

Vorlesung Algorithmische Kartografie – Projektbeschreibung

Problembeschreibung. Gegeben sei eine Karte, die Regionen und darin liegende Städte abbildet. Entwickeln Sie einen Algorithmus, der die Regionen so vereinfacht, dass sowohl die topologische Beziehung zwischen Regionsgrenzen als auch zwischen Regionsgrenzen und Städten durch die Vereinfachung nicht verändert wird. Nehmen Sie hierzu an, dass die Regionsgrenzen als Kantenzüge und die Städte als Punkte in der Ebene gegeben sind. Für einen gegebenen Parameter $k \in \mathbb{N}$ sollen die vereinfachten Regionen maximal k Kanten besitzen. Es steht Ihnen frei, die in der Vorlesung vorgestellten Algorithmen an die Problemstellung anzupassen, oder eigene Ansätze zu verwenden. Implementieren Sie Ihren Algorithmus in einer Programmiersprache Ihrer Wahl.

Ziel. Entwicklung und Implementierung eines Algorithmus zur Vereinfachung von Regionen, sodass die Vereinfachung topologieerhaltend ist und aus maximal k Kanten besteht.

Annahmen zu den Daten. Sie dürfen folgende Annahmen zu den gegebenen Kantenzügen und Punkten machen.

1. Alle Daten liegen im kartesischen Raum.
2. Kantenzüge schneiden sich nicht selbst.
3. Kantenzüge schneiden keine anderen Kantenzüge (außer an den Endpunkten).
4. Die gegebenen Punkte liegen nicht auf den gegebenen Kantenzügen.
5. Die Anzahl gegebener Kantenzüge ist von der Eingabe abhängig.
6. Die Anzahl gegebener Punkte ist von der Eingabe abhängig.

Bedingungen an die vereinfachten Regionen.

1. Die topologische Beziehung der Kantenzüge untereinander darf nicht verändert werden (Erhalt der kombinatorischen Einbettung).
2. Die topologische Beziehung der Kantenzüge und der gegebenen Punkten darf nicht verändert werden.
3. Die Endpunkte der Kantenzüge dürfen nicht verändert werden. Alle anderen Eckpunkte der Kantenzüge dürfen verändert werden, z.B. dürfen Eckpunkte gelöscht oder verschoben werden. Es dürfen auch neue Eckpunkte eingeführt werden.
4. Die Kantenzüge besitzen zusammen maximal k Kanten, wobei $k \in \mathbb{N}$ gegeben ist.

Programmiersprache. Sie dürfen eine Programmiersprache Ihrer Wahl verwenden. Sollten Sie nicht *Java* oder *Python* verwenden, so sprechen Sie dies bitte mit Ihrem Betreuer ab.

Format der Eingabedaten. Die Eingabedaten werden im GML-Format (Version 2.1.1.) zur Verfügung gestellt. Die Daten bestehen aus zwei Dateien: `lines_out.txt` und `points_out.txt`.

- `lines_out.txt`: Enthält zeilenweise die Kantenzüge der Regionen. Eine Zeile ist wie folgt aufgebaut. Zuerst kommt eine Identifikationsnummer eines Kantenzugs, gefolgt von einem GML-Eintrag für den Kantenzug.
- `points_out.txt`: Enthält zeilenweise die Koordinaten der Städte. Eine Zeile ist wie folgt aufgebaut. Zuerst kommt eine Identifikationsnummer der Stadt, gefolgt von einem GML-Eintrag für den entsprechenden Punkt.

Beispiele für Eingaben finden Sie auf der Vorlesungshomepage.

Format der Ausgabedaten. Die vereinfachten Regionen sollen dasselbe Format wie die gegebenen Regionen besitzen. Insbesondere sollen die Identifikationsnummern der Kantenzüge mit der Eingabe übereinstimmen, d.h., ein Kantenzug der Ausgabe besitzt dieselbe Identifikationsnummer wie der entsprechende Kantenzug der Eingabe. Sollte der Algorithmus nicht in der Lage sein, die Regionen so zu vereinfachen, dass sie aus k Kanten bestehen, dann soll dies entsprechend ausgegeben werden.

Abgabe. Es soll eine einzelne `.zip` Datei, die den Quellcode und alle nötigen Bibliotheken enthält, abgegeben werden. Wenn möglich soll eine ausführbare Datei mitgeliefert werden. Das Programm soll über die Kommandozeile ausführbar sein und folgende Parameter in der angegebenen Reihenfolge erwarten:

1. `MaxEdgesToKeep`: Maximale Anzahl an Kanten, die verbleiben dürfen. Vereinfachte Regionen dürfen auch weniger Kanten besitzen.
2. `LineInputFilePath`: Pfad der Datei, welche die Kantenzüge enthält.
3. `PointInputFilePath`: Pfad der Datei, welche die Punkte enthält.
4. `OutputFilePath`: Pfad der Datei, in der das Ergebnis gespeichert wird.

Fristen. Folgende Fristen müssen eingehalten werden.

1. Bis 13. Mai Treffen mit Betreuer. In diesem Treffen soll der verwendete Algorithmus besprochen werden. Den Termin bitte rechtzeitig mit dem Betreuer vereinbaren.
2. Bis 21. Juni Abgabe der Projektdateien.

Präsentation. Nach Ende der regulären Vorlesungstermine (voraussichtlich Anfang Juli), wird es einen Termin zur Präsentation der Projekte geben. Jedes Team soll an diesem Termin sein Projekt in einem 10 minütigen Vortrag vorstellen. Der Vortrag soll unter anderem folgenden Inhalt haben:

1. Vorstellung des verwendeten Algorithmus.
2. Präsentation der Ergebnisse.
3. Vor- und Nachteile der verwendeten Methode.
4. Eigene Bewertung der Ergebnisse in Hinsicht auf Qualität.