

Institut für Formale Methoden der Informatik
Abteilung Algorithmik
Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 296

Contraction Hierarchies für kontinuierliche Graphsimplifizierung

Seyed Pedram Hosseyni Damabi

Studiengang:	Informatik
Prüfer:	Prof. Dr. Stefan Funke
Betreuer:	Prof. Dr. Stefan Funke
begonnen am:	17.11.2015
beendet am:	12.05.2016
CR-Klassifikation:	G.2.2, I.3.5, J.2

Kurzfassung

Mithilfe von Contraction Hierarchies lassen sich kürzeste Pfade eines Graphen effizient finden. Hierbei findet vor der eigentlichen Suche des kürzesten Pfades eine Vorverarbeitung statt, deren zentrale Operation die Kontraktion von Knoten ist. Hierbei wird der entsprechende Knoten zunächst aus dem Graphen entfernt; falls dadurch die kürzeste-Wege-Distanz der benachbarten Knoten vergrößert wird, werden dem Graphen neue Kanten, sogenannte Shortcuts, hinzugefügt.

Durch die iterative Entfernung der Knoten kann jedoch auch eine Simplifizierung des Originalgraphen durchgeführt werden. Hierzu werden in dieser Arbeit verschiedene Kontraktionsreihenfolgen untersucht. Die betrachteten Kriterien zur Sortierung der Knoten sind die Edge Difference, die räumliche Dichte der Knoten, die durchschnittliche Kantendistanz der zum Knoten inzidenten Kanten sowie (bei Knoten vom Grad zwei) der Winkel der zum Knoten inzidenten Kanten. Hierzu wird ein Verfahren erläutert, mit dem die unterschiedlichen Kriterien miteinander kombiniert werden können.

Des Weiteren wird ein Verfahren vorgestellt, mit dem aus der Ausgabe der Vorverarbeitung ein simplifizierter Graph erzeugt werden kann. Dabei werden Knoten abhängig vom Zeitpunkt ihrer Kontraktion aus dem Graphen entfernt. Von den verbleibenden Kanten werden zu lange Shortcuts rekursiv entpackt, d.h. durch die ursprünglichen Kanten ersetzt.

Inhaltsverzeichnis

1	Einführung	7
2	Grundlagen	8
2.1	Allgemeines über Graphen	8
2.2	Contraction Hierarchies	8
2.2.1	Überblick	8
2.2.2	Kontraktion von Knoten	8
2.2.3	Vorverarbeitung	8
2.2.4	Suchanfrage	9
2.3	Methoden zur Darstellung von Online-Karten	10
2.4	Generalisierung von Karten	11
3	Simplifizierung mithilfe von Contraction Hierarchies	12
3.1	Vorverarbeitungsphase: Kontraktionsreihenfolge	12
3.1.1	Kriterien zur Sortierung der Knoten	12
3.1.2	Wahl einer geeigneten Knotenmenge	12
3.2	Darstellung des Graphen	14
3.2.1	Untere Grenze für Knotenlevel	14
3.2.2	Obere Grenze für Kantenlänge	15
4	Experimente	17
4.1	Allgemeines	17
4.1.1	Verwendeter Graph	17
4.1.2	Notation bei Kombination von Kriterien	17
4.1.3	Verwendete Software	17
4.2	Edge Difference	17
4.2.1	Verschiedene minLevel-Werte	17
4.2.2	Zusätzliche Erhöhung von maxLength	20
4.2.3	Veränderung des Zoom-Levels	23
4.3	Knotendichte	25
4.3.1	Allgemeines	25
4.3.2	Vergleich mit Edge Difference	27
4.3.3	Kombination von Knotendichte und Edge Difference	29
4.4	Durchschnittliche Kantendistanz	33
4.4.1	Allgemeines	33
4.4.2	Vergleich mit (2,1)-Kombination	36
4.4.3	Kombination	37
4.5	Winkel	41
4.6	Laufzeiten	44
5	Zusammenfassung	47

1 Einführung

Um die Übersichtlichkeit von Landkarten zu gewährleisten, wird bei ihrer Erstellung häufig eine sogenannte *Generalisierung* durchgeführt, beispielsweise werden Details, die für eine Vergrößerungsstufe nicht relevant sind, aus der Karte entfernt. Üblicherweise verwenden Online-Kartendienste wie OpenStreetMap hierzu explizite Regeln, mit denen festgelegt wird, welche Details die jeweiligen Vergrößerungsstufen enthalten sollen. Mithilfe dieser Regeln werden für jede Vergrößerungsstufe der Karte vorgefertigte Graphiken erzeugt.

Eine spezielle Methode der Generalisierung ist die *Simplifizierung*. Hierbei werden aus dem Graphen einer Karte Knoten entfernt, wobei die charakteristische Form der Karte dennoch erhalten bleiben soll.

Durch die Simplifizierung des Graphen entstehen neben der visuellen Vereinfachung auch andere Vorteile: Da der resultierende Graph weniger Knoten besitzt, ist sein Speicheraufwand sowie die Zeit, die zum Rendern der Karte benötigt wird, geringer als bei Verwendung des Originalgraphen.

Für die Simplifizierung von Karten wird in dieser Arbeit die Verwendung von *Contraction Hierarchies* untersucht. Bei Contraction Hierarchies handelt es sich um ein effizientes Routing-Verfahren, welches aus zwei Phasen besteht: Einer Vorverarbeitungsphase und einer Suchphase. Die zentrale Operation der Vorverarbeitungsphase ist die Kontraktion von Knoten. Hierbei werden die Knoten iterativ entfernt und ggf. neue Kanten hinzugefügt.

Noch wenig beachtet ist jedoch, dass die Vorverarbeitungsphase eine schrittweise Vereinfachung des Graphen darstellt, die zur Simplifizierung des Graphen verwendet werden kann.

Ziel dieser Arbeit ist, verschiedene Kontraktionsreihenfolgen zu untersuchen und ein geeignetes Verfahren zu entwickeln, um aus der Ausgabe der Vorverarbeitungsphase des Contraction Hierarchies Algorithmus einen simplifizierten Graphen bezüglich des Originalgraphen zu erzeugen.

2 Grundlagen

2.1 Allgemeines über Graphen

Definition: gerichteter Graph:

Ein *gerichteter Graph* $G = (V, E)$ besteht aus einer Knotenmenge V und einer Kantenmenge $E \subseteq V \times V$ [1].

Definition: Unabhängige Menge eines Graphen:

Sei $G = (V, E)$ ein (gerichteter) Graph. Die *unabhängige Menge* [1] der Knotenmenge von G ist eine Teilmenge der Knoten $V' \subset V$ so dass,

$$\forall v_1, v_2 \in V' : (v_1, v_2) \notin E$$

Geometrischer Graph: Ein Graph G wird als *geometrischer Graph* bezeichnet, wenn die Menge der Knoten von G Punkten im euklidischen Raum entsprechen [1]. Dies ist beispielsweise bei Graphen von Straßenkarten der Fall. Hierbei wird die Straßenkarte durch Punkte sowie Linien zwischen den Punkten dargestellt.

2.2 Contraction Hierarchies

2.2.1 Überblick

Mithilfe der von Geisberger et al. entwickelten Methode der *Contraction Hierarchies* [2] lassen sich kürzeste Pfade eines Graphen effizient finden.

Gegeben sei der gerichtete Graph $G = (V, E)$ mit Kantengewichtung $d : E \rightarrow \mathbb{R}$. Das Ziel ist die Bestimmung eines Pfades zwischen zwei Knoten, dessen Kosten bezüglich d minimal ist. Hierzu findet für G zuerst eine einmalige Vorverarbeitung statt, bei der dem Originalgraphen neue Kanten hinzugefügt werden. Die Suche des kürzesten Pfades findet mithilfe des neu erzeugten Graphen statt.

2.2.2 Kontraktion von Knoten

Die zentrale Operation der Vorverarbeitung ist die Kontraktion von Knoten, und wird im Folgenden erläutert:

Die Kontraktion des Knotens $v \in V$ findet statt, indem v zunächst aus der Knotenmenge entfernt wird. Falls dadurch die Länge eines kürzesten Pfades von $u \in V$ nach $w \in V$ mit $(u, v) \in E, (v, w) \in E$ vergrößert wird, wird dem Graph eine neue Kante (u, w) , die auch als *Shortcut* bezeichnet wird, mit Kantengewichtung $d(u, v) + d(v, w)$ hinzugefügt.

2.2.3 Vorverarbeitung

Bei der Vorverarbeitung wird jedem Knoten eine natürliche Zahl zugewiesen, die im Folgenden als *Level* des Knotens bezeichnet wird. Dies legt die Reihenfolge fest, in der

die Knoten kontrahiert werden. Die Knoten werden entsprechend dieser Reihenfolge wie im vorherigen Abschnitt erläutert kontrahiert.

Durch die Vorverarbeitung entsteht aus dem Originalgraphen $G = (V, E)$ ein Graph $G' = (V, E')$. Dieser enthält alle Kanten des Originalgraphen sowie alle durch Knotenkontraktion entstandenen Shortcuts.

Die Reihenfolge, in der die Knoten kontrahiert werden, kann zu Unterschieden in der Anzahl der Shortcuts, aber auch in der Dauer einer Suchanfrage führen. Falls beispielsweise die Knoten einer linearen Kette von Knoten nacheinander kontrahiert werden, kann die Suche eines kürzesten Pfades auf dieser Kette lineare Laufzeit haben. Falls jedoch in jedem Schritt die maximale unabhängige Menge der Knoten kontrahiert wird, benötigt die Suche logarithmische Laufzeit [2]. Hierbei sei angemerkt, dass die Kontraktion von mehreren Knoten in einem Schritt möglich ist, sofern diese eine unabhängige Menge bilden.

Die Bestimmung der Reihenfolge, in der die Knoten kontrahiert werden, erfolgt anhand von Kriterien. Ein häufig verwendetes Kriterium ist die *Edge Difference* [2]. Die Edge Difference eines Knotens v ist die Anzahl der Shortcuts, die durch Kontraktion von v entstehen abzüglich der zu v inzidenten Kanten.

Man beachte, dass die Edge Difference sowie weitere Eigenschaften der Knoten wie bspw. der Grad der Knoten durch Kontraktion anderer Knoten verändert werden kann. Das Kriterium, mithilfe dessen die Reihenfolge der Knoten festgelegt wird, sollte also im Laufe der Vorverarbeitung für die noch verbleibenden Knoten aktualisiert werden. Eine Strategie ist die Aktualisierung der Werte der Kriterien aller Knoten vor jedem Kontraktionsschritt; für andere Methoden wird auf [2] verwiesen.

2.2.4 Suchanfrage

Um einen kürzesten Pfad von einem Knoten $s \in V$ nach $t \in V$ zu finden, wird eine bidirektionale Dijkstra Suche durchgeführt. Die erste Suche beginnt bei s und betrachtet nur Kanten (v, w) mit $level(v) < level(w)$. Bei der zweiten Suche handelt es sich um eine Rückwärtssuche, die bei t beginnt und nur Kanten (v, w) betrachtet, für die $level(v) > level(w)$ gilt.

Falls ein kürzester Pfad von s nach t existiert, dann existiert ein Knoten $v \in V$, der bei beiden Suchanfragen gefunden wird und das höchste Level der in beiden Suchen gefundenen Knoten besitzt. Der kürzeste Pfad von s nach t ist der kürzeste Pfad von s nach v und anschließend von v nach t .

Die in dem resultierenden Pfad enthaltenen Shortcuts müssen anschließend *entpackt* werden, d.h. durch die ursprünglichen Kanten ersetzt werden.

2.3 Methoden zur Darstellung von Online-Karten

In diesem Abschnitt wird ein Überblick über Raster Kacheln und Vektor Kacheln, sowie ihre Vor- und Nachteile gegeben (vgl. [3] [4] [5]).

Raster Kacheln:

Eine weit verbreitete Methode zur Darstellung von Karten ist die Verwendung von *Raster Kacheln* (engl. *raster tiles*). Hierbei wird die Karte für jede Zoom-Stufe in rechteckige Gebiete aufgeteilt, aus denen anschließend Rastergraphiken erzeugt werden. Diese werden für jeden Kartenausschnitt vorgefertigt und auf einem Server des Kartendienstes gespeichert.

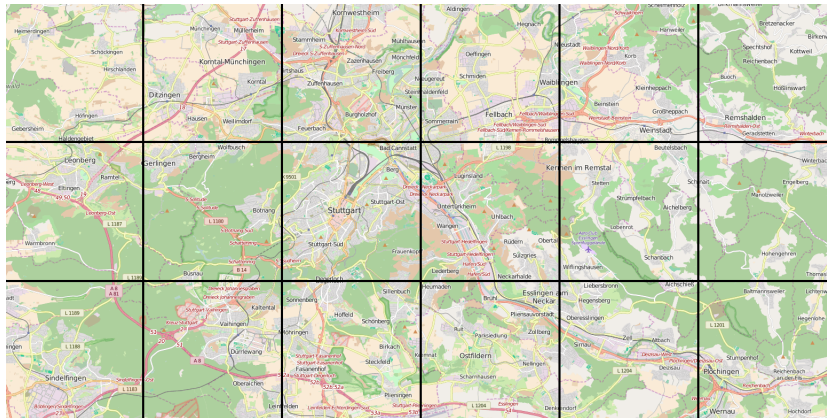


Abbildung 1: Original Kacheln der Region Stuttgart in OpenStreetMap.

Quelle: www.openstreetmap.org

Dieses Vorgehen hat mehrere Vorteile: Die Aufteilung der Karte in kleinere Gebiete ermöglicht es, für jeden Nutzer nur die relevanten Kartenausschnitte zu übertragen. Führt der Nutzer eine Vergrößerung des Kartenausschnittes durch oder verschiebt er den Kartenausschnitt, werden erneut nur die relevanten Kacheln übertragen. Außerdem müssen die benötigten Rastergraphiken nicht für jeden Nutzer erneut generiert werden, weshalb die angeforderten Kacheln relativ schnell bereitgestellt werden können.

Allerdings besitzt die Verwendung von Raster Kacheln auch Nachteile: Die erzeugten Raster Kacheln sind statisch, sie können nach ihrer Erzeugung nicht modifiziert werden. D.h. dass Änderungen der Kartendaten, wie bspw. neu eingefügte Straßen, nicht direkt nach der Modifizierung sichtbar sind, sondern erst nachdem die Kacheln neu erzeugt wurden.

Für die Erzeugung der Kacheln existieren verschiedene Programme. Für OpenStreetMap wird häufig *Mapnik* verwendet, bei dem das visuelle Aussehen der Karte mithilfe von Regeln festgelegt wird [4] [6].

Vektor Kacheln

Im Gegensatz zu Rastergraphiken bestehen Vektorgraphiken aus der Beschreibung der

einzelnen Graphikelemente. Um beispielsweise einen Kreis darzustellen, genügen ein paar charakteristische Eigenschaften, wie z.B. der Mittelpunkt und der Radius des Kreises.

Die Verwendung von Vektor Kacheln hat im Vergleich zu Raster Kacheln mehrere Vorteile: Da das Rendern beim Nutzer stattfindet, entstehen neue Möglichkeiten der Interaktion. Eine visuelle Veränderung des Kartenausschnittes, z.B. die Veränderung des Kartenstils oder das Hervorheben von bestimmten Kartenelementen ist ohne erneute Übertragung von Kacheln möglich.

Ein weiterer Vorteil ist, dass Veränderungen der Kartendaten schneller für den Nutzer sichtbar sind.

Ein Nachteil gegenüber Raster Kacheln ist, dass die Vektor Kacheln beim Nutzer gerendert werden müssen, weshalb der Nutzer spezielle Anwendungen benötigt.

2.4 Generalisierung von Karten

Um die Übersichtlichkeit von Landkarten sicherzustellen, wird abhängig vom Maßstab der Darstellung eine Vereinfachung durchgeführt. Diese Vereinfachung der Karte wird auch als *Generalisierung* [7] bezeichnet und beinhaltet u.A. die Entfernung von Details, aber auch die Vergrößerung und Vereinfachung von Objekten.

Einige konkrete Gründe für eine Generalisierung sind, dass durch Veränderung des Maßstabs (durch Veränderung der Zoom-Stufe der Darstellung) eine zu hohe Dichte von sichtbaren Objekten vorliegt, oder dass verschiedene Objekte zu einem einzelnen Objekt zusammenfallen, da der Abstand zwischen ihnen nicht mehr dargestellt werden kann [8].

