# CSE 6367 Assignment #2

Zahidur Talukder

March 2, 2021

## Problem 1

### (a)

In this problem we need to write a MATLAB function, that can project the given point in the image plane. The camera center co-located with origin i.e. $(0, 0, 0)$. I have find the straight line joining the camera center and the given points in 3D and the intersection of the image plane and straight line gives the co-ordinate of the projected point. The projection that I get for the points $P_{11} = [-1, -1, 2]^T$, $P_{12} = [-1, -1, 3]^T$, $P_{21} = [0, -1, 2]^T$, $P_{22} = [0, -1, 3]^T$, $P_{31} = [1, -1, 2]^T$ and $P_{32} = [1, -1, 3]^T$ are given in the figure-

```
Q11 =

   -0.5000   -0.5000    1.0000


Q12 =

   -0.3333   -0.3333    1.0000


Q21 =

         0   -0.5000    1.0000


Q22 =

         0   -0.3333    1.0000


Q31 =

    0.5000   -0.5000    1.0000


Q32 =

    0.3333   -0.3333    1.0000
```

Figure 1: Projection of the points

**(b)**

In this problem we need to write a MATLAB function Q= find_intersection(P11,,P12,P21,P22), that uses the previous function project_point to find the intersection of parallel line in the image plane. We computed the function and find the intersection is in $(0, 0, 1)$ co-ordinate. The visual representation of the parallel line and their intersection in 3d plane is given below:

