Mingchen Ma 998964072

1.

```
function [length,width] = shoe_measure
shoe_dollar = imread('shoe.jpg');
imshow(shoe_dollar);
hold on;
dollar_x = [774; 1166; 1070; 1368];
dollar_y = [984; 1132; 512; 584];
Rect_x = [0; 69.85; 0; 69.85];
Rect_y = [0; 0; 152; 152];
x1 = dollar_x(1,1);
x2 = dollar_x(2,1);
x3 = dollar_x(3,1);
x4 = dollar_x(4,1);
y1 = dollar_y(1,1);
y2 = dollar_y(2,1);
y3 = dollar_y(3,1);
y4 = dollar_y(4,1);
x1_ = Rect_x(1,1);
x2_ = Rect_x(2,1);
x3_ = Rect_x(3,1);
x4_ = Rect_x(4,1);
y1_ = Rect_y(1,1);
y2_ = Rect_y(2,1);
y3_ = Rect_y(3,1);
y4_ = Rect_y(4,1);
% Since we need at least 4 point thus, construct A as follow
A = [x1 y1 1 0 0 0 -x1_*x1 -x1_*y1 -x1_;
    0 0 0 x1 y1 1 -y1_*x1 -y1_*y1 -y1;
    x2 y2 1 0 0 0 -x2_*x2 -x2_*y2 -x2_;
    0 0 0 x2 y2 1 -y2_*x2 -y2_*y2 -y2;
    x3 y3 1 0 0 0 -x3_*x3 -x3_*y3 -x3_;
    0 0 0 x3 y3 1 -y3_*x3 -y3_*y3 -y3;
    x4 y4 1 0 0 0 -x4_*x4 -x4_*y4 -x4_;
    0 0 0 x4 y4 1 -y4_*x4 -y4_*y4 -y4;
   ];
% Found this way to extract the homography H from the A online google
[u,s,v] = svd(A);
H = v(:,9);
% These are the reference point i use for my shoe measurement estimation
x__ = 66;
y__ = 762;
x____= 826;
y____= 112;
x__1 = 186;
y__1 = 532;
x____1= 426;
y____1= 690;
% [a,b] = getpts
scatter(66,826);
scatter(788,112);
scatter(186,532);
scatter(426,690);
% According to the slides, caculate the x1 y1 prime and x2 y2 prime
% caculate the x3 y3 prime and x4 y4 prime then caculate the distance in between
```

```
x1_prime = (H(1,1)*x__ + H(2,1)*y__ + H(3,1))/(H(7,1)*x__ + H(8,1)*y__ + H(9,1));
y1_prime = (H(4,1)*x__ + H(5,1)*y__ + H(6,1))/(H(7,1)*x__ + H(8,1)*y__ + H(9,1));
x2_prime = (H(1,1)*x____ + H(2,1)*y____ + H(3,1))/(H(7,1)*x____ + H(8,1)*y____ + H(9,1));
y2_prime = (H(4,1)*x____ + H(5,1)*y____ + H(6,1))/(H(7,1)*x____ + H(8,1)*y____ + H(9,1));
x3_prime = (H(1,1)*x__1 + H(2,1)*y__1 + H(3,1))/(H(7,1)*x__1 + H(8,1)*y__1 + H(9,1));
y3_prime = (H(4,1)*x__1 + H(5,1)*y__1 + H(6,1))/(H(7,1)*x__1 + H(8,1)*y__1 + H(9,1));
x4_prime = (H(1,1)*x____1 + H(2,1)*y____1 + H(3,1))/(H(7,1)*x____1 + H(8,1)*y____1 + H(9,1));
y4_prime = (H(4,1)*x____1 + H(5,1)*y____1 + H(6,1))/(H(7,1)*x____1 + H(8,1)*y____1 + H(9,1));
length = sqrt((x1_prime - x2_prime)^2 + (y1_prime - y2_prime)^2);
width = sqrt((x3_prime - x4_prime)^2 + (y3_prime - y4_prime)^2);
% length of the shoe is 25cm
% width of the shoe is 9.8cm
end
```

