

# Modifikation des A\*-Algorithmus für energieeffizientes 3D-Routing

Dzenan DŽAFIĆ, Dominik FRANKE, Danni BAUMEISTER und Stefan KOWALEWSKI  
RWTH Aachen · dzafic@embedded.rwth-aachen.de

*Dieser Beitrag wurde durch das Programmkomitee als „reviewed paper“ angenommen.*

## Zusammenfassung

Um eine Route zu berechnen, greifen die meisten Navigationssysteme auf zweidimensionales Kartenmaterial zurück. In der Regel hat der Benutzer vor der Berechnung die Wahl, ob er die kürzeste oder schnellste Route berechnen lassen möchte. Navigationssysteme verwenden für die Routenberechnung oftmals den A\*-Algorithmus. Um eine energieeffiziente Route unter Berücksichtigung der Steigung von Streckenabschnitten zu berechnen, wie das bei dem eNav-Projekt der Fall ist, werden jedoch auch Steigungs-/Höheninformationen benötigt. Aus diesen zusätzlichen Daten kann eine dreidimensionale Karte generiert werden. Eine Modifikation des A\*-Algorithmus ermöglicht mit dieser Datenbasis eine Berechnung der energieeffizientesten Route bezüglich der Steigung. Somit können große Anstiege umfahren werden und eine effizientere Nutzung der Akkukapazität erzielt werden. Dieser Beitrag stellt die notwendigen Modifikationen und Optimierungen bei dem A\*-Algorithmus vor, mit deren Hilfe eine Berechnung der effizientesten Route bezüglich der Streckensteigung auf 3D-Kartenmaterial ermöglicht wird. Zudem wird durch einen Korrektheitsbeweis der heuristischen Funktion des A\*-Algorithmus gezeigt, dass die modifizierte Version des A\*-Algorithmus ebenfalls korrekt ist.

## 1 Einleitung

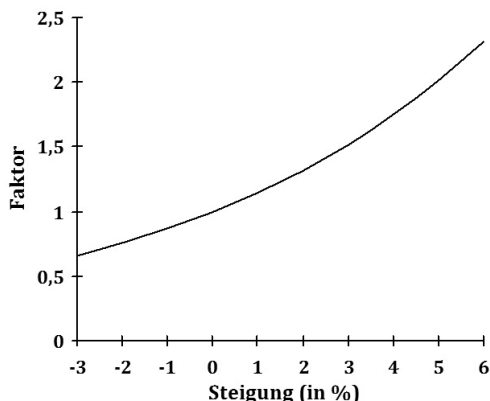
Während die meisten Navigationssysteme die schnellste und kürzeste Route berechnen, ermittelt eNav die effizienteste Route unter Berücksichtigung der Streckensteigung. Hierzu benötigt das Navigationssystem entweder Steigungsinformationen der Streckenabschnitte oder Höheninformationen der Streckenknoten. Aus den Höheninformationen der Streckenknoten kann man die Steigungsinformationen der Streckenabschnitte ableiten. eNav benötigt die Streckensteigung, um den Energieverbrauch für die unterschiedlichen Streckenabschnitte und somit für die gesamte Strecke zu berechnen. Weitere Faktoren wie beispielsweise der Straßenbelag und die Temperatur können die Effizienz ebenfalls beeinflussen, stehen jedoch nicht im Fokus dieser Arbeit. Der vorliegende Beitrag beschreibt eine Modifikation des A\*-Algorithmus, welche unter Berücksichtigung der Topologie eine energieeffiziente Route berechnen kann. Zudem wird die Korrektheit des modifizierten Algorithmus programmatisch bewiesen. Eine Zusammenfassung rundet den Beitrag ab.

## 2 Grundlagen

In den folgenden Unterkapiteln werden die Grundlagen für die Modifikation des A\*-Algorithmus (HART, NILSSON & RAPHAEL 1968) vorgestellt. Aus Platzgründen verweisen wir für eine detaillierte Beschreibung des A\*-Algorithmus auf (NORVIG & RUSSELL 2012). Dort wird auch die Rolle der Luftlinie ausführlich erläutert, auf welche in diesem Beitrag oft eingegangen wird. Weiterhin wird kurz auf Steigungs- und Höheninformationen eingegangen, welche eine wichtige Rolle für die Modifikation spielen.

