Dr. Song
Cristofer Belen
Richie Molina

AMAT 240-01 (Intro. to Linear Algebra)
Case Study: Computer Graphics in Automotive Design

Fall 2022
Group G1

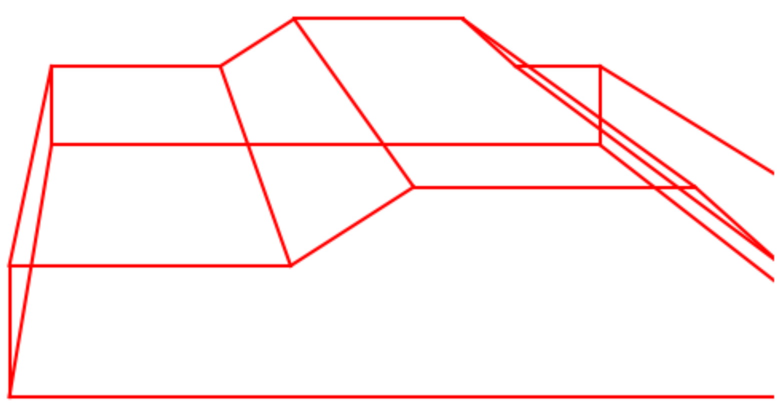
1. (15 points) Find the perspective projection using Toyota's data in matrix D using *(b, c, d)* = (−5, 10, 10) as center of projection. What is the matrix of the perspective projection? Present the outcome from your Python by attaching the figure of the vehicle viewed from (−5, 10, 10).

a. Matrix of Perspective Projection P1: $\begin{bmatrix} 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{10} & 1 \end{bmatrix}$

b. PD1

```
#1: Set up Perspective Projection PD1 on matrix D using P1(b, c, d) = (-5, 10, 10):

[[-6.2        -7.         -7.         -6.2        -3.         -1.66666667
  -1.6         0.66666667  1.6         6.         2.6         7.66666667
   4.2        10.33333333 10.33333333  4.2       ]
 [ 0.4        -6.         -2.66666667  2.4        2.4        -2.66666667
   3.6        -0.66666667  3.6        -0.66666667  2.4        -2.66666667
   2.4        -2.66666667 -6.         0.4       ]
 [ 0.          0.          0.          0.          0.          0.
   0.          0.          0.          0.          0.          0.
   0.          0.          0.          0.        ]
 [ 1.25        0.75        0.75        1.25        1.25        0.75
   1.25        0.75        1.25        0.75        1.25        0.75
   1.25        0.75        0.75        1.25      ]]
```



c.

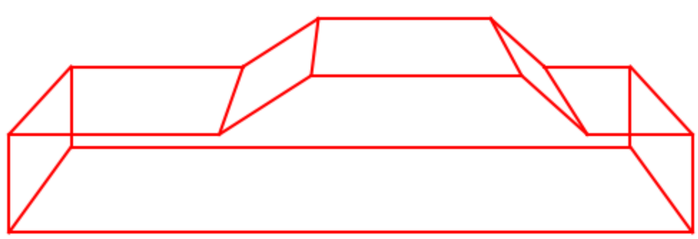
2. (15 points) Find the perspective projection using the Toyota data in matrix D using (b, c, d) = (0, 10, 25) as the center of projection. What is the matrix of the perspective projection? Present the outcome from your Python by attaching the figure of the vehicle viewed from (0, 10, 25).

a. Matrix of Perspective Projection P2: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\frac{10}{25} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{25} & 1 \end{bmatrix}$

b. PD2

```
#2: Set up Perspective Projection PD2 on matrix D using P2(b, c, d) = (0, 10, 25):

[[-5.90909091 -7.22222222 -7.22222222 -5.90909091 -2.27272727 -2.77777778
  -0.68181818 -0.83333333  2.95454545  3.61111111  4.09090909  5.
   5.90909091  7.22222222  7.22222222  5.90909091]
 [-0.90909091 -3.33333333 -0.55555556  1.36363636  1.36363636 -0.55555556
   2.72727273  1.11111111  2.72727273  1.11111111  1.36363636 -0.55555556
   1.36363636 -0.55555556 -3.33333333 -0.90909091]
 [ 0.          0.          0.          0.          0.          0.
   0.          0.          0.          0.          0.          0.
   0.          0.          0.          0.        ]
 [ 1.1         0.9         0.9         1.1         1.1         0.9
   1.1         0.9         1.1         0.9         1.1         0.9
   1.1         0.9         0.9         1.1       ]]
```



c.