The shoe length is about 25cm and width is about 9.8cm

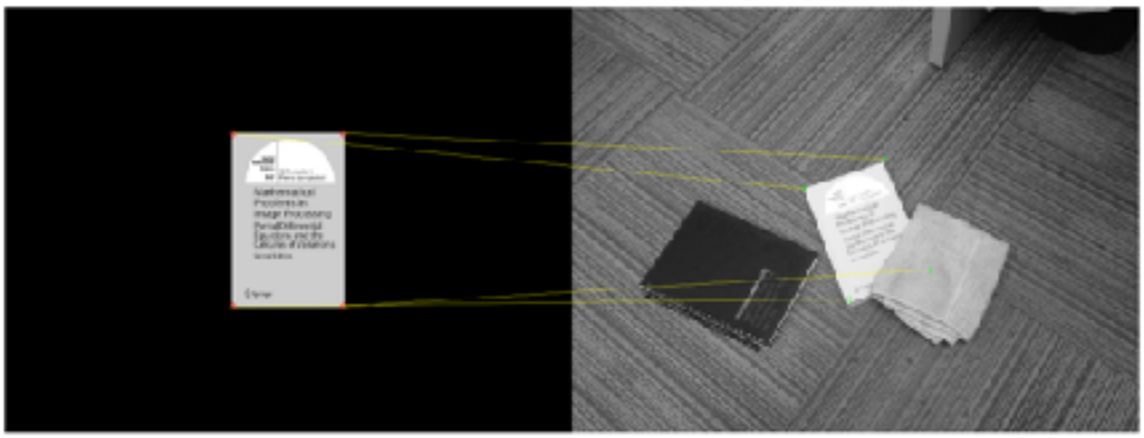
Output Image:

2a)

