

Navigation abseits von Straßen

Rainer Prüller¹, Mathias Schardt¹, Florian Schögg², Claudia Fösleitner³, Carmen Sommer³

¹Institut für Geodäsie, TU Graz · rainer.prueller@tugraz.at

²pentamap mapping services OG, Graz

³TeleConsult Austria GmbH, Graz

Zusammenfassung: Im Projekt RemoteNav („Remote sensing based navigation in remote areas“) wird ein Routingalgorithmus für Gebiete ohne Wege- und Straßeninfrastruktur entwickelt. Aufbauend auf Erdbeobachtungsdaten wie Satellitenbildern und detaillierten Oberflächenmodellen werden räumliche Indikatoren abgeleitet, welche die prinzipielle Erreichbarkeit von Orten beschreiben. Auf Basis dieser Indikatoren, die von Geländeabbrüchen über Vegetationsklassen bis zur Bodenbeschaffenheit reichen, werden bestehende graphenorientierte Routingalgorithmen erweitert. Die involvierten Bedarfsträger bestehend aus Bergrettungsorganisationen, Alpinpolizei und Partnern aus der Jagd- und Forstwirtschaft tragen zur praxistauglichen Umsetzung dieses Navigationssystems für unwegsames Gelände bei. Die entwickelte Navigationsanwendung wird in die bestehenden Applikationen Sarontar und Deermapper integriert.

Schlüsselwörter: Navigation, Fernerkundung, GPS, Laserscanning

Abstract: RemoteNav („Remote sensing based navigation in remote areas“) provides a routing algorithm for remote areas based on remote sensing data. Spatial indicators derived from earth observation data describe the accessibility of rough terrain, which is currently not implemented into existing routing algorithms. User requirements based on knowledge of the two involved user groups of alpine rescue teams and hunters provide detailed information to transform detected barriers into spatial parameters. Based on these routing constraints, existing navigation algorithms are significantly improved for rough terrain applications lacking roads and paths. The implemented RemoteNav navigation module will be demonstrated in the existing applications Sarontar and Deermapper.

Keywords: Navigation, remote sensing, GPS, laser scanning

1 Einführung

Die Verwendung von Navigationssystemen für Fahrzeuge, Fußgänger und den öffentlichen Verkehr ist in unserer Welt nicht mehr wegzudenken. Die bestehenden Navigationssysteme basieren auf Graphendaten und setzen damit eine Wege- und Straßeninfrastruktur voraus. Viele Anwendergruppen wie alpine Rettungsteams, Jäger oder Wanderer sind aber auch abseits dieser bekannten Straßeninfrastruktur im unwegsamen Gelände unterwegs, in dem graphenbasiertes Routing nicht mehr zur Anwendung kommen kann. RemoteNav hat sich zum Ziel gesetzt, diese Lücke durch die Entwicklung eines Navigationssystems für abgelegene Gebiete auf Basis von Fernerkundungsdaten zu schließen.

2 Datengrundlagen aus Fernerkundungsdaten

Fernerkundungsdaten sind eine der wenigen Datenquellen, die aktuelle Informationen zur Beschreibung von unwegsamem Gelände in abgelegenen Gebieten bieten, aus denen zuver-

