
Table of Contents

HW 4 Jesse Layman SID: 861135479	1
Exercise 7.2.	1
8.6	10

HW 4 Jesse Layman SID: 861135479

```
% Professor: Bir Bhanu,,
% TA: Vincent On,
% EE 146 - 021
```

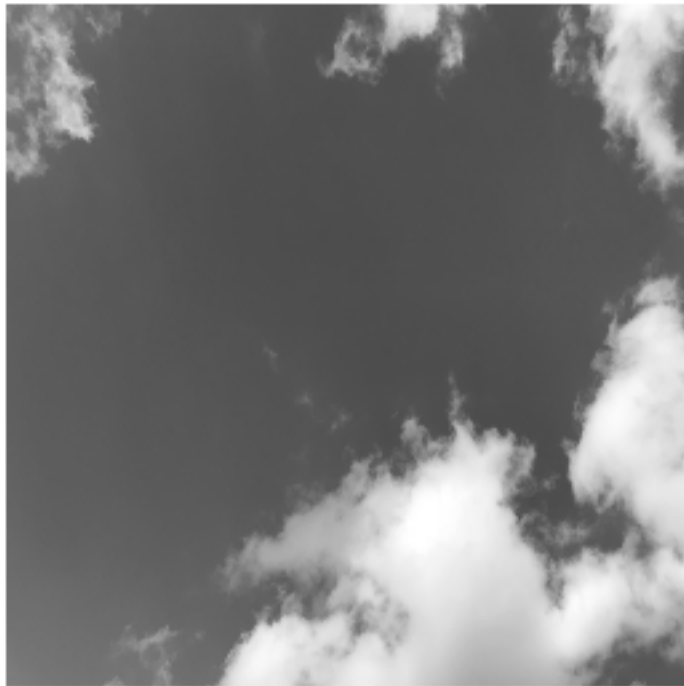
Exercise 7.2.

Conduct a series of experiments to determine how image contrast affects the performance of the Harris detector, and then develop an idea for how you might automatically determine the parameter t_H depending on image content.

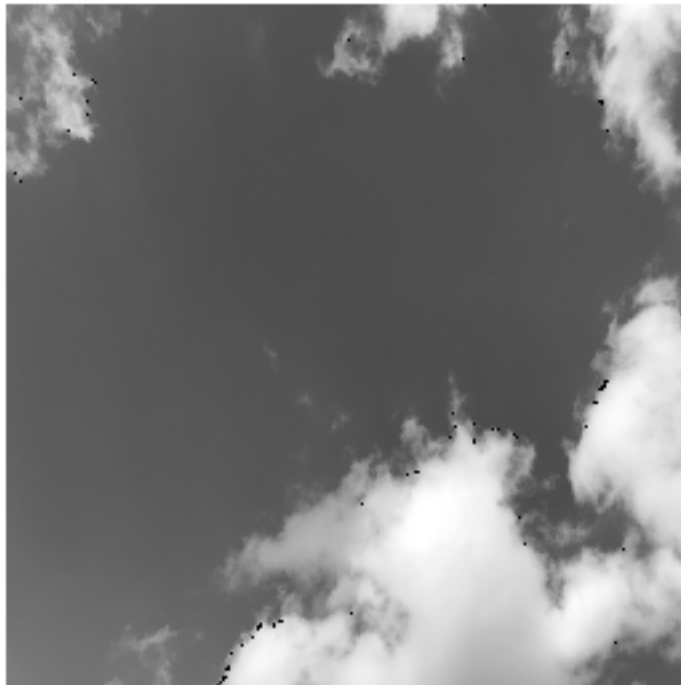
```
% Low contrast image
clear all
close all
H_gaus = fspecial('gaussian');
I_clouds = imread('/Users/jesselayman/Downloads/IMG_0690.JPG');
I_clouds = rgb2gray(I_clouds);
I_clouds = imresize(I_clouds,[512 512]);
I_cloudsd= double(I_clouds);
figure
imshow(I_clouds);

[Gx_c,Gy_c] = gradient(I_cloudsd);
Ac = Gx_c.^2;
Bc = Gy_c.^2;
Cc = Gx_c*Gy_c;
MA = conv2(Ac,H_gaus);
MB = conv2(Bc,H_gaus);
MC = conv2(Cc,H_gaus);
alpha = 0.08
for u=1:514
    for v = 1:514
        Q(u,v) = MA(u,v)* MB(u,v)- MC(u,v)^2 -alpha* [ MA(u ,v) +
        MB(u,v)]^2;
    end
end
figure
imshow(Q);
t = 11000;
for u=1:512
    for v = 1:512
        if Q(u,v) > t
```

```
        C(u,v) = 0;  
    else  
        C(u,v) = I_clouds(u,v);  
    end  
end  
end  
figure  
imshow(C)  
  
alpha =  
    0.0800
```