Zu den Methoden der Generalisierung gehören u.A. die Glättung von Linien, die Aggregation von Elementen, aber auch die Simplifizierung der Karte.

Bei der *Simplifizierung* der Karte ist das Ziel, die Anzahl der Knoten des Graphen der Karte zu verringern. Der resultierende Graph sollte die entsprechenden Objekte dennoch präzise genug darstellen, und der charakteristische Verlauf von Linien soll dabei beibehalten werden (s. Abb. 2). Zu den weiteren Vorteilen, die durch die Simplifizierung entstehen, gehört ein geringerer Speicheraufwand sowie ein schnelleres Rendern der Karte.

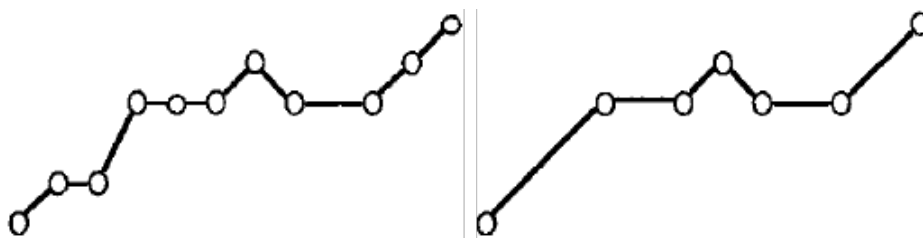


Abbildung 2: links: ursprüngliche Linie; rechts: nach Durchführung der Simplifizierung;
Quelle: [8]

Für die Beschreibung weiterer Methoden der Generalisierung wird auf [8] verwiesen.

3 Simplifizierung mithilfe von Contraction Hierarchies

3.1 Vorverarbeitungsphase: Kontraktionsreihenfolge

Im Folgenden wird ein Schema vorgestellt, mithilfe dessen die Kontraktionsreihenfolge der Vorverarbeitung festgelegt werden kann. Hierzu werden zunächst einige Kriterien erläutert, mit denen eine Sortierung der Knoten durchgeführt werden kann, sowie eine Methode, um verschiedene Kriterien miteinander zu kombinieren.

3.1.1 Kriterien zur Sortierung der Knoten

Edge Difference: Die Edge Difference eines Knotens ist die Anzahl der Shortcuts, die durch Kontraktion des Knotens entstehen minus der Anzahl der zum Knoten inzidenten Kanten [2].

Knotendichte: Bei diesem Kriterium wird ein Gitter über den Graphen gelegt und anschließend gezählt, wie viele Knoten sich in den jeweiligen Gitterzellen befinden. Der Wert eines Knotens ist die Anzahl der Knoten, die sich in der entsprechenden Gitterzelle befinden.

Durchschnittliche Kantendistanz: Bei diesem Kriterium ist der Wert eines Knotens die durchschnittliche Länge der zu ihm inzidenten Kanten. (Anmerkung: Als Länge einer Kante wurde der euklidische Abstand zwischen dem Start- und Endknoten der Kante gewählt.)

Winkel: Bei Knoten vom Grad zwei kann der Winkel der zum Knoten inzidenten Kanten in die Wahl der Kontraktionsreihenfolge einbezogen werden.

3.1.2 Wahl einer geeigneten Knotenmenge

In diesem Abschnitt wird ein Verfahren vorgestellt, mithilfe dessen verschiedene Kriterien zur Sortierung der Knoten miteinander kombiniert werden können, mit dem Ziel, in jedem Kontraktionsschritt eine geeignete Menge von Knoten zu kontrahieren.

Vorsortierung: Vor der Bestimmung der im nächsten Schritt zu kontrahierenden Knotenmenge wird für jeden noch verbleibenden Knoten v der Wert $in_deg(v) \cdot out_deg(v)$ berechnet, wobei $in_deg(v)$ und $out_deg(v)$ der Eingangs- bzw. Ausgangsgrad des Knotens v ist. Die Liste der Knoten wird anschließend nach aufsteigendem Wert sortiert, d.h. Knoten, bei denen der berechnete Wert gering ist befinden sich am Anfang der Liste. Der berechnete Wert ist die maximale Anzahl an Shortcuts, die durch Kontraktion des entsprechenden Knotens erzeugt werden können.

Berechnung der unabhängigen Menge: Im nächsten Schritt wird die unabhängige Menge der verbleibenden Knoten berechnet. Dies kann folgendermaßen gemacht werden:

Im ersten Schritt besteht die unabhängige Menge aus dem ersten Knoten der aktuellen Knotenliste. Danach wird jeder Knoten, durch dessen Hinzufügen die Menge unabhängig bleibt, hinzugefügt. Dies geschieht so lange, bis alle Knoten betrachtet wurden.

Priorität der Knoten festlegen: Prinzipiell kann jede Teilmenge der unabhängigen Menge kontrahiert werden. Um jedoch eine geeignete Teilmenge zu finden, wird für jeden Knoten eine Priorität bestimmt. Knoten mit einer guten Priorität werden im nächsten Schritt kontrahiert, während Knoten mit schlechteren Prioritäten in späteren Schritten kontrahiert werden.

Die Priorität kann durch ein in Abschnitt 3.1.1 beschriebenes Kriterium berechnet werden. Anschließend werden nur die Knoten, deren Priorität kleiner oder gleich dem Median der Prioritäten ist zur Menge der zu kontrahierenden Knoten hinzugefügt.

Kombination mehrerer Kriterien: Die Kombination mehrerer Kriterien kann durch lineare Kombination verschiedener Prioritäten erfolgen. Grundsätzlich treten jedoch zwei Probleme auf:

Die Werte der einzelnen Knoten können sich durch vorherige Kontraktionen ändern. Beispielsweise besitzen die Werte der Knotendichte eines Knotens im Graph von Baden-Württemberg vor dem ersten Kontraktionsschritt einen Mittelwert von ca. 912. Gegen Ende der Vorverarbeitung fällt dieser Wert auf Werte im einstelligen Bereich.

Ein anderes Problem ist, dass die verschiedenen Kriterien sich um mehrere Größenordnungen unterscheiden können. Der Graph von Baden-Württemberg besitzt vor dem ersten Kontraktionsschritt einen Edge Difference Mittelwert von ca. $-1,89$. Bei der Addition der Edge Difference Werte mit den Werten der Knotendichte haben die Edge Difference Werte also einen sehr geringen Einfluss.

Um diese Werte miteinander zu kombinieren, werden nach der Bestimmung der unabhängigen Menge für jedes Kriterium k die Werte der Knoten bestimmt und anschließend auf den Bereich $[0, 1]$ normiert. Im Anschluss daran wird für jedes Kriterium der arithmetische Mittelwert m bestimmt, und jeder normierte Wert des Kriteriums wird mit dem Faktor $|\frac{1}{m}|$ multipliziert, so dass die verschiedenen Kriterien alle den Mittelwert 1 besitzen. Dieser normierte und skalierte Wert eines Knotens v wird im Folgenden als $val_k(v)$ bezeichnet.

Um verschiedene Kriterien unterschiedlich zu gewichten, wird zu Beginn des Verfahrens für jedes Kriterium k ein Faktor c_k festgelegt. Die Priorität eines Knotens v ist

$$\sum_k c_k \cdot val_k(v)$$

Im Abschnitt 4 werden verschiedene Kriterien und verschiedene Koeffizienten experimentell untersucht.

3.2 Darstellung des Graphen

Durch die Vorverarbeitung des Originalgraphen findet zunächst das Gegenteil dessen statt, was erreicht werden soll: Statt einer Simplifizierung werden dem Graphen neue Kanten hinzugefügt.

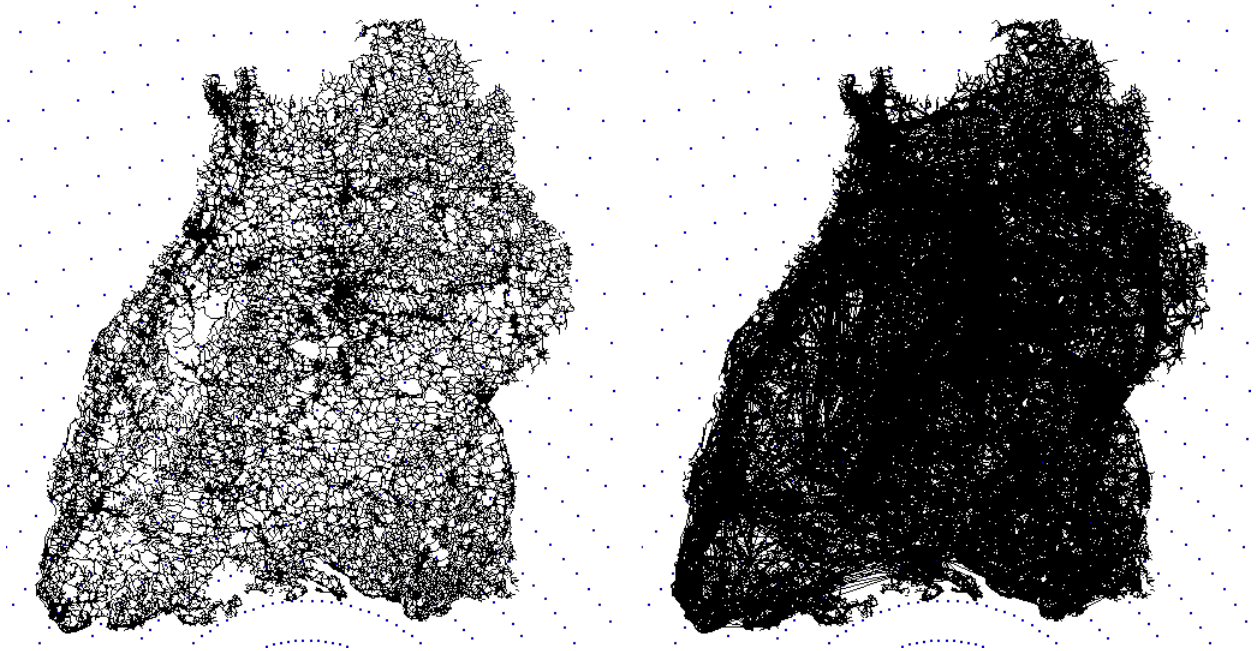


Abbildung 3: Graph von Baden-Württemberg
links: ohne zusätzliche Shortcuts
rechts: nach Durchführung der Vorverarbeitung

Im Folgenden wird gezeigt, wie aus der Ausgabe der Vorverarbeitung des Contraction Hierarchies Algorithmus ein simplifizierter Graph bezüglich des Originalgraphen erzeugt werden kann.

3.2.1 Untere Grenze für Knotenlevel

Sei $G = (V, E)$ der zu vereinfachende Originalgraph, $G' = (V, E')$ die um Shortcuts erweiterte Ausgabe der Vorverarbeitung und $l : V \rightarrow \mathbb{N}$ der in der Vorverarbeitungsphase festgelegte Level eines Knotens. (Der Level der Knoten definiert die Reihenfolge der Kontraktion, wobei im ersten Schritt alle Knoten mit Level 0 kontrahiert werden.)

Ein erster Schritt zur Vereinfachung des Graphen besteht darin, Knoten und ihre zugehörigen Kanten aus dem Graphen zu entfernen. Dies kann durch die Wahl einer unteren Grenze für die Knotenlevel erreicht werden, die im Folgenden *minLevel* genannt wird. Der resultierende Graph enthält also nur Knoten v , für die $l(v) \geq \text{minLevel}$ gilt.

Nachdem alle Knoten, deren Level zu klein ist, entfernt wurden, müssen alle zusätzlichen

Shortcuts entfernt werden. Dadurch bleiben alle Kanten übrig, deren Knoten direkt über der minLevel -Grenze ist. Dies wird in Abbildung 4 veranschaulicht.

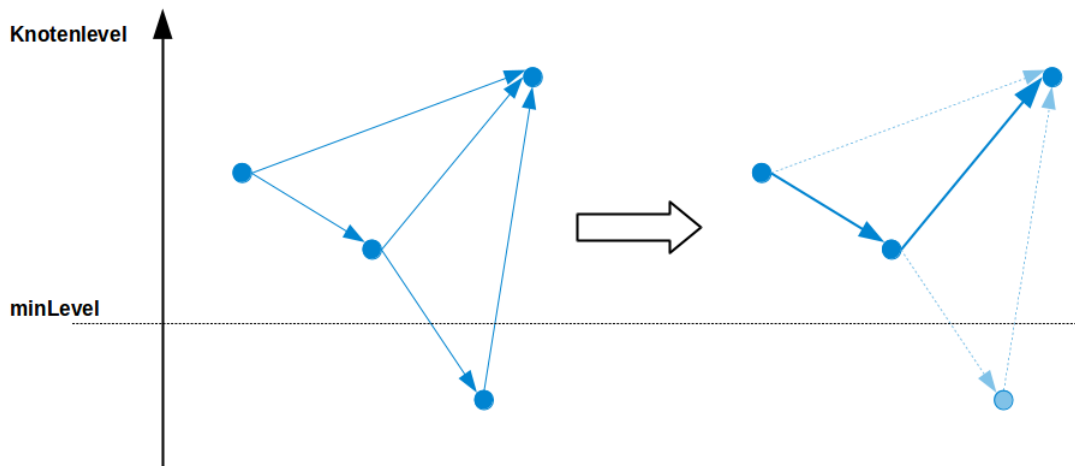


Abbildung 4: Wahl der Knoten und Kanten bei gegebenem minLevel

3.2.2 Obere Grenze für Kantenlänge

Nachdem alle Knoten, deren Level kleiner als minLevel ist, entfernt wurden, kann es sein, dass einige der verbleibenden Kanten zu lang sind. Die Ursache hierfür ist, dass die während der Vorverarbeitung entstehenden Shortcuts auch zwischen zwei Knoten entstehen können, die relativ weit voneinander entfernt sind. Die entstehenden Shortcuts führen zu einer sichtbaren Verschlechterung der Darstellung, wie im linken Teil der Abbildung 5 zu sehen ist. Bei einem zu hohen minLevel -Wert führt dies dazu, dass die Strukturen des ursprünglichen Graphen nicht mehr erkennbar sind.

Um dem entgegenzuwirken, wird eine Obergrenze für die Kantenlänge eingeführt, die im Folgenden als maxLength bezeichnet wird. Shortcuts, die länger als maxLength sind, werden so lange entpackt, bis sie maximal die Länge maxLength besitzen. Bei diesem Schritt werden die Level der Knoten nicht mehr beachtet, d.h. es kann vorkommen, dass dem Graphen wieder Knoten hinzugefügt werden, deren Level kleiner als minLevel ist.



Abbildung 5: Unterschiedliche Darstellungen in Abhängigkeit von `maxLength`:
links: ohne `maxLength`
mitte: moderate Wahl von `maxLength`
rechts: `maxLength` = 0

4 Experimente

4.1 Allgemeines

4.1.1 Verwendeter Graph

Als Ausgangsgraph für die weiteren Experimente wird der Graph von Baden Württemberg verwendet. Dieser enthält vor der Vorverarbeitung des Contraction Hierarchies Algorithmus 3540784 Knoten und 7181635 Kanten.

4.1.2 Notation bei Kombination von Kriterien

Bei Kombinationen von Kriterien werden die Gewichtungen als Tupel notiert. Bei einem 2-Tupel (a, b) gibt der erste Wert die Gewichtung des Edge-Difference-Kriteriums, und der zweite Wert die Gewichtung des Knotendichte-Kriteriums an. Bei einem 3-Tupel (a, b, c) gibt zusätzlich dazu der Wert c die Gewichtung des Durchschnittliche-Kantendistanz-Kriteriums an. Bei einem 4-Tupel (a, b, c, d) gibt zusätzlich dazu der vierte Wert die Gewichtung des Winkel-Kriteriums an.

4.1.3 Verwendete Software

Zur Durchführung der Vorverarbeitung wurde der *chconstructor* verwendet [9]. Hier wurde das in Abschnitt 3.1 vorgestellte Verfahren eingebaut. Die visuelle Darstellung des Graphen erfolgt mithilfe des *simplestGraphRendering* [10]. Hier wurde das in Abschnitt 3.2 vorgestellte Verfahren zur Wahl der entsprechenden Knoten und Kanten eingebaut.

4.2 Edge Difference

In diesem Abschnitt wird die Vereinfachung eines Graphen untersucht, bei dem die Kontraktionsreihenfolge nur durch Berücksichtigung der Edge Difference seiner Knoten festgelegt wurde. (Die Gewichtung des Edge-Difference-Kriteriums ist 1).

