

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет систем управления и робототехники

ОТЧЕТ
по дисциплине
«Практическая линейная алгебра»

по теме:
3D ПРЕОБРАЗОВАНИЯ

Студент:
Группа № R3435

Зыкин Л. В.

Предподаватель:
техник, ассистент

Догадин Е. В.

Санкт-Петербург
2025

СОДЕРЖАНИЕ

1	ХОД ВЫПОЛНЕНИЯ РАБОТЫ	3
1.1	Задание 1. Построение куба в пространстве	3
1.2	Задание 2. Масштабирование куба	4
1.3	Задание 3. Перенос куба	5
1.4	Задание 4. Поворот куба	6
1.5	Задание 5. Поворот куба вокруг вершины	8
1.6	Задание 6. Виртуальная камера	9
1.7	Задание 7. Перспективное отображение	10
1.8	Выводы	11

1 ХОД ВЫПОЛНЕНИЯ РАБОТЫ

1.1 Задание 1. Построение куба в пространстве

Для выполнения заданий используется код Python с библиотекой matplotlib для 3D визуализации.

Зачем использовать четырехкомпонентный вектор?

Четырехкомпонентный вектор (где последняя координата w) позволяет удобно записывать любые аффинные преобразования — такие как перенос, масштабирование и поворот — в виде одного матричного умножения. Благодаря этому все преобразования можно объединять и применять последовательно, не переходя к разным формулам для разных типов операций. После применения преобразования для получения обычных координат достаточно разделить x, y, z на w (если $w \neq 1$).

Как задать другие фигуры?

Для построения других многогранников достаточно изменить содержимое массивов координат вершин и индексов граней.

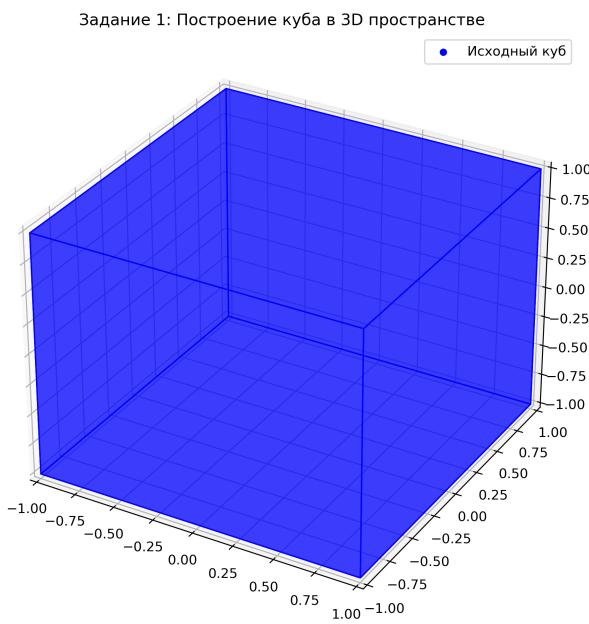


Рисунок 1 — Задание 1: Куб в трёхмерном пространстве

Задание 1: Пример тетраэдра в 3D пространстве

Тетраэдр

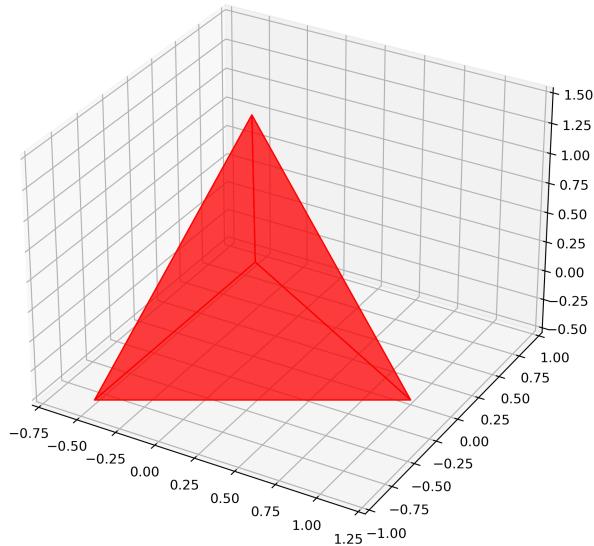


Рисунок 2 — Задание 1: Пример тетраэдра в трёхмерном пространстве

1.2 Задание 2. Масштабирование куба

Для изменения размеров куба используется матрица масштабирования:

$$S = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Здесь s_x , s_y , s_z — коэффициенты масштабирования по осям X , Y , Z соответственно. Применяя S к координатам вершин, получаем увеличенный или уменьшенный куб.

