Mingchen Ma 998964072

1.
a)
b)
MATLAB CODE

```
function harrisCorner
gray_building = double(rgb2gray(imread('building.jpg'));
[Ix, Iy] = imgradientxy(gray_building);
[cols, rows] = size(gray_building);
sigma = 3;
Ix_2 = Ix.^ 2;
Iy_2 = Iy.^ 2;
Ix_Iy = Ix.* Iy;
Gauss_x2 = imgaussfilt(Ix_2, sigma);
Gauss_y2 = imgaussfilt(Iy_2, sigma);
Gauss_xy = imgaussfilt(Ix_Iy, sigma);
threshold = 20000;
A = zeros(cols, rows);
for (col = 1:cols)
    for (row = 1:rows)
        M = [Gauss_x2(col, row) Gauss_xy(col, row); Gauss_xy(col, row) Gauss_y2(col, row)];
        R = det(M)/trace(M);

        A(col, row) = R;
    end
end

r = 4;


domain = fspecial('disk',  r) > 0;
output = ordfilt2(A, sum(sum( domain)),domain);

C = (A == output)&( A > threshold);

[x,y] = find(C);
imshow('building.jpg');
hold on
plot(y, x, 'or');
```

Output Image:
Explanation for 2b
So as r is increasing, the range of Non-maximum suppression is increasing so that the corner detected is decreasing

c)
d)
I didn't quite finish c and d… sorry

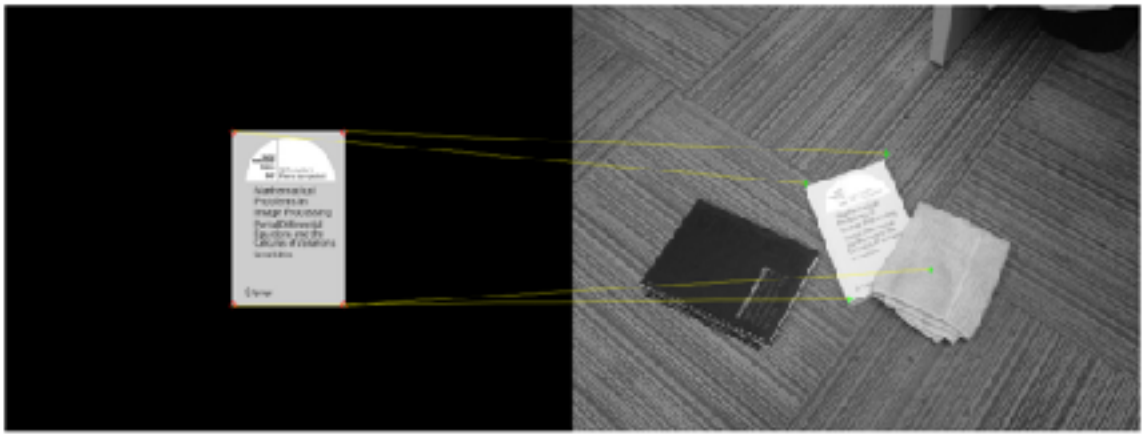
2.
a) b) c) d)

