1 Object manipulation

A set of transformation values were given with the question specification.

Student ID	δ	σ	θ	θ_c	ϕ_c	d_c
190018469	$\begin{bmatrix}0\\1\\1\end{bmatrix}$	$\begin{bmatrix}0.7\\0.7\\0.7\end{bmatrix}$	$\begin{bmatrix}0.1\\0.2\\0.1\end{bmatrix}$	0.15	0.07	12

Table 1: Transformation Values

1. a) The 4×4 transformation matrix for each operation

The matrices used are standard transformation matrices with the transformation values substituted.

Translating the object by vector δ

$$T_\delta = \begin{bmatrix}1 & 0 & 0 & \delta_x\\0 & 1 & 0 & \delta_y\\0 & 0 & 1 & \delta_z\\0 & 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 1\\0 & 0 & 1 & 1\\0 & 0 & 0 & 1\end{bmatrix}$$

Scaling the object by vector σ

$$S_\sigma = \begin{bmatrix}\sigma_x & 0 & 0 & 0\\0 & \sigma_y & 0 & 0\\0 & 0 & \sigma_z & 0\\0 & 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}0.7 & 0 & 0 & 0\\0 & 0.7 & 0 & 0\\0 & 0 & 0.7 & 0\\0 & 0 & 0 & 1\end{bmatrix}$$

Rotating the object around the x, y and z axes by the angles given by θ

$$\begin{aligned}
R_\theta &= R_{\theta_z} R_{\theta_y} R_{\theta_x}\\
&= \begin{bmatrix}\cos(\theta_z) & -\sin(\theta_z) & 0 & 0\\\sin(\theta_z) & \cos(\theta_z) & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos(\theta_y) & 0 & \sin(\theta_y) & 0\\0 & 1 & 0 & 0\\-\sin(\theta_y) & 0 & \cos(\theta_y) & 0\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & 0\\0 & \cos(\theta_x) & -\sin(\theta_x) & 0\\0 & \sin(\theta_x) & \cos(\theta_x) & 0\\0 & 0 & 0 & 1\end{bmatrix}\\
&= \begin{bmatrix}\cos(\theta_z) & -\sin(\theta_z) & 0 & 0\\\sin(\theta_z) & \cos(\theta_z) & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos(\theta_y) & \sin(\theta_x)\sin(\theta_y) & \sin(\theta_y)\cos(\theta_x) & 0\\0 & \cos(\theta_x) & -\sin(\theta_x) & 0\\-\sin(\theta_y) & \sin(\theta_x)\cos(\theta_y) & \cos(\theta_x)\cos(\theta_y) & 0\\0 & 0 & 0 & 1\end{bmatrix}\\
&= \begin{bmatrix}\cos(\theta_y)\cos(\theta_z) & \sin(\theta_x)\sin(\theta_y)\cos(\theta_z) - \sin(\theta_z)\cos(\theta_x) & \sin(\theta_x)\sin(\theta_z) + \sin(\theta_y)\cos(\theta_x)\cos(\theta_z) & 0\\\sin(\theta_z)\cos(\theta_y) & \sin(\theta_x)\sin(\theta_y)\sin(\theta_z) + \cos(\theta_x)\cos(\theta_z) & -\sin(\theta_x)\cos(\theta_z) + \sin(\theta_y)\sin(\theta_z)\cos(\theta_x) & 0\\-\sin(\theta_y) & \sin(\theta_x)\cos(\theta_y) & \cos(\theta_x)\cos(\theta_y) & 0\\0 & 0 & 0 & 1\end{bmatrix}\\
&= \begin{bmatrix}\cos 0.2\cos 0.1 & \sin 0.1\sin 0.2\cos 0.1 - \sin 0.1\cos 0.1 & \sin 0.1\sin 0.1 + \sin 0.2\cos 0.1\cos 0.1 & 0\\\sin 0.1\cos 0.2 & \sin 0.1\sin 0.2\sin 0.1 + \cos 0.1\cos 0.1 & -\sin 0.1\cos 0.1 + \sin 0.2\sin 0.1\cos 0.1 & 0\\-\sin 0.2 & \sin 0.1\cos 0.2 & \cos 0.1\cos 0.2 & 0\\0 & 0 & 0 & 1\end{bmatrix}\\
&\approx \begin{bmatrix}0.98 & -0.08 & 0.21 & 0\\0.1 & 0.99 & -0.08 & 0\\-0.2 & 0.1 & 0.98 & 0\\0 & 0 & 0 & 1\end{bmatrix}
\end{aligned}$$

