

Bachelorarbeit  
Automatische Erkennung  
von Gleisfeldern in Schienennetzen

Patrick Schulz

9. Dezember 2014

Matrikelnummer: 944808

Erstprüfer: Prof. Dr.-Ing. Jan-Henrik Haunert

Zweitprüfer: Johannes Ohrlein, M. Sc.



## Zusammenfassung

Es wird ein graphentheoretisches Verfahren vorgestellt zur automatischen Erfassung von Gleisfeldern, welches für die struktursuchende Generalisierung von Schienennetzen eine entscheidende Rolle spielt.

## Abstract

This paper introduces a graph-theoretical procedure how to identify fields in a railroad network, which is a significant method for the structure examined generalization of those networks.

**Keywords.** generalization, selection, railroad, network, graph theory

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Bestehende Arbeiten</b>	<b>4</b>
<b>3</b>	<b>Aufgabenstellung und Szenarien</b>	<b>9</b>
<b>4</b>	<b>Grundlagen / Voraussetzungen</b>	<b>10</b>
<b>5</b>	<b>Die Gleisfeldsuche</b>	<b>11</b>
5.1	Generierung des Gleisfeldgraphen $G_{\text{Gleisfeld}}$	11
5.2	Ergänzungen des Gleisfeldgraphen $G'_{\text{Gleisfeld}}$	12
5.3	Identifizierung der Randfeatures	13
5.4	Kursgraph $G_{\text{Kurs}}$	13
5.5	Die Wurzelsuche	14
5.5.1	Tokenverteilung durch Breitensuche	14
5.5.2	Auswertung der Tokenverteilung	16
<b>6</b>	<b>Implementierung</b>	<b>17</b>
<b>7</b>	<b>Experimente</b>	<b>18</b>
7.1	Testgebiet Porrentruy	19
7.2	Testgebiet Pratteln	20
7.3	Testgebiet Solothurn	21
<b>8</b>	<b>Fazit</b>	<b>21</b>
	<b>Literatur</b>	<b>22</b>
	<b>Abbildungsverzeichnis</b>	<b>22</b>
	<b>Anlagen</b>	<b>24</b>

# 1 Einleitung

Diese Bachelorarbeit befasst sich mit der Generalisierung von Schienennetzen. Unter Generalisierung versteht man einen Prozess, in dem der Inhalt einer Karte sinnvoll auf eine kleinere Fläche projiziert wird. Es gibt eine Vielzahl von Operatoren, die für die Generalisierung zur Anwendung kommen – darunter fallen die Geometrievereinfachung oder die Verdrängung zwischen den Geometrien, um gewisse Mindestabstände einhalten zu können. In dieser Arbeit wird allerdings nur der Selektion-Operator behandelt. Durch die Selektion wird bestimmt welche Gleise für die kleinere Karte übernommen werden sollen, um die Übersichtlichkeit in der kleineren Karte zu bewahren.

Aufgabensteller und Bereitsteller der Daten ist das Schweizer Bundesamt für Landestopografie (swisstopo) mit der Erwartung, geeignete Generalisierungs-Algorithmen für das Schienennetz der Schweiz zu finden, da dieser Arbeitsschritt bei swisstopo noch von Hand durchgeführt wird. Die Idee zur Identifizierung der Gleisfelder gaben indirekt die Ansprechpartner und Kartographen der swisstopo in Form einer Anforderungsanalyse weiter, die einen erheblichen Fokus auf Gleisfelder besitzt. Gleisfelder sind Schienenabschnitte, in denen mehrere Gleise mit einem geringen Abstand zueinander parallel verlaufen. Diese Strukturen besitzen eine hohe Dichte an Informationen, die bei einer Generalisierung somit den höchsten Bedarf an Informationsselektion fordert. Die Abbildung 1 zeigt ein solches Gleisfeld.



Abbildung 1: Beispielnetz bestehend aus einem einzigen Gleisfeld

Aufbauend für eine struktursuchende Generalisierung des Schienennetzes erscheint dieser Prozess des *Data Enrichments* als essentiell, denn nachdem man die Gleisfelder identifiziert hat, ist man in der Lage die Gleisfelder separat nach den Generalisierungsanforderungen zu bearbeiten. Die Abbildung 2 zeigt eine solche Bearbeitung eines Gleisfeldes, in der durch gegebene Bedingung entschieden werden muss, ob das Gleisfeld aufgrund seiner Größe nur ausgedünnt wird – und somit grob seine Charakteristik beibehält – oder nur eine einzige Linie beibehalten wird.

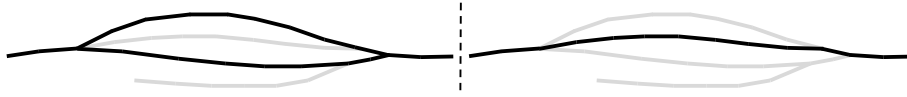


Abbildung 2: Mögliche Generalisierungsergebnisse für Abbildung 1; Links, wenn das Gleisfeld nach den Generalisierungsanforderungen existent ist, ansonsten Rechts

## 2 Bestehende Arbeiten

Es existieren bereits in der Literatur Ansätze zum Selektionsproblem von Straßen- und Flussnetzen, allerdings ist die Anwendbarkeit auf Schienennetze noch nicht behandelt worden. Die Ansprüche auf ein zusammenhängendes Netz und Wiedererkennbarkeit nach der Generalisierung sind Anforderungen, die für alle Netze gelten. Die Eigenschaften, welche das Schienennetz besonders machen, wurden bereits von Mark (1989) festgehalten:

”Railways have an even more constrained geometry than roads, since tight bends are never constructed, and gradients must be very low. Such smoothness should be preserved during generalization, even if curves must be exaggerated in order to achieve this.”

Der geläufige Prozess der Selektion für die Generalisierung in Netzwerken lässt sich in drei grobe Schritte unterteilen: Die Vorprozessierung, die Gewichtung der Kanten und die abschließende Selektionsstrategie. Unter der Vorprozessierung sind jegliche Ansätze unterzuordnen, die noch vor der eigentlichen Gewichtung stattfinden. Diese Ansätze selektieren z.B. im Vorfeld jegliche Kanten aus dem Netzwerk, um das Netzwerk auf eine gewisse Weise zu bereinigen. Ein anderer Ansatz bündelt mehrere Kanten zusammen, um die Kontinuität von Straßenzügen zu bewahren – der sogenannte Stroke-Ansatz (Thomson und Richardson, 1999). Die Bewahrung solcher Straßenzüge ist ein natürlicher, kognitiver Wunsch der menschlichen Wahrnehmung. Solch gebündelte Kanten erhalten dann bei der Gewichtung ein gemeinsames Gewicht und es können keine Teilmengen solcher Bündelungen selektiert werden – entweder ganz oder gar nicht. Für die Ermittlung der *Strokes* wird an jedem Knotenpunkt von einem Liniensegment überprüft, welches benachbarte Liniensegment den Linienverlauf am besten fortführt; ermittelt wird dies primär durch die Betrachtung des Schnittwinkels. Zwei Liniensegmente werden zu einem Stroke zusammengefasst, wenn beide Liniensegmente bei der vorhin erklärten Kontinuitätsprobe den jeweils Anderen gewählt haben. Die Abbildung 3 verdeutlicht die Ergebnisse, die beim Stroke-Ansatz entstehen sollten, wenn nur das Winkelkriterium in Betracht gezogen wird. Neben dem Winkelkriterium können zusätzlich in einem Straßennetz auch der Straßentyp und der Straßename in die Kontinuitätsprobe mit einbezogen werden.

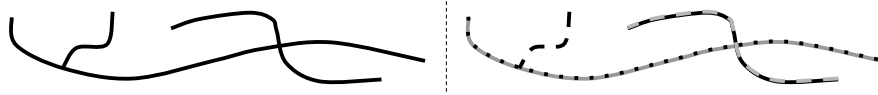


Abbildung 3: Links Beispielstraßennetz, Rechts die Strokes des Straßennetzes.

Auf Grundlage der Erkenntnisse von David M. Mark kann man behaupten, dass der Stroke-Ansatz aufgrund der Kontinuitätsbewahrung ein geeigneter Ansatz ist, der bei der Generalisierung von Schienennetzen zur Anwendung kommen sollte. Ein guter Anwendungsfall der Strokes existiert bereits seit mehreren Jahrzehnten noch vor der Veröffentlichung des Stroke-Ansatzes – Hierbei ist die Rede vom Hortonschen Ordnungssystem (Horton, 1945). Das Hortonsche Ordnungssystem ist ein klassischer Ansatz zur Unterteilung und Gewichtung von Flüssen in einem Flussnetz. Zur Ermittlung der zusammenhängenden Flüsse wurden schon damals die selben Methoden verwendet, wie man sie vom Stroke-Ansatz kennt. Jedes Flussegment erhält dabei eine Ordnungszahl, die die Wichtigkeit relativ zum gesamten Flussnetz indiziert. Alle Segmente des Hauptflusses erhalten hierbei die Ordnungszahl 1 und die Segmente aller Nebenflüsse, die direkt in den Hauptfluss münden, erhalten die nächsthöhere Ordnungszahl (in diesem Fall: 2). Diese Zuordnung wird rekursiv für jeden weiteren Nebenfluss durchgeführt. Die dadurch entstandene Hierarchie ist in der Abbildung 4 veranschaulicht. Aufbauend auf der entstandenen Hierarchie kann das Flussnetz nun iterativ abgebaut werden, beginnend bei den Flussegmenten mit der höchsten Ordnungszahl. Auf diese Weise ist gewährleistet, dass die Hauptflüsse bis zu ihren Quellen bestehen bleiben. Das Hortonsche Ordnungssystem von Robert E. Horton gilt bis heute als das meistgenutzte Ordnungssystem zur hierarchischen Unterteilung eines Flussnetzes.

