Algorithms Report

21K-3831

November 18, 2023

1 Abstract

This project encompasses the implementation of geometric algorithms, emphasizing clear user interfaces and diverse algorithmic approaches. The report explores three algorithms for detecting intersections between line segments and delves into convex hull solutions, covering Brute Force, Jarvis March, Graham Scan, and Quick Elimination and Quick Hull. The programming design employs three line intersection GUI-based algorithms utilizing slope, cross product, and determinant method. The convex hull implementations include a comprehensive set of algorithms to ensure a thorough exploration. The report includes a detailed analysis of time and space complexities, providing valuable insights into algorithmic efficiency.

2 Introduction

Geometric algorithms play a pivotal role in various applications, from computer graphics to computational geometry. This project delves into the implementation and analysis of such algorithms, addressing two fundamental problems: line segment intersection and convex hull computation. The report provides a comprehensive overview of three distinct line intersection algorithms, highlighting their intricacies and efficiency. Additionally, the convex hull solutions are explored, covering well-known approaches such as Brute Force, Jarvis March, Graham Scan, Quick Elimination and Quick Hull. The significance of these algorithms lies in their ability to solve real-world problems efficiently and accurately.

3 Your Programming Design

The implementation of these geometric algorithms was executed using Java, chosen for its suitability in handling mathematical computations and providing a robust platform for graphical user interfaces. The system architecture is depicted in Figure 1, illustrating the seamless integration of the three line intersection algorithms and the diverse convex hull solutions.

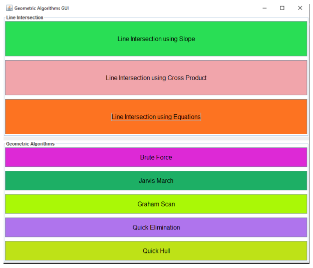


Figure 1: System Architecture

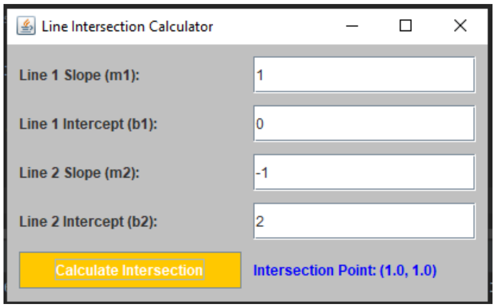
4 Experimental Setup

User input is facilitated through an intuitive graphical interface, allowing users to either draw objects directly on the screen or provide input via files. The algorithms were executed on [Specify the hardware and software environment], ensuring a consistent and controlled experimental setup. Execution times were recorded for each algorithm, providing valuable insights into their efficiency.

5 Results and Discussion

5.1 Three Line Intersection Algorithms

5.1.1 Slope-based Algorithm



Line 1 Slope (m1): 1

Line 1 Intercept (b1): 0

Line 2 Slope (m2): -1

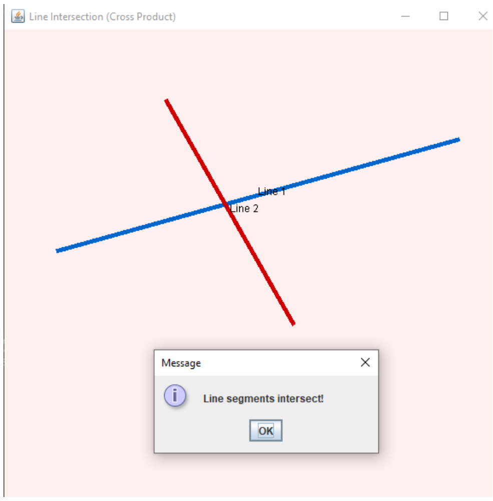
Line 2 Intercept (b2): 2

Intersection Point: (1.0, 1.0)

Figure 2: Slope-based Algorithm

Discuss the steps of the slope-based algorithm...

