EE 417 - Computer Vision
Post Lab Report

Data Generation for Camera Calibration

Name: Çiğdem Ceyda Düzgeç

Student ID: 23928

Date: 30.12.2020

Çiğdem Ceyda Düzgeç
23928

Harris Corners and Intersection Points

In our code we first needed to find Harris corners for this purpose we converted image to gray scale and applied Canny detector before using Hough Transform. When we use hough function we can get the theta and rho values. Then we used houghpeaks to locate peaks in the Hough transform matrix, we used threshold value here to get only the peaks we want to use. After that we created line which we will use later with houghlines function and we used theta and rho values here as parameters. Also, we have other parameters which are fillgap and minlenght. These parameters differentiate from image to image. Since we get the line we can plot our image. So in this step we plot lines with the line as green, the beginning point as yellow, ending point as red. Then we need to find two intersecting lines from the matrix of lines to hardcode them. After putting two lines' x and y values we need to get Thetas and Rhos. To do that we need to check if the beginning and points are in the line's matrix if they are we will get their theta and rho values. After finding theta and rho values of the lines we will use line equation which is x cos(theta) + y sin(theta) = rho for finding the intersection. Once we find line equations, we can solve both and find their intersections. For this purpose, we computed a matrix with cos and sin values and divide it to our rhos to solve them together. At this point we computed our intersections and plotted them but we will plot harris corners to compare with them. For this purpose we can use corners function.

Çiğdem Ceyda Düzgeç
23928

Appendix

a.lab5calibprep.m

```
clear all; close all; clc;

img = imread('calibrationObject.png');

[row, col, ch]=size(img);
if ch==3
    img_gray = rgb2gray(img);
else
    img_gray = img;
end

%% EDGE DETECTION
img_edge = edge(img_gray,'Canny');  % for post-lab use different
detectors
%img_edge = edge(img_gray,'log');
%imshow(img_edge);

%% HOUGH TRANSFORM - Extract Lines
[H,T,R] = hough(img_edge,'RhoResolution',0.8,'Theta',-
90:0.5:89); % try different parameters
P  = houghpeaks(H,60,'threshold',0.5*max(H(:)));
line = houghlines(img_edge,T,R,P,'FillGap',15,'MinLength',30);

%% PLOT HOUGHLINES

figure
subplot(1,2,1), imshow(img)    %original
subplot(1,2,2), imshow(img_gray) %grey image
hold on
for k = 1:length(line)
    xy = [line(k).point1; line(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',1,'Color','green'); % line
```

```
    plot(xy(1,1),xy(1,2),'x','MarkerSize',4,'Color','yellow');
% beginning point
    plot(xy(2,1),xy(2,2),'x','MarkerSize',4,'Color','red');    %
end point
    %len = norm(line(k).point1 - line(k).point2);  %you dont
need it
end

%% SELECT TWO INTERSECTING LINES MANUALLY

% Corner 1
 Line1_B = [41 131]; % beginning point
 Line1_E =[187 186];  % end point
 Line2_B = [48 83];
 Line2_E = [99 304];

%{
% Corner 2
 Line1_B = [41 131];
 Line1_E =[187 186];
 Line2_B = [82 125];
 Line2_E = [118 302];
% Corner 3
 Line1_B = [41 131];
 Line1_E =[187 186];
 Line2_B = [107 112];
 Line2_E = [143 319];
% Corner 4
 Line1_B = [42 161];
 Line1_E =[190 227];
 Line2_B = [141 133];
 Line2_E = [169 334];
% Corner 5
 Line1_B = [42 161];
 Line1_E =[195 264];
 Line2_B = [107 112];
 Line2_E = [143 319];
% Corner 6
 Line1_B = [42 161];
 Line1_E =[195 264];
 Line2_B = [82 125];
 Line2_E = [118 302];
% Corner 7
 Line1_B = [185 228];
 Line1_E =[316 131];
 Line2_B = [215 121];
 Line2_E = [228 317];
```

```
% Corner 8
 Line1_B = [185 228];
 Line1_E =[316 131];
 Line2_B = [246 175];
 Line2_E = [251 304];
%}
% Extract corresponding theta (T) and rho (R) values from the
output of 'houghlines' function
Rhos = [0;0];
Thetas = [0;0];

for k = 1:length(line)

    if(ismember(Line1_B, line(k).point1,
'rows')&&ismember(Line1_E, line(k).point2, 'rows'))
        Rhos(1,1) = line(k).rho ;
        Thetas(1,1) = line(k).theta ;
    end

    if(ismember(Line2_B,line(k).point1,
'rows')&&ismember(Line2_E,line(k).point2, 'rows'))
        Rhos(2,1) = line(k).rho ;
        Thetas(2,1) = line(k).theta ;
    end
end
%% PLOT INTERSECTING LINES

x_v = 0:size(img,1);
x_h = 0:size(img,2);
y_v = (Rhos(1,1) - x_v* cosd(Thetas(1,1)) )/ sind(Thetas(1,1));
y_h = (Rhos(2,1) - x_h* cosd(Thetas(2,1)) )/ sind(Thetas(2,1));

figure
imshow(img_gray)
hold on
plot(x_v,y_v,'Color','magenta');
plot(x_h,y_h,'Color','magenta');

%%  Solving the 2 line equations to find intersection point
(corner)
% A = [cosd(Thetas(1)) sind(Thetas(1))) ; cosd(Thetas(2))
sind(Thetas(2))];
A = [cosd(Thetas(1,1)) sind(Thetas(1,1)); cosd(Thetas(2,1))
sind(Thetas(2,1))];
C = A\Rhos; %Corner=inv(A)*Rhos

%% HARRIS CORNERS
```

```
corners = corner(img_edge, 'Harris');
plot(corners(:,1),corners(:,2),'s','MarkerSize',2,
'MarkerEdgeColor','blue','LineWidth',1);

%% PLOTTING CORNERS FOR COMPARISON
plot(C(1,1),C(2,1),'om', 'MarkerSize',4,'Color','magenta');
%hough, intersection
```