

Universität Bonn, Mathematisch-Naturwissenschaftliche Fakultät, Master of Science,
Mathematics

4. Januar 2014

Praktikumsbericht von

Pascal Huber (2219408), 8. Fachsemester, pascal.huber@aol.com

über ein Praktikum beim Fraunhofer-Institut für Eingebettete
Systeme und Kommunikationstechnik ESK

vom 15.04.2013 bis zum 27.09.2013

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 2 |
| 2 | Beschreibung der Praktikumsstelle | 2 |
| 2.1 | Die Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V. | 2 |
| 2.2 | Das Fraunhofer-Institut für Eingebettete System und Kommunikations- technik ESK und das Geschäftsfeld Automotive | 2 |
| 2.3 | Das Bewerbungsverfahren | 3 |
| 2.4 | Rahmenbedingungen des Praktikums | 3 |
| 3 | Aufgaben und Tätigkeiten - Ablauf des Praktikums | 4 |
| 3.1 | Überblick und Einordnung der Praktikumstätigkeit | 4 |
| 3.2 | Zielsetzung und nähere Beschreibung der Praktikumstätigkeit | 4 |
| 3.3 | Tätigkeit und Arbeitsergebnisse | 5 |
| 3.3.1 | Einarbeitungsphase | 5 |
| 3.3.2 | Routing-Algorithmen | 5 |
| 3.3.3 | Mapmatching-Algorithmen | 7 |
| 4 | Quellen | 10 |

1 Einleitung

2 Beschreibung der Praktikumsstelle

Das Fraunhofer-Institut für Eingebettete Systeme und Kommunikationstechnik ESK (kurz ESK) ist Teil der Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.

Kurze Einleitung und Überblick, was hier noch so kommt.

2.1 Die Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V.

Die Fraunhofer Gesellschaft sieht ihre zentrale Aufgabe in der Förderung von anwendungsorientierter Forschung zum unmittelbaren Nutzen für die Wirtschaft und zum Vorteil für die Gesellschaft. Zu diesem Zweck betreibt sie Vorlaufs- und Auftragsforschung in den Forschungsfeldern Gesundheit, Sicherheit, Kommunikation, Mobilität, Energie und Umwelt.

Die Fraunhofer Gesellschaft ist mit ungefähr 22.000 Mitarbeitern an 40 Standorten die größte Organisation für anwendungsorientierte Forschung in Europa und stellt somit einen Eckpfeiler der deutschen Forschungslandschaft dar. Das gesamte Finanzvolumen der Fraunhofer Gesellschaft betrug im Jahr 2012 ungefähr 1.9 Mrd. Euro, wobei der Großteil (etwa 70%) der Erträge durch Aufträge aus der Wirtschaft und von öffentlichen Forschungsprojekten erwirtschaftet wurde. Der restliche Anteil des Forschungsvolumens erhält die Fraunhofer Gesellschaft als Grundfinanzierung von Bund und Ländern.

Fußnote mit Quelle angeben

Intern ist die Fraunhofer Gesellschaft in 66 eigenständige Institute und einige weitere Einrichtungen unterteilt, die von einer Zentrale mit Sitz in München koordiniert werden. Die einzelnen Einrichtungen arbeiten weitestgehend unabhängig und agieren selbstständig auf dem Markt. Dies betrifft auch die Finanzierung durch Auftragsforschung.

Verbünde erwähnen?

2.2 Das Fraunhofer-Institut für Eingebettete System und Kommunikationstechnik ESK und das Geschäftsfeld Automotive

Das ESK ging 1999 zunächst nur als kleinere Einrichtung aus der Abteilung Systemtechnik/Telekommunikation des Fraunhofer Instituts für Festkörpertechnologie und wurde im Juli 2013 endgültig in ein dauerhaftes Fraunhofer Institut überführt. Das Institut hat seinen Sitz in unmittelbarer Nähe zur Fraunhofer Zentrale in München und ist mit ungefähr 80 ständigen Mitarbeitern eines der kleinsten Institute der Fraunhofer Gesellschaft.

