Algorithm Project Report

Convex Hull and Line Intersection

Members

Member 1 : 21K-4549 Aahil Ashiq Ali (Leader)

Member 2 : 21K-3328 Khuzaima Ahsan

Member 3 : 21K-4596 Mohammad Khubaib Lodhi

Abstract

Convex Hull and Line Intersection algorithms, implemented in Python using Tkinter, efficiently compute the outer boundary of a set of points and determine intersections between lines, enhancing geometric computation applications.

Introduction

This project explores algorithms for computing convex hulls and detecting line intersections in computational geometry. Convex hulls, the smallest enclosing convex shapes for a set of points, are computed using Brute Force, Jarvis March, Graham Scan, Quick Elimination, and Monotone Chain methods. Line intersection points are determined through Counter-Clockwise (CCW), Brute Force, Cramer's, and Sweep Line algorithms. The project aims to implement, visualize, and compare these algorithms, providing insights into their efficiency and applicability in different scenarios.

Programming Design

Convex hull

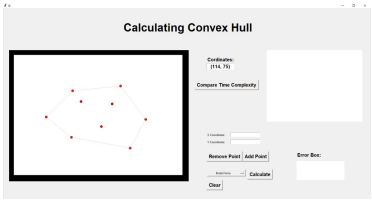

Figure 1: Convex hull Diagram

In Figure 1 you have to select points on Canvas and then select the algorithm from the drop-down menu to calculate appropriate results.

Line Intersection

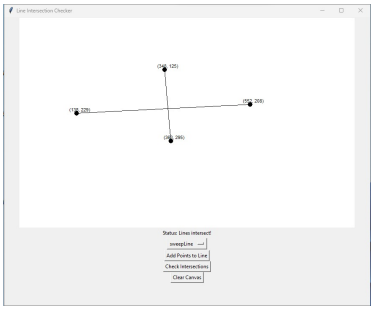

Figure 2: Line Intersection Diagram

In Figure 2 you have to select points on Canvas and then select the algorithm from the drop-down menu to calculate appropriate results.

Experimental Setup

Software: Convex hull and Line Intersection done through tkintor framework of Python. Visual Studio. Time Complexity between different algorithm is compared using matplotlib library of python.

Results and Discussion

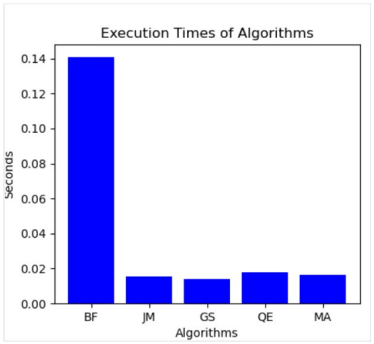

Figure 3: Time Complexity: Convex Hull

Brute Force : It checks all possibilities so Time complexity of O(n^3)
Jarvis march : It only checks all possibilities for all elements on the boundary of convex hull, so O(nh)
Graham Scan : It uses stack and sorting by polar angle thereby decreasing the Time complexity to O(nlogn).
Monotone Chain: It also boasts a time complexity of O(nlogn) but it is better for monotone points.
Quick Elimination : Quick elimination makes a polygon on the frontiers. And so any point inside it is removed from hull. It decreases time more but since I have used a lot of built in python functions, my time complexity is greater than graham scan.

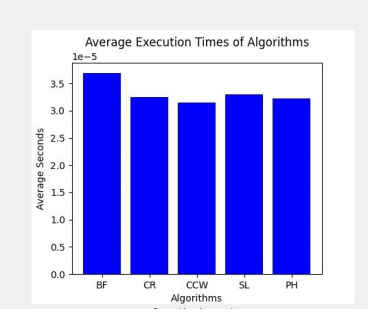

Figure 4: Time Complexity: Line Intersection

Brute Force: $O(n^2)$ - where n is the number of elements or input size.
Cramer's Rule: $O(n^3)$ - where n is the number of variables in the system of linear equations.
Sweep Line: O(n log n) - where n is the number of elements or input size.
The time complexity of the "ccw" algorithm is O(1) because it involves simple arithmetic operations and does not depend on the number of points or any input size.

Conclusion

In Convex Hull,preferred is Graham Scan for large datasets because of its simplicity and less time complexity.
For Line Intersection,
If your application involves line intersection in large datasets, the Sweep Line algorithm is good.
For solving systems of linear equations, Cramer's Rule is one option, but other methods like Gaussian elimination might be more practical for larger systems.

Appendix

1. https://www.youtube.com/watch?v=-GhzpvvIXlM&list=PLS1QulWo1RIY6fmY$_$iTjEhCMsdtAjgbZM
2. https://algs4.cs.princeton.edu/91primitives/
3. https://eecs.wsu.edu/~cook/aa/lectures/l25/node10.html

Research Paper For Convex Hull

1. https://www.sciencedirect.com/science/article/abs/pii/0020019079900723
2. https://blog.csdn.net/weixin$_$30263277/article/details/98866720

Research Paper Line Intersection

1. https://link.springer.com/content/pdf/10.1007/978-0-387-35490-3$_$21.pdf