

Robustes Mapmatching hoch aufgelöster, fahrzeugbasierter GPS-Tracks

Wolf A. Heidrich, Martin Schulze

Moritz Kessel, Martin Werner

Telemotive AG
Frankfurter Ring 115a
80807 München

wolf.heidrich@telemotive.de
martin.schulze@telemotive.de

Ludwig-Maximilians-Universität München
Oettingenstr. 67
80538 München

moritz.kessel@ifi.lmu.de
martin.werner@ifi.lmu.de

Abstract: Die Verwendung von mobilen GPS-Empfängern hat in den vergangenen Jahren drastisch zugenommen. Somit stehen mittlerweile nicht nur für den privaten Gebrauch auf dem eigenen Smartphone, sondern auch für den Telematikbereich hochgenaue Positionsdaten zur Verfügung. Diese können verwendet werden um darauf aufbauend fortschrittliche Dienste zu entwickeln. Um diese Positionsdaten sinnvoll nutzen zu können, ist eine „Mapmatching“ genannte Technik notwendig, die die einzelnen Positionsmessungen mit den Straßen einer digitalen Karte verknüpft. Diese Arbeit entwirft einen modifizierten Mapmatching-Ansatz, der auf der „Multiple Hypothesis Technique“ basiert und für zuverlässiges Mapmatching hoch aufgelöster GPS-Tracks in ländlichen, suburbanen und urbanen Gebieten optimiert ist. Dabei löst der vorgestellte Ansatz Probleme, die unter anderem durch Brücken, Tunnel und mehrdeutige Situationen beim Mapmatching entstehen können.

1 Einleitung

Die weiterhin zunehmende Verbreitung von günstigen und immer genauer werdenden GPS-Empfängern bietet die Möglichkeit, die generierten Positionsdaten zum Aufbau fortschrittlicher Dienste zu nutzen. Im Segment der tragbaren GPS-Empfänger, die heutzutage in vielen Smartphones zur Verfügung stehen, entwickeln sich die sogenannten ortsbasierten Dienste (location-based services, kurz: LBS) stetig fort und erfreuen sich einer immer größeren Akzeptanz durch den Endnutzer.

Jedoch ist diese Entwicklung nicht nur auf den Fußgänger begrenzt. Auch und insbesondere im Automobilbereich bietet die hohe Verfügbarkeit von Positionsdaten hoher Genauigkeit die Möglichkeit, diese zum Aufbau weiterer Dienste zu nutzen. Diese Dienste werden unter dem Oberbegriff „intelligente Transportsysteme“ (intelligent transportation system, kurz: ITS) zusammengefasst und umfassen Anwendungsgebiete sowohl im öffentlichen Bereich (z.B.: Verkehrsanalyse, Emissionsanalyse), als auch im privaten Bereich (z.B.: Navigation).

Zur Nutzung der gewonnenen Positionsdaten in Form von fortgeschrittenen Diensten, ist es meistens notwendig, diese auf eine digitale Karte zu projizieren. Diese Technik nennt sich „Mapmatching“. Je genauer das Mapmatching durchgeführt wird, desto genauer ist die Position des Anwenders in dem digitalen Straßennetz bekannt und desto besser kann ein auf diesen Daten aufsetzender Dienst seine Leistungen erbringen.

In dieser Arbeit stellen wir einen modifizierten MHT-Algorithmus vor, der insbesondere für die Verarbeitung von hoch aufgelösten GPS-Tracks in komplexen Situationen und zur Rekonstruktion von gefahrenen Strecken optimiert ist.

2 Verwandte Arbeiten

Das Thema Mapmatching ist seit einigen Jahren Gegenstand vieler Forschungsbemühungen. In der Literatur wird dabei zwischen zwei grundlegenden Methoden, der inkrementellen und der globalen, unterschieden.

Die inkrementelle Methode wird in der Literatur teilweise auch im Kontrast zur globalen Methode als „lokale Methode“ bezeichnet [L⁺09]. Sie zeichnet sich dadurch aus, dass für jede Positionsmessung jeweils das lokal bestmögliche Mapmatching-Ergebnis gesucht wird. Oftmals werden dabei die Positionsmessungen sequentiell verarbeitet, so dass das Mapmatching-Ergebnis eines Trackpoints P_t (t bezeichnet einen Zeitpunkt) direkt von dessen Vorgänger P_{t-1} abhängt [B⁺05]. Der Vorteil dieser Methode liegt darin, dass es mit ihr möglich ist, schnell und mit geringerem Rechenaufwand Mapmatching-Ergebnisse zu erhalten. Sie eignet sich insbesondere zur Echtzeitverarbeitung, da die Datenakquisition nicht abgeschlossen sein muss, um das Mapmatching durchführen zu können. Der Nachteil ist allerdings, dass die Qualität des Mapmatchings im Vergleich mit der globalen Methode schlechter abschneidet, da die zukünftigen Messdaten unbekannt sind und somit auch nicht in das Mapmatching mit einfließen können [W⁺07].