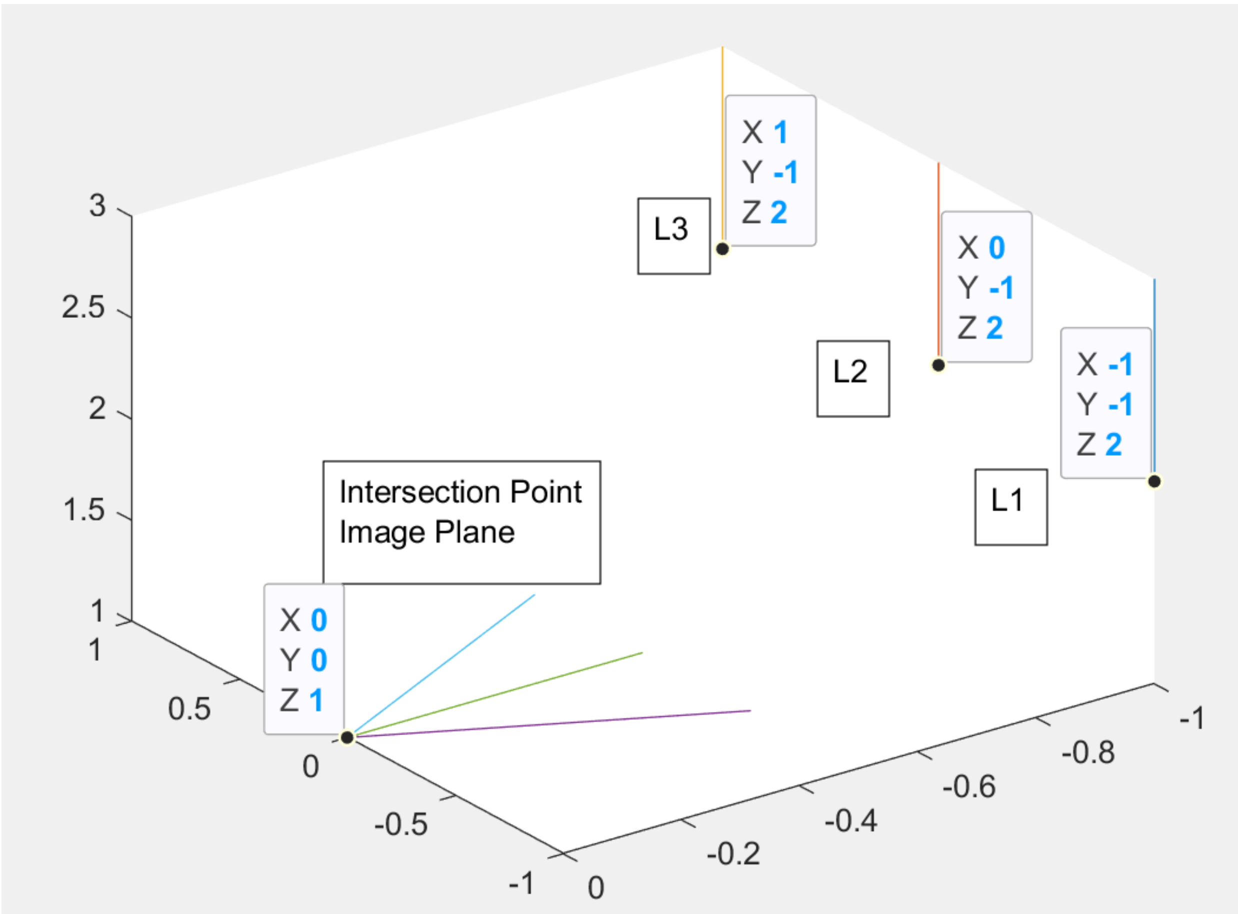


Figure 2: Intersection in the Image Plane

As We can see in the figure, the three parallel lines L1, L2 and L3 and their projection in the image plane, and the projecred line meets at $(0, 0, 1)$ co-ordinate.

### (c)

In this problem, We need to varify that, each pair of parallel lines meets at the same point. As we can see in the previous figure, 3 parallel lines meets at the same $(0, 0, 1)$ co-ordinate. The result of our simulation is also attached below:

```
l12 =

     0     0     1


l23 =

     0     0     1


l31 =

     0     0     1
```

Figure 3: Intersection in the Image Plane

### (d)

Three pairs of parallel lines are given, we need to write a "script pairwise_intersection" that uses previous function "find_intersection" for each pair. for this reason, I defined another function named "return_coordinate" that gives me 2 points from each line to use for "find_intersection" function. Then I was able to find the intersection in each pair in the image plain.

At first I find two points for each Parallel line using my defined function "return_coordinate". Then using four points for each group of parallel lines for the function "find_intersection" gives the intersection point in the Image plane. The Intersection points for first pair is $[0, 0, 1]$, for the 2nd pair is $[-.5333, -.4, 1]$ and for the last pair is $[-.4, 0, 1]$. Here I attached the pair of parallel lines in $\pi$ plane and their projection on the image plane.