1. b) Matrix multiplication concatenation of the operations

The order of transformations specified is translation, scaling and finally the three rotations in the x, y and z axes. When concatenated mathematically, these operations are carried out from right to left as follows.

$$M_{model} = R_\theta S_\sigma T_\delta = R_{\theta_z} R_{\theta_y} R_{\theta_x} S_\sigma T_\delta$$

1. c) Combined 4×4 transformation matrix M_{model}

Multiply the three transformations together as specified. R_θ's entries are written using indexing as r_{ij} for conciseness.

$$\begin{aligned}
M_{model} &= R_\theta S_\sigma T_\delta \\
&= \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_x & 0 & 0 & 0 \\ 0 & \sigma_y & 0 & 0 \\ 0 & 0 & \sigma_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \delta_x \\ 0 & 1 & 0 & \delta_y \\ 0 & 0 & 1 & \delta_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_x & 0 & 0 & \delta_x\sigma_x \\ 0 & \sigma_y & 0 & \delta_y\sigma_y \\ 0 & 0 & \sigma_z & \delta_z\sigma_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} r_{11}\sigma_x & r_{12}\sigma_y & r_{13}\sigma_z & \delta_x r_{11}\sigma_x + \delta_y r_{12}\sigma_y + \delta_z r_{13}\sigma_z \\ r_{21}\sigma_x & r_{22}\sigma_y & r_{23}\sigma_z & \delta_x r_{21}\sigma_x + \delta_y r_{22}\sigma_y + \delta_z r_{23}\sigma_z \\ r_{31}\sigma_x & r_{32}\sigma_y & r_{33}\sigma_z & \delta_x r_{31}\sigma_x + \delta_y r_{32}\sigma_y + \delta_z r_{33}\sigma_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} 0.68 & -0.06 & 0.14 & 0.09 \\ 0.07 & 0.69 & -0.06 & 0.64 \\ -0.14 & 0.07 & 0.68 & 0.75 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

1. d) The homogeneous coordinates of all points after the combined transformation

Multiply the original coordinates by the model matrix to get the model coordinates.

$$\begin{aligned}
V' &= \begin{bmatrix} P1' & P2' & \cdots & P10' \end{bmatrix} \\
&= M_{model} \begin{bmatrix} P1 & P2 & \cdots & P10 \end{bmatrix} \\
&= \begin{bmatrix} r_{11}\sigma_x & r_{12}\sigma_y & r_{13}\sigma_z & \delta_x r_{11}\sigma_x + \delta_y r_{12}\sigma_y + \delta_z r_{13}\sigma_z \\ r_{21}\sigma_x & r_{22}\sigma_y & r_{23}\sigma_z & \delta_x r_{21}\sigma_x + \delta_y r_{22}\sigma_y + \delta_z r_{23}\sigma_z \\ r_{31}\sigma_x & r_{32}\sigma_y & r_{33}\sigma_z & \delta_x r_{31}\sigma_x + \delta_y r_{32}\sigma_y + \delta_z r_{33}\sigma_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -2 & 2 & 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -0.5 & -0.5 \\ 3 & 3 & 3 & 3 & -3 & -3 & -3 & -3 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} -0.79 & 1.94 & 1.83 & -0.9 & -1.65 & 1.08 & 0.96 & -1.77 & -1.39 & 1.34 \\ -0.36 & -0.09 & 1.3 & 1.03 & -0.03 & 0.25 & 1.64 & 1.36 & 0.21 & 0.48 \\ 3.01 & 2.45 & 2.59 & 3.15 & -1.09 & -1.64 & -1.51 & -0.95 & 0.31 & -0.24 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{aligned}$$

2 Geometry

2. a) Expressing transformed points as P1', P2', P3', P4' and P5' as Cartesian vectors

Simply ignore the fourth entry of each homogeneous coordinate.

$$P1 \approx \begin{bmatrix} -0.79 \\ -0.36 \\ 3.01 \end{bmatrix}, P2 \approx \begin{bmatrix} 1.94 \\ -0.09 \\ 2.45 \end{bmatrix}, P3 \approx \begin{bmatrix} 1.83 \\ 1.3 \\ 2.59 \end{bmatrix}, P4 \approx \begin{bmatrix} -0.9 \\ 1.03 \\ 3.15 \end{bmatrix}, P5 \approx \begin{bmatrix} -1.65 \\ -0.03 \\ -1.09 \end{bmatrix}.$$

