

Ableitung eines routingfähigen Bahnnetzes aus nutzergenerierten Gleisdaten (OpenStreetMap) durch Generalisierung

PAUL CZIOSKA¹, FRANK THIEMANN¹, ROBIN GIESE² & HERMANN VOGT²

Karten spielen eine wichtige Rolle bei der Informationskommunikation im Rahmen des öffentlichen Verkehrs, etwa als Liniennetzplan, auf dem die aktuelle Position des Verkehrsmittels dargestellt wird. Um solche Karten zu erstellen, ist es sinnvoll, die vorhandenen Gleisdaten zu einer repräsentativen Graphstruktur zu aggregieren. In dieser Arbeit wird dazu ein Algorithmus vorgestellt, der einen Liniendatensatz im zweidimensionalen Raum nach topologischen Eigenschaften zusammenfasst und als Ergebnis einen routingfähigen, ungerichteten Graphen mit Geokoordinaten erzeugt. Parallel verlaufende Linien werden in einer repräsentativen Geometrie gebündelt, die zugrunde liegende Topologie wird beibehalten. Der Implementierungsansatz nutzt hierzu Techniken aus dem Bereich der Skelettierung von Polygonen.

- 1) Leibniz Universität Hannover, Institut für Kartographie und Geoinformatik (IKG)
- 2) HaCon Ingenieurgesellschaft mbH, Hannover

1 Einleitung

Das System des öffentlichen Personenverkehrs hat eine herausragende Bedeutung für die Infrastruktur eines Landes– allein in Deutschland reisen mehr als 30 Millionen Fahrgäste täglich mit Bus und Bahn (DESTATIS [2013]). Ein so großes und gleichzeitig komplexes System benötigt daher eine effektive Informationsstruktur. Karten, insbesondere Liniennetzpläne, sind hierbei ein integraler Bestandteil dieser Informationen. Sie stellen in übersichtlicher Weise die Vernetzung bzw. Verknüpfung der Verkehrsmittel schematisch dar. Immer häufiger wird zusätzlich auch die tatsächlich befahrene Route gezeigt, so dass der Nutzer sie als Layer auf einer unterlegten Karte sehen kann. Eine Erweiterung von rein statischen Kartendarstellungen des Liniennetzplanes um die zeitliche Dimension ist eine *Live-Map*, auf der die aktuell verkehrende Züge oder Busse mit ihrer (interpolierten) Echtzeit-Position auf dem Liniennetz angezeigt werden. Ein Nutzer erhält auf diese Weise leicht verständliche Informationen darüber, wo sich sein Verkehrsmittel derzeit befindet. Ein prominentes Beispiel ist das Zugradar der Deutschen Bahn (Abb. 1).

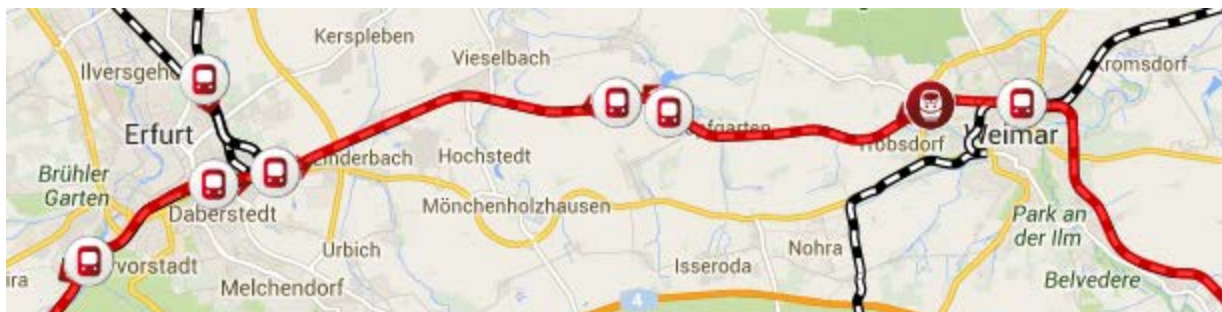


Abb. 1: Zugradar der Deutschen Bahn. Quelle: www.bahn.de/zugradar. Abgerufen am 06.01.2014

Für eine dynamische Darstellung der Zugposition und der befahrenen Route auf der Karte ist ein Routing auf dem Gleisnetz erforderlich. Eine Voraussetzung für die Anwendung von gängigen Routingalgorithmen ist, dass das zugrunde liegende Verkehrsnetz als Graph vorliegt. Ein Routing auf den Originaldaten ist zwar möglich, allerdings ist die Anzahl der Liniensegmente sehr hoch, wodurch der Routingalgorithmus verlangsamt wird.

