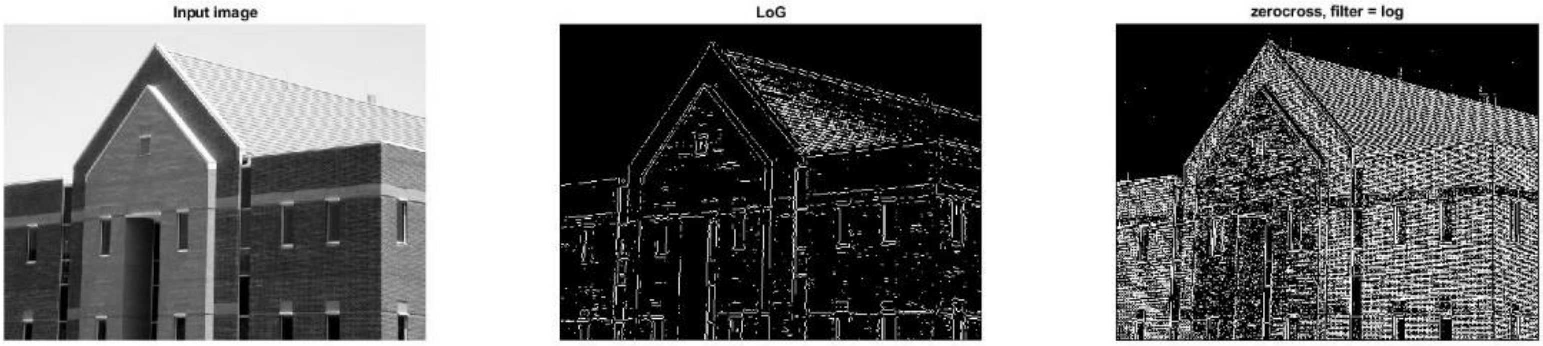
Digital image processing and vision systems – lab #8

Date performed: 18.05.2021	Group 2
Author name: Krzysztof Klimczyk	

1. Source codes and screenshots:

Task 8.2.1. – LoG

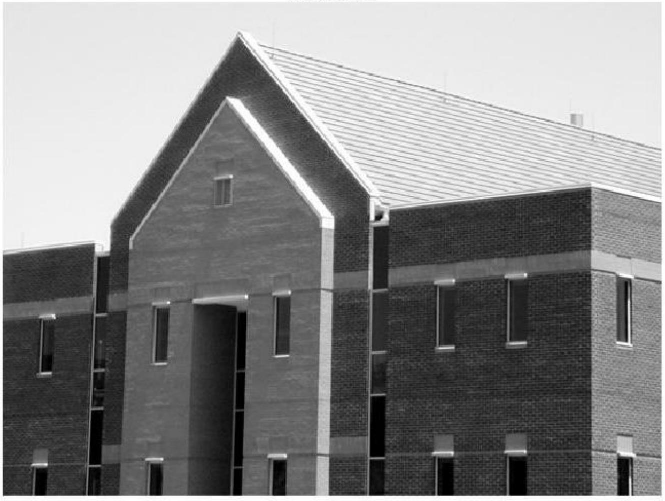
```
image = imread("house.png");

figure(1);
subplot(1,3,1);
imshow(image);
title('Input image')
subplot(1,3,2);
edge1 = edge(image,'log');
imshow(edge1);
title('LoG')
subplot(1,3,3);
h = fspecial('log');
edge1 = edge(image,'zerocross',[],h);
imshow(edge1);
title('zerocross, filter = log')
```



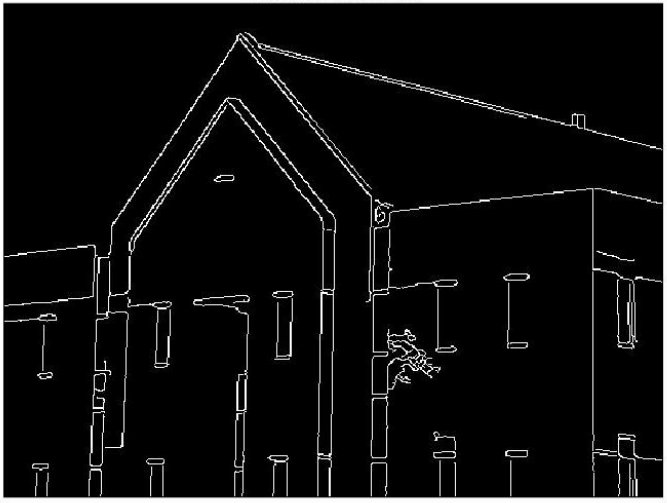
Task 8.2.2. – Canny

```
image = imread("house.png");

figure(1);
subplot(1,2,1);
imshow(image);
title('Input image')
subplot(1,2,2);
edge1 = edge(image,'canny',[0.01 0.30]);
imshow(edge1);
title('canny, thresh:[0.01 0.30]')
```