Bei der globalen Methode wird versucht, den gesamten Verlauf eines Tracks auf eine digitale Karte zu matchen. Hierbei werden mögliche Pfade in der Karte betrachtet und anschließend wird anhand verschiedener Messkriterien der am wahrscheinlichsten zu den Positionsmessungen passende Pfad ausgewählt [WSP06]. Als Messkriterium wird in der Literatur unter anderem die (schwache) Fréchet Distanz verwendet [B⁺05]. Der Vorteil dieser Methode besteht darin, dass sie meistens genauere Ergebnisse beim Mapmatching liefert als die inkrementelle Methode. Die Nachteile der globalen Methode sind, dass diese Genauigkeit durch eine höhere benötigte Rechenleistung erkaufte werden muss und dass sie sich nicht zum Mapmatching in Echtzeit eignet, da der gesamte Verlauf eines Tracks mit in die Berechnung einfließen muss [W⁺07].

Weiterhin werden Mapmatching-Algorithmen in die folgenden Klassen unterteilt: geometrische, topologische, wahrscheinlichkeitsbasierte sowie hybride und erweiterte Algorithmen [QON07].

Ein geometrischer Mapmatching-Algorithmus nutzt für seine Aufgabe nur die geometrischen Informationen der Ecken und Kanten einer digitalen Karte [QON07]. In der Literatur existieren im wesentlichen drei Ansätze für geometrische Algorithmen: das sogenannte

„Point-to-Point Mapmatching“, das „Point-to-Curve Mapmatching“ und das „Curve-to-Curve Mapmatching“.

Ein topologischer Mapmatching-Algorithmus verwendet für seine Aufgabe zusätzlich zu den geometrischen Methoden die Beziehungen zwischen Knoten und Kanten eines digitalen Straßennetzes. Greenfeld versucht in [Gre02] jede neu eingetroffene GPS-Positionsmessung P_t auf die letzte gematchte Kante A_i zu matchen, indem er zwischen der letzten gematchten GPS-Positionsmessung P_{t-1} und P_t linear interpoliert und anhand zweier Ähnlichkeitskriterien, der Richtung und der Nähe der beiden Kanten A_i und $\overline{P_{t-1}P_t}$ zueinander, bestimmt, ob die aktuelle GPS-Positionsmessung auf A_i gematcht werden kann. Diese Methode wird jedoch sehr stark von Ausreißern beeinflusst [QON07]. Fortgeschrittenere Ansätze verwenden zusätzliche semantische Informationen, wie zum Beispiel Abbiege- und Wendeverbote und Konnektivitäten von Kanten. Velaga et al. verwenden in [VQB09] die genannten zusätzlichen Informationen und versuchen außerdem zu erkennen, ob ein Link verlassen wurde. Wurde der Link nicht verlassen, wird davon ausgegangen, dass ein Trackpoint auf den letzten gematchten Link gematcht werden kann.

Die Klasse der wahrscheinlichkeitsbasierten Algorithmen verwendet Fehlerregionen um die GPS-Positionsmessungen herum, die oft elliptisch oder rechteckig sind und identifiziert Knoten und Kanten des digitalen Straßennetzwerks, die sich innerhalb dieser Fehlerregionen befinden [QON07]. Der Radius beziehungsweise Umfang einer solchen Fehlerregion soll anhand des Fehlers des zur Aufzeichnung der GPS-Positionsmessungen verwendeten Geräts abgeleitet werden können. Befindet sich mehr als ein Objekt innerhalb der Fehlerregion, so werden wie bei den topologischen Algorithmen Ähnlichkeitsmaße verwendet um das beste Mapmatching-Ergebnis zu ermitteln [Zha97]. Dieser Ansatz wurde von Ochieng et al. in [OQN03] derart verfeinert, dass die Fehlerregion nur noch dann erzeugt wird, wenn ein Fahrzeug einen Link verlässt. Dadurch wird der Rechenaufwand deutlich verringert und es sollen falsche Mapmatching-Ergebnisse, die auftreten können, wenn innerhalb einer Fehlerregion mehrere Kandidaten-Links identifiziert werden, obwohl das Fahrzeug den letzten gematchten Link nicht verlassen hat, verhindert werden. Die Erkennung, ob ein Fahrzeug einen Link verlassen hat, ist bei diesem Ansatz kritisch für das Funktionieren des Algorithmus.

Unter der Bezeichnung hybride und erweiterte Algorithmen werden alle Algorithmen zusammengefasst, die sich nicht in die obigen Kategorien einteilen lassen. Diese verwenden oft Kombinationen aus den obig genannten Algorithmen oder nutzen komplexere mathematische Methoden.