### 2.1 eNav

Das eNav-Projekt beschäftigt sich mit der Entwicklung eines Navigationssystems für Elektrorollstühle, welches eine energieeffiziente Nutzung der Akkukapazität durch Berücksichtigung der topologischen Streckensteigung gewährleistet. eNav verwendet hierzu den A\*-Algorithmus. Um eine energieeffiziente Route zu berechnen, ersetzt eNav bei der Routenberechnung die Kantengewichte, welche üblicherweise Angaben über die Distanz zweier Knoten beinhalten, mit dem Energieverbrauch für die Strecke zwischen den beiden Knoten. Bei der Berechnung des Energieverbrauchs wird die Streckensteigung anhand einer Verbrauchsfunktion berücksichtigt. Eine beispielhafte Verbrauchsfunktion wird in Abbildung 1 grafisch dargestellt. Weitere Informationen zur Funktion können aus (FRANKE et al. 2011) entnommen werden.

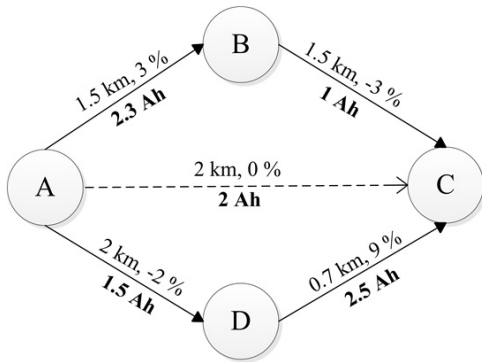


**Abb. 1:**  
Verbrauchsfunktion eines  
Elektrorollstuhls

Wie in Abbildung 1 erkennbar ist, steht der Energieverbrauch eines Elektrorollstuhls in einem exponentiellen Verhältnis zur Steigung. Dadurch kann eine längere Strecke mit geringer Steigung weniger Energie verbrauchen als eine kürzere Strecke mit relativ hoher Steigung. Die bei der Berechnung der Route benötigten Informationen sind in Abbildung 2 grafisch dargestellt. Ein abstrakter Graph, welcher dem A\*-Algorithmus als Grundlage dient, repräsentiert das Straßennetz. Zu jedem Streckenabschnitt sind die Distanz (km), die Streckensteigung (%) und der Verbrauch (Ah) angegeben. Die Luftlinie vom Start- zum Zielknoten wird durch eine gestrichelte Linie dargestellt und die Straßenkanten mithilfe durchgezogener Linien.

Im Vergleich zum Stand der ersten Veröffentlichung (FRANKE et al. 2011) haben sich einige Optimierungen ergeben. Derzeit wird nicht ausschließlich auf der Basis von OSM-Daten

geroutet, da die Datensätze zu wenige Höheninformationen enthalten. Aufgrund der mangelnden Höheninformationen hat der ursprüngliche eNav-Algorithmus eine 0 %-ige Steigung angenommen, wo keine Steigungsinformationen aus Höhendaten berechnet werden konnten oder angegeben waren. Diese Annahme hat die Qualität des Routingergebnisses verfälscht, was in FRANKE et al. (2011) durch einen *Relevanz*-Parameter bewertet wurde. Mit der im Folgenden vorgestellten Modifikation ist ein solcher Parameter nicht mehr notwendig.



