Presented to:
Sir Faisal Ali

Project by:

Shaikh Abdul Rafay
Minal Alam
Rayyan Ahmed

Implementation of Convex Hull and Line Intersection Algorithms

Shaikh Abdul Rafay 21K-3051
Minal Alam 21K-3072
Rayyan Ahmed 21K-3079

December 9, 2023

Abstract

This project's primary objective is to implement and contrast various convex hull algorithms, namely Graham Scan, Jarvis March, Brute Force, Quick Elimination and Andrew's Monotone Chain and Line Intersection Algorithms . The analysis focuses on assessing their performance, execution duration, and efficiency in constructing convex hulls using a given set of points within a two-dimensional plane.

1 Introduction

Convex hull algorithms serve as fundamental tools in computational geometry, finding wide-ranging applications in computer graphics, geographic information systems, and robotics. This project centers on the implementation and detailed assessment of five prominent convex hull algorithms. The focus lies in examining their performance characteristics across various computational domains.

Line intersection algorithms are fundamental in computational geometry, offering solutions across diverse fields like computer graphics, GIS, robotics, and CAD. They're pivotal for identifying intersection points between lines, crucial in collision detection, route planning, and shape analysis.

This project aims to implement and analyze line intersection algorithms, exploring their efficiency, accuracy, and computational complexity. Through this scrutiny, the study aims to reveal their performance in varied scenarios and suitability across computational geometry applications.

2 Programming Design

2.1 Programming Language and Tools Used

The implementation was done in Java utilizing the Java Swing library for graphical user interface development. Java Swing provides a robust set of tools for building interactive graphical applications, making it suitable for visualizing convex hull construction algorithms. Line Intersection was implemented with Python and using the relevant Python libraries such as tkinter for gui and matplotlib for graphs and numpy for computations.

2.2 System Diagram

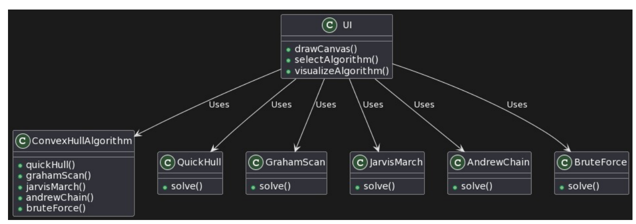

Figure 1: Class

2.3 Discussion

The algorithms were chosen due to their distinct methodologies in addressing the convex hull problem. Graham Scan, Jarvis March, Brute Force, Quick Elimination, and Andrew's Monotone Chain were employed for comparison, aiming to assess their effectiveness and efficiency.

3 Experimental Setup

The experiments were conducted on a machine with an Intel Core i7 with 4Gen processor. Different data points were used to test the algorithms' performance.

4 Results and Discussion

4.1 Execution Time Comparison

The execution times of each algorithm were recorded and compared. Table 1 summarizes the results.

Algorithm	Execution Time (ms)
Brute Force	$O(n^3)$
Graham Scan	O(n log n)
Jarvis March	O(nh)
Andrew's Monotone Chain	O(n log n)
Quick Elimination	O(n log n)
Counter Clockwise	$O(n^3)$
Line By Line	$O(n^3)$
Bounding Box	$O(n^3)$

Table 1: Execution Time Comparison

4.2 Algorithm Performance Comparison

The accuracy and efficiency of the algorithms in constructing the convex hull for different data points were visually compared.