2. b) Expressing the three lines from P1' to P2', P4' and P5' as Cartesian vectors v1, v2 and v3

Subtract the initial point from the target point to get the vector line between them.

$$v_1 = P2' - P1' \approx \begin{bmatrix} 1.94 \\ -0.09 \\ 2.45 \end{bmatrix} - \begin{bmatrix} -0.79 \\ -0.36 \\ 3.01 \end{bmatrix} = \begin{bmatrix} 2.73 \\ 0.27 \\ -0.56 \end{bmatrix}$$

$$v_2 = P4' - P1' \approx \begin{bmatrix} -0.9 \\ 1.03 \\ 3.15 \end{bmatrix} - \begin{bmatrix} -0.79 \\ -0.36 \\ 3.01 \end{bmatrix} = \begin{bmatrix} -0.11 \\ 1.39 \\ 0.14 \end{bmatrix}$$

$$v_3 = P5' - P1' \approx \begin{bmatrix} -1.65 \\ -0.03 \\ -1.09 \end{bmatrix} - \begin{bmatrix} -0.79 \\ -0.36 \\ 3.01 \end{bmatrix} = \begin{bmatrix} -0.87 \\ 0.33 \\ -4.1 \end{bmatrix}$$

2. c) Showing that vectors v1, v2 and v3 are all orthogonal to each other

Two vectors are orthogonal if their dot product is equal to 0.

$$\begin{aligned} v_1 \cdot v_2 &\approx \begin{bmatrix} 2.73 \\ 0.27 \\ -0.56 \end{bmatrix} \cdot \begin{bmatrix} -0.11 \\ 1.39 \\ 0.14 \end{bmatrix} \\ &= 2.73 \times -0.11 + 0.27 \times 1.39 \ - 0.56 \times 0.14 \\ &\approx -0.30 + 0.38 - 0.08 \\ &= 0 \end{aligned}$$

$$\begin{aligned} v_1 \cdot v_3 &\approx \begin{bmatrix} 2.73 \\ 0.27 \\ -0.56 \end{bmatrix} \cdot \begin{bmatrix} -0.87 \\ 0.33 \\ -4.1 \end{bmatrix} \\ &= 2.73 \times -0.87 + 0.27 \times 0.33 + -0.56 \times -4.1 \\ &\approx -2.38 + 0.09 + 2.30 \\ &\approx 0 \end{aligned}$$

$$\begin{aligned} v_2 \cdot v_3 &\approx \begin{bmatrix} -0.11 \\ 1.39 \\ 0.14 \end{bmatrix} \cdot \begin{bmatrix} -0.87 \\ 0.33 \\ -4.1 \end{bmatrix} \\ &= -0.11 \times -0.87 + 1.39 \times 0.33 + 0.14 \times -4.1 \\ &\approx 0.10 + 0.46 - 0.57 \\ &\approx 0 \end{aligned}$$

2. d) Showing that transformed points P1', P2', P3' and P4' are coplanar (all lie on the same plane)

First, get an equation for the plane. This can be found by taking the dot product of the normal with a point on the plane. The normal can be found by taking the cross product of the vectors v_1 and v_2.

$$\begin{aligned} n \approx \begin{bmatrix} 2.73 \\ 0.27 \\ -0.56 \end{bmatrix} \times \begin{bmatrix} -0.11 \\ 1.39 \\ 0.14 \end{bmatrix} &= \begin{bmatrix} 0.27 \times 0.14 - (-0.56) \times 1.39 \\ -0.56 \times -0.11 - 2.73 \times 0.14 \\ 2.73 \times 1.39 - 0.27 \times (-0.11) \end{bmatrix} \\ &= \begin{bmatrix} 0.27 \times 0.14 - (-0.56) \times 1.39 \\ -0.56 \times -0.11 - 2.73 \times 0.14 \\ 2.73 \times 1.39 - 0.27 \times (-0.11) \end{bmatrix} \\ &\approx \begin{bmatrix} 0.04 + 0.78 \\ 0.06 - 0.38 \\ 3.79 + 0.03 \end{bmatrix} \\ &= \begin{bmatrix} 0.82 \\ -0.32 \\ 3.82 \end{bmatrix} \approx \begin{bmatrix} 0.81 \\ -0.31 \\ 3.82 \end{bmatrix} \text{ (Computed)} \end{aligned}$$