Eine populäre Datenquelle für Schieneninfrastrukturdaten sind nutzergenerierte Geodaten, etwa aus dem Portal OpenStreetMap (OSM). Insbesondere bei solchen von Laien aufgenommenen Daten ist jedoch nicht sichergestellt, dass die Liniensegmente topologisch korrekt miteinander verknüpft sind, wie SCHMITZ ET AL. [2008] in ihrer Untersuchung feststellen. Die Daten können zwar für ein Routing verwendet werden, allerdings sind resultierende Fehler in der Routenwahl dadurch möglich.

Um diese Probleme zu lösen, wird in diesem Beitrag ein Algorithmus vorgestellt, der eine Menge an Linien im zweidimensionalen Raum zu einer repräsentativen Geometrie zusammenfasst. Dabei werden folgende Anforderungen gestellt:

- Parallel verlaufende Linien werden in einer repräsentativen Linie gebündelt, um eine Vereinfachung des Netzes zu erreichen und das Routing zu beschleunigen (Abb. 2a).
- Die zugrunde liegende Topologie der Originaldaten wird gewahrt, um bei Abbiegemöglichkeiten (beispielsweise eine Weiche im Schienennetz) das Routing über diese Stelle zu ermöglichen (Abb. 2b), jedoch bei Kreuzungen ohne Verknüpfung der Gleise auch im Graph keine Verbindung einzufügen (Abb. 2c).
- Das Ergebnis ist ein ungerichteter Graph aus Kanten und Knoten. Um ein Routing auf dem Verkehrsgraphen zu ermöglichen, werden punktförmige Daten (Bahnhöfe, Haltestellen etc.) als Start- und Zielknoten integriert (Abb. 2d).
- Kleinere topologische Fehler in den Originaldaten, wie fehlende Anschlüsse zweier Liniensegmente, werden behoben, sofern sie als solche erkannt werden.