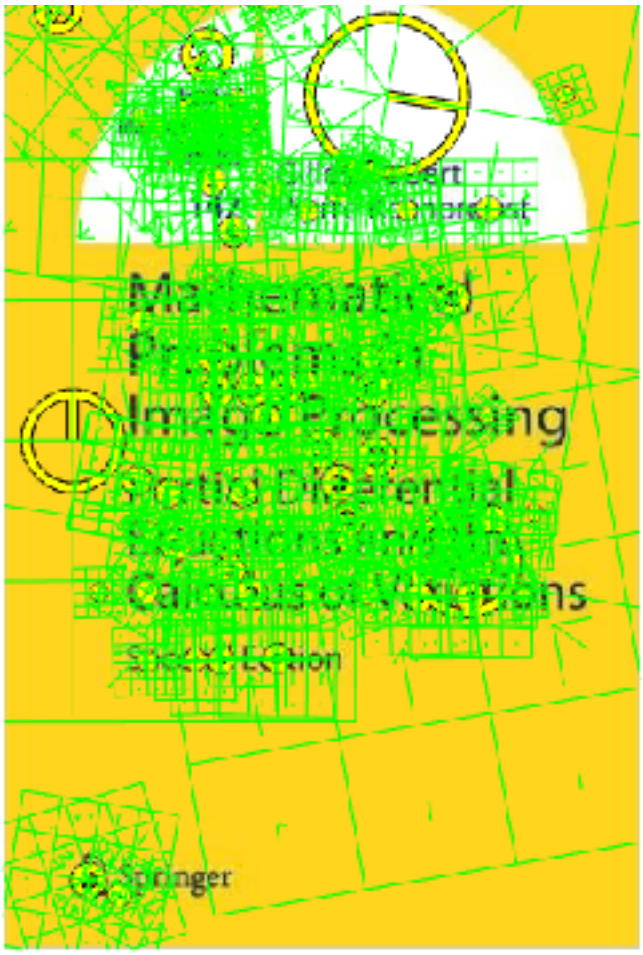
```
function [matches, scores] = ExtractSIFT
%book
img = imread('book.jpg');
figure; imshow(img)

img = single(rgb2gray(img)) ;
[fa,da] = vl_sift(img);
perm = randperm(size(fa,2)) ;
sel = perm(1:100) ;
h1 = vl_plotframe(fa(:,sel)) ;
h2 = vl_plotframe(fa(:,sel)) ;
set(h1,'color','k','linewidth',3) ;
set(h2,'color','y','linewidth',2) ;

h3 = vl_plotsiftdescriptor(da(:,sel),fa(:,sel)) ;
set(h3,'color','g') ;

%findBook
find = imread('findBook.JPG');
```