Überschrift zu lang?

Der Forschungsgegenstand des ESK sind Verfahren und Methoden der Informations- und Kommunikationstechnik (kurz IKT) mit Schwerpunkten auf der Adaptivität (selbstständig anpassende Systeme) von verteilten und vernetzten Systemen und Software-Methodik. Dies umfasst beispielsweise die Konzeption und Analyse neuer Ansätze in der Softwa-

Quelle angeben und hinzufügen, dass die meisten Mitarbeiter Informatiker und Elektrotechniker sind

Wie weit soll ich hier ins Detail gehen?

reentwicklung für vernetzte Systeme (Parallelisierung und Multicore-Software), die Entwicklung und Verbesserung von zuverlässigen und effizienten Funknetzen in industriellen Anlagen oder im Fahrzeugbereich (M2M, Car-to-X) oder die Optimierung von Software für adaptive verteilte Systeme, d.h. Systeme, die sich selbständig an Änderungen in der Umgebung oder im Ressourcenbedarf anpassen (z. Bsp. Steuergeräte im Automobilbereich).

Hier muss noch eine Erklärung hin oder soll ich es gleich ganz weglassen?

Das ESK gliedert sich in die drei Geschäftsfelder Automotive (Fahrzeugkommunikation), Industrial Communication (Software- und Kommunikationslösungen für industrielle Anwendungen) und Telecommunication (u.a. Verbesserung von Leitungsnetzen). Meine Praktikumsstelle wurde im Rahmen des Geschäftsfeldes Automotive ausgeschrieben, weshalb ich diesen und insbesondere den Leistungsbereich *Automotive Connectivity* näher beschreiben möchte.

Hervorheben? Schreiben, dass Automotive größtes Geschäftsfeld ist

Um die Sicherheit und die Effizienz im Straßenverkehr zu steigern, ist eine immer stärkere Vernetzung von Fahrzeugen sowohl untereinander als auch mit der Umgebung notwendig. Das Geschäftsfeld Automotive arbeitet deshalb an Hardware- und Softwarelösungen, die die Zuverlässigkeit und die Dienstqualität dieser Vernetzung gewährleisten soll und unterstützt somit Fahrzeughersteller und Dienstleister bei der Steuergeräte-Entwicklung und der Fahrzeug-Umwelt-Vernetzung (sog. Car-to-X Systeme).

In diesem Rahmen entwickelt der Leistungsbereich Automotive Connectivity Software-Frameworks, die die aus der Vernetzung entstehenden Daten verarbeiten und somit zum Beispiel die Integration von kooperativen Fahrerassistenzsystemen (z.Bsp. kooperativer Tempomat) ermöglichen.

Zu viele Details? Ist das verständliche?

2.3 Das Bewerbungsverfahren

Nachdem ich mich zum Ende des Wintersemesters 2012/13 entschlossen hatte, im darauf folgenden Sommersemester ein mehrmonatiges Industriepraktikum zu absolvieren, machte ich mich zu Beginn der Semesterferien auf die Suche nach einem geeigneten Praktikumsplatz im Großraum München. Als Hauptinformationsquelle diente mir das Internet. Dabei fand ich unter anderem auch die Stellenausschreibung für mein späteres Praktikum auf dem Karriereportal der Fraunhofer-Gesellschaft, für das ich mich online bewerben konnte. Nur wenige Tage nach der Bewerbung wurde ich telefonisch zu einem Bewerbungsgespräch eingeladen, in dem mich mein späterer Betreuer zu meiner Motivation und meinem Vorwissen befragte und anschließend den genauen Inhalt des Praktikums darlegte. Das Gespräch hatte weniger den Charakter eines Bewerbungsgesprächs, sondern war vielmehr eine nette Unterhaltung, an dessen Ende mir der Praktikumsplatz bereits zugesichert wurde. Wenige Tage nach diesem Gespräch wurde der genaue zeitliche Rahmen des Praktikums festgelegt und ich erhielt meinen Arbeitsvertrag.