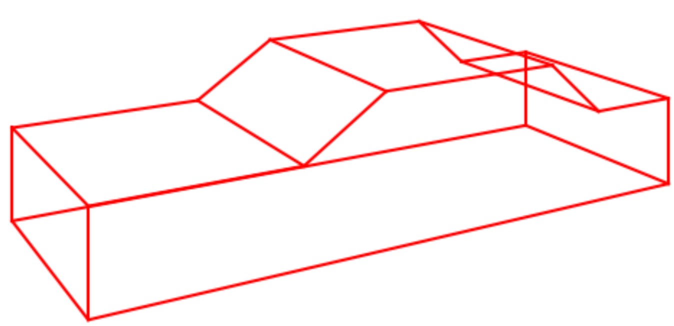
3. (25 points) Rotate the Toyota 30° about the y-axis, then perform the perspective projection with center of projection (0,10,25). What is the matrix of the rotation? Present the outcome from your Python by attaching the figure of the rotated vehicle viewed from (0, 10, 25). How does this figure compare with that in Question 2?

a. Matrix of Rotation R1 where $\varphi = 30^\circ\, or \frac{\pi}{6}$: $\begin{bmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

b. PRD1

```
#3: PRD1: A 30° Rotation about y-axis then Perspective Proj. on matrix D using P2(b, c, d) = (0, 10, 25):

[[-7.19124697 -5.58996595 -5.58996595 -7.19124697 -3.29447727 -1.0598404
  -1.77259663  0.66839119  1.35861333  4.15432657  2.24979483  5.12968641
   3.59950352  6.59304377  6.59304377  3.59950352]
 [-2.54439487 -5.31789496 -2.12666684  0.06902073  0.83544596 -1.00304372
   2.53454551  1.09525532  3.05315878  1.82336389  1.92590558  0.53216658
   2.19136926  0.8951283  -1.50089057  0.13646643]
 [ 0.          0.          0.          0.          0.          0.
   0.          0.          0.          0.          0.          0.
   0.          0.          0.          0.        ]
 [ 0.95660254  0.78339746  0.78339746  0.95660254  1.03660254  0.86339746
   1.07160254  0.89839746  1.15160254  0.97839746  1.17660254  1.00339746
   1.21660254  1.04339746  1.04339746  1.21660254]]
```



c. Using how the perspective projection matrix was created for the figure in Question 2 by the center of projection at (0, 10, 25), the perspective projection matrix P is multiplied against the Dimension matrix D. Before this matrix operation, however, Toyota's y-axis is first rotated by phi1 (counterclockwise since it's positive) as the Matrix of Rotation R1 uses φ = 30° or $\frac{\pi}{6}$ with respect to the Perspective Projection RD1 (0, 10, 25) and matrix D. $\mathbb{R}^3$ is then found by (x, y, z, 1) as homogenous coordinates for the points (x, y, z) in $\mathbb{R}^3$. In general, (X, Y, Z, H) are homogenous coordinates for (x, y, z) if $H \neq 0$. Therefore, we can divide (x, y, z) in each column space by the corresponding entry in the fourth row H. The (x, y) coordinates are displayed in section a in PRD1 and subsequently plotted with ax4 and by the adjacency matrix C denoted in part b.

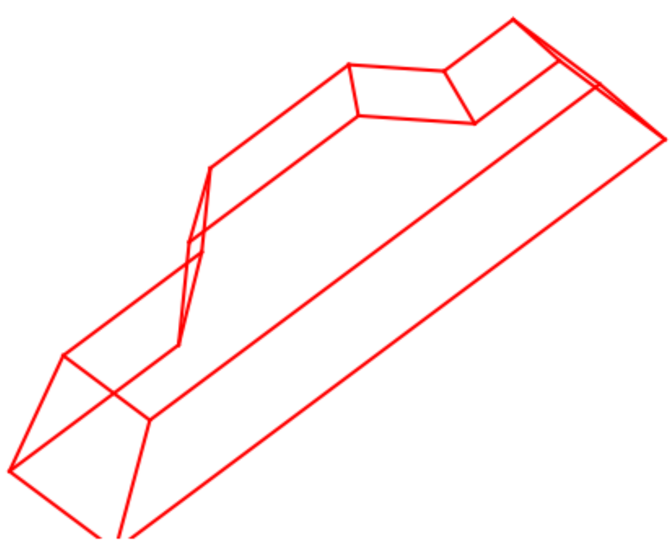
4. (25 points) Rotate the Toyota 45° about the z-axis, then perform the perspective projection with the center of projection (0,10,25). What is the matrix of the rotation? Present the outcome from your Python by attaching the figure of the rotated vehicle viewed from (0, 10, 25). How does this figure compare with that in Question 2?