4.2.1 Convex Hull Algorithms

1. **Brute Force**

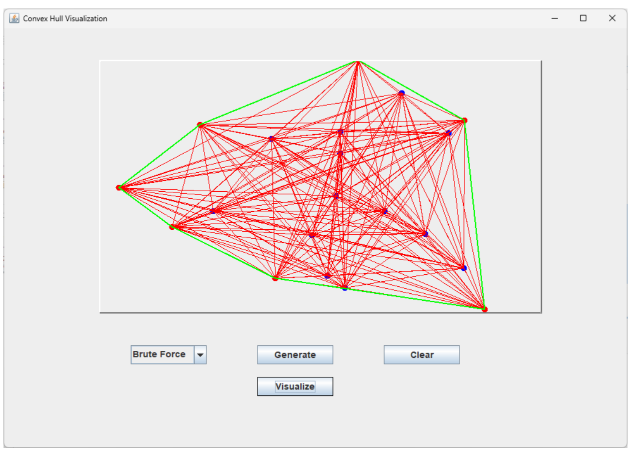

Figure 2: Brute Force Visualization

2. **Graham Scan**

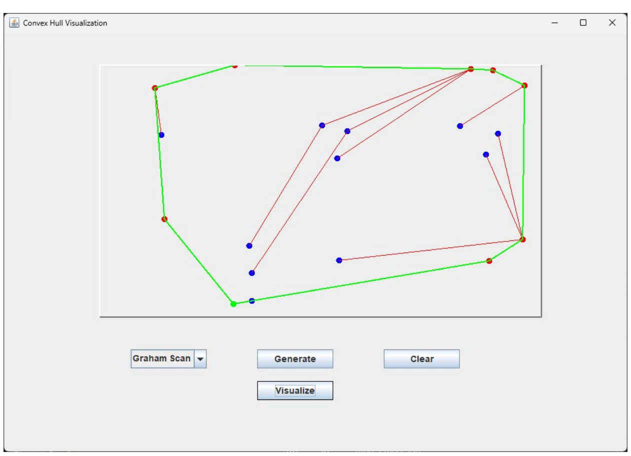

Figure 3: Graham Scan Visualization

3. **Jarvis March**

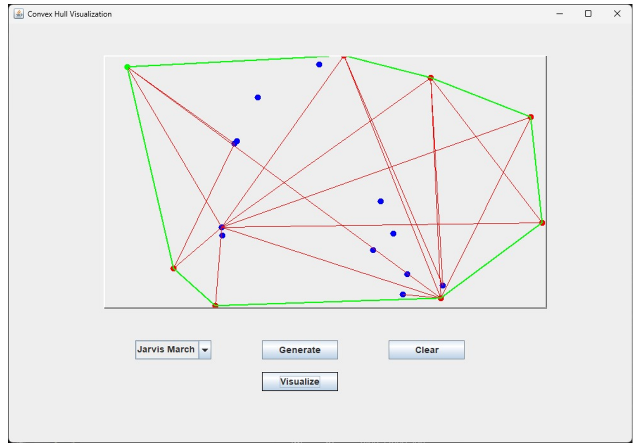

Figure 4: Jarvis March Visualization

4. **Andrew Monotone Chain**

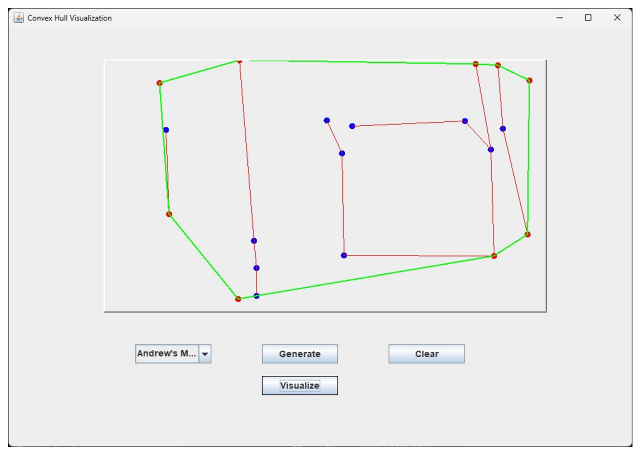

Figure 5: Andrew Monotone Chain Visualization

5. **Quick Elimination**

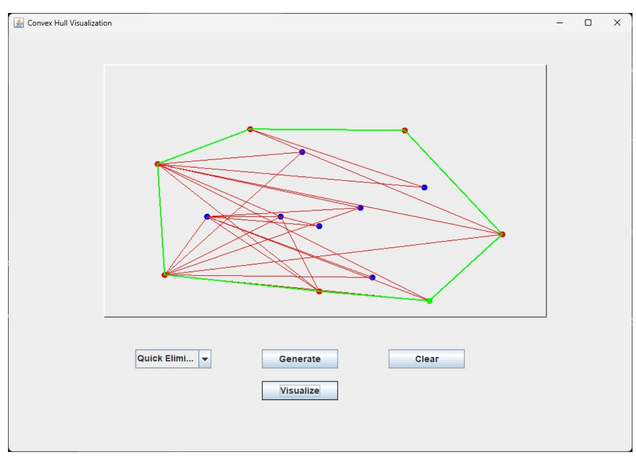

Figure 6: Quick Elimination Visualization

4.2.2 Line Intersection Algorithms

1. **Counter Clockwise**

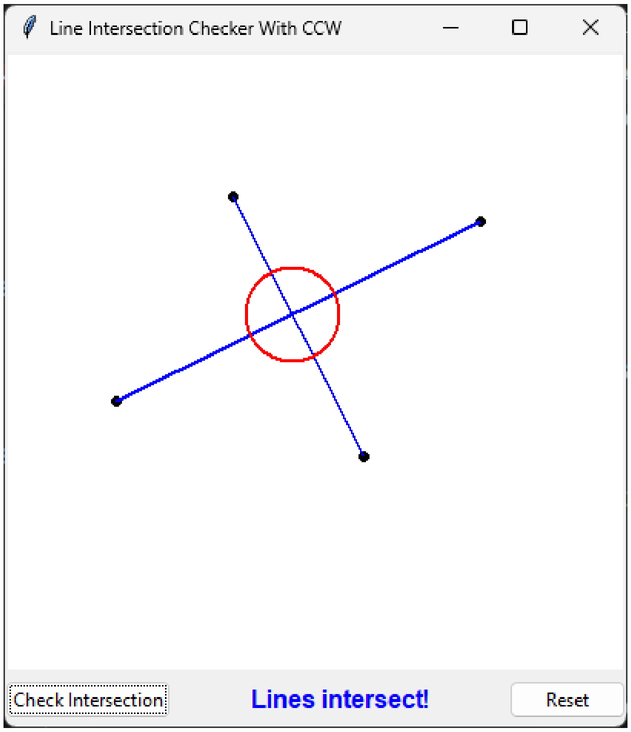

Figure 7: Counter Clockwise Visualization

2. **Line By Line**

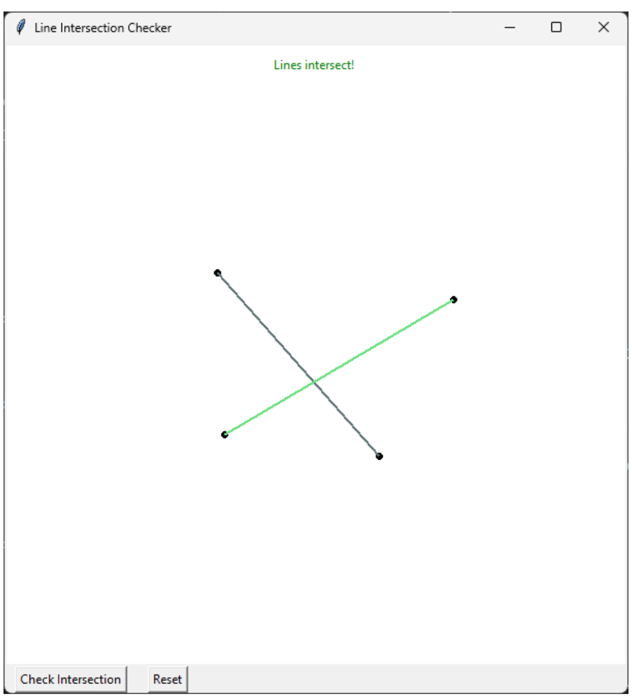

Figure 8: Cross Product Visualization

3. **Bounding Box**

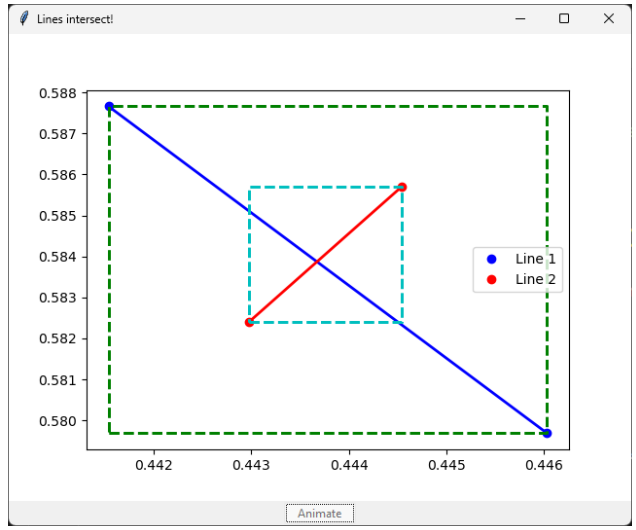

Figure 9: Bounding Box Visualization

4.3.1 Convex Hull Algorithms