**Abb. 2:**  
Routing-Graph (Stand 2011)

## 2.2 OpenStreetMap

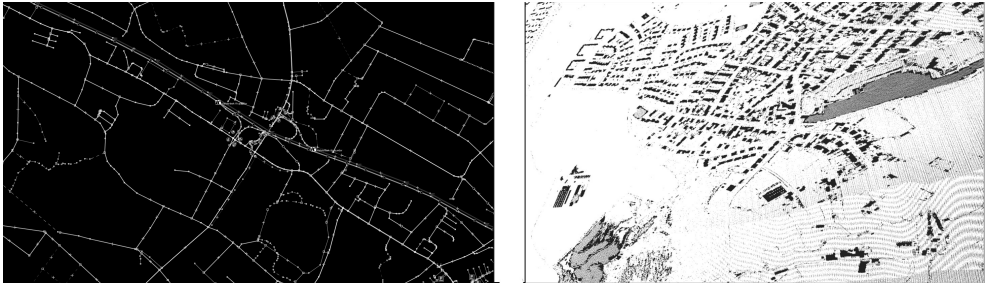
Bei dem Versuch frei verfügbares Kartenmaterial für die Zwecke von eNav zu finden, ist erkennbar geworden, dass es keine gleichwertigen freien Alternativen zu OpenStreetMap (OSM) gibt. Die Datendichte der Weltkarte sowie die Verlässlichkeit der Daten ist unseren Erfahrungen nach in keinem anderen frei verfügbaren Open-Source-Projekt derart hochwertig (RAMM & TOPF 2010). Zusätzlich bietet OSM eine gute Basis, um die Daten, welche im XML-Format vorliegen, um eigene Informationen zu erweitern. Dazu gibt es *Tags*, die eine Eigenschaft beschreiben und in Form eines *Key-Value Paares* in den OSM-Dateien abgespeichert werden. Für das eNav Projekt sind besonders die Tags mit dem Key *Altitude* und *Elevation* nützlich, welche Höheninformationen zu Knoten speichern. Diese spielen bei der Modifikation des A\*-Algorithmus im Folgenden eine wichtige Rolle.

## 2.3 Laserdaten

Die in OSM in der Regel fehlenden bzw. ungenauen Höhenangaben müssen für das eNav-Projekt anderweitig ersetzt oder ergänzt werden. Die Ergebnisse eines geographischen Laserscans der Bezirksregierung in Köln sind für das eNav-Projekt an der RWTH Aachen gut geeignet (BEZIRKSREGIERUNG KÖLN 2010). Die Nutzung der Daten um Aachen herum ist ausschließlich für Forschungszwecke gestattet. Abbildung 3 präsentiert einen Kartenabschnitt mit OSM-Daten (links, visualisiert mit JOSM<sup>1</sup>) und den gleichen Abschnitt mit den Lasermessdaten der Bezirksregierung (rechts, visualisiert mit OpenFlipper<sup>2</sup>).

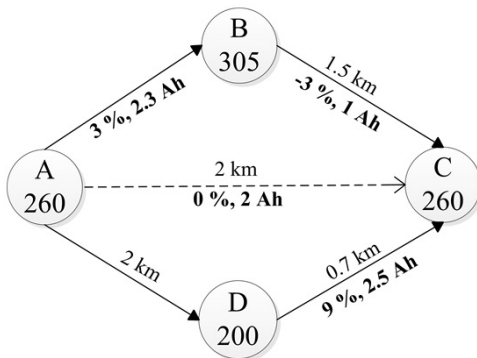
<sup>1</sup> <http://josm.openstreetmap.de/>

<sup>2</sup> <http://www.openflipper.org/>



**Abb. 3:** Aachen in OSM- und Laserscandarstellung

Die Lasermessdaten werden von einem Flugzeug aus mithilfe eines Laserstrahls, der die Erdoberfläche mit einer Laserabtastung von ca. 200 kHz abtastet, alle 4-5 Jahre erneut vermessen. Die Sensoren im Flugzeug registrieren die Reflexion des Lasers an der Erdoberfläche. Der relative Abstand zwischen dem Lasermessgerät im Flugzeug und der Erdoberfläche wird anschließend anhand der vergangenen Zeit bestimmt. Ein Vorteil der Lasermessdaten ist die Genauigkeit der Datenpunkte, welche mit einer Abweichung von  $\pm 20$  cm angegeben wird. Dies ist für die Zwecke von eNav zufriedenstellend und stellt im Vergleich zu GPS-Daten eine enorme Verbesserung dar, da die Abweichung der OSM-Daten mit  $\pm 5$  m angegeben wird. Sobald man beide Datensätze hat, OSM- und Lasermessdaten, müssen diese noch zusammengeführt werden, um später als Grundlage für den modifizierten A\*-Algorithmus dienen zu können. Ein Parser übernimmt diese Aufgabe und generiert aus der Punktwolke des Laserscans und den entsprechenden Knoten der OSM-Datenbank dreidimensionales Kartenmaterial. Diese Veränderung führt zu einer Änderung des Routinggraphen aus Abbildung 2 (vgl. Abb. 4). Da nun auch in den Knoten Höheninformationen vorhanden sind, können die Streckensteigungen zwischen den Knoten berechnet werden.