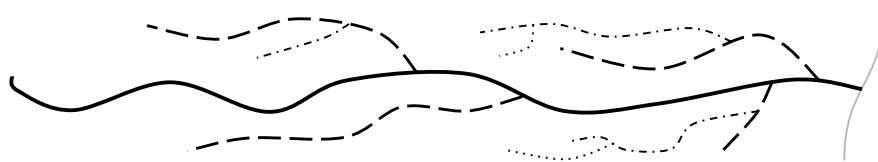


Abbildung 4: Hortonsches Ordnungssystem: Ordnungszahl der einzelnen Flüsse ist durch die Strichlinienart gekennzeichnet: durchgezogen → 1, gestrichelt → 2, punkt-gestrichelt → 3, gepunktet → 4

Neben dem Hortonschen Ordnungssystem gibt es eine ganze Reihe an Möglichkeiten seine Kanten im Netz zu gewichten, um die Wichtigkeit einer bestimmten Kante in Relation zu den anderen Kanten zu verdeutlichen. Eine offensichtliche Variante ist die Einbeziehung der Linielänge. Auch andere bestehende Eigenschaften, wie der Straßentyp oder die Verkehrsintensität, wie viele Menschen die Verkehrsverbindung im

Durchschnitt nutzen, sind geeignete Gewichtungsgrundlagen. Die Ermittlung der Verkehrsintensität kann über einen längeren Zeitraum empirisch ermittelt werden oder man setzt auf Verfahren, wie die Betweenness-Centrality (Knoke und Burt, 1983) oder dem PageRank [(Page et al., 1999)].

Der PageRank ist ein von Google-Mitgründer Larry Page entwickelter Algorithmus zur Gewichtung von Webseiten auf Basis von Referenzen zwischen den einzelnen Webseiten. Dieses Prinzip kann auch für die Gewichtung von Straßen herangezogen werden: Aus einem gegebenen Straßennetz lassen sich die Konnektivitäten der einzelnen Straßensegmente zueinander in einem Nachbarschaftsgraphen verdeutlichen. Jeder Knoten des Nachbarschaftsgraphen erhält ein Startgewicht. Der PageRank lässt sich nach mehreren Iterationen der folgenden Arbeitsschritte ermitteln. Jeder Knoten verteilt einen Großteil seines Gewichts an seine Nachbarn – der genaue Anteil ist durch den Dämpfungsfaktor  $D$  festgelegt. Das verteilte Gewicht wird gleichmäßig und unabhängig von allen anderen Grapheneigenschaften auf die Nachbarknoten aufgeteilt.

Ein Beispiel: Ein Knoten  $P$  hat ein Gewicht von 10 Einheiten und besitzt 3 Nachbarknoten. Bei einem Dämpfungsfaktor  $D = 0.9$  muss der Knoten  $P$  9 Einheiten seines Gewichtes an seine Nachbarn abgeben. Jeder Nachbar erhält genau 3 Einheiten von Punkt  $P$ . Dieses Ereignis findet in einer Iteration simultan bei allen Punkten statt, daher wird das in der selben Iteration erhaltene Gewicht erst in der nächsten Iteration weiterverteilt. Für die weitere Verständlichkeit des Verfahrens folgt eine tabellarische Berechnung vom PageRank (Tabelle 1) für die kleine Instanz in der Abbildung 5.

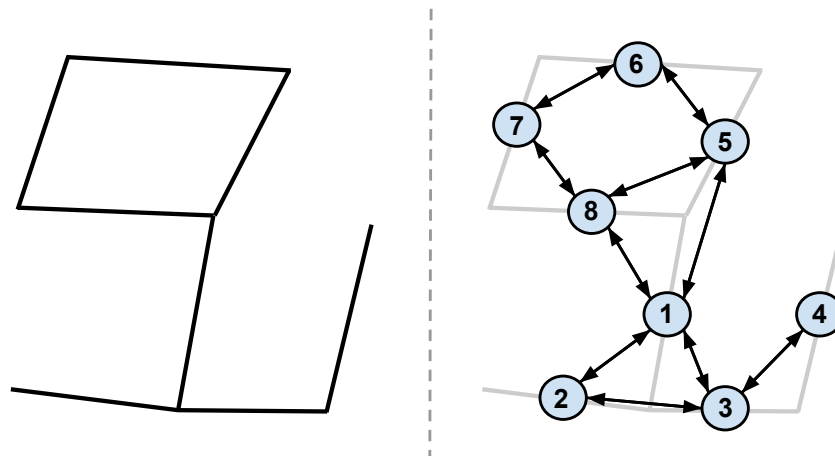


Abbildung 5: Links das gegebene Straßennetz, rechts der daraus generierte Nachbarschaftsgraph.

Knoten Startgewicht	1 10	2 10	3 10	4 10	5 10	6 10	7 10	8 10
Vert. Knoten 1	5,00	1,25	1,25		1,25			1,25
Knoten 2	2,50	5,00	2,50					
Knoten 3	1,67	1,67	5,00	1,67				
Knoten 4			5,00	5,00				
Knoten 5	1,67				5,00	1,67		1,67
Knoten 6					2,50	5,00	2,50	
Knoten 7						2,50	5,00	2,50
Knoten 8	1,67				1,67		1,67	5,00
Iteration 1	12,50	7,92	13,75	6,67	10,42	9,17	9,17	10,42
Iteration 10	16,05	8,16	12,37	4,18	11,80	7,82	7,82	11,80

Tabelle 1: Tabellarische Berechnung der ersten Iteration vom PageRank für den Graphen in Abbildung 5 mit einem Dämpfungsfaktor  $D = 0.5$ .

Nachdem alle bestehenden Kanten im Graphen gewichtet sind, steht die Wahl der Selektionsstrategie an. Von einem trivialen Problem ist hier keines Falls die Rede: Würde man versuchen nur die  $n$ -teuersten Kanten zu selektieren, wird man mit einer sehr hohen Wahrscheinlichkeit kein zusammenhängendes Netzwerk erhalten. Die Selektion könnte man als Optimierungsproblem formulieren und einem mathematischen Solver die Entscheidung überlassen. Diese Herangehensweise ist allerdings nur für kleine Instanzen effizient lösbar. Als Alternative bietet sich der *GreedySelection-Algorithmus* an, der in der Theorie schlechte Ergebnisse liefern kann im Vergleich zum Optimum, in der Praxis an realen Instanzen aber anwendbar ist (Chimani et al., 2014). Der Algorithmus entfernt bzw. *isst* mit jeder Iteration eine Kante aus dem Graphen – bevorzugt Kanten mit dem geringsten Gewicht, achtet aber gleichzeitig darauf, dass der gesamte Graph nicht in zwei gespaltene Mengen aufgeteilt wird. Auf diese Weise versucht der Algorithmus die Summe der Gewichte aller verbliebenen Kanten nach jeder Iteration maximal zu halten. In der Abbildung 6 ist das Verhalten vom GreedySelction-Algorithmus nochmal veranschaulicht. Im nächsten Iterationsschritt würde in der Abbildung 6 der Knoten mit dem Gewicht 7 wegfallen. Der Prozess der Entfernung der Kanten wird so oft wiederholt, bis die Ausdünnung des Netzwerkes zufriedenstellend ist. Als Leitwert für die Anzahl der Kanten, die im Graphen übrig bleiben sollen, kommt das *Wurzelgesetz* zur Anwendung (Töpfer und Pillewizer, 1966).

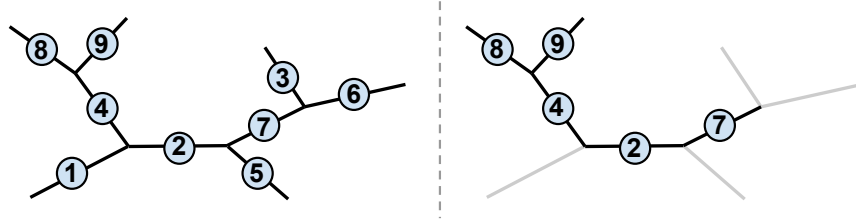


Abbildung 6: Links der gegebene Graph mit Gewichtung, rechts der selbe Graph nach der vierten Iteration vom *GreedySelection*-Algorithmus.

Friedrich Töpfer erkannte in den 60er Jahren einen gewissen Zusammenhang, was die Anzahl der selektierten Objekte in einer generalisierten Karte betrifft. Er formulierte den für sich erkannten Zusammenhang in eine Gleichung, die als “the Radical Law” bzw. als das Wurzelgesetz publiziert wurde. Diese Gleichung liefert für viele Generalisierungsalgorithmen einen akzeptablen Leitwert für die Anzahl der zu selektierenden Objekte.

$$n' = n * \sqrt{s/s'} \quad (1)$$

Die Variable  $n$  beschreibt die Anzahl der Features, die auf der Karte dargestellt werden; die Variable  $s$  steht für den Kartenmaßstab. Wenn man nun  $s'$  einen Wert zuweist, welcher den Maßstab der generalisierten Karten repräsentiert, lässt sich die ungefähre Anzahl der Features herleiten, die in der generalisierten Karte angezeigt werden sollen.

Ein anderer Ansatz, den man zu den Vorprozessierungs Ansätzen zählen kann, versucht Hauptumschlagpunkte in Straßennetzen zu erkennen, um diese danach zu vereinfachen (Mackaness und Mackechnie, 1999). Bei den Hauptumschlagpunkten kann es sich um Strukturen beliebigen Maßstabes handeln: Ein Kreisverkehr, der zu einer einfachen Kreuzung konvertiert wird, oder ganze Ballungsräume, in denen danach alle eingehenden Hauptstraßen an nur wenigen Stellen aufeinander treffen.

