UNIVERSITY OF TEXAS AT ARLINGTON

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

6367 COMPUTER VISION SPRING 2021

ASSIGNMENT 2

Nudrat Nawal Saber

1001733394

Problem 1a

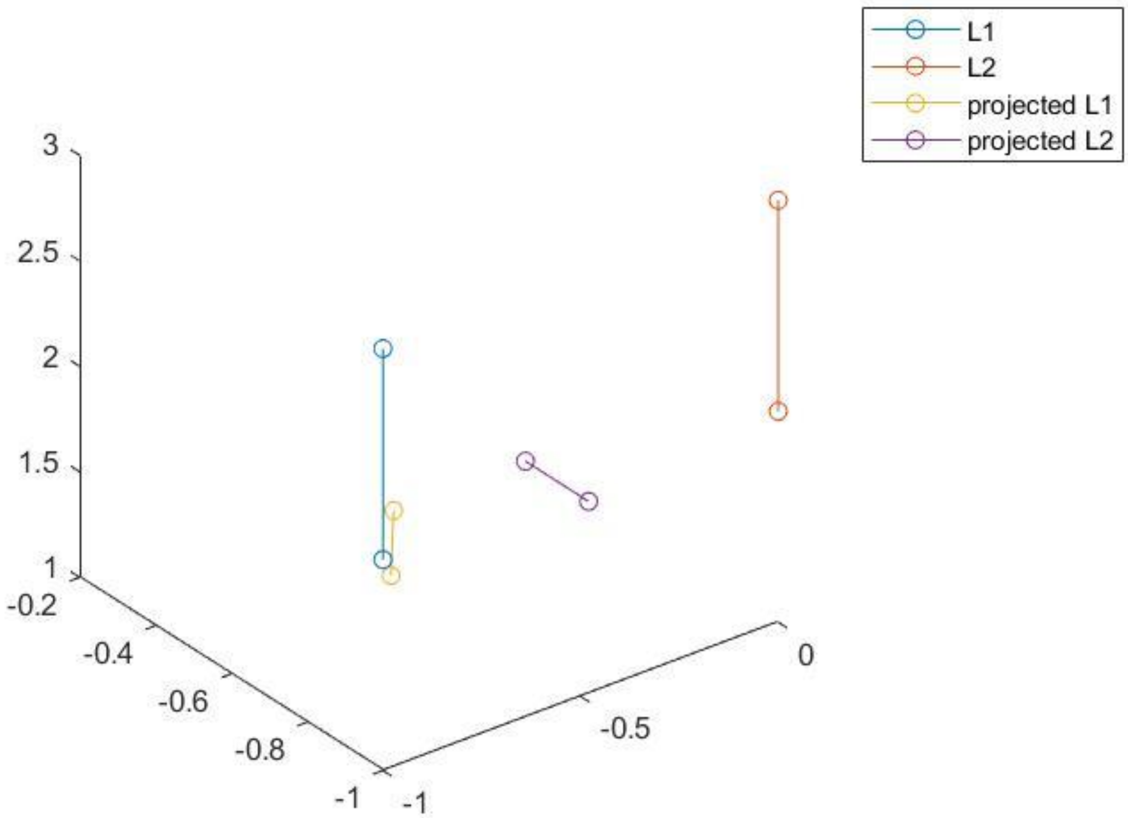
The project_point(p) function to compute the projection of a given point on the image plane can be seen attached as project_point.m file. Calling the problem1_a.m it will compute the projection points from the actual points given as inputs. We can express the line as (1−λ)p+λq(1−λ)p+λq with a pair of points (p,q) in the intersection.

Projected point and the actual point can be seen below

```
newpoint1 =

   -0.5000   -0.5000    1.0000
   -0.3333   -0.3333    1.0000
         0   -0.5000    1.0000
         0   -0.3333    1.0000
    0.5000   -0.5000    1.0000
    0.3333   -0.3333    1.0000


originalpoint =

    -1    -1     2
    -1    -1     3
     0    -1     2
     0    -1     3
     1    -1     2
     1    -1     3
```

