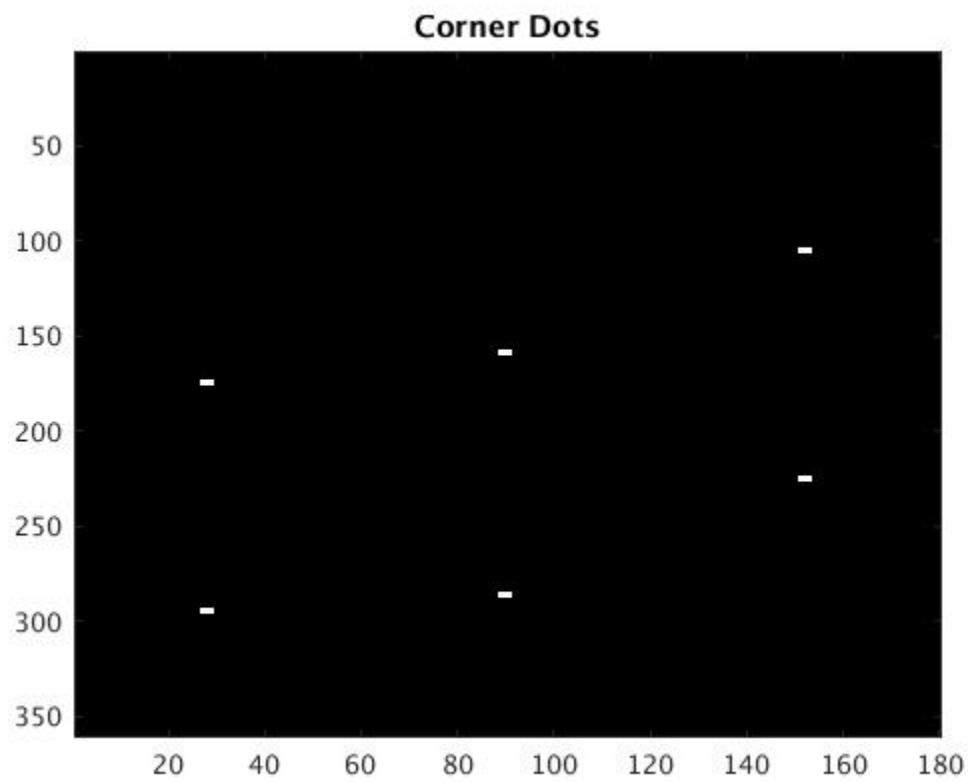
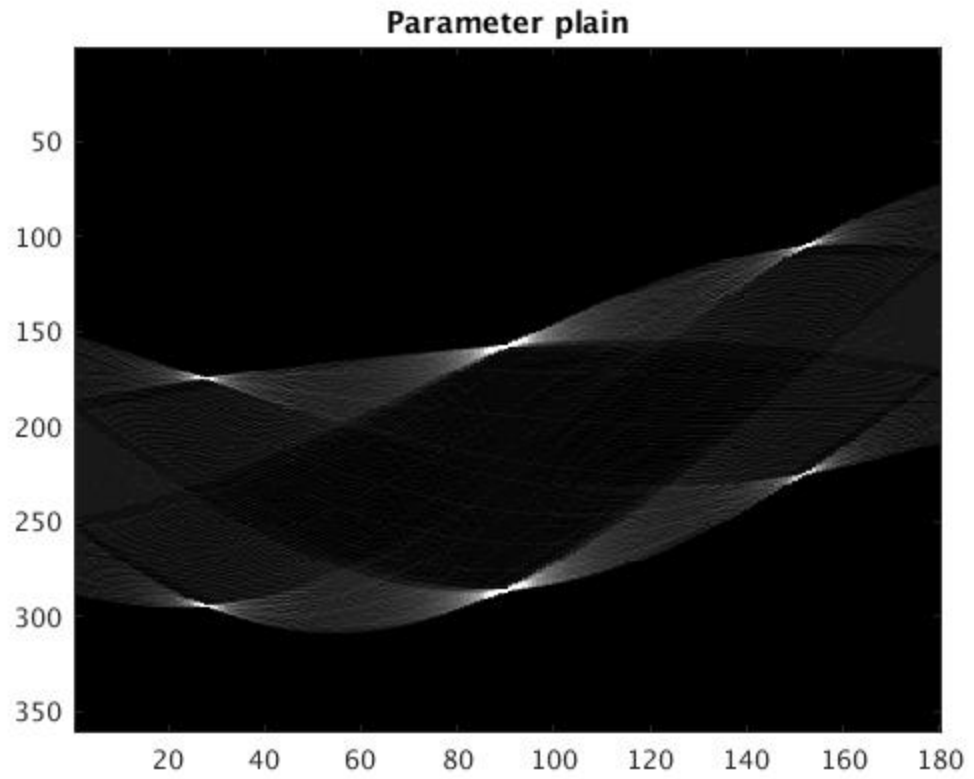
```
Found line: r = -75.5, theta = 152
```

```
There are 2 parrellel lines with angle: 28
```

```
There are 2 parrellel lines with angle: 90
```

```
There are 2 parrellel lines with angle: 152
```





Used function q3b2g:

```
function q3b2g(im)

figure(), image(im), colormap(gray(2)), title('Original');

% B
imEdges = edge(im, 'canny');
figure(), image(imEdges), colormap(gray(2)), title('canny Edges');

% C
edgesPoint = findEdgePoints(imEdges);

% D
diameter = @(im) round(norm(size(im)));

d = diameter(imEdges);
parameterPlain = zeros(d, 180);

% E
Rho = @(x,y, theta) round(x*cos(theta) + y * sin(theta) + d/2);

for J = 1:length(edgesPoint)
    for i=1:180
        theta = deg2rad(i);
        x = edgesPoint(J,1);
        y = edgesPoint(J,2);
        r = Rho(x, y, theta);
        parameterPlain(r, i) = parameterPlain(r, i) + 1;
    end
end
figure(), image(parameterPlain), colormap(gray(32)), title('Parameter plain');

% F
thresh = 20;

parameterPlain(parameterPlain<thresh) = 0;
temp = parameterPlain;
parameterPlain2Plot = zeros(size(parameterPlain));
bolding = -1:1;

while 1
    [val, ind] = max(temp(:));
    if val == 0
        break
    end
    [indI, indJ] = ind2sub(size(temp), ind);
    area = -10:10;
    temp(area + indI, area + indJ) = 0;
    parameterPlain(area + indI, area + indJ) = 0;
    parameterPlain(ind) = 1;
    parameterPlain2Plot(bolding + indI, boliding + indJ) = 128;
    fprintf("Found line: r = %g, theta = %g\n",indI,indJ);
end
figure(), image(parameterPlain2Plot), colormap(gray(128)), title('Corner D
```

```

line_angles = sum(parameterPlain);
oneLines = find(line_angles == 1) -1;
oneLines(oneLines<1) = [];
line_angles(oneLines) = line_angles(oneLines) + 1 ;

single_line_angles = line_angles;
single_line_angles(single_line_angles>1) = 1;
vertical_lines = single_line_angles(1:90) + single_line_angles(91:180);
vertical_lines(vertical_lines < 2) = 0;
if find(vertical_lines > 1)
    fprintf("There are Vertical lines!!\n");
end
parallel_lines_angles = find(line_angles>1);
if find(parallel_lines_angles > 1)
    for angle=parallel_lines_angles
        fprintf("There are %g parrellel lines with angle: %g\n", line_angl
    end
end

end
end

```

Published with MATLAB® R2017a