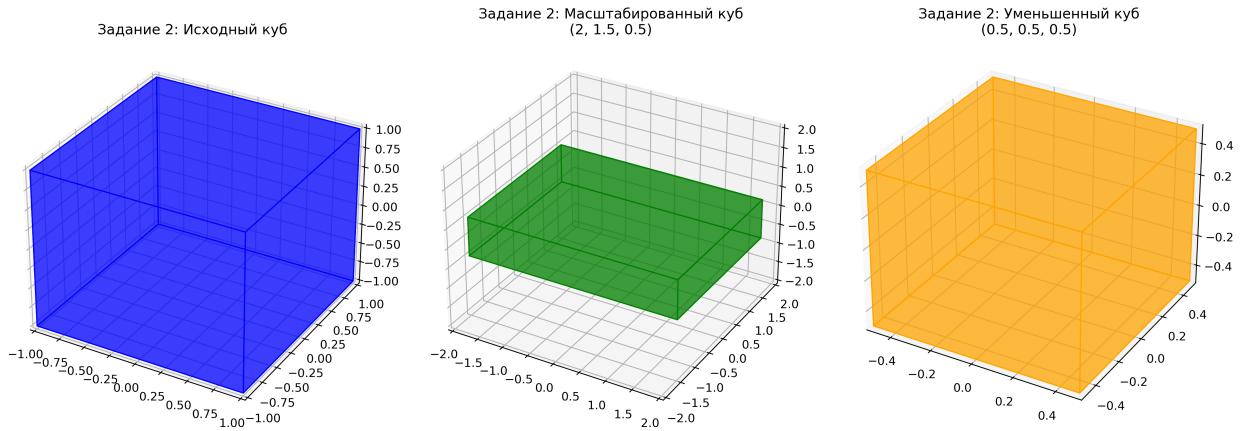


Рисунок 3 — Задание 2: Сравнение масштабирования куба: исходный, увеличенный (2, 1.5, 0.5) и уменьшенный (0.5, 0.5, 0.5)

1.3 Задание 3. Перенос куба

Для перемещения объекта в пространстве используется матрица сдвига:

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Здесь t_x, t_y, t_z — величины смещения по соответствующим осям. Итоговые координаты получаются умножением T на вектор вершин.

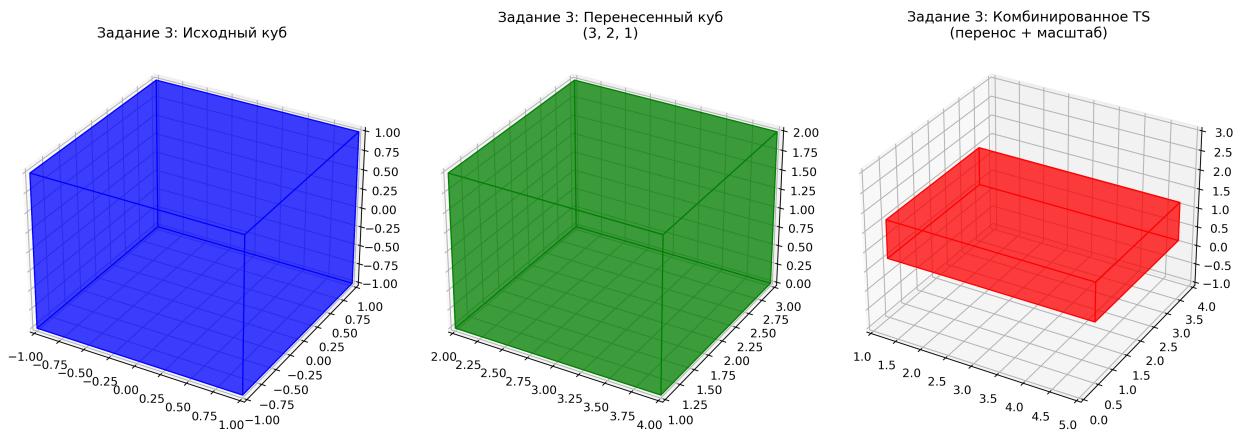


Рисунок 4 — Задание 3: Сравнение переноса куба: исходный, перенесенный (3, 2, 1) и комбинированное преобразование TS