**Abb. 4:**  
Routing-Graph (Stand 2013)

### 3 3D-Routing mit eNav

In einem ersten Schritt zeigen wir, dass die Berechnung der effizientesten Route auf dreidimensionalem Kartenmaterial ohne eine Anpassung des A\*-Algorithmus zu einer fehlerhaften Berechnung führt. Wie im Folgenden klar wird, ist ein wichtiger Grund hierfür die

Verfügbarkeit der Höhendaten der Knoten (im Gegensatz zu dem in (Franke et al. 2011) vorgestellten Ansatz).

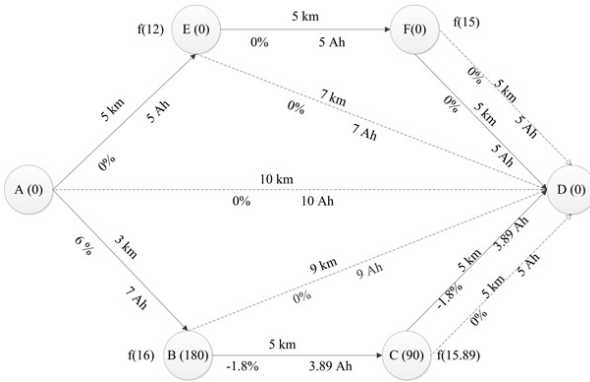


Abb. 5.1: OSM Beispiel

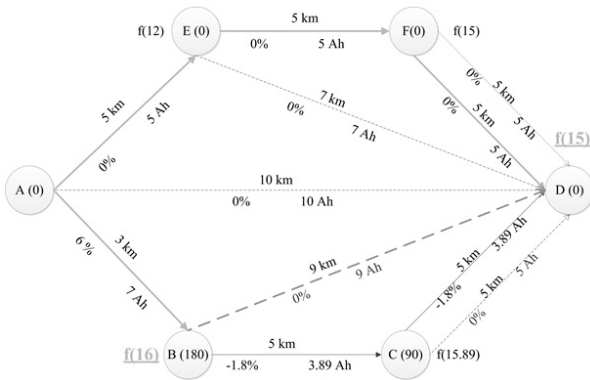


Abb. 5.2: OSM Beispiel

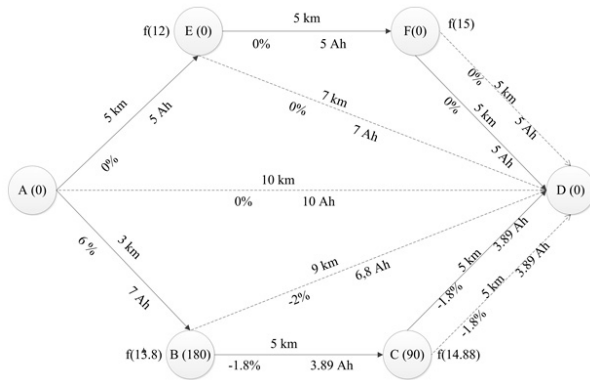


Abb. 5.3: Dreidimensionaler Lösungsansatz

Die gestrichelten Linien stellen die Luftlinien dar, bei denen eine Steigung von 0 % angenommen wird, da Höheninformationen nicht in den Knoten vorhanden sind. Angenommen es soll eine Route von A nach D gefunden werden (siehe Abb. 1). Dann wird aufgrund der fehlenden Steigung der Luftlinie von A nach D über E und F geroutet. Da die errechnete f-Funktion bei B 16 beträgt und somit 15 bei F überbietet, wird jedoch die effizientere Route von A nach D über B und C nicht gefunden. Der Grund dafür ist, dass der Punkt B höher als der Punkt D liegt, was man jedoch nur mit Höheninformationen in den Knoten erkennen kann (siehe Abb. 2). Zu beachten ist, dass aufgrund der Kanteninformationen die Höhe nicht errechnet werden kann, da die Route nicht vorab bekannt ist.

