

Routing über Flächen mit SpiderWebGraph

Dzenan Dzafic, Stefan Klug, Dominik Franke und Stefan Kowalewski

Lehrstuhl Informatik 11, Embedded Software, RWTH Aachen University ·
dzafic@embedded.rwth-aachen.de

Full paper double blind review

Zusammenfassung

Navigationssysteme und Routenplaner können problemlos über Graphen mit Knoten und Kanten die kürzeste, schnellste oder energieeffizienteste Route finden. In dem Fall, dass sich auf der Route zum Ziel ein Polygon befindet, wird dieses umfahren. In einem realen Beispiel sieht dies wie folgt aus: Angenommen ein Fußgänger will von A nach B über einen Marktplatz oder großen Parkplatz gehen. Ein herkömmlicher Routenplaner bzw. ein herkömmliches Navigationssystem führt den Fußgänger um die Fläche herumgeführt statt hinüber. Dieser Beitrag befasst sich mit der Herausforderung, effizient über freie Flächen zu routen. Als Erstes wird das Routingverhalten von acht etablierten Routenplanern untersucht. Dann wird ein Lösungsansatz (BAUER et al. 2014) vorgestellt und evaluiert. Danach wird im Kernteil dieses Papers der SpiderWebGraph (SWG) vorgestellt, welcher sowohl als Alternative als auch als Ergänzung zum vorgestellten Algorithmus (BAUER et al. 2014) zu betrachten ist. Abschließend geht das Paper auf die Vorteile des SpiderWebGraph für das vorhandene eNav-System (DZAFIC et al. 2014) ein und verdeutlicht die Herausforderungen und Implikationen für das energieeffiziente Routen über Polygone.

1 Einleitung

Durch die stetig steigende Zunahme der Smartphonennutzung in der Gesellschaft, nutzen immer mehr Fußgänger und Rollstuhlfahrer Navigations-Apps. Was jedoch viele Navigationshersteller vernachlässigen, ist, dass die Fußgänger und Rollstuhlfahrer auch Flächen überqueren können, wie z. B. quer über einen Marktplatz oder andere Flächen. Zurzeit werden Fußgänger von den etablierten Routenplanern um überquerbare Flächen herum geleitet (siehe Abb. 1 durchgezogene Linie), da in dem zugrunde liegenden Kartenmaterial nur in vereinzelt Fällen quer durch die Fläche führende Wege enthalten sind. Dies kann mehrere Ursachen haben. Eine mögliche Ursache ist die konzeptuelle Entwicklung von Navigationssystemen für motorisierte Fahrzeuge, welche nur über Straßen und nicht über Flächen geleitet werden sollen. Es besteht eine Möglichkeit, Fußgänger- und Fahrradprofile auszuwählen, jedoch werden die Anforderungen dieser Zielgruppen noch nicht konsequent umgesetzt.