Исследование коммутативности преобразований

Важно отметить, что порядок применения матриц имеет значение. Рассмотрим композиции TS (перенос, затем масштабирование) и ST (масштабирование, затем перенос):

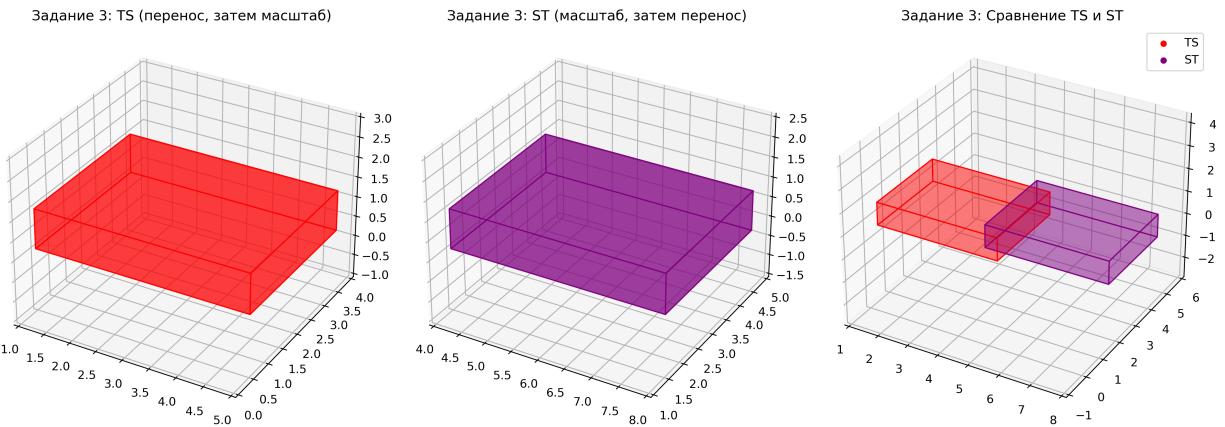


Рисунок 5 — Задание 3: Сравнение TS и ST: преобразования не коммутируют

Как видно из рисунка, $TS \neq ST$, что демонстрирует некоммутативность аффинных преобразований.

1.4 Задание 4. Поворот куба

Для поворота используются стандартные матрицы вращения вокруг осей:

– Вокруг X :

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

– Вокруг Y :

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Вокруг Z :

$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Поворот вокруг произвольной оси

Для поворота вокруг произвольной оси v на угол θ используется формула Родрига:

$$v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, \quad J = \frac{1}{\|v\|} \begin{bmatrix} 0 & -v_z & v_y & 0 \\ v_z & 0 & -v_x & 0 \\ -v_y & v_x & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R_v(\theta) = e^{J\theta}$$