4.2.1 Verschiedene minLevel-Werte

Im Folgenden wird untersucht, welchen Einfluss verschiedene Werte von minLevel auf das Aussehen des Graphen, aber auch auf die Anzahl der Knoten und Kanten haben. Für diesen Abschnitt ist maxLength = 0.

Tabelle 1 zeigt die Anzahl der Knoten und Kanten der vereinfachten Graphen bei verschiedenen minLevel-Werten. Der Anteil der Knoten und Kanten bezieht sich dabei im Folgenden immer auf den Originalgraphen.

Wie in Tabelle 1 zu sehen ist, kann bereits durch einen geringen minLevel-Wert eine signifikante Reduktion der Knoten- und Kantenanzahl erreicht werden. Die sichtbaren Unterschiede zwischen dem Originalgraph und dem Graph mit $\text{minLevel} = 5$ sind zwar gering, jedoch enthält der resultierende Graph nur noch ca. 60% der ursprünglichen Knoten und Kanten. Auch bei $\text{minLevel} = 25$ bleibt die Gesamtstruktur des Graphen gut erhalten, obwohl der Graph nur noch ca. 15% der ursprünglichen Knoten und Kanten besitzt.

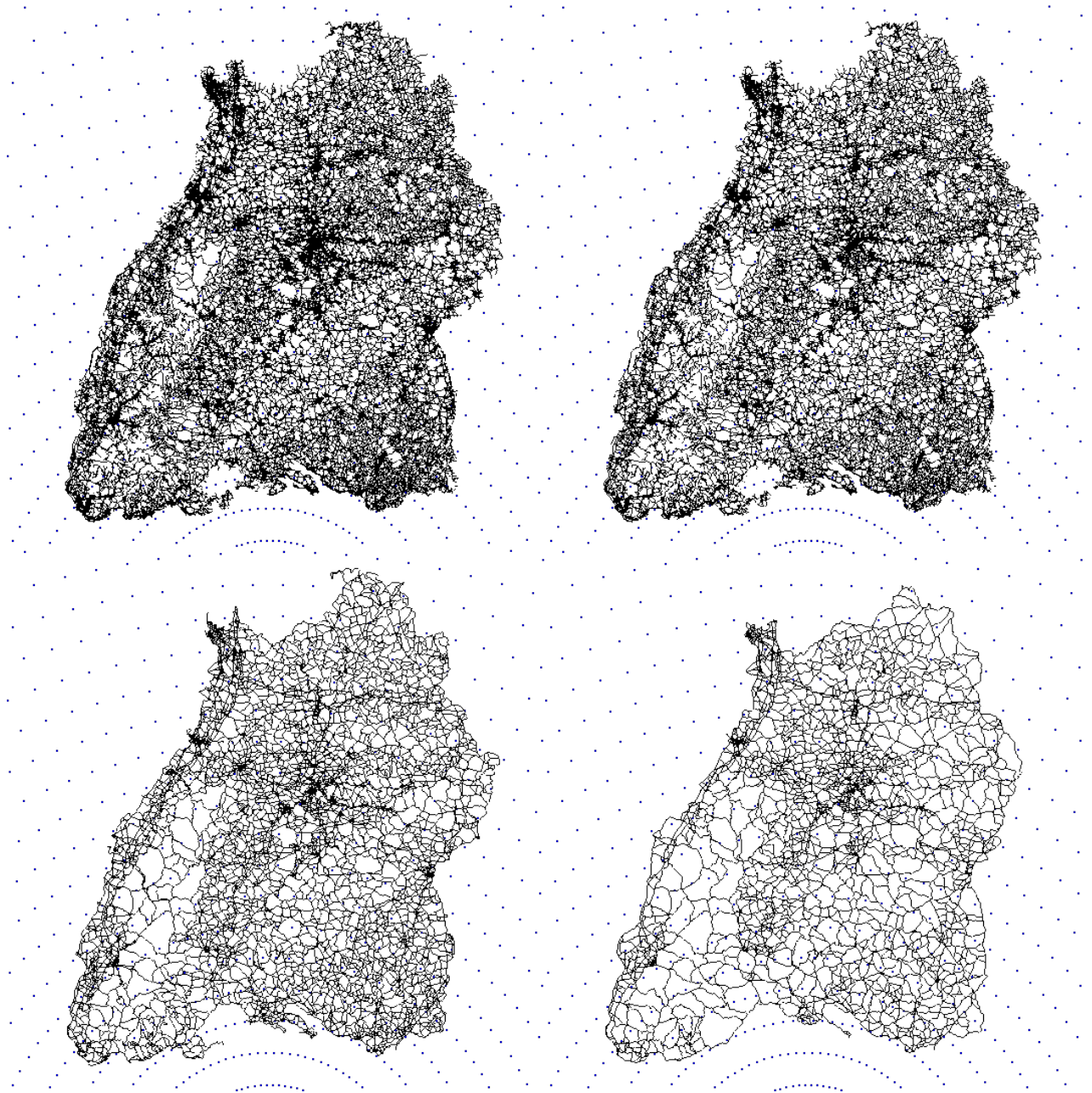


Abbildung 6: links oben: Originalgraph
links unten: $\text{minLevel} = 15$

rechts oben: $\text{minLevel} = 5$
rechts unten: $\text{minLevel} = 25$

minLevel	maxLength	Anzahl Knoten	Anzahl Kanten	Anteil(Knoten)	Anteil(Kanten)
5	0	2,20 Mio.	4,29 Mio.	62,06%	59,79%
15	0	0,88 Mio.	1,60 Mio.	24,75%	22,28%
25	0	0,55 Mio.	0,97 Mio.	15,64%	13,49%
35	0	0,35 Mio.	0,58 Mio.	9,86%	8,08%
45	0	0,24 Mio.	0,39 Mio.	6,89%	5,38%
55	0	0,17 Mio.	0,27 Mio.	4,93%	3,70%

Tabelle 1: Verschiedene Parameter und die Anzahl der Knoten und Kanten des resultierenden Graphen

Eine weitere Erhöhung von minLevel ist, wie in Abbildung 7 zu sehen ist, zwar möglich, würde den Graphen jedoch zu stark verändern. Bei einem zu hohem minLevel-Wert fallen außerdem viele Bereiche vom Rand des Graphen weg.

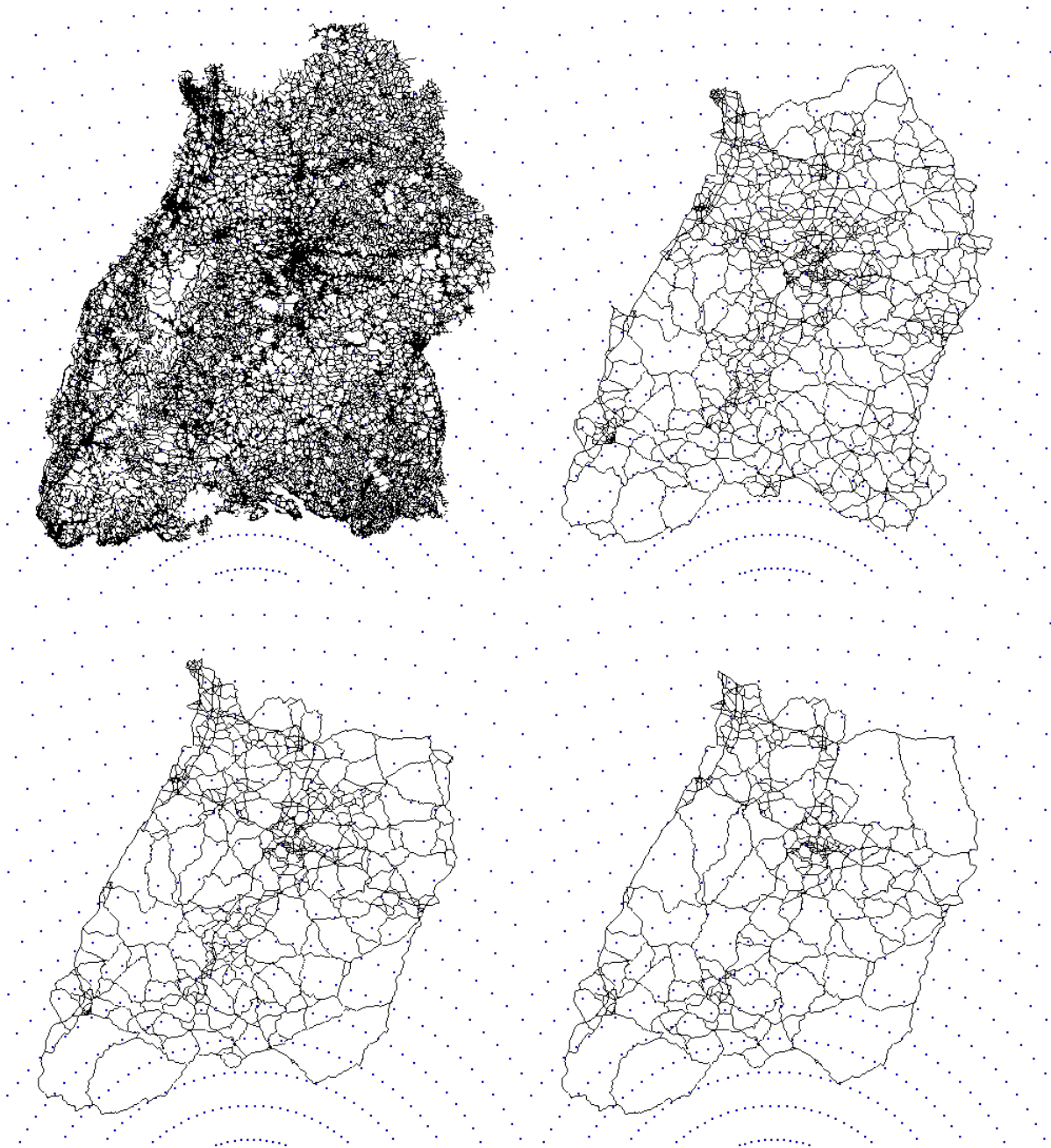


Abbildung 7: links oben: Originalgraph
links unten: minLevel = 45

rechts oben: minLevel = 35
rechts unten: minLevel = 55

4.2.2 Zusätzliche Erhöhung von maxLength

Eine weitere Reduktion der Knoten- und Kantenanzahl kann durch Erhöhung von maxLength erreicht werden. Für die in Tabelle 1 gezeigten minLevel-Werte wurde der maxLength-Wert so lange erhöht, so dass nur minimale Unterschiede sichtbar waren. Die resultierende Anzahl der Knoten und Kanten ist in Tabelle 2 zu sehen. (Auch hier bezieht sich

der Anteil der Knoten und Kanten auf den Originalgraphen).

Wie in Abbildung 8 für $minLevel = 25$ zu sehen ist, sind die Unterschiede, die durch eine geringe Erhöhung von $maxLength$ entstehen, kaum sichtbar. Dennoch enthält der resultierende Graph nur noch ca. ein Prozent so viele Knoten und Kanten wie der Originalgraph.

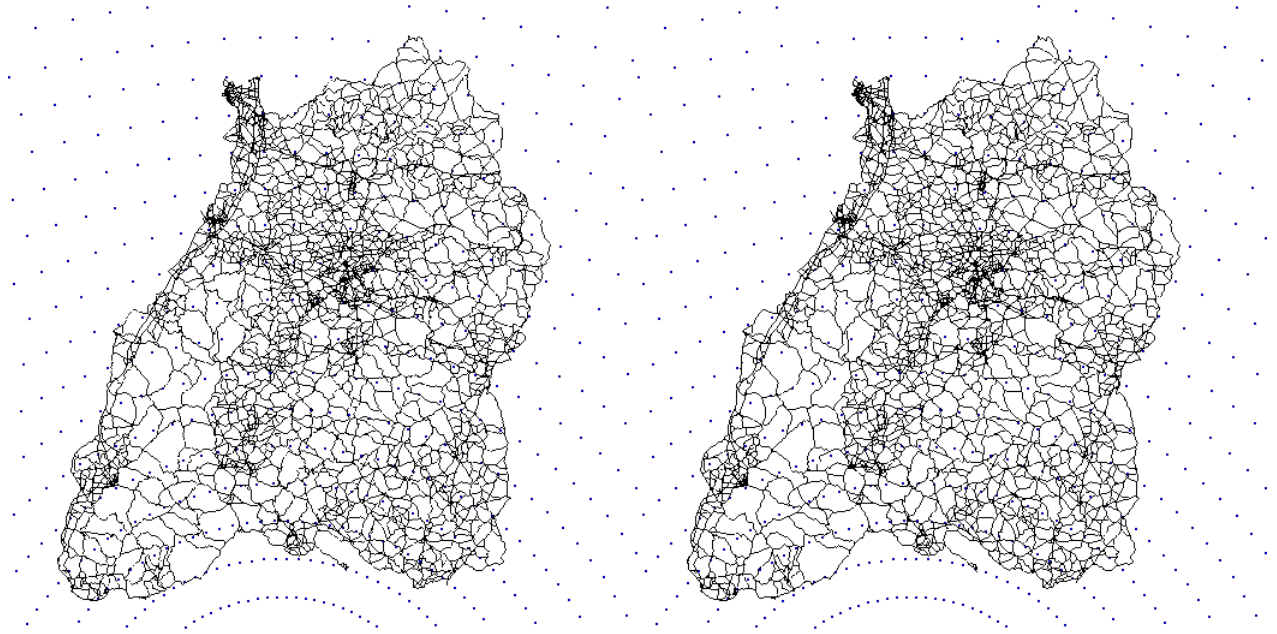


Abbildung 8: $minLevel = 25$
links: $maxLength = 0$ rechts: $maxLength = 0,01$

minLevel	maxLength	Anzahl Knoten	Anzahl Kanten	Anteil(Knoten)	Anteil(Kanten)
5	0,05	0,35 Mio.	0,75 Mio.	10,02%	10,41%
15	0,01	61 Tsd.	0,14 Mio.	1,73%	1,92%
25	0,01	41 Tsd.	82 Tsd.	1,16%	1,15%
35	0,01	27 Tsd.	50 Tsd.	0,77%	0,70%
45	0,01	20 Tsd.	34 Tsd.	0,55 %	0,47%
55	0,01	15 Tsd.	24 Tsd.	0,42%	0,33 %

Tabelle 2: Reduzierung der Anzahl der Knoten und Kanten durch Erhöhung von $maxLength$

Zu großer maxLength-Wert:

Bei den im vorherigen Abschnitt betrachteten Werten wurde darauf geachtet, dass durch die Wahl von maxLength kaum sichtbare Verschlechterungen auftreten. Dass eine zu große Erhöhung des maxLength-Wertes zu deutlich sichtbaren Shortcuts führt wird in Abbildung 9 gezeigt. Der minLevel-Wert beträgt bei allen vier Kartenausschnitten 25.

Wie anhand der beiden unteren Teilabbildungen zu sehen ist, führt eine zu starke Erhöhung von maxLength dazu, dass viele Strukturen nicht mehr erkennbar sind. Bei $\text{maxLength} = 0,3$ sind in den meisten Fällen nicht einmal grobe Straßenverläufe erkennbar, wie besonders im linken unteren Bereich des Graphen deutlich zu sehen ist.