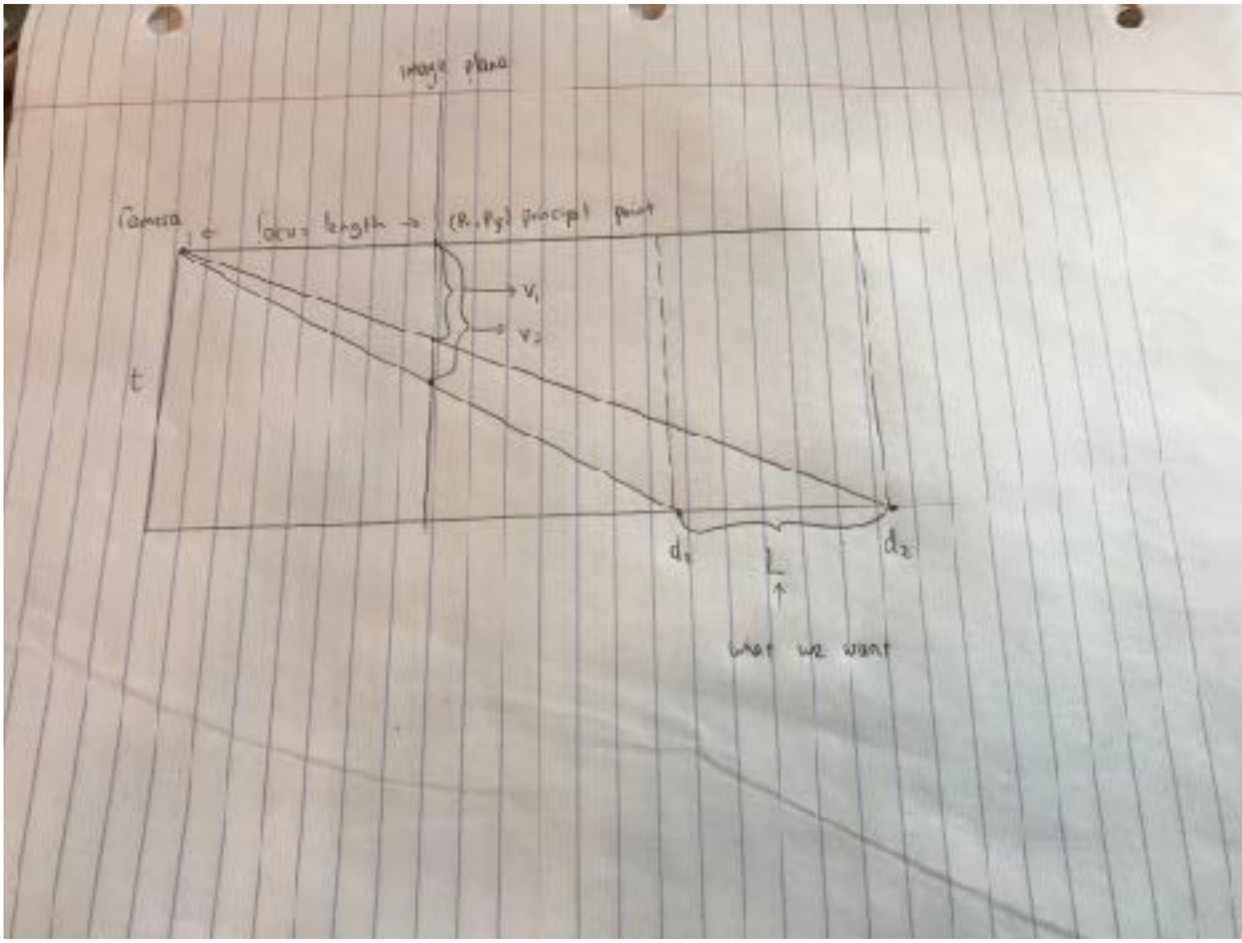
```
function SIFT %book
img = imread('book.jpg');
% figure; imshow(img)
img = single(rgb2gray(img)) ; [fa,da] = vl_sift(img);
perm = randperm(size(fa,2)) ; sel = perm(1:100) ;
% h1 = vl_plotframe(fa(:,sel)) ;
% h2 = vl_plotframe(fa(:,sel)) ; set(h1,'color','k','linewidth',3) ; set(h2,'color','y','linewidth',2) ;
% h3 = vl_plotsiftdescriptor(da(:,sel),fa(:,sel)) ; set(h3,'color','g') ;
%findBook
threshold = 0.7;
findBook = imread('findBook.JPG');

% figure; imshow(findBook)
findBook = single(rgb2gray(findBook)) ; [fb,db] = vl_sift(findBook);
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') ;
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,findBook,transpose(fa(1:2, can)), transpose(fb(1:2, min2(1, can))),
'montage','Parent',ax);

book_pts = transpose(fa(1:2, can));
find_book_pts = transpose(fb(1:2, min2(1, can)));
ransac(book_pts, find_book_pts);
end

function ransac(input_data,output_data)
%  figure;plot(input_data,output_data,'o');hold on;
 number = size(input_data, 1);
 bestInLier = 0;
 bestA = [];
 for i = 1:10

     id = randperm(number,3);
     id1 = id(:,1);
     id2 = id(:,2);
     id3 = id(:,3);
%      A_x = zeros(3,2);
%      A_y = zeros(3,2);
     x1 = input_data(id1, 1);
     y1 = input_data(id1, 2);
     x1_ = output_data(id1, 1);
     y1_ = output_data(id1, 2);
     x2 = input_data(id2, 1);
```

```
     y2 = input_data(id2, 2);
     x2_ = output_data(id2, 1);
     y2_ = output_data(id2, 2);
     x3 = input_data(id3, 1);
     y3 = input_data(id3, 2);
     x3_ = output_data(id3, 1);
     y3_ = output_data(id3, 2);
     p = [x1 y1 0 0 1 0;
         0 0 x1 y1 0 1;
         x2 y2 0 0 1 0;
         0 0 x2 y2 0 1;
         x3 y3 0 0 1 0;
         0 0 x3 y3 0 1;
     ];
    p_ = [x1_ y1_ x2_ y2_ x3_ y3_]';
    A = (inv(p'*p)) * p' * p_;
%     disp(A);
%     if it's invertible, anotherwise I dont know
    inlier = 0;
    for n = 1: size(input_data,1)
        x__ = input_data(n, 1);
        y__ = input_data(n, 2);
        point = [output_data(n, 1);output_data(n, 2)];
        M = [x__ y__ 0 0 1 0;0 0 x__ y__ 0 1];
        Q = M * A;
        D = sum((point - Q).^2);
%         disp(D)
        if D < 7
            inlier = inlier + 1;
        end
    end
    if inlier > bestInLier
%         disp(1);
        bestInLier = inlier;
        bestA = A;
    end
 end
imshow('book.jpg');hold on;
% [n,m]= size(imread('book.jpg'));
% disp(m);
% disp(n);
% disp(n);
% disp(size(bestA));
p1 =[1 1 0 0 1 0;
    0 0 1 1 0 1]*bestA;
p2 = [1 499 0 0 1 0;
    0 0 1 499 0 1]*bestA;
p3 =[320 1 0 0 1 0;
    0 0 320 1 0 1]*bestA;
p4 = [320 499 0 0 1 0;
    0 0 320 499 0 1]*bestA;
% scatter(n,1);
% scatter(1,m);
% scatter(n,m);
```

```
p_2 = vertcat([p1, p2, p3, p4]);
disp(p_2);
figure; ax = axes;
showMatchedFeatures(single(rgb2gray(imread('book.jpg'))),single(rgb2gray(imread('findBook.J
PG'))),[1 1;1 499;320 1;320 499],p_2', 'montage','Parent',ax);
end
% scatter(p1);
% scatter(p2);
% scatter(p3);
% scatter(p4);

%  figure;plot(bestPoint1,bestPoint2,'o');hold on;
```



