

Отчет по полиномиальной аппроксимации

Постановка задачи:

Проект направлен на разработку эффективных алгоритмов генерации полилиний, сжатия и измерения ошибок.

Метод решения:

1. Генерация полилиний

Для генерации полилиний были разработаны два алгоритма:

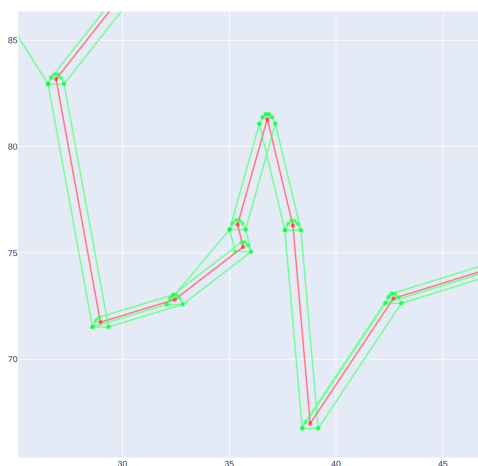
Алгоритм 1:

- Генерировать случайные точки на плоскости.
- Интерпретировать эти точки как граф.
- Применить задачу о путешествующем продавце (TSP) для поиска непересекающегося пути.

Алгоритм 2:

- Создайте непересекающиеся окружности в пространстве.
- Получите две хорды в каждой окружности и создайте извилистые пути между ними (кластеры).
- Соединить кластеры с помощью эвристического подхода.

2. Сжатие полилиний



Алгоритм сжатия является частичной реализацией алгоритма, представленного в *“Searching for a Compressed Polyline with a Minimum Number of Vertices”*.

Шаги следующие:

- Создать треугольную сетку вокруг каждой точки полилинии.
- Постройте выпуклую оболочку.
- Позволить каждой вершине переместиться в другую вершину в сетке или исчезнуть, при условии, что новая полилиния без этой вершины остается на выпуклом оболочке исходной полилинии.

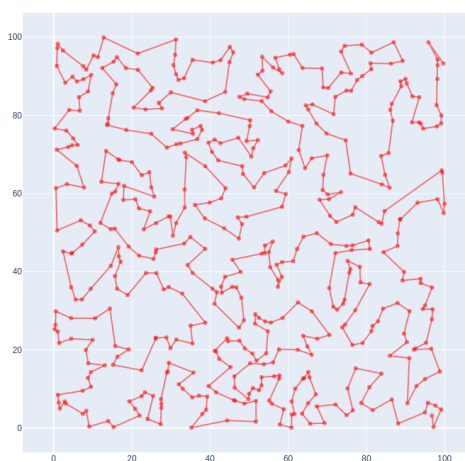
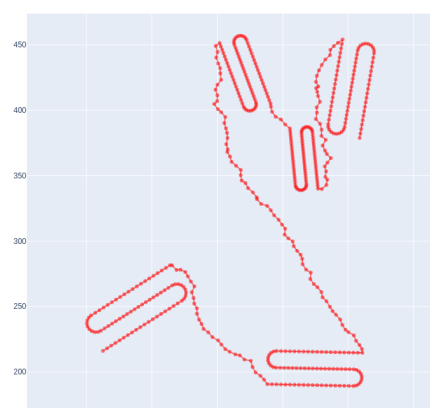
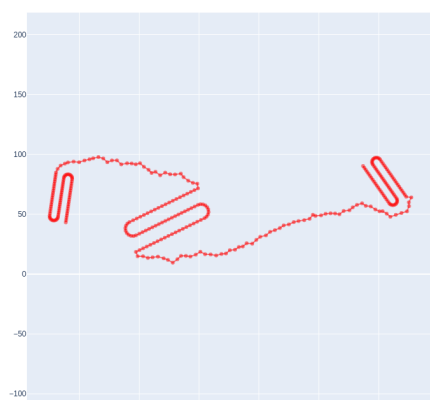
3. Измерение ошибок

Используются четыре типа измерения ошибок:

