



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

**Лабораторна робота №3**  
**З дисципліни «Технології Computer Vision»**  
*ДОСЛІДЖЕННЯ АЛГОРИТМІВ ФОРМУВАННЯ ТА*  
*ОБРОБКИ ВЕКТОРНИХ ЦИФРОВИХ ЗОБРАЖЕНЬ*

Виконала  
студент кафедри  
ІСТ ФІОТ,  
групи ІА-12:  
Яковенко Д. О.

Перевірив:  
пос. Баран Д. Р.

Київ 2024

## I. Мета:

Виявити, дослідити та узагальнити особливості реалізації алгоритмів формування та обробки векторних цифрових зображень на прикладі застосування алгоритмів інтерполяції, апроксимації та згладжування складних 3D растрових об'єктів та застосування технологій видалення невидимих граней та ребер.

## II. Завдання:

*Лабораторія провідної IT-компанії реалізує масштабний проект розробки універсальної платформи з цифрової обробки зображень для задач Computer Vision. Платформа передбачає розташування back-end компоненти на власному хмарному сервері з наданням повноважень користувачам заздалегідь адаптованого front-end функціоналу універсальної платформи. Цим формується унікальна для потреб замовника ERP система з технологіями Computer Vision.*

*Замовниками ресурсів платформи є: державні та комерційні компанії, що розробляють медичне обладнання з діагностування захворювань за візуальною інформацією; автоматизації аграрного бізнесу в аспекті обліку посівних територій за даними з БПЛА; візуального контролю безпекових заходів на об'єктах критичної інфраструктури: аеропорти, торгівельно-розважальні центри, житлові комплекси тощо.*

### Завдання I рівня – максимально 7 балів.

Здійснити виконання завдання лабораторної роботи із застосуванням алгоритму інтерполяції для побудови векторного зображення 2D, 3D графічного об'єкту.

Паралелепіед. Метод інтерполяції: кривими Безьє.

## III. Результати виконання лабораторної роботи.

### 3.1. Синтезована математична модель перетворень графічних об'єктів відповідно до індивідуального завдання.

Згідно з умовами завдання, було створено математичну модель для обробки графічного об'єкту - паралелепіеда. Ця модель втілює операцію створення векторного зображення тривимірного паралелепіеда через аксонометричну проекцію, використовуючи криві Безьє для інтерполяції.

**Криві Безьє** - це параметричні криві, які широко використовуються у комп'ютерній графіці для створення гладких кривих. Вони визначаються набором контрольних точок, які визначають форму кривої. Для створення векторного зображення паралелепіеда використовуються криві Безьє другого порядку (квадратичні криві Безьє), щоб інтерполювати між вершинами паралелепіеда.

Квадратичну криву Без'є задаємо трьома опорними точками:  $P_0$ ,  $P_1$  та  $P_2$ :

$$\mathbf{B}(t) = (1 - t)^2 \mathbf{P}_0 + 2t(1 - t) \mathbf{P}_1 + t^2 \mathbf{P}_2, \quad t \in [0, 1].$$

### 3.2. Блок-схема алгоритму та її опис.

Застосування синтезованих моделей здійснюється у порядку, що відображає суть алгоритму реалізації завдань лабораторної роботи.

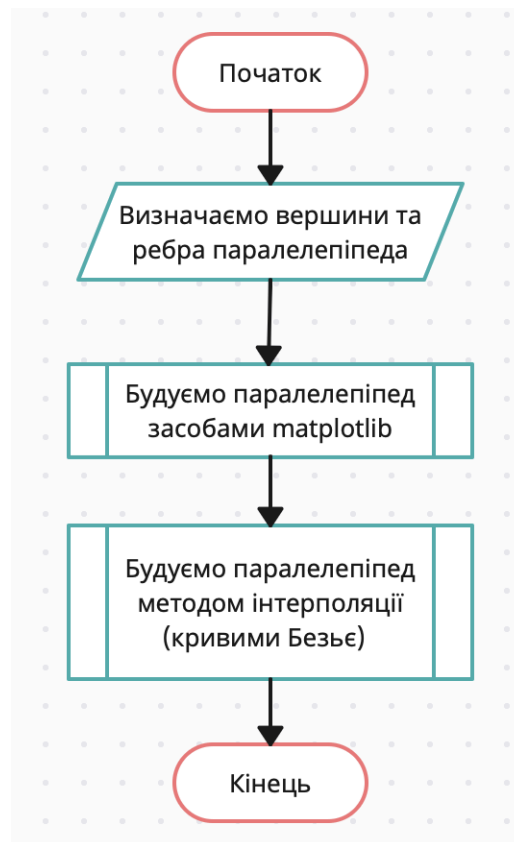


Рис.1. Блок-схема алгоритму програми для завдання третьої складності.

Розглянемо даний алгоритм. Робота алгоритму розпочинається із визначення вершин та ребер паралелепіпеда. Далі будується цей паралелепіпед спочатку вбудованими засобами matplotlib, а потім - методом інтерполяції: кривими Безьє. Побудовані паралелепіпеди мають бути однаковими. На цьому робота алгоритму завершується.

### 3.3. Опис структури проекту програми в середовищі PyCharm.

Для реалізації розробленого алгоритму мовою програмування Python з використанням можливостей інтегрованого середовища PyCharm сформовано проект.

Проект базується на лінійній бізнес-логіці функціонального програмування та має таку структуру.