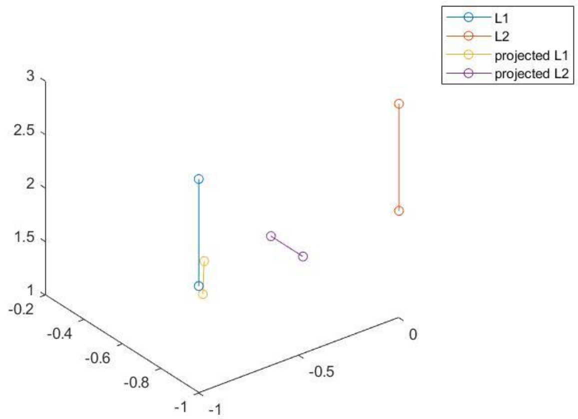
Problem 1b

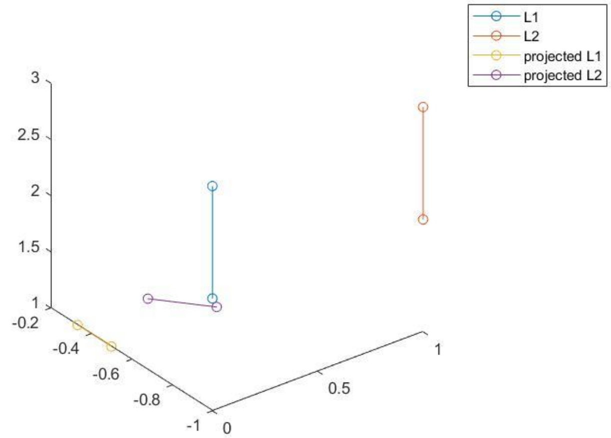
We have applied the find_intersection() function in find_intersection.m file. We use the formula from 1a but for 2 lines L1, L2. If we plot using the given points and plot it using matlab 3d plotter polt3 we will get the following output

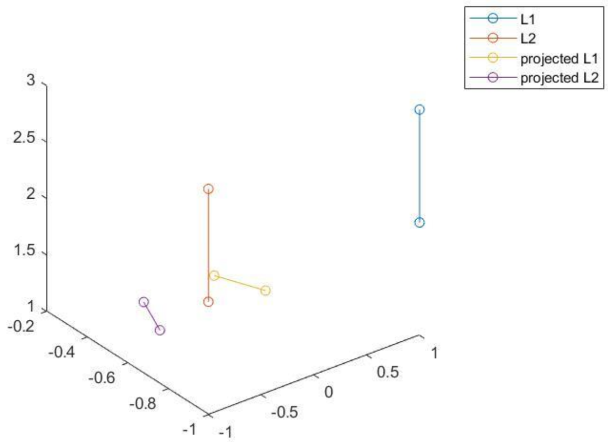


Problem 1c

For getting the output we need to execute problem1_c.m file where it uses the find_intersection function. If we compare the lines L1,L2,L3 and their projection and intersection line we can verify the claim that intersection for each pair of parallel lines are same.