lässige Indikatoren für die Ableitung von Kostenparameter zum Aufbau eines Navigationsalgorithmus abgeleitet werden können. Methoden der Fernerkundung bieten hier Ansätze für die objektive und standardisierte Erfassung von für die Fußgängernavigation sowohl positiven als auch negativen Indikatoren. Diese Indikatoren der Gelände- und Vegetationsstrukturen umfassen die Landbedeckung, die Topographie und die Phänologie. Positive Indikatoren der Landbedeckung sind bspw. offene Flächen wie Wiesen oder Lichtungen, negative Indikatoren können Jungwuchs, Latschen oder Sumpfgebiete sein. Topographische Parameter berücksichtigen Faktoren, die aus der Geländebeschaffenheit abgeleitet werden. Darunter werden Indikatoren verstanden, die einerseits die Geländerauigkeit beschreiben und andererseits die Reliefenergie bzw. die zurückzulegenden positiven und negativen Höhenunterschiede berücksichtigen. Negativ für die Navigation werden dabei steiles Gelände, große Höhenunterschiede oder Abbruchkanten gesehen. Als positive Faktoren werden flaches Gelände oder ein geringer bzw. negativer aufsummierter Höhenunterschied zwischen Start- und Zielpunkt der Navigation gesehen. Phänologische und saisonale Parameter beschreiben jahreszeitliche Einflüsse auf den Navigationsalgorithmus, wie bspw. der Vegetationsgrad, die Schneebedeckung oder der Wasserstand von Gewässern. Neben diesen aus Erdbeobachtungsdaten ableitbaren Informationen müssen aber auch anthropologische Einflüsse berücksichtigt werden, die rein aus Fernerkundungsdaten nicht abgeleitet werden können. Dazu zählen z. B. Zäune, Sperr- und Schutzgebiete oder administrative Grenzen, die nicht überwunden werden können bzw. nicht überwunden werden dürfen. Auch die Einbindung von bestehenden Straßengraphen ist Teil der Datenharmonisierung, da eine Route über eine Forststraße ungleich geringere Kosten aufweist wie z. B. jene durch dichten Unterwuchs oder über Wasserflächen hinweg. Um also in der Lage zu sein, die idealen Routen im alpinen und unwegsamen Gelände zwischen zwei vorgegebenen Punkten zu berechnen, müssen diese räumlichen Informationen von Gelände- und Vegetationsbarrieren bekannt sein und in einer harmonisierten GIS-Datenbank zusammengeführt werden. Die wesentlichen Informationen dafür werden aus Satelliten- bzw. Luftbildern, LIDAR-Daten und bestehenden topographischen Karteninformationen wie z. B. Open Street Map (OSM) oder Open Government Data (OGD) abgeleitet. Die Harmonisierung und einheitliche Zusammenführung dieser Indikatoren und Parameter sorgt dafür, dass das Navigationssystem auf einer gemeinsamen und einheitlichen Datenbasis aufgesetzt werden kann.

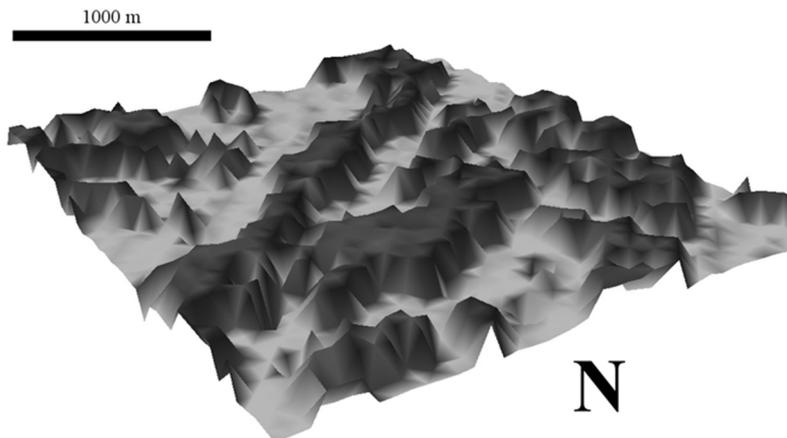


Abb. 1: 3D-Visualisierung des Kostenrasters

Die Zusammenführung aller erwähnten Navigationsparameter für das Routing wesentlicher Faktoren wird in einem Kostenraster zusammengefasst. In Abbildung 1 ist ein solcher Kostenraster in einer dreidimensionalen Ansicht dargestellt, wobei die Erhöhungen im Kostengebirge schwer überwindbare Geländebereiche darstellen und die Täler die leichter durchschreitbaren Abschnitte. Das Ziel des Routingalgorithmus ist nun, einen Weg durch dieses Kostengebirge zu finden, der ein Minimum an zu überwindbaren Kosten verursacht. Die Zusammenführung der einfließenden Navigationsparameter erfolgt additiv mit Formel 1 für jeden einzelnen Rasterpunkt im Untersuchungsgebiet.

$$\overline{L_{Kosten}} = \frac{1}{n} \sum_{i=0}^n \omega_i * \overline{L}_i \quad (1)$$