Input image



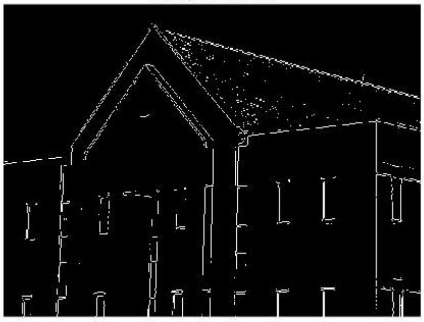
canny, thresh:[0.01 0.30]



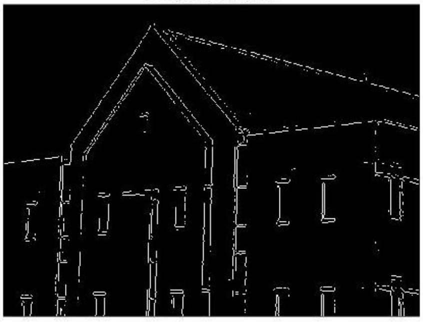
Task 8.2.3. – Comparison of edge detection

```
image = imread("house.png");

figure(1);
subplot(1,3,1);
edge1 = edge(image,'sobel',0.13);
imshow(edge1);
title('sobel, thresh:0.13')
subplot(1,3,2);
edge1 = edge(image,'log',0.009);
imshow(edge1);
title('LoG, thresh:0.009')
subplot(1,3,3);
edge1 = edge(image,'canny',[0.011 0.30]);
imshow(edge1);

title('canny, thresh:[0.011 0.30]')
```

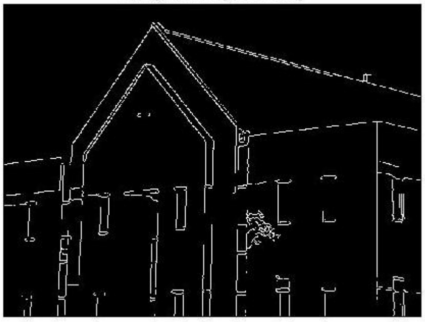
sobel, thresh:0.13



LoG, thresh:0.009

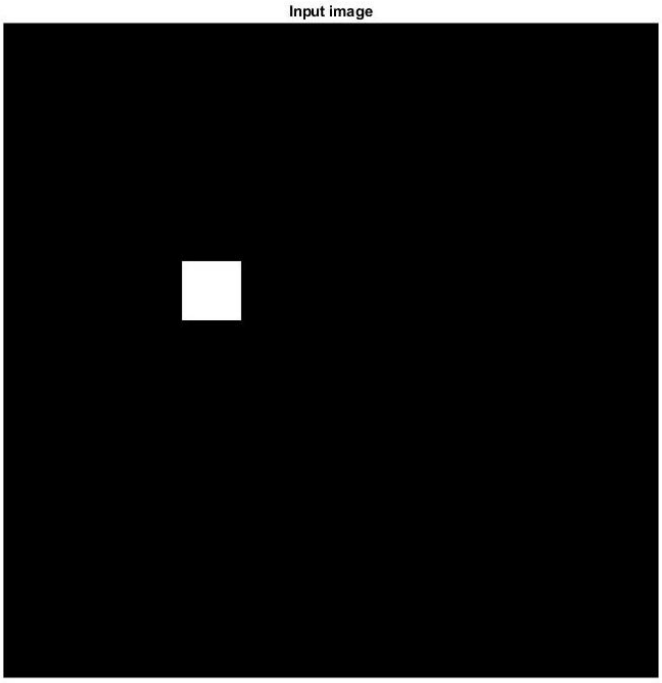


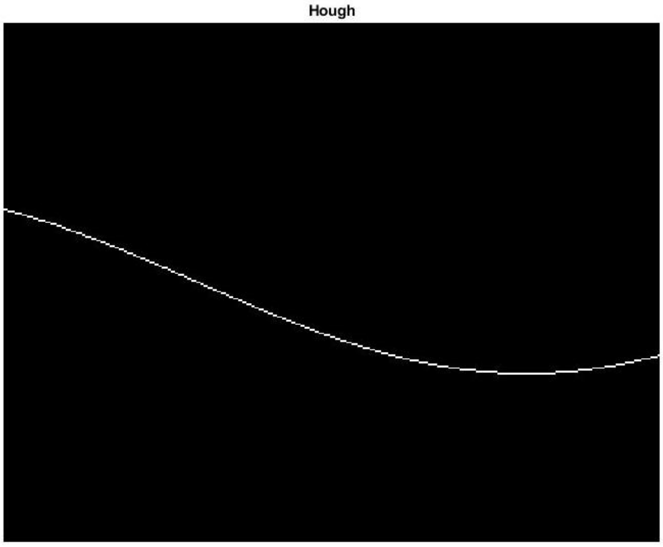
canny, thresh:[0.011 0.30]