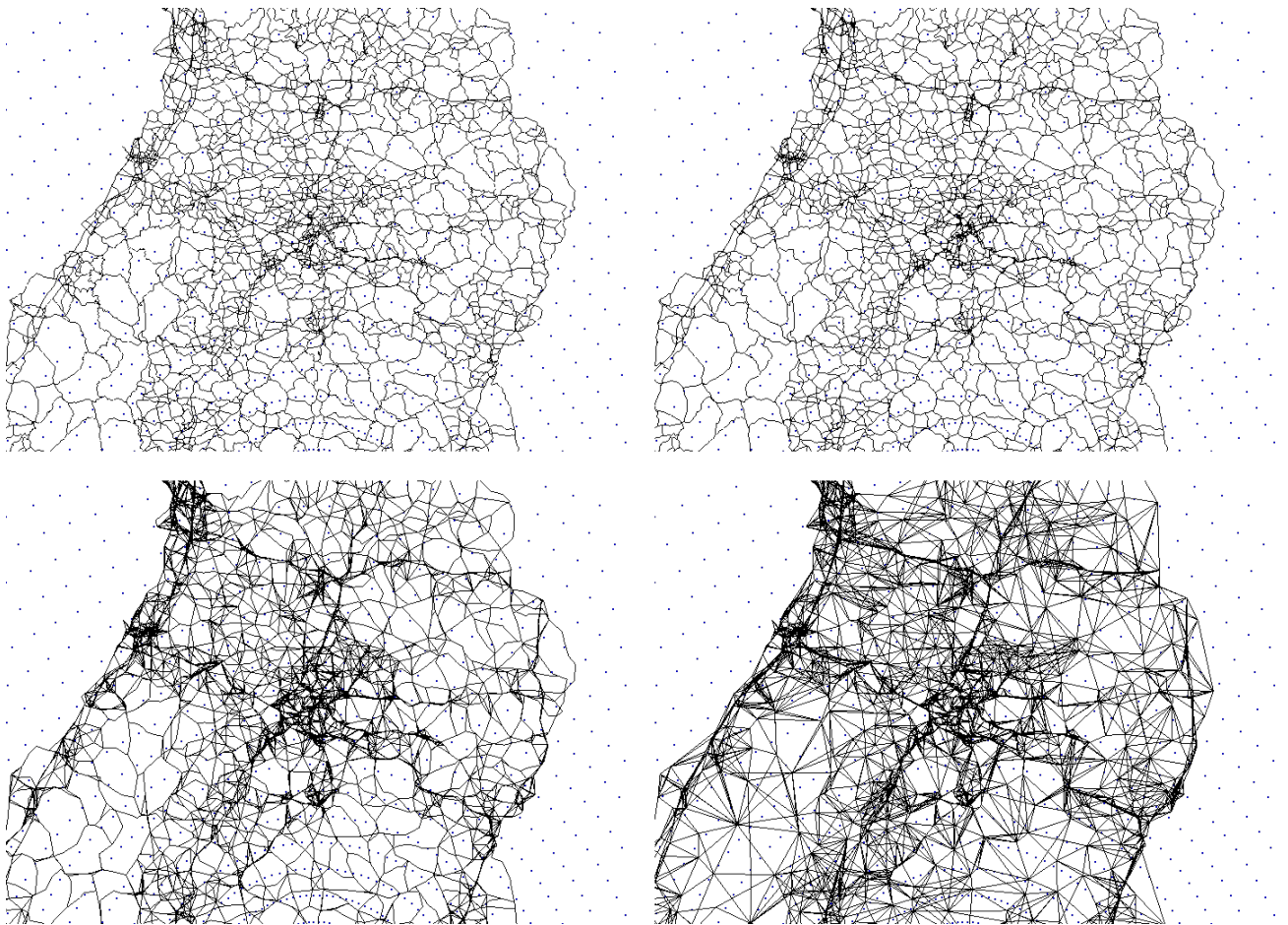


Abbildung 9: minLevel = 25

links oben: maxLength = 0

links unten: maxLength = 0,07

rechts oben: maxLength = 0,01

rechts unten: maxLength = 0,3

4.2.3 Veränderung des Zoom-Levels

Wird die Zoom-Stufe der Darstellung vergrößert, dann müssen die Werte für minLevel und maxLength angepasst werden. Dies hat verschiedene Gründe. Zum einen muss sichergestellt werden, dass relevante Straßen sichtbar sind, d.h. dass insbesondere bei der höchsten Zoom-Stufe der Wert für minLevel 0 sein sollte, da hierdurch alle Straßen sichtbar werden. Zum anderen muss der Wert für maxLength verringert werden, vor allem bei einem hohen minLevel-Wert, da es ansonsten sehr sichtbare Shortcuts gibt.

Abbildung 10 zeigt beispielhaft einige minLevel- und maxLength-Kombinationen für eine höhere Zoom-Stufe.



Abbildung 10: links oben: Originalgraph
rechts oben: minLevel = 5, maxLength = 0,05
links unten: minLevel = 15, maxLength = 0,01
rechts unten: minLevel = 25, maxLength = 0,01

Bei den beiden unteren Teilabbildungen ist zu sehen, dass viele durch die Kontraktion hinzugefügten Shortcuts zu lang sind. Um diesem Effekt entgegenzuwirken, muss bei beiden minLevel Werten ein geringerer maxLength Wert gewählt werden. Außerdem sind im Vergleich zum Originalgraphen die meisten Städte nicht mehr zu sehen. Bei dieser Zoom-Stufe wäre es besser, wenn kleinere Städte zumindest ansatzweise erkennbar sind, was durch die Wahl eines geringeren minLevel-Wertes erreicht werden kann.

Ein für diese Zoom-Stufe angemessener minLevel-Wert könnte z.B. 5 sein, mit einem maxLength-Wert von 0.002. Der resultierende Graph ist in Abbildung 11 zu sehen.

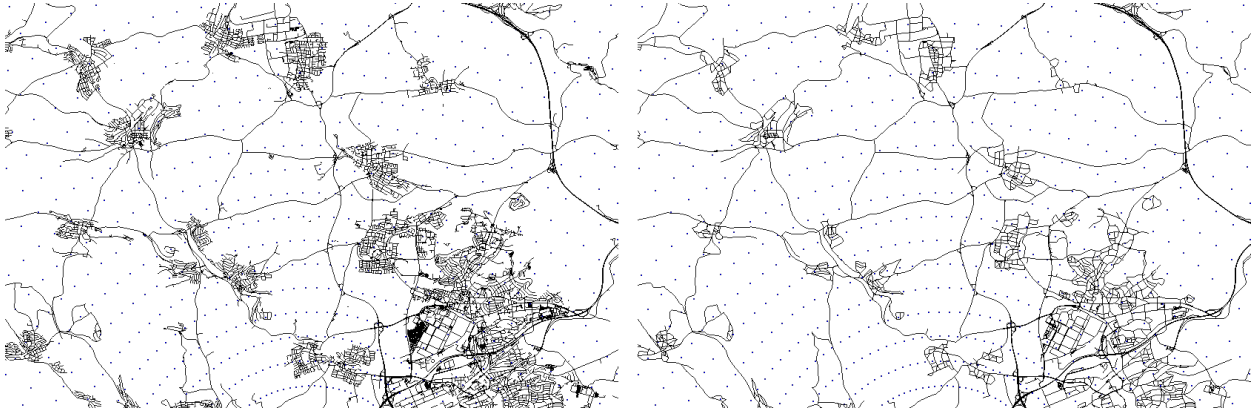


Abbildung 11: links: Originalgraph rechts: minLevel = 5, maxLength = 0,002

4.3 Knotendichte

4.3.1 Allgemeines

Bei diesem Kriterium wird zunächst ein Gitter über den Graphen gelegt. Vor jedem Kontraktionsschritt wird für jede Gitterzelle gezählt, wie viele Knoten sich in ihr befinden. Der Wert eines Knotens ist die Anzahl der Knoten, die sich in der entsprechenden Gitterzelle befinden.

Wird für dieses Kriterium ein positiver Koeffizient gewählt, so werden zuerst die Knoten kontrahiert, die eher weniger Knoten in ihrer (räumlichen) Umgebung haben. Das führt dazu, dass dichtere Gebiete länger erhalten bleiben, d.h. dass insbesondere größere Städte nach vielen Kontraktionsschritten noch erkennbar sind. Wird ein negativer Koeffizient gewählt, so findet das Gegenteil statt. Die Vereinfachung des resultierenden Graphen verläuft eher gleichmäßig.

Dieser Effekt wird in Abbildung 12 dargestellt, wobei beide Graphen vergleichbar viele Knoten und Kanten besitzen (links: 1,02 Mio. Knoten und 1,91 Mio. Kanten; rechts: 1,02 Mio. Knoten und 1,90 Mio. Kanten).

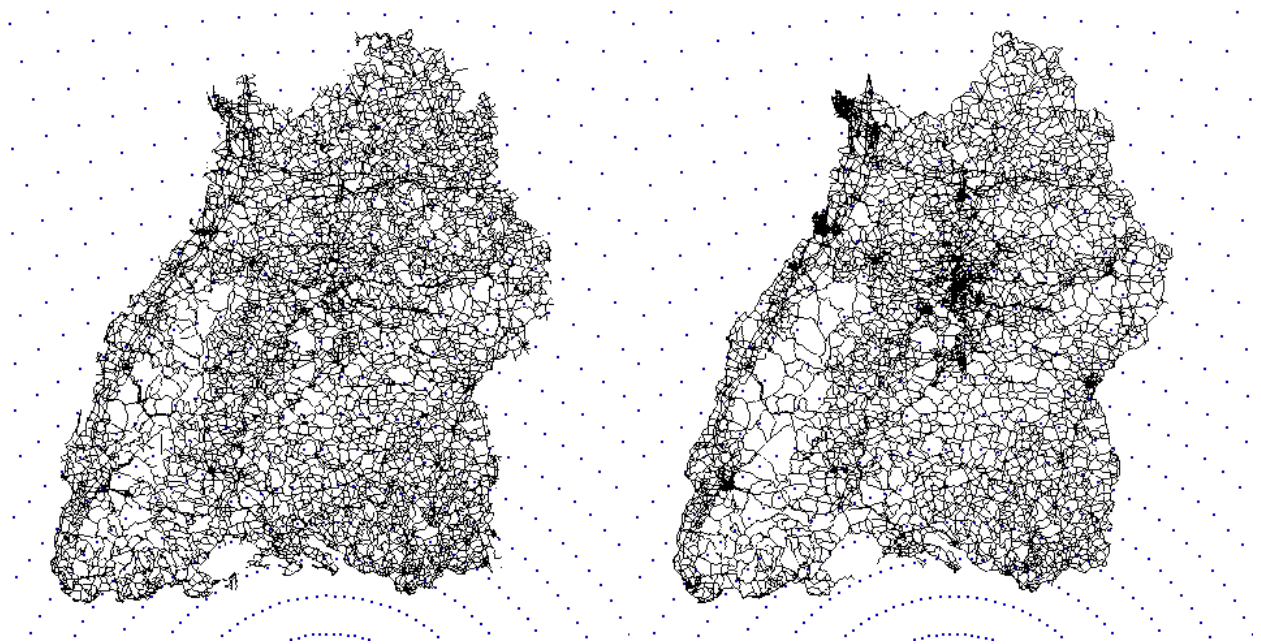


Abbildung 12: links: Gewichtung = -1 rechts: Gewichtung = 1
(bei vergleichbar vielen Knoten und Kanten)

Im weiteren Verlauf dieses Abschnittes wird die Gewichtung dieses Kriteriums auf einen positiven Wert gesetzt, da dadurch auch bei einer geringen Knoten- und Kantenanzahl größere Städte besser erkennbar sind als bei einer negativen Gewichtung.

Größe der Gitterzellen:

Ein wichtiger Parameter dieses Kriteriums ist die Größe der Gitterzellen. Diese wird im

Folgenden nicht explizit festgelegt, sondern relativ zum Wert $lat_{max} - lat_{min}$, wobei lat_{max} und lat_{min} die kleinsten bzw. größten Breitengrad-Werte der Knoten sind. Im Folgenden bezeichnet k die Anzahl der Teile, in die der Abstand $lat_{max} - lat_{min}$ unterteilt wird. Die quadratischen Gitterzellen besitzen somit eine Seitenlänge von $\frac{lat_{max} - lat_{min}}{k}$. Für den Graph von Baden-Württemberg gilt: $lat_{max} - lat_{min} = 2.27899$.

Anmerkung: Im obigen Beispiel (Abbildung 12) wurde $k = 25$ gewählt.

Die Größe der Gitterzellen hat einen großen Einfluss auf die Ausgeprägtheit von Städten bei geringer Knoten- und Kantenanzahl. Dies ist in Abbildung 13 für $k = 10$ und $k = 50$ zu sehen. Obwohl beide Graphen vergleichbar viele Knoten und Kanten enthalten (links: 1,06 Mio. Knoten und 1,98 Mio. Kanten; rechts: 1,04 Mio. Knoten und 1,93 Mio. Kanten), sind bei $k = 50$ viele Städte deutlich ausgeprägter dargestellt. Dies ist besonders im linken oberen und rechten Bereich gut zu sehen.

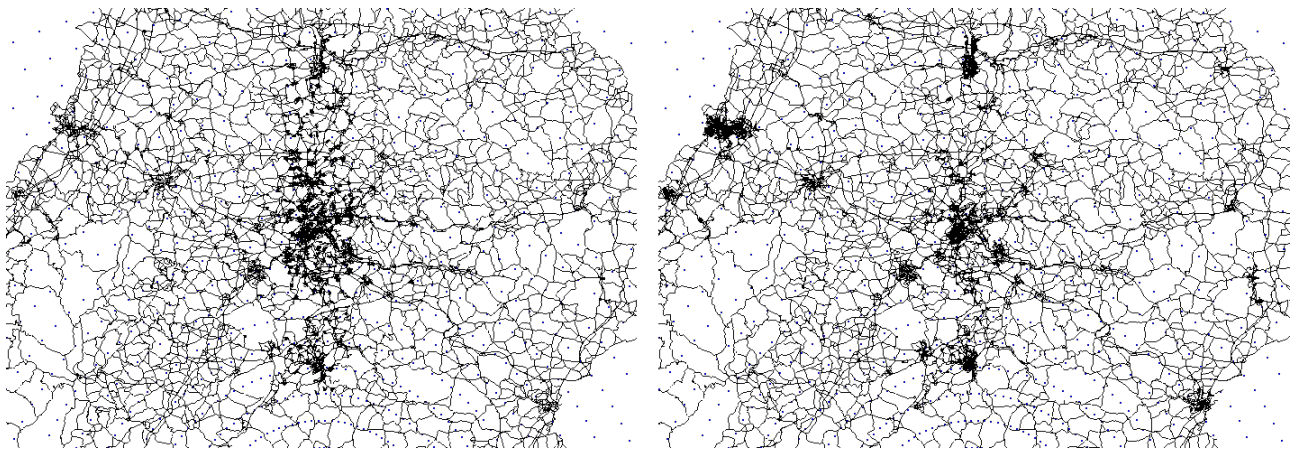


Abbildung 13: links: $k=10$ rechts: $k=50$
(bei vergleichbar vielen Knoten und Kanten)

Um einen geeigneten Wert für k zu finden, wurden die Werte 10, 25, 50, 75, 100, 200 und 1000 getestet. Hierzu wurden die Graphen bei etwa gleicher Knoten- und Kantenanzahl miteinander verglichen und dabei untersucht, wie deutlich kleinere Städte zu sehen sind. Die beste Darstellung konnte bei $k = 100$ festgestellt werden.

Abbildung 14 zeigt den Graphen von Baden-Württemberg mit $k = 10$, $k = 100$ und $k = 1000$ bei vergleichbar vielen Knoten und Kanten ($k = 10$: 0,93 Mio. Knoten und 1,71 Mio. Kanten; $k = 100$: 0,91 Mio. Knoten und 1,68 Mio. Kanten; $k = 1000$: 0,91 Mio. Knoten und 1,66 Mio. Kanten).

