ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

канд. техн. наук, доцент		А. В. Аграновский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 3

ИССЛЕДОВАНИЕ АФФИННЫХ ПРЕОБРАЗОВАНИЙ В ПРОСТРАНСТВЕ

по курсу:

КОМПЬЮТЕРНАЯ ГРАФИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4326		Г. С. Томчук
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

1 Цель работы

Целью работы является Цель изучение теоретических основ линейных геометрических преобразований в трёхмерном пространстве и анализ особенностей практической реализации аффинных преобразований в трёхмерной графике.

2 Формулировка задания

Необходимо реализовать программу отражения икосаэдра относительно плоскости, не параллельной координатным плоскостям. Начало координат находится вне икосаэдра (вариант 20).

Результат преобразований представляется в экранных координатах. Для всех вариантов фигура должна отображаться в контурном виде без удаления невидимых линий. Рисование контура фигур по матрице координат вершин можно (но не обязательно) осуществлять с помощью специализированных библиотек. Аффинные преобразования необходимо выполнять только путем комбинации рассмотренных выше матричных вычислений (можно, но не обязательно, использовать библиотеки программ для матричных вычислений). Использование созданных другими авторами программ аффинных преобразований не допускается.

3 Теоретические сведения

Аффинные преобразования в трёхмерном пространстве — это преобразования, которые сохраняют прямолинейность объектов, параллельность линий и их относительное расположение, но могут изменять углы и размеры. Ключевая особенность аффинных преобразований — это возможность описания их с помощью матриц, что упрощает их применение на практике. Основные виды аффинных преобразований включают:

Сдвиг (трансляция) — перемещение фигуры на фиксированное расстояние вдоль одной или нескольких осей координат. Это преобразование задается вектором смещения и изменяет положение объекта без изменения его формы или размеров.

Масштабирование — изменение размеров объекта вдоль осей координат. При этом масштабирование может быть как однородным (равномерным по всем осям), так и неоднородным (разным для каждой оси), что может приводить к деформации объекта.

Поворот — вращение объекта вокруг одной из осей (или произвольной оси) на заданный угол. Ось и угол задают параметры преобразования, которое может использоваться для поворота объектов в трёхмерной сцене.

Отражение — зеркальное отображение объекта относительно плоскости. Это преобразование меняет ориентацию объекта, при этом его форма сохраняется, но может измениться направление осей.

Сжатие/растяжение (сдвиг) — преобразование, которое меняет форму объекта, сжимая или растягивая его вдоль оси или плоскости. Это может приводить к аффинным деформациям, когда объект изменяет свою форму, но не нарушает топологию.

Аффинные преобразования можно описать с помощью матрицы размером 4×4, что позволяет удобно комбинировать несколько преобразований в одну операцию. Например, последовательность поворота, масштабирования и сдвига можно представить в виде произведения

соответствующих матриц. Это делает матричный подход ключевым инструментом для работы с аффинными преобразованиями в компьютерной графике и 3D-моделировании.

4 Описание алгоритма решения

Инициализация данных. Задаём координаты вершин икосаэдра в пространстве. Для этого используем известные координаты икосаэдра, нормализованные для удобства работы. Определяем центр икосаэдра и смещаем его в произвольную точку пространства, чтобы начало координат находилось вне фигуры.

Определение плоскости отражения. Плоскость задаётся уравнением вида Ax+By+Cz+D=0, где A, B и C — компоненты нормального вектора к плоскости, а D — смещение плоскости от начала координат. На основе этого уравнения формируем нормальный вектор n=[A,B,C] и нормализуем его.

Построение матрицы отражения. Матрица отражения относительно плоскости вычисляется по формуле:

$$R = I - 2 \cdot \frac{nn^T}{n \cdot n}$$

где I — единичная матрица, а n^T — векторное произведение нормального вектора на самого себя. Это позволяет отразить фигуру относительно произвольной плоскости.

Смещение координат фигуры. Чтобы применить преобразование, сначала перемещаем икосаэдр так, чтобы его центр совпадал с началом координат. Это делается путём вычитания вектора центра фигуры из координат всех её вершин.

Применение матрицы отражения. К каждой вершине икосаэдра применяется матричное умножение с матрицей отражения. В результате этого шага каждая вершина изменяет своё положение согласно преобразованию.

Возвращение фигуры на исходную позицию. После применения отражения возвращаем икосаэдр обратно в исходное положение путём

прибавления вектора смещения, который был вычтен на шаге 4.

Визуализация результата. Используем графическую библиотеку, такую как matplotlib, для отображения результата. Фигура отображается в контурном виде, и линии рёбер не скрываются. Построение фигуры до и после отражения осуществляется в одной системе координат, чтобы наглядно увидеть разницу.