Soll ich das lieber weglassen?

Fahrtkosten erwähnen?

Soll ich noch hinzufügen welche Erfahrungen ich aus der Bewerbung mitnehme?

2.4 Rahmenbedingungen des Praktikums

Meine Praktikumsstelle war als Vollzeitpraktikum ausgelegt, sodass ich 39 Stunden in der Woche im ESK arbeiten sollte. Es gab zwar keine festen Arbeitszeiten, allerdings

mussten alle Mitarbeiter in bestimmten Kernzeiten anwesend sein.

Während meines Praktikums hatte ich einen festen Betreuer, der am selben Projekt arbeitete und den ich auch schon beim Bewerbungsgespräch kennen gelernt hatte. Einen festen Besprechungstermin gab es zwar nicht, allerdings konnte ich meinen Betreuer bei Fragen problemlos persönlich oder auch per E-Mail um Rat fragen.

Da mein Praktikum über einen Zeitraum von mehr als fünf Monaten ging, erhielt ich einen eigenen Arbeitsplatz in einem der Praktikantenräume des ESK. Neben mir arbeiteten ungefähr 20 weitere Studenten als Hilfskräfte oder schrieben ihre Abschlussarbeiten im Institut, sodass die Praktikantenräume immer sehr gut gefüllt waren.

Mehr über die Betreuung schreiben?

Doofe Formulierung... der ganze Abschnitt gefällt mir nicht

Auch über Urlaubstage und Aufwandsentschädigung schreiben? (Habe ich im Internet gelesen...)

Überschrift ändern?

Ermöglichung klingt komisch

3 Aufgaben und Tätigkeiten - Ablauf des Praktikums

3.1 Überblick und Einordnung der Praktikumstätigkeit

Wie bereits beschrieben entwickelt der Leistungsbereich Automotive Connectivity Systeme zur Vernetzung von Fahrzeugen untereinander und mit der Umwelt, also sogenannte Car-to-X Systeme. Ziel solcher Systeme ist eine gesteigerte Verkehrssicherheit und -effizienz durch die Ermöglichung neuer Dienste wie personalisiertes Routing, eine genauere Erfassung der Verkehrslage und die Integration von kooperativen Fahrerassistenzsystemen.

Neben einer zuverlässigen und leistungsstarken Datenübertragung benötigen Car-to-X System auch eine Vielzahl von Algorithmen, die die gewonnenen Daten in Relation zueinander setzen und für spätere Anwendungen weiterverarbeiten. Ein wichtiger Baustein in diesem Kontext ist die Integration von Kartendaten zur Integration von Navigationsinformationen.

Im Rahmen meines Praktikums sollte ich an dieser Stelle ansetzen und für das vom ESK entwickelte *ezCar2X* Software-Framework kartenbezogene Algorithmen auswählen, implementieren und testen. Diese Algorithmen sollen die Verwendung von Kartenmaterial zu Navigations- und Routingzwecken ermöglichen. Das *ezCar2X* Projekt besteht dabei aus einer Sammlung von C++-Bibliotheken, die eine zügige Erstellung von prototypischen Anwendungen für Car-to-X Systeme ermöglichen soll. Mögliche Abnehmer sind dabei beispielsweise Fahrzeughersteller oder Zulieferer von Verkehrsinfrastruktur.

Zwei Projekte ansprechen, in der das Software-Framework voraussichtlich Verwendung findet? (Mobincity, ACM)

3.2 Zielsetzung und nähere Beschreibung der Praktikumstätigkeit

Ziel meines Praktikums war die Erweiterung des *ezCar2X* Software-Frameworks um kartenbezogene Algorithmen zur Ermöglichung von positions- und navigationsbezogenen Anwendungen. Dabei sollte ich einige einfache Vertreter dieser Algorithmen implementieren, die als Basis für komplexere Verfahren dienen sollen. Unter kartenbezogenen Algorithmen versteht man dabei im Wesentlichen zwei Arten von Algorithmen . Auf der

Komische Formulierung? Grammatik?

