Project 5 solution

1. Figure of the original image

Figure of the LoG filter(σ = 3.5, n = 21)

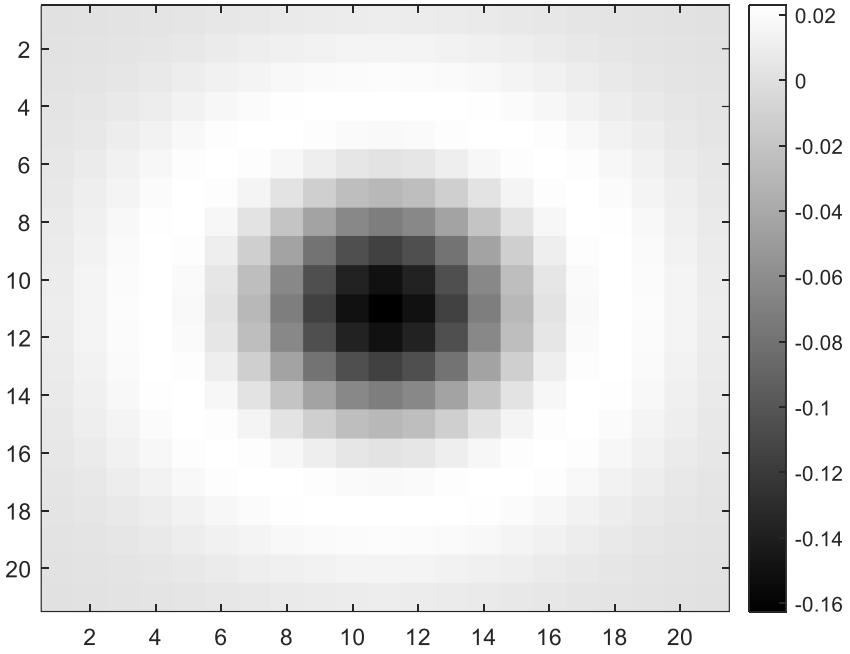


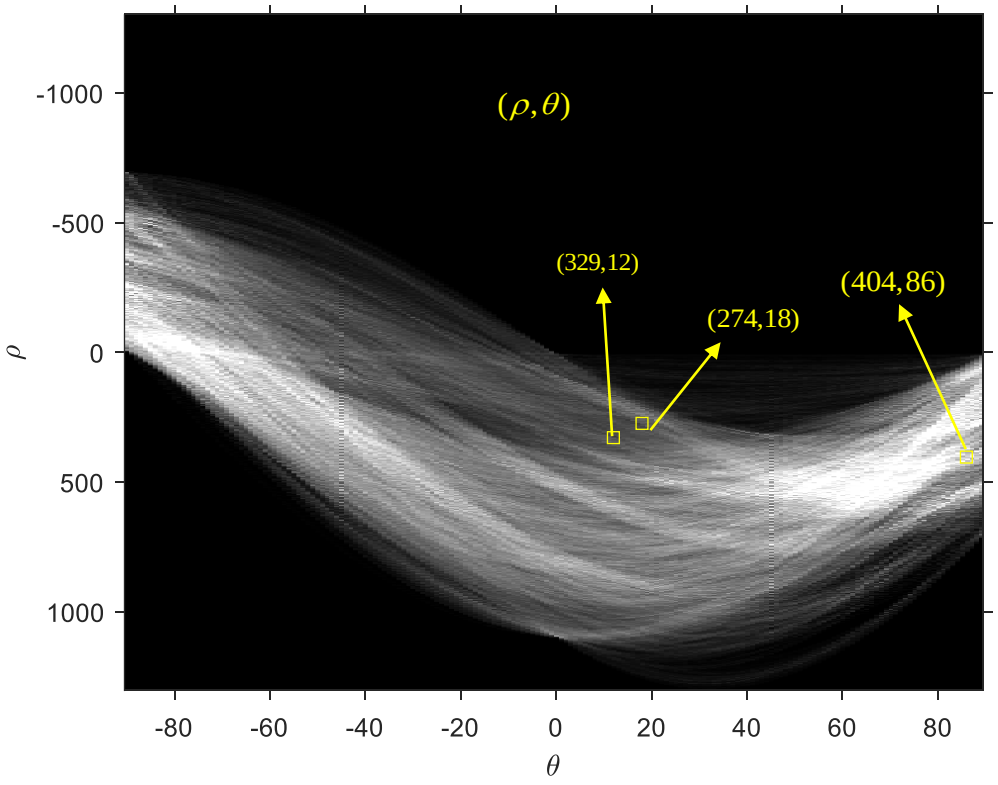
Figure of the LoG image

Binary images by zero-crossings with threshold of 0% of max(LoG)