Отображение 3D-проекции на экране. Преобразуем трёхмерные координаты вершин в экранные координаты с использованием выбранной проекции (например, ортографической или перспективной) и выводим результат.

5 Выбор языка программирования и используемых библиотек

Для написания программы я выбрал язык программирования Python. Данный язык имеет богатую экосистему библиотек, в том числе и необходимых для работы с отрисовкой и графикой. Помимо этого, я уже имел опыт работы с данным языком, что ускорило и облегчило рабочий процесс.

Используемые библиотеки:

- NumPy — используется для работы с матрицами и векторами, что упрощает вычисления аффинных преобразований, таких как поворот, сдвиг и масштабирование. Это стандартная библиотека для эффективных математических операций в Python.
- Matplotlib — отвечает за визуализацию.

Обе библиотеки широко используются в научных и инженерных вычислениях, что делает их подходящими для задач, связанных с компьютерной графикой и аффинными преобразованиями.

6 Описание разработанной программы

Программа (см. Приложение А) выполняет отражение икосаэдра относительно произвольной плоскости в трёхмерном пространстве с использованием матричных преобразований.

Основные этапы программы:

Инициализация икосаэдра:

Программа задаёт координаты вершин икосаэдра и смещает его в

произвольную точку пространства, чтобы начало координат находилось вне фигуры.

Определение плоскости отражения:

Пользователь задаёт уравнение плоскости $Ax+By+Cz+D=0$, относительно которой будет происходить отражение.

Построение матрицы отражения:

На основе нормального вектора к плоскости вычисляется матрица отражения.

Преобразование координат:

Икосаэдр смещается в начало координат для удобства вычислений, после чего к его вершинам применяется матрица отражения.

Визуализация результата:

Программа отображает икосаэдр до и после преобразования в виде трёхмерной модели с контуром рёбер, используя библиотеку matplotlib.

Разработанная программа демонстрирует использование аффинных преобразований в трёхмерном пространстве с применением матричной алгебры и визуализирует результат преобразования.