Als umfassende Ausarbeitung zur Generalisierung von Straßennetzen ist die Masterarbeit von Benz (2013) von der Universität Zürich zu empfehlen: Diese geht nochmals sehr ausführlich auf die bestehenden Ansätze der Generalisierung von Netzwerken ein.



### 3 Aufgabenstellung und Szenarien

Gegeben sind die Rohdaten vom Schienennetz in Form von auf Vektoren basierenden Linien; die Linien repräsentieren im Datenbestand die geometrische Lage der Gleise. Gesucht sind Teilmengen der Linien aus dem Datenbestand, die jeweils im Kontext der anderen Linien ein Gleisfeld darstellen. Zusätzlich kann man weitere Eigenschaften der Gleisfelder herausfinden, wie die Wurzeln eines Gleisfeldes: Unter der Wurzel eines Gleisfeldes versteht man einen Gleisabschnitt, in den idealerweise alle anderen Gleise des Gleisfeldes münden. Dieses Ideal, in dem alle Gleisfelder auf beiden Seiten jeweils nur eine Wurzel besitzen wie in der Abbildung 7, ist im Datenbestand allerdings nicht durchgehend vorhanden. Im Falle, dass einzelne Gleise vom Gleisfeld abdriften (d.h. sie verändern ihre Ausrichtung), existieren anstelle der Wurzel dann nur mehrere Subwurzeln, in die jeweils nur ein Bruchteil der Gleise vom Gleisfeld münden, die in ihrer Summe aber alle Gleise des Gleisfeldes abdecken (siehe Abbildung 8). Es kann außerdem der Fall eintreten, dass gar keine Wurzeln auf einer Seite entstehen können, wenn jedes Gleis auf der besagten Seite des Gleisfeldes in eine Sackgasse führt.



Abbildung 7: Beispielgleisfeld; Die Punkte sind die Wurzeln des Gleisfeldes.

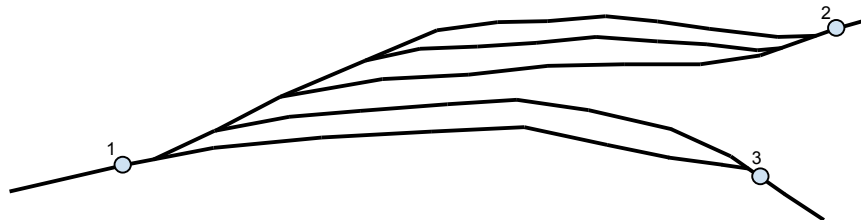


Abbildung 8: Beispielgleisfeld, wo die zwei unteren Gleise vom Rest abdriften und eine eigene Subwurzel (3) bilden. Der Rest sammelt sich an der anderen Subwurzel (2)

In den darauffolgenden Kapiteln werden die einzelnen Schritte beschrieben, die vom Autor durchgeführt wurden, um die gewünschten Gleisfelder aus dem Schienennetz identifizieren zu können.

## 4 Grundlagen / Voraussetzungen

Für die Gleisfeldsuche müssen die Rohdaten noch vorprozessiert werden, um effektiv mit ihnen arbeiten zu können. Jede Linie im Datenbestand muss in ihre Liniensegmente aufgebrochen werden. Der Grund hierfür sind die Anforderungen zur Berechnung der Linienausrichtungen im Gleisfeldgraphen  $G_{\text{Gleisfeld}}$  in Kapitel 5.1. Die einzelnen Mittelpunkte sind eine geeignete Repräsentation für die einzelnen Linien, wie in der Abbildung 9 zu erkennen ist. Diese Mittelpunkte bilden die Knotengrundlage  $V$  für jeden Graphen  $G$ , der für die Gleisfeldererkennung generiert wird. Es ist wichtig, dass die einzelnen Liniensegmente bzw. Knoten eine Referenz zu ihrem Ursprungsgleis besitzen, um eine gewisse Form der Rückverfolgung zu ermöglichen.

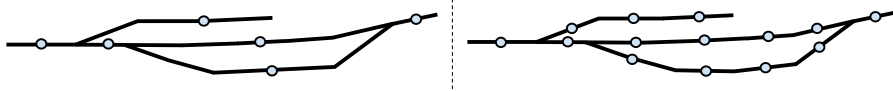


Abbildung 9: Zerlegung der Linien (links) in die einzelnen Liniensegmente (rechts); Die Kreise repräsentieren jeweils die Mittelpunkte der Linien

Nach der Linienzerlegung muss für den Datenbestand ein Nachbarschaftsgraph  $G_{\text{Nachbar}} = (V, E)$  generiert werden: Ein Nachbarschaftsgraph  $G_{\text{Nachbar}}$  enthält die Information über die geometrischen Beziehungen zwischen den Linien. In  $G_{\text{Nachbar}}$  werden die Liniensegmente aus dem Gleisnetz als Knoten  $v \in V$  repräsentiert und zwei Liniensegmente, die sich jeweils an einem Ende ihrer Randpunkte berühren, erhalten eine Kante  $e \in E$  zwischen ihren Knoten. In der Abbildung 10 ist an den drei Mündungspunkten verdeutlicht, dass benachbarte Knoten eine Kante erhalten, selbst wenn diese Pfade in der Realität mit gewöhnlichen Schienenfahrzeugen nicht direkt befahrbar sind. Sind die Linien zerlegt und existiert der Nachbarschaftsgraph  $G_{\text{Nachbar}}$ , kann man mit den eigentlichen Arbeitsschritten der Gleisfeldsuche fortfahren.

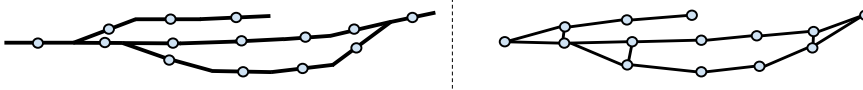


Abbildung 10: Der zerlegte Beispielgraph aus Abbildung 9 (links) und der daraus resultierende Nachbarschaftsgraph (rechts)

## 5 Die Gleisfeldsuche

In diesem Kapitel werden nun die Hauptarbeitsschritte vorgestellt, die zur Gleisfeldsuche benötigt werden. Der gesamte Prozess der Gleisfeldsuche lässt sich folgendermaßen zusammenfassen: Es werden aus einem selbst generierten Gleisfeldgraphen die Zusammenhangskomponenten erfasst, welche grob die Gleisfelder repräsentieren. Wenn man diese Zusammenhangskomponenten erfasst hat, wird die darin befindliche Menge von Liniensegmenten als ein unvollständiges Gleisfeld deklariert, welches im Folgeschritt bis zu seinen (Sub-)Wurzeln durch eine Breitensuche erweitert wird. Die gefundenen (Sub-)Wurzeln der Gleisfelder sind ein nützliches Nebenprodukt für die anschließende Selektion.

### 5.1 Generierung des Gleisfeldgraphen $G_{\text{Gleisfeld}}$

Der Gleisfeldgraph  $G_{\text{Gleisfeld}} = (V, E)$  erfasst die Parallelität zwischen zwei Liniensegmenten. Die Liniensegmente werden wie in  $G_{\text{Nachbar}}$  durch die Knotenmenge  $V$  repräsentiert ( $V_{\text{Gleisfeld}} = V_{\text{Nachbar}}$ ). Zwei Liniensegmente erhalten zwischen ihren beiden Graphknoten eine Kante  $e \in E$  in  $G_{\text{Gleisfeld}}$ , wenn sie sich geometrisch nahe sind, einen annähernd gleichen Ausrichtungswinkel besitzen und in  $G_{\text{Nachbar}}$  keine gemeinsame Kante besitzen. Die Parameter, wie weit genau die Linien voneinander entfernt sein dürfen ( $\Delta_{\text{dist}}$ ) und wie stark die Winkel sich differenzieren dürfen ( $\Delta_{\text{arc}}$ ), müssen für die gewünschten Ansprüche vordefiniert werden. Bei der Wertewahl der beiden Parameter  $\Delta_{\text{dist}}$  und  $\Delta_{\text{arc}}$  ist zu beachten, dass beide nicht zu groß gewählt werden: Ein zu hoher Wert für  $\Delta_{\text{dist}}$  sorgt für unerwünschte Kanten zwischen den Linien vom selben Gleis, wodurch dann Zusammenhangskomponenten auf alleinstehenden Gleisen entstehen. Ein zu hoher Wert für  $\Delta_{\text{arc}}$  sorgt für unerwünschte Kanten an Mündungspunkten von mehreren Gleisfeldern, wodurch diese im Gleisfeldgraphen in den Folgeschritten nicht mehr voneinander unterscheidbar sein werden.

Die Parameter sind ideal gesetzt, wenn bei jedem Gleisfeld eine Zusammenhangskomponente in  $G_{\text{Gleisfeld}}$  entsteht, die mindestens ein Liniensegment von jedem Gleis beinhaltet. Die Abbildung 11 zeigt, wie der Gleisfeldgraph aussehen kann – die rot markierten Kanten sind die, die mindestens vorhanden sein sollten, um das Gleisfeld in den Folgeschritten komplett zu identifizieren.

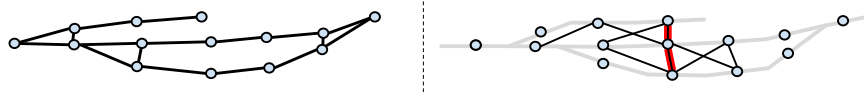


Abbildung 11: Nachbarschaftsgraph  $G_{\text{Nachbar}}$  aus Abbildung 10 (links) und der daraus resultierende Gleisfeldgraph  $G_{\text{Gleisfeld}}$  (rechts) bei annehmbaren Parametern.