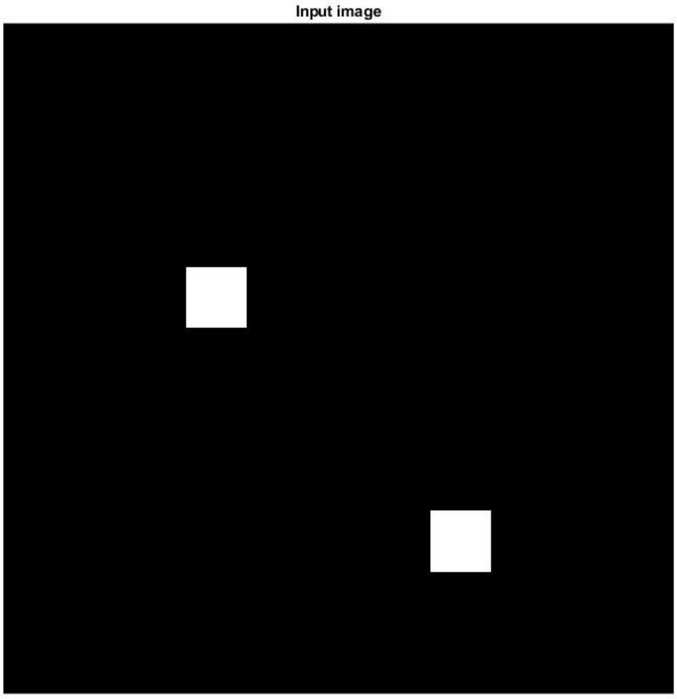


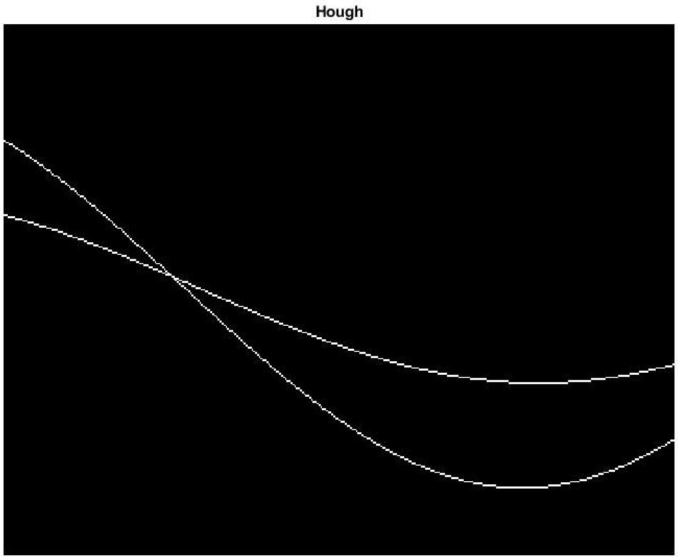
Task 8.3.1 – Hough transform for points

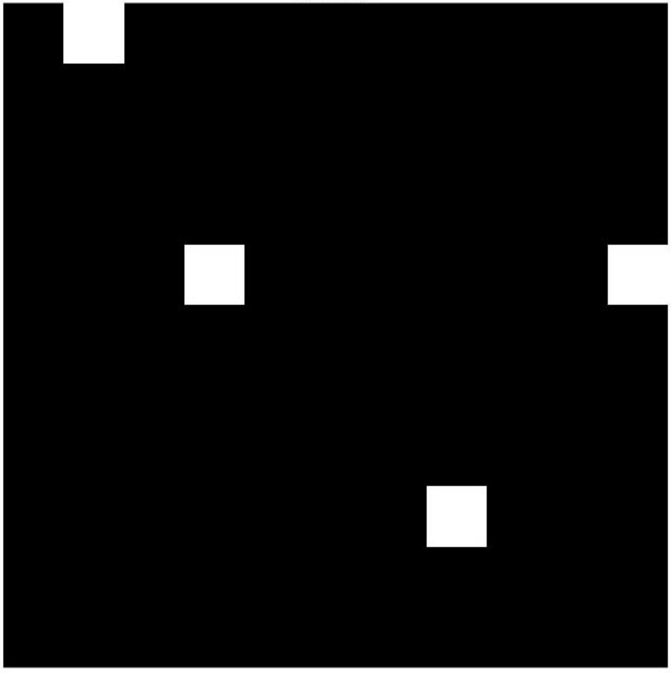
```
image = zeros(11,11);

figure(1);
subplot(1,2,1);
image(5,4) = 1;
imshow(image);
title('Input image')
subplot(1,2,2);
[H,theta,rho] = hough(image,'RhoResolution',0.1,'ThetaResolution',0.5);
imshow(H,[]);
title('Hough',[])


figure(2);
subplot(1,2,1);
image(5,4) = 1;
image(9,8) = 1;
imshow(image);
title('Input image')
subplot(1,2,2);
[H,theta,rho] = hough(image,'RhoResolution',0.1,'ThetaResolution',0.5);
imshow(H,[]);
title('Hough',[])

figure(3);
subplot(1,2,1);
image(5,4) = 1;
image(9,8) = 1;
image(1,2) = 1;
image(5,11) = 1;
imshow(image);
title('Input image')
subplot(1,2,2);
[H,theta,rho] = hough(image,'RhoResolution',0.1,'ThetaResolution',0.5);
imshow(H);
title('Hough',[])

figure(4);
subplot(1,2,1);
O=theta(90);
p=rho(135);
x = 0:0.1:10;
y =(p-x*cosd(O))/sind(O);

imshow(image);
hold on;
plot(x+1,y+1);
```