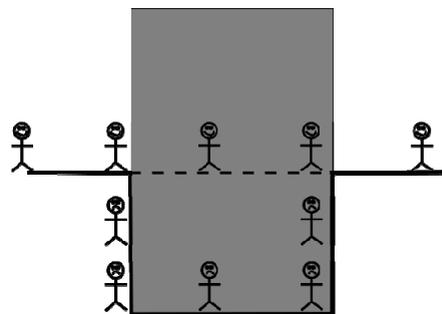


Abb. 1: Routen über freie Flächen

Eine weitere Ursache für die Vernachlässigung des Flächenroutings ist wahrscheinlich, dass Fußgänger intuitiv freie Flächen überqueren. Dies wird von den Herstellern als selbstverständlich angesehen. Jedoch kann vorkommen, dass aufgrund der Zoomstufe und der geringen Displaygröße eines Smartphones nicht immer sichtbar ist, dass der direkte Weg genommen werden kann. Eine solche inkorrekte Anzeige kann durch das in diesem Paper vorgestellte Verfahren vermieden werden. Unter einer inkorrekten Anzeige ist zu verstehen, dass Fußgänger nicht quer über Flächen, sondern an deren Rändern entlang geleitet werden. Abgesehen von der inkorrekten Anzeige des kürzesten Weges erfolgt auch eine inkorrekte Berechnung des kürzesten Weges. Im späteren Verlauf wird anhand eines Beispiels in Kapitel 5 die inkorrekte Berechnung gezeigt. Der Algorithmus von (BAUER et al. 2014) befasst sich mit den visuellen Mängeln und löst sie effizient. Der SpiderWebGraph, welcher in diesem Paper vorgestellt wird, konzentriert sich auf die korrekte Berechnung der Route.

2 Grundlagen

Bevor die oben genannten Herausforderungen behandelt werden, erfolgen in den Unterkapiteln zunächst Begriffserläuterungen, welche für das Verständnis dieses Paper von Bedeutung sind. Diese beschränken sich im Sinne der Seitenvorgabe auf das Wesentliche.

2.1 A*-Algorithmus

Der A*-Algorithmus berechnet den kürzesten Pfad von A nach B. Die Vorgehensweise ist mit Dijkstra zu vergleichen. Zusätzlich zu den Kantengewichten verwendet der A*-Algorithmus eine Heuristik verwendet, um die Suche zielgerichtet zu leiten, wodurch die Laufzeit verringert wird. Ein Beispiel für eine solche Heuristik ist die Luftlinie zwischen den Zwischenknoten und dem Ziel (HART et al. 1968, NORWIG & RUSSELL 2011).

2.2 Graph

In diesem Paper ist der Begriff Graph stets als gerichteter Graph zu verstehen. Dieser ist wie folgt definiert:

Ein gerichteter Graph $G = (V, E)$ besteht aus einer Knotenmenge V und einer Menge von gerichteten Kanten E , sodass jeder gerichteten Kante $e = (u, v)$ eindeutig ein geordnetes Paar (u, v) von Knoten aus V zugeordnet ist. Der Knoten u heißt der Anfangsknoten, v der Endknoten der Kante $e = (u, v)$ (TITTMANN 2011, OTTMANN & WIDMAYER 2012).

2.3 Korrektheit

Das Wort „korrekt“ wird in diesem Paper häufig verwendet. Es ist damit Folgendes gemeint:

*Mit einer korrekt gefundenen Route ist gemeint, dass die Summe der Kantengewichte der gefundenen Route (**kürzeste Route**) nicht durch eine andere Route unterboten werden kann. Auch nach der Betrachtung von Flächen muss die gefundene Route noch stets die kürzeste sein.*

Abbildung 6 zeigt ein Beispiel, in welchem nicht die kürzeste Route gefunden wird.

2.4 OpenStreetMap

Das Projekt OpenStreetMap (OSM) wurde 2004 mit der Zielsetzung eine frei verfügbare Weltkarte zur Verfügung zu stellen gestartet (RAMM & TOPF 2010). Der Unterschied zu anderen Kartendatenbanken besteht darin, dass bei dem Projekt jeder angemeldete Benutzer Karteninformationen in die Karte einpflegen kann. Diese Informationen können in Form einer XML-Datei von OSM heruntergeladen werden. So eine XML-Datei besteht hauptsächlich aus drei verschiedenen Arten von Elementen: Nodes, Ways und Relations. Nodes stellen in OSM Knoten mit eindeutigen Koordinaten dar. Sie können außerdem noch Zusatzinformationen in Form von Key-Value-Paaren, den sogenannten Tags, enthalten. Ways verbinden mehrere Nodes. Auch sie können durch Tags mit weiteren Informationen versehen werden. Relations beschreiben die Beziehungen zwischen mehreren Ways und Nodes. Begehbare Flächen werden in OSM mit zirkulären Ways und dem Tag „area=yes“ gekennzeichnet. Durch die Flächen können vereinzelt Wege verlaufen, die es gängigen Routenplanern teilweise über Flächen zu routen. Dadurch eignet sich das Datenmaterial von OSM besonders gut für das Routen über Flächen.