## 5.2 Ergänzungen des Gleisfeldgraphen $G'_{\text{Gleisfeld}}$ .

Jedes Gleisfeld sollte im Idealfall nur durch eine Zusammenhangskomponente in  $G_{\text{Gleisfeld}}$  repräsentiert werden. Bei gekrümmte Gleisfeldern wie in der Abbildung 12 entstehen allerdings keine einheitlichen Zusammenhangskomponenten. Dieser Sachverhalt sollte kein Anlass sein die Werte der Gleisfeldgraph-Parameter  $\Delta_{\text{arc}}$  oder  $\Delta_{\text{dist}}$  anzuheben. Für diese Fälle muss der Graph künstlich um ein paar Kanten ergänzt werden. Diese Ergänzungen erfolgen durch einen Nachbarschaftsvergleich: Wenn zwei Zusammenhangskomponenten über mindestens zwei verschiedene Knotenpaare im Nachbarschaftsgraphen verbunden sind, sollen die Kanten der Knotenpaare ergänzt werden – siehe dazu Abbildung 13.

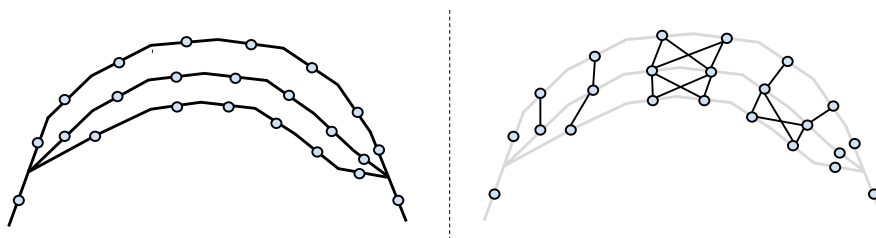


Abbildung 12: Beispielgraph (links) und der daraus resultierende Gleisfeldgraph  $G_{\text{Gleisfeld}}$  (rechts) bei annehmbaren Parametern.

Die Gleisfelder sind nach diesem Arbeitsschritt in den Zusammenhangskomponenten grob identifizierbar, sind allerdings nicht vollständig. Die Vervollständigung erfolgt über die Suche der Wurzeln des Gleisfeldes, in die im Nachhinein jede Linie dem Gleisfeld zugeordnet wird, die sich auf einem Pfad zwischen einer Wurzel und den Linien der Zusammenhangskomponente befindet.

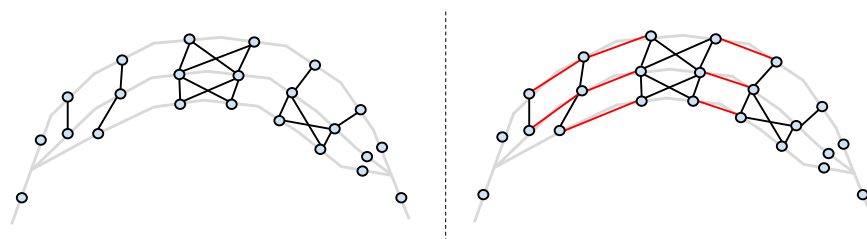


Abbildung 13: Gleisfeldgraph  $G_{\text{Gleisfeld}}$  (links) aus der Abbildung 12 und die künstliche Ergänzung  $G'_{\text{Gleisfeld}}$  (rechts) verdeutlicht durch rote Kanten.

### 5.3 Identifizierung der Randfeatures

Um die Wurzeln effizient durch eine Breitensuche finden zu können, müssen zuerst die Randfeatures aus der Menge der Liniensegmente des unvollständigen Gleisfeldes identifiziert werden. Unter den Randfeatures versteht man die Knoten aus den Zusammenhangskomponenten des Gleisfeldgraphen  $G'_{\text{Gleisfeld}}$ , welche eine Kante zu einem Knoten im Nachbarschaftsgraphen  $G_{\text{Nachbar}}$  besitzen, die nicht zu derselben Zusammenhangskomponente gehören ( $V_{\text{Out}}$ ).

Die gefundenen Randfeatures werden in zwei Gruppen aufgeteilt, je nachdem in welche Ausrichtung die jeweiligen Randfeatures in Richtung  $V_{\text{Out}}$  zeigen. Auf diese Weise erhält man die gewünschten Seiten eines Gleisfeldes, die im Anschluss für die Wurzelsuche separat behandelt werden können (siehe Abbildung 14 rechts). Man beachte, dass ein Randfeature auch in beiden Randfeature-Gruppen sich befinden kann, wenn an beiden Seiten des Randfeatures adjacente Kanten zu einem Knoten  $V_{\text{Out}}$  da sind.

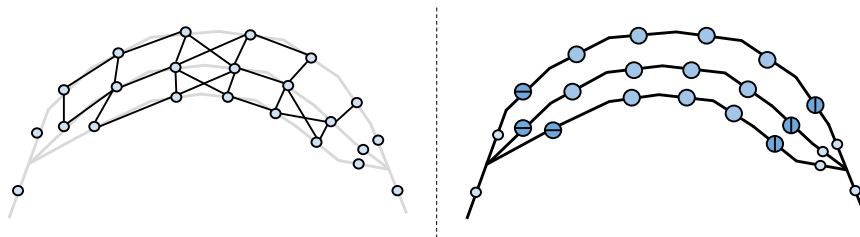


Abbildung 14: Gleisfeldgraph  $G'_{\text{Gleisfeld}}$  (links) aus Abbildung 13 und die daraus resultierende Linienmenge des *unvollständigen* Gleisfeldes in dunkelblau (rechts). Die Knoten mit den senkrechten oder waagerechten Strichen sind die Randfeatures des unvollständigen Gleisfeldes.

### 5.4 Kursgraph $G_{\text{Kurs}}$

Für die Breitensuche zu den Wurzeln wird nicht der Nachbarschaftsgraph  $G_{\text{Nachbar}}$  verwendet, sondern ein Kursgraph  $G_{\text{Kurs}}$ . Der Kursgraph  $G_{\text{Kurs}}$  ist ein gerichteter Teilgraph vom Nachbarschaftsgraphen ( $E_{\text{Kurs}} \subseteq E_{\text{Nachbar}}$ ). Dabei werden nur Kanten zwischen den Knoten übernommen, die eine vorbestimmte Richtung aufweisen (siehe Abbildung 15). Die Überlegung dahinter ist die Folgende: Wenn wir die Randfeatures mit ihrer Ausrichtung in den Raum kennen; dann ist es sinnvoll die vorgegebene Richtung von den Randfeatures weiterverfolgt, um zur Wurzel zu gelangen. Auf diese Weise simuliert man auch die Manövrierfähigkeit eines Schienenfahrzeugs in die vorgegebene Richtung.

Jede Randfeature-Gruppe  $V_{\text{Out}}$  bedient sich aufgrund ihrer eigenen Ausrichtung auch an einem eigenen Kursgraphen  $G_{\text{Fluss}}$ , weshalb es unsinnig ist jeden Kursgraphen vollständig zu generieren. Es empfiehlt sich, anstatt den Graphen vorzuprozessieren, nur die lokalen Nachbarschaften für eine Linie aus  $G_{\text{Nachbar}}$  abzuleiten, wenn diese während der Breitensuche benötigt werden.

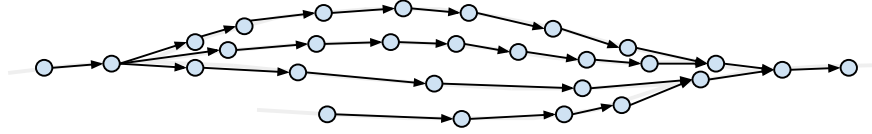


Abbildung 15: Kursgraph  $G_{\text{Kurs}}$  für das Gleisfeld aus Abbildung 7 mit der vorgegebenen Richtung nach *rechts*.

## 5.5 Die Wurzelsuche

Zu diesem Zeitpunkt des Prozesses liegt eine Menge von unvollständigen Gleisfeldern zur Weiterverarbeitung vor, die jeweils auf beiden Seiten eine Gruppe von Randfeatures  $V_{\text{Out}}$  besitzen. In den nächsten beiden Arbeitsschritten versuchen wir für jede Randfeature-Gruppe eine Linie zu finden, in welche all die Randfeatures aus der Gruppe münden – die sogenannten Wurzeln. Wie aber auch schon in der Aufgabenstellung erläutert, ist die ideale Wurzel nicht immer gegeben, so dass man auch nach Subwurzeln Ausschau halten muss. Für diese Anforderung werden an dieser Stelle in der Breitensuche Token verwendet. Ein Token ist ein Informationsträger, der auf eine bestimmte Position gesetzt wird, um ortsbezogene Informationen zu vermitteln. Für eine ungefähre bildliche Vorstellung von Token kann man sich das Märchen von Hänsel und Gretel vor Augen halten, wo die Brotkrümel, die auf dem Weg in den tiefen Wald von Hänsel verteilt werden und den Weg nach Hause weisen. Diese Brotkrümel sind im übertragenen Sinne Token.

Diese Token werden in den nächsten beiden Schritten durch die Breitensuche verteilt, und die entstandene Verteilung wird daraufhin ausgewertet, um die Wurzel bzw. Subwurzeln zu ermitteln. Um die Übersichtlichkeit in den Abbildungen von der Wurzelsuche zu gewährleisten, werden vom unvollständigen Gleisfeld aus nur noch die Randfeatures und der relevante Teil des Kursgraphen  $G_{\text{Kurs}}$  zu den Wurzeln abgebildet (siehe Abbildung 16). Die Knotenpunkte der Randfeatures werden alle mit Großbuchstaben gekennzeichnet; alle anderen Knotenpunkte erhalten Kennzeichnungen durch Kleinbuchstaben.