Substituting in a point on the plane, such as P1', we can then write the equation of the plane in Cartesian form.

$$\begin{aligned} 0.81x - 0.31y + 3.82z &= 0.81 \times -0.79 - 0.31 \times -0.36 + 3.82 \times 3.01 \\ &\approx -0.64 + 0.11 + 11.50 \\ &= 10.97 \approx 10.98 \text{ (Computed)} \end{aligned}$$

Given that $v1$ and $v2$ were used to find the normal and they are composed of points $P1'$, $P2'$ and $P4'$ respectively, we know that these three points are coplanar. We can check if $P3'$ also lies on this plane by substituting its Cartesian coordinates into the equation of the plane.

$$\begin{aligned} 0.81x - 0.31y + 3.82z &= 0.81 \times 1.83 - 0.31 \times 1.3 + 3.82 \times 2.59 \\ &\approx 1.48 - 0.40 + 9.89 \\ &= 10.97 \approx 10.98 \text{ (Computed)} \end{aligned}$$

Given that substituting in $P3'$ satisfies the equation of the plane when accounting for rounding error, $P1'$, $P2'$, $P3'$ and $P4'$ must be coplanar.

3 Camera positioning

3. a) Show the individual transformations required to position the camera in terms of variable names and angles

The camera matrix is defined in the question as a transformation followed by a rotation in the x and then another in the y axis. The matrices used are standard transformation matrices with the transformation values substituted.

Translating the object by vector d_c in the z axis

$$T_{camera} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_c \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotating the object around the x and y axes by θ_c and ϕ_c

$$\begin{aligned}
R_{camera} &= R_{\phi_c} R_{\theta_c} = R_{y_c} R_{x_c} \\
&= \begin{bmatrix} \cos(\phi_c) & 0 & \sin(\phi_c) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\phi_c) & 0 & \cos(\phi_c) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_c) & -\sin(\theta_c) & 0 \\ 0 & \sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(\phi_c) & \sin(\phi_c)\sin(\theta_c) & \sin(\phi_c)\cos(\theta_c) & 0 \\ 0 & \cos(\theta_c) & -\sin(\theta_c) & 0 \\ -\sin(\phi_c) & \sin(\theta_c)\cos(\phi_c) & \cos(\phi_c)\cos(\theta_c) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(0.07) & \sin(0.07)\sin(0.15) & \sin(0.07)\cos(0.15) & 0 \\ 0 & \cos(0.15) & -\sin(0.15) & 0 \\ -\sin(0.07) & \sin(0.15)\cos(0.07) & \cos(0.07)\cos(0.15) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} 1.0 & 0.01 & 0.07 & 0 \\ 0 & 0.99 & -0.15 & 0 \\ -0.07 & 0.15 & 0.99 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Full Camera Matrix

$$
\begin{aligned}
M_{camera} &= R_{camera}T_{camera} = R_{y_c}R_{x_c}T_{d_c} = R_{\phi_c}R_{\theta_c}T_{d_c} \\
&= \begin{bmatrix} \cos(\phi_c) & \sin(\phi_c)\sin(\theta_c) & \sin(\phi_c)\cos(\theta_c) & 0 \\ 0 & \cos(\theta_c) & -\sin(\theta_c) & 0 \\ -\sin(\phi_c) & \sin(\theta_c)\cos(\phi_c) & \cos(\phi_c)\cos(\theta_c) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(\phi_c) & \sin(\phi_c)\sin(\theta_c) & \sin(\phi_c)\cos(\theta_c) & d_c\sin(\phi_c)\cos(\theta_c) \\ 0 & \cos(\theta_c) & -\sin(\theta_c) & -d_c\sin(\theta_c) \\ -\sin(\phi_c) & \sin(\theta_c)\cos(\phi_c) & \cos(\phi_c)\cos(\theta_c) & d_c\cos(\phi_c)\cos(\theta_c) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(0.07) & \sin(0.07)\sin(0.15) & \sin(0.07)\cos(0.15) & 12\sin(0.07)\cos(0.15) \\ 0 & \cos(0.15) & -\sin(0.15) & -12\sin(0.15) \\ -\sin(0.07) & \sin(0.15)\cos(0.07) & \cos(0.07)\cos(0.15) & 12\cos(0.07)\cos(0.15) \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} 1.0 & 0.01 & 0.07 & 0.83 \\ 0 & 0.99 & -0.15 & -1.79 \\ -0.07 & 0.15 & 0.99 & 11.84 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