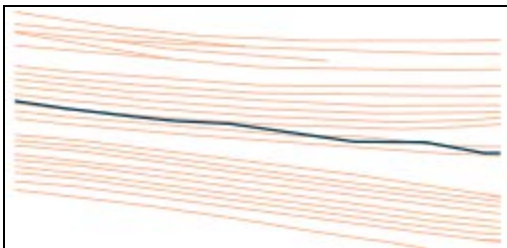


Abb. 2a: Parallel verlaufende Linien zusammenfassen

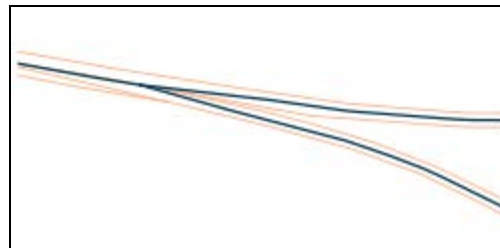


Abb. 2b: Abzweigungen mit einem Knoten verbinden

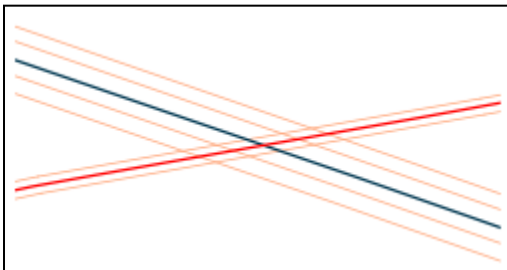


Abb. 2c: Kreuzungen nicht verbinden, d.h. kein Knoten einführen

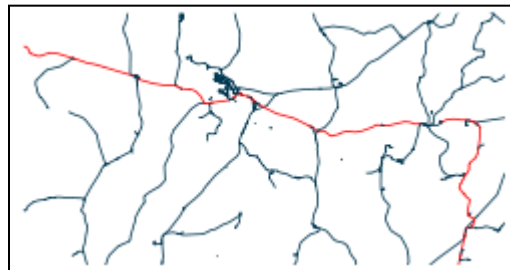


Abb. 2d: Integration punktförmiger Objekte

2 Verwandte Arbeiten

2.1 Ableitung eines Verkehrsgraphen

Die Ableitung eines routingfähigen Netzgraphen aus nutzergenerierten Geodaten ist in der Regel eine einfache Operation und in der Literatur gut dokumentiert, jedoch wird in den meisten Publikationen der Fokus auf Straßendaten gelegt, um ein Auto- bzw. Fußgängerouting zu ermöglichen, Bahnstrecken werden hingegen nicht thematisiert. REHRL ET AL [2012] stellen etwa ein Verfahren vor, mit dem ein Verkehrsgraph aus OpenStreetMap-Daten generiert werden kann, um ihn anschließend auf verschiedene Qualitätskriterien hin zu überprüfen. Dabei wird das gesamte Straßennetz als Grundlage für das Routing verwendet, es findet keine Generalisierung oder Korrektur von Topologiefehlern statt. In der Zukunft wird dieser Schritt voraussichtlich auch an Wichtigkeit verlieren, wie eine Untersuchung von NEIS ET AL. [2011] zeigt: Demnach ist die Anzahl der topologischen Fehler in den OSM-Daten in der Zeit zwischen 2007 und 2011 in Deutschland immer weiter zurückgegangen.

Eisenbahnverbindungen wurden allerdings bisher nicht systematisch analysiert und für ein Routing verwendet. Eine eigene Untersuchung kommt jedoch zu dem Schluss, dass auch die Bahnstrecken topologisch zu einem großen Teil korrekt verknüpft sind: In einem Testdatensatz wurden bei insgesamt 175499 Knoten lediglich 44 Fehler gefunden.

2.2 Kartographische Generalisierung

Im Kontext der kartographischen Generalisierung ist der in diesem Beitrag vorgestellte Algorithmus dem Teilbereich des Zusammenfassens (engl. „merging“) zuzuordnen. Für die Generalisierung von Liniennetzwerken aus Straßen oder Flüssen wird häufig das Prinzip der guten Fortführung (engl. „good continuation“) verwendet. Dabei wird ein Liniennetzwerk so ausgedünnt, dass die Winkel an Kreuzungen bzw. Knoten möglichst klein bleiben (vgl. THOMSON U. RICHARDSON [1999]). Diese Methode löst jedoch nicht das Problem der Zusammenfassung mehrerer paralleler Linien, so dass der Ansatz nicht für die Fragestellung dieses Beitrages verwendet werden kann.

2.3 Zusammenfassen von Trajektorien

Im Bereich der Analyse von Bewegungstrajektorien wurden verschiedene Methoden entwickelt, die sich grundsätzlich dazu eignen, mehrere parallel verlaufende Linien zu bündeln und einen Graph zu generieren. Eine Vielzahl von Algorithmen zum Zusammenfassen von Trajektorien basiert auf einer Clusterung der zugrunde liegenden Daten basierend auf ihrer Ähnlichkeit. Das Grundprinzip dabei ist, durch das Clustern Trajektorien zu identifizieren, die sich aufgrund ihrer räumlichen Lage und dem Bewegungsablauf ähnlich sind. WANG ET AL. [2006] stellen verschiedene Ähnlichkeitsmerkmale für Trajektorien vor, die als Basis einer Clusterung verwendet werden können. Sind mehrere Trajektorien als zusammengehörend identifiziert, kann ein Graph daraus abgeleitet werden. LEE ET AL. [2007] ziehen dafür auch Sub-Trajektorien in Betracht. Eine Methode von CAO U. KRUMM [2009] simuliert eine Gravitation zur Bündelung der Linien. In ZHANG ET AL [2010] wird ein Straßenmittelpunkt anhand von Profilen senkrecht zur genähert angenommenen Straße ermittelt. Im Gegensatz zum in diesem Beitrag vorgestellten Operator fokussieren die genannten Methoden allerdings nicht auf den Bahnverkehr.

3 Vorstellung des Operators

Der Grundgedanke des Zusammenfassungs-Operators ist die Verschmelzung von Linien-segmenten aufgrund ihrer räumlichen Nähe bei gleichzeitiger Wahrung der zugrunde liegenden Streckentopologie. Dazu wird eine Art morphologische Operation durchgeführt, indem die Gleise zunächst gepuffert (Dilatation) und anschließend wieder verdünnt werden (Erosion) – letzteres wird durch eine Skelettierung erreicht.