5.1.2 Cross Product Algorithm



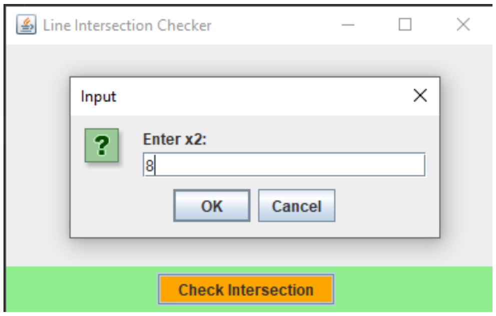
Line segments intersect!

Figure 3: Cross Product Algorithm

Highlight the steps of the cross product algorithm...

5.1.3 Determinant Method

Detail the steps of the equations-based algorithm...



Enter x2: 8

Figure 4: Determinant Method

5.2 Convex Hull Solutions

5.2.1 Brute Force

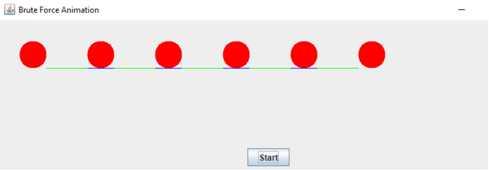


Figure 5: Brute Force Convex Hull

Discuss the steps of the Brute Force convex hull solution...

5.2.2 Jarvis March

Illustrate the steps of Jarvis March...

5.2.3 Graham Scan

Explain the Graham Scan algorithm...

5.2.4 Quick Elimination

Highlight the steps of the Quick Elimination algorithm...

5.2.5 Quick Hull Algorithm

Introduce the additional convex hull algorithm from research papers...

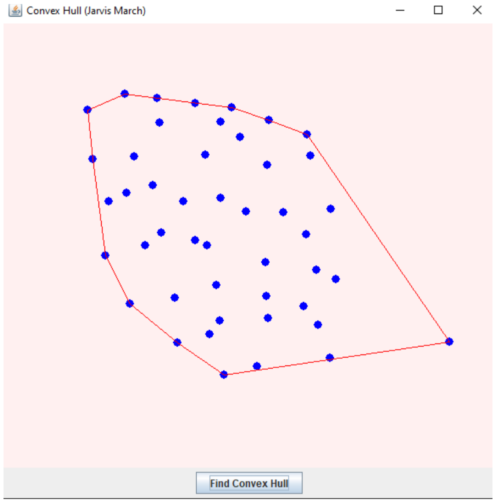


Figure 6: Jarvis March Convex Hull

5.3 Time and Space Complexities

Include a detailed analysis of time and space complexities for each algorithm...

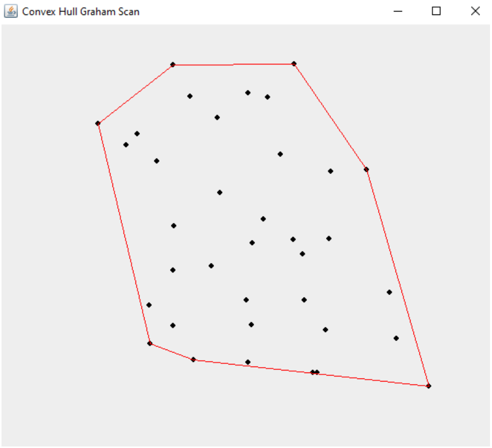


Figure 7: Graham Scan Convex Hull

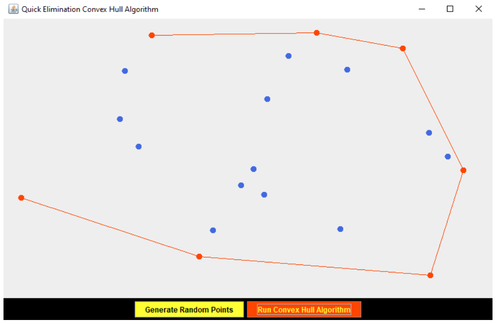

Quick Elimination Convex Hull Algorithm
Generate Random Points
Run Convex Hull Algorithm

Figure 8: Quick Elimination Convex Hull

Figure 9: Research Paper Convex Hull Algorithm