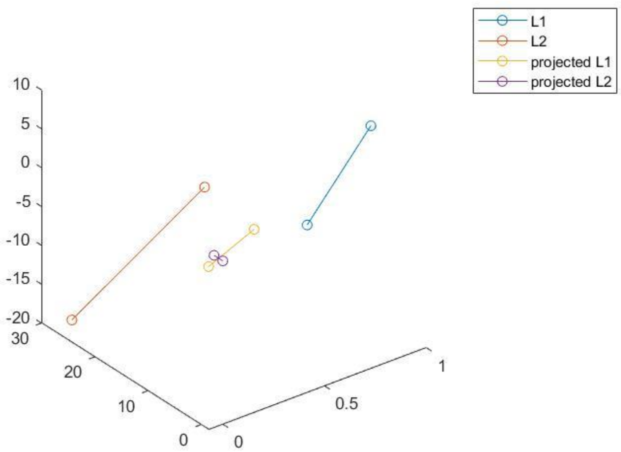


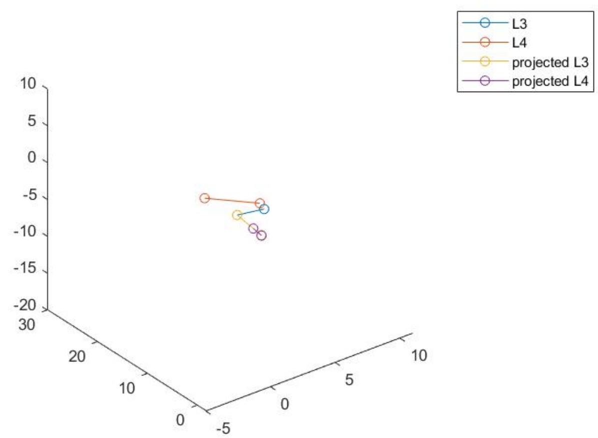


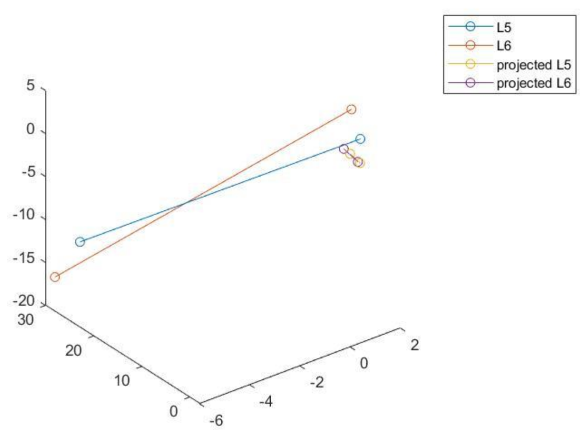
If we check from the above plot of 3 lines all the parallel lines from each plot have intersection point at infinity.

Problem 1d

We have attached pairwise_intersection.m for the solution. Here we apply the program over 3 given equation to verify the 3 points are colinear. To run the program we have to execute the given .m file. It will generate 5 plots as find_intesection will generate plots when calling the function. We will consider the last 3 plot as our final output. The output images are given below from which we can see that all points are collinear as they all intersect on infinity







Problem 2a

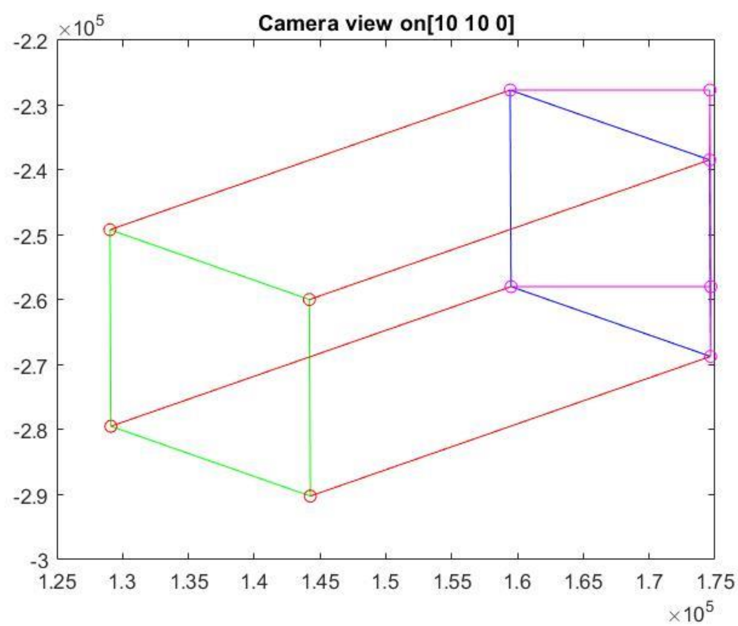
MATLAB function P C = project points(P W, R, t) that takes as input an N × 3 vector of points with coordinates in the world frame and returns as output an N × 2 vector of coordinates of points in the camera frame can be seen in the project_points.m file.

Problem 2b

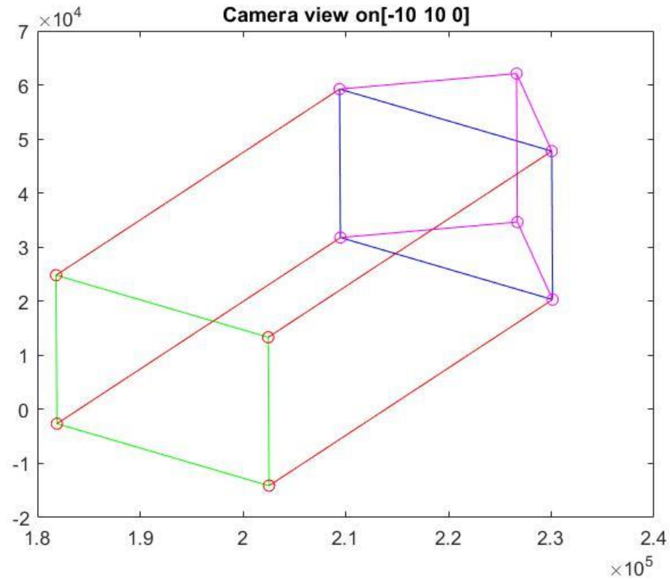
Calling the problem_2.m file will execute the project_point function to determine the projection of the image on given camera positions.

