Geometric Algorithms (Line Intersection & Convex Hull)

Project Report

Professor Dr. Farrukh Salim (BCS-5E)

Design & Analysis of Algorithms (CS-2009)

Muhammad Talha (K21-3349)
Muhammad Hamza (K21-4579)
Muhammad Salar (K21-4619)

Foundation of Advancement of Science and Technology
National University of Computer and Emerging Sciences
Department of Computer Science
Karachi, Pakistan
Monday, November 27, 2023

Abstract

This project focuses on the implementation and analysis of geometric algorithms with varying computational complexities. Two main problems are addressed:

- determining the intersection of two line segments.
- solving the Convex Hull problem using different algorithms.

The programming design is presented using Python programming language and Jupyter Notebook. The experimental setup involves demonstrating the algorithms, calculating their time complexities, and presenting the results and discussions. The conclusion summarizes the findings, and references are provided for further exploration.

Contents

1 Introduction

Geometric algorithms play a crucial role in various applications, from computer graphics to computational geometry. This project delves into two specific problems: line segment intersection and Convex Hull computation. The former involves exploring different techniques using counter-clockwise operation and vector cross products and line sweep as an additional approach. The latter employs brute force, Jarvis-March, Graham scan, Quick Hull, and Monotone Chain.

2 Programming Design

The implementation is conducted using Python, chosen for its suitability in handling geometric computations. This section delves into the thoughtful design choices made, emphasizing the clarity and efficiency intrinsic to the implementation. The line segment intersection problem involves an exploration of diverse techniques, leveraging counter-clockwise operations, vector cross products, and line sweep approach. Counter-clockwise operations are instrumental in determining the relative orientation of points and aiding in discerning whether two line segments intersect. The utilization of vector cross-products provides a robust method for evaluating the convexity of polygons formed by these line segments. The incorporation of line sweep further enhances efficiency by exploiting the sorted order of events along a sweep line, thereby facilitating the identification of intersection points. The Convex Hull problem is addressed through a selection of algorithms, each with distinct strengths and applications. Brute Force, while computationally intensive, serves as a baseline for comparison and ensures the correctness of subsequent, more optimized algorithms. Jarvis-March, known for its simplicity, showcases a convex hull construction based on selecting the point with the smallest polar angle. Graham scan, leveraging the angular sorting of points, demonstrates an efficient convex hull construction. Quick Hull, a divide-and-conquer algorithm, demonstrates efficiency by recursively dividing the point set and constructing convex hulls for each subset. Monotone Chain, employed for its adaptability to the convex hull problem, efficiently computes the upper and lower hulls separately before merging them to obtain the complete convex hull.

3 Experimental Setup

The program draws objects on the screen using randomly selected points. The report describes the steps of each algorithm, highlighting key components of their implementation. Furthermore, a discussion on how time and space complexities are calculated for each algorithm is included. This ensures a comprehensive understanding of the computational efficiency of the implemented solutions.

4 Results and Discussion

Execution times are provided, allowing for a comparison of the various algorithms. The discussion delves into the strengths and weaknesses of each algorithm, providing insights into their practical applicability. Graphs or tables may be included to visually represent the performance differences between the algorithms.

4.1 Line Intersection

Algorithm	Time Taken
Counter-Clockwise	0.0000066757
Vector Cross Product	0.0000071526
Line Sweep	0.0023283958

4.2 Convex Hull

Algorithm	Time Complexity	Time Taken	Data-set
Graham Scan	O($nLog(n)$)	$2.992 * 10^{-3}$	500
Jarvis-March	O(nh)	$5.746 * 10^{-5}$	500
Quick Hull	O($nLog(n)$)	$3.679 * 10^{-3}$	500
Monotone Chain	O($nLog(n)$)	$9.785 * 10^{-4}$	500
Brute Force	O(n^3)	$2.264 * 10^{3}$	500

5 Conclusion

The project concludes by summarizing the key findings and insights gained through the implementation and analysis of geometric algorithms. During the implementation and analysis of the geometric algorithms, several noteworthy observations, challenges, and areas for potential future research or improvements emerged.

- Unexpected Observations:
  Some algorithms displayed varying performance based on the characteristics of input data. For example, Quick Hull exhibited exceptional speed on well-distributed point sets but faced challenges with pathological cases, emphasizing the importance of understanding algorithmic behaviour under diverse scenarios.

- Challenges Faced:
  Fine-tuning Quick Hull for better performance in edge cases posed a significant challenge. Balancing the divide-and-conquer approach with efficient merging strategies proved to be non-trivial.

- Potential Areas for Future Research or Improvements:
    1. Dynamic Input Handling:
       Extending the system to handle dynamic input updates in real-time could enhance its practical utility. This involves exploring efficient algorithms for incremental updates, enabling the system to adapt to changing geometric configurations.
    2. Parallelization for Efficiency:
       Investigating the feasibility of parallelizing certain algorithms, particularly Convex Hull computations, may lead to significant performance improvements. This could involve exploring parallel computing frameworks or algorithms specifically designed for parallel execution.
    3. Hybrid Approaches:
       Researching and implementing hybrid algorithms that combine the strengths of different approaches could yield improved overall performance. For instance, combining the strengths of Jarvis-March and Quick Hull for Convex Hull computation might result in a more robust solution.

6 References

- WikiBooks (Monotone Chain Algorithm 1)
- GeekForgeeks (Monotone Chain Algorithm 2)
- Wikipedia (Line Sweep Algorithm 1)
- GeekForgeeks (Line Sweep Algorithm 2)
- Wikipedia (QuickHull Algorithm 1)
- GeekForgeeks (QuickHull Algorithm 2)
- Introduction to Algorithms by Thomas H. Cormen (Reference Book)