### 5.5.1 Tokenverteilung durch Breitensuche

Die Tokenverteilung erfolgt separat für die beiden Kursgraphinstanzen des Gleisfeldes. Von jedem Randfeature aus wird im Kursgraphen  $G_{\text{Kurs}}$  eine Breitensuche durchgeführt. Dabei wird bei jedem besuchten Knoten ein Token hinterlassen, welches die bisher zurückgelegte Strecke und das Ursprungs-Randfeature speichert. Diese Token werden im nächsten Arbeitsschritt ausgewertet, um die Wurzel bzw. die Subwurzeln auf der jeweilig behandelten Seite des Gleisfeldes herauszufinden. Die Token besitzen außerdem jeweils einen Verweis auf den zuvor platzierten Token, der in Richtung des Ursprungs-Randfeatures führt – diese Angabe ist für die abschließende Rückverfolgung nach der Findung der Wurzeln erforderlich.

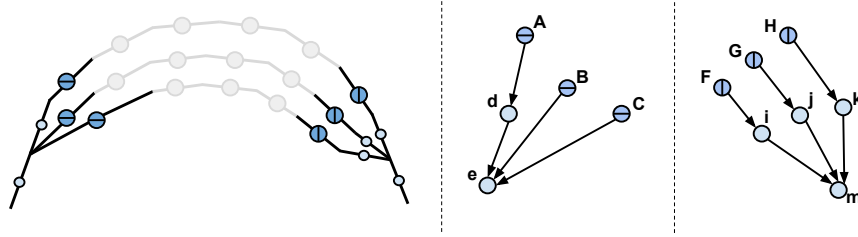


Abbildung 16: Links, Gleisfeld aus der Abbildung 14 mit seinen Randfeatures; Rechts die relevanten Bestandteile des Gleisfeldes für die Wurzelsuche in Form von zwei Kursgraph-Instanzen.

derlich.

Wenn bei einem Knotenpunkt mehrere Token vom selben Ursprungs-Randfeature vorhanden sind, dann soll nur das Token mit der geringsten zurückgelegten Distanz behalten werden – die anderen Token vom selben Randfeature werden verworfen (siehe Abbildung 17). Auf diese Weise hat man ein geeignetes Abbruchkriterium für die Tokenverteilung, wenn die Anzahl der Token auf einem Knotenpunkt der Anzahl in dem aktuellen Durchgang beteiligten Ursprungs-Randfeatures entspricht.

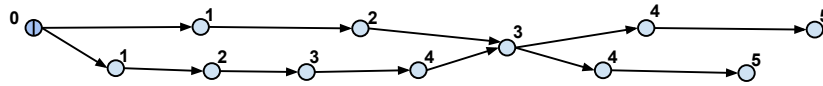


Abbildung 17: Beispiel für eine Tokenverteilung von einem einzelnen Randfeature aus. Die Ziffern über den Knoten beschreiben die kürzeste Distanz zum Randfeature.

Als geeignetes Maß für die Breitensuche sind sowohl die Anzahl der Knoten, die bisher besucht wurden, als auch die euklidische Distanz betrachtbar – Voraussetzung für die Wahl der Anzahl besuchter Knoten als Maß ist allerdings eine gewisser Grad an Homogenität, was die Digitalisierung der Linien im Datenbestand betrifft: Die Linien sollten annähernd gleich lang sein und auch an den Mündungspunkten gleichgut digitalisiert sein, andernfalls führt dieses Maß zu unerwünschten, ungleichmäßigen Suchradien. Die Wahl der zurückgelegten euklidischen Distanz als Maß verspricht da mehr Konsistenz, wenn die Linien von ihrer geometrischen Lage nicht von der Realität abweichen. Aufgrund der Übersicht wird in den Abbildungen 17 und 18 zur Wurzelsuche die Anzahl der besuchten Knoten als Maß verwendet. Man beachte, dass die Randfeatures selbst auch Token besitzen, die jeweils auf die Randfeatures verweisen, auf denen sie liegen, mit der Distanz null (*Nulltoken*).

Es ist wichtig, dass der Radius nicht zu kurz gewählt wird; noch wichtiger ist, dass überhaupt ein Suchradius definiert wird, da die Breitensuche

nicht auf den kompletten Graphen angewendet werden soll – dieses wäre eine Verschwendung von Rechnerressourcen. Idealerweise sollte die Größe des Suchradius dynamisch aus den Eigenschaften des bestehenden, unvollständigen Graphen abgeleitet werden (zB. die Anzahl der Randfeatures).

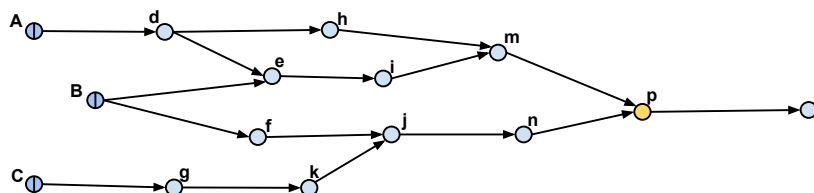


Abbildung 18: Kursgraph-Instanz: Die Knoten A, B & C bilden die Randfeature-Gruppe; der Knoten p ist die gesuchte Wurzel für diese Instanz, die es vom Algorithmus zu finden gilt.

Knoten	A	B	C
A	0		
B		0	
C			0
d	1		
e	2	1	
f		1	
g			1
h	2		
i	3	2	
j		2	3
k			2
m	3	3	
n		3	4
p	4	4	5
q	5	5	6

Tabelle 2: Ergebniss der Tokenverteilung für die Abbildung 18

### 5.5.2 Auswertung der Tokenverteilung

Nach der Tokenverteilung durch die Breitensuche folgt die Auswertung der entstandenen Verteilung, um die Wurzeln heraus zu finden. Gesucht werden eine minimale Anzahl an (Sub-)Wurzeln, die gemeinsam alle Randfeatures abdecken. Es werden alle Linien, die über ein Token verfügen, als Wurzelkandidaten angesehen. Die absoluten Favoriten sind die Linien mit den meisten Token, da von diesen Linien aus die meisten Randfeatures erreichbar sind. Unter diesen Favoriten ist der, mit dem geringsten Abstand



zu dem von ihm am weitesten entfernten Randfeature, eine gesuchte Wurzel. Wenn ein Randfeature keine Wurzel mit anderen Randfeatures bildet, wird es wegen des *Nulltoken* selbst zu einer Subwurzel. Der nun folgende Pseudo-Code für einen Greedy-Algorithmus zur Bestimmung der Wurzeln macht sich die obigen Erkenntnisse zu Nutzen:

**Data :** Liste  $K$  der Linien mit ihren Token.  
 sortiere  $K$  primär absteigend nach Anzahl der Token in den Linien;  
 sortiere  $K$  sekundär aufsteigend nach höchste Distanz bei den Token;  
 Liste  $L \leftarrow$  alle Randfeatures aus der Instanz.;  
**while**  $L$  nicht leer &  $K$  nicht leer **do**  
      $P \leftarrow$  nimm die oberste Linie von  $K$ ;  
     Liste  $U \leftarrow$  alle Randfeatures aus den Token von  $P$ ;  
     **if** mind. ein Randfeature von  $U \in L$  **then**  
          $\rightarrow P$  ist eine Wurzel;  
         entferne alle Randfeature  $R \in L$ , wenn  $R \in U$ ;  
     **end**  
**end**

**Algorithmus 1 :** Greedy-Algorithmus zur Bestimmung der Wurzeln aus der Tokenverteilung.

Für die Kursgraph-Instanz aus der Abbildung 18 entsteht bei den Wurzel-Kandidaten aus der Tabelle folgende Reihenfolge:  $p \rightarrow q \rightarrow e \rightarrow i \rightarrow j \rightarrow m \rightarrow n \rightarrow A \rightarrow B \rightarrow C \rightarrow d \rightarrow f \rightarrow g \rightarrow h \rightarrow k$ . Der Greedy-Algorithmus hätte dann die Linie  $p$  aus der Liste genommen und erkannt, dass alle Randfeatures in  $p$  münden und ihn zur Wurzel deklariert. Da keine weiteren Randfeatures in der Liste  $L$  mehr vorhanden sind, endet der Algorithmus an dieser Stelle und  $p$  bleibt als einzige Wurzel. Nachdem alle Wurzeln einer Kursgraph-Instanz gefunden wurden, erfolgt der abschließende Schritt der Rückverfolgung, in dem alle Linien erfasst werden, die sich auf dem kürzesten Pfad zwischen den Wurzeln und den auf sie bezogenen Randfeatures befinden. Diese Linien werden dem Gleisfeld zugeordnet. Das Gleisfeld ist nun vollständig und kann für die lokale Selektion verwendet werden, welche in dieser Arbeit nicht weiter behandelt wird.

## 6 Implementierung

Das entstandene Programm zur Gleisfeldidentifizierung ist ein Eclipse-Projekt basierend auf der Programmiersprache Java. Es werden im Projekt zwei verschiedene Bibliotheken verwendet: Die geotools-Bibliothek ist eine mächtige Open-Source-Java-Bibliothek für GIS-Fragestellung, die im Kontext der Bachelorarbeit in erster Linie zur Visualisierung und Bearbeitung der Shapefiles verwendet wird. Der graphentechnische Teil ist mittels der generischen Graph-Klassen der JGraphT-Bibliothek umgesetzt worden.

Bei der Wahl der Parameter für die Generierung des Gleisfeldgraphen haben sich für den verwendeten Datensatz die Werte  $\delta_{dist} = 10$  Meter und  $\delta_{arc} = 0.8^\circ$  etabliert. Diese Werte wurden auch für die Ergebnisse verwendet, die im Kapitel 7 vorgestellt werden. Für die Abfragen zur räumlichen Nähe, was zur Erstellung des Gleisfeldgraphen nötig war, wurden alle Linien in einen Quadtree einsortiert mit den kartesischen Koordinaten der Geometrien als räumliche Indizes.

