**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №3**

**з навчальної дисципліни «Технології Computer Vision»**

**Тема:**

**ДОСЛІДЖЕННЯ АЛГОРИТМІВ ФОРМУВАННЯ ТА ОБРОБКИ ВЕКТОРНИХ**

**ЦИФРОВИХ ЗОБРАЖЕНЬ**

**Виконав:**

Студент 3 курсу кафедри ІПІ ФІОТ,

Навчальної групи ІП-11

Лошак В.І.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

**Київ 2024**

**І. Мета:**

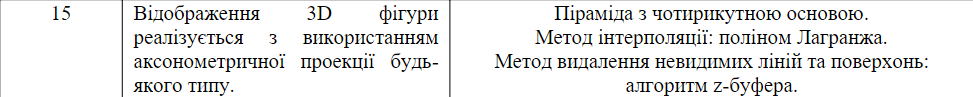
Виявити дослідити та узагальнити особливості реалізації алгоритмів формування та

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.

**ІІ. Завдання:**

Здійснити синтез математичних моделей та розробити програмний скрипт, що забезпечує *реалізацію векторних алгоритмів над 2D, 3D графічними примітивами та з цифровими зображеннями*.

Технічні умови реалізації завдання:



**Завдання ІІІ рівня – максимально 9 балів.**

Здійснити виконання завдання ІІ рівня складності:

Здійснити виконання завдання лабораторної роботи із застосуванням алгоритму інтерполяції для побудови векторного зображення 2D, 3D графічного об’єкту та алгоритму видалення невидимих ліній та поверхонь.

Реалізувати розробку програмного скрипта No2, що реалізує виділення контору обраного об’єкту на цифровому растровому зображенні. За необхідності передбачити корекцію кольору цифрового растрового зображення для покращення якості виділення контору обраного об’єкту.

Цифрове зображення обрати самостійно.

**ІІІ. Результати виконання лабораторної роботи.**

**Завдання І.**

1. ***Рендеринг в векторній формі***

Для початку налаштуємо процес рендерингу піраміди використовуючи бібліотеку graphics.py. Для перенесення координат чотирикутної піраміди на 2d поверхню використаємо аксонометричну проекцію. З кодом що реалізує цей процес можна ознайомитись в файлі з кодом програми, тут наведу лише загальний алгоритм проекції.

A screenshot of a computer program

Description automatically generated

Рис.1 — Функції загального призначення що повертають матриці для обертання 3D фігури

*Код що виконує початковий рендеринг піраміди що задана в векторній формі:*  
pyramid\_vertices = np.array([

    [0, 0, 0, 1],  # Base vertices

    [100, 0, 0, 1],

    [100, 100, 0, 1],

    [0, 100, 0, 1],

    [50, 50, 100, 1]  # Top vertex

])

win = GraphWin("Pyramid Axonometric Projection", 400, 400)

win.setBackground('white')

rotated\_vertices = pyramid\_vertices @ rotation\_matrix\_x(-80)

rotated\_vertices = rotated\_vertices @ rotation\_matrix\_z(20)

projection\_matrix = get\_projection\_matrix()

vertices\_2d = project\_vertices(rotated\_vertices, projection\_matrix)

vertices\_2d += np.array([200, 200])  # Center the pyramid in the window

draw\_pyramid(win, vertices\_2d)

win.getMouse()

win.close()

*Результат:*

A screenshot of a computer

Description automatically generated

1. ***Растеризація зображення***

Для растеризації використаємо алгоритм Брезенхема. В результаті виконання було отримано набір ліній, що відповідають ребрам піраміди. Ці лінії складаються з набору точок що містять координати х та у відповідно.