Um dieses Problem zu lösen, wird dreidimensionales Routing eingeführt. Für die Berechnung der Luftlinie werden die 3D-Koordinaten der Knoten betrachtet. Aufgrund der Höheninformationen und der Entfernung kann mittels trigonometrischen Formeln die Steigung berechnet werden. Mit dieser Methode beträgt die Steigung der Luftlinie von B nach D -2 % statt 0 % (siehe Abb. 5.3).

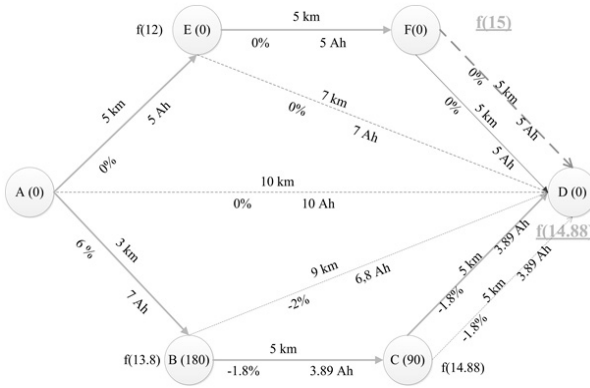


Abb. 5.4: Dreidimensionaler Lösungsansatz (A→D)

Das Routen von A nach D erfolgt über die Zwischenkonten B und C und nicht mehr über E und F. Der Grund für diese Änderung ist die Berechnung der Luftlinie, welche für die Strecke zwischen den Knoten B und D ein Gefälle von -2 % annimmt. Dadurch gestaltet sich der Verbrauch geringer als bei Annahme einer waagerechten Luftlinie. Somit wird der Fehler des ersten Szenarios behoben.

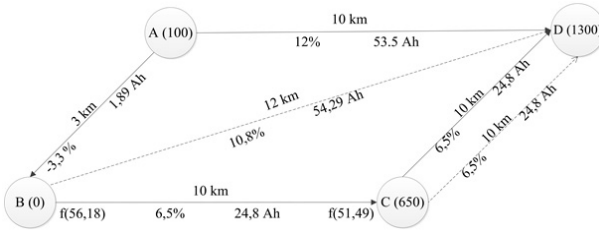


Abb. 5.5: Problem beim 3D Routing

Dieser 3-dimensionale Lösungsansatz ist jedoch noch immer nicht optimal. Das zweite Szenario (siehe Abb. 5.5) verdeutlicht, dass es möglich ist, sich erst weiter vom Zielknoten zu entfernen und danach eine effizientere Route zu berechnen. Wie im Kapitel Grundlagen bereits beschrieben, besteht die Möglichkeit, dass eine lange Route mit geringer Steigung weniger Energie verbraucht als eine kurze Route mit einer großen Steigung. Somit kann der Fehler, der Abbildung 4.6 zu entnehmen ist, entstehen: Der direkte Luftlinienverbrauch von B nach D weist einen größeren Verbrauch auf als der von A nach D auf direktem Wege. Da der A\*-Algorithmus auf Basis der f-Funktion entscheidet, wird die direkte Route gefunden und nicht die effizienteste.

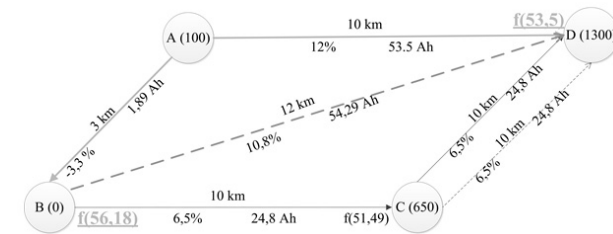


Abb. 5.6: Problem beim 3D Routing (A→D)

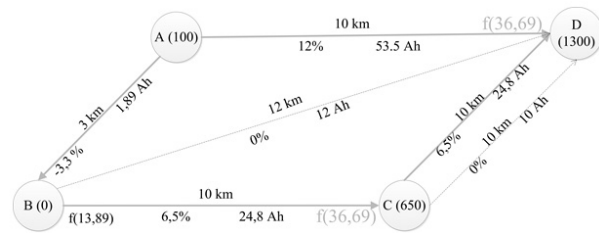


Abb. 5.7: Behebung des Problems

Um das beschriebene Problem zu beheben, wird für die obere Schranke die Luftliniensteigung gleich 0 gesetzt.