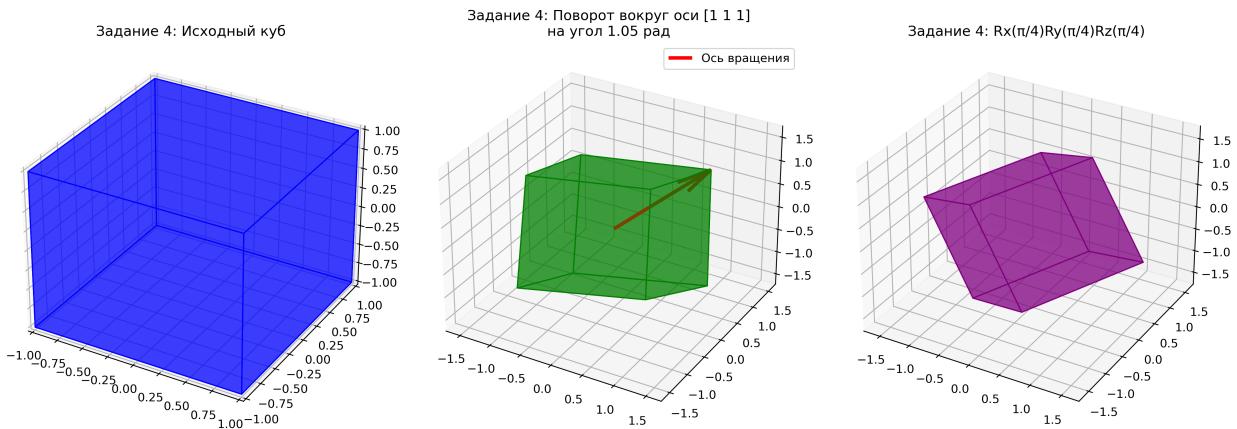


Рисунок 6 — Задание 4: Анализ поворотов: исходный куб, поворот вокруг произвольной оси $(1,1,1)$ и композиция поворотов вокруг координатных осей

Ответы на вопросы задания:

- Матрицы $R_x(\theta)$, $R_y(\phi)$, $R_z(\psi)$ получаются при выборе вектора v вдоль осей x , y или z соответственно.
- Тройка матриц $R_x(\theta)R_y(\phi)R_z(\psi)$ достаточна для описания всех возможных вращений в 3D пространстве (теорема Эйлера).
- Ось вращения можно восстановить, найдя собственный вектор матрицы поворота, соответствующий собственному значению 1.

1.5 Задание 5. Поворот куба вокруг вершины

Чтобы повернуть куб относительно одной из его вершин, нужно найти матрицу преобразования, которая осуществит поворот таким образом, что центр вращения будет совпадать с одной из его вершин.

Найдем матрицу преобразования:

Пусть P — координаты вершины, вокруг которой происходит поворот, R — матрица поворота вокруг оси, проходящей через начало координат. Тогда искомая матрица имеет вид:

$$M = T \cdot R \cdot T^{-1}$$

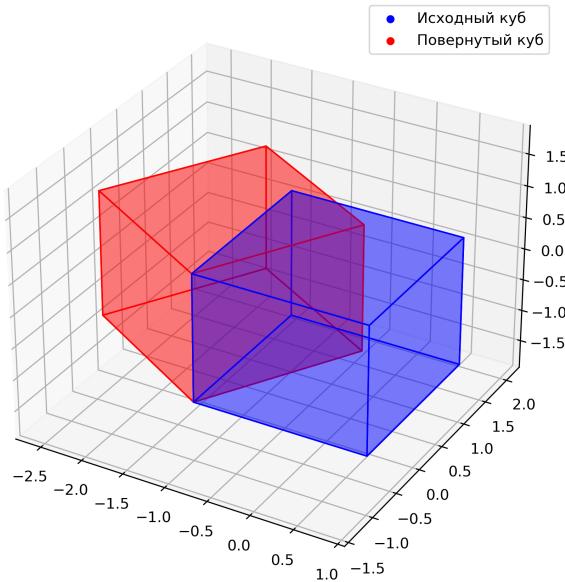
где T — матрица переноса, перемещающая вершину P в начало координат:

$$T = \begin{bmatrix} 1 & 0 & 0 & -P_x \\ 0 & 1 & 0 & -P_y \\ 0 & 0 & 1 & -P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T^{-1} — обратная матрица переноса:

$$T^{-1} = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Задание 5: Поворот куба вокруг вершины 0
на угол $\pi/3$ вокруг оси Z



