

# Softwareprojekt: Rekonstruktion metrischer Graphen

Terese Haimberger, Lea Helmers, Jiang Hongliang, Mahmoud Kassem, Daniel Theus, Moritz Walter

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung: Rekonstruktion metrischer Graphen</b>	<b>2</b>
<b>2</b>	<b>Organisation und Umsetzung des Algorithmus in Java</b>	<b>2</b>
<b>3</b>	<b>Arbeit in den einzelnen Gruppen</b>	<b>3</b>
3.1	Preprocessing . . . . .	3
3.2	Reconstruction . . . . .	3
3.3	Visualisierung . . . . .	3
<b>4</b>	<b>Testen?</b>	<b>3</b>
<b>5</b>	<b>Schwierigkeiten und Verbesserungsvorschläge</b>	<b>3</b>
<b>6</b>	<b>Zusammenfassung</b>	<b>3</b>

# 1 Einleitung: Rekonstruktion metrischer Graphen

## Motivation und Problemstellung

Im Rahmen des Softwareprojekts „Anwendungen effizienter Algorithmen“ haben wir uns damit befasst, einen Algorithmus umzusetzen, der aus einer Punktmenge den zugrundeliegenden Graphen sowie dessen Metrik rekonstruiert. Dadurch soll Struktur in große Mengen geometrischer Daten gebracht werden, was deren Analyse und Weiterverarbeitung erleichtert. Zu verarbeitende Daten können Netzwerke im weitesten Sinne sein, wie beispielsweise GPS-Daten, Strom- und Nachrichtennetze oder astrologische Daten. Häufig sind diese Daten gestört oder enthalten Ausreißer und sind zudem im Allgemeinen sehr umfangreich. Ziel des Algorithmus ist es, die Datenmenge kompakt durch ihre wichtigsten Verzweigungen darzustellen, wodurch sie auf ihre wichtigen Aspekte reduziert wird. So kann eine einfachere Visualisierung und die weitere Analyse und Verarbeitung der Daten ermöglicht werden.

Als Grundlage für unsere Arbeit diente uns ein Paper [1], welches einen Algorithmus für die Rekonstruktion metrischer Graphen beschreibt und dessen Richtigkeit beweist. Die Eingabe besteht dabei aus einem metrischen Raum  $(Y, d_y)$ , der aus den Rohdaten konstruiert wird und dem ein metrischer Graph  $(X, d_x)$  zugrundeliegt, den es zu rekonstruieren gilt. Ziel des Algorithmus ist es, diesen zugrundeliegenden Graphen durch einen metrischen Graphen  $(\hat{X}, d_{\hat{x}})$  anzunähern und dabei weitestgehend die Abstände von  $(X, d_x)$  in  $(\hat{X}, d_{\hat{x}})$  beizubehalten.

## Der Algorithmus

## 2 Organisation und Umsetzung des Algorithmus in Java

Da allen Studenten im Team die objektorientierte Programmiersprache Java geläufig war und sie uns durchaus als geeignet für die Aufgabe schien, haben wir uns entschlossen, unser Programm darin zu schreiben. Für das Arbeiten an unserer Software haben wir uns in drei verschiedene Gruppen aufgeteilt. Die eine Gruppe hat sich mit der Vorverarbeitung der Rohdaten beschäftigt, im Wesentlichen also aus den Rohdaten den metrischen Raum  $(Y, d_y)$  für die Eingabe des Algorithmus konstruiert. Die zweite Gruppe hat sich mit der Implementierung des Algorithmus, genauer der Rekonstruktion von  $(\hat{X}, d_{\hat{x}})$ , in Java auseinandergesetzt, während die dritte für die Visualisierung des Graphen  $(\hat{X}, d_{\hat{x}})$  zuständig war. Damit wir stets alle über die Arbeit unserer Teammitglieder informiert waren haben wir unsere Ergebnisse bei `github.com` gespeichert und stets aktualisiert. (GENAUER ERKLÄREN?) Um Aufgaben und den Stand der Arbeit daran für alle zugänglich zu dokumentieren haben wir zusätzlich `trello.com` genutzt, wo man ein virtuelles Notizbrett erstellen kann. (GENAUER ERKLÄREN!).

Einschränkung auf GPS und Buchstaben

### **3 Arbeit in den einzelnen Gruppen**

#### **3.1 Preprocessing**

#### **3.2 Reconstruction**

#### **3.3 Visualisierung**

### **4 Testen?**

### **5 Schwierigkeiten und Verbesserungsvorschläge**

### **6 Zusammenfassung**

### **Literatur**

- [1] Mridul Aanjaneya u. a. “Metric graph reconstruction from noisy data”. In: *International Journal of Computational Geometry & Applications* 22.4 (2012), S. 305–325.