3. b) The combined 4×4 transformation matrix M_{view} that performs all operations in the correct order

The camera transformation matrix can be inverted to obtain the view matrix, which applies the transformations in the correct order (in this case, one translation and two rotations).

$$
\begin{aligned}
M_{view} &= M_{camera}^{-1} = T_{camera}^{-1}R_{x_c}^{-1}R_{y_c}^{-1} = T_{camera}^{-1}R_{\theta_c}^{-1}R_{\phi_c}^{-1} = T_{view}R_{view} \\
&= \begin{bmatrix} \cos(\phi_c) & \sin(\phi_c)\sin(\theta_c) & \sin(\phi_c)\cos(\theta_c) & d_c\sin(\phi_c)\cos(\theta_c) \\ 0 & \cos(\theta_c) & -\sin(\theta_c) & -d_c\sin(\theta_c) \\ -\sin(\phi_c) & \sin(\theta_c)\cos(\phi_c) & \cos(\phi_c)\cos(\theta_c) & d_c\cos(\phi_c)\cos(\theta_c) \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} \cos(0.07) & \sin(0.07)\sin(0.15) & \sin(0.07)\cos(0.15) & 12\sin(0.07)\cos(0.15) \\ 0 & \cos(0.15) & -\sin(0.15) & -12\sin(0.15) \\ -\sin(0.07) & \sin(0.15)\cos(0.07) & \cos(0.07)\cos(0.15) & 12\cos(0.07)\cos(0.15) \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\
&\approx \begin{bmatrix} 1.0 & 0.01 & 0.07 & 0.83 \\ 0 & 0.99 & -0.15 & -1.79 \\ -0.07 & 0.15 & 0.99 & 11.84 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\
&\approx \begin{bmatrix} 1.0 & 0 & -0.07 & 0 \\ 0.01 & 0.99 & 0.15 & 0 \\ 0.07 & -0.15 & 0.99 & -12.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

3. c) The combined 4×4 model-view matrix that combines M_{model} and M_{view}

Simply concatenate the two matrices together in the correct order.

$$
\begin{aligned}
M_{mv} &= M_{view} M_{model} \\
&\approx \begin{bmatrix} 1.0 & 0 & -0.07 & 0 \\ 0.01 & 0.99 & 0.15 & 0 \\ 0.07 & -0.15 & 0.99 & -12.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.68 & -0.06 & 0.14 & 0.09 \\ 0.07 & 0.69 & -0.06 & 0.64 \\ -0.14 & 0.07 & 0.68 & 0.75 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} 0.69 & -0.06 & 0.1 & 0.04 \\ 0.05 & 0.7 & 0.05 & 0.74 \\ -0.1 & -0.04 & 0.69 & -11.35 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

3. d) The homogeneous coordinates of all points after the model-view matrix transformation

Multiply the original coordinates by the model-view matrix to get the model-view coordinates.

$$
\begin{aligned}
V'' &= \begin{bmatrix} P1'' & P2'' & \cdots & P10'' \end{bmatrix} \\
&= M_{view} \begin{bmatrix} P1' & P2' & \cdots & P10' \end{bmatrix} \\
&= M_{mv} \begin{bmatrix} P1 & P2 & \cdots & P10 \end{bmatrix} \\
&\approx \begin{bmatrix} 0.69 & -0.06 & 0.1 & 0.04 \\ 0.05 & 0.7 & 0.05 & 0.74 \\ -0.1 & -0.04 & 0.69 & -11.35 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -2 & 2 & 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -0.5 & -0.5 \\ 3 & 3 & 3 & 3 & -3 & -3 & -3 & -3 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} -1.0 & 1.77 & 1.65 & -1.12 & -1.57 & 1.19 & 1.07 & -1.7 & -1.41 & 1.35 \\ 0.08 & 0.3 & 1.69 & 1.48 & -0.2 & 0.01 & 1.4 & 1.19 & 0.24 & 0.46 \\ -9.03 & -9.43 & -9.51 & -9.11 & -13.18 & -13.58 & -13.66 & -13.26 & -11.82 & -12.22 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{aligned}
$$