Die Kursgraphen werden während der Breitensuche nur lokal generiert, da ein kompletter Kursgraph für ein Gleisfeld nie benötigt wird. Außerdem wird kein instanziieller Graph aufgebaut, sondern die implementierte Runner-Klasse, die die Breitensuche ausführt, vergleicht die vorgegebene Ausrichtung mit den adjazenten Linien im Nachbarschaftsgraphen  $G_{\text{Nachbar}}$ , die als potenzielle Nachfolger in Frage kommen.

## 7 Experimente

In diesem Kapitel werden die Ergebnisse, die aus der eigenen Implementierung des Verfahrens zur Gleisfeldsuche entstammen, demonstriert und verglichen. Es werden drei Testgebiete vorgestellt, in denen zuvor versucht wurde die einzelnen Gleisfelder manuell zu identifizieren. In den Folgetexten werden die identifizierten Gleisfelder als Ergebnisse von Maschine (automatisiert) und Mensch (manuell) bezeichnet. Bei der manuellen Identifizierung der Gleisfelder ist festzustellen, dass die Auseinanderhaltung von Gleisfeldern bzw. die Zuordnung einzelner Gleise zu einem Gleisfeld alles andere als trivial ist. Hierzu folgen noch Beispiele im Testgebiet.

Die Abbildungen von den Testgebieten sind folgendermaßen zu interpretieren: Alle dargestellten Linien repräsentieren die Gleise im Testgebiet. Von den Testgebieten sind mehrere identische Instanzen nebeneinander dargestellt, weil die erkannten Gleisfelder sich überlappen und in einer Abbildung mit nur einer Instanz nicht voneinander unterscheidbar wären. Die vom Menschen erkannten Gleisfelder sind nummeriert und an der breitesten Stelle mit einer grauen Hilfslinie markiert. Diese grauen Hilfslinien werden in den Erörterungen fortan als *Gitarrengriff* bezeichnet. Die Farbe der Gleise beschreibt, ob Mensch oder Maschine oder beide das betroffene Gleis zum Gleisfeld zugeordnet haben: Orange steht für die Zuordnung nur durch den Menschen, blau nur durch die Maschine und grün bedeutet, dass sowohl Mensch als auch Maschine das betroffene Gleis zum Gleisfeld zugeordnet haben. Rote Gleise sind Markierungen, auf die im Text gesondert Bezug genommen wird. Aufgrund der Größe der Abbildungen sind diese als Anlagen vom restlichen Text ausgesondert.

## 7.1 Testgebiet Porrentruy

Bei dem ersten Testgebiet in der Abbildung 20 handelt es sich um das Schienennetz von der Gemeinde Porrentruy im Kanton Jura an der französischen Grenze. In diesem Schienennetz sind zwei große Gleisfelder zu erkennen: Ein mehrspuriges Abstellgleisfeld (Nummer 1) und ein Durchfahrtsgleis, welches sich von Süden aus nach Norden auf der Anhöhe des Bahnhofes in mehrere Gleise aufteilt (Nummer 2). Von den vier ausgehenden Gleisen nach Norden driften die zwei rechten Gleise auf das Abstellgleisfeld (Nummer 1) ab. Wenn das ganz rechte Gleis (s. rote Markierung am Gitarrengriff) eine Rangiermöglichkeit hätte, um wieder zu den restlichen Gleisen nach links zu gelangen, könnte man das Gleisfeld Nummer 2 an der Wurzel am nördlichen Durchfahrtsgleis abschließen. Stattdessen entstehen zwei Subwurzeln – ein Szenario, wie es bereits in Kapitel 3 beschrieben und in der Abbildung 8 dargestellt wurde. Die beiden rechten Gleise sind auch Teil des Abstellgleisfeldes (Nummer 1), aufgrund ihrer Beschaffenheit.

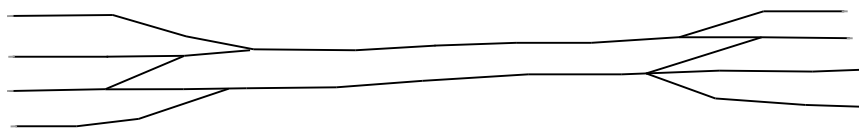
Beim Vergleichen der beiden Ergebnisse von Mensch und Maschine zeigt sich, dass die Ergebnisse von der Maschine etwas umfassender ausgefallen sind, als die Ergebnisse vom Menschen. Beim Abstellgleisfeld (Nummer 1) sind zwei blaue Schienen zu erkennen, die zwar für das menschliche Auge offensichtlich abdriften und eine Sackgasse bilden, an ihren Ursprüngen allerdings noch für wenige Meter parallel zum restlichen Gleisfeld verlaufen und somit für die Maschine noch Teil des Gleisfeldes sind. Die südliche Wurzel vom Abstellgleis (Nummer 1) ist von der Maschine um ein Liniensegment später erfasst worden als vom Menschen (siehe roter Pfeil in Abbildung 20). In einer GIS-Analyse ist festzustellen, dass es sich bei dem kurzen Liniensegment noch um eine Mündungsschiene vom linken Gleis handelt und somit keine Wurzel sein kann. Aufgrund der ungeschickten Überlappung der Linien konnte dieser Fall nur von der Maschine erkannt werden.

Beim Gleisfeld Nummer 2 sind von der Maschine drei Gleise mehr dem Gleisfeld zugeordnet worden als vom Menschen. Aufgrund der Parallelität von allen drei Gleisen zum restlichen Gleisfeld sind die Zuordnungen nachvollziehbar, sie sind allerdings auf den ersten Blick nicht eindeutig: Das linke blaue Gleis im Gleisfeld Nummer 2 besitzt einen eindeutig größeren Abstand zum restlichen Gleisfeld, das obere blaue Gleis ist ein Ausläufer von einem grünen Gleis und das rechte blaue Gleis verläuft parallel zum rechten grünen Gleis, welches vom restlichen Gleisfeld abdriftet und eine Subwurzel erzeugt. Die Aufnahme des rechten blauen Gleises sorgt für eine Verschiebung der Subwurzel weiter nach oben.

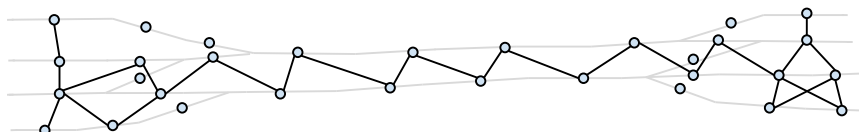
## 7.2 Testgebiet Pratteln

Bei dem nächsten Testgebiet (Abbildung 21) handelt es sich um eine Gleisstruktur, die in einem Industriegebiet von Pratteln, südöstlich von Basel, am Rhein gelegen ist. In diesem Testgebiet sind mehrere Gleisfelder zu erkennen, die in ein zentrales Gleisfeld (Nummer 4) münden. Die Herausforderung war, die Gleisfelder 1, 4 und 5 an ihrer gemeinsamen Kreuzung trennbar zu machen. Bei den restlichen Gleisfeldern bestand die Herausforderung darin, die Gleisfelder zu erkennen bei geringerer Anzahl paralleler Gleise.

Am Gleisfeld Nummer 1 sind sich Mensch und Maschine bei der Identifizierung einig. Am Gleisfeld Nummer 2 weist die Maschine ein kurzes Abstellgleis dem Gleisfeld zu, welches aufgrund seiner Länge keine Relevanz besitzt für die Selektion; ansonsten sind sich auch hier Mensch und Maschine einig. Beim Gleisfeld Nummer 3 beginnen die ersten Probleme: Obwohl es gewollt ist, dass mehrspurige Linien wie die Doppelspur als Gleisfelder erkannt werden, führen diese in der Anwendung zu unerwünschten Beziehungen zwischen zwei Gleisfeldern. Gleisfeld, die über solche Doppelspuren verbunden sind, wie zwischen dem Gleisfeld 3 und 6, können von der Maschine nicht mehr getrennt werden. Der Sachverhalt ist in der Abbildung 19a und 19b nochmals skizziert. Die gefundenen Gleisfelder Nummer 4 von Mensch und Maschine sind in diesem Fall wieder identisch. Beim Gleisfeld Nummer 5 kommt es seitens der Maschine zu einer nicht erklärlichen Verlängerung des Gleisfeldes, obwohl dieses schon vorher an einem anderen Mündungspunkt abschließbar gewesen wäre (siehe rote Markierung). Das Gleisfeld Nummer 7 weist einen sehr großen Zwischenabstand zwischen zwei Gleisen auf, der den Mindestabstand  $\delta_{dist}$  überschreitet, was es der Maschine eindeutig schwerer macht das Gleisfeld komplett zu erfassen. Die unteren orangen Gleise sind von der Maschine als ein Paar Gleisfelder für sich erkannt worden; diese sind in der Abbildung 21 als rote Gleise markiert worden.



(a) Zwei Gleisfelder, verbunden über eine Doppelspur.



(b) Der resultierende Gleisfeldgraph  $G_{\text{Gleisfeld}}$  für die Abbildung 19a

Abbildung 19: Zwei Gleisfelder, die von der Maschine nicht differenzierbar sind.

### 7.3 Testgebiet Solothurn

In dem dritten und letzten Testgebiet (Abbildung 22) handelt es sich um ein Schienennetz aus der Stadt Solothurn im gleichnamigen Kanton. Die dort auffindbare Schienenstruktur ist eine einzige Masche, die nur schwer zu differenzieren ist.