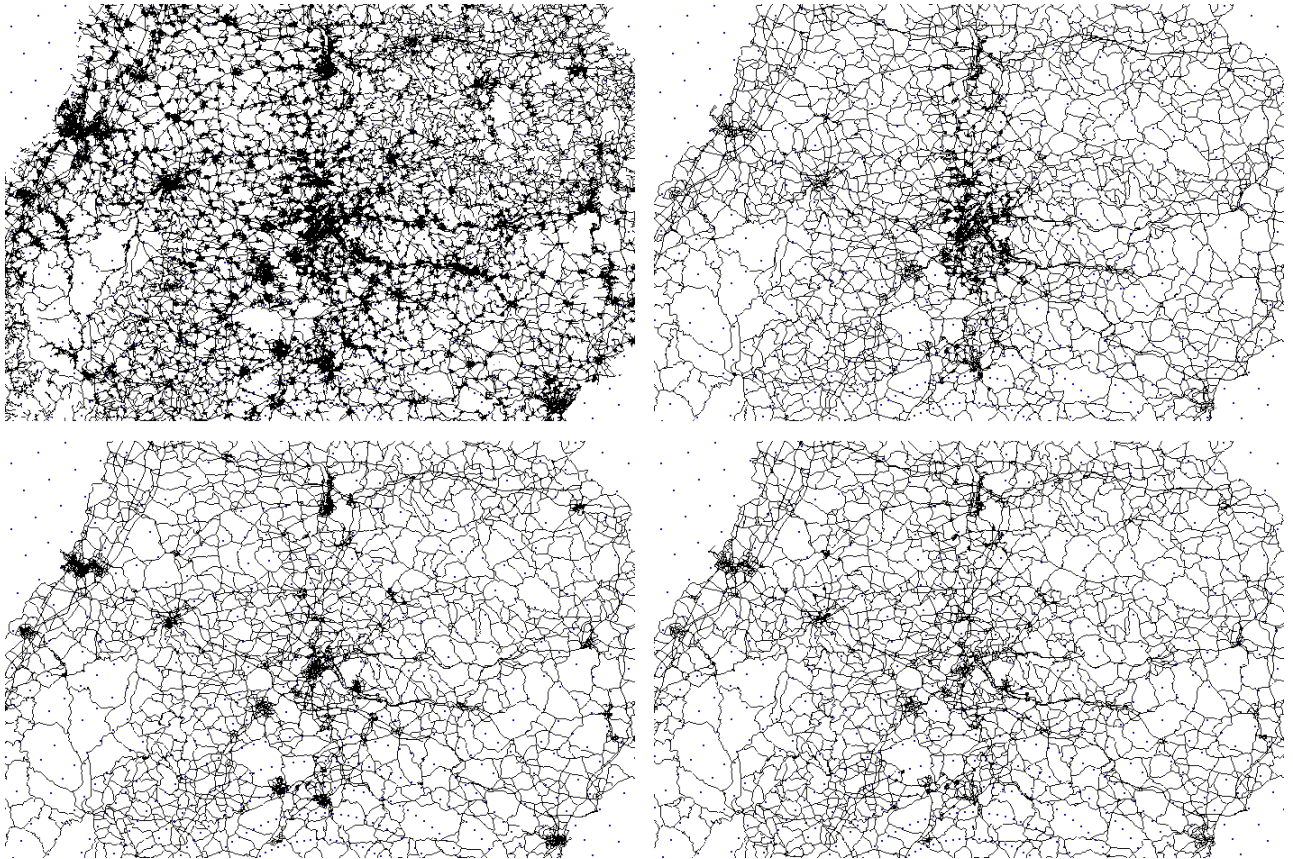


Abbildung 14: links oben: Originalgraph
links unten: $k = 100$

rechts oben: $k = 10$
rechts unten: $k = 1000$

Wie zu sehen ist, sind im Vergleich zu $k = 10$ bei $k = 100$ viele Städte deutlich ausgeprägter. Allerdings führt eine weitere Erhöhung von k nicht dazu, dass dieser Effekt verstärkt wird, wie bei $k = 1000$ zu sehen ist. Im Vergleich zu $k = 100$ sind bei $k = 1000$ viele Städte deutlich geringer ausgeprägt. Dieser Effekt konnte bereits bei $k = 200$ festgestellt werden.

4.3.2 Vergleich mit Edge Difference

Im direkten Vergleich zum Edge-Difference-Kriterium kann mithilfe des Knotendichte-Kriteriums eine bessere Erkennbarkeit von Städten erreicht werden. Dies gilt besonders dann, je höher die Zoom-Stufe ist. In Abbildung 15 sind beide Varianten bei vergleichbar vielen Knoten und Kanten (links: 1,16 Mio. Knoten und 2,15 Mio. Kanten; rechts: 1,13 Mio. Knoten und 2,12 Mio. Kanten) dargestellt.

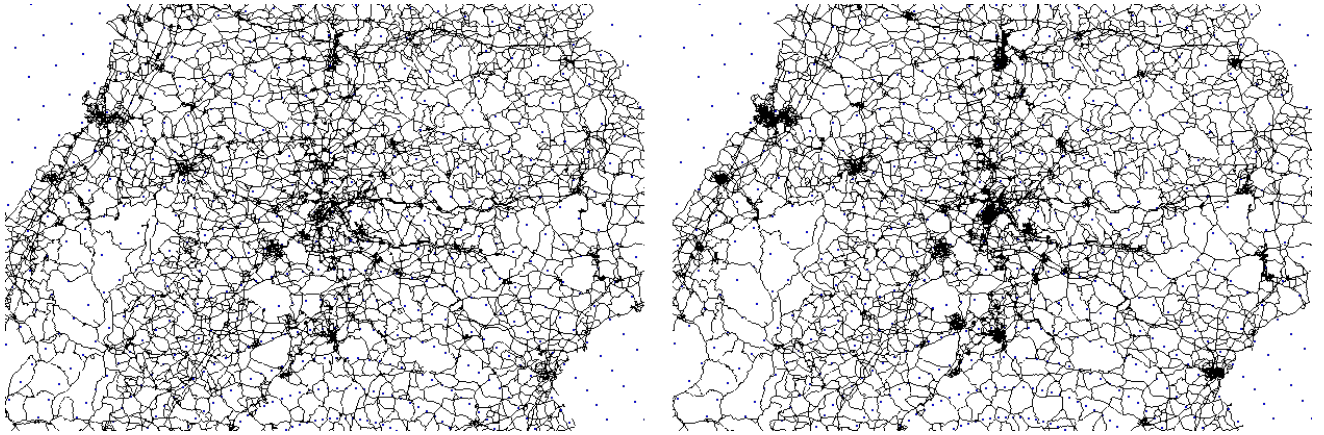


Abbildung 15: links: nur Edge Difference
rechts: nur Knotendichte (pos. Gewichtung, $k = 100$)

Es ist deutlich zu sehen, dass bei Verwendung des Knotendichte-Kriteriums viele Städte ausgeprägter dargestellt sind. Dieser Effekt ist bei einer geringeren Kantenanzahl noch deutlicher sichtbar, wie in Abbildung 16 zu sehen ist. Auch hier enthalten beide Graphen vergleichbar viele Knoten und Kanten (links: 0,93 Mio. Knoten und 1,72 Mio. Kanten; rechts: 0,91 Mio. Knoten und 1,68 Mio. Kanten).

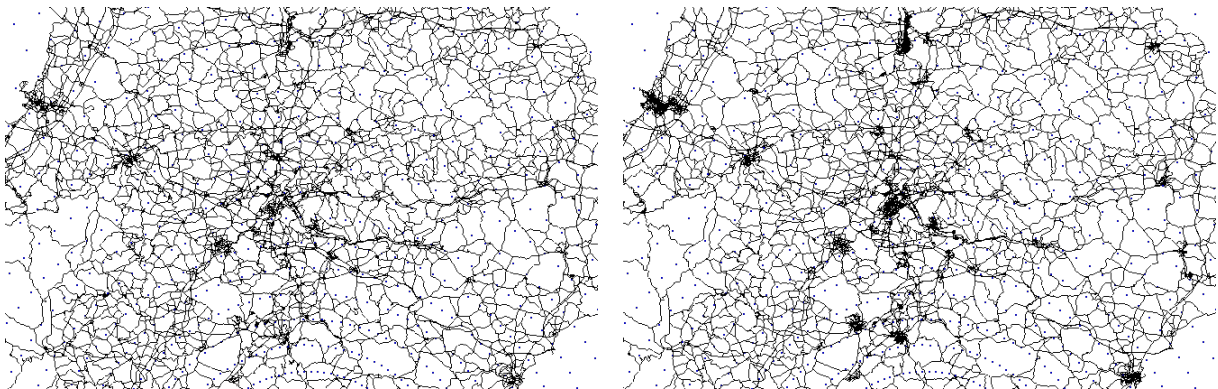


Abbildung 16: links: nur Edge Difference
rechts: nur Knotendichte (pos. Gewichtung, $k = 100$)

Allerdings besitzt die Kontraktion mithilfe des Knotendichte-Kriteriums im Vergleich zum Edge-Difference-Kriterium auch einen Nachteil:

Bei einem etwa gleich aussehendem Kartenausschnitt kann eine Erhöhung von max-Length dazu führen, dass bei Verwendung des Knotendichte-Kriteriums deutlich mehr Shortcuts sichtbar werden als unter Verwendung des Edge-Difference-Kriteriums.

In den Abbildung 17 und 18 ist dieser Effekt deutlich zu sehen. Abbildung 17 zeigt einen Kartenausschnitt, bei dem als Kriterium nur die Edge Difference verwendet wurde. Beide Teilabbildungen zeigen den Graphen mit minLevel-Wert 2. Bei der linken Teilabbildung

beträgt der `maxLength`-Wert 0. Der Graph der nach Erhöhung des `maxLength`-Wertes auf 0,2 entsteht ist in der rechten Teilabbildung zu sehen.

In Abbildung 18 ist derselbe Kartenausschnitt dargestellt, allerdings unter Verwendung des Knotendichte-Kriteriums. Auch hier sind beide `minLevel`-Werte auf 2 gesetzt. Hier ist durch eine Erhöhung des `maxLength`-Wertes auf 0,2 ein deutlich sichtbarer Unterschied feststellbar. Im Vergleich zum rechten Teil der Abbildung 17 sind viel mehr sichtbare Shortcuts enthalten, die zu einer deutlichen Verschlechterung der Darstellung führen.

(Bei `minLevel` = 2 und `maxLength` = 0 enthält der Graph bei Verwendung des Edge-Difference-Kriteriums 3,06 Mio. Knoten und 6,15 Mio. Kanten. Bei Verwendung des Knotendichte-Kriteriums sind es 3,30 Mio. Knoten und 6,66 Mio. Kanten.)



Abbildung 17: Nur Edge Difference, `minLevel` = 2:
links: `maxLength` = 0 rechts: `maxLength` = 0,2



Abbildung 18: Nur Knotendichte (pos. Gewichtung, $k = 100$), `minLevel` = 2:
links: `maxLength` = 0 rechts: `maxLength` = 0,2

4.3.3 Kombination von Knotendichte und Edge Difference

Eine naheliegende Möglichkeit, um diesen Nachteil des Knotendichte-Kriteriums zu verringern, ist die Kombination mit dem Edge-Difference-Kriterium. Hierzu wurden für die Gewichtungen des Edge-Difference- und Knotendichte-Kriteriums die Werte (1,1), (2,1) und (1,2) genauer untersucht. Dabei scheint die Gewichtung des Edge-Difference-Kriteriums mit 2 und die des Knotendichte-Kriteriums mit 1 die beste Kombination zu

sein, da bei Betrachtung der Graphen die deutlich sichtbaren Shortcuts am geringsten ausgeprägt waren, wie in Abbildung 19 beispielhaft gezeigt wird. Auch hier besitzen beide Graphen vergleichbar viele Knoten und Kanten (links: 2,09 Mio. Knoten, 4,33 Mio. Kanten; rechts: 2,09 Mio. Knoten, 4,27 Mio. Kanten).

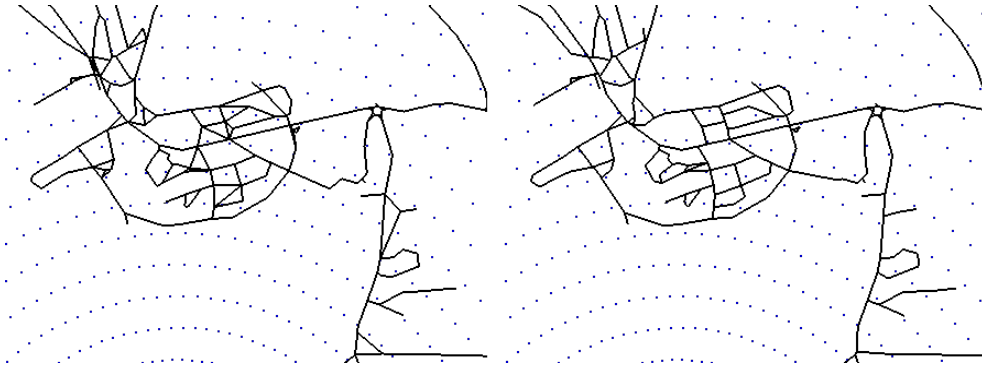


Abbildung 19: $\text{minLevel} = 2$, $\text{maxLength} = 0,2$
Koeffizienten: links: (1,1) rechts: (2,1)

Bei der Ausgeprägtheit von Städten wurden beim Vergleich der drei Kombinationen nur geringe Unterschiede festgestellt. Abbildung 20 zeigt die Kombinationen (1, 1) und (2, 1) für vergleichbar viele Knoten und Kanten (links: 0,90 Mio. Knoten, 1,66 Mio. Kanten; rechts: 0,90 Mio. Knoten, 1,64 Mio. Kanten).

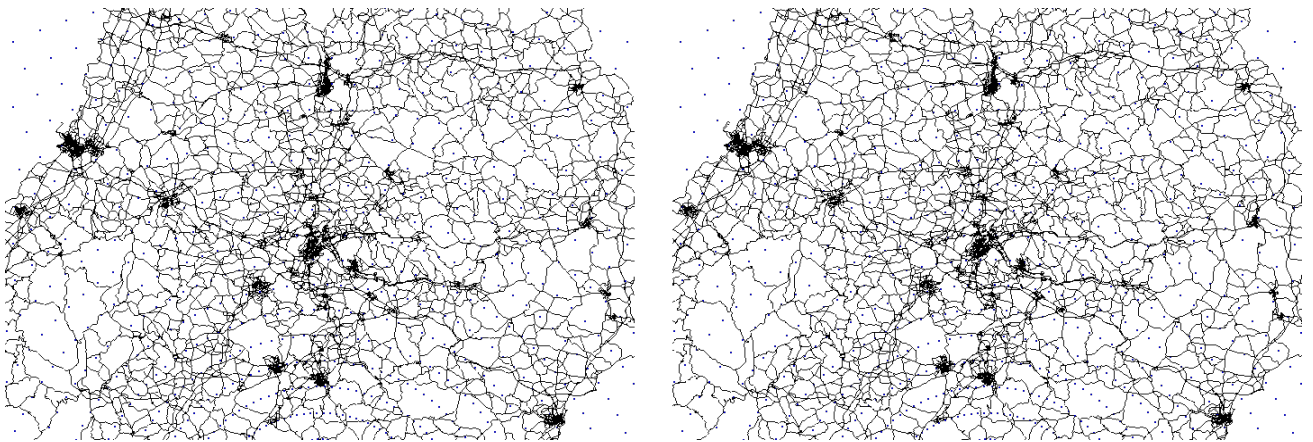


Abbildung 20: $\text{minLevel} = 17$, $\text{maxLength} = 0$
Koeffizienten: links: (1,1) rechts: (2,1)

Bei den drei betrachteten Kombinationen ist die (2, 1)-Kombination den anderen Kombinationen vorzuziehen. Im Folgenden wird diese Kombination mit den Einzelkriterien verglichen.

Einfluss der Kombination auf sichtbare Shortcuts:

Beim Vergleich der (2,1)-Kombination mit den jeweiligen Einzelkriterien sind deutliche Unterschiede sichtbar. Bei der Betrachtung von verschiedenen Kartenausschnitten wurden bei Verwendung der (2,1)-Kombination generell weniger sichtbare Shortcuts beobachtet als bei alleiniger Verwendung des Knotendichte-Kriteriums.

In Abbildung 21 ist dies für den bereits betrachteten Kartenausschnitt dargestellt. Durch die Kombination des Knotendichte-Kriteriums mit dem Edge-Difference-Kriterium ist eine deutliche Verringerung der sichtbaren Shortcuts zu sehen, die der alleinigen Verwendung des Edge-Difference-Kriteriums sehr nahe kommt.

