Assignment 1 Report

KevinLi yl9314

The Screenshot and Recording videos are too big to upload to Github, so here’s the link to google drive!

<https://drive.google.com/drive/folders/1TL43iExgp_Km1PZ3rXtyzXQZp1lMgRAg?usp=drive_link>

Description

This assignment involves the implementation of Graham's scan algorithm to compute the convex hull of a set of 2D points and saving the resulting convex hull as an OBJ file. The code successfully accomplishes these tasks.

Operating System and Compiler

Operating System: macOS Big Sur (Version 11.2.3)

Compiler: g++ (Apple clang version 13.0.0)

Convex Hull Implementation

Graham's Scan Algorithm

The code correctly implements Graham's scan algorithm to compute the convex hull of a set of 2D points. The algorithm begins by identifying the lowest-leftmost point in the input point cloud and then sorts all input points in counterclockwise order concerning this reference point. It subsequently constructs the convex hull incrementally by checking for right turns and eliminating middle points as required. The implementation produces accurate convex hulls for various input datasets, and the results were validated by visual inspection using Meshlab.

Point In Polygon Test Implementation

The code includes an implementation of the Point In Polygon test based on the even-odd rule. This algorithm determines whether a given point is inside or outside a polygon by projecting a ray from the query point to a point far outside the polygon and counting the number of intersections with the polygon edges. The implementation was tested with various datasets and consistently provided correct results for identifying points inside and outside the polygon.

Results and Comments

The convex hull algorithm efficiently computes the convex hull of a point cloud.

The code compiles and runs successfully on macOS using the g++ compiler.

The implementation adheres to the template provided in the starter code.

The code saves the resulting convex hull as an OBJ file, which is compatible with visualization tools like Meshlab.

This assignment enhanced my understanding of computational geometry algorithms and their implementation in C++. The provided code fulfills the specified tasks effectively and produces reliable results.

Both implemented algorithms produce correct results for the provided datasets.

The Point In Polygon test correctly classifies points as inside or outside the polygon using the even-odd rule.