4 In-place manipulation

To create a transformation matrix that scales the object in place, we must first reset the object to the origin. Then we can scale the object and finally undo the reset. In this instance, we take the inverse of the model-view matrix, scale the object and then re-apply the model-view matrix. To simplify the symbolic calculation of the inverse of the model-view matrix, we can break it down into the inverse of the model matrix and the camera matrix.

$$
\begin{aligned}
M_{scaleIP} &= M_{mv} M_{scale} M_{mv}^{-1} = M_{mv} M_{scale} M_{model}^{-1} M_{camera} \\
&\approx \begin{bmatrix} 0.69 & -0.06 & 0.1 & 0.04 \\ 0.05 & 0.7 & 0.05 & 0.74 \\ -0.1 & -0.04 & 0.69 & -11.35 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1.41 & 0.11 & -0.2 & -2.45 \\ -0.12 & 1.42 & -0.08 & -1.98 \\ 0.2 & 0.1 & 1.41 & 15.94 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} 1.97 & 0.08 & -0.14 & -1.69 \\ 0.08 & 1.01 & -0.01 & -0.13 \\ -0.14 & -0.01 & 1.02 & 0.25 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Applying this transformation matrix to the transformed points then scales the object in place.

$$
\begin{aligned}
V''' &= \begin{bmatrix} P1''' & P2''' & \cdots & P10''' \end{bmatrix} \\
&= M_{scaleIP} \begin{bmatrix} P1'' & P2'' & \cdots & P10'' \end{bmatrix} \\
&\approx \begin{bmatrix} 1.97 & 0.08 & -0.14 & -1.69 \\ 0.08 & 1.01 & -0.01 & -0.13 \\ -0.14 & -0.01 & 1.02 & 0.25 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\times \begin{bmatrix} -1.0 & 1.77 & 1.65 & -1.12 & -1.57 & 1.19 & 1.07 & -1.7 & -1.41 & 1.35 \\ 0.08 & 0.3 & 1.69 & 1.48 & -0.2 & 0.01 & 1.4 & 1.19 & 0.24 & 0.46 \\ -9.03 & -9.43 & -9.51 & -9.11 & -13.18 & -13.58 & -13.66 & -13.26 & -11.82 & -12.22 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\
&\approx \begin{bmatrix} -2.38 & 3.15 & 3.03 & -2.5 & -2.96 & 2.57 & 2.45 & -3.08 & -2.79 & 2.73 \\ -0.02 & 0.41 & 1.8 & 1.37 & -0.31 & 0.12 & 1.51 & 1.08 & 0.13 & 0.56 \\ -8.83 & -9.63 & -9.71 & -8.91 & -12.98 & -13.78 & -13.86 & -13.06 & -11.62 & -12.42 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{aligned}
$$

We can check the approximate distance between P1 and P2 as well as P1 and P4 to ensure our calculations are correct. As seen below, the distance between P1 and P4 stays the same while the distance between P1 and P2 doubles, as expected.

```
P1P4Original: 2. P1P4Model: 1.40. P1P4ModelView: 1.41. P1P4ScaledModelView: 1.40.
P1P2Original: 4. P1P2Model: 2.80. P1P2ModelView: 2.81. P1P2ScaledModelView: 5.60.
```

5 Advanced tasks

5. a) Expressing camera coordinate axes in world coordinate axes

From lectures, the camera matrix converts world coordinates into camera coordinates. As such, the camera coordinate axes can be read as the columns from the camera matrix.

$$M_{camera} \approx \begin{bmatrix} 1.0 & 0.01 & 0.07 & 0.83 \\ 0 & 0.99 & -0.15 & -1.79 \\ -0.07 & 0.15 & 0.99 & 11.84 \\ 0 & 0 & 0 & 1 \end{bmatrix} \implies v_1^c \approx \begin{bmatrix} 1.0 \\ 0 \\ -0.07 \end{bmatrix}, v_2^c \approx \begin{bmatrix} 0.01 \\ 0.99 \\ 0.15 \end{bmatrix}, v_3^c \approx \begin{bmatrix} 0.07 \\ -0.15 \\ 0.99 \end{bmatrix}.$$

To return to the world coordinates, we can multiply camera coordinates by the inverse of the camera matrix (i.e. the view matrix). The world coordinate axes can then be read as the columns of the resultant matrix, which is the identity matrix.

$$M_{world} = I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \implies v_1^w = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_2^w = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, v_3^w = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