Input image

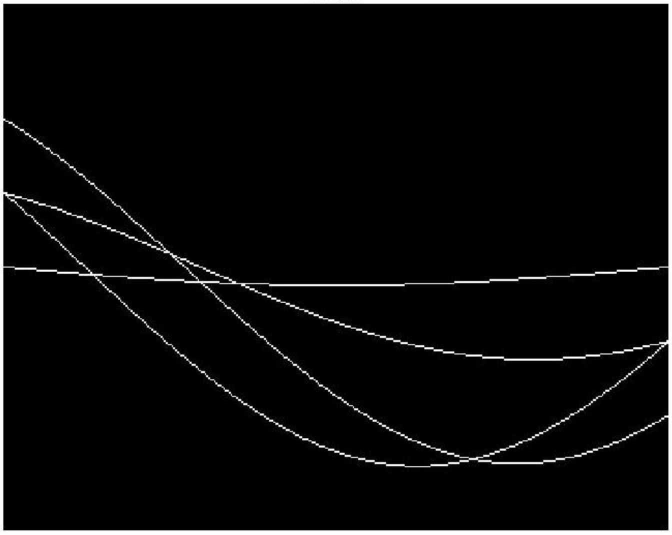

Hough


Input image

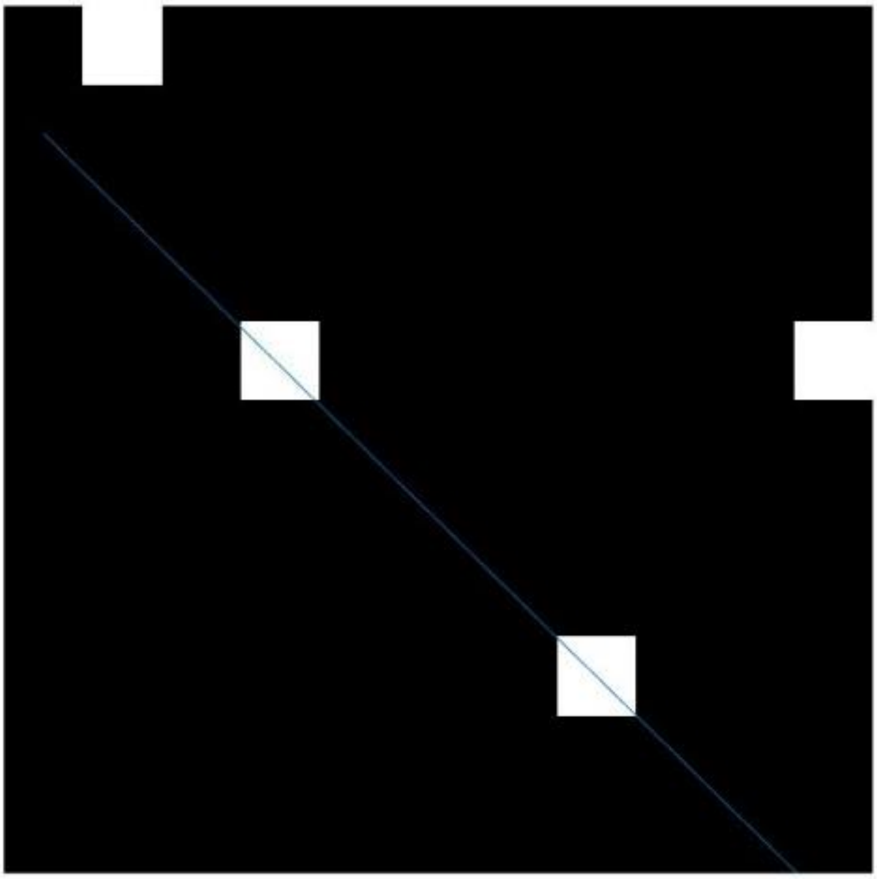

Hough



Input image

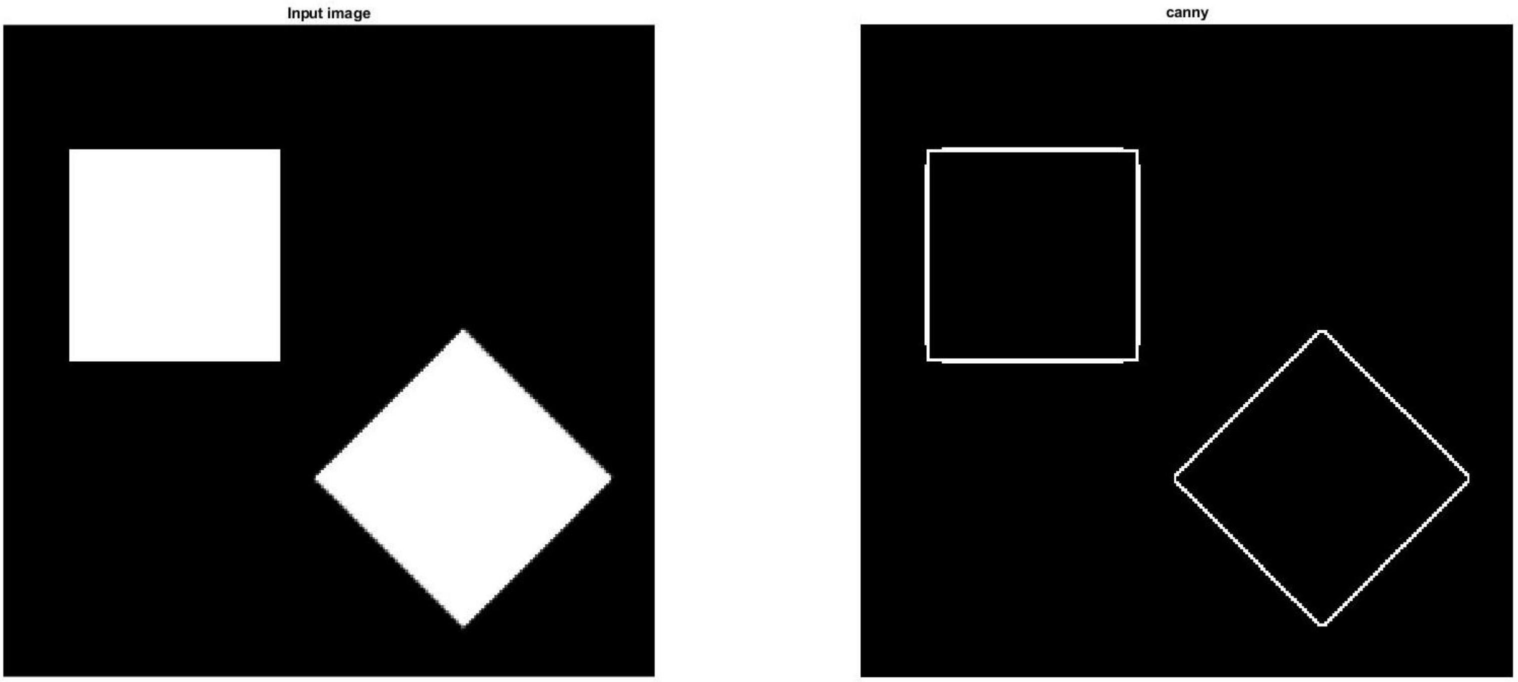


Hough



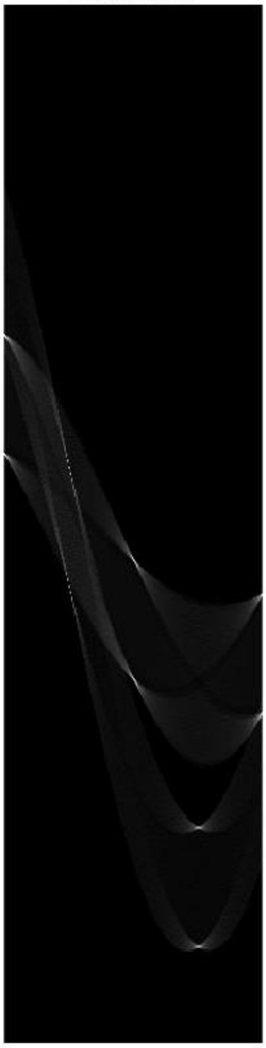
Task 8.3.2 – Hough transform for object

```matlab
image = imread("squares.png");

figure(1);
subplot(1,2,1);
imshow(image);
title('Input image')
subplot(1,2,2);
edge1 = edge(image,'canny');
imshow(edge1);
title('canny');

figure(2);
subplot(1,2,1);
[H,T,R] = hough(edge1);
imshow(H,[]);
title('Hough space')
subplot(1,2,2);
P = houghpeaks(H,10);
imshow(H,[]);
hold on;
plot(P(:,2),P(:,1),'o');
title('Hough peaks')

lines = houghlines(edge1,T,R,P,'FillGap',5,'MinLength',7);
figure, imshow(edge1), 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(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
      max_len = len;
      xy_long = xy;
   end
end
title('Hough lines')
```