$a.$ Matrix of Rotation R2 where $\varphi = 45° \, or \frac{\pi}{4}$: $\begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & 0 \\ \sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$b.$ PRD2

```
#4: PRD2: A 45° Rotation about z-axis and Perspective Proj. on matrix D using P2(b, c, d) = (0, 10, 25):

[[-2.89270956 -3.53553391 -5.49971941 -4.49977043 -1.92847304 -2.3570226
  -1.76776695 -2.16060405  0.80353043  0.98209275  2.57129739  3.14269681
   3.85694608  4.71404521  6.67823071  5.46400695]
 [-4.55491604 -7.78934182 -5.82515632 -2.94785517 -0.37655778 -2.68245951
   1.71262134 -0.12901836  4.28391873  3.01367845  4.12321264  2.8172599
   5.40886133  4.3886083   2.42442279  3.80180047]
 [ 0.          0.          0.          0.          0.          0.
   0.          0.          0.          0.          0.          0.
   0.          0.          0.          0.        ]
 [ 1.1         0.9         0.9         1.1         1.1         0.9
   1.1         0.9         1.1         0.9         1.1         0.9
   1.1         0.9         0.9         1.1       ]]
```



$c.$

d. Using how the perspective projection matrix was created for the figure in Question 2 by the center of projection at (0, 10, 25), the perspective projection matrix P is multiplied against the Dimension matrix D. Before this matrix operation, however, Toyota's z-axis is first rotated by phi2 (counterclockwise since it's positive) as the Matrix of Rotation R2 uses φ = 45° or $\frac{\pi}{4}$ with respect to the Perspective Projection RD2 with respect to (0, 10, 25) and D. $\mathbb{R}^3$ is then found by (x, y, z, 1) as homogenous coordinates for the points (x, y, z) in $\mathbb{R}^3$. In general (X, Y, Z, H) are homogenous coordinates for (x, y, z) if $H \neq 0$. Therefore, we can divide (x, y, z) in each column space by the corresponding entry in the fourth row H. The (x, y) coordinates are displayed in section a in PRD1 and subsequently plotted with ax5 and by the adjacency matrix C denoted in part b.

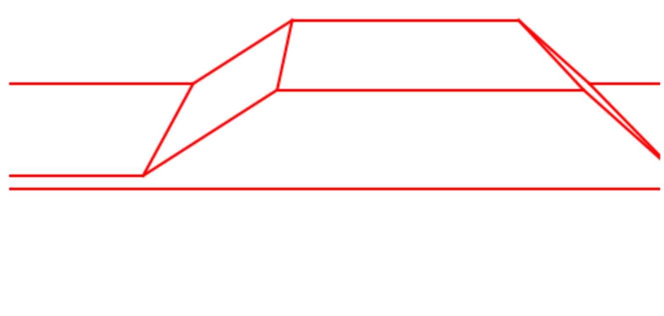
5. (25 points) Zoom in on the Toyota with a zoom factor of 150%, then perform the perspective projection with center of projection (0, 10, 25). What is the matrix of zoom? Present the outcome from your Python by attaching the figure of the zoomed vehicle viewed from (0, 10, 25). How does this figure compare with that in Question 2?

a. The Matrix of Zoom: $\begin{bmatrix} 1.5 & 0 & 0 & 0 \\ 0 & 1.5 & 0 & 0 \\ 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

b. PqD

```
#5: PqD: A 150% zoom factor then Perspective Proj. #2 on matrix D using P2(b, c, d) = (0, 10, 25):

[[ -8.47826087 -11.47058824 -11.47058824  -8.47826087  -3.26086957
   -4.41176471  -0.97826087  -1.32352941   4.23913043   5.73529412
    5.86956522   7.94117647   8.47826087  11.47058824  11.47058824
    8.47826087]
 [ -1.30434783  -5.29411765  -0.88235294   1.95652174   1.95652174
   -0.88235294   3.91304348   1.76470588   3.91304348   1.76470588
    1.95652174  -0.88235294   1.95652174  -0.88235294  -5.29411765
   -1.30434783]
 [  0.           0.           0.           0.           0.
    0.           0.           0.           0.           0.
    0.           0.           0.           0.           0.
    0.        ]
 [  1.15         0.85         0.85         1.15         1.15
    0.85         1.15         0.85         1.15         0.85
    1.15         0.85         1.15         0.85         0.85
    1.15      ]]
```



c.

