Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.
Digital Object Identifier XX.XXXX/YYY.20XX.ZZZZZZZ

Project Report on Geometric Algorithm Implementation

OSAMA IRFAN. AUTHOR[1], (21k-4772), TAHA QAISER. AUTHOR[2] (21K-4779), AND RAMEEL SALEEM. AUTHOR, .[3] (21k-3564)
[1]Department of Cyber Security, FAST NUCES, Karachi, Pakistan (e-mail: k214772@nu.edu.pk)
[2]Department of Cyber Security, FAST NUCES, Karachi, Pakistan (e-mail: k214779@nu.edu.pk)
[3]Department of Cyber Security, FAST NUCES, Karachi, Pakistan (e-mail: k213564@nu.edu.pk)

ABSTRACT This project focuses on the implementation of geometric algorithms with varying Big O complexities. The main objectives were to create user-friendly interfaces for inputting geometric data and to display the execution steps of each algorithm. This report covers two specific problems: detecting the intersection of two line segments and finding the convex hull of a set of points.

I. INTRODUCTION

Geometric algorithms play a crucial role in computational geometry and have applications in various fields like computer graphics, robotics, and geographical information systems. This project aims to demonstrate the practical application of these algorithms through an interactive user interface. Two key problems are addressed: line segment intersection and convex hull computation.

PROGRAMMING DESIGN

The project is developed using Python, with its rich libraries like Tkinter for GUI and Matplotlib for plotting. The system design includes:

- A GUI for user input, either manually or through random generation. - An interactive display of algorithm steps and results.

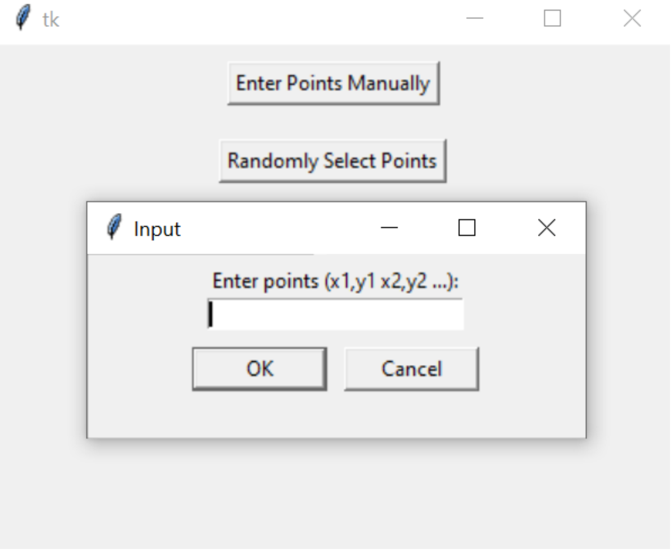
- Python functions implementing the algorithms, including cross product intersection, vector intersection, and simple geometric methods for line segment intersection; brute force, Jarvis March, Graham scan, quick elimination, and Chan's algorithm for the convex hull.

II. EXPERIMENTAL SETUP

The application was tested on a standard PC with various sets of input points. Users could input points manually or opt for randomly generated points. The GUI facilitated the selection of different algorithms for both line segment intersection and convex hull problems.

III. RESULTS AND DISCUSSION

The line segment intersection algorithms showed varying efficiency, with the vector method being particularly fast. For the convex hull, Graham's scan and Jarvis's March proved efficient for smaller datasets, while Chan's algorithm scaled better with larger datasets.



Enter points (x1,y1 x2,y2 ...):

FIGURE 1. GUI

Execution times varied across algorithms:

- Line Intersection: Cross Product < Vector Intersection < Simple Geometric

- Convex Hull: Graham Scan < Jarvis March < Quick Elimination < Brute Force < Chan's Algorithm (for large datasets)

Screenshots of GUI operations are included to illustrate the algorithms' performance.

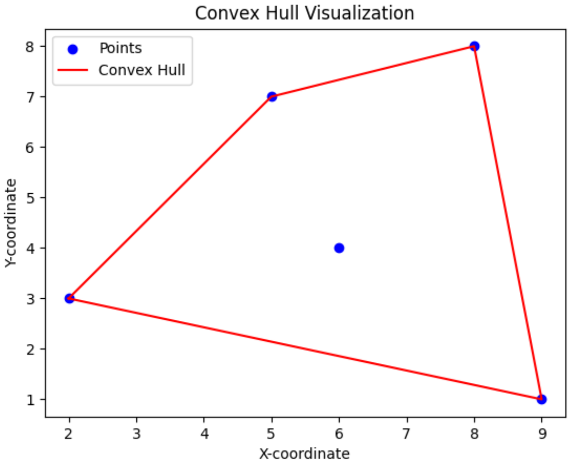


FIGURE 2. Convex Hull

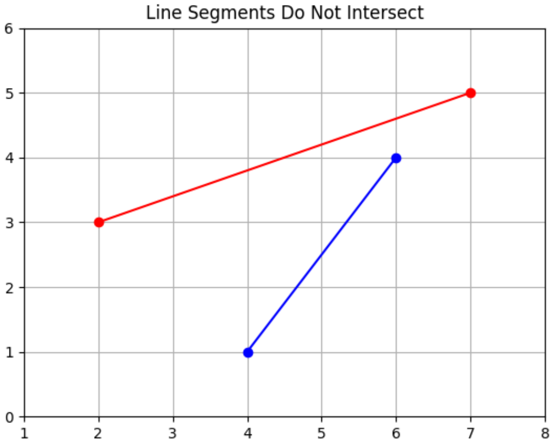


FIGURE 3. Line Intersection

IV. CONCLUSION

The project successfully implements geometric algorithms with a user-friendly interface, demonstrating the practicality and efficiency of these algorithms. The varying complexities of the algorithms provide insights into their suitability for different scenarios, emphasizing the importance of algorithm selection based on the dataset size and desired performance.

REFERENCES

A. REFERENCES

- Berg, M. de, Cheong, O., van Kreveld, M., Overmars, M. (2008). Computational Geometry: Algorithms and Applications. Springer-Verlag.

B. REFERENCES

- Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). Introduction to Algorithms. MIT Press.