The output image when the camera is placed at the following positions: (i) [10, 10, 0], (ii) [-10, 10, 0], (iii) [0, 0, 10], (iv) [10, 0, 0], and (v) [10, 10, 10] can be seen below:

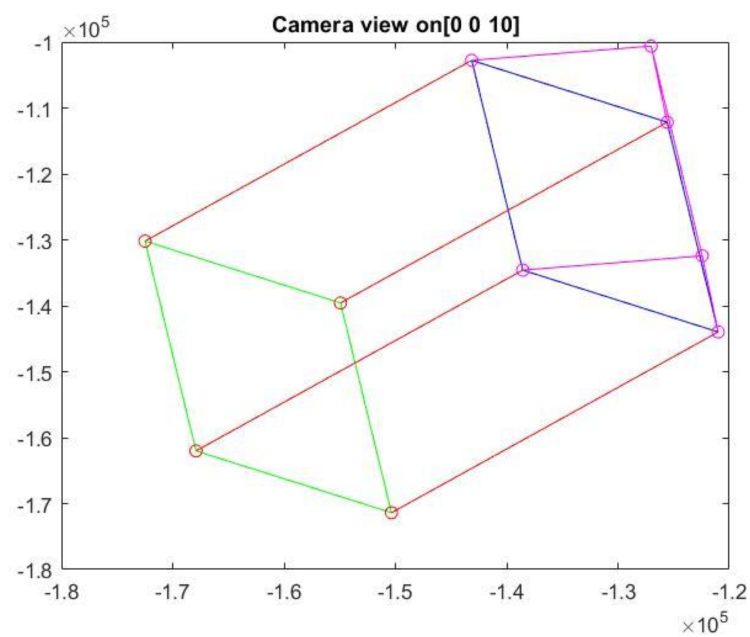
1. Camera position on [10,10,0]



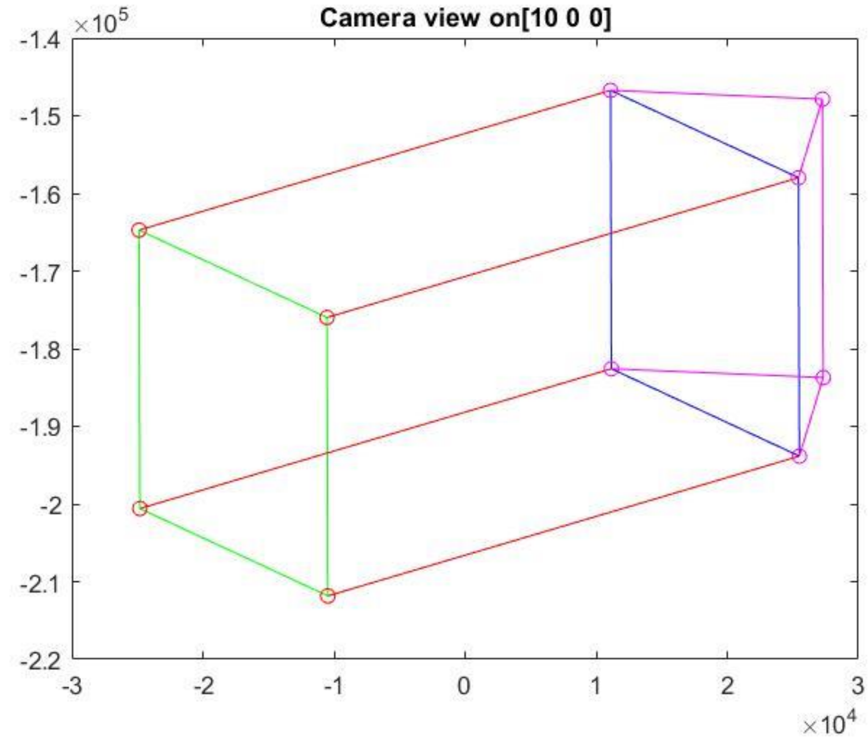
2. Camera position on [=10,10,0]