Задание 5: Поворот куба вокруг вершины 0
на угол $\pi/2$ вокруг оси Z

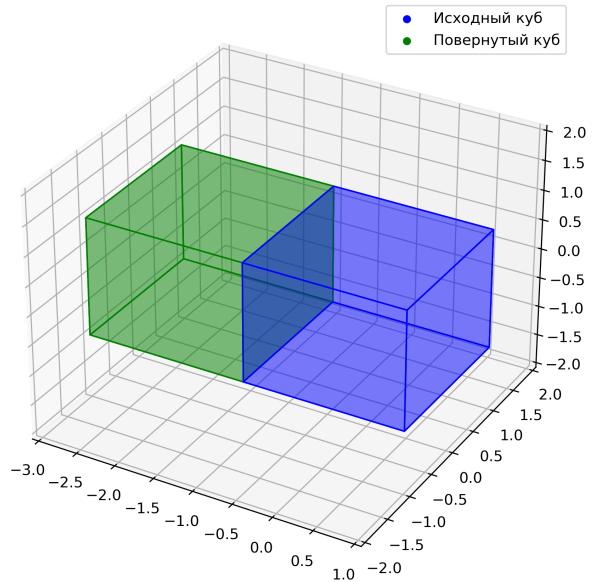


Рисунок 7 — Задание 5: Поворот куба вокруг вершины 0: на угол $\pi/3$ (слева) и $\pi/2$ (справа)
вокруг оси Z

1.6 Задание 6. Виртуальная камера

Для имитации движения камеры сцены используются матрицы вращения и матрица переноса камеры:

$$T_{camera} = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Здесь t_x, t_y, t_z — положение камеры. Отрицательный знак означает, что движение камеры в одну сторону эквивалентно движению сцены в противоположную.

Обратная матрица позволяет вернуть камеру в исходное положение.

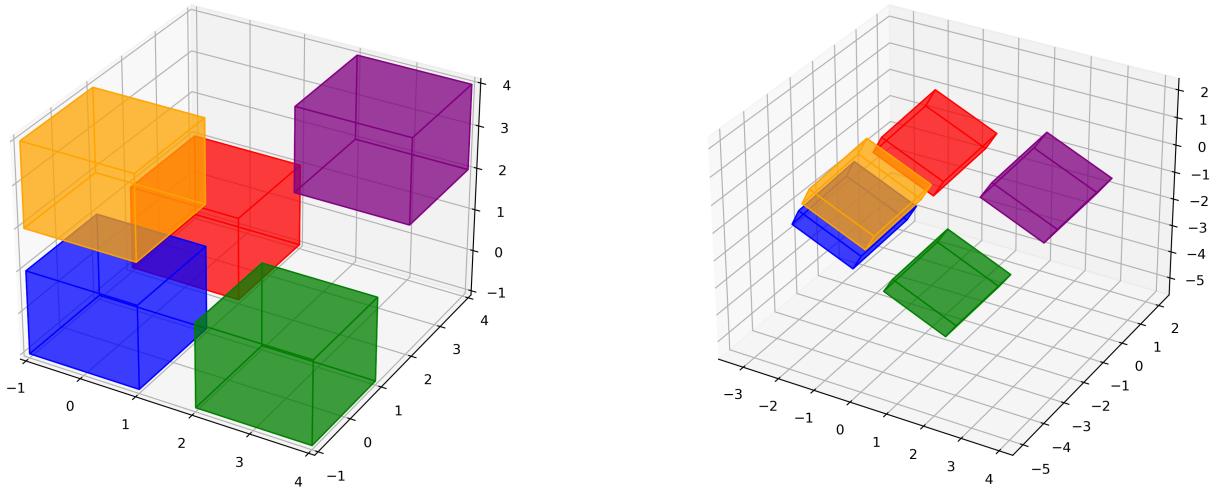


Рисунок 8 — Задание 6: Сцена с несколькими кубами: исходная (слева) и после применения камеры (справа)

1.7 Задание 7. Перспективное отображение

Для создания эффекта перспективы используется специальная матрица проекции:

$$M_{proj} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -1 \\ 0 & 0 & \frac{2fn}{f-n} & 0 \end{bmatrix}$$

Параметры:

- $s = \frac{1}{\tan(\frac{\text{fov}}{2})}$ — масштаб по X и Y (fov — угол обзора).
- f, n — дальняя и ближняя плоскости отсечения.
- Остальные элементы отвечают за нормализацию глубины и перспективу.