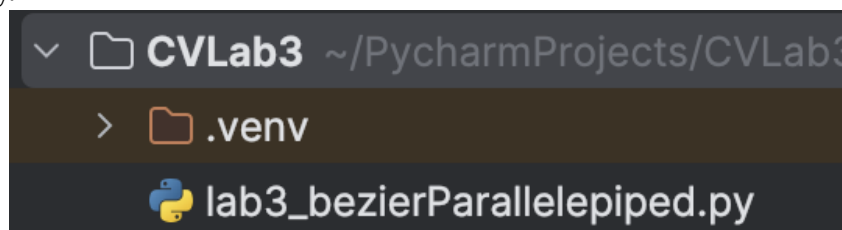


Рис.2. Структура проекту.

CVLab3 – головний каталог проекту

lab3\_bezierParallelepiped.py – файл програмного коду лабораторної роботи;

### 3.4. Результати роботи програми відповідно до завдання.

Результатом роботи програми є програма, в якій при її запуску отримуємо такий результат:

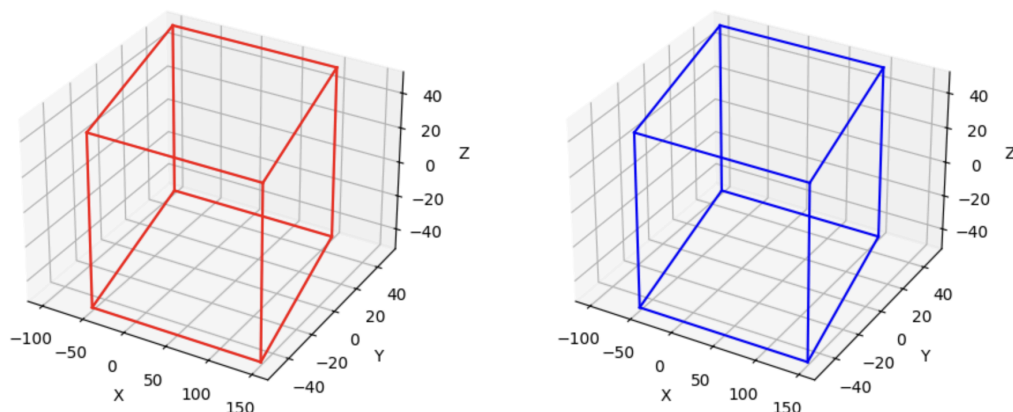


Рис.3. Результат обробки зображення.

Червоний паралелепіпед був побудований засобами matplotlib, тоді як синій – методом інтерполяції (кривими Безьє). Бачимо що паралелепіпеди з вигляду нічим один від одного не відрізняються, в точності відповідають один одному.

Таким чином, представлені результати у повному обсязі відповідають завданню лабораторної роботи.

### 3.5. Програмний код.

Програмний код послідовно реалізує алгоритми для виконання завдання першого рівня складності.

Для спрощення програмного коду і раціоналізації обчислень застосовано функціональні механізми створення підпрограм.

При цьому використано можливості Python бібліотек: numpy; matplotlib.

```
import numpy as np
import matplotlib.pyplot as plt

# Точки паралелепіпеда
points = np.array([
    [-50, -50, -50],
    [150, -50, -50],
    [100, 50, -50],
    [-100, 50, -50],
    [-50, -50, 50],
    [150, -50, 50],
    [100, 50, 50],
    [-100, 50, 50]
])

# Ребра паралелепіпеда
edges = [[0, 1], [1, 2], [2, 3], [3, 0], [4, 5], [5, 6], [6, 7], [7, 4], [0, 4], [1, 5], [2, 6], [3, 7]]

fig = plt.figure()
manager = plt.get_current_fig_manager()
manager.set_window_title('Паралелепіпед')
ax = fig.add_subplot(111, projection='3d')
```

```

# Малювання кожного ребра
for edge in edges: ax.plot3D(*zip(*points[edge]), color='r')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
plt.show()

# Функція обчислення кривої Безьє
def bezier_curve(P, t):
    T = np.tile(t, (3, 1)).T
    return (1 - T) ** 2 * P[0] + 2 * (1 - T) * T * P[1] + T ** 2 * P[2]

fig = plt.figure()
manager = plt.get_current_fig_manager()
manager.set_window_title('Паралелепіпед. Метод інтерполяції: кривими Безьє')
ax = fig.add_subplot(111, projection='3d')

# Побудова ребер
for edge in edges:
    p1 = points[edge[0]]
    p2 = points[edge[1]]
    P = np.array([p1, (p1 + p2) / 2, p2]) # Контрольні точки для кривої

    # Інтерполяція кривими Безьє
    t = np.linspace(0, 1, 100)
    B = bezier_curve(P, t)
    ax.plot(B[:, 0], B[:, 1], B[:, 2], 'b')

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
plt.show()

```

### 3.6. Аналіз результатів відлагодження та верифікації результатів роботи програми.

Результати відлагодження та тестування довели працездатність розробленого коду.

Верифікація функціоналу програмного коду, порівняння отриманих результатів з технічними умовами завдання на лабораторну роботу доводять, що завдання виконано у повному обсязі.

#### IV. Висновки.

У ході виконання лабораторної роботи проведено дослідження алгоритмів побудови векторного зображення графічного об'єкту – паралелепіпду із застосуванням алгоритму інтерполяції, а саме кривими Безьє з використанням можливостей алгоритмічної мови високого рівня Python. В результаті виконання я здійснила побудову такого об'єкту двома методами та вивела результати у вікні програми.

Виконала: студент Яковенко Д.О.