2.5 eNav

Das Projekt eNav ist ein Navigationssystem mit dem Ziel, die Mobilität von Nutzern von Elektrorollstühlen zu steigern, indem die Berechnung von energieeffizienten Routen eine bessere Ausnutzung der Akkukapazität erlaubt. Neben der Mobilität steht auch der Komfort im Fokus. Um die Energieeffizienz zu steigern, berücksichtigt der Routenplaner von eNav zusätzliche Faktoren. Zurzeit fließen neben der Straßenlänge auch die Steigung und der Bodenbelag mit in die Berechnung der Route ein. Eine Untersuchung hat ergeben, dass sich der Energieverbrauch exponentiell zur Steigung verhält (FRANKE et al. 2011). Als Folge dessen ergibt sich, dass oftmals Routen existieren, welche im Vergleich zur kürzesten Route länger dafür jedoch energieeffizienter sind. Des Weiteren ist der Energieverbrauch des Rückwegs oft ein anderer als der des Hinweges. Durch die Berücksichtigung des Bodenbelags kann neben dem Reibungswiderstand auch der Komfort erhöht werden. Der Benutzer kann zusätzlich zur Wahl von Start und Ziel auch Bodenbeläge ausschließen. Somit kann beispielsweise das unbequeme Schütteln, welches Kopfsteinpflaster verursacht, vermieden werden (DZAFIC et al. 2014, FRANKE et al. 2011).

3 State of the Art

Dieses Kapitel stellt die Ergebnisse einer Untersuchung von acht etablierten Routenplanern bezüglich ihres Verhaltens auf routen über Flächen dar. Als Untersuchungsort wird der Marktplatz von Aachen verwendet. Mit dem Ziel von der linken auf die rechte Seite zu gelangen (siehe Abb. 2). Wie zu erkennen, ist es keinem der aufgeführten Routenplanern gelungen, über den Marktplatz zu routen. Es ist zu erwähnen, dass bei allen, die über ein Fußgängerprofil verfügen, dieses gewählt wurde.



1. <http://www.bing.com/maps>



2. <http://www.falk.de>



3. <https://www.google.de/maps>



4. <http://www.mapandroute.de>



5. <http://www.viamichelin.de>



6. <http://en.mappy.com>



7. <http://openrouteservice.org>



8. <https://de.maps.yahoo.com>

Abb. 2: Routenplanervergleich

4 Related Work

Auf der AGIT 2014 stellt (BAUER et al. 2014) einen Algorithmus vor, welcher sich derselben Herausforderung wie dieses Paper stellt und präsentiert eine visuelle Lösung, wodurch der Pfad auf dem Routenplaner den Marktplatz überquert wird anstatt ihn zu umfahren. Dafür wird das Kartenmaterial von OSM verwendet. Jedoch findet die visuelle Korrektur nach der Berechnung der Route statt und hat somit keinen Einfluss auf die gewählte Route.

Das folgende Beispiel zeigt die Vorgehensweise des Algorithmus (siehe Abb. 3). Angenommen ein Benutzer will von S nach T. Als erstes findet der A*-Algorithmus eine kürzeste Route und liefert die relevanten Knoten und Kanten zurück. Danach geht der Algorithmus von (BAUER et al. 2014) die Route rückwärts entlang und überprüft, ob Knoten entlang eines Polygons verlaufen. Dies ist möglich, da in OSM Polygone durch Ways mit dem Untertag „area=yes“ beschrieben werden. Wenn sowohl der Nachfolger als auch der Vorgänger Polygonknoten sind, werden der Knoten und die entsprechenden Kanten aus dem Graphen entfernt. Vorgänger- und Nachfolgerknoten werden durch eine neue Kante miteinander verbunden (siehe Abb. 4). Der beschriebene Vorgang wird so oft wiederholt, bis ein direkter Weg über das Polygon verläuft (siehe Abb. 5). Die verbleibende Route wird dann auf dem Display so gezeichnet, dass der Benutzer über das Polygon geleitet wird. Der Algorithmus funktioniert korrekt, solange der in Schritt 1 gefundene Pfad tatsächlich der kürzeste ist. Die Abbildung 6 zeigt ein Gegenbeispiel.