Wiederholung Algorithmen

einen Seite werden Mapmatching-Verfahren benötigt, die die Fahrzeugposition auf einer gegebenen digitalen Karte abbilden. Dazu werden Positionsinformation (GPS-Position) mit vorhandenem Kartenmaterial in Beziehung gebracht. Auf der anderen Seite sind Routing-Algorithmen notwendig, um die nach einem vorgegebenen Kriterium beste Route zum Ziel zu errechnen.

Im Zuge meines Praktikums sollte ich mich zunächst einen Überblick über mögliche Mapmatching- und Routing-Algorithmen machen und anschließend eine Auswahl solcher Algorithmen implementieren und testen. Parallel wurde von meinem Betreuer ein Verwaltungssystem für digitale Kartendaten implementiert, auf das sich die von mir geschriebenen Algorithmen stützen sollten und bei dem ich ihn gegebenenfalls unterstützen sollte. Das Kartenmaterial für dieses Verwaltungssystem stammt vom OpenStreetMap-Projekt, welches den Vorteil hat, dass die Daten sehr vielfältig und präzise und gleichzeitig lizenzkostenfrei sind.

Grammatik falsch?

Da meine Betreuer und ich die einzigen Mitarbeiter waren, die sich mit der Integration von kartenbezogenen Anwendungen befassen, war es fast vollständig mir überlassen, welche Schwerpunkte ich setzen und welche Funktionalitäten ich implementieren sollte.

Ist dieser letzte Satz vielleicht unnötig?

3.3 Tätigkeit und Arbeitsergebnisse

3.3.1 Einarbeitungsphase

Bevor ich mit der eigentlichen Arbeit, also der Auswahl und Implementierung der Algorithmen anfangen konnte, musste ich zunächst meinen Arbeitsplatz einrichten und mir einen Überblick über die für mich relevanten Teile des ezCar2X-Frameworks machen. Nachdem ich mich mit dem bestehenden Source-Code vertraut gemacht hatte, begann ich eine Übersicht von möglichen Routing- und Mapmatching-Algorithmen zu erstellen, die mir als Grundlage für meine Arbeit dienen sollte. Dabei bezog ich meine Informationen hauptsächlich aus wissenschaftlichen Papern, die ich im Internet fand und aus Gesprächen mit meinem Betreuer, der bereits in seiner Studienzeit viel mit Mapmatching-Verfahren gearbeitet hatte.

Ist dieser Abschnitt notwendig?

"Wissenschaftliche Paper" klingt komisch

Grammatik falsch? Satz unnötig?

3.3.2 Routing-Algorithmen

Um mir den Einstieg in die Programmierarbeit zu erleichtern, entschied ich mich mit der Implementierung der Routing-Algorithmen zu beginnen, da ich das grundlegende Verfahren, nämlich den Dijkstra-Algorithmus schon aus meinem Studium kannte. Im Folgenden möchte ich zunächst die Grundidee des Routing-Verfahrens vorstellen und anschließend kurz auf die von mir implementierten Algorithmen eingehen.

"Nämlich" hört sich doof an

Das Verfahren Die Aufgabe eines Routing-Algorithmus ist die Bestimmung einer 'besten' Route zwischen zwei gegebenen Positionen. Die Güte einer Route wird hierbei durch ein zuvor festgelegtes Kriterium definiert, meistens handelt es sich dabei entweder um die Reisezeit oder die Gesamtlänge der Route.

Mathematisch gesehen wird das Straßennetz als Graph aufgefasst, in dem die Knoten die Kreuzungen und die Kanten die dazwischen liegenden Straßenabschnitte repräsentieren.

Jeder Kante wird anschließend ein (ein nicht-negatives) Gewicht zugeordnet, das sich aus dem zuvor gewählten Kriterium ableitet, beispielsweise die Länge des Straßenabschnitts in Metern. Zur Bestimmung des besten Weges zwischen zwei Positionen wird nun ein von der Startposition ausgehender Dijkstra-Algorithmus verwendet.