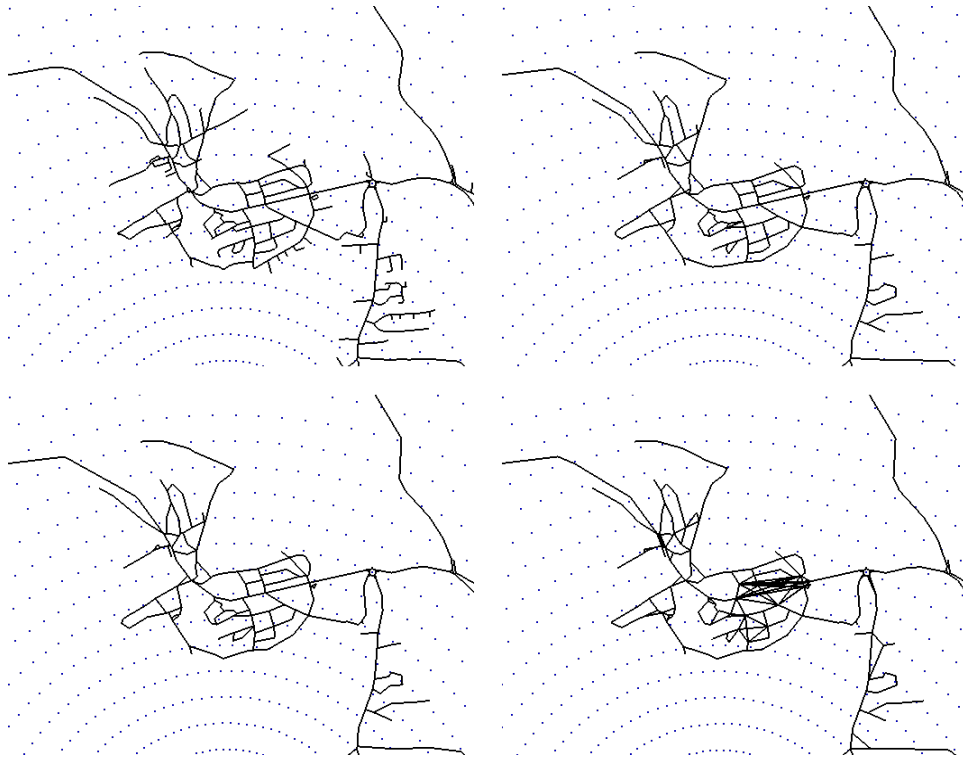


Abbildung 21: links oben: Originalgraph;
 bei allen anderen: minLevel = 2, maxLengh = 0,2
 rechts oben: (2,1)-Kombination
 links unten: nur Edge Difference
 rechts unten: nur Knotendichte (k=100)

Einfluss der Kombination auf die Ausgeprägtheit von Städten:

Beim Vergleich der (2,1)-Kombination mit der alleinigen Verwendung des Knotendichte-Kriteriums konnte bezüglich der Ausgeprägtheit von Städten bei Betrachtung mehrerer minLevel-Werte keine signifikante Verschlechterung festgestellt werden.

Abbildung 22 zeigt dies für einen Kartenausschnitt mit vergleichbar vielen Knoten und Kanten ((2,1)-Kombination: 0,90 Mio. Knoten und 1,64 Mio. Kanten; nur Knotendichte:

0,91 Mio. Knoten und 1,68 Mio. Kanten; nur Edge Difference: 0,93 Mio. Knoten und 1,72 Mio. Kanten).

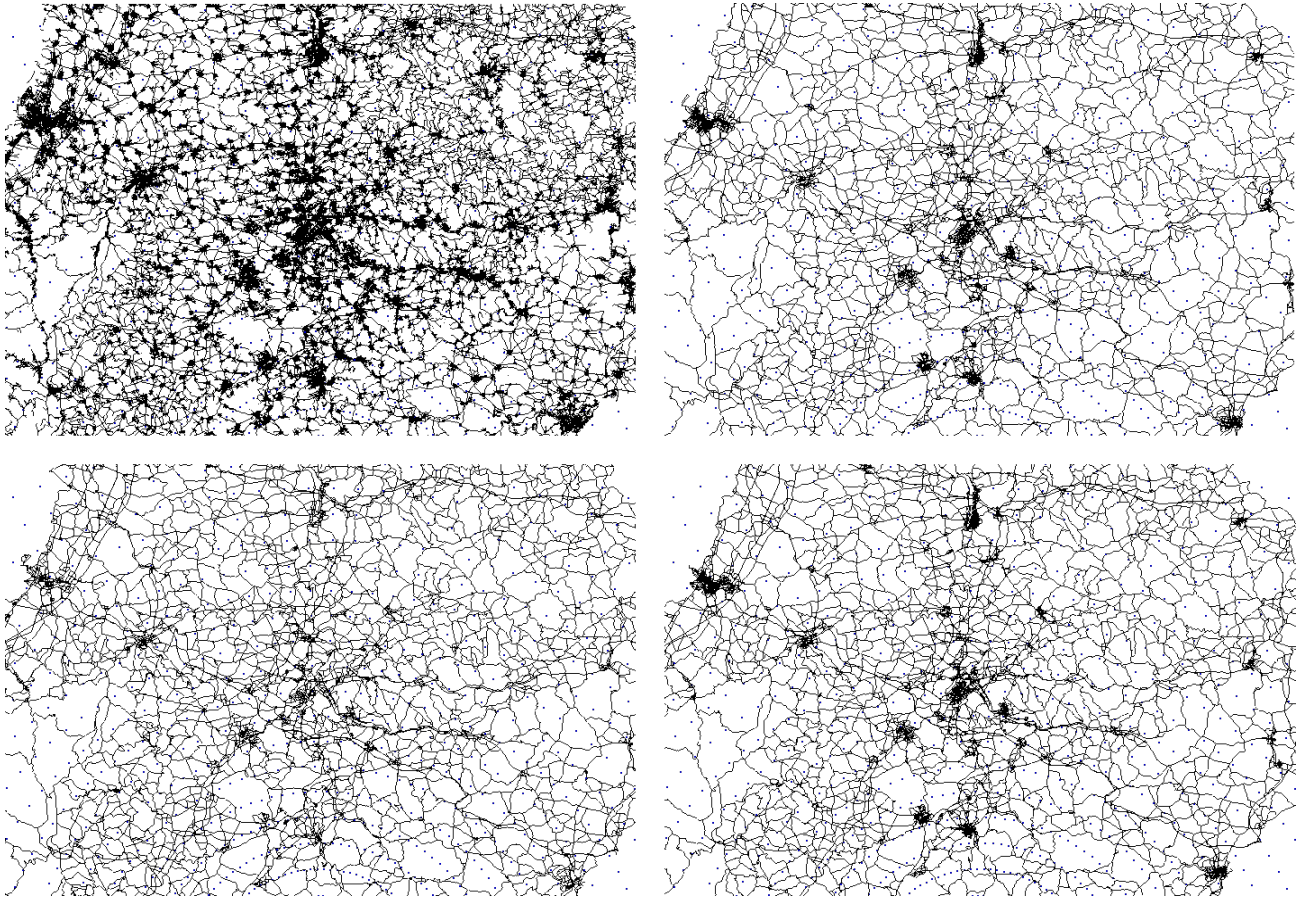


Abbildung 22: links oben: Originalgraph
rechts oben: (2,1)-Kombination
links unten: nur Edge Difference
rechts unten: nur Knotendichte ($k=100$)
(bei allen Graphen: $\text{maxLength} = 0$)

4.4 Durchschnittliche Kantendistanz

4.4.1 Allgemeines

Bei diesem Kriterium ist die Priorität eines Knotens der arithmetische Mittelwert der Längen der zu ihm inzidenten Kanten. Auch hier hat das Vorzeichen der Gewichtung einen sichtbaren Einfluss auf das Kontraktionsverhalten.

Im linken Teil von Abbildung 23 wurde eine positive Gewichtung gewählt, während im rechten Teil eine negative Gewichtung verwendet wurde. Beide Graphen besitzen vergleichbar viele Knoten und Kanten (links: 1,16 Mio. Knoten und 2,19 Mio. Kanten; rechts: 1,13 Mio. Knoten und 2,11 Mio. Kanten).

Wie zu sehen ist, verläuft die Vereinfachung des Graphen bei einer positiven Gewichtung wesentlich gleichmäßiger als bei einer negativen Gewichtung. Bei einer negativen Gewichtung sind Städte deutlicher ausgeprägt, obwohl dies nicht so stark wie unter Verwendung des Knotendichte-Kriteriums geschieht. Außerdem verschwinden bei einer negativen Gewichtung mehr kleinere Straßen als bei einer positiven Gewichtung, wie beispielsweise im oberen Bereich der Abbildung 23 gut zu sehen ist.

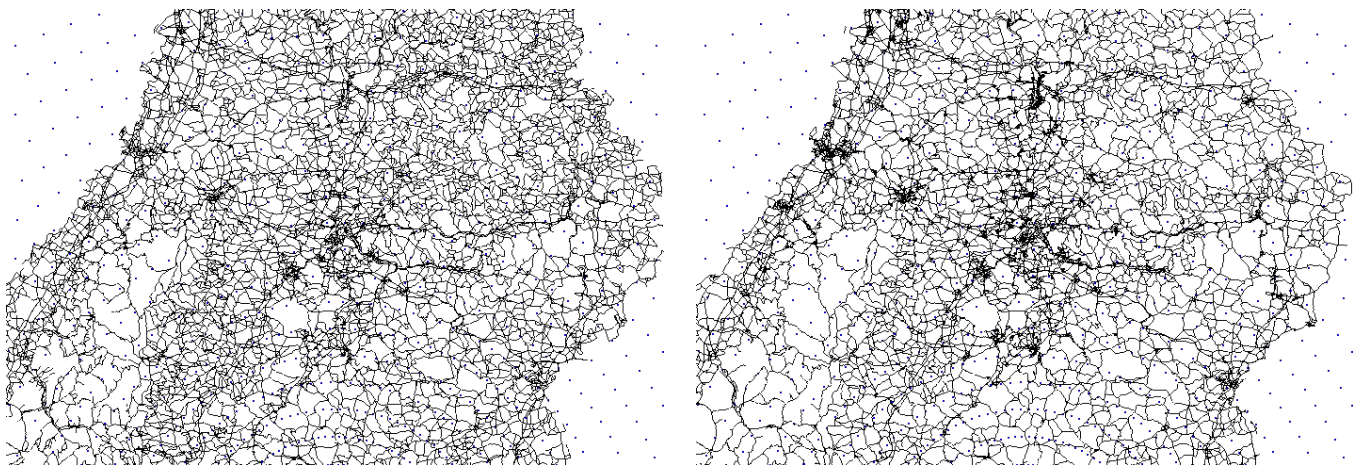


Abbildung 23: links: positive Gewichtung rechts: negative Gewichtung
(bei beiden Graphen: $\text{maxLength} = 0$)

Dieser Effekt ist besonders bei höheren Zoom-Stufen sichtbar, wie in Abbildung 24 zu sehen ist. (Hier handelt es sich um die gleichen Graphen wie in Abbildung 23; der einzige Unterschied ist die höhere Zoom-Stufe und der veränderte Kartenausschnitt).

Das hat zur Folge, dass bei einer positiven Gewichtung bei Vergrößerung des Kartenausschnittes der Wert von minLevel stärker verringert werden muss als bei einer negativen Gewichtung, da sonst bei den verbleibenden Städten nur wenige Strukturen erkennbar sind.

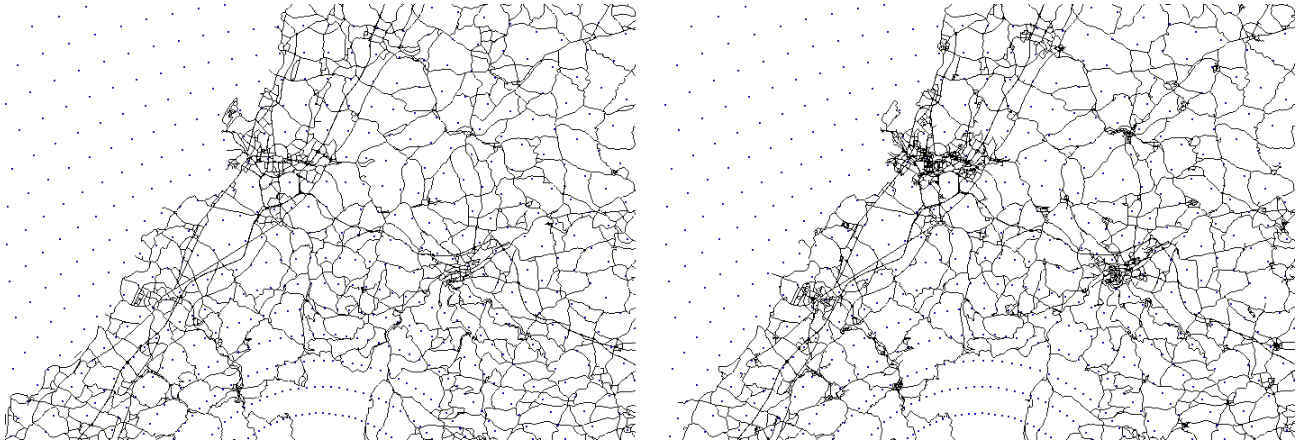


Abbildung 24: links: positive Gewichtung rechts: negative Gewichtung
(bei beiden Graphen: $\text{maxLength} = 0$)

Charakteristische Form von Straßenverläufen

Im Folgenden wird näher untersucht, welchen Einfluss die Gewichtung des Kriteriums auf die Form von Straßenverläufen hat. Dazu wird, wie in Abbildung 25 gezeigt wird, maxLength auf einen festen Wert gesetzt und minLevel schrittweise erhöht.

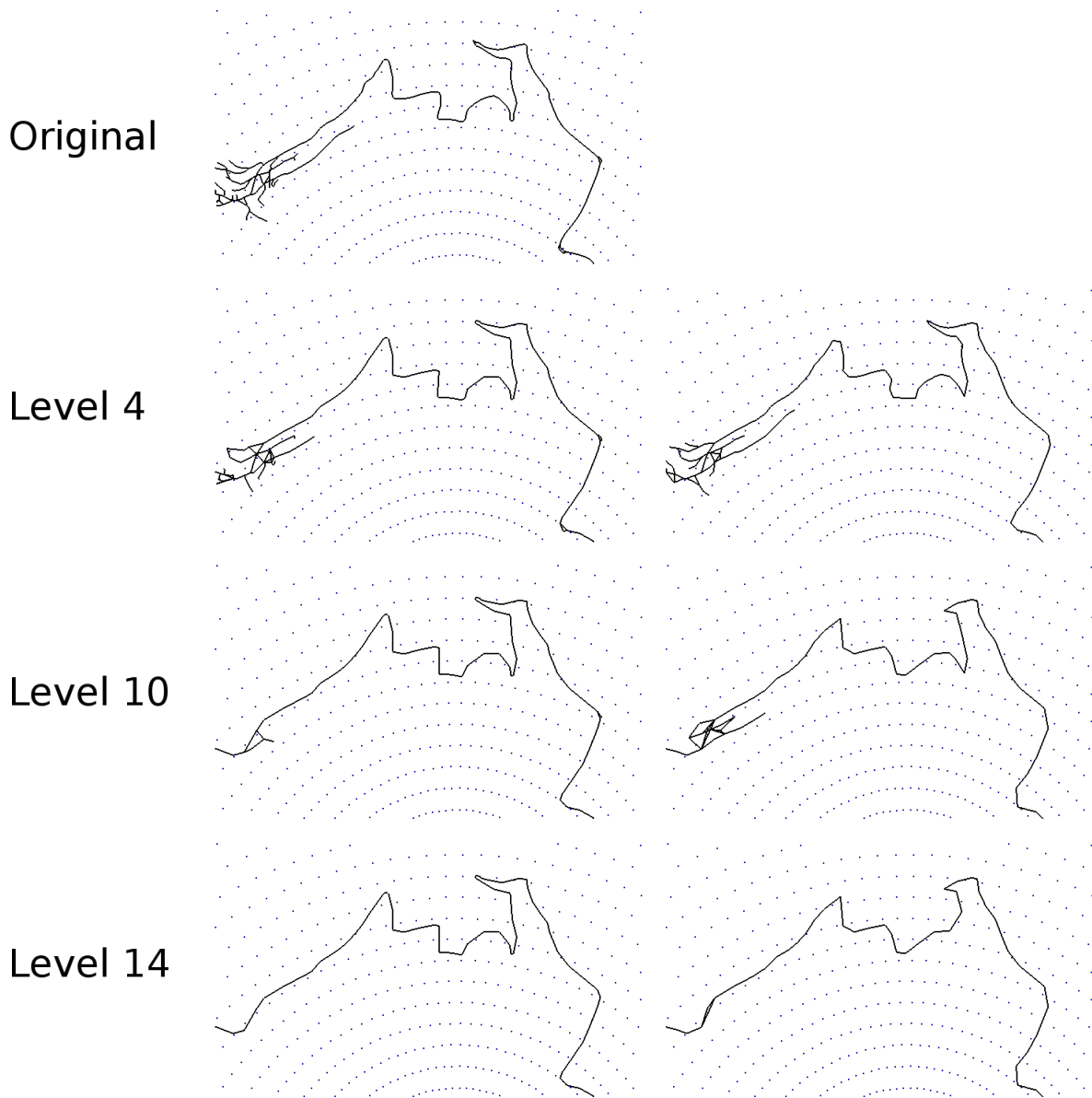


Abbildung 25: maxLength : 0,002

links: negative Gewichtung

rechts: positive Gewichtung

Eine weitere Erhöhung des minLevels hat bei diesem Ausschnitt keinen sichtbaren Einfluss (abgesehen vom der Entfernung der gesamten Straße bei sehr starker Erhöhung).

