# MAIN

%% read image

road = imread('road.png');

%% Pre-processing

%smooting, hessian detector, and non maximum supression

road\_smooth = gaus\_filt(double(road),5);

[road\_1d,road\_ori] = sobel\_filt(road\_smooth,0);

hessian\_image = hessian\_det(road\_smooth, 400);

nms\_image = non\_max\_sup\_matrix(hessian\_image, 1);

nms\_image = binarize(nms\_image);

[c,r] = find(nms\_image);

interest\_point = [r,c];

%eliminate the interest point on trees and ground

a = find(0.73\*interest\_point(:,1)+interest\_point(:,2)<189|interest\_point(:,2)>280);

interest\_point1 = setdiff(interest\_point, interest\_point(a,:),'rows');

imshow(road);

hold on;

plot(interest\_point1(:,1),interest\_point1(:,2),'r\*');

pause;

%% RANSAC

%line1

[ransac\_line\_coe1, inlier1] = ransacline(interest\_point1,50,0.8);

x = linspace(0,548,400);

y1 = ransac\_line\_coe1(1)+ransac\_line\_coe1(2)\*x;

%line2

interest\_point2 = setdiff(interest\_point1, inlier1,'rows');

[ransac\_line\_coe2, inlier2] = ransacline(interest\_point2,30,0.9);

y2 = ransac\_line\_coe2(1)+ransac\_line\_coe2(2)\*x;

%line3

interest\_point3 = setdiff(interest\_point2, inlier2,'rows');

[ransac\_line\_coe3, inlier3] = ransacline(interest\_point3,20,0.95);

y3 = ransac\_line\_coe3(1)+ransac\_line\_coe3(2)\*x;

%line4

interest\_point4 = setdiff(interest\_point3, inlier3,'rows');

[ransac\_line\_coe4, inlier4] = ransacline(interest\_point4,20,0.95);

y4 = ransac\_line\_coe4(1)+ransac\_line\_coe4(2)\*x;

inlier\_all = [inlier1;inlier2;inlier3;inlier4];

imshow(road);

hold on;

plot(x,y1,x,y2,x,y3,x,y4, 'LineWidth',1.5);

plot(inlier\_all(:,1),inlier\_all(:,2),'y\*');

pause;

%% Hough

H = hough\_trans(407,548,interest\_point1);

%get the local maximum interest point

hough\_max = sort(H(:),'descend');

hough\_max = hough\_max(1:20);

[r1,c1] = find(H==18);

y5 = hough\_inver(r1,c1);

[r2,c2] = find(H==16);

y6 = hough\_inver(r2(1),c2(1));

[r3,c3] = find(H==14);

y7 = hough\_inver(r3(3),c3(3));

[r4,c4] = find(H==13);

y8 = hough\_inver(r4(4),c4(4));

imshow(road);

hold on;

plot(x,y5,x,y6,x,y7,x,y8,'LineWidth',1.5);

# HESSIAN DETECTOR

function [hessian\_image] = hessian\_det(X,threshold)

[m,n] = size(X);

X = double(X);

hessian\_x = zeros(m,n);

hessian\_y = zeros(m,n);

hessian\_xx = zeros(m,n);

hessian\_yy = zeros(m,n);

hessian\_xy = zeros(m,n);

sobel\_filter\_x = [-1,0,1;-2,0,2;-1,0,1];

sobel\_filter\_y = [-1,-2,-1;0,0,0;1,2,1];

for i = 1:(m-2)

for j = 1:(n-2)

temp\_X = X(i:i+2,j:j+2);

hessian\_x(i,j) = sum(sum(sobel\_filter\_x .\* temp\_X));

hessian\_y(i,j) = sum(sum(sobel\_filter\_y .\* temp\_X));

end

end

for i = 1:(m-2)

for j = 1:(n-2)

temp\_XX = hessian\_x(i:i+2,j:j+2);

temp\_YY = hessian\_y(i:i+2,j:j+2);

hessian\_xx(i,j) = sum(sum(sobel\_filter\_x .\* temp\_XX));

hessian\_yy(i,j) = sum(sum(sobel\_filter\_y .\* temp\_YY));

hessian\_xy(i,j) = sum(sum(sobel\_filter\_y .\* temp\_XX));

end

end

hessian\_image = hessian\_xx.\*hessian\_yy -hessian\_xy.^2;

hessian\_image = hessian\_image.\*(hessian\_image > threshold);

end

# RANSAC

function [ransac\_line\_coe, inlier\_max] = ransacline(interest\_point, threshold,p)

N = 1;

sample\_count = 0;

s = RandStream('mlfg6331\_64');

e\_min = 1;

count = 0;

while N > sample\_count

sample = datasample(s,interest\_point,2,'Replace',false);

inlier\_num = 0;

inlier = [];

for i = 1 : size(interest\_point,1)

P = interest\_point(i,:)';

d = abs(det([sample(1,:)'-sample(2,:)',P-sample(1,:)']))/norm(sample(1,:)'-sample(2,:)');

if d <= threshold

inlier\_num = inlier\_num + 1;

inlier = [inlier;P'];

end

end

e = 1 - inlier\_num/size(interest\_point,1);

if e <= e\_min

e\_min = e;

inlier\_max = inlier;

N = round(log(1-p)/log(1-(1-e\_min)^2));

sample\_count = sample\_count+ 1;

end

count = count +1;

end

X = [ones(length(inlier\_max),1),inlier\_max(:,1)];

Y = inlier\_max(:,2);

ransac\_line\_coe = regress(Y,X);

end

# HOUGH

function H = hough\_trans(m,n,interest\_point)

H = zeros(m,n);

for i = 1 : length(interest\_point)

for j = 1:n

theta = j\*pi/n;

p = interest\_point(i,1)\*cos(theta) + interest\_point(i,2)\*sin(theta);

if p > m | p <= 0

continue

else

p = floor(p);

y = m-p;

H(y,j) = H(y,j) +1;

end

end

end

# HOUGH INVERSE

function y= hough\_inver(r,c)

x = linspace(0,548,400);

theta = c\*pi/548;

p = 407-r;

y = p/sin(theta)-x\*cot(theta);

end