

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: Ф.М. Шавандрин
Преподаватель: Г.С. Филиппов
Группа: М8О-308Б-19
Дата: 20.12.2021
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 7: 10-гранная прямая правильная призма.

Описание

Для выполнения данного задания я использовал библиотекой matplotlib для Python, которая идеально подходит для работы с трехмерными изображениями. Для построения правильной призмы я задал список вершин (v) и список ребер (edges). Для отрисовки вершин я использовал функцию scatter, а для отрисовки ребер — Poly3DCollection.

Чтобы переключаться между различными проекциями и показывать или скрывать невидимые линии, я воспользовался кнопками (Button) из matplotlib.widgets. Для инициализации каждой кнопки нужно задать координаты её расположения и функцию, которая будет выполняться при нажатии пользователем на кнопку. В моей программе есть 5 кнопок: показать две ортографические проекции с разных сторон, показать изометрическую проекцию, показать и скрыть невидимые линии.

Для отображения и удаления невидимых линий я изменяю параметр alpha, отвечающий за степень прозрачности сторон фигуры.

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

Исходный код

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

figure = plt.figure()
figure.subplots_adjust(bottom=0.3)
figure.patch.set_facecolor('lightblue')
axis_obj = figure.add_subplot(111, projection='3d')

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

v = np.array(v)
# adding vertexes to plot
axis_obj.scatter3D(v[:, 0], v[:, 1], v[:, 2])
```

```

# sides of prism
edges = [[v[i % 20],
          v[(i + 1) % 20],
          v[(i + 3) % 20],
          v[(i + 2) % 20]] for i in range(0, 19, 2)]

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

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

# making buttons
# isometric button

def isometric_button_function(param):
    axis_obj.view_init(35, 45)
    plt.draw()

isometric_axis = figure.add_axes([0.35, 0.23, 0.31, 0.06])
isometric_button = Button(isometric_axis, "Show isometric projection")
isometric_button.on_clicked(isometric_button_function)

# ortographic top button
def ortographic_top_function(param):
    axis_obj.view_init(90)
    plt.draw()

ortographic_top_axis = figure.add_axes([0.15, 0.13, 0.31, 0.06])
ortographic_top_button = Button(ortographic_top_axis, "Show ortographic top")
ortographic_top_button.on_clicked(ortographic_top_function)

# ortographic front button
def button_callback_ortographic_front(event):
    axis_obj.view_init(0)
    plt.draw()

button_ax_ortographic_front = figure.add_axes([0.15, 0.03, 0.31, 0.06])
button_ortographic_front = Button(button_ax_ortographic_front, "Show ortographic
front")
button_ortographic_front.on_clicked(button_callback_ortographic_front)

# show invisible lines button
def show_invisible_lines_function(event):
    axis_obj.add_collection3d(Poly3DCollection(edges, alpha=0.5,
edgecolors='black'))
    plt.draw()

show_lines_axis = figure.add_axes([0.55, 0.13, 0.31, 0.06])
button_show_button = Button(show_lines_axis, "Show invisible lines")
button_show_button.on_clicked(show_invisible_lines_function)

# remove invisible lines button
def remove_lines_function(event):
    axis_obj.add_collection3d(Poly3DCollection(edges, alpha=1,
edgecolors='black'))
    plt.draw()