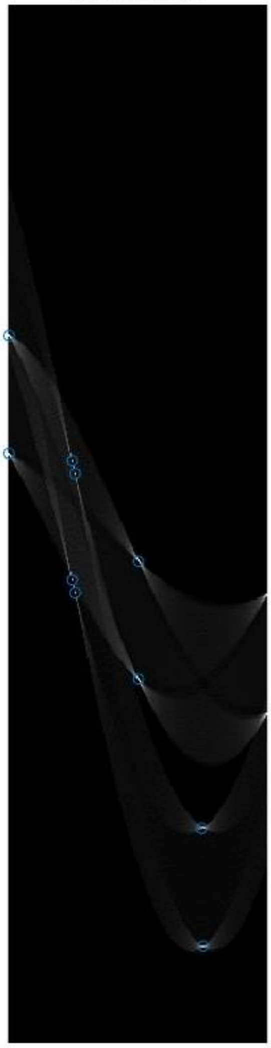


Input image

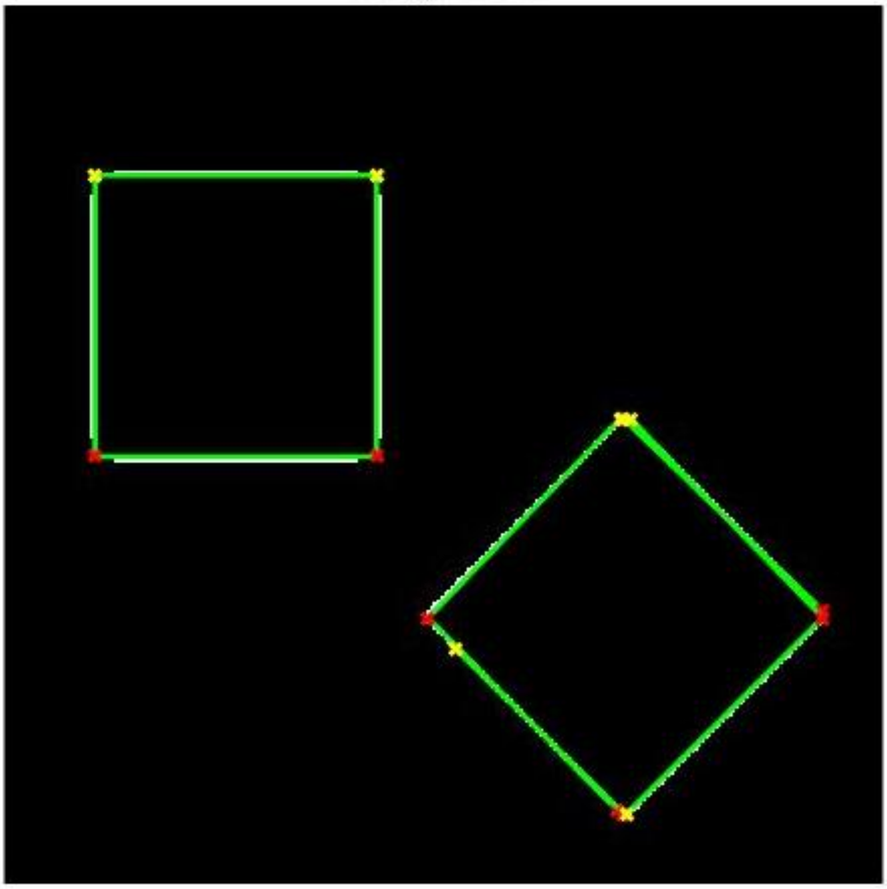
canny



Hough space



Hough peaks

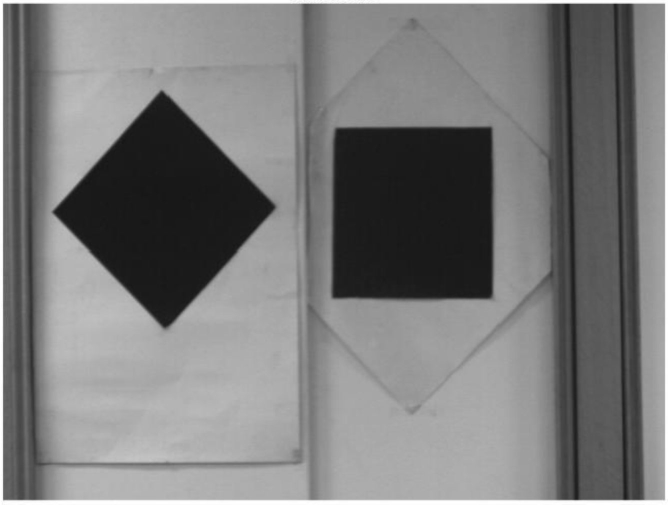


Hough lines

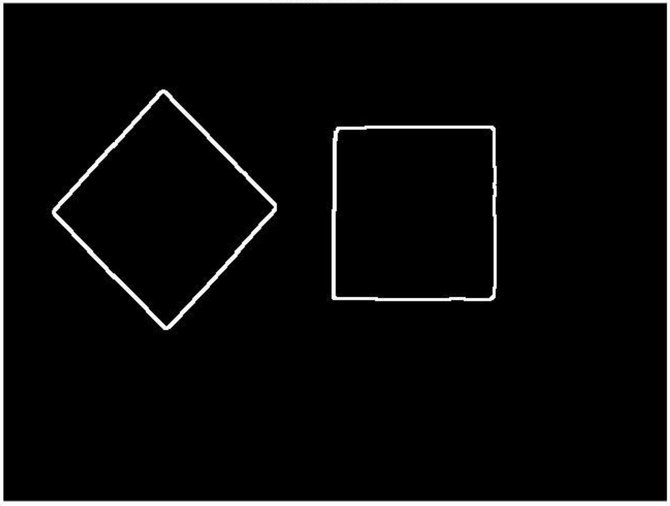
Task 8.3.3 – Hough transform for real image

```
image = imread("lab112.png");

figure(1);
subplot(1,2,1);
imshow(image);
title('Input image')
subplot(1,2,2);
edge1 = edge(image,'canny',0.56);
se = strel('square',3);
edge1 = imdilate(edge1,se);
imshow(edge1);
title('canny, thresh:0.56');

figure(2);
subplot(1,2,1);
[H,T,R] = hough(edge1);
imshow(H,[]);
title('Hough space')
subplot(1,2,2);
P = houghpeaks(H,8);
imshow(H,[]);
hold on;
plot(P(:,2),P(:,1),'o');
title('Hough peaks')

lines = houghlines(edge1,T,R,P,'FillGap',5,'MinLength',7);
figure, imshow(edge1), 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(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
      max_len = len;
      xy_long = xy;
   end
end
title('Hough lines')

figure();
subplot(1,2,1);
image = imread("house.png");
edge1 = edge(image,'canny',0.65);
imshow(edge1);
title('Edges');
subplot(1,2,2);
[H,T,R] = hough(edge1);
imshow(H,[]);
hold on;
P = houghpeaks(H,8);
plot(P(:,2),P(:,1),'o');
title('Peaks on hough space');
```