5. b) Calculating and applying a suitable perspective matrix

From lectures, a suitable perspective matrix can be found given the bounds of the view volume (frustum between the near and far plane). We can define the bounds by finding the minimum and maximum point in each dimension in the camera coordinates after applying all our previous transformations. In the z dimension we need to use the opposite operations and flip the sign to get a positive distance from the camera. These steps give the following bounds:

$$\begin{bmatrix} x_{min} & x_{max} \\ y_{min} & y_{max} \\ -z_{max} & -z_{min} \end{bmatrix} = \begin{bmatrix} \text{left} & \text{right} \\ \text{bottom} & \text{top} \\ \text{near} & \text{far} \end{bmatrix} \approx \begin{bmatrix} -3.08 & 3.15 \\ -0.31 & 1.8 \\ 8.83 & 13.86 \end{bmatrix}$$

From this, we can define a perspective matrix:

$$NSH = \begin{bmatrix} \frac{2\times\text{near}}{\text{right}-\text{left}} & 0 & \frac{\text{right}+\text{left}}{\text{right}-\text{left}} & 0 \\ 0 & \frac{2\times\text{near}}{\text{top}-\text{bottom}} & \frac{\text{top}+\text{bottom}}{\text{top}-\text{bottom}} & 0 \\ 0 & 0 & -\frac{\text{far}+\text{near}}{\text{far}-\text{near}} & \frac{-2\times\text{far}\times\text{near}}{\text{far}-\text{near}} \\ 0 & 0 & -1 & 0 \end{bmatrix} \approx \begin{bmatrix} 2.83 & 0 & 0.0112 & 0 \\ 0 & 8.37 & 0.706 & 0 \\ 0 & 0 & -4.511 & -48.66 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

By applying this matrix to the transformed points, we get the non-normalised perspective transformed points.

$$\begin{bmatrix} -6.83 & 8.82 & 8.47 & -7.18 & -8.52 & 7.13 & 6.79 & -8.87 & -8.05 & 7.61 \\ -6.44 & -3.38 & 8.22 & 5.16 & -11.78 & -8.73 & 2.87 & -0.19 & -7.1 & -4.04 \\ -8.82 & -5.2 & -4.84 & -8.46 & 9.9 & 13.52 & 13.88 & 10.26 & 3.75 & 7.37 \\ 8.83 & 9.63 & 9.71 & 8.91 & 12.98 & 13.78 & 13.86 & 13.06 & 11.62 & 12.42 \end{bmatrix}$$

Each point can be normalised by dividing by its weight w, which corresponds to the fourth entry. This gives the final perspective transformed points P'''':

$$\begin{bmatrix} -0.77 & 0.92 & 0.87 & -0.81 & -0.66 & 0.52 & 0.49 & -0.68 & -0.69 & 0.61 \\ -0.73 & -0.35 & 0.85 & 0.58 & -0.91 & -0.63 & 0.21 & -0.01 & -0.61 & -0.33 \\ -1.0 & -0.54 & -0.5 & -0.95 & 0.76 & 0.98 & 1.0 & 0.79 & 0.32 & 0.59 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Rendering the shape with these vertices in WebGL gives the image shown in Figure 1. This shows the front (white) and bottom (blue) faces of the shape with some rotation and perspective foreshortening.

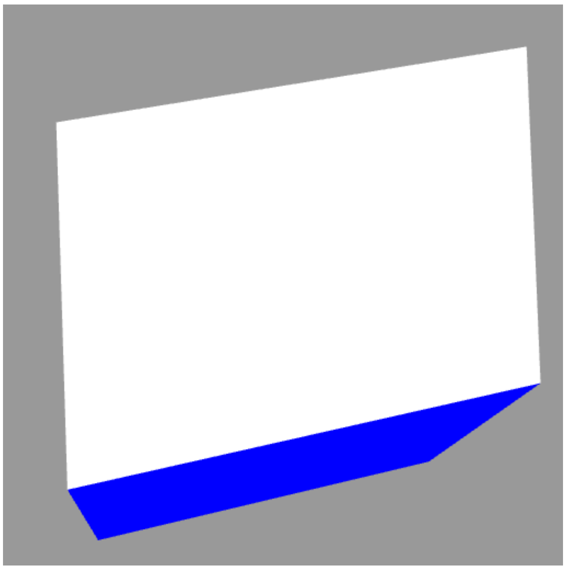

Figure 1: The shape with all transformations applied