Für den Fall, dass der in Schritt 1 gefundene Pfad nicht der kürzeste ist, kommt der SpiderWebGraph zum Zuge.

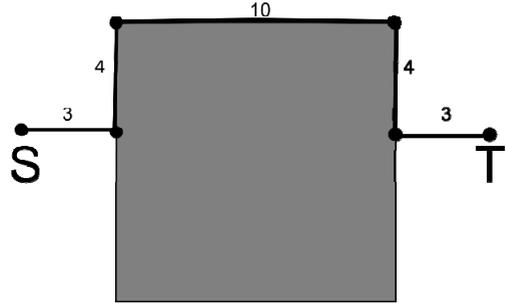


Abb. 3: Erster Schritt

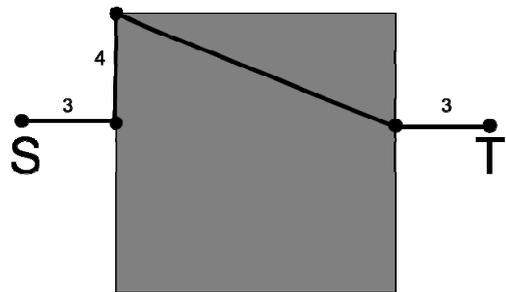


Abb. 4: Zweiter Schritt

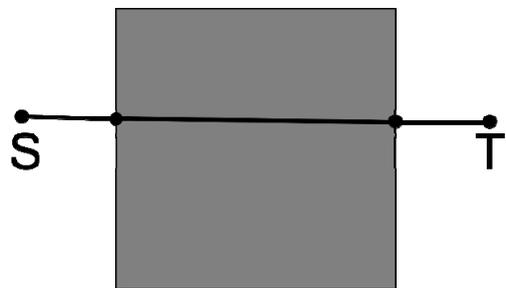


Abb. 5: Dritter Schritt

5 SpiderWebGraph

Bevor der SpiderWebGraph (SWG) vorgestellt wird, folgt ein Beispiel zur Verdeutlichung, warum die gängigen Routenplaner nicht nur falsche Anzeigen, sondern auch falsche Routen liefern können.

Angenommen, es liegt ein Fall wie in Abbildung 6 vor. Der Benutzer müsste wieder von S nach T und es existiert eine zweite Route, welche nicht entlang des Polygons verläuft (siehe gestrichelte Linie in Abb. 6). Der A*-Algorithmus sucht den kürzesten Pfad und findet den gestrichelten Pfad, welcher eine Gesamtlänge von 20 Metern hat. Der Weg entlang des Polygons wird verworfen, da dieser mit 24 Metern länger als der gestrichelte ist. Als nächstes folgt Schritt zwei des Algorithmus von (BAUER et al. 2014), welcher den Pfad entlang geht und nach Polygonknoten sucht. Da kein Polygonknoten auf dem Pfad ist, wird der Algorithmus die vollständige Route unverändert zurückgeben. Der Benutzer wird dann über eine Route von 20 Metern geführt, obwohl er mit 16 Metern über die Fläche gelangen kann (siehe Abb. 5).