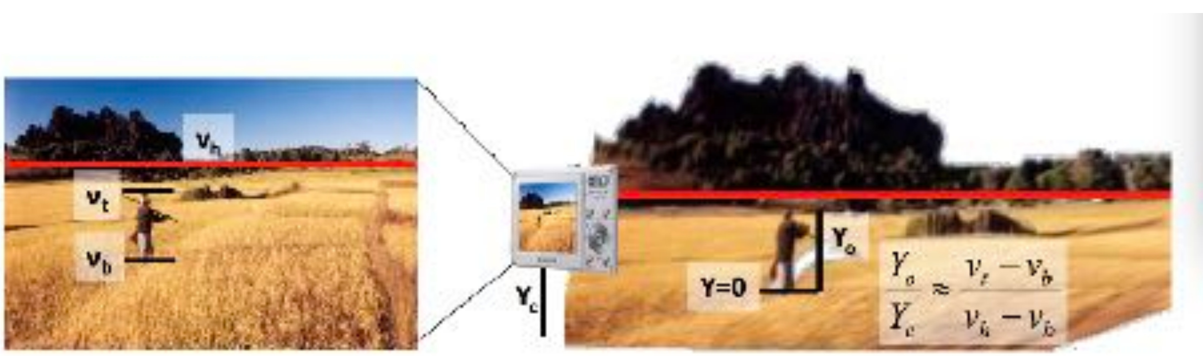
3.a)
Let L be the distance between adjacent railway ties
v1 is the length from camera centre to d1 that intersect the image plane
v2 is the length from camera centre to d2 that intersect the image plane
v1 and v2 are measurable using pixel coordinate on the image
Let f be the focus length that is extract from K
t is the distance from the ground to the camera that is given as well
d1 and d2 are the real position of the the adjacent railway ties according to v1 and v2
Due to similar triangle geometry we know that
d1/f = t/v1 Similarly for d2, d2/f = t/v2
thus,
L = d2 - d1
   = t*f/v1 - t*f/v2
As shown in the graph below to demonstrate