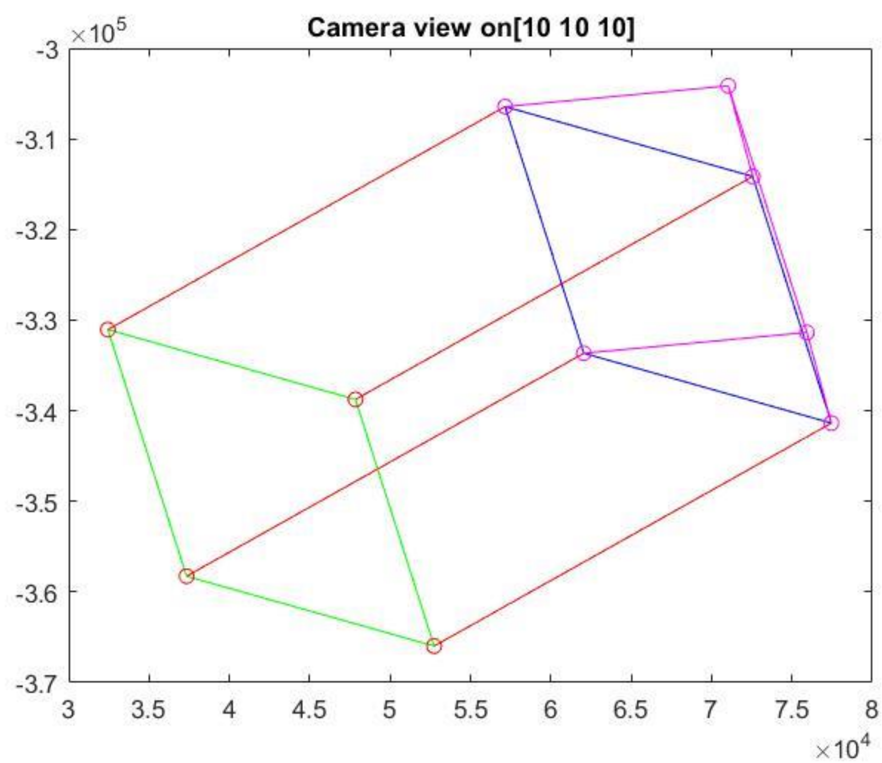
3. Camera position on [0,0,10]



4. Camera position on [10, 0, 0]



5. Camera position on [10, 10, 10]



Extra Credit

1(a)

If a line is defined by two points A and B ,by using the homogeneous coordinates and Plücker matrix of the line ,the matrix will be ,

$$L=AB^T-BA^T$$

If π is a vector that represents a plane ,its intersection with the line is

$$X=L\pi=(B^T\pi)A-(A^T\pi)B$$

So, the intersection of a line defined by two points and a plane can be computed with two dot products, two scalar products, a vector addition and then three divisions for the dehomogenization.

If the last coordinate of $L\pi$ is zero, then the line is parallel to the plane; if the entire product vanishes, then the line lies on the plane. Let's say there's a plane in 3d space, with a normal vector n of $<X_1, Y_1, Z_1>$. The point (X_2, Y_2, Z_2) lies on the plane as well. There is also a line that passes through points (X_3, Y_3, Z_3) and (X_4, Y_4, Z_4) .

Here, $A=(X_3, Y_3, Z_3,1)^T$, $B=(X_4, Y_4, Z_4,1)^T$ and $\pi=(X_1, Y_1, Z_1,- X_1X_2- Y_1 Y_2-Z_1Z_2)^T$.The elements of the latter are the coefficients of the point-normal equation of the plane.

1(b)

All affine transformations that have non-isotropic scaling will result in a circle being stretched and squashed into an ellipse. Next, let's prove the more general result. The quadratic equation of a general conic

$$C=Ax^2+Bxy+Cy^2+Dx+Ey+F$$

The value B^2-4AC, is called the discriminant of the conic section. If the discriminant is equal to 0 then the conic is classified as a parabola, if it is positive an ellipse and if negative a hyperbola. To find the classification of a conic after an affine transformation, let's find the discriminant of the transformed conic. The equation of transformation of a conic under a point transformation is-

$$C'=H^{-T}CH^{-1}$$

The determinant of the transpose of a matrix is the same as that of the matrix itself, the determinant will be $(\det(A))^2(ac-b^2/4)$. As the square term will always be positive, this discriminant will be 0 when $B^2/4=AC$, positive when $B^2/4<AC$ and negative when $B^2/4>AC$.

Thus, the classification of the affinely transformed conic is the same as the classification of the original conic and a circle can only be mapped into ellipse ,but can not be parabola or hyperbola under an affine transformation.