Als Mensch erkennt man zwei große Gleisfelder, die parallel zueinander verlaufen und nur durch eine geringe Lücke voneinander differenzierbar wären (Gleisfeld Nummer 1 und 2). Genauso gut könnten diese auch zu einem großen Gleisfeld zusammenfasst werden (Nummer 3). Die Maschine sieht hingegen noch weitere Beziehungen zwischen den restlichen Gleisen und bildet ein viel größeres Gleisfeld als Nummer 3, welches ohnehin schon umfangreich ist. Diese entstandene Instanz von der Maschine ist alles andere als einfach zu generalisieren in den Folgeschritten. Dieses Ergebnis kann wieder auf die Sensibilität der Maschine zurückgeführt werden, die zwar in den Testgebieten zu interessanten Ergänzungen geführt hat, in diesem Beispiel allerdings scheitert. Solch riesige Gleisfelder entstehen auch adäquat bei den Schienenstrukturen in Großstädten, wie Basel oder Zürich. Für die Handhabung solcher Fälle sind eindeutig weitere Überlegungen notwendig.

## 8 Fazit

Mit dem beschriebenen Verfahren ist man in der Lage, bis zu einem gewissen Komplexitätsgrad erfolgreich Gleisfelder und deren Wurzeln für die struktursuchende Generalisierung von Schienennetzen identifizieren zu können. Die Analyse der Testgebiete sollte die Sensibilität der Implementierung aufzeigen. Die geometrischen Beziehungen zwischen den Gleisen werden von der Maschine viel einfacher erkannt als vom Menschen. Diese Sensibilität hat aber auch ihre Kehrseite; Mehrspurige Verbindungen zwischen den Gleisfeldern machen diese unzertrennlich, was zu riesigen, unüberschaubaren Gleisfeldgruppen führt und die Selektion wieder in solchen Instanzen erschwert. Um diesem Effekt entgegenzuwirken sind weitere Überlegungen notwendig. Insgesamt kann man mit den Ergebnissen aus den Testgebieten zufrieden sein. Als Ausblick für diese Arbeit könnte man versuchen, die gefundenen unüberschaubaren Gleisfeld-Instanzen weiter zu verarbeiten oder das Regelwerk bei der Generierung des Gleisfeldgraphen zu optimieren. Auf jeden Fall ist mit dieser Arbeit eine Grundlage geschaffen, die durch weitere Verbesserungszyklen eine vielversprechende Lösung zur automatischen bzw. semi-automatischen Generalisierung von Schienennetzen bietet.

## Literatur

- Benz, S. (2013). *Algorithmen zur automatischen Selektion von Straßennetzwerken*. M. Sc., Universität Zürich.
- Chimani, M., T. van Dijk und J. Haunert (2014). *How to Eat a Graph: Computing Selection Sequences for the Continuous Generalization of Road Network*. In: *Proc. 22st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (2014)*.
- Horton, R. (1945). *Erosional development of streams and their drainage basins; hydrophysical approach to quantitative morphology*. Geological society of America bulletin, 56(3):275–370.
- Knoke, D. und R. Burt (1983). *Prominence*. Applied network analysis, S. 195–222.
- Mackaness, W. und G. Mackechnie (1999). *Automating the detection and simplification of junctions in road networks*. GeoInformatica, 3(2):185–200.
- Mark, D. (1989). *Conceptual basis for geographic line generalization*. In: *Proc. 9th International Symposium on Computer-Assisted Cartography (Auto-Carto 9)*, S. 68–77.
- Page, L., S. Brin, R. Motwani und T. Winograd (1999). *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66, Stanford University.
- Thomson, R. und D. Richardson (1999). *The "good continuation" principle of perceptual organization applied to the generalization of road networks*. In: *Proc. 19th International Cartographic Conference (ICC 2011)*, S. 1215–1225.
- Töpfer, F. und W. Pillewizer (1966). *The principles of selection*. The Cartographic Journal, 3(1):10–16.

## Abbildungsverzeichnis

1	Beispielnetz bestehend aus einem einzigen Gleisfeld . . . . .	3
2	Mögliche Generalisierungsergebnisse für Abbildung 1; Links, wenn das Gleisfeld nach den Generalisierungsanforderungen existent ist, ansonsten Rechts . . . . .	4
3	Links Beispielstraßennetz, Rechts die Strokes des Straßennetzes. . . . .	5
4	Hortonsches Ordnungssystem: Ordnungszahl der einzelnen Flüsse ist durch die Strichlinienart gekennzeichnet: durchgezogen → 1, gestrichelt → 2, punkt-gestrichelt → 3, gepunktet → 4 . . . . .	5
5	Links das gegebene Straßennetz, rechts der daraus generierte Nachbarschaftsgraph. . . . .	6

6	Links der gegebene Graph mit Gewichtung, rechts der selbe Graph nach der vierten Iteration vom <i>GreedySelection</i> -Algorithmus. . . . .	8
7	Beispielgleisfeld; Die Punkte sind die Wurzeln des Gleisfeldes. . . . .	9
8	Beispielgleisfeld, wo die zwei unteren Gleise vom Rest abdriften und eine eigene Subwurzel (3) bilden. Der Rest sammelt sich an der anderen Subwurzel (2) . . . . .	9
9	Zerlegung der Linien (links) in die einzelnen Liniensegmente (rechts); Die Kreise repräsentieren jeweils die Mittelpunkte der Linien . . . . .	10
10	Der zerlegte Beispielgraph aus Abbildung 9 (links) und der daraus resultierende Nachbarschaftsgraph (rechts) . . . . .	10
11	Nachbarschaftsgraph $G_{\text{Nachbar}}$ aus Abbildung 10 (links) und der daraus resultierende Gleisfeldgraph $G_{\text{Gleisfeld}}$ (rechts) bei annehmbaren Parametern. . . . .	11
12	Beispielgraph (links) und der daraus resultierende Gleisfeldgraph $G_{\text{Gleisfeld}}$ (rechts) bei annehmbaren Parametern. . . . .	12
13	Gleisfeldgraph $G_{\text{Gleisfeld}}$ (links) aus der Abbildung 12 und die künstliche Ergänzung $G'_{\text{Gleisfeld}}$ (rechts) verdeutlicht durch rote Kanten. . . . .	12
14	Gleisfeldgraph $G'_{\text{Gleisfeld}}$ (links) aus Abbildung 13 und die daraus resultierende Linienmenge des <i>unvollständigen</i> Gleisfeldes in dunkelblau (rechts). Die Knoten mit den senkrechten oder waagerechten Strichen sind die Randfeatures des unvollständigen Gleisfeldes. . . . .	13
15	Kursgraph $G_{\text{Kurs}}$ für das Gleisfeld aus Abbildung 7 mit der vorgegebenen Richtung nach <i>rechts</i> . . . . .	14
16	Links, Gleisfeld aus der Abbildung 14 mit seinen Randfeatures; Rechts die relevanten Bestandteile des Gleisfeldes für die Wurzelsuche in Form von zwei Kursgraph-Instanzen. . . . .	15
17	Beispiel für eine Tokenverteilung von einem einzelnen Randfeature aus. Die Ziffern über den Knoten beschreiben die kürzeste Distanz zum Randfeature. . . . .	15
18	Kursgraph-Instanz: Die Knoten A, B & C bilden die Randfeature-Gruppe; der Knoten p ist die gesuchte Wurzel für diese Instanz, die es vom Algorithmus zu finden gilt. . . . .	16
19	Zwei Gleisfelder, die von der Maschine nicht differenzierbar sind. . . . .	20
20	Schienennetz Porrentruy – Vergleich der erkannten Gleisfelder vom menschlichen Auge und vom Algorithmus. . . . .	24
21	Schienennetz Pratteln – Vergleich der erkannten Gleisfelder vom menschlichen Auge und vom Algorithmus. . . . .	25
22	Schienennetz Solothurn – Vergleich der erkannten Gleisfelder vom menschlichen Auge und vom Algorithmus. . . . .	26

