Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2 по курсу «Компьютерная графика»

Студент: И. П. Моисеенков
Преподаватель: Г. С. Филиппов
Группа: М8О-308Б-19

Дата: 06.11.2021
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №2

Каркасная визуализация выпуклого многогранника. Удаление невидимых линий.

Задача: Разработать формат представления многогранника и процедуру его каркасной отрисовки в ортографической и изометрической проекциях. Обеспечить удаление невидимых линий и возможность пространственных поворотов и масштабирования многогранника.

Обеспечить автоматическое центрирование и изменение размеров изображения при изменении размеров окна.

Вариант 11: 10-гранная прямая правильная призма.

1 Описание

Для выполнения данного задания я воспользовался библиотекой matplotlib для python. Эта библиотека позволяет довольно просто работать с трехмерными изображениями.

Для построения призмы мне понадобилось задать список вершин (v) и список сторон (sides). Для отрисовки вершин используется функция scatter, для отрисовки сторон - Poly3DCollection.

Чтобы переключаться между различными проекциями, а также чтобы возможность показать и скрыть невидимые линии, я воспользовался кнопками (Button) из matplotlib.widgets. Для инициализации каждой кнопки нужно задать координаты ее расположения и функцию, которая будет выполняться при нажатии пользователем на кнопку. В моей программе есть 5 кнопок: показать/скрыть невидимые линии, показать изометрическую проекцию и показать две ортографические проекции.

Для удаления/отображения невидимых линий я просто манипулирую параметром alpha, который отвечает за степень прозрачности сторон фигуры. Чем меньше alpha, тем прозрачнее фигура. Если пользователь просит показать невидимые линии, то я устанавливаю alpha=0.5. Если их нужно скрыть - alpha=1.

Для отображения проекций фигуры используется функция view_init, которая устанавливает ракурс обзора.

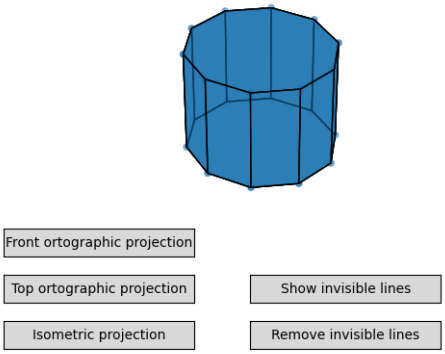
В своей программе я ограничился двумя ортографическими проекциями - передней и верхней, т.к. боковая проекция просто совпадала бы с передней.

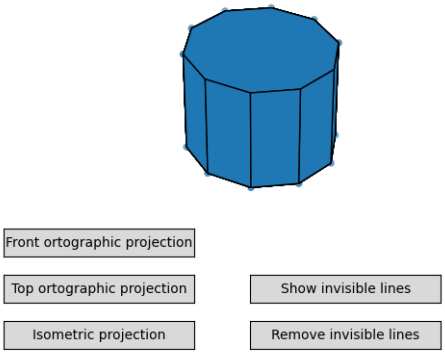
2 Исходный код

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

fig = plt.figure()
fig.subplots_adjust(bottom=0.3)
ax = fig.add_subplot(111, projection='3d')

vertices_num = 10 # amount of vertices in prism
v = [] # vertices of prism
for i in range(vertices_num):
    v.append([np.cos(2 * np.pi * i / vertices_num), np.sin(2 * np.pi * i / vertices_num), 0])
    v.append([np.cos(2 * np.pi * i / vertices_num), np.sin(2 * np.pi * i / vertices_num), 1])

v = np.array(v)
ax.scatter3D(v[:, 0], v[:, 1], v[:, 2]) # adding vertices to plot

sides = [[v[i % (2 * vertices_num)],
         v[(i+1) % (2 * vertices_num)],
         v[(i+3) % (2 * vertices_num)],
         v[(i+2) % (2 * vertices_num)]] for i in range(0, 19, 2)]

sides.append([v[i] for i in range(0, 19, 2)])
sides.append([v[i] for i in range(1, 20, 2)])

ax.add_collection3d(Poly3DCollection(sides, alpha=0.5, edgecolors='black')) # adding sides to plot


def button_callback_remove(event):
    ax.add_collection3d(Poly3DCollection(sides, alpha=1, edgecolors='black'))
    plt.draw()


button_ax_remove = fig.add_axes([0.5, 0.05, 0.31, 0.06])
button_remove = Button(button_ax_remove, "Remove invisible lines")
button_remove.on_clicked(button_callback_remove)


def button_callback_show(event):
    ax.add_collection3d(Poly3DCollection(sides, alpha=0.5, edgecolors='black'))
    plt.draw()


```

```
45 button_ax_show = fig.add_axes([0.5, 0.15, 0.31, 0.06])
46 button_show = Button(button_ax_show, "Show invisible lines")
47 button_show.on_clicked(button_callback_show)
48
49
50 def button_callback_isometric(event):
51     ax.view_init(35, 45)
52     plt.draw()
53
54
55 button_ax_isometric = fig.add_axes([0.1, 0.05, 0.31, 0.06])
56 button_isometric = Button(button_ax_isometric, "Isometric projection")
57 button_isometric.on_clicked(button_callback_isometric)
58
59
60 def button_callback_ortographic_top(event):
61     ax.view_init(90)
62     plt.draw()
63
64
65 button_ax_ortographic_top = fig.add_axes([0.1, 0.15, 0.31, 0.06])
66 button_ortographic_top = Button(button_ax_ortographic_top, "Top ortographic projection
       ")
67 button_ortographic_top.on_clicked(button_callback_ortographic_top)
68
69 def button_callback_ortographic_front(event):
70     ax.view_init(0)
71     plt.draw()
72
73
74 button_ax_ortographic_front = fig.add_axes([0.1, 0.25, 0.31, 0.06])
75 button_ortographic_front = Button(button_ax_ortographic_front, "Front ortographic
       projection")
76 button_ortographic_front.on_clicked(button_callback_ortographic_front)
77
78 ax.grid(None)
79 ax.axis('off')
80 plt.show()
```

3 Результат работы





4 Выводы

Выполнив вторую лабораторную работу по компьютерной графикой, я потренировался в построении трехмерных изображений с помощью библиотеки matplotlib. Также я изучил еще один виджет для интерактивного взаимодействия с пользователем - button.