Da im Prinzip alle Routing-Algorithmen auf diesem einfachen Verfahren aufbauen, möchte ich kurz die wesentlichen Schritte des Dijkstra-Algorithmus in Erinnerung rufen. Die Grundidee des Algorithmus ist es, im jeden Schritt immer der Kante zu folgen, die zu diesem Zeitpunkt das geringste Gesamtgewicht zum Startknoten besitzt. Im einzelnen werden folgende Schritte ausgeführt:

1. Weise dem Startknoten ein Gesamtgewicht von 0 und allen anderen Knoten ein Gesamtgewicht von ∞ zu.
2. Solange der Zielknoten nicht besucht wurde, wähle unter allen unbesuchten Knoten denjenigen mit dem geringsten Gesamtgewicht aus.
 - a. Markiere den ausgewählten Knoten als 'besucht'.
 - b. Berechne für alle unbesuchten Nachbarknoten die Summe aus dem aktuellen Gesamtgewicht und den Kosten der verbindenden Kante.
 - c. Falls das Gesamtgewicht eines unbesuchten Nachbarknoten größer als die berechnete Summe ist, ersetze es durch die Summe und aktualisiere den Vorgänger.

Vielleicht richtiger Pseudo-Code oder sogar einfach weglassen? Relaxation benennen?

Dijkstra-Algorithmus Bei der Implementierung des Dijkstra-Algorithmus musste ich den obigen Algorithmus an ein paar Besonderheiten beim Routing in Straßennetzen anpassen. Das größte Problem stellten dabei Abbiegeverbote dar. Da in diesem Fall Paare von Kanten betrachtet werden müssen, änderte ich den Algorithmus dahingehend, dass dieser nicht nur Knoten sondern Knoten-Kanten-Paare betrachtet. Der resultierende Algorithmus lieferte schließlich korrekte Ergebnisse.

Das Hauptproblem der Dijkstra-Suche besteht darin, dass der Algorithmus einen großen Suchraum (Menge der besuchten Knoten) betrachten muss, was zu hohen Laufzeiten führt. Die eigentliche Schwierigkeit beim Routing besteht also darin, den Dijkstra-Algorithmus mit Hilfe von Zusatzinformationen so zu verbessern, dass einerseits die Korrektheit des Algorithmus gewahrt bleibt und andererseits die Laufzeit verringert wird.

Dieser Satz gefällt mir nicht!

Bidirektionale Suche Als ersten Ansatz zur Reduktion der Laufzeit wählte ich das Verfahren der bidirektionalen Suche. Dabei wird zusätzlich zu der obigen Dijkstra-Suche simultan eine weitere von der Zielposition ausgehende Suche durchgeführt. Der Algorithmus terminiert sobald ein Knoten sowohl von der Vorwärts- als auch von der Rückwärtssuche besucht wurde. Dieses Verfahren halbiert den Suchraum und verringert entsprechend die Laufzeit. Bei der Implementierung stellte sich vor allem die korrekte Formulierung des Abbruchkriteriums als schwierig heraus, da ich aufgrund der Abbiegeverbote wie im einfachen Dijkstra-Algorithmus Knoten-Kanten-Paare betrachten musste.

A*-Algorithmus Ein effektiverer Weg die Laufzeit des Dijkstra-Algorithmus zu verringern ist der Einsatz von Schätzfunktionen (Heuristiken), die zusätzliche Informationen (zum Beispiel geographische Positionen) verwenden, um den Suchraum zu verkleinern. Der resultierende Algorithmus wird A*-Algorithmus genannt.

Bemerkung
in Klammer
unnötig?