Der Algorithmus besteht aus 5 Schritten, die im Folgenden näher erläutert werden:

1. Erzeugung eines vereinigten Puffers aller Liniensegmente
2. Entfernung einiger Löcher des Gesamt-Polygons
3. Triangulation des Gesamt-Polygons
4. Einbeziehen der Originaldaten und Ableitung des Skeletts
5. Hinzufügen von Punktinformationen

Erzeugung eines vereinigten Puffers aller Liniensegmente

Zunächst wird von jedem Liniensegment ein Polygon durch eine Puffer-Operation gebildet. Der Pufferabstand ist manuell vom Benutzer zu definieren - je größer er gewählt wird, desto eher werden nahe beieinander liegende Liniensegmente vereint. Alle Puffer-Polygone werden anschließend zu einem großen Polygon vereinigt.

Entfernung der Löcher

In größeren Schienenkomplexen (etwa im Bahnhofsbereich) kann es dazu kommen, dass die gepufferten Einzelschienen nicht zu einem geschlossenen Polygon führen. Daher werden alle Löcher im Gesamtpolygon auf ihre Größe hin überprüft. Ist die Fläche kleiner als ein vom Anwender vorgegebener Schwellenwert, wird das Loch entfernt. Dieses Vorgehen ist wichtig, um eine Vielzahl von kleinen, eher unbedeutenden Löchern zu entfernen, die sonst die Form des Ergebnisses erheblich stören. Wird die Fläche allerdings zu groß gewählt, werden möglicherweise Kurvenradien oder Abstände zwischen zwei Strecken nicht mehr korrekt wiedergegeben, da zu viele Informationen verloren gehen. In der Praxis haben sich hier Werte zwischen 4000 und 10000 m² als sinnvoll erwiesen.

Triangulation des Gesamt-Polygons

Um aus den flächenhaften Daten wieder Linien zu generieren, wird eine Mittelachsen-Transformation des Gesamt-Polygons durchgeführt. Zur Berechnung einer Mittelachse im Vektorraum gibt es verschiedene Ansätze in der Literatur (vgl. HAUNERT & SESTER [2008]).

Eine Möglichkeit zur Approximation der Mittelachse basiert auf einer Triangulation der Polygonpunkte, wie sie von CHITHAMBARAM ET AL. [1991] vorgestellt wird. Die Mittelachse ergibt sich dabei durch die Verbindung der Mittelpunkte der Dreiecksseiten, welche keine ursprüngliche Polygonseiten darstellen. Diese Idee wird im folgenden aufgegriffen und für die speziellen Anforderungen adaptiert.

Zur Triangulation empfehlen PENNINGA ET AL. [2005] eine *Conforming Delaunay Triangulation* (vgl. etwa BERN U. EPPSTEIN [1992]), um den Verlauf des Skeletts zu glätten. Dabei werden zusätzliche Steiner-Punkte an den Polygonseiten eingefügt, um eine gleichförmige Gestalt der Dreiecke zu gewährleisten.

Erzeugung der Mittelachse durch Einbeziehen der Originaldaten

Die repräsentative Liniengeometrie wird nun aus dem triangulierten Polygon und den Originaldaten abgeleitet. Dazu werden alle Dreiecke iterativ durchlaufen und die folgenden Bearbeitungsschritte ausgeführt:

1. Suche der Linien im Originaldatensatz im Bereich des Dreiecks
2. Korrektur von offensichtlichen Topologiefehlern
3. Konnektivitätsanalyse der Linien innerhalb des Dreiecks
4. Einfügen von Verbindungslinien zwischen den Dreieckskanten
5. Optional: Rekursive Streckenverfolgung

Um Kreuzungen sicher von Abzweigungen unterscheiden zu können, ist es erforderlich, die Strecken (Gleisstränge mit gleichem Start- und Zielpunkt) über eine weitere Entfernung hinweg zu verfolgen, um zu entscheiden, ob sie sich an einem Punkt vereinigen (Abzweigung, Abb. 3a) oder ohne gemeinsamen Knotenpunkt wieder trennen (Kreuzung, Abb. 3b).

