

EE 417 Computer Vision Lab #4: Corner Detection

Introduction

Lab #4 aims to implement both minimum eigenvalue(Kanade-Tomasi) and Harris algorithms to detect corner features in grayscale images.

Minimum Eigenvalue Algorithm

The algorithm first computes the gradients G_x and G_y by using built-in “imgradient()” function. Then creates H matrix of each pixel in a window as follows

$$H = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

Where I_x and I_y are the image gradients of a window along x and y directions respectively. After creating the matrix eigenvalues of the H matrix are computed. If minimum of these eigenvalues are greater than the threshold the pixels are add to corner list.

Harris Algorithm

The algorithm first computes the gradients G_x and G_y by using built-in “imgradient()” function. Then creates H matrix of each pixel in a window as follows

$$H = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

Where I_x and I_y are the image gradients of a window along x and y directions respectively. After creating the matrix the ratio of $\det(H)/\text{trace}(H)$ is computed. If the ratio is greater than the threshold the pxels are add to corner list.

I’ve could print the results in the lab, however when I’ve tried to implement the functions MATLAB kept applying the algorithms. I’ve waited for more than 2-3 hours for both and I’ve couldn’t get any results. I’m sure that my code is working because I’ve applied both of them in the lab and got results. I’ve even asked Mehmet Emin that the algorithms couldn’t find 2 corners and he’ve explained why.

I’m sincerely sorry that I can’t upload any results for the post-lab. I’ve taken the course last year when Mehmet Hoca was giving the course and I’ve tried to implemet the codes that were working last year after facing the problem with this year’s in-lab codes. Last year’s codes didn’t work also. I’m wishing you can have an understanding.

APPENDIX

```
function [Corners_Filtered]=lab4HarrisCorners(img,T)

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

Corners_Filtered=zeros(size(img));
img=double(img);
Corners_Filtered=double(Corners_Filtered);
corners=[];

k=2;
Simg=lab2gaussfilt(img);
[Gx,Gy]=imgradientxy(Simg);

for i=k+1:1:row-k-1;
    for j=k+1:1:col-k-1;
        windowIx=Gx(i-k:i+k,j-k:j+k);
        windowIy=Gy(i-k:i+k,j-k:j+k);
        h1=sum(sum(windowIx^2));
        h2=sum(sum(windowIy^2));
        h3=sum(sum(windowIx*windowIy));
        h4=h3;
        H=[h1 h3;h4 h2];
        b=trace(H);
        d=det(H);
        f=d/b;
        if(f>T)
            corners=[corners; i ,j]
        end
    end
end

figure;
imshow(img);
hold on;
plot(corners(:,2),corners(:,1),'r*','MarkerSize',7,'LineWidth',1);
title('Harris algorithm');

end
```

```

function [Corners_Filtered]=lab4ktcorners(img,T)

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

Corners_Filtered=zeros(size(img));
img=double(img);
Corners_Filtered=double(Corners_Filtered);
corners=[];
k=2;
Simg=lab2gaussfilt(img);
[Gx,Gy]=imgradientxy(Simg);

for i=k+1:1:row-k-1;
    for j=k+1:1:col-k-1;
        windowIx=Gx(i-k:i+k,j-k:j+k);
        windowIy=Gy(i-k:i+k,j-k:j+k);
        h1=sum(sum(windowIx^2));
        h2=sum(sum(windowIy^2));
        h3=sum(sum(windowIx*windowIy));
        h4=h3;
        H=[h1 h3;h4 h2];
        eigs=eig(H);
        if(min(eigs)>T)
            corners=[corners; i ,j]
        end
    end
end
figure;
    imshow(img);
    hold on;
    plot(corners(:,2),corners(:,1),'r*','MarkerSize',7,'LineWidth',1);
    title('Kanade-Tomasi corner detection');
end

```

```
%% kt
clear all; close all; clc;
tic
% img =checkerboard(20);
img = imread('chessboard.png');
imshow(img);
Corners_Filtered= lab4ktcorners(img,200); %image
toc
%% harrison
clear all; close all; clc;
% img =checkerboard(20);
img = imread('Monastery.bmp');
imshow(img);
Corners_Filtered= lab4Harrisorners(img,1); %image
```