

```

%%                                     Harris Corner Detection
%   V Khagesh Kumar
%   170102070
%   ECE
%   This program will take 2 minutes to compute.

clear;

original_image = imread('IITG.jpg');
image = double(rgb2gray(original_image));

%% Hyper parameters
alpha = 0.06; %constant for response functino R
gradient_x = [-1 0 1; -2 0 2; -1 0 1];
gradient_y = [-1 -2 -1; 0 0 0; 1 2 1];
f = 1; %3x3 gradient filter
window = 3;
W = ones(window,window); %consider uniform as of now
ff = floor(window/2);

%% Compute overhead
image = padarray(image,[f,f],'replicate');
[m,n,~] = size(original_image);
R = zeros(m,n); %Response Function
Mx = zeros(m,n);
My = zeros(m,n);

for i=1+f:m+f
    for j=1+f:n+f
        Mx(i-f,j-f) = sum(sum( gradient_x.*image(i-f:i+f,j-f:j+f) ));
        My(i-f,j-f) = sum(sum( gradient_y.*image(i-f:i+f,j-f:j+f) ));
    end
end

for i=1+ff:m-ff
    for j=1+ff:n-ff
        Ix = Mx(i-ff:i+ff,j-ff:j+ff);
        Iy = My(i-ff:i+ff,j-ff:j+ff);

        a_11 = sum(sum(W.*Ix.*Ix));
        a_12 = sum(sum(W.*Ix.*Iy));
        a_21 = a_12;
        a_22 = sum(sum(W.*Iy.*Iy));

        M = [a_11 a_12; a_21 a_22];
        R(i,j) = det(M) - alpha*trace(M)^2;
    end
end

%% Non-maximal Supression with Appropriate Threshold
R(R<1000000000) = 0; %Threshold

R_nonmax = R(2:m-1,2:n-1);
R_nonmax(R_nonmax<=R(1:m-2, 1:n-2) | ...

```

```

R_nonmax<=R(1:m-2, 2:n-1) | ...
R_nonmax<=R(1:m-2, 3:n ) | ...
R_nonmax<=R(3:m, 1:n-2) | ...
R_nonmax<=R(3:m, 2:n-1) | ...
R_nonmax<=R(3:m, 3:n ) | ...
R_nonmax<=R(2:m-1, 1:n-2) | ...
R_nonmax<=R(2:m-1, 3:n ) = 0;
final_image = R_nonmax>0;

final_image = padarray(final_image,[1,1]);
[X,Y] = find(final_image==1);
%imshow(original_image);
imshow(rgb2gray(original_image));
hold on;
plt = scatter(Y,X,10, '+', 'r');
title('Harris Corner Detection followed by Non-maximul Supression');

saveas(plt, 'IITG_1_gray.jpg');

```