Wird während der Iteration ein Dreieck mit drei innenliegenden Kanten gefunden und wird jede der drei Kanten von mindestens einem Gleis durchquert, ist dieses Dreieck der Startpunkt einer nun folgenden Rekursion (Grüne Dreiecke 1 in Abb. 3). Im Normalfall werden zwei Kanten des Dreiecks als Eingang von Strecken betrachtet, die dritte Kante, in der beide Strecken weiterlaufen, stellt den Ausgang dar. Anhand dieser Ausgangskante wird nun das nächste Dreieck gesucht, welches sich anschließt. Wird in diesem Dreieck noch keine Verbindung zwischen den verschiedenen Strecken gefunden, wird die rekursive Funktion erneut mit dem nächsten Dreieck aufgerufen. Sobald in einem Dreieck eine Verbindung festgestellt wird (Abb. 3a), werden die beiden Strecken miteinander verbunden und die Rekursion endet. Verlassen beide Strecken das Dreieck durch unterschiedliche Kanten (Abb. 3b), wird von einer Kreuzung ausgegangen. Dieses Konzept kann prinzipiell mit beliebig vielen Strecken gleichzeitig durchgeführt werden. Neben den vorgestellten Standardfällen gibt es allerdings noch weitere Konstellationen (Sonderfälle), die hier nicht näher beschrieben werden, wie z.B. Ringe oder Kollisionen.

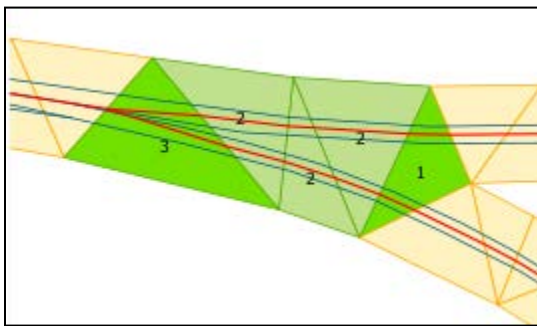


Abb. 3a: Vereinigung zweier Strecken.

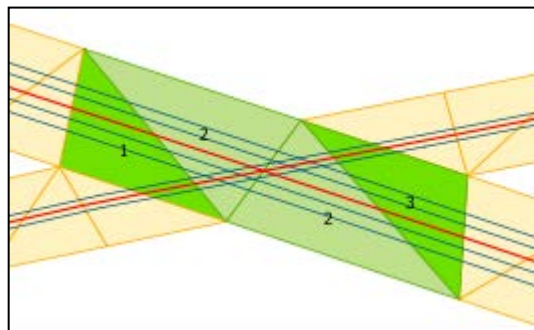


Abb. 3b: Vereinigung zweier Strecken.

Legende

Blau: Gleisdaten - Rot: Realgraph

Grüne Dreiecke 1: Strecken treffen an unterschiedlichen Kanten ein

Grüne Dreiecke 2: Strecken werden verfolgt - noch keine Vereinigung

Grüne Dreiecke 3: Strecken vereinigen oder trennen sich (Kreuzung)

4 Ergebnisse

4.1 Beispiele

Die folgenden Beispiele zeigen die Ergebnisse eines Testlaufs mit allen Gleisdaten Deutschlands aus dem OpenStreetMap-Datensatz. Abbildung 4 zeigt den Bahnhof Osnabrück mit dem Verkehrsgraph in Rot und unterlegter OpenStreetMap-Karte, die Abbildungen 5 und 6 zeigen Ausschnitte aus dem Ergebnisdatensatz – die blaue Linie stellt hier die repräsentative Linie dar, die gelben Linien die originalen Gleisdaten. In Abbildung 7 wird gezeigt, dass an Kreuzungen topologisch korrekt kein Knotenpunkt eingefügt wurde.