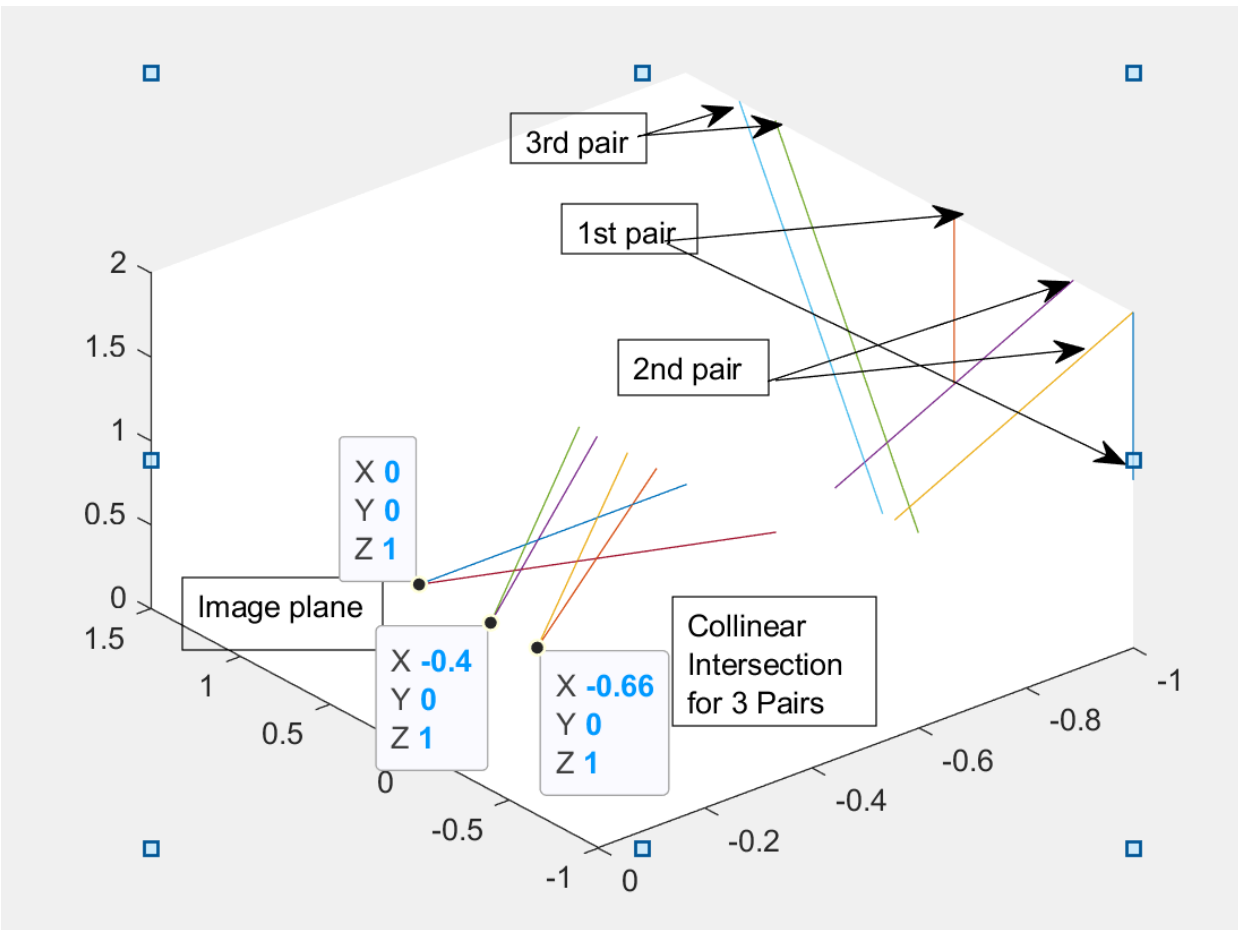


Figure 4: Intersection in the Image Plane for each pair of parallel lines

As we can see in the figure, there are 3 pairs of parallel line in $\pi$ plane and their intesection in the Image plane(Lower plane).

Since the determinant of the three points are 0, these points are collinear.

## Problem 2

### (a)

For this particular problem, we need to define a function called "project_points" that takes world frame 3D coordinate as input and return the image plane 2d coordinate as output. That means the function actually converts 3D object to a 2D image. Based on the position of the camera the image is changed. But if camera center moves from World cordinate to a different coordinate we need a translation matrix (Composed of Rotation Matrix and Transform Matrix) that can covert the world cordinate in camera coodinate system. So for solving this issue I have defined a function named "get_translation" that can return the translation matrix of the world coordinate to the camera coordinate. The function I defined is based on the direction provided for the assignment.

Since there is no skew in the camera and the scale factor is $\alpha = 200$ and $\beta = 200$ and the Image center is at $(50, 50)$. For the general case if camera center position is $(X_c, Y_c, Z_c)$, the focal length is the distance from the camera center to the image center.

The translation matrix that we get will be 4×4 matrix. So if the cordinate in in a World co-ordinate is $(X, Y, Z)$, then multiplying this with the Translation Matrix results their value in camera co-ordinate $(X_{cam}, Y_{cam}, Z_{cam})$ and using the formula

$$RT^{(4\times4)}_{translation} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \begin{pmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{pmatrix}$$

$$X_{im} = X_{cam} * \alpha / Z_{cam} + F_{focal}$$

$$Y_{im} = Y_{cam} * \beta / Z_{cam} + F_{focal}$$

I have attached the fucnction in the report. Based on the function we will see the output for different camera postion in the next problem.

**(b)**

**i**

In this particular problem, we need to find the image of that house based on different camera position. In the first Problem, the camera is placed at $(10, 10, 0)$. The image that we get is-