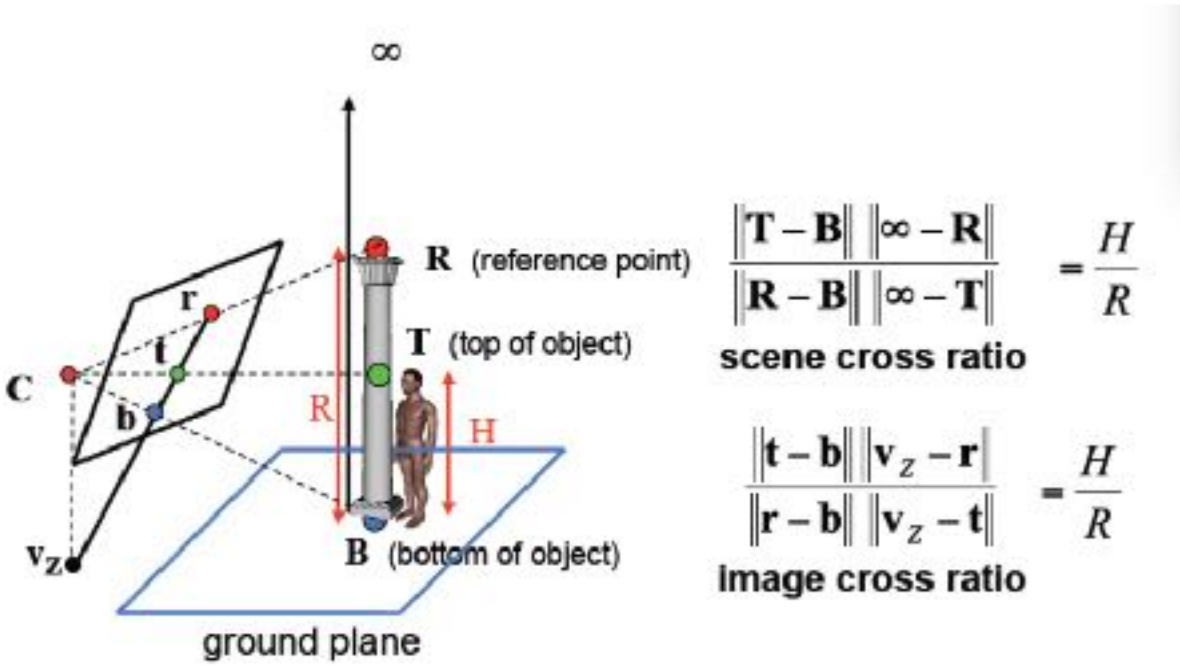
b)

So, according to https://courses.engr.illinois.edu/cs543/sp2011/materials/3dscene_book_svg.pdf
Page 6 and 7
I decide to use Yo/Yc = (vt-vb)/(vh-vb)
Where Yo is the height of the person in real world
Where Yc is height of the camera to the real ground
vh is the horizon that I get by my estimation , vt is the line that measure of the top of the person in the image
vb is the line that measure of the bottom of the person in the image

In this case we know that all the vertical vanish point is at infinity as shown in the graph below



The following graph is more explanatory, basically we are using the projective geometry, that the cross ratio of 4 collinear points in this case vz,b,t,r remain the same on the projective transformation
Therefore since vz is infinity as we know therefore we can use this formula derived from the cross ratio of 4 collinear points
There for Yo/Yc = (vt-vb)/(vh-vb) this formula can be used in determining the height of the man




```
function [x,y] = man
man = imread('man.jpg');
% figure
% imshow('shoe.jpg');
% [x,y] = getpts;
% pt1:(774,984) pt2:(1166,1132) pt3:(1070, 512) pt4:(1368,584)
M = size(man, 1);
N = size(man, 2);
% size of the image is 640, 427
% the horizon should be a line at 193 by my estimate
% since the coordinate system is origin at the top left so
% some modification of the coordinate
figure
imshow(man);hold on;
line([0,640],[200,200]);
scatter(149.03,71.36);
scatter(158.26,360);
Y = ((360 - 71.36)/(360 - 200)) * 95
```

[x,y] = getpts
% top (149.03,71.36) bottom (158.26,360)
% The man’s height is about 171.3800cm

end
The man’s height is about 171 cm with some human error allowed

4.
a)
K = [721.5 0 609.6;
0 721.5 172.9;
0 0 1]