$\overline{L_{Kosten}}$ stellt in Formel 1 den normalisierten Gesamtkostenlayer dar, ω_i entspricht dem Gewicht des Kostenlayers \overline{L}_i mit dem Wertebereich $[0, 1]$ und \overline{L}_i spiegelt die normalisierten Kosten eines einzelnen Navigationsparameters wider. Dazu ist es a-priori notwendig, jeden einzelnen Navigationsparameter zu normalisieren, sprich auf einen Wert zwischen 0 und 1 zu bringen. Die Normalisierung wird durch die Division des Kostenwertes durch den theoretisch maximal auftretbaren Wert gewährleistet. Bei der Hangneigung wären das z. B. 90° . Diese Vorgehensweise ermöglicht die Übertragung des entwickelten und kalibrierten Algorithmus auf andere Testgebiete. Eine weitere Möglichkeit wäre zudem, die Gewichte nicht gleichverteilt auf den Kostenwert zu applizieren, sondern über eine nichtlineare Funktion. Untersuchungen in Feldtests werden in dazu weitere Aufschlüsse liefern. Die Gewichte jedes einzelnen Eingangsparameters werden in erster Näherung mit 1 angenommen, das einer gleichgewichteten Verwendung aller Eingangsparameter entspricht. Durch die Kalibrierung dieses Modells durch Feldtests werden die Gewichte der einzelnen Navigationsparameter für die unterschiedlichen Nutzerprofile (Wanderer mit Leichtgepäck vs. Bergretter mit einer Vielzahl an technischem Equipment) bestimmt. Bei der Berechnung des dargestellten Kostenrasters ist zu bemerken, dass die im Testgebiet vorhandenen Forststraßen einen großen Einfluss auf das Ergebnis haben. Durch die vielerlei für die Wegfindung positiven topographischen und vegetationsspezifischen Eigenschaften der Straßeninfrastruktur, wie z. B. geringe Geländerauigkeit, geringe Hangneigung, frei von jeglicher Vegetation, keine Wasserläufe etc., wird das vorhandene Wegenetz mehrfach positiv bewertet. Dies könnte möglicherweise auf eine positive Überschätzung der Straßen und Wege führen. Weitere Untersuchungen aus abgeleiteten Feldtests und Kalibrierungen des entwickelten Navigationsalgorithmus werden weiteren Aufschluss darüber liefern.

3 Routingansätze

Als Grundlage für die untersuchten Routingansätze wird ein Kostenraster verwendet, das aus einer Vielzahl von Navigationsparametern abgeleitet wird und das in unterschiedlichen Rasterweiten von 1×1 m bis 100×100 m vorgerechnet wird. Alle für die Navigation verwendeten Eingangsparameter werden räumlich durch additive Kombination und Gewichtung in diesem Raster abgebildet. Die Kostenwerte der einzelnen Rasterzellen werden CSV-basiert dem Routingalgorithmus zur Verfügung gestellt. Diese CSV-Datei enthält zeilenweise die normalisierten Kostenwerte für alle Rasterzellen im Operationsgebiet. Für die effiziente Ver-

speicherung und Analyse dieses Kostenrasters wurde ein Übergang von relationalen Datenbankansätzen auf fileobjektbasierte Ansätze durchgeführt. Die Vorteile dieses Ansatzes sind Performancegründe hinsichtlich Datenmenge, Speichervolumen auf mobilen Geräten und signifikante Performancegewinne in der Verarbeitung und Analyse der Daten. Die oberste Prämisse dabei ist die Praxistauglichkeit des Ergebnisses für die Integration in die bestehenden Applikationen Sarontar und Deermapper. Die Nutzer sollen in so kurzer Zeit wie möglich die so effizient wie möglich gespeicherten Navigationsdaten auf ihr Endgerät herunterladen können. Als Datenbankmanagementsystem (DBMS) für mobile Endgeräte wird der objektorientierte Datenbankansatz Realm verwendet (<https://realm.io/>). Um die Matrix-Darstellung (Rasterform) des übermittelten Kostenrasters abzubilden, wird eine Liste von Listen erstellt. Eine Liste, die eine Reihe in der Matrix repräsentiert, enthält die jeweiligen Kostenwerte für die Navigation. Eine weitere Liste speichert alle Reihen in aufsteigender Reihenfolge.