Wie in Abschnitt 3.3.2 gesehen, wählt der Dijkstra-Algorithmus intern in jeden Schritt denjenigen unbesuchten Knoten v mit den geringsten Gesamtkosten $c(v)$ aus. Der A*-Algorithmus dagegen berücksichtigt zusätzlich zu diesen Gesamtkosten eine Schätzfunktion $h(v)$, sodass derjenige Knoten besucht wird, für den die Summe

$$c(v) + h(v) \quad (1)$$

am kleinsten ist. Die Schätzfunktion $h(v)$ soll die verbleibenden Kosten von v bis zur Zielposition abschätzen und damit diejenigen Knoten v bevorzugen, die im Sinne der gewählten Heuristik am wahrscheinlichsten auf einer besten Route liegen.

Damit der A*-Algorithmus ein korrektes Ergebnis liefert, muss die Heuristik $h(v)$ zwei Bedingungen erfüllen: Auf der einen Seite darf sie die benötigten Gesamtkosten um von v zum Zielknoten b zu gelangen nicht überschätzen, d.h. es muss gelten $h(v) \leq \text{dist}_c(v, b)$ und auf der anderen Seite muss eine gewisse Dreiecksungleichung erfüllt sein.

Diesen Abschnitt etwas
klarer aufschreiben

Für mein Praktikum wählte ich als Heuristik zum einen die euklidische Distanz zwischen v und b und zum anderen eine Schätzfunktion die auf vorberechneten Kosten von zuvor zufällig gewählten Knoten, sogenannten Landmarken beruht (ALT-Algorithmus). Bevor ich diese Heuristiken implementieren konnte, musste ich aufgrund der durch die Abbiegeverbote nötigen Änderungen zunächst zeigen, dass die gewählten Schätzfunktion die beiden oben genannten Bedingungen erfüllen. In den anschließenden Tests zeigte sich, dass vor allem der ALT-Algorithmus die Laufzeit bei geeigneter Wahl der Landmarken deutlich verkürzt. Eine weitere Verbesserung konnte ich schließlich durch Kombination des A*-Algorithmus mit der bidirektionalen Suche erreichen.

Muss ich hier
mehr ins Detail
gehen?

Genauer auf
meine Ergebnisse
eingehen.

Hier noch näher
darauf eingehen?

3.3.3 Mapmatching-Algorithmen

Damit ein Routing-Algorithmus die beste Route zu einem gegebenen Zielpunkt berechnen kann, muss zunächst bestimmt werden, an welcher Position der digitalen Karte sich das zu routende Fahrzeug befindet. Da sowohl das Kartenmaterial als auch die Positionsinformationen (in den meisten Fällen GPS-Daten) mit Fehlern behaftet sind, sind spezielle Verfahren notwendig, um die Positionsdaten mit dem vorhandenen Kartenmaterial abzugleichen. Solche Verfahren werden Map Matching Algorithmen genannt.

Schreibweise von
Map-Matching
im gesamten
Dokument verbessern

Das Ziel eines Map Matching Algorithmus ist es also, für eine gegebene Abfolge von gemessenen Positionsdaten (GPS-Track) eine entsprechende Folge von bereinigten Positionen, sogenannten Match-Punkten zu finden. Dazu wird zunächst jeder Position des GPS-Tracks ein Straßenabschnitt der digitalen Karte zugeordnet und anschließend durch Projektion auf diesen Abschnitt der Match-Punkt bestimmt. Hierbei stellt sich vor allem die Wahl des richtigen Straßenabschnitts als schwierig heraus.

Im Gegensatz zum Routing gibt es beim Map Matching eine große Anzahl von verschiedenen Lösungsansätzen, die von einfachen geometrischen Verfahren, über topologische

Algorithmen bis hin zu komplexeren Methoden, die Kalman Filter oder Fuzzy Logic verwenden, gehen.

Satzkonstruktion
unschön.

Aus Zeitgründen einigten mein Betreuer und ich mich darauf, dass ich zunächst ein paar einfache geometrische Verfahren und anschließend zwei topologische Map Matching Algorithmen implementieren sollte. Da mein Betreuer seine Diplomarbeit über dieses Thema geschrieben hatte, konnte er mich mit den nötigen wissenschaftlichen Artikeln zu diesem Thema versorgen, sodass mir eine langwierige Einarbeitungsphase erspart blieb. Im folgenden möchte ich einen kurzen Überblick über die von mir implementierten Map Matching Verfahren geben.