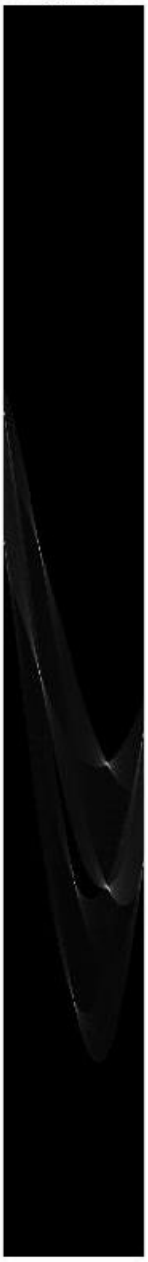
Input image



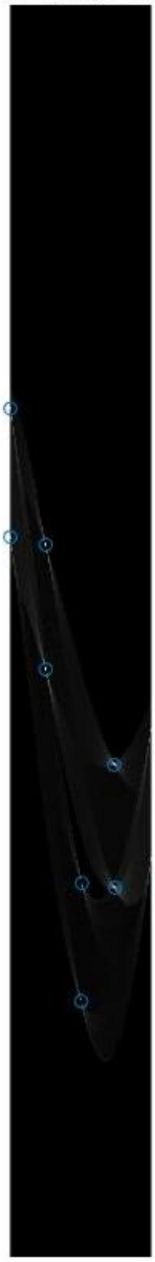
canny, thresh:0.56



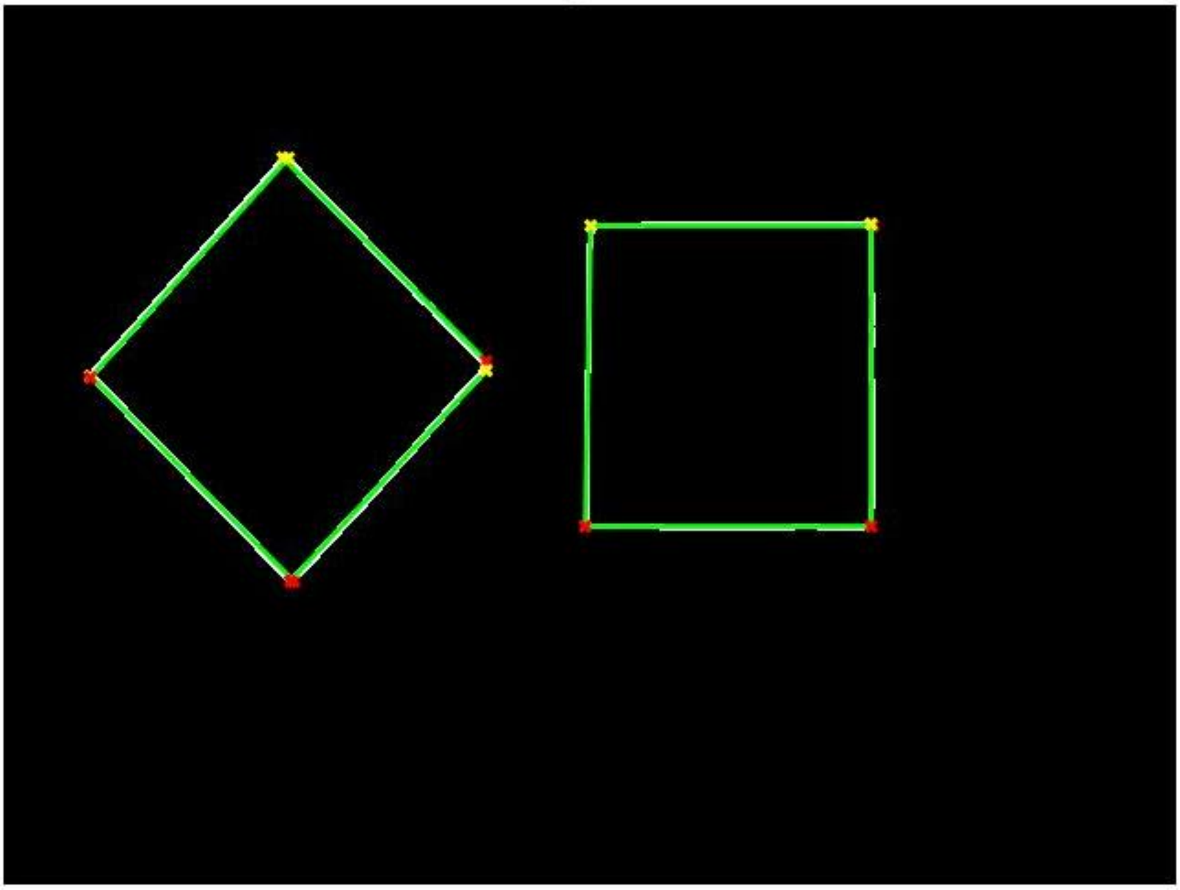
Hough space



Hough peaks

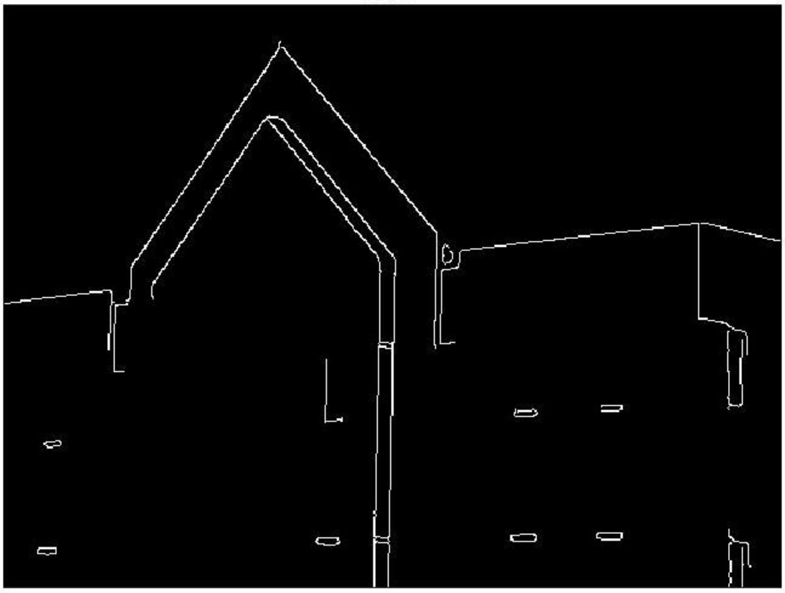


Hough lines



Peaks on hough space

Edges



2. Conclusions:

The LoG method finds the main edges of the entire house and returns a little of noice where are bricks.
With default settings, the zerocross method with log filter gives much more edge of house bricks and roof tiles.
The result with canny method in proper threshold gives a mostly clean image of main edges of building.
Comparing three methods of detecting edges, in my opinion, the best result is obtained by canny method with a specific threshold as a two-element array. Only this method can be specified by 2 element threshold.
If the filter kernel h is not provided to the zerocross method, the Laplacian of Gaussian kernel is used by default, so both methods will give the same result.

Every curve in hough space is represented by one point in the 11x11 image. If there is an intersection of two curves and it is called maximum.
Coordinates of maximum can be transformed to a straight line passing through two points whose curves intersected in the hough space.

With more complicated images the coordinations of maxima can be obtained by houghpeaks function.

The real image lab112.png required additional operation(dilation) to increase the thickness of square edges. This operation corrects the final result of the hough transform. The result of line detections is satisfying.