Dadurch wird die Heuristik wie folgt definiert: Bei einem Gefälle der Luftlinie wird die Steigung mit in den Verbrauch einberechnet. Falls der Zielknoten höher liegt als der Kno-

ten, von dem aus die Luftlinie ermittelt wird, nimmt man für den Verbrauch eine Steigung von 0 % an.

Daher wird auch in diesem Szenario die effizienteste Route als solche gefunden und ausgewählt (siehe Abb. 5.7). Im folgenden Kapitel werden die beschriebenen Schranken getestet und validiert.

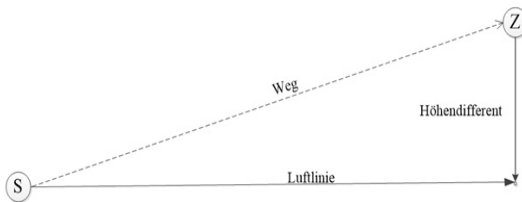
## 4 Beweis

An dieser Stelle wird die in Kapitel 3 vorgestellte Heuristik bewiesen. Zu beweisen ist, dass die beiden Schranken nicht unterschritten werden können und somit eine gültige Abbruchbedingung für den A\*-Algorithmus existiert. Ohne einen Beweis der Korrektheit der Schranken bestünde die Möglichkeit, dass sich der eigentliche A\*-Algorithmus fehlerhaft verhält. Dabei wird die Gültigkeit der oberen und der unteren Schranke gezeigt. Wir beginnen mit der oberen Schranke:

Wir definieren (*Ist-Verbrauch*)  $f = \text{Normalverbrauch} * \text{Distanz} * 1,15^{\text{Steigung}}$ , für Steigung  $> 0$

$l_0$  (*Luftlinien*) =  $\text{Normalverbrauch} * \text{Distanz} * 1,15^0$ , für Steigung = 0

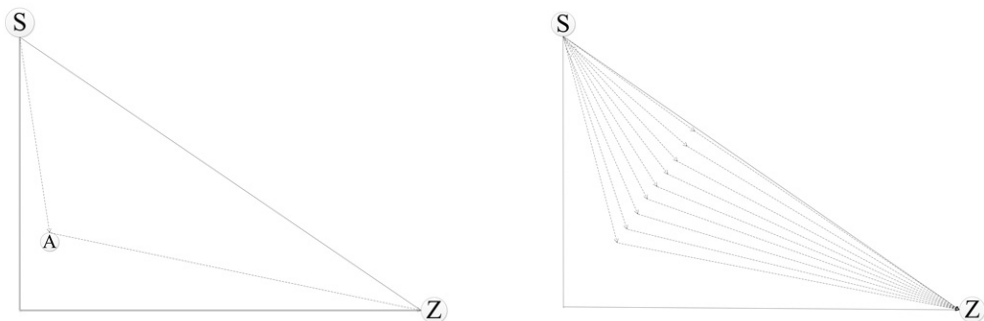
Zu zeigen ist:  $f(x) > l_0(x)$



**Abb. 6:**  
Luftlinie

Diese Ungleichung lässt sich graphisch belegen (siehe Abb. 6). Angenommen wir wollen von Knoten S zum Knoten Z routen, wobei der Zielknoten Z höher liegt als S. Dann lässt sich aufgrund von trigonometrischen Gesetzen zeigen: Sei  $l_0$  die Ankathete und  $f$  die Hypotenuse. Aufgrund der Höhendifferenz gilt somit mit Satz des Pythagoras (GERICKE 1984), dass  $f(x) > l_0(x)$ . Zudem ist zu erwähnen, dass durch die Annahme der Steigung größer als 0 % auch der Faktor  $1,15^{\text{Steigung}}$  größer wird (FRANKE et al. 2011).

Die Betrachtung der unteren Schranke lässt sich in zwei Fälle unterteilen.