d. Similar how the perspective projection matrix was created in Question 2 by using the center of projection P2 at (0, 10, 25), the dimension matrix D is first multiplied by the zoom matrix q which is stored into qD. Similar to how rotations are processed, we will then find the product of P2 and qD. The Zoom at the same perspective projection from Question 2 is simply zoomed in by 150% on (x, y, z, 1).

6. (5 points) Include Python codes in plain text at end of your report.

```
"""

Dr. Song

AMAT 240-01 FA22

Project 2: Case Study: Computer Graphics in Automotive Design

Cristofer Belen

John Molina

"""


import math

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.pyplot import plot, ion, show



# Plots the coorinates of the car
```

```
D = np.array([[-6.5, -6.5, -6.5, -6.5, -2.5, -2.5, -0.75, -0.75, 3.25,
3.25, 4.5, 4.5, 6.5, 6.5, 6.5, 6.5],

               [-2, -2, 0.5, 0.5, 0.5, 0.5, 2, 2, 2, 2, 0.5, 0.5, 0.5, 0.5,
-2, -2],

               [-2.5, 2.5, 2.5, -2.5, -2.5, 2.5, -2.5, 2.5, -2.5, 2.5, -
2.5, 2.5, -2.5, 2.5, 2.5, -2.5],

               [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]])


# Adjacency matrix to connect all the lines of the car

C = np.array([[0, 1, 0, 1, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0, 1],

              [1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0],

              [0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

              [1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

              [0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0],

              [0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],

              [0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0],

              [0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0],

              [0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0],

              [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0],

              [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0],

              [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0],

              [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1],

              [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0],

              [0, 1, 0, 0, 0, 0, 0 ,0 ,0, 0, 0, 0, 0, 1, 0, 1],

              [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0]])
```

```
# Plots all points using the coordinates matrix D & its adjacency matrix C.

f, ax1 = plt.subplots(1)

for i in range(16):

    for j in range(i):

        if C[i, j] == 1:

            ax1.plot([D[0, i], D[0, j]], [D[1, i], D[1, j]], 'r')

ax1.axis('off')

plt.xlim(-7.5, 7.5)

plt.ylim(-7.5, 7.5)




# Perspective Projection #1

# P1 = perspective,  D = matrix for car, PD1 = perspective(-5,10,10)

# Question 1

P1 =np.array([[1, 0, 1/2, 0],

             [0, 1, -1, 0],

             [0, 0, 0, 0],

             [0, 0, -1/10, 1]])

PD1 =np.matmul(P1, D)

PD1[0, :]= PD1[0, :]/PD1[3, :]

PD1[1, :]= PD1[1, :]/PD1[3, :]


f, ax2 = plt.subplots(1)

for i in range(16):
```

```
        for j in range(i):

            if C[i, j] == 1:

                ax2.plot([PD1[0, i], PD1[0, j]], [PD1[1, i], PD1[1, j]], 'r')


    ax2.axis('off')

    plt.xlim(-7.5, 7.5)

    plt.ylim(-7.5, 7.5)




    # Perspective Projection #2

    # P2 = perspective  D = matrix for car, PD2= perspective(0,10,25)

    # Question 2

    P2 =np.array([[1, 0, 0, 0],

                 [0, 1, -10/25, 0],

                 [0, 0, 0, 0],

