Project Report

Muhammad Talha, Syed Ismail, Ibad Khan

November 27, 2023

Abstract

This project explores the implementation and comparative analysis of convex hull algorithms in Python, along with an investigation into line intersection methods. Convex hulls are fundamental computational geometry structures widely used in various applications, such as computer graphics, image processing, and geographic information systems. Our project focuses on coding and evaluating different convex hull algorithms, including Graham's Scan, Jarvis March, and Quick Elimination and Chans Algorithm, in terms of their efficiency and robustness.

1 Introduction

In this project, we dive into the world of convex hull and line intersection algorithms, essential tools in solving geometric problems across fields like computer graphics and geographic information systems. Through the implementation of well-known algorithms like QuickHull, Jarvis March, Graham Scan, and Chan's Algorithm and line intersection algorithms, we aim to unravel the efficiency and versatility of these techniques.

2 Programming Design

This project is implmented in Python Pygame library to display GUI where the codebase has been organized into separate files, Each algorithm is encapsulated in its respective file, promoting algorithmic abstraction and facilitating potential future extensions. The main menu, centralizing user interaction, is implemented in a dedicated file (MainMenu.py).This design choice prioritizes clarity, ease of use, and adaptability for future algorithmic additions.

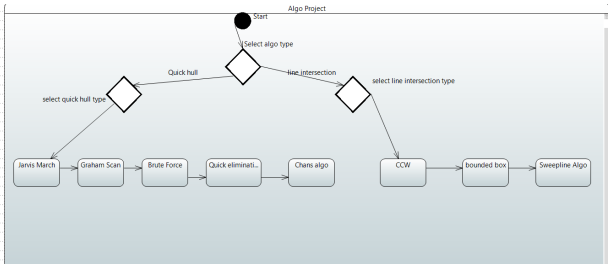

Figure 1: System Diagram

3 Experimental Setup

The experimental setup involves a comprehensive evaluation of convex hull and line intersection algorithms implemented in Python. The project considers three distinct sets of 2D points for convex hull algorithms—randomly distributed points (Set A), points arranged in a circular pattern (Set B), and a densely clustered point set (Set C). For line intersection algorithms, random line segments (Set X), collinear line segments (Set Y), and intersecting line segments (Set Z) are employed.

