Geometric Algorithms Implementation and Analysis

21K-4716 S. Uddin, 21K-3067 Y. Ali, 21K-4702 B. Allahwala

November 26, 2023

Appendix

Content for the Appendix:

- Abstract
- Introduction
- Programming Design
- Experimental Setup
- Results and Discussion
- Conclusion
- References

Abstract

This project focuses on the implementation of geometric algorithms with varying Big Oh complexities. While also elucidating the calculation of Time and Space complexities for each algorithm.

Introduction

This project focuses on implementing practical geometric algorithms. The main goals include tackling challenges related to line segment intersection and convex hull computation

Programming Design

We'll use PYTHON as the programming language due to its versatility, strong support for graphical interfaces, and extensive libraries for algorithmic implementations.

Line Intersection

Method 1: Cross-Product

Cross Product Method: Applies the cross product concept to ascertain the intersection of two line segments. It involves examining the orientation of line segments at their endpoints. The time and space complexity is O(N).

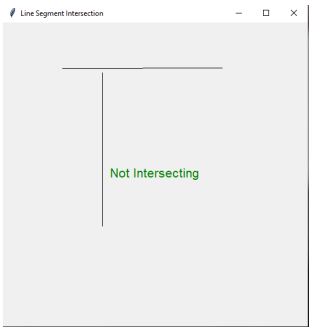

Figure 1: Method 1 Figure

Method 2: Parametric Line Equation

Parametric Equation Check: uses the lines' parametric equations to determine whether they intersect. Getting Parameters: Determines the line equations' parameters and assesses if the intersection is within the parameters of the two line segments. O(N) complexity in time.

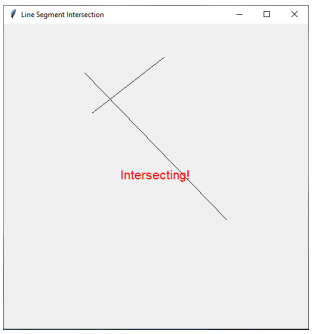

Figure 2: Method 2 Figure

Convex Hull

Graham Scan

The pivot point is first chosen based on which point has the lowest y-coordinate. The remaining points are then sorted by the algorithm according to polar angle. The convex hull is found by processing the sorted points. Complexity of Time: O(n log n) Complexity of Space: O(n).

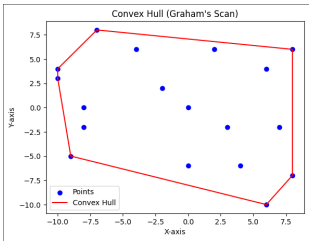

Figure 3: Graham Scan Figure

Quick Elimination

divides and conquers in order to determine the convex hull. As the diameter's ends, it chooses the two places with the lowest and maximum x-coordinates. Next, points are split into two sets according to where they are on the line that these endpoints form. Every subset receives a recursive application of the procedure. Time Complexity: Average O(n log n), worst-case O(n2) Complexity of Space: O(n).

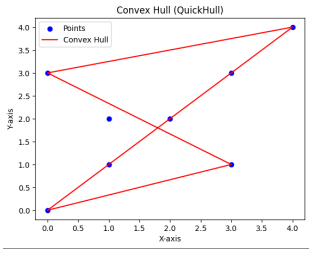

Figure 4: Quick Elimination Figure

Brute-Force

To find the convex hull, the brute-force convex hull algorithm examines every possible pairing of three points. It is inefficient for large datasets because to its cubic time complexity of O(n3) and space complexity of O(n).

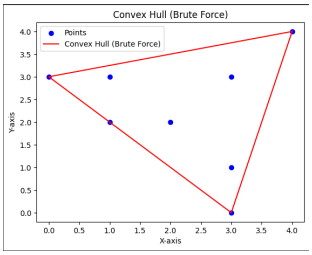

Figure 5: Brute Force Figure

Jarvis-March

A straightforward method for determining the convex hull of a set of points in the plane is Jarvis's March, commonly referred to as the Gift-Wrapping algorithm. Time Complexity: O(nh), where h is the number of points on the convex hull and n is the number of input points. Complexity of Space: O(h).

Monotone-Chain

The Monotone Chain Convex Hull algorithm exhibits a time complexity of O(n log n), primarily driven by the initial sorting step of the input points based

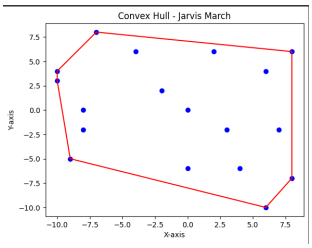

Figure 6: Jarvis March Figure

on their x-coordinates. The construction of the upper and lower hulls and their subsequent combination contribute linearly to the overall efficiency of the algorithm, resulting in a space complexity of O(n).

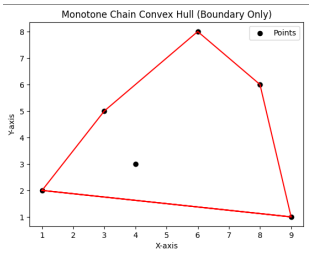

Figure 7: Monotone-Chain Figure

Experimental Setup

Using Matplotlib and Tkinter tools, We created a graphical user interface (GUI) for PyCharm that displayed geometric objects and Matplotlib for algorithmic outputs. The code is available for additional research and possible improvements.

Results and Discussion

The intended results were achieved. All the algorithms worked smoothly and provided the intended outcome.

Conclusion

In concluding this project proposal for the implementation and analysis of geometric algorithms, we recognize the significance of advancing our understanding of fundamental computational geometry problems. The proposed project is designed to provide a comprehensive exploration of geometric algorithms, encompassing line segment intersection and convex hull solutions

References

1. CHATGPT
2. Slides
3. https://www.youtube.com/watch?v=B2AJoQSZf4Mt=353s
4. https://www.youtube.com/watch?v=bbTqI0oqL5U
5. https://www.tutorialspoint.com/Graham-Scan-Algorithm