4.3 Comparison

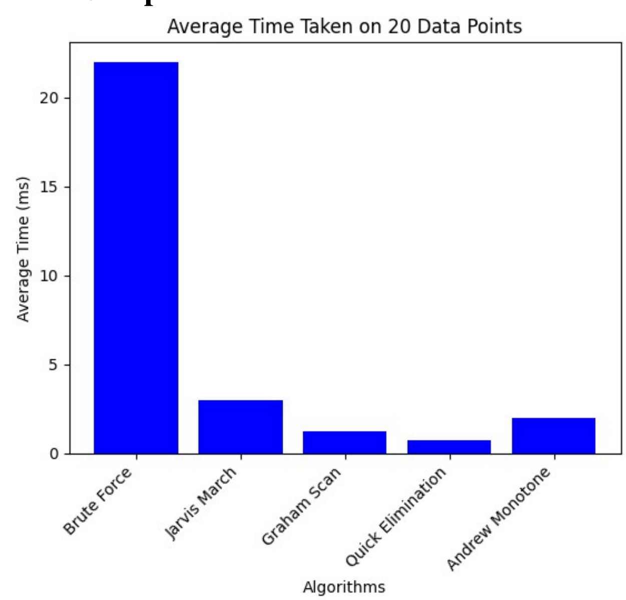

Figure 10: Time Comparison Of Convex Hull Algorithms

Notably, the ”Brute Force” algorithm exhibits a comparatively higher time requirement, while the ”Jarvis March” and ”Graham Scan” algorithms demonstrate more efficient performances. The ”Quick Elimination” and ”Andrew Monotone” algorithms showcase minimal time requirements, suggesting their effectiveness in rapidly identifying the convex hulls within the given data points. This analysis provides insights into the computational efficiency of different convex hull algorithms in handling a set of 20 data points.

4.3.2 Line Intersection Algorithms

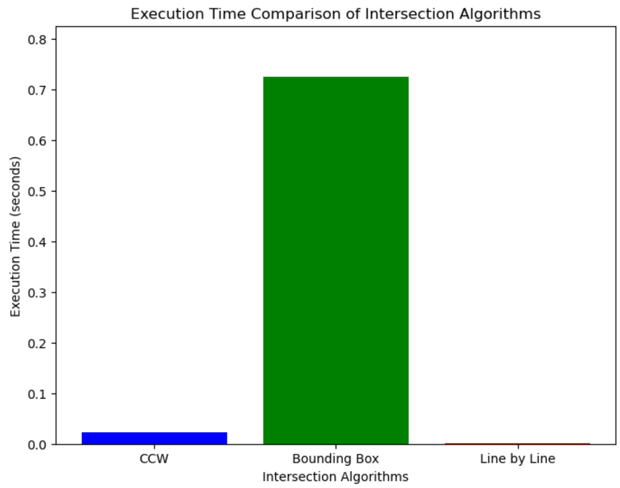

Figure 11: Time Comparison Of Line Intersection Algorithms

The execution times indicate that each algorithm has its own trade-offs, and the choice depends on the specific needs of the application. The ”CCW” algorithm is particularly efficient, the ”Bounding Box” algorithm provides simplicity, and the ”Cross Product” test offers a balance between accuracy and computational cost.

5 Conclusion

The study presented a comparative analysis of various convex hull algorithms. Each algorithm showed distinct performance characteristics and trade-offs. Based on the experiments conducted, conclusions were drawn regarding the most efficient algorithms for specific scenarios.

6 References

Chan, T. M. (1996). ”Optimal output-sensitive convex hull algorithms in two and three dimensions.” Discrete Computational Geometry, 16(4), 361–368. DOI: 10.1007/BF02573991

Preparata, F. P., Hong, S. J. (1981). ”Convex hulls of finite sets of points in two and three dimensions.” Communications of the ACM, 24(10), 620–624. DOI: 10.1145/358728.358773

Kirkpatrick, D. G., Seidel, R. (1986). ”The ultimate planar convex hull algorithm?” SIAM Journal on Computing, 15(1), 287–