High contrast image

```
I_fence = imread('/Users/jesselayman/Downloads/IMG_0693_2.JPG');
I_fence = rgb2gray(I_fence);
I_fence = imresize(I_fence,[512 512]);
figure
imshow(I_fence);
I_fenced= double(I_fence);

[Gx_f,Gy_f] = gradient(I_fenced);
Af = Gx_f.^2;
Bf = Gy_f.^2;
Cf = Gx_f*Gy_f;
MAf = conv2(Af,H_gaus);
MBf= conv2(Bf,H_gaus);
MCf = conv2(Cf,H_gaus);
alphaf = 0.08
for uf=1:514
    for vf = 1:514
        Qf(uf,vf) = MAf(uf,vf)* MBf(uf,vf)- MCf(uf,vf)^2 -alpha*
        [ MAf(uf ,vf) + MBf(uf,vf)]^2;
    end
end
figure
imshow(Qf);
tf = 100000;
```

```

for uf=1:512
    for vf = 1:512
        if Qf(uf,vf) > tf
            Cf(uf,vf) = 0;
        else
            Cf(uf,vf) = I_fence(uf,vf);
        end
    end
end
figure
imshow(Cf)
ITEST = HarrisCorners(I_fence,0.04,1000000,0.01);
figure
imshow(ITEST)
function Harris = HarrisCorners(I ,alpha,th ,dmin)
H_gaus = fspecial('gaussian');
Id= double(I);
[Gx,Gy] = gradient(Id);
A = Gx.^2;
B = Gy.^2;
C = Gx*Gy;
MA = conv2(A,H_gaus);
MB= conv2(B,H_gaus);
MC = conv2(C,H_gaus);
    for u=1:size(I)
        for v = 1:size(I)
            Q(u,v) = MA(u,v)* MB(u,v)- MC(u,v)^2 -alpha* (MA(u,v) +
MB(u,v))^2;
        end
    end
    for u=1:size(I)
        for v = 1:size(I)
            if Q(u,v)>th || IsLocalMax(Q,u,v)
                Corner(u,v) = Q(u,v);
            end
        end
    end
    Harris = CleanUpCorners(Corner,dmin);
end

function B = IsLocalMax(Q, u, v)
N = GetNeighbors(Q, u, v);
    if Q(u, v) > max(N)
        B = 1;
    else B = 0;
    end
end

function N = GetNeighbors(Q,u,v)
if (u-1)==0
    u = u+1;
end
if (v-1) ==0