**Abb. 7:** Luftlinie mit unterschiedlichem Gefälle

Fall 1: Es ist zu überprüfen, ob es eine effizientere Route geben kann, als die direkte Route zwischen zwei Knoten. Um diesen Beweis mit vollständiger Induktion nachzuweisen, führen wir einen Zwischenknoten A zwischen den Startknoten S und den Zielknoten Z ein. Durch eine Computersimulation, die den Knoten A in der Höhe variiert und zu jeder möglichen Höhe (mit einer Granularität von 0,1 mm) die Effizienz (den Verbrauch) der Route berechnet, weisen wir programmatisch nach, dass es keine effizientere Route zwischen S und Z gibt. Dazu ist zu beachten, dass laut der Straßenverkehrsordnung die maximale Steigung für von Autos befahrene Straßen  $\pm 12\%$  nicht übersteigt (FORSCHUNGSGESELLSCHAFT FÜR STRAßEN UND VERKEHRSWESSEN 1995). Die Simulation ergibt (siehe Abb. 7), dass unter den genannten Voraussetzungen keine Route effizienter ist, als die direkte von S nach Z. Noch bleibt zu zeigen, dass es keine Route gibt, die effizienter ist und mehrere Zwischenknoten benutzt. Dazu führen wir folgenden Gegenbeweis mithilfe der Induktion:

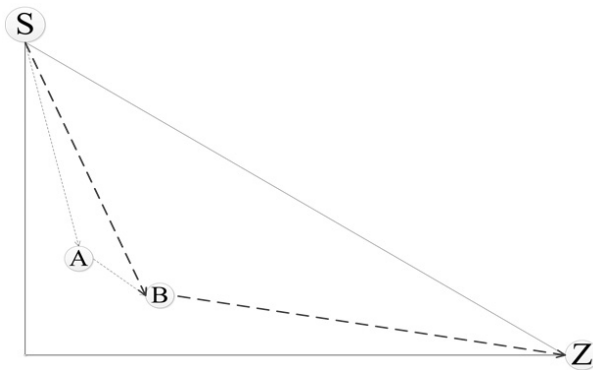
Angenommen es gäbe zwei Zwischenknoten A und B (siehe Abb. 8) für die gilt:

$$f_{(S,A)} + f_{(A,B)} + f_{(B,Z)} < f_{(S,Z)}$$

Wie bereits gezeigt, ist ein Zwischenknoten A nicht effizienter als die direkte Route von S nach B.

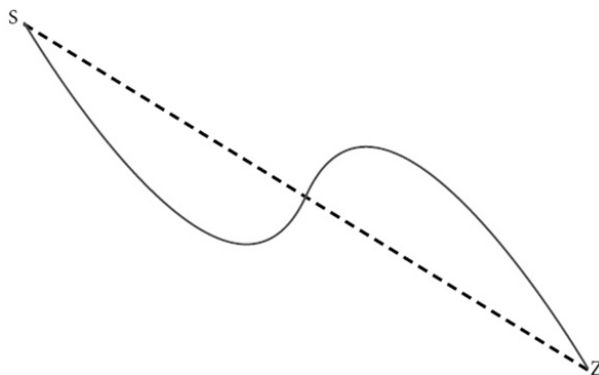
Somit gilt  $f_{(S,A)} + f_{(A,B)} \geq f_{(S,B)}$  und damit auch:

$$f_{(S,B)} + f_{(B,Z)} \leq f_{(S,A)} + f_{(A,B)} + f_{(B,Z)} < E_{(S,Z)}$$



**Abb. 8:** Einfügen eines weiteren Zwischenknotens

Damit ergibt sich ein Widerspruch zur Simulation. Aufgrund dessen lässt sich über Induktion zeigen, dass eine Zunahme einer beliebigen Anzahl von Punkten nie zu einer effizienteren Route führen kann als die direkte Route. Die Anzahl der Kanten lässt sich durch Induktion immer auf zwei reduzieren. Es ist zu beachten, dass sich aufgrund der exponentiellen Entladung eines Akkus nicht darauf schließen lässt, dass der direkte Weg der effizienteste ist.



**Abb. 9:** Slalomparcour

Fall 2: Angenommen es handelt sich um eine slalomartige Route, sodass das Gefälle flacher wird, die Distanz allerdings zunimmt. Sei die Verbrauchsfunktion der direkten Route  $\text{Normalverbrauch} * \text{Distanz} * 1,15^{\text{Steigung}}$ , dann führte eine flachere Steigung zum Anwachsen der Distanz und des Faktors mit der



Steigung im Exponenten. Somit kann eine slalomartige Route auch nicht effizienter sein, als die direkte Route.