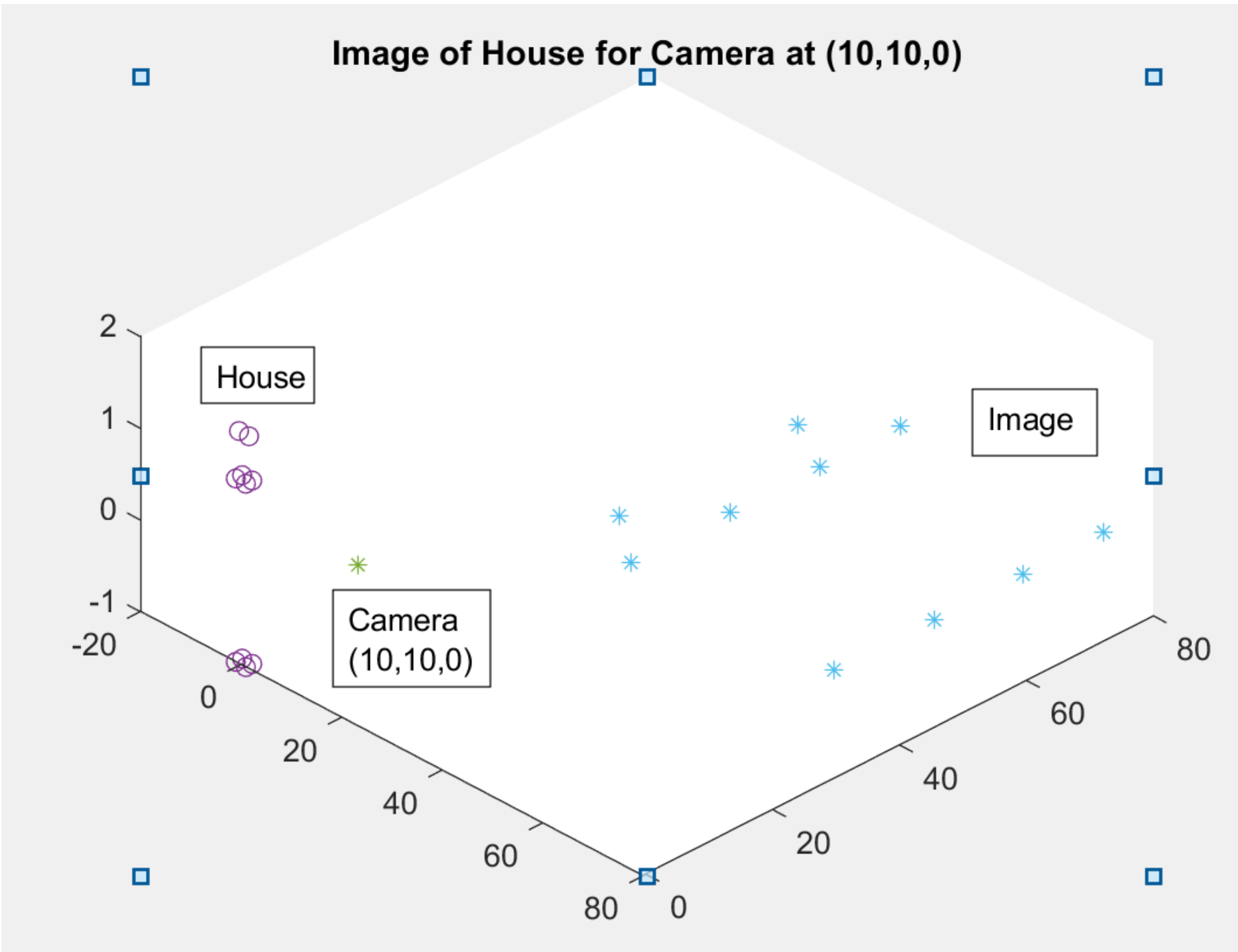


Figure 5: Image of the house placing Camera at (10,10,0)

**ii**

In this particular problem, we need to find the image of that house based on different camera position. In the first Problem, the camera is placed at $(-10, 10, 0)$. The image that we get is-

