



National University
of computer and emerging sciences

GEOMETRIC ALGORITHMS (LINE INTERSECTION & CONVEX HULL)

PROJECT REPORT

PROFESSOR DR. FARRUKH SALIM (BCS-5E)

DESIGN & ANALYSIS OF ALGORITHMS (CS-2009)

MUHAMMAD TAHIR (K21-4503)

INSHA JAVES (K21-3279)

SABIKA SHAMEEL (K21-4606)

Abstract

This project delves into the realm of geometric algorithms, specifically focusing on the design and implementation of algorithms for Convex Hull construction and Line Intersection detection. The Convex Hull algorithms explored include Brute Force, Jarvis March, Graham Scan, Quick Elimination, and A New Approach Algorithm, each meticulously examined for their distinct steps and interactive features within a user interface. Additionally, Line Intersection methods such as the Cross Product Method, Slope Intersection Method, and CCW Method are implemented with equal scrutiny.

Contents

0.1	Introduction	2
0.2	Programming Design	2
0.3	Experimental Setup	2
0.4	Results and Discussion	2
	0.4.1 Line Intersection	2
	0.4.2 Convex Hull	2
0.5	Insights	2
0.6	Conclusion	2
0.7	References	3

0.1 Introduction

This project explores geometric algorithms, focusing on Convex Hull construction and Line Intersection detection. It includes five Convex Hull algorithms and three Line Intersection methods within an interactive user interface for hands-on learning. Emphasis is placed on analyzing time and space complexities and enhancing accessibility with a user-friendly interface. The goal is to contribute to geometric algorithm knowledge while providing an engaging learning experience.

0.2 Programming Design

Implemented in C#, our programming design prioritizes clarity and readability, taking inspiration from a c# implementation renowned for handling geometric computations. The project incorporates a suite of geometric algorithms, each meticulously chosen for its specific strengths and applications.

In tackling the line segment intersection problem, the implementation explores diverse techniques, including counter-clockwise operations, vector cross products, and the application of parametric equations. Counter-clockwise operations play a crucial role in determining the relative orientation of points, aiding in identifying intersecting line segments. Vector cross-products offer a robust method for evaluating the convexity of polygons formed by these line segments. The utilization of parametric equations enhances efficiency by providing an alternative approach to assessing the intersection points.

Addressing the Convex Hull problem involves a selection of algorithms, each offering unique advantages. The Brute Force algorithm, while computationally intensive, serves as a baseline for comparison, ensuring the correctness of subsequent, more optimized algorithms. Jarvis-March, known for its simplicity, constructs a convex hull based on selecting the point with the smallest polar angle. Graham Scan leverages angular sorting for efficient convex hull construction. Quick Hull, A New Way algorithm, demonstrates efficiency by recursively dividing the point set and constructing convex hulls for each subset. The New Way algorithm, starts firstly with Quick hull and in its second step Graham scan is applied to the remaining points.

In essence, the C# implementation reflects the thoughtful design and algorithmic exploration observed in the c# reference, maintaining a focus on clarity, efficiency, and a comprehensive approach to solving geometric problems, now incorporating the use of parametric equations for line segment intersection analysis.

0.3 Experimental Setup

The experimental setup assesses the performance of implemented geometric algorithms by generating random points to draw diverse objects on the screen. The report outlines each algorithm's steps using these points, providing visual representations and user interaction for a comprehensive understanding.

A key focus is the calculation of time and space complexities for each algorithm. The report details the methodology used, offering insights into computational efficiency. This analysis helps gauge algorithmic performance, revealing strengths and limitations across different scenarios.

0.4 Results and Discussion

The below table shows the results alongside their time complexity of three of the Line Intersection Algorithms.

0.4.1 Line Intersection

Algorithm	Time Taken(s)
Counter-Clockwise	0.00320005
Vector Cross Product	0.000365
Parametric Equation	0.002537958

Attached below are the results alongside their time complexity of five of the Convex Hull Algorithms.

0.4.2 Convex Hull

Algorithm	Time Complexity	Time Taken(s) on N(100)
Graham Scan	$O(n \log(n))$	0.00432353
Jarvis-March	$O(nh)$	0.31947946
Quick Elimination	$O(n \log(n))$	0.735836029
Brute Force	$O(n^3)$	0.0458802776
A New Way	$O(n \log(n))$	0.0015172615

0.5 Insights

Brute Force: Despite its theoretical inefficiency $O(n^3)$, the execution time of 29 seconds underscores its impracticality for large data sets.

Jarvis March: With a relatively faster execution time of 0.31947946 seconds, Jarvis March demonstrates improved efficiency compared to Brute Force but is still outperformed by more sophisticated algorithms.

Graham Scan: Boasting a time complexity of $O(n \log n)$, Graham Scan executes swiftly in 0.00432353 seconds, showcasing its efficiency for large data sets.

Quick Elimination: Exhibiting a time complexity of $O(n \log n)$ on average, Quick Elimination achieves an execution time of 0.735836029 seconds, demonstrating competitive performance but with a potential worst-case scenario of $O(n^2)$.

A New Way Algorithm: Leading the efficiency frontier with time complexity of $O(n \log n)$, Chan's Algorithm impressively executes in a mere 0.0015172615 seconds, establishing its superiority, especially for large and diverse data sets.

0.6 Conclusion

The interplay between time complexity and real-world execution time underscores the importance of algorithm selection based on the nature and size of the input data. While

Brute Force may be infeasible for large data sets, more sophisticated algorithms like Graham Scan and A New Way's Algorithm showcase remarkable efficiency, laying the foundation for informed algorithmic choices in computational geometry.

0.7 References

- Researchgate (A New Way Algorithm)
- Semisignal (Brute Force Algorithm)
- CISE (Parametric Equation Algorithm)
- Wikipedia (QuickHull Algorithm)
- GeekForgeeks (QuickHull Algorithm)
- Introduction to Algorithms by Thomas H. Cormen (Reference Book)