Binary images by zero-crossings with threshold of 4% of max(LoG)

2. Figure of *Hough parameter space* and possible cells for license plate

(ρ , θ) = (404, 86), (329, 12), (274, 18)



3.

Figure of linked edges alone

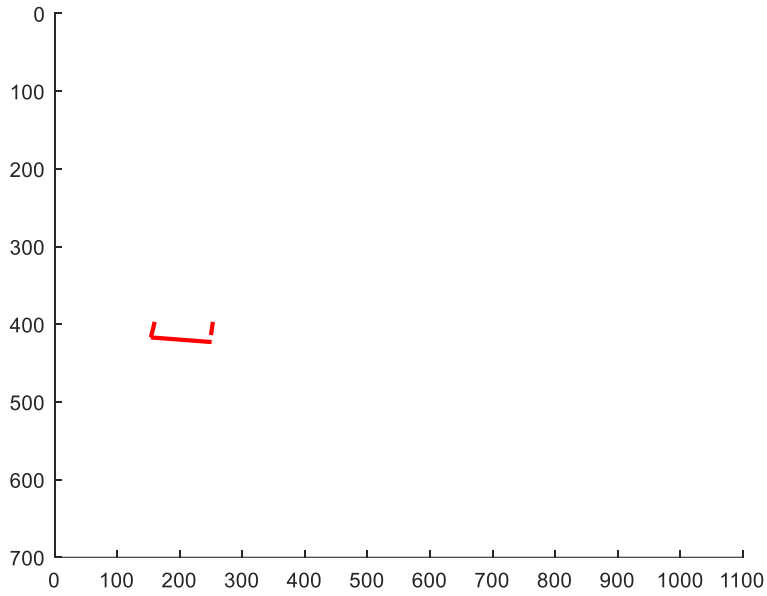


Figure of linked edges overlapped on the original image

4. source code

```
clear all
close all
%% Laplacian of Gaussian   (1/(sigma^4))*((y*y+x*x)-2*(sigma*sigma))*
img_ori=imread('Car On Mountain Road.tif');
img_ori = im2double(img_ori);
```

```
M = size(img_ori,1);
N = size(img_ori,2);

n =21; sigma = 3.5; % n = 6*sigma + 1 (odd number)
h = zeros(n,n);
h_n = (n-1)/2;

[x,y] = meshgrid(-h_n:h_n, -h_n:h_n);


% The two parts of the LoG equation
a = (x .^ 2 + y .^ 2 - 2 * sigma ^ 2) / sigma ^ 4;
b = exp( - (x .^ 2 + y .^ 2) / (2 * sigma ^ 2) );
%b = b / sum(b(:));
% The LoG filter
LoG = a .* b;

% The normalized LoG filter
nLoG = LoG - mean2(LoG);
figure(1)
image(nLoG,'CDataMapping','scaled')
colormap(gray)
colorbar


%% zero padding and sptial filtering

zero_img = zeros(M+floor(n/2)*2,N+floor(n/2)*2);
zero_img(floor(n/2)+1:floor(n/2)+M,floor(n/2)+1:floor(n/2)+N) =
img_ori;
for i = floor(n/2)+1:floor(n/2)+M
    for j = floor(n/2)+1:floor(n/2)+N
        img(i-floor(n/2), j-floor(n/2)) = sum(sum(zero_img(i-
floor(n/2):i+floor(n/2), j-floor(n/2):j+floor(n/2)) .* nLoG));
    end
end

%img = imfilter(255*(img_ori),nLoG,'replicate');
```

```
figure(2);
out = img - min(min(img)); out = 255 * (out / max(max(out)));
imshow(uint8(out));

%% check zero-crossing
[f, k] = size(img);
fk = zeros(f,k);

fk_zct_0 = zeros(f,k);
fk_zct = zeros(f,k);

threshold = 0;
threshold1 = 0.04;
for i=2:f-1
    for j=2:k-1

       if (img(i,j+1)>=0 && img(i,j-1)<0) || (img(i,j+1)<0 &&
img(i,j-1)>=0)
           fk(i,j)= img(i,j);
           fk_zct_0(i,j) = abs(img(i,j+1)-img(i,j-
1))>max(max(img))*threshold;
           fk_zct(i,j) = abs(img(i,j+1)-img(i,j-
1))>max(max(img))*threshold1;
       elseif (img(i+1,j)>=0 && img(i-1,j)<0) || (img(i+1,j)<0 &&
img(i-1,j)>=0)
           fk(i,j)= img(i,j);
           fk_zct_0(i,j) = abs(img(i+1,j)-img(i-
1,j))>max(max(img))*threshold;
           fk_zct(i,j) = abs(img(i+1,j)-img(i-
1,j))>max(max(img))*threshold1;
       elseif (img(i+1,j+1)>=0 && img(i-1,j-1)<0) || (img(i+1,j+1)<0
&& img(i-1,j-1)>=0)
           fk(i,j)= img(i,j);
           fk_zct_0(i,j) = abs(img(i+1,j+1)-img(i-1,j-
1))>max(max(img))*threshold;
           fk_zct(i,j) = abs(img(i+1,j+1)-img(i-1,j-
1))>max(max(img))*threshold1;;