```

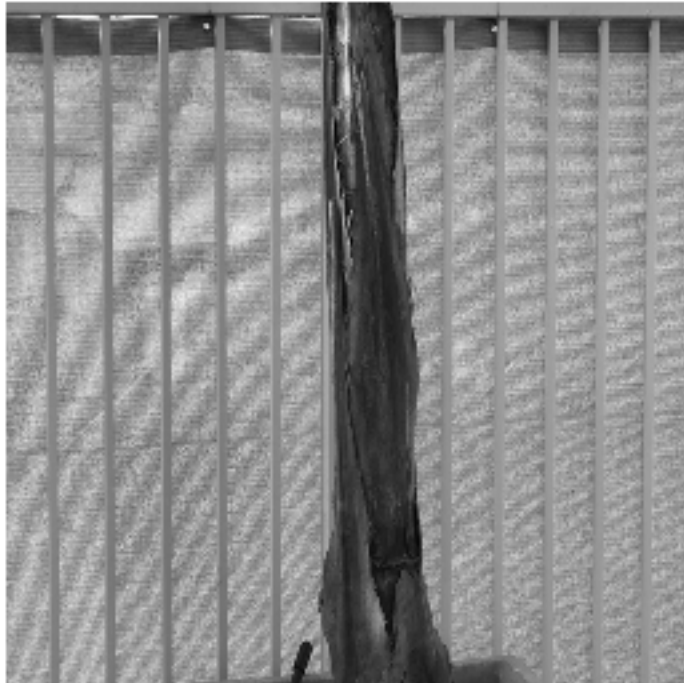
```

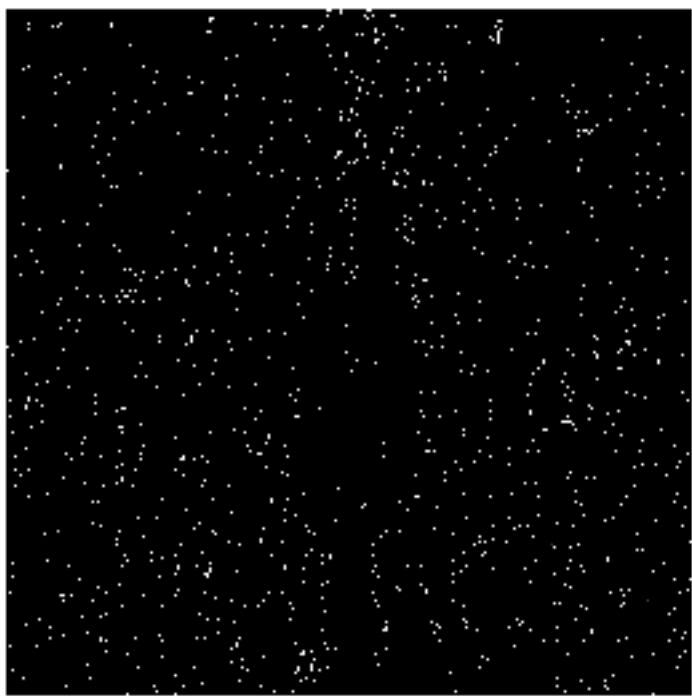
        v = v+1;
    end
    if (u+1) >size(Q)/2
        u = u-1;
    end
    if (v+1) >size(Q)/2
        v = v-1;
    end
    N = [Q(u+1,v),Q(u+1,v-1),Q(u,v-1),Q(u-1,v-1),
        Q(u-1, v), Q(u-1, v+1), Q(u, v+1), Q(u+1, v+1)];
    end

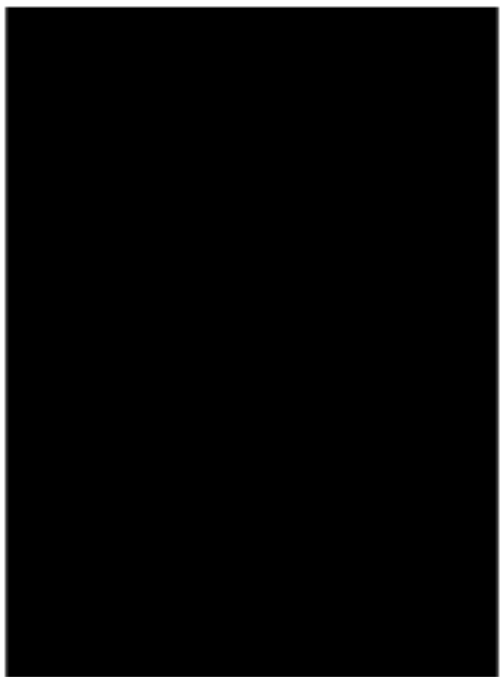
    function Cornerclean = CleanUpCorners(C, dmin)
    C = sort(C,'descend');
    cSize = size(C);
    Cornerclean = zeros(cSize(1),cSize(2));
    while max(C)>0
        i=1;
        c0 = C(i);
        C(i) = [];
        Cornerclean(i) = c0;
        for r = 1:cSize(1)
            for j = 1:cSize(2)
                if dist(c0,C(j)) < dmin
                    C(j) = [];
                end
            end
        end
        i = i+1;
    end

    alphaf =
        0.0800

```







8.6

Published with MATLAB® R2017b