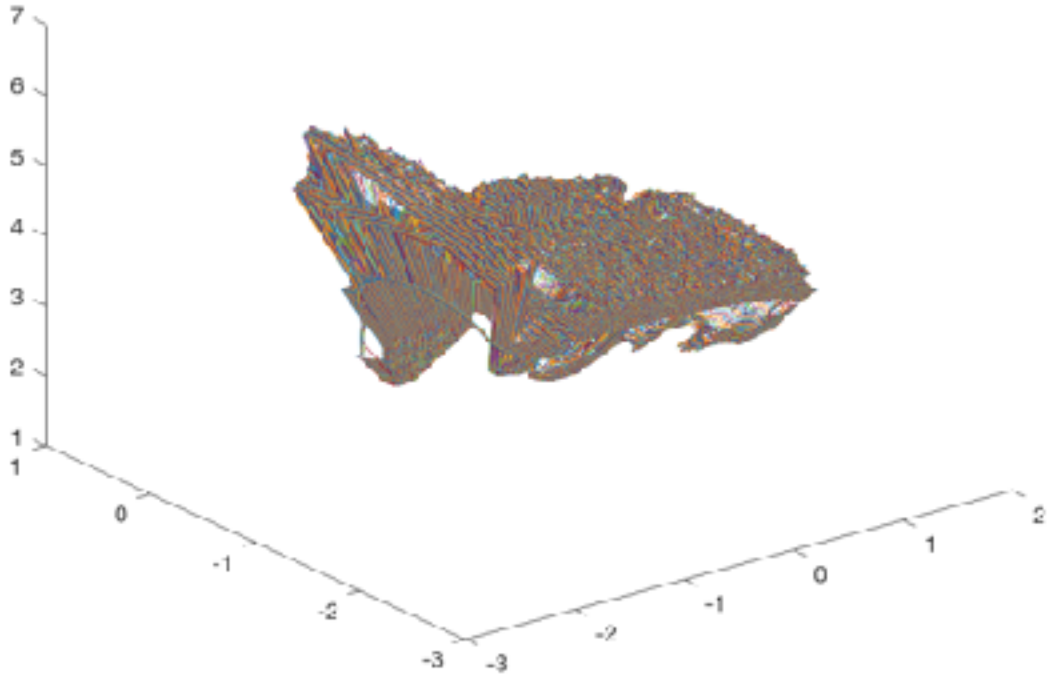
b)
f(x,y) = (x, - 1.7, y)

Bonus
function [x, y, z] = bonus(depth, labels)

% Depth Intrinsic Parameters
fx_d = 5.8262448167737955e+02;
fy_d = 5.8269103270988637e+02;
px_d = 3.1304475870804731e+02;
py_d = 2.3844389626620386e+02;

```
% Rotation
R = -[ 9.9997798940829263e-01, 5.0518419386157446e-03, ...
  4.3011152014118693e-03, -5.0359919480810989e-03, ...
  9.9998051861143999e-01, -3.6879781309514218e-03, ...
  -4.3196624923060242e-03, 3.6662365748484798e-03, ...
  9.9998394948385538e-01 ];

R = reshape(R, [3 3]);
R = inv(R');

% 3D Translation
t_x = 2.5031875059141302e-02;
t_z = -2.9342312935846411e-04;
t_y = 6.6238747008330102e-04;

y = zeros(size(depth));
x = zeros(size(depth));
z = zeros(size(depth));

for i=1:size(depth, 1)
   for j = 1:size(depth, 2)
      y(i, j) = depth(i, j)*(i - px_d)/fx_d;
      x(i, j) = depth(i, j)*(j - py_d)/fy_d;
      z(i, j) = depth(i, j);
   end

end


avg = zeros(3, 4);

for index=1:4
   idx = find(labels==index);
   obj = [y(idx)'; x(idx)'; z(idx)'];
   avg(:, index) = sum(obj, 2)./length(idx);

end

[max_avg, max_idx] = max(avg);

imagesc(labels == max_idx(1));
```

OUPUT IMAGE


7
6
5
4
3
2
1
1
0
-1
-2
-3
-3
-2
-1
0
1
2