Пояснения:

Такая матрица позволяет корректно отображать удалённые объекты меньшими, чем ближние, и задаёт параметры видимости сцены (угол обзора, глубина, соотношение сторон).

Задание 7: Сцена без перспективы

Задание 7: Сцена с перспективной проекцией
(fov=60°, near=0.1, far=100)

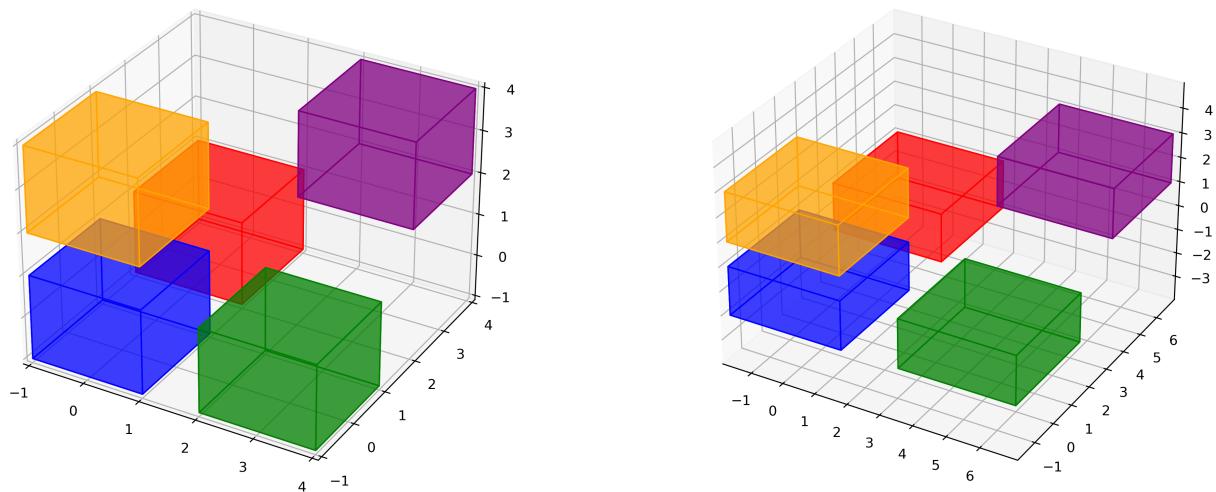


Рисунок 9 — Задание 7: Сравнение сцены без перспективы (слева) и с перспективной проекцией (справа)

Задание 7: Перспективная проекция - вид сбоку

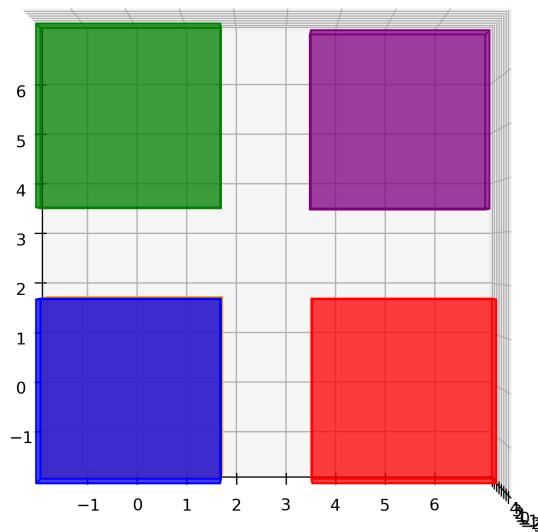


Рисунок 10 — Задание 7: Перспективная проекция - вид сбоку для демонстрации эффекта перспективы

1.8 Выводы

В ходе работы были реализованы и исследованы основные преобразования в 3D: масштабирование, перенос, поворот, а также перспективная проекция и виртуальная камера. Показано, как с помощью матриц можно гибко

управлять положением и видом объектов в пространстве, а также формировать реалистичное изображение сцены.

Ключевые результаты:

- Демонстрирована некоммутативность аффинных преобразований ($TS \neq ST$).
- Реализована формула Родрига для поворота вокруг произвольной оси.
- Показано, как комбинировать преобразования для сложных эффектов.
- Исследована перспективная проекция и её влияние на визуализацию 3D-сцен.