## Anlagen

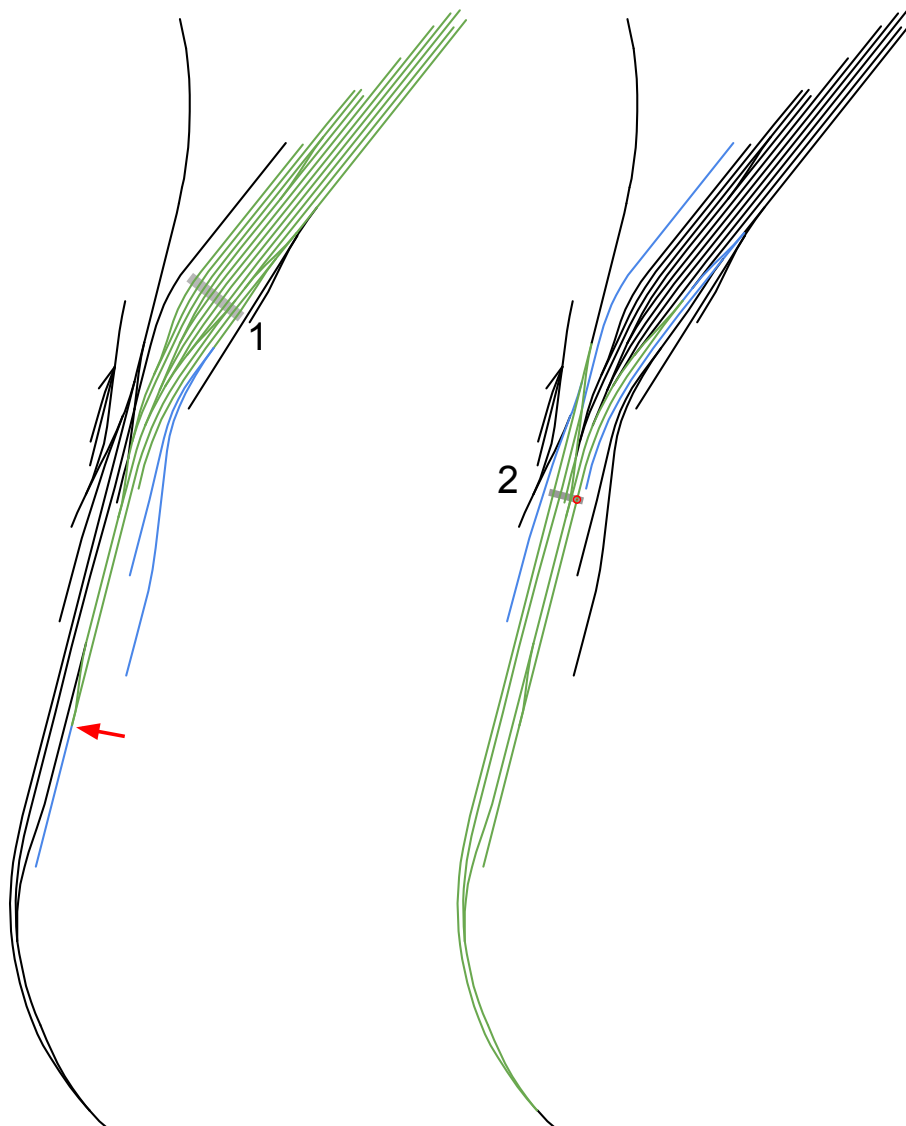


Abbildung 20: Schienennetz Porrentury – Vergleich der erkannten Gleisfelder vom menschlichen Auge und vom Algorithmus.



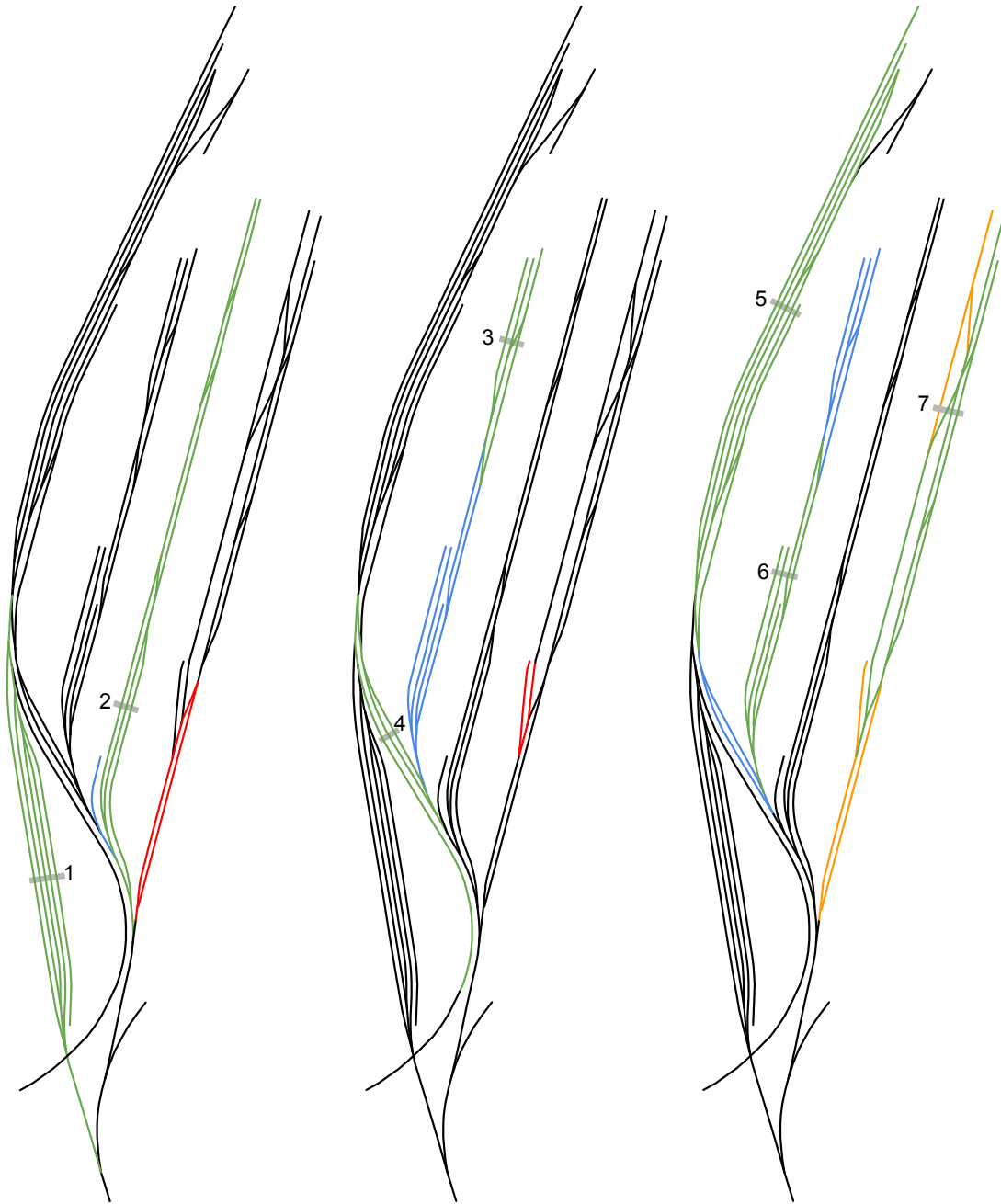


Abbildung 21: Schienennetz Pratteln – Vergleich der erkannten Gleisfelder vom menschlichen Auge und vom Algorithmus.

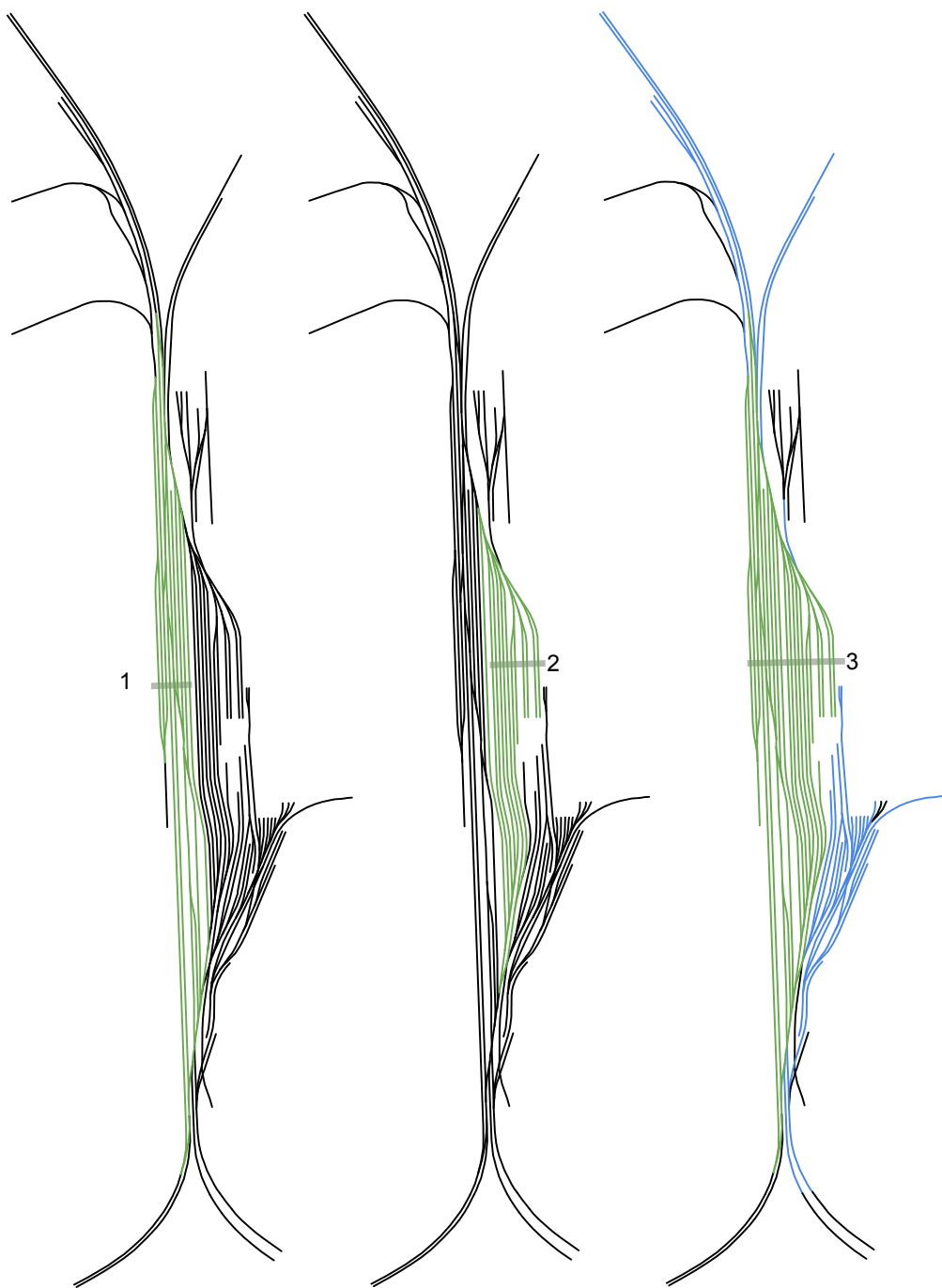


Abbildung 22: Schienennetz Solothurn - Vergleich der erkannten Gleisfelder vom menschlichen Auge und vom Algorithmus.