                 [0, 0, -1/25, 1]])

    PD2 =np.matmul(P2, D)

    PD2[0, :]= PD2[0, :]/PD2[3, :]

    PD2[1, :]= PD2[1, :]/PD2[3, :]




    f, ax3 = plt.subplots(1)

    for i in range(16):

        for j in range(i):
```

```
        if C[i, j] == 1:

            ax3.plot([PD2[0, i], PD2[0, j]], [PD2[1, i], PD2[1, j]], 'r')

ax3.axis('off')

plt.xlim(-7.5, 7.5)

plt.ylim(-7.5, 7.5)




# Rotate it 30 degrees about y-axis

# R1 = rotate, P2 is perspective of (0,10,25), PRD is P times RD

# Question 3

phi1=math.pi/6

R1 = np.array([[math.cos(phi1), 0, math.sin(phi1), 0],

              [0, 1, 0, 0],

              [-math.sin(phi1), 0, math.cos(phi1), 0],

              [0, 0, 0, 1]])

RD1 = np.matmul(R1, D)

PRD1= np.matmul(P2, RD1)

PRD1[0, :]= PRD1[0, :]/PRD1[3, :]

PRD1[1, :]= PRD1[1, :]/PRD1[3, :]

f, ax4 = plt.subplots(1)

for i in range(16):

    for j in range(i):

        if C[i, j] == 1:

            ax4.plot([PRD1[0, i], PRD1[0, j]], [PRD1[1, i], PRD1[1, j]],
'r')
```

```
ax4.axis('off')

plt.xlim(-7.5, 7.5)

plt.ylim(-7.5, 7.5)


# Rotate it 45 degrees about z-axis

# R = rotate, P is perspective of (0,10,25) PRD is P times RD

# Question 4

phi2=math.pi/4

R2 = np.array([[math.cos(phi2), -math.sin(phi2), 0, 0],

              [math.sin(phi2), math.cos(phi2), 0, 0],

              [0, 0, 1, 0],

              [0, 0, 0, 1]])

RD2 = np.matmul(R2, D)

PRD2= np.matmul(P2, RD2)

PRD2[0, :]= PRD2[0, :]/PRD2[3, :]

PRD2[1, :]= PRD2[1, :]/PRD2[3, :]

f, ax5 = plt.subplots(1)

for i in range(16):

    for j in range(i):

        if C[i, j] == 1:

            ax5.plot([PRD2[0, i], PRD2[0, j]], [PRD2[1, i], PRD2[1, j]],
'r')

ax5.axis('off')

plt.xlim(-7.5, 7.5)

plt.ylim(-7.5, 7.5)
```

```
# q = to our zoom array

# Question 5

q= np.array([[1.5, 0, 0, 0],

             [0, 1.5, 0, 0],

             [0, 0, 1.5, 0],

             [0, 0, 0, 1]])

qD=np.matmul(q, D)

PqD= np.matmul(P2, qD )

PqD[0, :]= PqD[0, :]/PqD[3, :]

PqD[1, :]= PqD[1, :]/PqD[3, :]

f, ax6 = plt.subplots(1)

for i in range(16):

    for j in range(i):

        if C[i, j] == 1:

            ax6.plot([PqD[0, i], PqD[0, j]], [PqD[1, i], PqD[1, j]], 'r')

ax6.axis('off')

plt.xlim(-7.5, 7.5)

plt.ylim(-7.5, 7.5)

show()




print("Toyota's dimensions D:\n", D)

print("\n")
```

```
print("Adjacency Matrix C:\n", C)

print("\n")

print("Perspective Matrix P1 in Q1:\n", P1)

print("\n")

print("\n")

print("VVV this Perspective Matrix is used often throughout this report.
VVV\n")

print("Perspective Matrix P2 in Q2:\n", P2, "\n")

print("^^^ this Perspective Matrix is used often throughout this report.
^^^")

print("\n")

print("30° counterclockwise rotation along y-axis matrix:\n", R1)

print("\n")

print("45° counterclockwise rotation along z-axis matrix:\n", R2)

print("\n")

print("phi1 = 30° or pi/6: ", phi1)

print("phi2 = 45° or pi/4: ", phi2)

print("\n")

print("#1: Set up Perspective Projection P1 on matrix D using (b, c, d) =
(-5, 10, 10):\n\n", PD1, "\n\n")



print("#2: Set up Perspective Projection P2 on matrix D using (b, c, d) =
(0, 10, 25):\n\n", PD2, "\n\n")

print("#3: 30° Rotation about y-axis then Perspective Proj. on matrix D
using P2(b, c, d) = (0, 10, 25):\n\n", PRD1, "\n\n")

print("#4: 45° Rotation about z-axis and Perspective Proj. on matrix D
using P2(b, c, d) = (0, 10, 25):\n\n", PRD2, "\n\n")
```

```
print("#5: 150% zoom factor then Perspective Proj. #2 on matrix D using
P2(b, c, d) = (0, 10, 25):\n\n", PqD)
```