*Блоксхема алгоритму:*

A diagram of a flowchart

Description automatically generated

*Результат:*

A screenshot of a computer

Description automatically generated

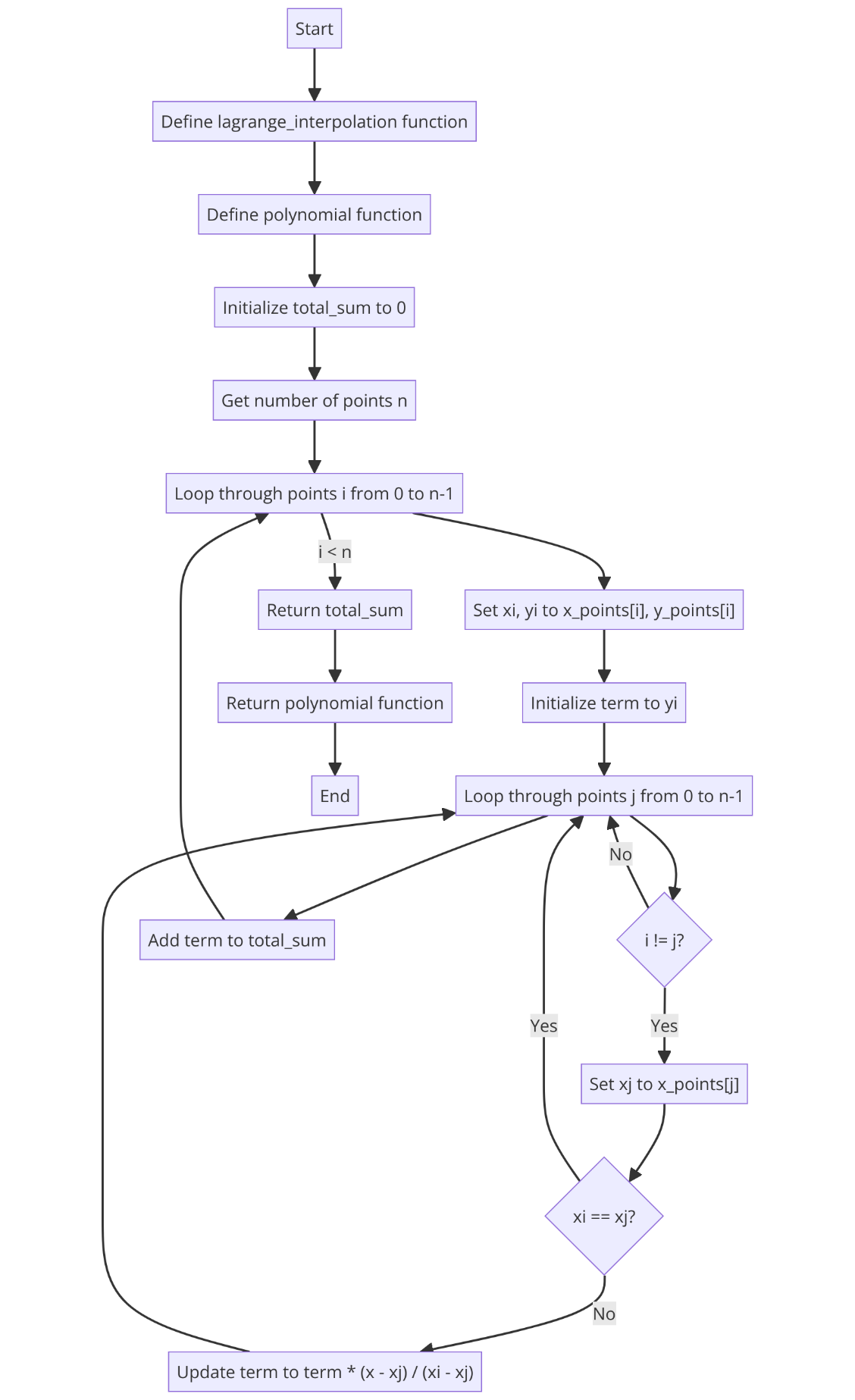
1. ***Векторизація зображення з використанням інтерполяції Лагранжа.***

Для векторизації використаємо інтерполяцію Лагранжа. Інтерполяція Лагранжа — це числовий метод, що використовується для знаходження полінома, який точно проходить через заданий набір даних точок. Основна ідея полягає у побудові інтерполяційного полінома як лінійної комбінації базисних поліномів, кожен з яких дорівнює нулю в усіх заданих точках даних, крім однієї. Поліном конструюється за формулою

де Li(x)— це базисні поліноми Лагранжа, визначені як

забезпечуючи інтерполяцію полінома P(x)P(x)P(x) через точки даних (xi,yi), i=0,1,…,n

*Блоксхема алгоритму:*



*Результат:*

A screenshot of a computer

Description automatically generated

**Завдання ІІ.**

1. ***Реалізувати розробку програмного скрипта No2, що реалізує виділення контору обраного об’єкту на цифровому растровому зображенні.***

Обране для обробки зображення:



Для виділення контурів ми використовуємо метод contour з бібліотеки matplotlib. Всередині цього методу використовується алгоритм Marching Squares для визначення послідовності пікселів з однаковою інтенсивністю

*Код:*

im = array(Image.open(r'book.jpg').convert('L'))

image = Image.open(r"book.jpg")

figure()

contour(im, origin='image')

axis('equal')

show()

contour(im, levels=[170], colors='black', origin='image')

axis('equal')

show()

*Результат:*

A green and blue background with a rectangular object

Description automatically generated

Після виділення лише контурів що переходять певний threshold ми отримуємо контур шуканого об’єкту.

A black and white drawing of a rectangle

Description automatically generated

**IV. Висновки.**

В ході цієї лабораторної роботи я дослідив та узагальнив особливості реалізації алгоритмів формування та обробки векторних цифрових зображень. Я зосередив увагу на застосуванні алгоритмів інтерполяції, апроксимації та згладжування для складних 3D растрових об'єктів, а також на методах видалення невидимих граней та ребер. Ця робота дала змогу глибше зрозуміти, як можна ефективно перетворювати растрові зображення в векторні формати, покращуючи їхню якість за допомогою інтерполяції та згладжування, а також ефективно використовувати техніки відображення 3D-моделей з приховуванням невидимих елементів.

Виконав: студент ФІОТ Лошак В.І. ІП-11