```

```
       elseif (img(i-1,j+1)>=0 && img(i+1,j-1)<0) || (img(i-1,j+1)<0
&& img(i+1,j-1)>=0)
           fk(i,j)=img(i,j);
           fk_zct_0(i,j) = abs(img(i-1,j+1)-img(i+1,j-
1))>max(max(img))*threshold;
           fk_zct(i,j) = abs(img(i-1,j+1)-img(i+1,j-
1))>max(max(img))*threshold1;
       end
    end
end
figure(3); imshow(fk_zct_0);
figure(4); imshow(fk_zct);

fk_zct = logical(fk_zct)
%% Hough trandform

%using 4% max(LoG) as the threshold
[H,T,R] = hough(fk_zct,'RhoResolution',1,'Theta',-90:1:89);

%% find possible cells for license plate in Hough transform
%{
%% Using houghpeaks function and find possible cells for license
plate by try and error
P = houghpeaks(H,300,'threshold',ceil(0.2*max(H(:))));
lines = houghlines(fk_zct,T,R,P,'FillGap',5,'MinLength',7);
figure, imshow(img_ori), 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','red');

   % 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
```

```
% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',3,'Color','red');
%}
P=[900 1633 1578;5 103 109].';
figure(5);
%imshow(H,[],'XData',T,'YData',R,'InitialMagnification','fit');
imshow(imadjust(rescale(H)),'XData',T,'YData',R,'InitialMagnification
','fit');

xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;
set(findobj(get(gca,'Children'),'LineWidth',0.5),'LineWidth',10);
plot(T(P(:,2)),R(P(:,1)),'s','color','yellow');


% extract line segments based on Hough transform
lines = houghlines(fk_zct_0,T,R,P,'FillGap',5,'MinLength',5);
lines1 = houghlines(fk_zct_0,T,R,[900 5],'FillGap',5,'MinLength',5);
lines2 = houghlines(fk_zct_0,T,R,[1633
103],'FillGap',5,'MinLength',5);
lines3 = houghlines(fk_zct_0,T,R,[1578
109],'FillGap',5,'MinLength',5);

%% License plate
% only plot line segment

figure(6), hold on;
axis([0 1100 0 700]);
set(gca, 'ydir', 'reverse');

for k = 4
   xy = [lines1(k).point1; lines1(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
end

for k = 8
   xy = [lines2(k).point1; lines2(k).point2];
```

```
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
end
for k = 12
   xy = [lines3(k).point1; lines3(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
end

figure(7), imshow(img_ori), hold on;

for k = 4
   xy = [lines1(k).point1; lines1(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
end

for k = 8
   xy = [lines2(k).point1; lines2(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
end
for k = 12
   xy = [lines3(k).point1; lines3(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
end
```