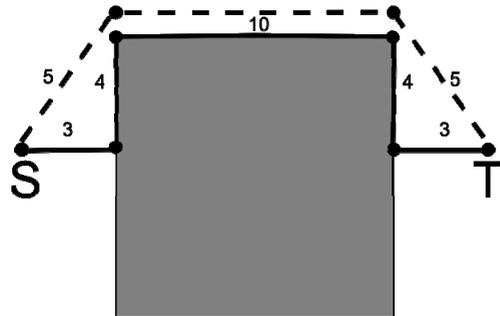


Abb. 6: Gegenbeispiel

Um dieses Phänomen zu beheben, erzeugt der SWG-Algorithmus einen neuen Graphen. Bei diesem wird eine Art Spinnennetz über dem Polygon erzeugt (siehe Abb. 7), sodass auf der Fläche viele Zwischenknoten und Kanten entstehen. Diese dürfen nur eingefügt werden, wenn sich dort kein Hindernis wie beispielsweise ein Brunnen befindet. Dabei können natürlich nur statische Hindernisse, welche in das Kartenmaterial integriert sind, betrachtet werden. Bewegliche Hindernisse wie parkende Autos werden nicht beachtet. Die neu erstellten Knoten werden mit allen anderen Knoten in einem fest definierten Radius verbunden. Dadurch entsteht eine spinnennetzartige Struktur. Somit wird das Spinnennetz automatisch mit den OSM-Knoten am Rand des Polygons verbunden.

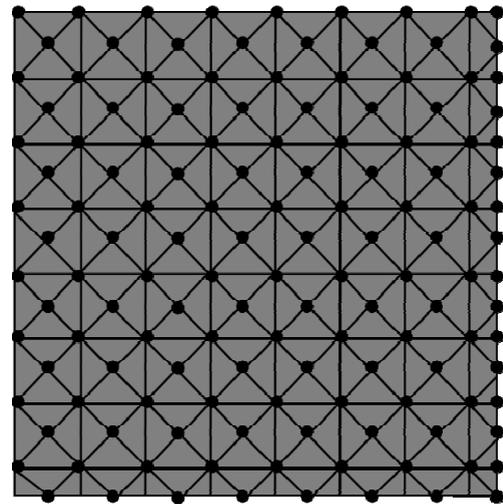


Abb. 7: Spinnennetz

Eine existierende Anbindung an das Straßennetz wird für das Routing vorausgesetzt. Durch das Auffüllen der Polygone mit Knoten und Kanten ergibt sich für den A*-Algorithmus die Möglichkeit, über Flächen zu routen.

Bei Anwendung des SWG-Algorithmus auf das Beispiel aus Abbildung 6 kann nun die tatsächlich kürzeste Route berechnet werden, welche nicht mehr eine Länge von 20 Metern (gestrichelte Linie) sondern von nunmehr 16 Metern besitzt. Die Genauigkeit des Algorithmus hängt mit der Dichte des Netzes zusammen. Als Konsequenz ergibt sich, dass die Korrektheit des kürzesten Pfades noch immer nicht ganz gegeben ist. Nichtsdestotrotz liefert der Ansatz des SWG immer noch eine höhere Berechnungsgenauigkeit als die bisherigen Ansätze. Angenommen die Kante an der rechten Polygonseite in Abbildung 9 ist 1 Meter höher als die Kante an der linken Seite, dann liefert der Algorithmus eine Gesamtlänge von 16,4 Metern. Dieses Ergebnis ergibt sich wie folgt:

$$3 + \sqrt{1+1} + 10 - 1 + 3 = 16,4$$

Die Wurzel 2 kommt durch den ersten Teil der Diagonalen zustanden. Dadurch muss der eine Meter von den 10 abgezogen werden und dann noch jeweils die 3 Meter der Seitenkanten. Die korrekte Länge müsste jedoch 16,04 Meter (siehe Abb. 10) sein:

$$3 + \sqrt{1+100} + 3 = 16,04$$

Die Berechnung erklärt sich analog zu obiger.