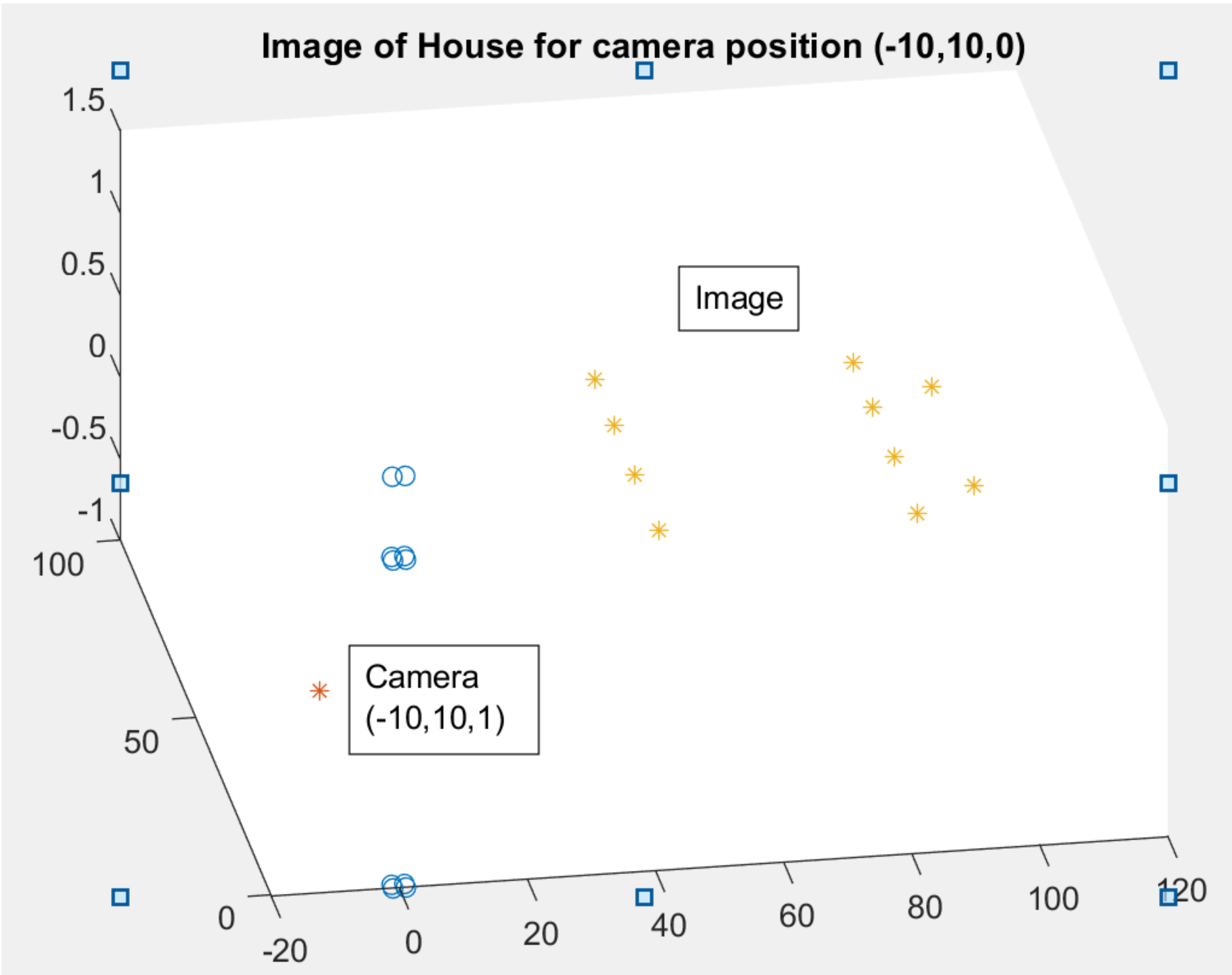


Figure 6: Image of the house placing Camera at (-10,10,0)

**iii**

In this particular problem, we need to find the image of that house based on different camera position. In the first Problem, the camera is placed at $(0, 0, 10)$. The image that we get is-