Insgesamt ist hiermit der Beweis programmatisch mithilfe der vollständigen Induktion abgeschlossen und die Korrektheit der oberen und unteren Schranke gezeigt.

## 5 Fazit und Ausblick

Obwohl zur Berechnung des Verbrauchs auf den einzelnen Streckenabschnitten nur die Streckensteigung notwendig ist, sind die Höheninformationen der Streckenknoten für eine korrekte Berechnung der effizientesten Route mit dem A\*-Algorithmus notwendig. Dabei spielen sie bei der Berechnung der Luftlinien während der Laufzeit des A\*-Algorithmus eine Schlüsselrolle. Dazu sei zu erwähnen, dass es sich nicht um eine bloße Auswechslung der Kantengewichte handelt, da der Algorithmus sonst, wie im Beweis gezeigt, nicht korrekt ausgeführt werden könnte. Die heuristische Funktion muss sich deshalb von derjenigen unterscheiden, die bspw. zur Berechnung der schnellsten Route verwendet wird. Somit ermöglicht eine korrekte Heuristik mithilfe dieser Luftlinie das Routen in 3D-Räumen. Die Ergebnisse dieser Arbeit können auch für weitere Projekte, in denen der A\*-Algorithmus zur Berechnung von Routen in 3D-Räumen zum Einsatz kommt, verwertet werden. Der modifizierte A\*-Algorithmus steht ab sofort auf allen üblichen eNav-Clients (Android, Web, ...) zur Verfügung. Als nächsten Schritt sehen wir die Herausforderungen an, welche sich durch die Berücksichtigung der Streckenbeschaffenheit bei der Berechnung der effizientesten Route ergeben, und planen diesen ebenfalls in eNav zu integrieren. Anschließend ist eine Kopplung von eNav mit rollstuhlrouting<sup>3</sup> und wheelmap<sup>4</sup> geplant.

Diese Arbeit wurde durch das UMIC Research Centre, RWTH Aachen Universität, unterstützt.

## Literatur

- BEZIRKSREGIERUNG KÖLN (2011), Topographische Reliefinformationen, Köln.
- FORSCHUNGSGESELLSCHAFT FÜR STRABEN UND VERKEHRSWESEN (1995), Richtlinien für die Anlage von Straßen – Teil: Linienführung. Berlin.
- FRANKE, D., DZAFIC, D., WEISE, C. & KOWALEWSKI, S. (2011), Entwicklung eines mobilen Navigationssystems für Elektrofahrzeuge auf Basis von OpenStreetMap-Daten. In: Konferenz für Freie und Open Source Software für Geoinformationssysteme (FOSS-GIS), Heidelberg, 92-99.
- FRANKE, D., DZAFIC, D., WEISE, C. & KOWALEWSKI, S. (2011), Konzept eines Mobilen OSM-Navigationssystems für Elektrofahrzeuge. In: Symposium für Angewandte Geoinformatik (AGIT), Salzburg, Österreich, 2011, 148-157.

---

<sup>3</sup> [www.rollstuhlrouting.de](http://www.rollstuhlrouting.de)

<sup>4</sup> [www.wheelmap.org](http://www.wheelmap.org)

- 
- FRANKE, D., DZAFIC, D., BAUMEISTER, D. & KOWALEWSKI, S. (2012), Energieeffizientes Routing für Elektrorollstühle. In: 13. Aachener Kolloquium Mobilität und Stadt (AMUS/ACMOTE), Aachen, 65-67.
- GERICKE, H. (1984), Mathematik in Antike und Orient. Berlin.
- HART, P. E., NILSSON, N. J. & RAPHAEL, B. (1968), A Formal Basis for the Heuristic Determination of Minimum Cost Paths. In: IEEE (Ed.), Transactions on Systems Science and Cybernetics. Menlo Park, California, USA.
- NORWIG, P. & RUSSELL, S. (2011), Künstliche Intelligenz – Ein Moderner Ansatz. München.
- RAMM, F. & TOPF, J. (2010), OpenStreetMap – Die freie Weltkarte nutzen und mitgestalten. Berlin.