Die bereitgestellten Navigationsparameter aus der CSV-Datei stellen direkt die jeweiligen Werte pro „Bildpixel“ dar. Diese Bewertungen sind normiert und somit im Bereich von 0 bis 1. Je höher der Wert, desto schwieriger ist der jeweilige Bereich durchquerbar. Ein Wert von 1 bedeutet, dass der Bereich nicht durchquerbar ist und wird somit bei der Navigation nicht in Betracht gezogen. Das Routingnetzwerk bildet sich aus den einzelnen Rasterzellen mit den jeweiligen Navigationswerten. Die anfängliche Überlegung, Rasterzellen mit ähnlichen Werten zusammenzufassen, wurde wieder verworfen, da dies die Navigationsgenauigkeit unnötig verringert. Abbildung 2 zeigt die Erstellung eines virtuellen Routingnetzwerkes aufbauend auf einem vorberechneten Kostenraster im Untersuchungsgebiet. Das Netzwerk ist virtuell, da es nicht wirklich erstellt wird, sondern durch die Matrixstruktur repräsentiert wird. Daraus geht hervor, dass ein Knotenpunkt, der in der Mitte eines Bildpixels liegt, jeweils acht Nachbarzellen besitzt – sofern dieser nicht am Bildrand liegt oder ein Pixel mit dem Wert 1 als Nachbar hat. Wie bereits erwähnt, werden Werte von 1 nicht in die Navigation miteinbezogen und somit auch bei der Erstellung des virtuellen Netzwerkes vernachlässigt.

0.74	0.62	0.68	0.53	0.27
0.37	1.00	0.48	0.43	0.21
0.70	0.66	0.41	0.15	0.68
0.39	0.55	0.16	0.12	1.00
0.35	0.73	0.14	0.28	0.24

Abb. 2:
Matrixstruktur der Kostenverteilung

Für die Routenkalkulation muss der Netzwerkgraph aus einer Reihe von Knoten und Kanten bestehen. Die Pixelmittelpunkte bilden dabei die Knoten und die Verbindungslinien zu allen Pixelnachbarn (ausgenommen zu jenen mit Werten von 1) die Kanten. Der nächste Schritt ist, den Graphen zu bewerten. Zu diesem Zweck wird jeder Kante ein Wert zugewiesen, der

die Kosten entlang dieser Kante widerspiegelt. Diese Kosten beschreiben, wie schwierig es ist, die Kante zu passieren. Ein Weg zwischen dem ausgewählten Start- und Endknoten ist eine Folge von Kanten. Die Summe der Kostenwerte aller Kanten innerhalb des Weges bildet die Gesamtkosten des Weges. Jener Weg mit den minimalen Gesamtkosten aller möglichen Wege zwischen Start- und Endknoten wird kürzester Weg genannt, der dem Nutzer als bester Weg zwischen Start- und Endpunkt vorgeschlagen wird. Für die Berechnung dieses optimalen Weges zwischen zwei Punkten in einer Knoten- und Kantenstruktur kommen zwei Routingansätze in die engere Auswahl. Dies sind der Dijkstra- und der A-Stern-Algorithmus.

Dijkstra-Algorithmus

Das Prinzip des Dijkstra-Algorithmus ist es, den kürzesten Weg von einem Knoten zu allen anderen zu berechnen. Ist ein bestimmter Zielknoten definiert – wie es in RemoteNav der Fall ist – wird der Algorithmus unterbrochen, sobald der kürzeste Weg zum Zielknoten gefunden ist. Der Algorithmus breitet sich somit in alle Richtungen um den Startknoten gleichmäßig aus und hat somit eine weniger gute Performance in extrem vermaschten und großen Routinggraphen.

A-Stern-Algorithmus

Der A-Stern-Algorithmus ist eine Erweiterung von Dijkstra und vor allem für große Datensätze von Vorteil. Der A-Stern-Algorithmus ist der klassische Best-First-Search-Algorithmus. Er verwendet bei der Ermittlung des kürzesten Weges zwischen einem Start- und Zielknoten eine Heuristik (ungenaueres Wissen) und dadurch wird eine vielversprechende Richtung bei der Suche zuerst eingeschlagen. Der Schätzwert bis zum Ziel muss immer optimistisch sein, d. h. die tatsächliche Distanz muss größer sein. Die Verwendung der Luftliniendistanz als Schätzwert bietet sich dabei in den meisten Fällen an. Der A-Stern-Algorithmus breitet sich somit nicht wie der Dijkstra-Algorithmus in alle Richtungen gleich aus, sondern folgt einer zielorientierten Suche und verringert somit die vorgesehene Zeit, den kürzesten Weg zu finden.