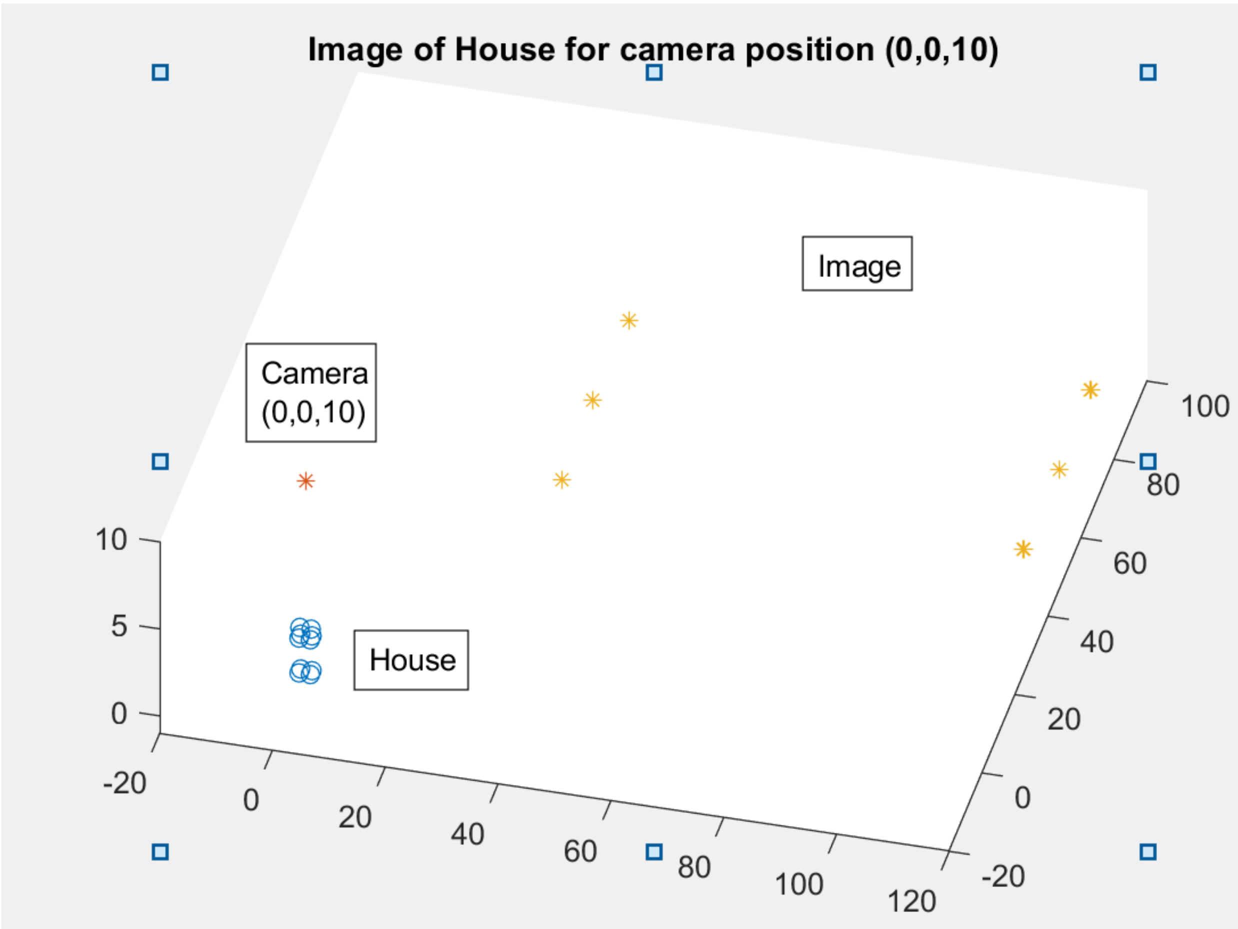


Figure 7: Image of the house placing Camera at (0,0,10)

**iv**

In this particular problem, we need to find the image of that house based on different camera position. In the first Problem, the camera is placed at $(10, 0, 0)$. The image that we get is-