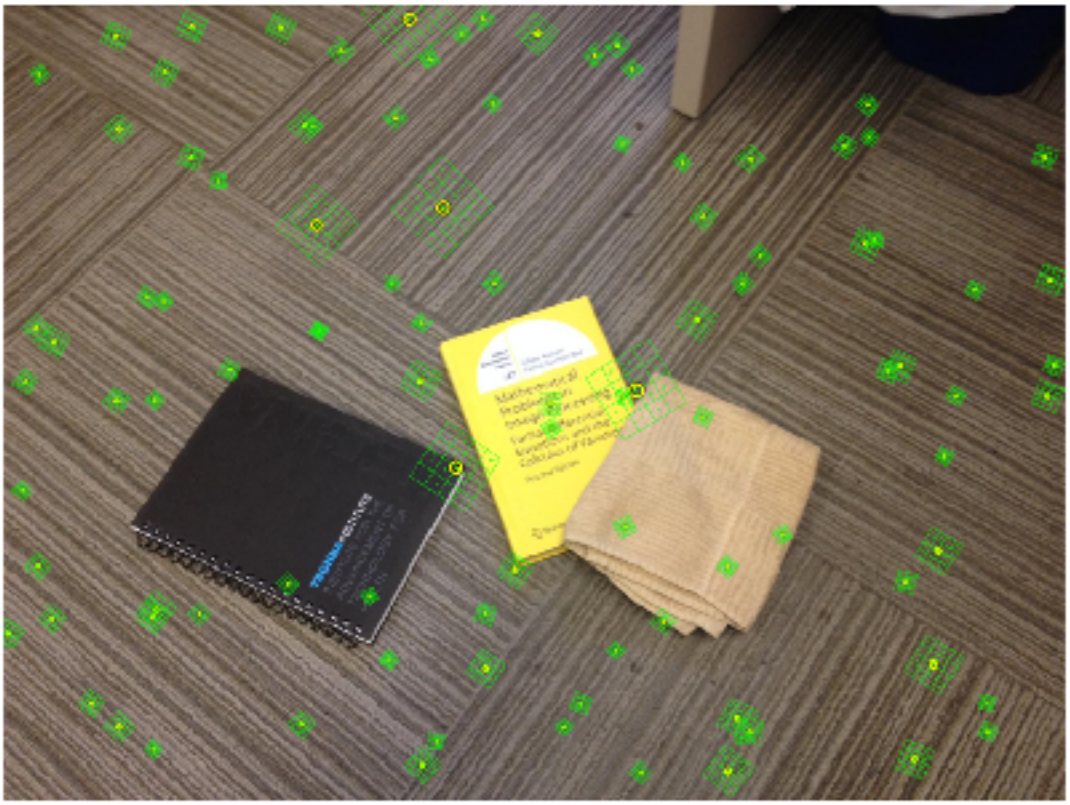
```
figure; imshow(find)

find = single(rgb2gray(find)) ;
[fb,db] = vl_sift(find);
perm = randperm(size(fb,2)) ;
sel = perm(1:100) ;
h1 = vl_plotframe(fb(:,sel)) ;
h2 = vl_plotframe(fb(:,sel)) ;
set(h1,'color','k','linewidth',3) ;
set(h2,'color','y','linewidth',2) ;

h3 = vl_plotsiftdescriptor(db(:,sel),fb(:,sel)) ;
set(h3,'color','g') ;
threshold = 0.8;


distance = pdist2(transpose(da), transpose(db));
min = zeros(2, size(da, 2));
min2 = zeros(2, size(da, 2));

for i=1:size(da, 2)
    [Mi, index] = sort(distance(i, :));

    min(:, i) = Mi(1:2);
    min2(:, i) = index(1:2);
end

can = find((min(:, 1)/min(:,2)) < threshold);




figure; ax = axes;
showMatchedFeatures(img,find,transpose(fa(1:2, can)), transpose(fb(1:2, min2(1, can))),
'montage','Parent',ax);
title(ax, 'book matches');


k = 10;
[~, idx] = sort(min(1, :));

p = [];
p_prime = [];


for i=1:k
    x_1 = fa(1, idx(i));
    y_1 = fa(2, idx(i));

    x_2 = fb(1, min2(1, idx(i)));
    y_2 = fb(2, min2(1, idx(i)));

    p_1 = [x_1,y_1,0,0,1,0;
           0,0,x_1,y_1,0,1]';
```

```
    p_2 = [x_2,y_2];
    p = [p,p_1];
    p_prime = [p_prime, p_2];
end
p = transpose(p);
p_prime = transpose(p_prime);

% compute affine transformation
affine = inv(transpose(p)*p)*transpose(p)*p_prime;

corner = [1,1;
      1,499;
      320,1;
      320,499];

find_matrix = [1 1 0 0 1 0;       % top left corner
               0 0 1 1 0 1;
               1 499 0 0 1 0;
               0 0 1 499 0 1;
               320 1 0 0 1 0;
               0 0 320 1 0 1;
               320 499 0 0 1 0;
               0 0 320 499 0 1
               ];

transformation = find_matrix*affine;
transformation = transpose(transformation);

corner_2 = [transformation(1:2);
             transformation(3:4);
             transformation(5:6);
             transformation(7:8)
            ];


showMatchedFeatures(img,find,corner, corner_2, 'montage','Parent',ax);
```

OUTPUT IMAGE




Springer



3.
a)
```
function [G] = gaussian(s, k)
   x = linspace(-k, k, 20*k);
   G = exp(-(x.^2) / (2*s^ 2));
   G = G* (1/((2*pi)^0.5 *s));
```
b)
```
function [LoG] = laplacian(s, k)
   x = linspace(-k, k, 20*k);
   LoG = -1/s^4 * (1 - x.^2/(2*s^2)).*exp(-x.^2 / (2*s^2));
```

c)
```
G1 = gaussian(7, 20);
G2 = gaussian(6, 20);

DoG = G1 - G2;
```

Mingchen Ma 998964072

```
LoG = laplacian(7, 20);

x = linspace(-20, 20, 400);
plot(x, DoG, x, LoG);
```