

Computational Geometry Aufgabe 3

Aufgabenstellung:

Ziel der Aufgabe ist es ein Programm zu schreiben, in welches gegebene Liniensegmente eingelesen werden können, welches dann die Anzahl an Schnittpunkten liefert. Hierbei soll ein Sweepeline-Algorithmus zum Einsatz kommen.

Vorgehensweise:

Wie in Aufgabe 1 wird zunächst die Datei mit den Liniensegmenten Zeilenweise eingelesen. Allerdings werden die einzelnen Koordinaten hier bereits für jedes Segment nach X-Koordinate sortiert. Dies erspart eine spätere Sortierung beim Erstellen der Start/Ende Events. Nach dem Einlesen werden Events für den Start und das Ende eines Liniensegments erstellt. Alle Events werden in X-Richtung sortiert. Die Datenstruktur dafür ist ein `std::set`, welches als Event Queue dient. Für die Sweepeline-Struktur wird ein `std::vector` verwendet. Diese Struktur unterstützt den notwendigen „swap“ von zwei Elementen, jedoch leider nicht das Einfügen in $O(1)$, sondern nur in $O(n)$, allerdings ist die Sweepeline in der Regel deutlich kleiner als die gesamte Anzahl an Liniensegmenten, weshalb dies nicht allzu sehr ins Gewicht fällt.

Auswertung:

Es ergeben sich folgende Messergebnisse (20 Durchläufe, gemittelt Resultat):

Average test results for file 's_1000_10.dat':	Average test results for file 's_1000_10.dat':
input time: 2.7 ms	input time: 2.65 ms
intersection calculation time: 4.05 ms	intersection calculation time: 9.75 ms
total time taken: 6.75 ms	total time taken: 12.4 ms
number of intersections found: 796	number of intersections found: 796
Average test results for file 's_1000_1.dat':	Average test results for file 's_1000_1.dat':
input time: 2.75 ms	input time: 2.2 ms
intersection calculation time: 2.05 ms	intersection calculation time: 7.75 ms
total time taken: 4.8 ms	total time taken: 9.95 ms
number of intersections found: 7	number of intersections found: 11
Average test results for file 's_10000_1.dat':	Average test results for file 's_10000_1.dat':
input time: 21.6 ms	input time: 18.85 ms
intersection calculation time: 25.05 ms	intersection calculation time: 601 ms
total time taken: 46.65 ms	total time taken: 619.85 ms
number of intersections found: 725	number of intersections found: 732
Average test results for file 's_100000_1.dat':	Average test results for file 's_100000_1.dat':
input time: 195.6 ms	input time: 198.75 ms
intersection calculation time: 777.5 ms	intersection calculation time: 63252.4 ms
total time taken: 973.1 ms	total time taken: 63451.2 ms
number of intersections found: 77117	number of intersections found: 77125

Links ist das Ergebnis des Sweepeline-Algorithmus zu sehen, rechts das des „naiven“ Ansatzes. Für die Dateien `s_1000_1`, `s_10000_1` und `s_100000_1` können über die Sweepeline nicht alle Schnittpunkte ermittelt werden, denn bei den restlichen Schnittpunkten handelt es sich um Grenzfälle, welche in der hier Implementierten Lösung nicht behandelt wurden. Lediglich in `s_1000_10` liefern beide das gleiche Ergebnis. Es ist anzunehmen, dass dort die Grenzfälle nicht vorhanden sind, welche bei der Sweepeline zu Problemen führen (z.B. Vertikale Liniensegmente, Liniensegmente bestehend aus einem Punkt, ...). Bei 100000 Liniensegmenten wird das Ausmaß des Zeit-Komplexitätsunterschiedes der beiden Algorithmen deutlich: Während mit der Sweepeline die Schnittpunkte innerhalb von einer Sekunde gefunden werden können, so benötigt der „naive“ Ansatz schon über eine Minute.