

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

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 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.

Neue Datei für P3:

- s_1000_10.dat:
Schnittpunkte: 796, Laufzeit: 98.032 ms

Dateien aus P1 (reduziert nach Kriterien aus Vorlesung):

- 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: 77111, Laufzeit: 556070 ms

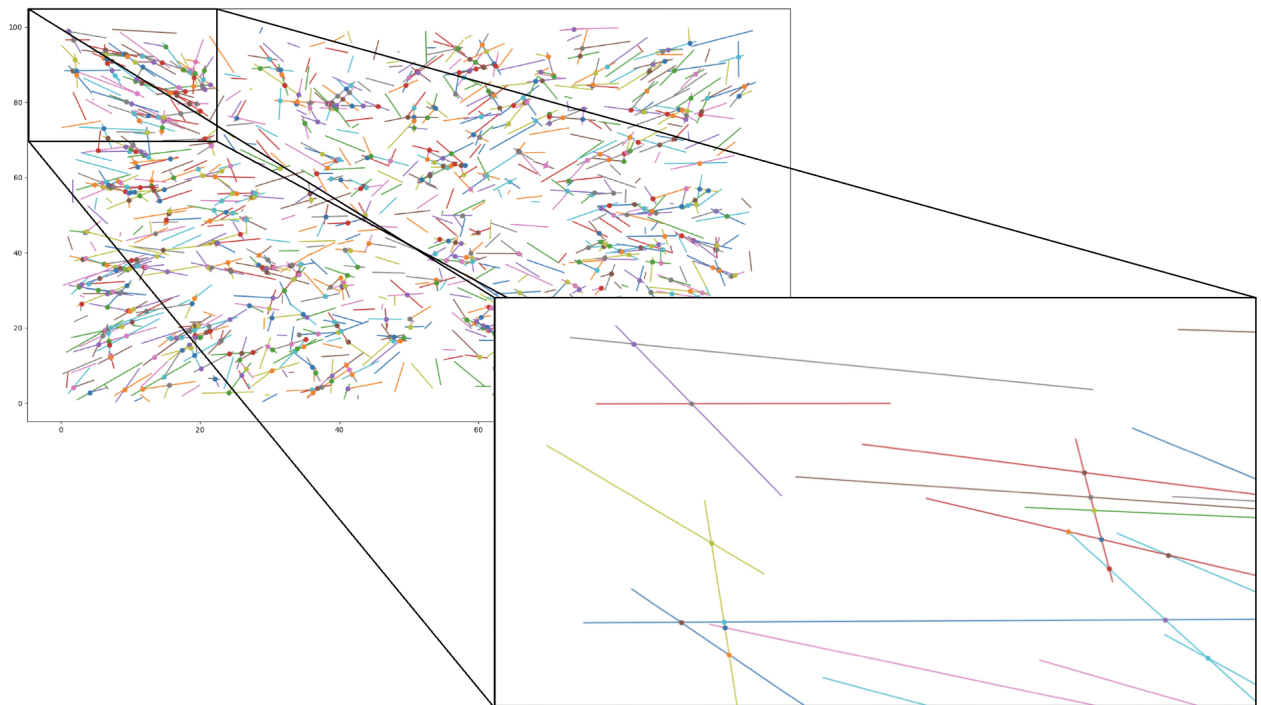
Es konnte bei allen Dateien eine längere Laufzeit festgestellt werden. Dies ist darauf zurückzuführen, dass die verwendete Datenstruktur (std::vector) sortieren, einfügen, finden und entfernen von Elementen mit linearer Komplexität ausführt. Diese Operationen werden mehrfach pro Durchlauf ausgeführt und sind damit sehr zeitaufwändig.

Im Weiteren wurde jedoch festgestellt, dass die Laufzeit des Programmes abhängig von der Anzahl der gefundenen Schnittpunkte ist. Dies zeigt sich durch den Vergleich der Laufzeiten der Dateien s_1000_10.dat und s_1000_1.dat. Diese haben dieselbe Anzahl an Linien aber einen signifikanten Unterschied in der Anzahl der Schnittpunkte und damit der Laufzeit.

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.



Nachdem das Programm aus dem ersten Praktikum bereits validiert wurde, wird dieses als Referenz für den Sweep Line Algorithmus verwendet. Hieraus hat sich ergeben, dass die Anzahl der Schnittpunkte für beide Programme den selben Wert hat (sowohl für die s_1000_10.dat als auch für die reduzierten, ursprünglichen Dateien).

Im Weiteren wurden die Ergebnisse mit denen anderer Gruppen verglichen. Diese sind zu gleichen Ergebnisse gekommen.