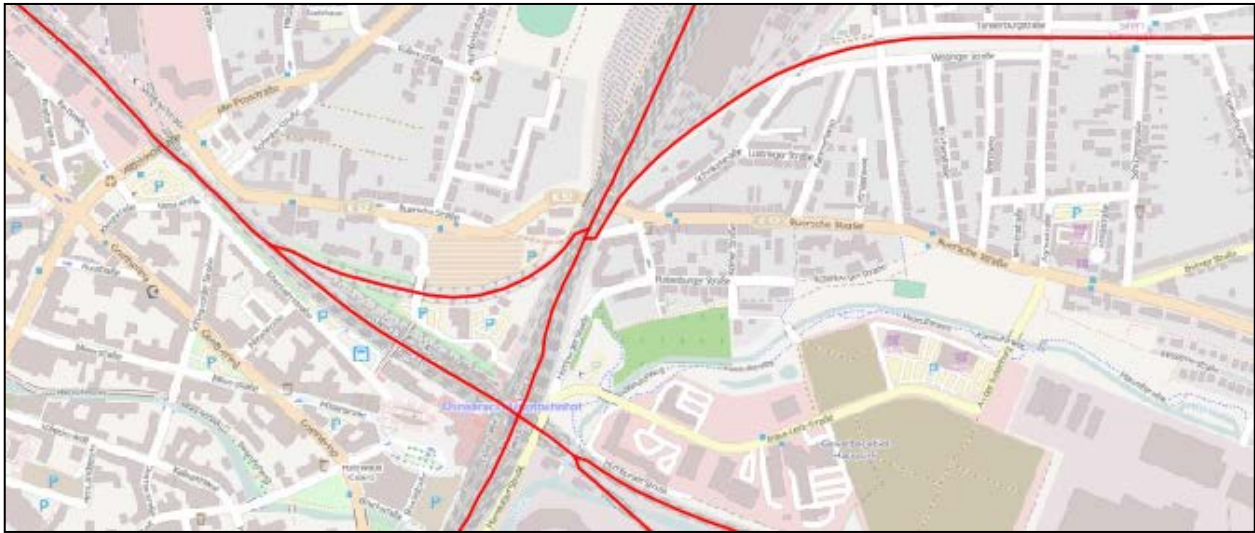


Abb. 4: Zusammengefasste Linien über OpenStreetMap-Karte, Beispiel Osnabrück

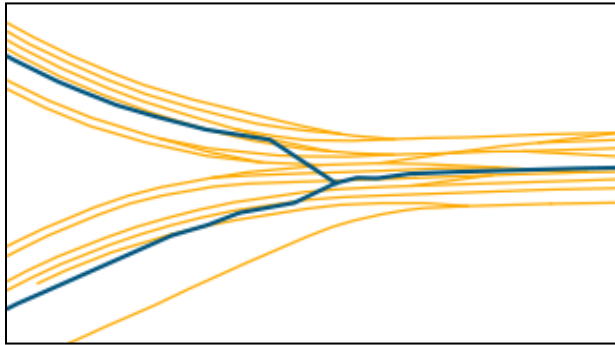


Abb. 5: Abzweigung. Beispiel Hameln

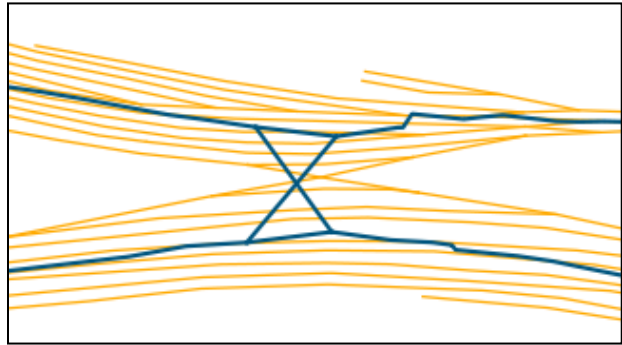


Abb. 6: Mehrere Abzweigungen. Beispiel Kreiensen

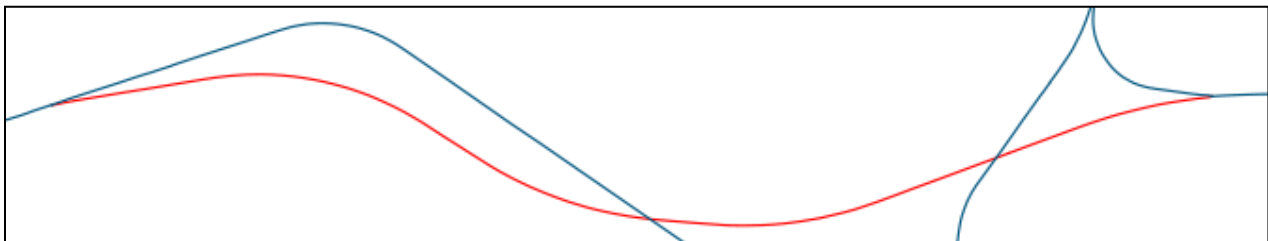


Abb. 7: Kreuzungen ohne Übergang. Beispiel Schnellfahrstrecke Hannover-Göttingen

4.2 Evaluierung

Um die Eignung des Ergebnisses im Hinblick auf die Routingfähigkeit und topologische Korrektheit zu überprüfen, wurden mehrere Zugfahrten auf einem Testdatensatz durch ein Routing simuliert. Als Basis wurden folgende Daten verwendet:

- Gleisdaten aus OpenStreetMap (Bereich Norddeutschland)
- Bahnhofskordinaten aus OpenStreetMap (Bereich Norddeutschland)
- Fahrplandaten aus www.bahn.de und www.der-metronom.de

Aufgrund von Datenschutzbestimmungen seitens der Deutschen Bahn konnte nicht automatisiert auf eine Vielzahl von Verbindungen zugegriffen werden, so dass zum Testen vier Verbindungen in Norddeutschland ausgewählt wurden.