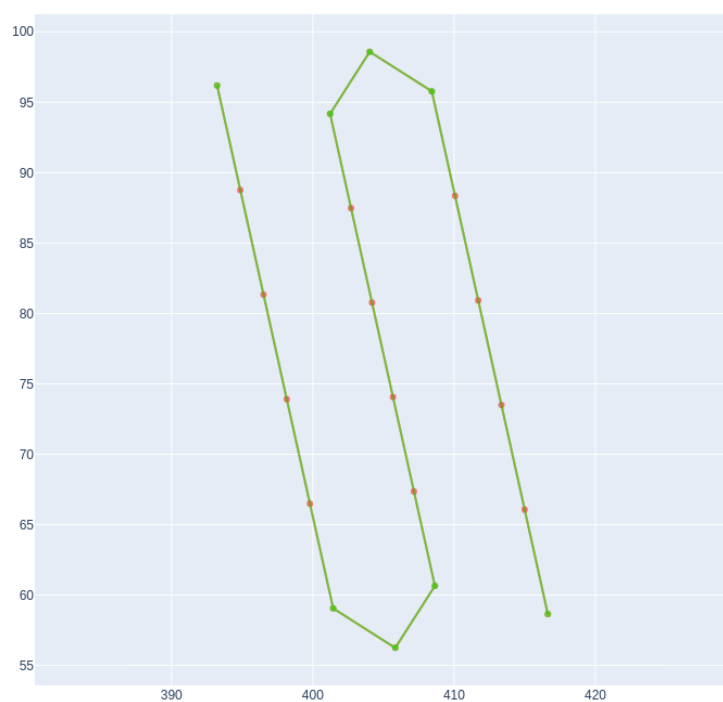
1. Разница площадей между исходной и упрощенной полилинией.
2. Максимальное расстояние между любой точкой исходной полилинии и ближайшей к ней точкой упрощенной полилинии.
3. Средняя разность углов между последовательными сегментами исходной полилинии и соответствующими сегментами упрощенной полилинии.
4. Общее расстояние между любой точкой исходной полилинии и ближайшей к ней точкой упрощенной полилинии.

Результаты тестирования:

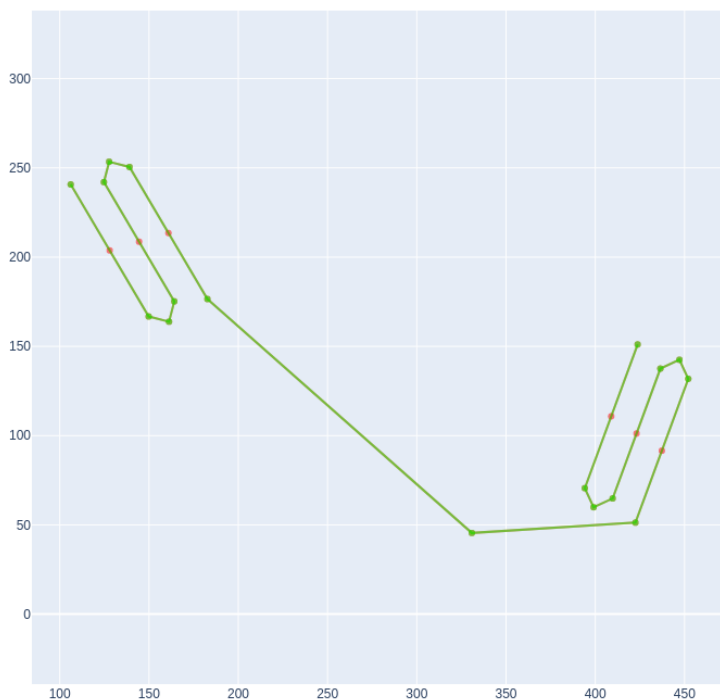
1. Генерация полилиний



2. Сжатие полилиний и Измерение ошибок



Area difference: -0.0005398
Angle difference: 0.916241
Hausdorff difference: 15.2105
Full distance: 129.411



Area difference: -0.0295514
Angle difference: 0.688272
Hausdorff difference: 38.7847
Full distance: 231.117

Как запустить тесты локально:

Запустите нужный тест в `prj.test/polyline_test.cpp`, результат будет автоматически открываться в браузере.

Параметры алгоритма:

Список параметров можно найти в `prj.lab/polyline/Utils/parms`

Заключение:

В рамках проекта были успешно разработаны алгоритмы генерации полилиний, сжатия и измерения ошибок. Алгоритмы генерации полилиний создавали непересекающиеся траектории, а алгоритм сжатия минимизировал интегральное квадратичное отклонение и количество вершин. При измерении ошибок учитывались различные факторы, такие как разница площадей, максимальное расстояние, средняя разница углов и общее расстояние. Однако из-за сложности алгоритма, представленного в статье, алгоритм сжатия может работать только с небольшими полилиниями. Проблема быстро становится экспоненциальной, поскольку количество возможных путей, которые необходимо рассмотреть, равно $(C + 1)^n$ где C - количество точек в каждой сетке, а n - количество вершин в полилинии.