Durch Abbildung 9 und 10 wird ersichtlich, dass eine Kombination aus dem Algorithmus von (BAUER et al. 2014) und dem SWG-Algorithmus zu einem noch besseren Ergebnis als die alleinige Verwendung von SWG (siehe Abb. 13) führt. Das heißt, im ersten Schritt wird aus dem OSM-Graphen ein neuer Graph mit der Netzstruktur erstellt. Im zweiten Schritt findet der A*-Algorithmus darauf den kürzesten Pfad und abschließend erfolgt eine Korrektur der angezeigten Route (siehe Abb. 11) durch den Algorithmus von (BAUER et al. 2014).

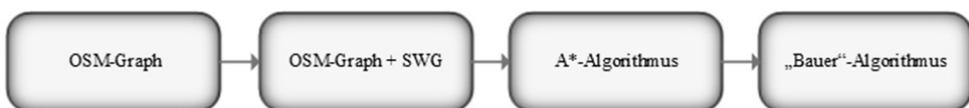


Abb. 11: Kombination von Bauer und SWG

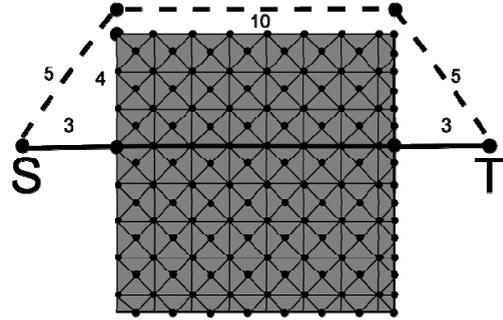


Abb. 8: Routing mit SWG

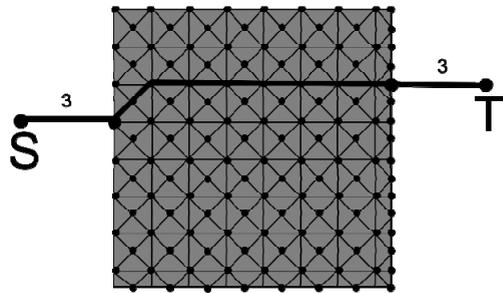


Abb. 9: SWG-Abweichung

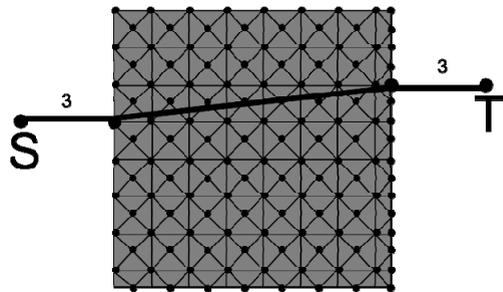


Abb. 10: Korrekte Route

eNav-System die Möglichkeit, spezielle Bodenbeläge wie großes Kopfsteinpflaster auszuschließen.

7 Fazit und Ausblick

Zu Beginn erfolgt eine Evaluierung der gängigen Routenplaner mit dem Fokus auf Routen über Flächen. Das daraus resultierende Ergebnis zeigt, dass keiner der untersuchten Routenplaner sich der Herausforderung gestellt hat, einen Fußgänger über eine Fläche zu navigieren statt um diese herum, obwohl im Sommer 2014 auf der AGIT ein Lösungsansatz von Herrn Clemens Bauer vorgestellt worden ist. Ihm ist es gelungen, einen Algorithmus zu entwickeln, welcher eine gerade Linie durch ein Polygon findet. Jedoch muss erst eine Route mit einem Polygon vorhanden sein. Dieser Herausforderung widmet sich das vorliegende Paper mithilfe des SpiderWebGraphen. Die Idee dieses Algorithmus ist es, OSM-Polygone mit Knoten und Kanten so aufzufüllen, dass eine spinnnetzartige Struktur entsteht. Diese Struktur ermöglicht es dem A*-Algorithmus, Routen über begehbbare bzw. befahrbare Flächen zu berechnen. Die Genauigkeit der Routenberechnung hängt von der Dichte des Graphen ab. Abschließend wird eine Kombination von SpiderWebGraphen und (BAUER et al. 2014) vorgestellt, sodass die Vorteile beider Mechanismen zum Vorschein kommen (siehe Abb. 13). Die Nachteile beider Mechanismen werden aufgehoben, denn durch den SpiderWebGraph wird sichergestellt, dass die kürzeste Route gefunden wird. Der Algorithmus (BAUER et al. 2014) sorgt für eine glatte Zeichnung auf der Anzeige. Die kombinierte Variante lässt sich nicht immer anwenden. Im Falle von eNav, welches mithilfe von 3D-Routing Hügel umfährt, um Energie zu sparen, würde der Algorithmus (BAUER et al. 2014) einen Elektrorollstuhlfahrer auf direktem Weg über Hügel leiten.