Wie zu sehen ist, bleibt bei negativer Gewichtung des Kriteriums die Form des Straßenverlaufes besser erhalten als bei einer positiven Gewichtung. Eine mögliche Erklärung hierfür ist, dass runde Straßenabschnitte aus vielen kleineren Kanten bestehen. Diese werden bei einer negativen Gewichtung des Kriteriums später kontrahiert als bei einer positiven Gewichtung.

Der in Abbildung 25 gezeigte Effekt wurde auch bei anderen Straßenverläufen beobachtet

und ist stärker, je größer der Wert von `maxLength` ist.

4.4.2 Vergleich mit (2,1)-Kombination

Ausgeprägtheit von Städten:

Im direkten Vergleich mit der (2,1)-Kombination wurde beobachtet, dass die Ausgeprägtheit von Städten bei der (2,1)-Kombination deutlich besser ist als bei alleiniger Verwendung des Durchschnittliche-Kantendistanz-Kriteriums.

Dieser Unterschied ist in Abbildung 26 zu sehen. Hier wurde für das Durchschnittliche-Kantendistanz-Kriterium eine negative Gewichtung gewählt. Die beiden Graphen enthalten eine vergleichbare Knoten- und Kantenanzahl (bei beiden Graphen: 1,20 Mio. Knoten und 2,25 Mio. Kanten).

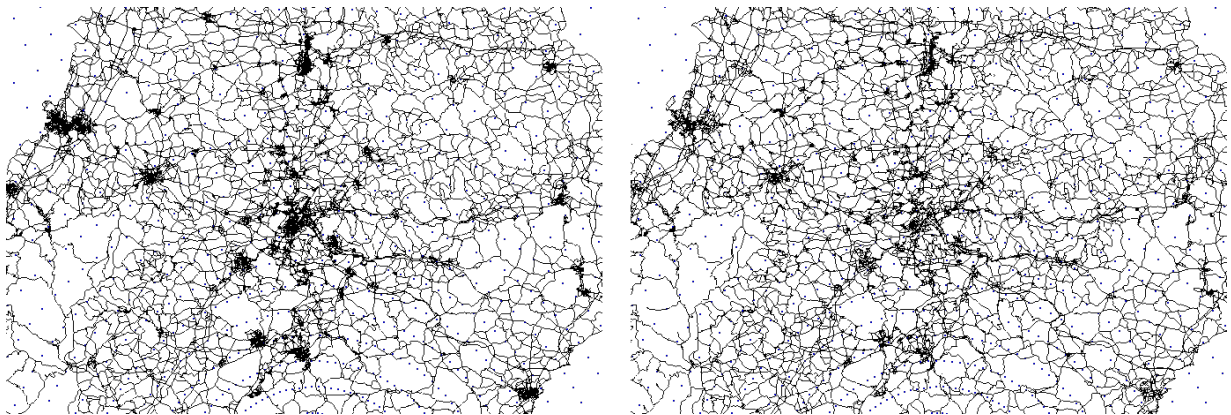


Abbildung 26: links: (2,1)-Kombination
rechts: nur durchschnittliche-Kantendistanz (negativ gewichtet)
bei beiden: `maxLength` = 0

Straßenverläufe:

Bezüglich der Simplifizierung von Straßenverläufen wurde beobachtet, dass bei alleiniger Verwendung des Durchschnittliche-Kantendistanz-Kriteriums Straßenverläufe tendenziell besser erhalten bleiben als bei Verwendung der (2,1)-Kombination. Außerdem sind Kurven der simplifizierten Straßenverläufe bei der (2,1)-Kombination eher eckiger, während bei alleiniger Verwendung der durchschnittlichen Kantendistanz eher runder sind.

Ein Beispiel hierfür ist in Abbildung 27 zu sehen. Beim gewählten Kartenausschnitt wurden zuerst die `minLevel`-Werte so eingestellt, so dass beide Graphen vergleichbar viele Kanten und Knoten besitzen. Bei der (2,1)-Kombination sind dies 1,57 Mio. Knoten und 3,01 Mio. Kanten. Bei alleiniger Verwendung des Durchschnittliche-Kantendistanz-Kriteriums sind dies 1,55 Mio. Knoten und 2,96 Mio. Kanten.

Anschließend wurde der `maxLength` Wert bei beiden Graphen auf 0,002 erhöht. Das Resultat ist in Abbildung 27 zu sehen.



Abbildung 27: links: Originalgraph
mitte: nur durchschnittliche Kantendistanz (neg. gewichtet)
rechts: (2,1)-Kombination

Bei der Betrachtung von Straßenverläufen wurden vereinzelt auch Verschlechterungen beobachtet, wie in Abbildung 28 zu sehen ist. Hier ist der Straßenverlauf bei Verwendung der (2,1)-Kombination deutlich genauer als bei alleiniger Verwendung des Durchschnittliche-Kantendistanz-Kriteriums.



Abbildung 28: links: Originalgraph
mitte: nur durchschnittliche Kantendistanz (neg. gewichtet)
rechts: (2,1)-Kombination

4.4.3 Kombination

Im Folgenden wird eine Kombination des Durchschnittliche-Kantendistanz-Kriteriums mit der bereits erfolgten Kombination des Edge-Difference- und Knotendichte-Kriteriums genauer untersucht. Dabei wird eine negative Gewichtung des Durchschnittliche-Kantendistanz-Kriteriums verwendet. Die untersuchten Kombinationen umfassen die Gewichtung mit den Werten -1 und -2 . Die Gewichtung des Edge-Difference-Kriteriums wird auf 2 , die des Knotendichte-Kriteriums wird auf 1 gesetzt (mit $k=100$). Die Kurzschreibweise der Gewichtungen wird wie bereits in Abschnitt 4.1.2 erläutert als 3-Tupel (a, b, c) notiert, wobei a die Gewichtung des Edge-Difference-Kriteriums, b die Gewichtung des Knotendichte-Kriteriums und c die Gewichtung des Durchschnittliche-Kantendistanz-Kriteriums ist.

Vergleich von (2,1,-1) und (2,1,-2):

Bezüglich des Kontraktionsverhaltens bei Straßenverläufen wurden die Kombinationen (2,1,-1) und (2,1,-2) für $\text{minLevel} = 14$ und $\text{maxLength} = 0,002$ genauer untersucht. Hierbei konnten im direkten Vergleich jedoch nur sehr geringe Unterschiede festgestellt werden.

Auch bei anderen Werten für minLevel und maxLength konnten keine signifikanten Unterschiede bezüglich der Form von Straßenverläufen festgestellt werden.

Bezüglich der Entstehung sichtbarer Shortcuts wurden oft nur geringe Unterschiede festgestellt, wobei bei der (2,1,-1)-Kombination geringfügig weniger sichtbare Shortcuts beobachtet wurden als bei der (2,1,-2)-Kombination. (Generell ist bei einer stärkeren Gewichtung des Edge-Difference-Kriteriums (im Vergleich zu den anderen Gewichtungen) eine geringere Anzahl an Shortcuts zu sehen.)

Ein Beispiel für einen Unterschied bezüglich sichtbarer Shortcuts ist in Abbildung 29 zu sehen.



Abbildung 29: links: (2,1,-1)-Kombination
rechts: (2,1,-2)-Kombination
bei beiden: $\text{minLevel} = 4$, $\text{maxLength} = 0,1$

Da die Unterschiede bezüglich des Straßenverlaufes sehr gering sind und bei einer Gewichtung des Durchschnittliche-Kantendistanz-Kriteriums mit -1 eher weniger sichtbare Shortcuts entstehen, wird im Folgenden diese Kombination mit der (2,1,0)-Kombination verglichen.

Vergleich von (2,1,0) mit (2,1,-1)

Durch die Erweiterung um die durchschnittliche Kantendistanz (mit Gewichtung -1) konnten nur sehr geringe Verbesserungen festgestellt werden.

Abbildung 30 zeigt einen Kartenausschnitt, auf dem mehrere Straßenverläufe zu sehen sind. Wie beim Vergleich der beiden Teilabbildungen zu sehen ist, kann durch Verwendung des Durchschnittliche-Kantendistanz-Kriteriums keine signifikante Verbesserung erzielt werden.

Auch bei höheren Zoom-Stufen und anderen minLevel - und maxLength -Werten konnten

keine deutlichen Verbesserungen beobachtet werden.

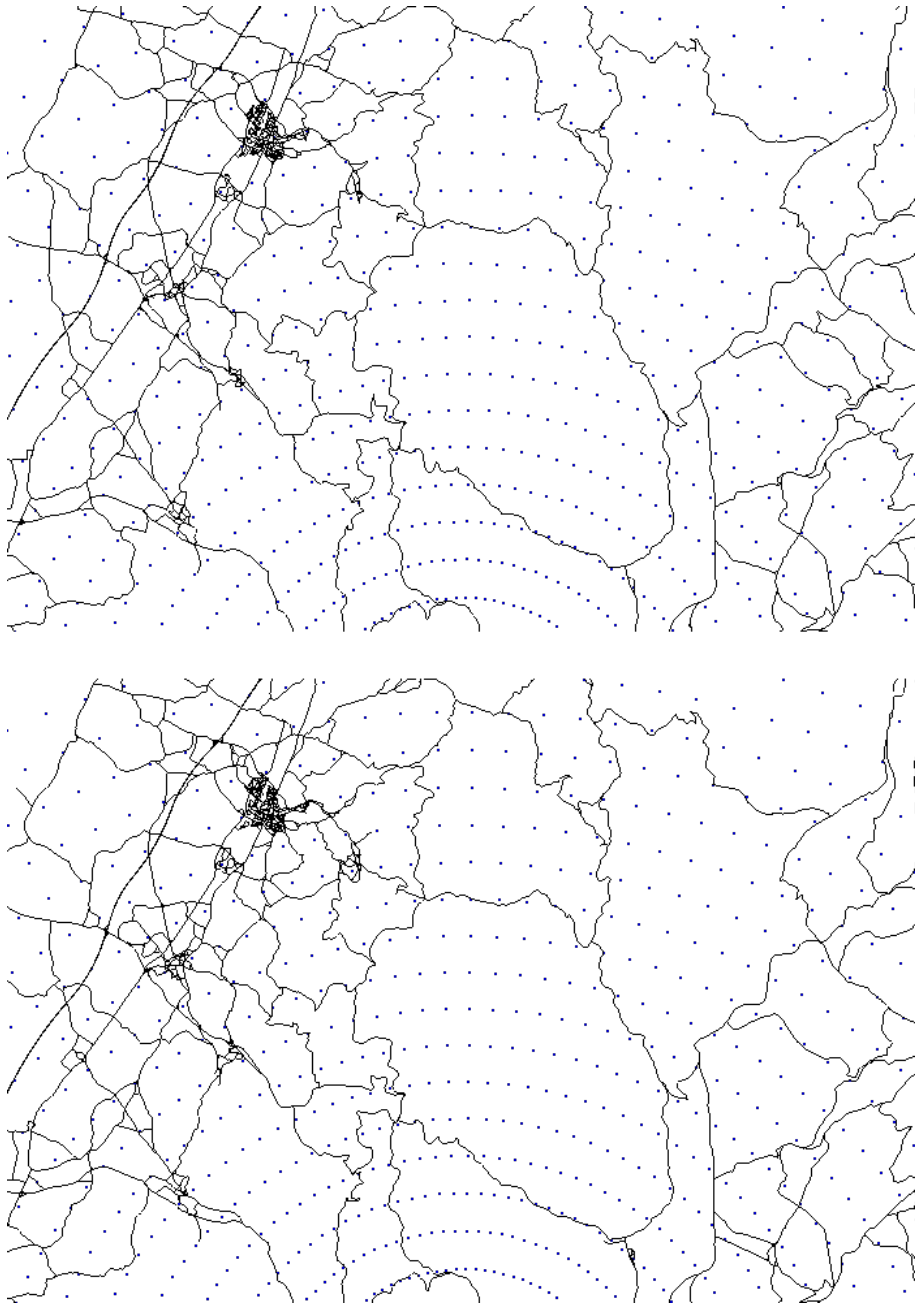


Abbildung 30: oben: (2,1,0)-Kombination
 unten: (2,1,-1)-Kombination
 bei beiden: minLevel = 14, maxLength = 0,003

Bei der (2,1,-1)-Kombination wurden jedoch deutlich mehr sichtbare Shortcuts beobachtet als bei der (2,1,0)-Kombination. Ein Beispiel hierfür ist in Abbildung 31 zu sehen.

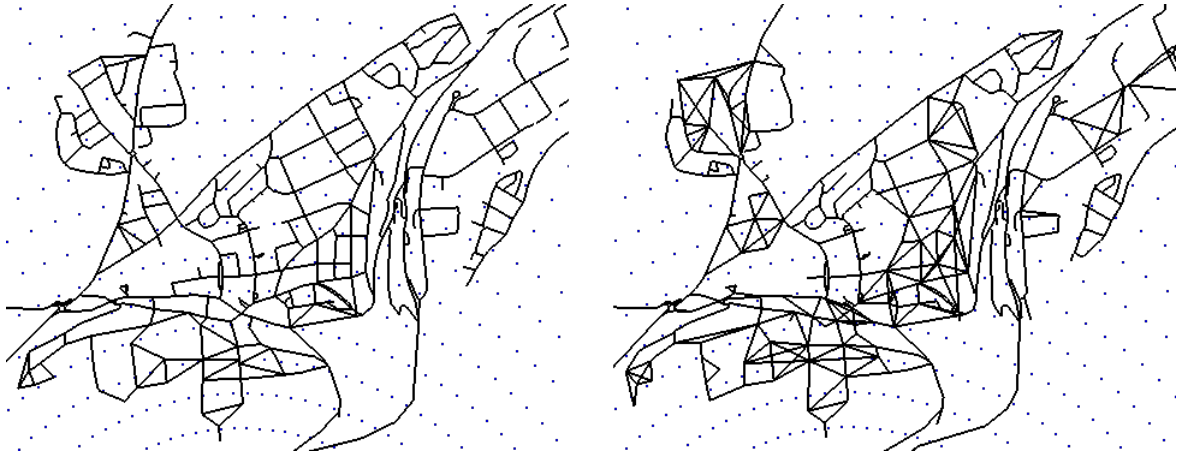


Abbildung 31: links: $(2,1,0)$ -Kombination
rechts: $(2,1,-1)$ -Kombination
bei beiden: $\text{minLevel} = 3$, $\text{maxLength} = 0,01$

Zusammengefasst kann gesagt werden, dass die Veränderungen der Form von Straßenverläufen durch zusätzliche Betrachtung der durchschnittlichen Kantendistanz gering sind. Es entstehen jedoch deutlich mehr sichtbare Shortcuts, die das Gesamtbild der Karte deutlich verschlechtern können.

Somit hat die Erweiterung der $(2,1)$ -Kombination um das Durchschnittliche-Kantendistanz-Kriterium mehr Nachteile als Vorteile und ist daher nicht sinnvoll.

4.5 Winkel

Im Folgenden wird untersucht, welchen Einfluss die Verwendung des Winkels als Kriterium besitzt. Da das Kriterium nur für Knoten vom Grad zwei berechnet werden kann, wird es nicht einzeln betrachtet, sondern in Kombination mit anderen Kriterien.

Die Gewichtung des Edge-Difference-Kriteriums wird auf zwei, die des Knotendichte-Kriteriums auf eins (mit $k = 100$) gesetzt. Das Durchschnittliche-Kantendistanz-Kriterium wird nicht verwendet, um Wechselwirkungen bei der Kontraktion von Straßenverläufen zu verringern. Die Kombination der Gewichtungen wird, wie in Abschnitt 4.1.2 erläutert, als 4-Tupel notiert, wobei der letzte Wert die Gewichtung des Winkel-Kriteriums ist.

Um einen verstärkten Einfluss des Winkel-Kriteriums feststellen zu können, wird für die Gewichtung ein vergleichsweise großer Wert verwendet. Daher werden im Folgenden die Gewichtungen $(2,1,0,-3)$ und $(2,1,0,3)$ näher betrachtet.