Soll ich noch
aufschreiben,
dass ich im Ge-
gensatz zum
Routing erst
noch Testfälle
für das Map
Matching her-
aussuchen muss-
te?

Point-To-Node Map Matcher

Das einfachste Map Matching Verfahren ordnet jeder Position eines GPS-Tracks denjenigen Knoten des Straßennetzes zu, der die geringste Distanz zu dieser Position besitzt. Ein solches Verfahren wird Point-To-Node Map Matching genannt. Die Implementierung dieses Map Matchers war unkompliziert, allerdings zeigte sich schon nach wenigen Beispielanwendungen, dass die gelieferten Ergebnisse mehr als unbefriedigend waren. Grund hierfür ist, dass der Point-To-Node Map Matcher nicht die Abfolge der einzelnen Positionen des GPS-Tracks berücksichtigt und deshalb öfters zwischen unzusammenhängenden Straßenabschnitten hin und her springt. Darüber hinaus ist dieses Verfahren sehr stark von der Anzahl der Knoten im Kartenmaterial abhängig. Dies wird durch den Point-To-Road Map Matcher behoben.

Ist das zu viel,
wenn ich über
jeden einzelnen
Map-Matcher
schreiben?

Point-To-Road Map Matcher Im Gegensatz zum Point-To-Node Map Matcher bestimmt dieses Verfahren zunächst die Entfernung zu allen umliegenden Straßenabschnitten und wählt schließlich den Abschnitt mit der kleinsten Distanz aus. Anschließend wird durch Projektion der Match-Punkt auf dem gewählten Straßenabschnitt bestimmt. Dieses Verfahren berücksichtigt zwar auch nicht die zeitliche Abfolge der GPS-Punkte, lieferte allerdings in den Tests deutlich bessere Ergebnisse als der Point-To-Node Map Matcher. Die meisten Probleme traten dabei in der Nähe von Kreuzungen oder bei dicht beieinander liegenden Straßen auf.

Die Implementierung des Point-To-Road Map Matchers war im Vergleich zu der des vorhergehenden Algorithmus aufwendiger, was vor allem an der Berechnung der Projektionen auf die Straßenabschnitte lag.

Soll ich hier
noch irgendwas
anderes schrei-
ben? Ich habe
das Gefühl, dass
ich nicht wirk-
lich viel über
meine Arbeit
schreibe!

Topologisch gewichtete Map Matcher Topologische Map Matcher beziehen neben rein geometrischen Informationen auch die Topologie des Straßennetzes in den Map Matching Prozess mit ein. Unter Topologie versteht man hierbei alle Informationen die den Zusammenhang von Straßenabschnitten untereinander beschreiben. Dadurch wird auch die Abfolge der Positionen im GPS-Track berücksichtigt.

Im einzelnen gehen topologische Map Matching Algorithmen wie folgt vor. Zunächst wird durch eine Radiussuche um die aktuelle GPS-Position eine Menge von möglichen Straßenabschnitten bestimmt. Anschließend erhält jeder dieser Kandidaten ein Gewicht,

Falsches Wort?

das durch die Geometrie, die Topologie und weiteren Kriterien berechnet wird. Der Straßenabschnitt mit dem geringsten Gewicht wird schließlich ausgewählt und der Match-Punkt wie im Point-To-Road Map Matcher bestimmt. Die beiden von mir implementierten Map Matching Algorithmen bezogen für die Gewichts Berechnung zusätzlich die Geschwindigkeit und den Kurs des Fahrzeugs sowie Informationen zu Abbiegeverboten und Einbahnstraßen mit ein.

oder beziehen?

Ich habe noch einen weiteren ähnlichen Map Matcher geschrieben, soll ich den auch noch beschreiben?

Weitere Tätigkeiten beschreiben?

4 Quellen

- Homepage der Fraunhofer Gesellschaft: www.fraunhofer.de