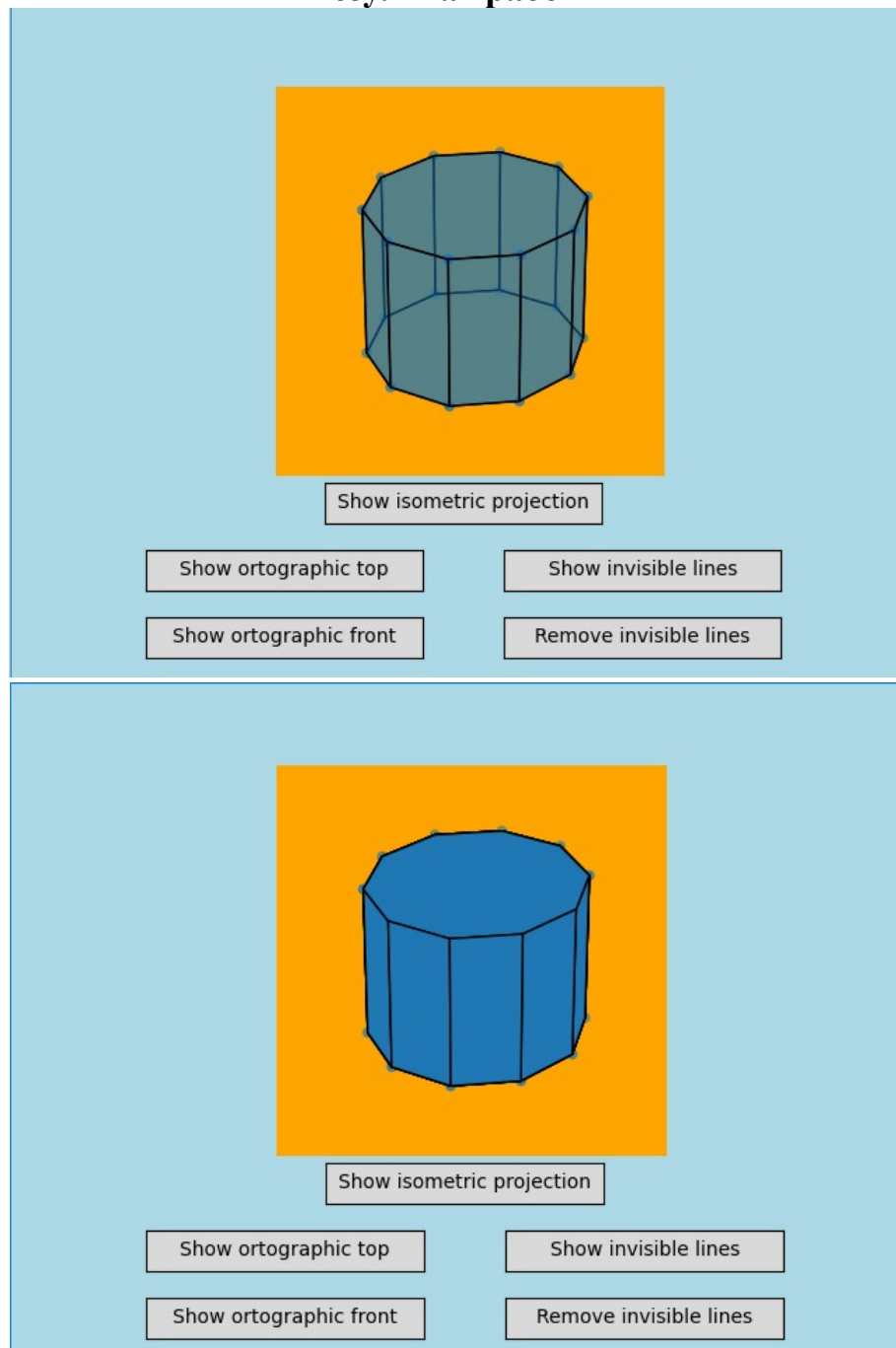
remove_lines_axis = figure.add_axes([0.55, 0.03, 0.31, 0.06])
remove_button = Button(remove_lines_axis, "Remove invisible lines")
remove_button.on_clicked(remove_lines_function)

axis_obj.grid(None)

```

```
axis_obj.axis('off')  
axis_obj.patch.set_facecolor("orange")  
plt.show()
```

Результат работы



Выводы

Выполнив вторую лабораторную работу по курсу «Компьютерная графика», закрепил навыки работы с библиотекой `matplotlib` для Python, изучил различные виды виджетов, которые предоставляет данная библиотека.