CS 6160 Project 3 Performance Report

After successfully completing all three algorithms, namely Quick Hull, Graham Scan, and Gift- Wrapping, a performance analysis was conducted on a set of 1000 randomly generated points. A performance test class was designed to iterate through 10 times the number of points in each iteration to ensure robust testing. Three different randomly generated point sets were utilized to better understand how the algorithms scale.

One configuration of random points consisted of regions of points arranged in the shape of a triangle. The next configuration featured three distinct points placed in a triangular formation, with the fourth vertex of the four-sided polygon coming from a region of randomly generated points in the middle of the triangle. The final configuration featured randomly generated points within a square region.

Upon running the performance test class on the various point configurations, different combinations of performances were observed from the three algorithms. Quick Hull's performance slightly increased as the number of points per iteration increased in all three random point configurations. Graham Scan's performance increased in the first point setup and remained relatively stable in the other two configurations as the number of points per iteration increased. The Gift-Wrapping algorithm, on the other hand, experienced a slight decrease in performance as the number of points increased per iteration. A similar trend was observed when the maximum number of points was increased from 1000 to 10000 and the number of iterations from 100 to 1000.

Based on the experimental results, it can be concluded that the Quick Hull algorithm provides the best performance out of the three algorithms. Graham Scan's performance is also similar to Quick Hull, but as the number of points increases, Quick Hull becomes slightly more efficient than Graham. The Gift-Wrapping algorithm is the worst performing algorithm overall, and its performance further deteriorates as the number of points increases.