Beim Vergleich dieser beiden Kombinationen konnten keine eindeutigen Tendenzen bezüglich der Veränderung der Form von Straßenverläufen erkannt werden. Diese Beobachtung wird in Abbildung 32 dargestellt, bei dem ein Kartenausschnitt mit mehreren Straßenverläufen zu sehen ist. Ohne Erhöhung des `maxLength`-Wertes, d.h. bei `minLevel = 10` und `maxLength = 0` besitzen beide Kombinationen vergleichbar viele Knoten und Kanten ($(2,1,0,3)$ -Kombination: 1,56 Mio. Knoten und 2,94 Mio. Kanten; $(2,1,0,-3)$ -Kombination: 1,62 Mio. Knoten und 3,14 Mio. Kanten).

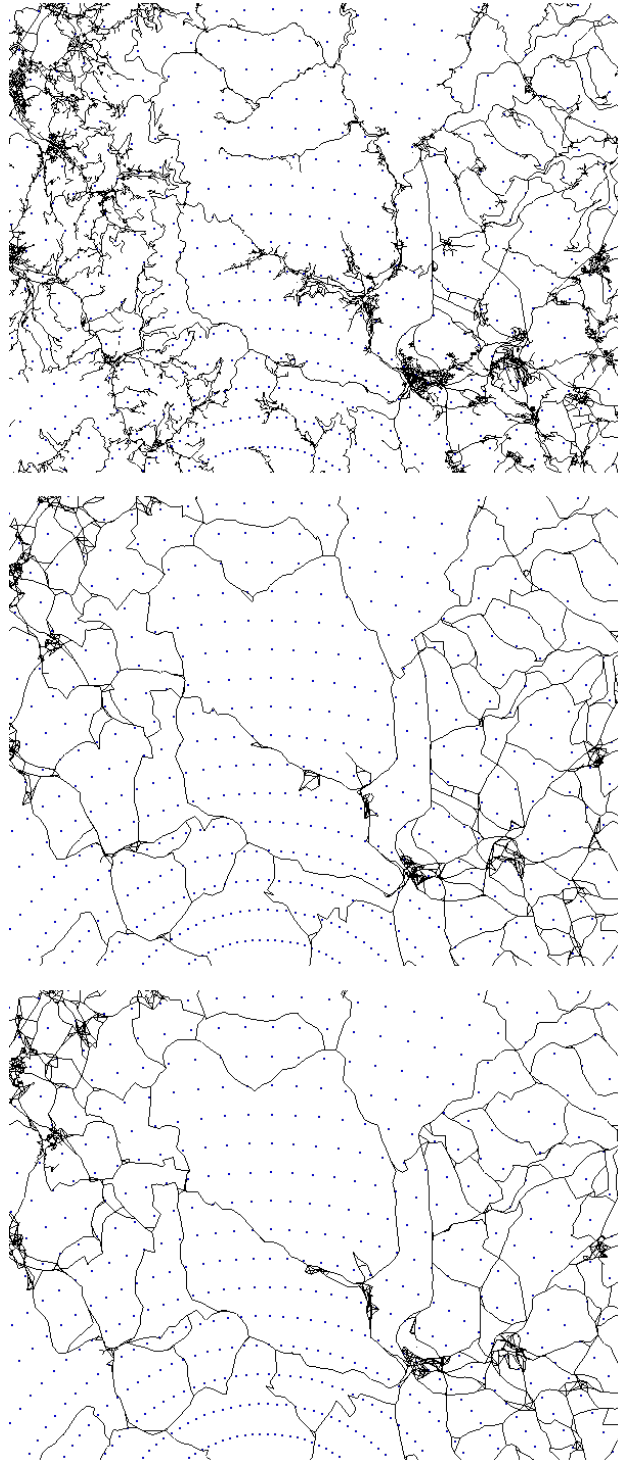


Abbildung 32: oben: Originalausschnitt
mitte: (2,1,0,3)
unten : (2,1,0,-3)
mitte und unten: minLevel = 10, maxLength = 0,01

Auf dem ersten Blick scheint bei negativer Gewichtung des Winkel-Kriteriums eine stärkere Glättung von Straßenverläufen stattzufinden, wie in Abbildung 33 für einen vergrößerten Kartenausschnitt zu sehen ist.

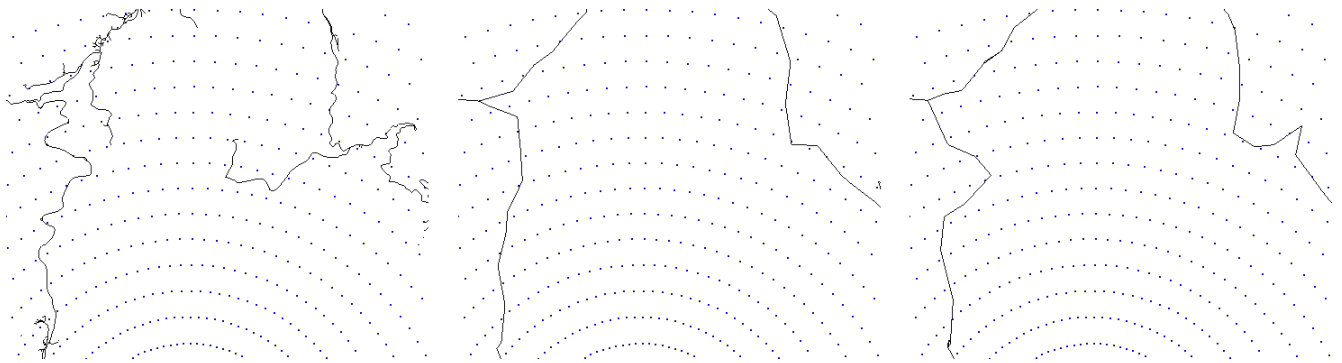


Abbildung 33: links: Originalausschnitt

mitte: $(2,1,0,-3)$

rechts: $(2,1,0,3)$

mitte und rechts: $\text{minLevel} = 10$, $\text{maxLength} = 0,01$

Bei genauerer Betrachtung kann man jedoch feststellen, dass bei einigen Straßenverläufen keine signifikanten Unterschiede zu sehen sind, und dass bei einigen Straßenverläufen sogar das Gegenteil dessen stattfindet, d.h. dass bei positiver Gewichtung eine stärkere Glättung stattfindet, wie in Abbildung 34 zu sehen ist.

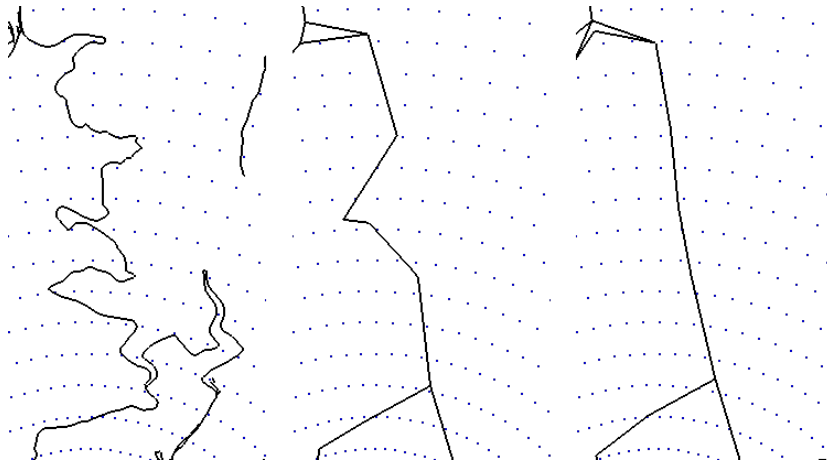


Abbildung 34: links: Originalausschnitt

mitte: $(2,1,0,-3)$

rechts: $(2,1,0,3)$

mitte und rechts: $\text{minLevel} = 10$, $\text{maxLength} = 0,01$

Dieser Effekt wurde auch bei $\text{maxLength} = 0,005$ und $\text{maxLength} = 0,02$ beobachtet.

Da bei Betrachtung der Winkel von Knoten von Grad zwei ein widersprüchliches Simplifikations-Verhalten beobachtet wurde, wird von der Verwendung dieses Kriteriums eher abgeraten. Möglicherweise wäre es besser, statt einer lokalen Betrachtung der Winkel einzelner Knoten ganze Kurven und Straßenabschnitte zu betrachten.

4.6 Laufzeiten

Verwendetes System:

Die Experimente wurden mithilfe eines Intel Celeron N2940 Quad-Core Prozessors mit 4 GB Arbeitsspeicher durchgeführt. Das verwendete Betriebssystem ist Lubuntu 15.10 (Kernel: 4.2.0-35).

Vorverarbeitung:

Wie bereits erwähnt, wurde zur Durchführung der Vorverarbeitung der `chconstructor` [9] verwendet. Hierzu wurde das vorgestellte Verfahren zur Festlegung der Kontraktionsreihenfolge eingebaut.

In Tabelle 3 sind für einige Kriterien die Gesamtzeit zur Durchführung der Vorverarbeitung angegeben. (Für die Kriterien wurden nur positive Gewichtungen verwendet.)

Kriterium	benötigte Zeit	maximales Level
nur Edge Difference	14,87 Minuten	104
nur Knotendichte (k=100)	10,51 Minuten	119
nur durchschnittliche Kantendist.	7,09 Minuten	149
(2,1)-Kombination (k=100)	22 Minuten	107

Tabelle 3: Laufzeiten (Vorverarbeitung)

Erstellung des simplifizierten Graphen

In den Tabellen 4 und 5 sind typische Laufzeiten für die Erstellung des simplifizierten Graphen angegeben. Genauer gesagt handelt es sich um die Zeit, die benötigt wird, um aus der Kantenmenge die entsprechenden Kanten des simplifizierten Graphen zu markieren. (Hierbei wurde nur ein Prozessorkern verwendet).

Die Ausgabe der Vorverarbeitung besitzt bei alleiniger Verwendung des Edge-Difference-Kriteriums 12,98 Mio. Kanten. Bei Verwendung der (2,1)-Kombination sind es 13,46 Mio. Kanten.

Wie zu sehen ist, nimmt die benötigte Zeit mit steigendem `minLevel` ab, wobei gleichzeitig auch die Anzahl der Knoten und Kanten des simplifizierten Graphen abnimmt. Für einen festen `minLevel`-Wert ist die Laufzeit geringer, je höher der `maxLength`-Wert ist.

In beiden Tabellen ist bei Erhöhung von `minLevel` von null auf eins eine Erhöhung der benötigten Zeit zu sehen, obwohl die Anzahl der Knoten und Kanten abnimmt. Dies ist darauf zurückzuführen, dass in beiden Fällen `maxLength = 0` ist. Bei `minLevel = 0` gibt

es keine Shortcuts, es werden also keine zu langen Shortcuts entpackt. Bei $\text{minLevel} = 1$ gibt es aber Shortcuts, die entpackt werden müssen. Dadurch ist der Zeitaufwand höher.

minLevel	maxLength	benötigte Zeit (in Sekunden)	Anzahl Knoten	Anzahl Kanten
0	0	0,77104	3,54 Mio.	7,18 Mio.
1	0	0,888131	3,31 Mio.	6,69 Mio.
2	0	0,861577	3,06 Mio.	6,15 Mio.
3	0	0,818645	2,77 Mio.	5,52 Mio.
4	0	0,780154	2,47 Mio.	4,88 Mio.
5	0	0,75864	2,20 Mio.	4,29 Mio.
10	0	0,60908	1,26 Mio.	2,36 Mio.
20	0	0,558814	0,69 Mio.	1,23 Mio.
30	0	0,512269	0,43 Mio.	0,73 Mio.
40	0	0,496034	0,29 Mio.	0,47 Mio.
50	0	0,465941	0,21 Mio.	0,32 Mio.
20	0,001	0,483677	0,38 Mio.	0,70 Mio.
20	0,002	0,439449	0,23 Mio.	0,42 Mio.
20	0,005	0,39597	0,096 Mio.	0,19 Mio.
20	0,01	0,367997	0,050 Mio.	0,10 Mio.
20	0,1	0,36096	5,5 Tsd.	26,6 Tsd.

Tabelle 4: Laufzeiten (Erstellung des simplifizierten Graphen): nur Edge-Difference

minLevel	maxLength	benötigte Zeit (in Sekunden)	Anzahl Knoten	Anzahl Kanten
0	0	0,805931	3,54 Mio.	7,18 Mio.
1	0	0,86984	3,43 Mio.	6,95 Mio.
2	0	0,867009	3,28 Mio.	6,62 Mio.
3	0	0,862219	3,08 Mio.	6,20 Mio.
4	0	0,849084	2,85 Mio.	5,70 Mio.
5	0	0,839228	2,60 Mio.	5,18 Mio.
10	0	0,761128	1,57 Mio.	3,01 Mio.
20	0	0,730202	0,75 Mio.	1,34 Mio.
30	0	0,623674	0,43 Mio.	0,74 Mio.
40	0	0,596491	0,30 Mio.	0,48 Mio.
50	0	0,560754	0,21 Mio.	0,33 Mio.
20	0,001	0,587286	0,40 Mio.	0,75 Mio.
20	0,002	0,504939	0,24 Mio.	0,46 Mio.
20	0,005	0,428086	0,11 Mio.	0,22 Mio.
20	0,01	0,393216	65,1 Tsd.	0,14 Mio.
20	0,1	0,37268	23,4 Tsd.	72,8 Tsd.

Tabelle 5: Laufzeiten (Erstellung des simplifizierten Graphen): (2,1)-Kombination
(k=100)

5 Zusammenfassung

Bei der Simplifizierung von Graphen mithilfe von Contraction Hierarchies lässt sich durch geeignete Wahl von `minLevel` und `maxLength` eine signifikante Reduzierung der Knoten- und Kantenanzahl erreichen.

Die Wahl der Kontraktionsreihenfolge hat dabei einen starken Einfluss auf das Aussehen des simplifizierten Graphen. Generell lässt sich sagen, dass durch eine hohe Gewichtung des Edge-Difference-Kriteriums die Anzahl der sichtbaren Shortcuts verringert wird. Durch Kombination mit dem Knotendichte-Kriterium kann eine stärkere Ausprägtheit von Städten erreicht werden, wodurch auch bei einer weiteren Verringerung der Knoten- und Kantenanzahl die charakteristische Form des Graphen erhalten bleibt.

Bei der Untersuchung der Gewichtung der Kriterien konnte festgestellt werden, dass auch das Vorzeichen der Gewichtung einen großen Einfluss auf die Simplifizierung hat. Beispielsweise führt eine positive Gewichtung des Knotendichte-Kriteriums dazu, dass Städte auch bei hohem `minLevel`-Wert gut sichtbar sind, während eine negative Gewichtung zu einer gleichmäßigeren Vereinfachung des Graphen führt.

Von den untersuchten Kriterien und deren Kombinationen ist die Kombination des Edge-Difference-Kriteriums (mit Gewichtung 2) mit dem Knotendichte-Kriterium (mit Gewichtung 1 und $k = 100$) den anderen Kombinationen vorzuziehen. Eine weitere Kombination mit anderen Kriterien führte entweder zu keinen oder zu minimalen Verbesserungen, hatte jedoch den Nachteil, dass eine deutliche Zunahme der sichtbaren Shortcuts stattfand.

Referenzen

- [1] W. Dörfler und W. Peschek, *Einführung in die Mathematik für Informatiker*. Hanser, 1988.
- [2] R. Geisberger, P. Sanders, D. Schultes und D. Delling, “Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks,” in *Experimental Algorithms*, pp. 319–333, Springer, 2008.
- [3] “Comparing Vector and Raster Mapping for Internet Applications.” An ESRI White Paper, August 2006.
- [4] M. Haklay und P. Weber, “Openstreetmap: User-Generated Street Maps,” *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.
- [5] J. Gaffuri, “Toward Web Mapping with Vector Data,” in *Geographic Information Science*, pp. 87–101, Springer, 2012.
- [6] <http://wiki.openstreetmap.org/wiki/Mapnik> (letzter Seitenzugriff: 11.05.2016).
- [7] A. Ruas, *Encyclopedia of GIS*, ch. Map Generalization, pp. 631–632. Boston, MA: Springer US, 2008.
- [8] K.S. Shea und R.B. McMaster, “Cartographic Generalization in a Digital Environment: When and How to Generalize,” in *Proceedings of AutoCarto*, vol. 9, pp. 56–67, 1989.
- [9] <https://theogit.fmi.uni-stuttgart.de/nusserae/chconstructor> (letzter Seitenzugriff: 11.05.2016).
- [10] <https://github.com/invor/simplestGraphRendering> (letzter Seitenzugriff: 11.05.2016).

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.

Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

(Ort, Datum, Unterschrift)