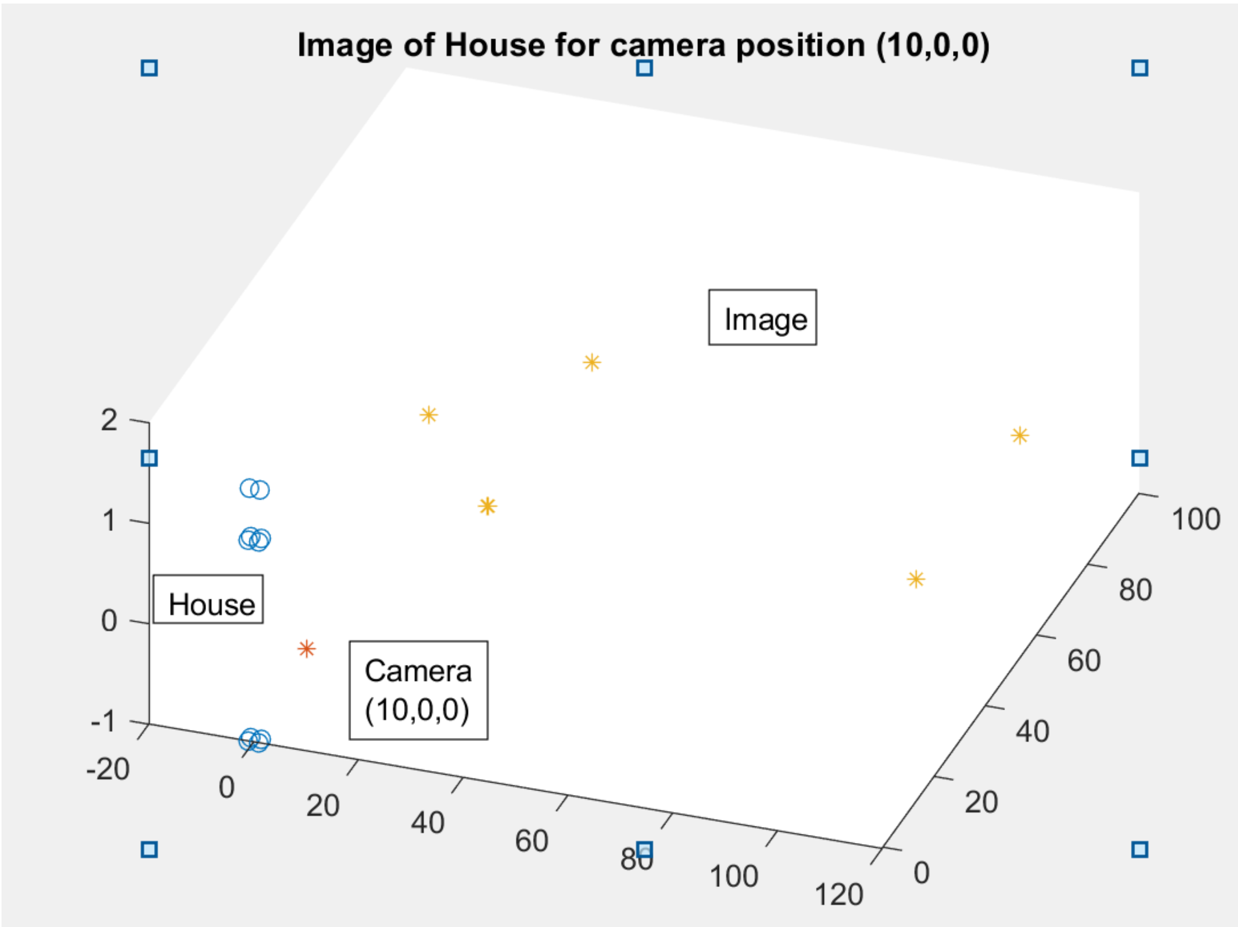


Figure 8: Image of the house placing Camera at (10,0,0)

**v**

In this particular problem, we need to find the image of that house based on different camera position. In the first Problem, the camera is placed at $(10, 10, 10)$. The image that we get is-

