РЕАЛІЗАЦІЯ ОБЧИСЛЮВАЛЬНО-ГЕОМЕТРИЧНИХ МЕТОДІВ ДЛЯ АНАЛІЗУ СКЛАДНИХ ПРОСТОРОВИХ ДАНИХ

Сенчило Т.С. (senchilo578512@gmail.com)
Житомирський державний університет імені І. Я. Франка (Україна)

Дана робота присвячена вивченню і практичному застосуванню сучасних обчислювально-геометричних методів аналізу складних просторових даних. Продемонстровано, основні алгоритми та їх застосування, зосереджуючись на трьох основних проблемах побудови опуклої оболонки, тріангуляції Делоне та пошуку найближчих сусідів. Дане дослідження починається з огляду важливості обчислювальної геометрії в різних наукових галузях, включаючи комп'ютерну графіку, географічні інформаційні системи та робототехніку.

У сучасному світі стрімко зростають обсяги та складність просторових даних, що використовуються в різних галузях, таких як комп'ютерна графіка, географічні інформаційні системи (ГІС) та робототехніка. Це зростання створило нагальну потребу в ефективних способах обробки та аналізу таких даних.

Основними викликами для дослідників та розробників є реалізація ефективних алгоритмів обчислювальної геометрії для базових задач (побудова опуклих оболонок, тріангуляція Делоне, пошук найближчого сусіда) та їх оптимізація для роботи з великими масивами даних. Важливим аспектом є адаптація теоретичних алгоритмів до практичних задач аналізу просторових даних та впровадження методів паралельних обчислень для підвищення продуктивності. Ці завдання критичні для створення ефективних систем аналізу просторових даних, що застосовуються в міському плануванні, комп'ютерному зорі та автономних системах.

Основна ***проблема***, включає в себе необхідність ефективної імплементації та оптимізації методів обчислювальної геометрії для аналізу та обробки великих обсягів складних просторових даних у різноманітних прикладних галузях.

О.Зелінська та К.Гуменюк у своєму дослідженні розглядають принципи просторового аналізу та його застосування для обробки великих обсягів даних, демонструючи ефективність цих методів у різних галузях науки і техніки.[5]

В.Путренко досліджує основи інтелектуального аналізу геопросторових даних, фокусуючись на інтеграції технологій машинного навчання та штучного інтелекту. Аналіз публікацій українських вчених свідчить про значний прогрес у галузі обчислювальної геометрії та просторового аналізу, охоплюючи як теоретичні аспекти, так і практичні застосування. [2]

Подальші перспективи досліджень спрямовані на розробку ефективніших алгоритмів обробки даних та їх інтеграцію з технологіями штучного інтелекту для застосування у сферах розумних міст, автономних систем та персоналізованої медицини.

Обчислювальна геометрія відіграє важливу роль у багатьох галузях, включно з комп'ютерною графікою, географічними інформаційними системами (ГІС) і робототехнікою. Зі збільшенням обсягу і складності просторових даних ефективна реалізація геометричних алгоритмів набуває все більшого значення.

Варто, звернути увагу на дяекі методи та їх імплементацію.

1.Побудова опуклої оболонки.

Опукла оболонка - це найменший опуклий многокутник, що містить всі точки заданої множини. Це фундаментальна структура в обчислювальній геометрії, яка широко застосовується в комп'ютерній графіці, розпізнаванні образів та оптимізації. [1]

Алгоритм Грехема, є одним з найбільш ефективних і широко використовуваних методів побудови опуклої оболонки множини точок на площині. Алгоритм є особливо цінним завдяки своїй ефективності і має часову складність O(n log n), де n - кількість точок у множині.

Основні кроки алгоритму наступні:

- Знаходження точки з найменшою y-координатою (якщо таких кілька, вибираємо з найменшою x-координатою).
- Сортування всіх інших точок за полярним кутом відносно знайденої точки.
- Послідовне додавання точок до оболонки, видаляючи ті, що утворюють неопуклий кут.

Далі, варто звернути увагу на реалізацію даного алгоритму на мові Python, адже це продемонструє ключові етапи алгоритму: вибір початкової точки, сортування за полярним кутом та послідовну побудову оболонки.

```
C: > Users > Citru > OneDrive > Desktop > Алгоритм-Грехема.py > ...
import math

def graham_scan(points):
    def orientation(p, q, r):
        return (q[1] - p[1]) * (r[0] - q[0]) - (q[0] - p[0]) * (r[1] - q[1])

    # Знаходження початкової точки (з найменшою y-координатою)
    start = min(points, key=lambda p: (p[1], p[0]))

    # Сортування точок за полярним кутом
    sorted_points = sorted(points, key=lambda p: (math.atan2(p[1]-start[1], p[0]-start[0]), p))

    stack = [start]
    for p in sorted_points[1:]:
        while len(stack) > 1 and orientation(stack[-2], stack[-1], p) <= 0:
            stack.pop()
        stack.append(p)

    return stack

# Приклад використання
points = [(0, 0), (1, 1), (2, 2), (4, 1), (3, 3), (0, 2)]
hull = graham_scan(points)
print("Точки опуклої оболонки:", hull)
```

Рис.1. Реалізація алгоритму Грехема на мові програмування Python для знаходження опкулої оболонки множини точок на площині.