4 Results and Discussion

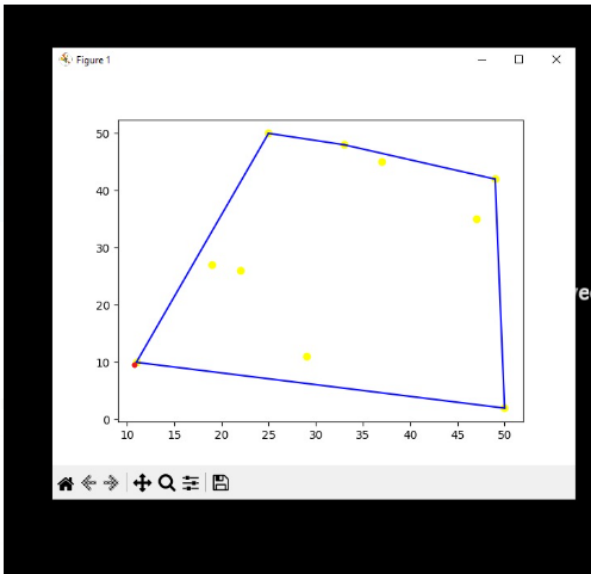

Figure 2: Brute force

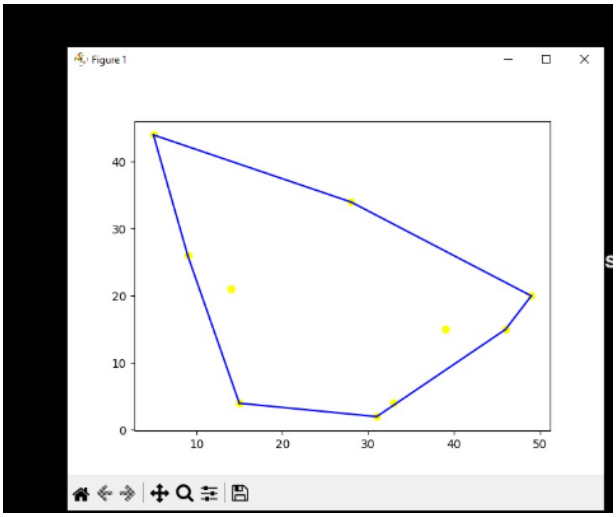

Figure 4: Graham scan

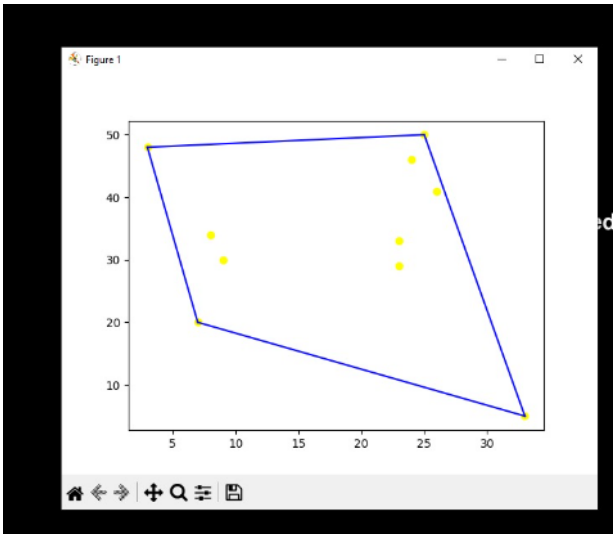

Figure 3: Jarvis March

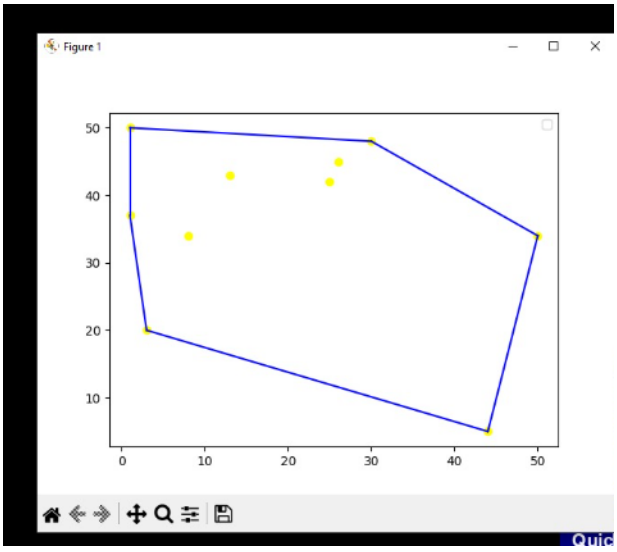

Figure 5: Quick hull

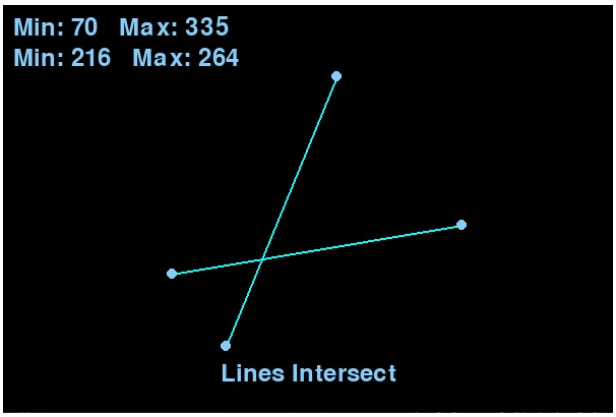

Figure 6: Bounded box line intersection

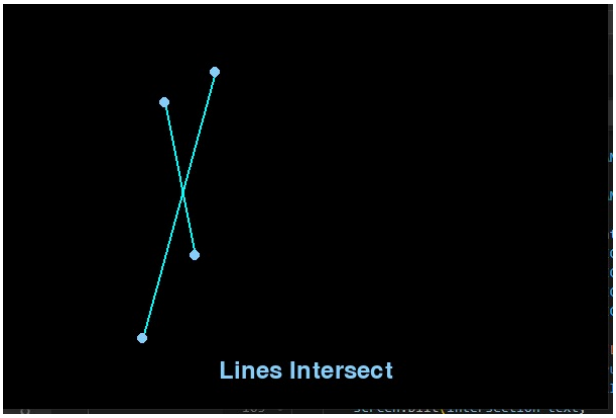

Figure 8: Sweepline polygon

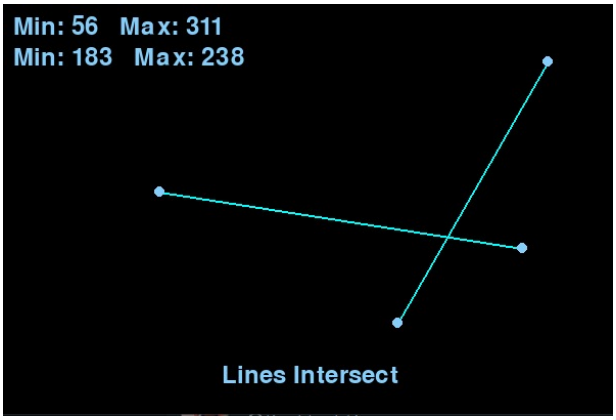

Figure 7: CCW orientation

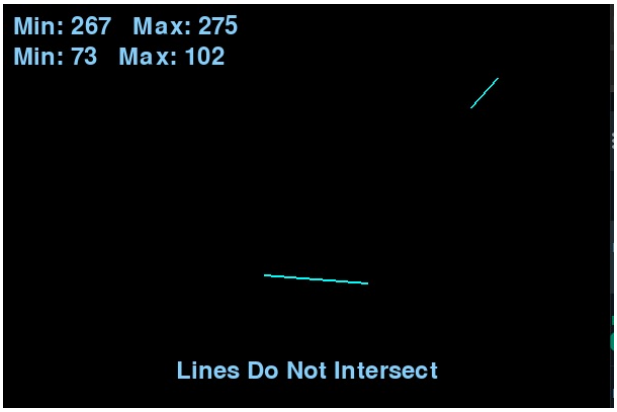

Figure 9: No intersection bounded box

Algorithm	Time Complexity
Brute Force	$O(n^3)$
Jarvis March	$O(nh)$
Graham Scan	$O(nlogn)$
Quick Elimination	$O(n^2)$
Chan's Algorithm	$O(nlogh)$

Algorithm	Space Complexity
Brute Force	$O(h)$
Jarvis March	$O(n+h)$
Graham Scan	$O(n+h)$
Quick Elimination	$O(n)$
Chan's Algorithm	$O(n+h)$

Algorithm	Time (ms)	Time (s)
Brute Force	2.62594	150
Jarvis March	0.49686	20
Graham Scan	0.62274	7
Quick Elimination	-	-
Chan's Algorithm	2.00223	10

5 Discussion

Chan's Algorithm achieves O(n log h) time complexity using a divide-and-conquer approach. Jarvis March has a worst-case time complexity of O(nh) and selects points based on polar angles. Brute Force compares all point pairs, resulting in $O(n^3)$.

6 Conclusion

In conclusion, the convex hull algorithms (such as Chan's, Jarvis March, Brute Force, Quick Elimination, and Graham Scan) and line intersection algorithms we explored in this project provide diverse solutions for geometric problems. Each algorithm comes with its own set of trade-offs in terms of time and space complexity, making them suitable for different scenarios. Understanding and implementing these algorithms enables us for efficient computation of convex hulls and identification of intersections in various applications.

7 References

https://www.geeksforgeeks.org/convex-hull-using-graham-scan/
https://graphics.stanford.edu/courses/cs268-16-fall/Notes/cmsc754-lects.pdf
https://www.geeksforgeeks.org/given-a-set-of-line-segments-find-if-any-two-segments-intersect/
https://keymakr.com/blog/what-are-bounding-boxes/
https://jeffe.cs.illinois.edu/teaching/compgeom/notes/14-convexhull.pdf
https://bryceboe.com/2006/10/23/line-segment-intersection-algorithm/