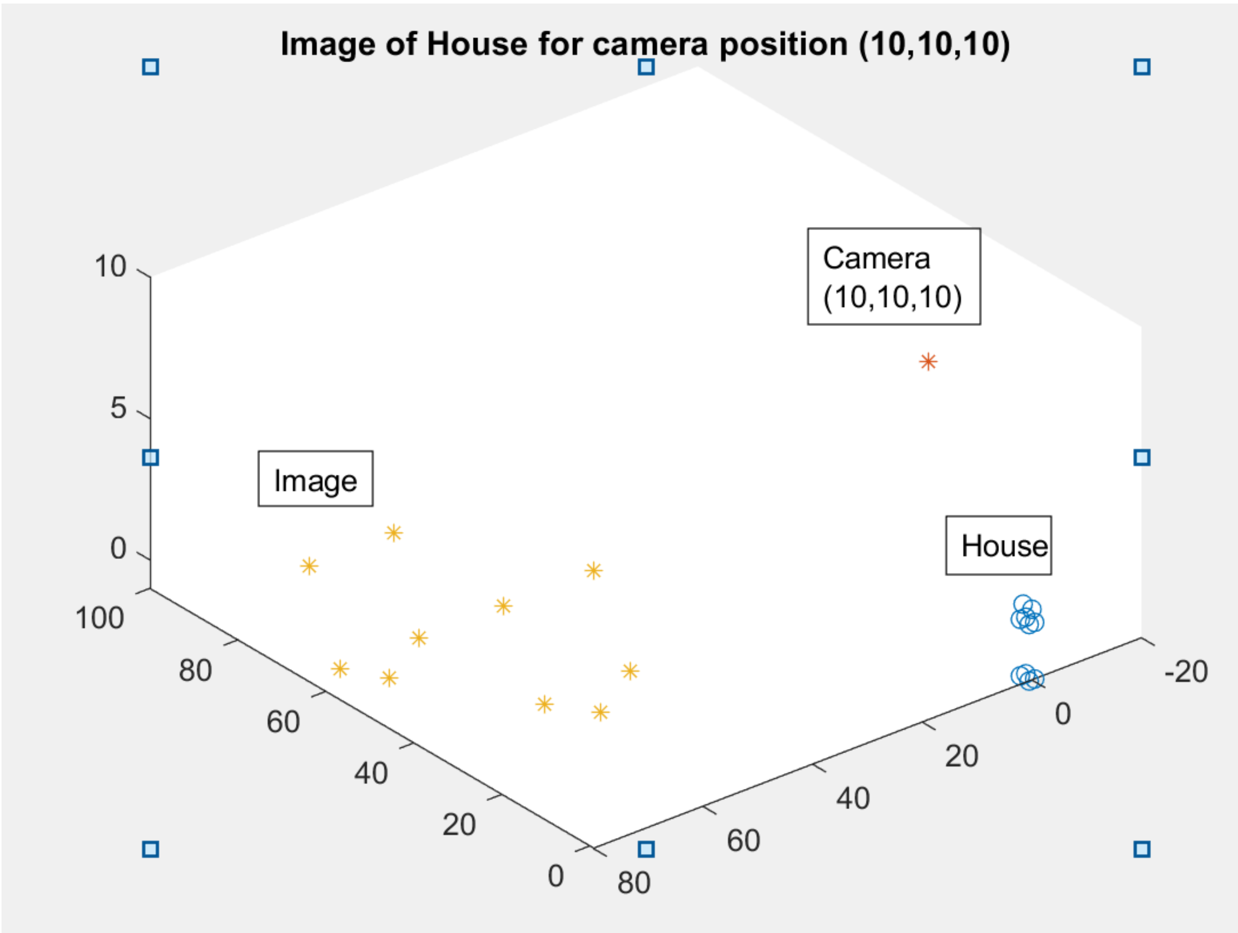


Figure 9: Image of the house placing Camera at (10,10,10)

## Extra Question 1

We know that the point of intersection $X$ of a line with Plücker matrix $\mathcal{L}$ and a plane $\pi$ is given by

$$X = \mathcal{L}\pi$$

Expanding this using Plücker coordinates $l_{12}, l_{13}, l_{14}, l_{24}, l_{42}, l_{34}$ as elements of the skew-symmetrix matrix $\mathcal{L}$, we get

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} 0 & l_{12} & l_{13} & l_{14} \\ -l_{12} & 0 & l_{23} & -l_{42} \\ -l_{13} & -l_{23} & 0 & l_{34} \\ -l_{14} & l_{42} & -l_{34} & 0 \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix}$$

The matrix product

$$\begin{pmatrix} 0 & l_{23} & -l_{42} \\ -l_{23} & 0 & l_{34} \\ l_{42} & -l_{34} & 0 \end{pmatrix} \begin{pmatrix} \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix}$$

can be represented as the cross product

$$-b \times \pi = \pi \times b$$

where $\boldsymbol{b}$=$[l_{34}, l_{42}, l_{23}]$ and $\boldsymbol{\pi}$= $\pi_2$,$\pi_3$,$\pi_4$. We can now rewrite the matrix product as follows,

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} = \begin{pmatrix} 0 & a \\ -a & -[b] \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix}$$

where $\boldsymbol{a}$=$[l_{12}, l_{13}, l_{14}]$. Then we get,

$$X_1 = a.\pi$$

$$X = -[b] \times \pi - \pi_1 a = \pi \times b - \pi_1 a$$

Where $X$=$[X_1, X_2, X_3]$. This is the equation of the line-plane intersection in Plücker coordinates $\mathcal{L}$=$(a, b)$ where $\boldsymbol{a}$=$[l_{12}, l_{13}, l_{14}]$ and $\boldsymbol{b}$=$[l_{34}, l_{42}, l_{23}]$ and $a.b = 0$. Similarly from the expansion of equation $\pi = \mathcal{L}.X$, we can get the equation a plane defined by point and line in Plücker coordinates to be

$$\pi_1 = b.X$$

$$\pi = X \times a - X_1 b$$

## Extra Question 2

All affine transformation that have non isotropic scalling will result in a circle being mapped to an ellipse. The quadratic equation of a general conic can be represented in matrix form as-

$$C = \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix}$$

The euclidean classification of a conic is based on the determinant of the upper left 2×2 matrix of its quadratic form. this value, $ac - b^2/4$, is also called the discriminant of the conic section. I the discriminant is 0 theb the conic is parabola, if positive an ellipse and negative a hyperbola.

lets find the discrimant of transformed conic for finding its classification after affine transformation. The equation of the transformation of a conic under a point transformation is given by $C' = H^{-T}CH^{-1}$. As the inverse of an affine transformation is also an affine transformation. We can write the conic as-

$$C' = \begin{pmatrix} A^T & 0 \\ P^T & 1 \end{pmatrix} \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix} \begin{pmatrix} A & p \\ 0 & 1 \end{pmatrix}$$

The discriminant of this transormed conic will be the discriminant of the following sub-matrix

$$A^T \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix} A$$

As the determinant of the transpose of a matrix is the same as that of the matrix itself, the above mentioned determinant will be $(det(A))^2(ac - b^2/4)$. As the square term will always be positive, this discriminant will be ) when $b^2/4 = ac$, positive when $b^2/4 < ac$ and negative when $b^2/4 > ac$. Hence the classification of the affine transformed conic is the same as the classification of the original conic and an ellipse can only be transformed to an ellipse and never to a parabola or a hyperbola under an affine transformation.