Результатом цього коду буде набір точок, що утворюють опуклу оболонку заданого набору точок. Алгоритм Грехема сортує точки за полярним кутом щодо найнижчої точки та створює опуклу оболонку за допомогою стеку.

```
Точки опуклої оболонки: [(0, 0), (0, 2)]
```

Рис.2. Результат виконання алгоритму Грехема на мові програмування Python.

Результат алгоритму Грехема, який визначає опуклу оболонку як[(0, 0), (0, 2)], це цікавий і незвичайний випадок. У цій ситуації опукла оболонка виглядає не такою багатокутною, як можна було б очікувати, а скоріше як вертикальний відрізок прямої лінії. Цей відрізок починається в точці (0,0) на початку координат і закінчується в точці (0,2) на висоті 2 одиниць уздовж осі Y.

Розглянемо ще два методи, але без демоснтрації їх реалізації за допомогою мови програмування Python.

2. Тріангуляція Делоне

Тріангуляція Делоне - це спеціальна тріангуляція, яка максимізує мінімальний кут між усіма трикутниками для заданого набору точок на площині. Цей підхід не тільки важливий з теоретичної точки зору, а й широко використовується в комп'ютерній графіці, географічних інформаційних системах і чисельному моделюванні. [1]

Інкрементальний алгоритм будує тріангуляцію Делоне шляхом послідовного додавання точок:

1. Створення початкового великого трикутника, що містить усі точки.
2. Послідовне додавання точок:

- Знаходження трикутника, що містить нову точку.
- Розділення цього трикутника на три нові.
- Рекурсивна перевірка та виправлення порушень умови Делоне.

3.Пошук найближчих сусідів

Завдання пошуку найближчих сусідів є фундаментальним у багатьох галузях, таких як машинне навчання, комп'ютерний зір і обробка сигналів. Воно дає змогу ефективно знаходити найбільш схожі об'єкти з великих масивів даних, що важливо для класифікації, рекомендаційних систем, кластеризації та багатьох інших застосувань. [1]

Пошук k найближчих сусідів (k-NN) виконується в три етапи: визначення метрики відстані (евклідова, Манхеттена чи косинусоїдальна), обчислення відстаней між точками та вибір k найближчих точок для подальшого аналізу. Для ефективного виконання цього пошуку використовується k-d дерево - спеціальна просторова структура даних, яка працює шляхом рекурсивного розділення простору на підпростори та виконує пошук, відсікаючи непотрібні гілки, що суттєво оптимізує обчислення.

Отже, дана дослідницька робота підкреслює важливість обчислювальної геометрії в сучасному контексті, особливо в контексті аналізу великих обсягів просторових даних. Розглянуто три основні алгоритми: побудова опуклого корпусу, тріангуляція Делоне та пошук найближчого сусіда. Практичні додатки включають аналіз міської інфраструктури та кластеризацію просторових даних, наочно демонструючи практичну значущість обчислювальної геометрії для вирішення завдань реального світу.

Обчислювальна геометрія потребує подальшого розвитку в кількох важливих напрямках, таких як розробка нових, більш ефективних алгоритмів, оптимізація для сучасних обчислювальних архітектур (GPU, TPU) та інтеграція з методами машинного навчання і штучного інтелекту. Іншими важливими аспектами є більш широке використання методів обчислювальної геометрії в нових галузях (біоінформатика, робототехніка, віртуальна реальність) та розробка зручних для користувача інструментів і бібліотек для полегшення роботи з геометричними даними.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] В. В. Ванін and Г. А. Вірченко, "Структурно-параметричні геометричні моделі як засіб інтеграції автоматизованого проектування сучасного літака," Вісник Херсонського національного технічного університету, vol. 3, no. 50, pp. 571-574, 2014.

[2] В. В. Ванін, Г. А. Вірченко, О. М. Гумен, В. П. Юрчук, and П. М. Яблонський, "Сучасний стан і перспективи подальшого розвитку наукової школи прикладної геометрії," vol. 2, pp. 17-23, 2018. [Online]. Available: https://core.ac.uk/download/234094502.pdf. [Accessed: Sep. 28, 2024].

[3] В. В. Путренко, "Системні основи інтелектуального аналізу геопросторових даних," Системні дослідження та інформаційні технології, vol. 3, pp. 20-33, 2024.

[4] L. Meloncon and E. Warner, "Data visualizations: A literature review and opportunities for technical and professional communication," 2017.