

Praktikumsaufgabe 3

in der Vorlesung

Computational Geometry

Bestimmung von Schnittpunkten aus einem
gegebenen Satz von Strecken mit Hilfe des
Sweep Line-Algorithmus

| | |
|------------------|---------------------------------|
| Team: | Christopher Hinz, Tobias Gruber |
| Studiengruppe: | Master Informatik |
| Studiensemester: | 1. Semester |
| Schwerpunkt: | Embedded Computing |

11.04.2022

Sommersemester 2022

Inhaltsverzeichnis

| | | |
|----------|-------------------------------------|----------|
| 1 | Einführung | 3 |
| 2 | Methoden | 3 |
| 3 | Ergebnisse | 3 |
| 4 | Verifikation und Validierung | 3 |

1 Einführung

Im Rahmen des 3. Praktikums sollen die Koordinaten von Linien aus verschiedenen dat-Dateien eingelesen und auf Schnittpunkte überprüft werden. Hierzu soll neben der Anzahl an gefundenen Schnittpunkten auch die Laufzeit des Programms ausgegeben werden.

2 Methoden

Zur Lösung der Problemstellung soll ein Line Sweep Algorithmus implementiert werden. Dieser und alle benötigten Hilfsfunktionen wurden hierbei in C++ implementiert. Hierbei sei folgendes vorweggenommen. Die Dateien des ersten Praktikums enthalten Linien die nach den getroffenen Einschränkungen des Line Sweep Algorithmus aussortiert oder ignoriert werden müssen. Hierfür wurde eine Routine implementiert die jede dat-Datei auf einliest und überprüft. Es mussten hierbei nur bei den Dateien des ersten Praktikums Linien aussortiert werden, was deren leicht veränderte Größe (z.B 997 statt 1001) im nächsten Abschnitt erklärt.

3 Ergebnisse

Nun sollen die Laufzeiten verglichen werden. Um diese Neue Datei für P3:

- s_1000_10.dat:
Schnittpunkte: 789, Laufzeit: 82.164 ms

Dateien aus P1:

- s_1000_1.dat:
reduziert auf 997, Schnittpunkte: 4, Laufzeit: 38.592 ms
- s_10000_1.dat:
reduziert auf 9997, Schnittpunkte: 725, Laufzeit: 1956.78 ms
- s_100000_1.dat:
reduziert auf 99985, Schnittpunkte: 76087, Laufzeit: 556070 ms

4 Verifikation und Validierung

Zur Verifikation wurden verschiedene Testfälle definiert, getestet und überprüft. Außerdem wurden insbesondere die Ergebnisse mit matplotlib geplottet. Hierzu wurden neben den gegebenen Strecken aus den dat-Dateien die gefundenen Schnittpunkte geplottet. Damit wurde im Laufe der Implementierung der Algorithmus nachgebessert. Hierbei wurden noch nicht gefundenen Schnittpunkte und die beteiligten Strecken isoliert und

als Testfälle verwendet.

Untenstehendes Bild zeigt den erwähnten Plot. Zu sehen sind alle 1000 Strecken (s_1000_10.dat) und jeder gefundene Schnittpunkt als Punkt. Durch Vergrößern bestimmter Bereiche können die Resultate einfach und sicher überprüft werden. So kann in kurzer Zeit grafisch überprüft werden ob alle gefundenen Schnittpunkte tatsächlich Schnittpunkte sind und ob es Schnittpunkte gibt die noch nicht als solche erkannt werden.