Da der A-Stern-Algorithmus in großen Netzwerkgraphen durch die Verwendung einer Heuristik eine gute Performance bietet, fiel die Wahl in der Umsetzung von RemoteNav auf diesen Algorithmus. Beim A-Stern-Algorithmus werden die besuchten Knoten so bewertet, dass der bisher zurückgelegte Weg vom Startknoten mit der Schätzung des Weges bis zum Ziel addiert wird. Die Knoten werden nach dieser Bewertung gereiht. Der Knoten mit der minimalen Bewertung wird immer als nächster verfolgt.

4 Kalibrierung und Validierung

Um den Routingalgorithmus zu kalibrieren, werden in Feldtests vorbestimmte Routen zwischen zwei Punkten von Probanden abgegangen. Es wird daraus abgeleitet, wie der Einfluss von unterschiedlichen Vegetationsklassen und topographischen Bedingungen auf das Vorankommen der zu Fuß agierenden Personen ist. Basierend auf den über GPS gesammelten Daten wird die Geschwindigkeit der Probanden durch unterschiedliche Geländeprofile ermittelt. Die Geschwindigkeit des Vorankommens in den unterschiedlichen Geländeabschnitten wird

umgekehrt proportional in Kostenwerte umgelegt und damit der Zusammenhang der einzelnen Navigationsparameter gewichtet. Zusätzlich gibt es unterschiedliche Profile von Nutzern, die den Grad der Mobilität beschreibt. So ist es ein großer Unterschied, ob die optimale Route für einen sportlichen Wanderer mit Leichtgepäck oder für einen Bergretter mit einer Vielzahl an technischem Equipment berechnet wird. Das gleiche Gelände ist für den einen Nutzer gangbar, für den anderen stellt dieses aber ein unüberwindbares Hindernis dar.

Als Testgebiet wurde ein ca. 4 km² großes Gebiet im alpinen Gelände ausgewählt. Diese Größe stellt ein realistisches Jagdrevier für die Anwendung in der Jagdsoftware Deermapper dar. Davon ausgehend kann aber auch abgeschätzt werden, wie sich die Performance des Routingalgorithmus in größeren Gebieten verhält. Die für das Testgebiet berechnete CSV-Datei mit den Kostenwerten mit einer Rastergröße von 1 × 1 m hat eine Größe von 27 MB, das für den einmaligen Download auf ein mobiles Endgerät im Rahmen ist. Wird die Datei auch noch komprimiert, kann die Größe auf ungefähr 7 MB verringert werden. Eine weitere Möglichkeit der Speicherplatzoptimierung ist, direkt eine vorab erstellte Realm-Datei zum Download zur Verfügung zu stellen. Da Realm aber bislang nur für mobile Betriebssysteme zur Verfügung steht, konnte dieser Schritt mit der im Konsortium verfügbaren Serverinfrastruktur noch nicht implementiert werden. Die Dateigröße bleibt zwar in etwa gleich, jedoch spart dies die Zeitdauer des Befüllens der Datenbank ein. Das Einlesen der CSV-Navigationsparameter inklusive deren Verspeicherung in Realm-Objekten beansprucht für das 4 km² große Gebiet auf einem handelsüblichen Smartphone ungefähr 6 Minuten. In Anbetracht dessen, dass das Laden der Kostenparameter einmalig pro Region erfolgt, ist die Zeitdauer der Erstellung der Datenbank akzeptabel. Beim Einlesen der entsprechenden Realm-Datei verringerte sich die Zeitdauer für das Einlesen der Daten auf unter 1 Minute.

Erste Tests des Routingalgorithmus auf Desktop-Basis belegen eine adäquate Arbeitsweise, da der berechnete Weg zuerst entlang des leichter durchquerbaren Bereiches führt (z. B. entlang vorhandener Forststraßen oder Wanderwege) und erst danach in den schwerer durchquerbaren Bereich (z. B. dichter Unterwuchs, steiles Gelände) mündet, um an das Ziel zu gelangen. Der vom Algorithmus durchsuchte Bereich rund um den berechneten Weg ist relativ schmal, d. h. der A-Stern-Algorithmus leistet mit Miteinbeziehung der Luftliniendistanz als Heuristik gute Dienste. Die nächsten Schritte bezüglich Routenkalkulation sind weitere Tests mit unterschiedlichen Navigationsparametern, sowie Performancetests auf Desktopcomputern und in weiterer Folge auch auf mobilen Endgeräten.