Abb. 13: Kombination aus SWG und dem Bauer-Algorithmus

Dieser Beitrag zeigt, dass es mit kleinen Modifikationen des Routinggraphen möglich ist, über Polygone nahezu korrekt zu routen und diese Route dementsprechend auch anzuzeigen. Es bleibt abzuwarten, bis sich dieser Ansatz in der Realität etabliert.

Diese Arbeit wurde durch das UMIC Research Centre und Center for European Research on Mobility (CERM), RWTH Aachen University, unterstützt.

Literatur

- BAUER, C., AMLER, A., LADSTÄTTER, S. & LULEY, P. M. (2014), Optimierte Wegefindung für Fußgänger basierend auf vorhandenen OpenStreetMap-Daten. In: STROBL, J. et al. (Hrsg.), *Angewandte Geoinformatik 2014*. Berlin/Offenbach, 408-413.
- DZAFIC, D., BAUMEISTER, D., FRANKE, D. & KOWALEWSKI, S. (2014), Integration von Bodenbelagsinformationen zum energieeffizienten Routen von Elektrorollstühlen. In: STROBL, J. et al. (Hrsg.), *Angewandte Geoinformatik 2014*. Berlin/Offenbach, 451-460.
- DZAFIC, D., FRANKE, D., BAUMEISTER, D. & KOWALEWSKI, S. (2013), Modifikation des A*-Algorithmus für energieeffizientes 3D-Routing. In: STROBL, J. et al. (Hrsg.), *Angewandte Geoinformatik 2013*. Berlin/Offenbach, 414-423.
- FRANKE, D., DZAFIC, D., BAUMEISTER, D. & KOWALEWSKI, S. (2012), Energieeffizientes Routing für Elektrorollstühle. In: 13. Aachener Kolloquium Mobilität und Stadt (AMUS/ACMOTÉ), AACHEN, 2012, 65-67.
- FRANKE, D., DZAFIC, D., WEISE, C. & KOWALEWSKI, S. (2011), Konzept eines Mobilien OSM-Navigationssystems für Elektrofahrzeuge. In: STROBL, J. et al. (Hrsg.), *Angewandte Geoinformatik 2011*. Wichmann, Berlin/Offenbach, 148-157.
- HART, P. E., NILSSON, N. J. & RAPHAEL, B. (1968), A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE (Ed.), *Transactions on Systems Science and Cybernetics*. Menlo Park, California, USA.
- NORWIG, P. & RUSSELL, S. (2011), *Artificial Intelligence – A Modern Approach*. München.
- OTTMANN, T. & WIDMAYER, P. (2012), *Algorithmen und Datenstrukturen*. Heidelberg.
- RAMM, F. & TOPF, J. (2010), *OpenStreetMap – Die freie Weltkarte nutzen und mitgestalten*. Berlin.
- TITTMANN, P. (2011), *Graphentheorie – Eine anwendungsorientierte Einführung*. Leipzig.