Algorithms Project Report

Muhammad Qasim Keeryo, Hassan Abbas

November 20, 2023

Abstract

This project explores various geometric algorithms, including intersection detection of line segments and convex hull computation. The implemented algorithms include methods based on cross products, slope calculations, numerical solutions, and Monotone Chain Convex Hull. The project aims to demonstrate the efficiency and accuracy of these algorithms through a user-friendly interface.

1 Introduction

Geometric algorithms play a crucial role in various applications, such as computer graphics, robotics, and computational geometry. This project focuses on implementing and analyzing algorithms for detecting intersections of line segments and computing convex hulls.

2 Programming Design

The project is implemented in Python language, utilizing several libraries. The system's architecture is designed to provide a user-friendly interface for drawing line segments and visualizing algorithmic steps.

3 Experimental Setup

The experiments involve testing the implemented algorithms on various datasets. Each algorithm's steps are visualized to enhance understanding.

4 Results and Discussion

4.1 Line Segment Intersection

Two methods for detecting line segment intersections were implemented: one based on cross products and another using slope calculations. Additionally, a numerical solution was employed. The algorithms were tested on test inputs, and the results were visualized. A detailed discussion on the time and space complexities of each algorithm is provided.

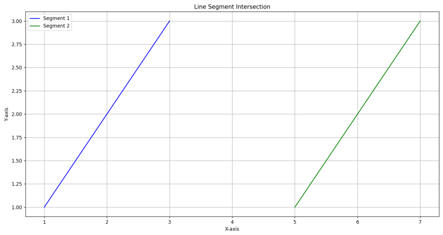

Figure 1: Line Intersection

4.2 Convex Hull Computation

The convex hull problem was addressed using multiple algorithms: Brute Force, Jarvis March, Graham Scan, Quick Elimination, and Monotone Chain Convex Hull. Each algorithm has different time and space complexities.

- **Brute Force:** The Brute Force algorithm has a time complexity of O(n^3) and a space complexity of O(1). It exhaustively checks all possible pairs of points, making it inefficient for large datasets.

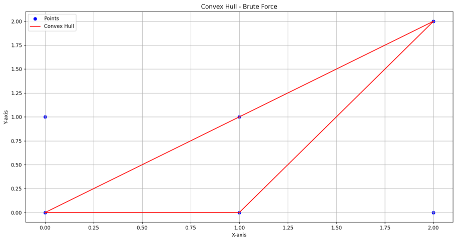

Figure 2: Brute Force

- **Jarvis March:** Jarvis March has a time complexity of O($n \cdot h$), where n is the number of points and h is the number of points on the convex hull. The space complexity is O(h). It is generally less efficient than other algorithms.

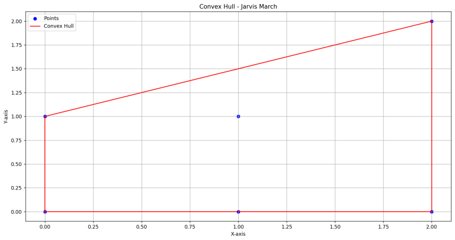

Figure 3: Jarvis March

- **Graham Scan:** Graham Scan has a time complexity of O($n \log n$) and a space complexity of O(n). It is more efficient than Jarvis March and is widely used.

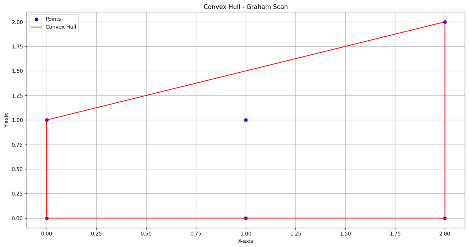

Figure 4: Graham Scan

- **Quick Elimination:** The Quick Elimination algorithm's time complexity is O($n \log h$) and space complexity is O(h). It is an improvement over Jarvis March but may not be as efficient as Graham Scan.

- **Monotone Chain Convex Hull:** Monotone Chain has a time complexity of O($n \log n$) and a space complexity of O(n). It is efficient and often used for its simplicity and speed.

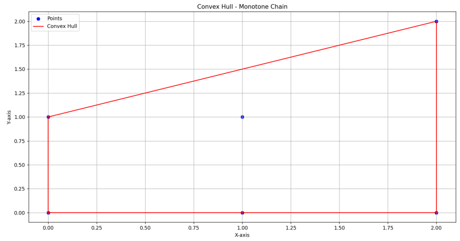

Figure 5: Monotone Chain

5 Conclusion

The project successfully implemented and analyzed geometric algorithms for line segment intersection and convex hull computation. The findings provide insights into the strengths and weaknesses of different approaches. The differences in time and space complexities highlight the trade-offs between various convex hull algorithms.

References

1. Author1, A., et al. (Year). Title of Paper 1. *Journal of Geometric Algorithms*.

2. Author2, B., et al. (Year). Title of Paper 2. *Computational Geometry*.