Die Auswertung des Routings erfolgte einerseits visuell durch Abgleich der gerouteten Strecke mit der realen Fahrtroute und andererseits automatisch durch eine Überprüfung der benötigten Fahrzeit. Gibt es topologische Fehler in den Daten, so dass der Zug einen großen Umweg nehmen müsste, steigt die berechnete Geschwindigkeit des Fahrzeugs an, da eine wesentlich weitere Strecke in der gleichen Zeit gefahren werden muss. Sobald eine unrealistische Geschwindigkeit im Fahrtverlauf auftritt, wird eine entsprechende Warnung ausgegeben.

Der visuelle Vergleich zeigte keinerlei Auffälligkeiten, auch der Test der benötigten Geschwindigkeiten verlief unauffällig, da alle Züge die normale Route nehmen konnten.

4.3 Untersuchungen zur Skalierbarkeit

Um die mittlere Laufzeit experimentell zu bestimmen, wurde der Zeitbedarf des Algorithmus bei unterschiedlich großen Datenmengen bestimmt. Gleichzeitig wurde das Maximum der Speicherplatzbelegung bei jeder Berechnung erfasst und aufgezeichnet. Das Ergebnis ist in Abbildung 8 gezeigt. Anhand der Kurvenform kann von einer linearen Zeitkomplexität ausgegangen werden.

Eine Berechnung des Bahn-Skeletts von Norddeutschland hatte in der Testberechnung einen Zeitbedarf von etwa 15-30 Minuten, für ganz Deutschland wurden etwa 1,5 Stunden benötigt.

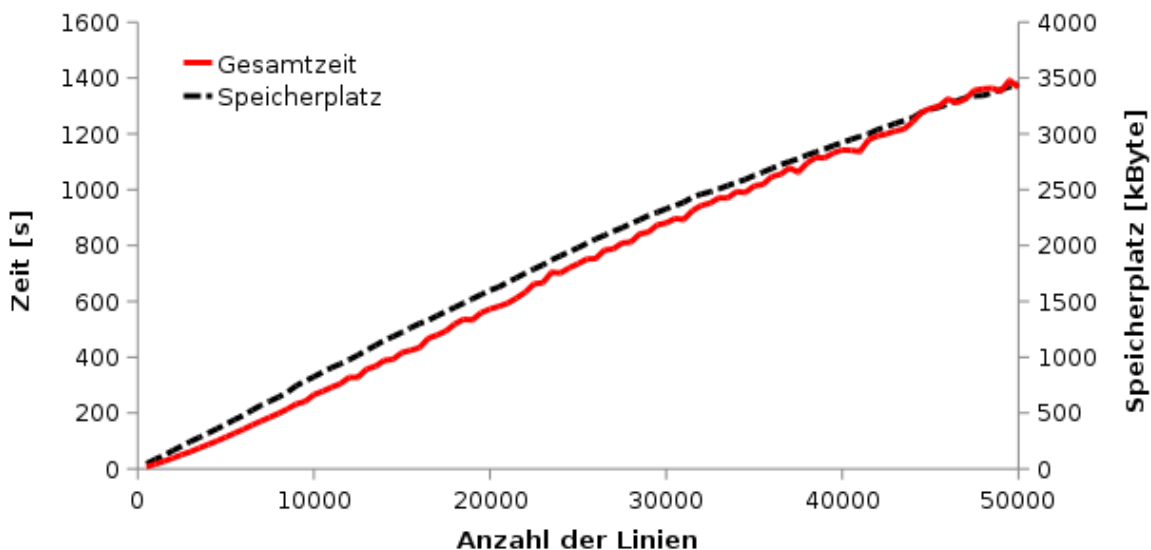


Abb. 8: Zeit- und Speicherkomplexität des Algorithmus

5 Fazit

In diesem Beitrag wurde eine Methode vorgestellt, um aus Liniendaten einen repräsentativen Graphen mit Geokoordinaten unter Berücksichtigung der Netztopologie der Originaldaten abzuleiten. Der Operator verwendet dabei Methoden der Triangulation und des Geometriypwechsels (Dimensional Collapse). Kern des Algorithmus ist die rekursive Verfolgung von Strecken auf der Basis von Dreiecken.