7 Скриншоты, иллюстрирующие результаты работы программы

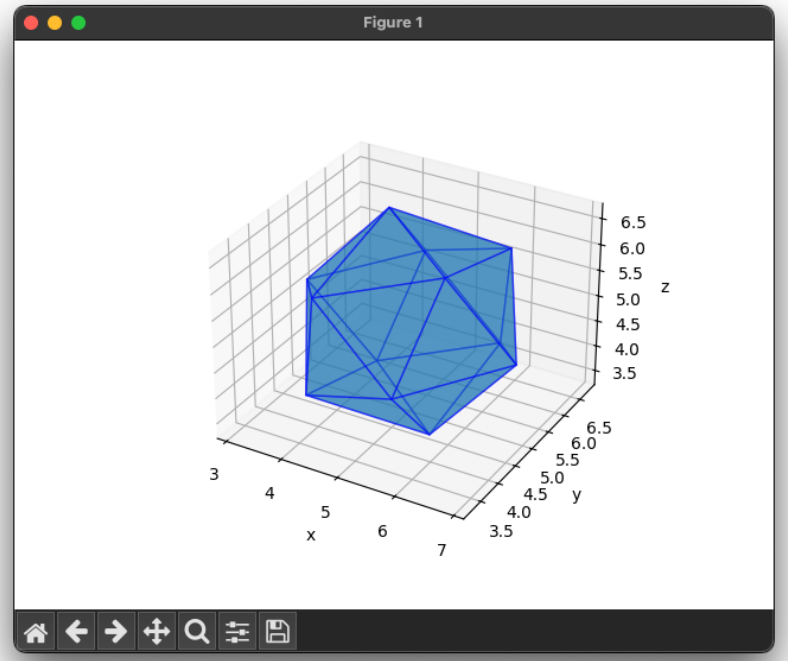

Рисунок 1

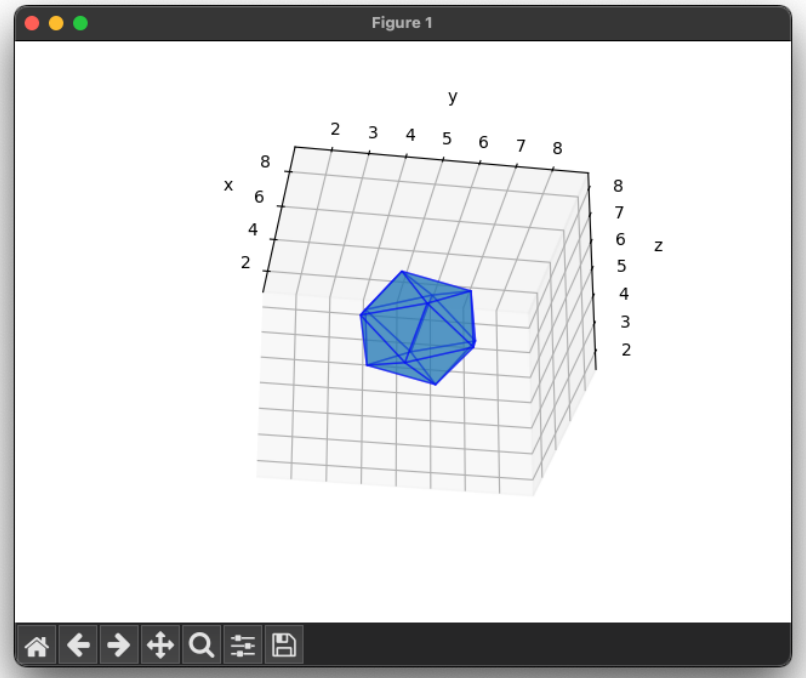

Рисунок 2

8 Вывод

В ходе выполнения работы были изучены теоретические основы аффинных преобразований в трёхмерном пространстве, включая их реализацию с помощью матричных операций. Разработанная программа позволила применить отражение икосаэдра относительно произвольной плоскости, наглядно демонстрируя принцип действия преобразований. Задача успешно показала важность использования матриц для выполнения линейных преобразований и их применения в задачах компьютерной графики.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аграновский, А. В. Использование методов преобразования координат для формирования растровых изображений. Учебно-методическое пособие / А. В. Аграновский. — СПб.: Редакционно-издательский центр ГУАП, 2024. — 40 с.
2. Бобылев, С. И. Основы компьютерной графики: учебное пособие / С. И. Бобылев. — М.: Издательство МГТУ им. Н. Э. Баумана, 2010. — 224 с.
3. Ласков, А. В. Математические основы компьютерной графики / А. В. Ласков, Ю. Г. Лоскутов. — СПб.: Питер, 2008. — 352 с.
4. NumPy Documentation [Электронный ресурс] — Режим доступа: https://numpy.org/doc/, свободный. — (дата обращения: 29.09.2024).

ПРИЛОЖЕНИЕ А

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.art3d import Poly3DCollection

# Вершины икосаэдра
phi = (1 + np.sqrt(5)) / 2  # ≈ 1.618 — золотая пропорция.
Используется для описания правильных многогранников, таких как
икосаэдр
vertices = np.array([
    [-1, phi, 0], [1, phi, 0], [-1, -phi, 0], [1, -phi, 0],
    [0, -1, phi], [0, 1, phi], [0, -1, -phi], [0, 1, -phi],
    [phi, 0, -1], [phi, 0, 1], [-phi, 0, -1], [-phi, 0, 1]
])

# Грани икосаэдра (связи между вершинами)
faces = [
    [0, 11, 5], [0, 5, 1], [0, 1, 7], [0, 7, 10], [0, 10, 11],
    [1, 5, 9], [5, 11, 4], [11, 10, 2], [10, 7, 6], [7, 1, 8],
    [3, 9, 4], [3, 4, 2], [3, 2, 6], [3, 6, 8], [3, 8, 9],
    [4, 9, 5], [2, 4, 11], [6, 2, 10], [8, 6, 7], [9, 8, 1]
]


# Функция отражения относительно произвольной плоскости
def reflection(vertices, A, B, C, D):
    # Матрица отражения относительно плоскости Ax + By + Cz + D = 0
    n = np.array([A, B, C])  # Нормальный вектор плоскости,
относительно которой происходит отражение
    n = n / np.linalg.norm(n)  # Нормализуем нормальный вектор
    reflection_matrix = np.eye(3) - 2 * np.outer(n, n)

    # Преобразуем вершины
    transformed_vertices = np.dot(vertices, reflection_matrix.T)
    return transformed_vertices


# Пример: отражение фигуры относительно плоскости 4x + 2y + 3z - 7 =
0
# Применяем отражение
transformed_vertices_centered = reflection(vertices, 4, 2, 3, -7)

center = np.array([5, 5, 5])  # Центр икосаэдра (координаты,
относительно которых будем его смещать)

# Смещаем фигуру от центра координат
transformed_vertices = transformed_vertices_centered + center

# Рисуем трансформированную фигуру
fig = plt.figure()  # Пространство, на котором будет рисоваться
график
ax = fig.add_subplot(111, projection='3d')  # Объект осей, на которых
рисуется график
```

```
ax.add_collection3d(Poly3DCollection([transformed_vertices[face] for
face in faces], alpha=0.5, edgecolor='b'))

ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')

plt.show()
```