CS478 - Computational Geometry

Implementing Three Voronoi Diagram Computation Algorithms and Comparing Their Performance

PROJECT PROPOSAL

Kerem Erdal - 21901596
Emre Erdal - 21901597

Instructor: Uğur Güdükbay

25/02/2023
The main aim of the project is to implement a program in order to deal with calculation and visualization of the “Voronoi Diagram” of some points that are generated randomly in two dimensions. This program will handle calculation of the “Voronoi Diagram” by using three different approaches. These approaches include the “Randomized Incremental Algorithm”, “Fortune's Algorithm”, and “The Flipping Algorithm”.

In the program, there will be randomly generated points in a 2D plane according to various distributions taken in the input, and the graphical output as a result of the “Voronoi Diagram” will be visible after the calculations. The program will have a user-friendly interface that would allow specifying the number of points, zooming in/out, and translating during the visualization of the “Voronoi Diagram”. In addition, to make the graphical output more clear and understandable, the different “Voronoi Cells” will have different colors. Lastly, the steps of the “Fortune’s Algorithm” will be visible while visualizing the “Voronoi Diagram” as a step by step animation.

Brief description of these mentioned algorithms can be done as follows. In the Randomized Incremental Algorithm, a random vertex can be added incrementally at each step to update the Voronoi diagram. At the end, the final diagram will be computed after adding all vertices, which would have expected worst case running time of $O(n \log n)$. Furthermore, in Fortune's Algorithm, a sweep line moves across the plane and the points to the left of the line will be considered to construct the Voronoi diagram. The final result would be obtained as the sweep line reaches to the end, and it will have an expected runtime of $O(n \log n)$. Finally, The Flipping Algorithm starts with an arbitrary triangulation using lexicographic triangulation and one of the edges of non-Delaunay triangles can be flipped until no triangle is non-Delaunay. At the end, Voronoi diagram can be obtained in an overall runtime of $O(n^2)$.

To sum up, this program will calculate and visualize the “Voronoi Diagram” of some points that are generated randomly by user specified parameters via three different algorithms. Finally, this program will be tested for different random points and the performance of the calculations will be reported. After making sufficient tests, these different algorithms will be compared in the report.