Die Ergebnisse des Algorithmus sind topologisch und visuell zu einem sehr großen Anteil korrekt, auch wenn Fehler nicht ausgeschlossen werden können. Zur Evaluierung der Ergebnisse wurde ein Testdatensatz durch ein Routing mit realen Fahrplandaten in seiner Topologie bestätigt. Die Laufzeit des Operators befindet sich in einem akzeptablem Bereich - sowohl die Zeit- als auch die Speicherkomplexität ist linear, so dass auch größere Datensätze mit begrenztem Zeit- und Ressourcenverbrauch berechnet werden können.

Ein wesentlicher Nachteil des Algorithmus ist seine Richtungs- bzw. Reihenfolgeabhängigkeit, d.h. bei verschiedenen Eingabedaten kann es auch in Bereichen, die kongruent sind, zu unterschiedlichen Ergebnissen kommen. Dadurch ist eine kachelbasierte Aufteilung der Daten zur Parallelisierung der Prozesse (und einer damit verbundenen Beschleunigung der Rechenzeit) nicht möglich. Hierfür sind Ansätze zur realisieren, die mit überlappenden Partitionierungen arbeiten (vgl. etwa THIEMANN ET AL. [2013]).

6 Literaturverzeichnis

BERN, M. & EPPSTEIN, D., 1992: Mesh generation and optimal triangulation. In: Computing in Euclidean geometry (1), S. 23–90

CAO, L. & KRUMM, J., 2009: From GPS traces to a routable road map. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, S. 3–12

CHITHAMBARAM, R.; BEARD, K. & BARRERA, R., 1991: Skeletonizing polygons for map generalization. In: Technical papers, ACSM-ASPRS Convention, Cartography and GIS/LIS (2), S. 44–54

DESTATIS (2013): Verkehrsmittelbestand und Infrastruktur.

<https://www.destatis.de/DE/ZahlenFakten/Wirtschaftsbereiche/TransportVerkehr/UnternehmenInfrastrukturFahrzeugbestand/Tabellen/Verkehrsinfrastruktur.html>

Version: August 2013. – Abgerufen am 12.08.2013

HAUNERT, J. H. & SESTER, M., 2008: Area collapse and road centerlines based on straight skeletons. In: Geoinformatica, 12(2), 169-191.

LEE, J.; HAN, J. & WHANG, K., 2007: Trajectory clustering: a partition-and-group framework. In: Proceedings of the 2007 ACM SIGMOD international conference on Management of data ACM, S. 593–604

NEIS, P.; ZIELSTRA, D. & ZIPF, A., 2011: The Street network evolution of crowdsourced maps: OpenStreetMap in Germany 2007–2011. In: Future Internet 4 (1), S. 1–21

PENNINGA, F.; VERBREE, E.; QUAK, W. & OOSTEROM, P., 2005: Construction of the planar partition postal code map based on cadastral registration. In: GeoInformatica 9 (2), S. 181–204

REHRL, K.; EDLINGER, K. M.; FRIEDWAGNER, A.; HAHN, B.; LANGTHALER, T.; WAGNER, A. & WIMMER, M., 2012: Evaluierung von Verkehrsgraphen für die Berechnung von länderübergreifenden Erreichbarkeitspotenzialen am Beispiel von OpenStreetMap. In : Strobel, J. et al.(Hg.), 2012: Angewandte Geoinformatik.

SCHMITZ, S.; ZIPF, A. & NEIS, P., 2008: New Applications based on collaborative geodata – the case of Routing. In: XXVIII INCA international congress on collaborative mapping and space technology, Gandhinagar, India, 2008.

THIEMANN, F.; WERDER, S.; GLOBIG, T.; SESTER, M., 2013: Investigations into partitioning of generalization processes in a distributed processing framework. In: 26. International Cartographic Conference. 25.-30.8.2013. Dresden: International Cartographic Association, 2013. http://www.icc2013.org/contxt/medien/upload/proceeding/395_proceeding.pdf

THOMSON, R. C., & RICHARDSON, D. E., 1999: The ‘good continuation’ principle of perceptual organization applied to the generalization of road networks.

WANG, X.; TIEU, K.; GRIMSON, E., 2006: Learning semantic scene models by trajectory analysis. In: Computer Vision–ECCV, S. 110–123.

ZHANG, L., F. THIEMANN AND M. SESTER, 2010: Integration of GPS Traces with Road Map, Proceedings of the Workshop on Computational Transportation Science in conjunction with ACM SIGSPATIAL, San Jose, USA, 2010.