Pink et al. verwenden in [PH08] zum Beispiel einen erweiterten Kalman-Filter zur Stabilisierung der GPS-Rohdaten und nehmen Veränderungen an den digitalen Karten vor, indem das Straßennetz zwischen den Stützpunkten anstatt mit der sonst üblichen linearen Interpolation mithilfe von kubischen Splines interpoliert wird. Dadurch soll in Situationen, die bei linearer Interpolation des Straßennetzes mehrdeutig wären, ein eindeutiges Mapmatching-Ergebnis herbeigeführt werden.

Syed et al. verwenden in ihrem dreistufigen Algorithmus Fuzzy Logic für die Identifikation von Links, die ein korrektes Mapmatching-Ergebnis darstellen könnten [SC04].

Schüssler et al. nutzen in [SA09] die MHT für ihren Mapmatching-Algorithmus. Die Multiple-Hypothesis-Technique wurde ursprünglich von Reid für die gleichzeitige Verfolgung mehrerer Ziele mithilfe eines Radarsystems entwickelt [Rei79]. Wird sie in einem Mapmatching-Algorithmus verwendet, so werden während des Mapmatchings verschiedene Hypothesen gebildet. Der Mapmatching-Algorithmus von Schüssler et al. ist zweistufig. Beim initialen Mapmatching, mit dem das Mapmatching am ersten Trackpoint beginnt, wird innerhalb eines Radius von 750 Metern um den aktuellen Trackpoint nach Kandidaten-Links gesucht. Wurden mindestens 25 Kandidaten-Links gefunden, bricht die Suche ab. Wurden weniger als 25 Kandidaten-Links gefunden, wird der Suchradius vergrößert, bis mindestens 25 Kandidaten-Links gefunden wurden. Den Kandidaten-Links wird mit einem Gewichtungsverfahren ein Gewicht in Form einer Punktzahl zugeordnet. Anschließend wird für jeden der Kandidaten-Links eine Hypothese erstellt, so dass der Algorithmus mit mindestens 25 Hypothesen initialisiert ist. Das Gewicht einer Hypothese ist die Summe der Gewichte ihrer gematchten Links. Für jeden weiteren Trackpoint existieren nun zwei Möglichkeiten, nach denen der Algorithmus fortfahren kann: der Trackpoint hat den aktuellen Link nicht verlassen oder der Trackpoint hat den aktuellen Link verlassen. Im ersten Fall wird der Trackpoint auf den letzten gematchten Link der Hypothese gematcht. Im zweiten Fall wird nach Kandidaten-Links gesucht, die mit dem letzten gematchten Link verbunden sind. Für jeden dieser Kandidaten-Links wird nun eine neue Hypothese mit dem Kandidaten-Link als letztem Link und Mapmatching-Ergebnis für den Trackpoint erstellt. Wurden alle Hypothesen für einen Trackpoint verarbeitet, wird eine Reduktion vorgenommen. Die Menge der Hypothesen wird so lange um die Hypothese mit dem höchsten Gewicht verringert, bis nur noch eine vorher definierte, konstante Anzahl an Hypothesen existiert. Anschließend fährt der Algorithmus mit dem nächsten Trackpoint fort.

Die genannten Arbeiten decken nur einen kleinen Teil der Algorithmen ab, die dieser Kategorie zugeordnet werden können. Für eine – zumindest bis zum Jahr 2007 – vollständigere Übersicht wird die Literatur von [QON07] empfohlen.

3 Algorithmus

Einen Mapmatching-Algorithmus auf Basis der MHT zu verwenden, basiert auf der Idee, dass dadurch einige allgemein auftretende Probleme beim Mapmatching, wie z.B. mehrdeutige Situationen, inhärent gelöst werden können, ohne dass eine gesonderte Behandlung notwendig wird. Der Kern des Algorithmus besteht aus drei Stufen, die dem Algorithmus aus [VQB09] entlehnt sind: dem initialen Mapmatching, dem Mapmatching auf einem Link und dem Mapmatching an einer Junction. Diese werden im Folgenden „MatchModes“ genannt. Vier der fünf verwendeten Gewichtsfunktionen zur Identifikation von Kandidaten-Links werden ebenfalls aus [VQB09] übernommen. Dies sind die Gewichtsfunktionen für die Distanz, das Heading, die Konnektivität und die Abbiege-Einschränkungen. Alle Gewichtsfunktionen fließen gleich stark gewichtet in die Bewertung des Gesamtgewichts eines Kandidaten-Links ein. Bei der MHT wird, wie der Name schon sagt, nicht nur ein mögliches Mapmatching-Ergebnis erzeugt, sondern mehrere. Aus diesen wird am

Ende des Mapmatchings dann das beste Ergebnis ausgewählt. Diese möglichen Mapmatching-Ergebnisse werden Hypothesen genannt, gelten für einen gesamten Track, sind insofern einzigartig, als dass keine Hypothese einer anderen Hypothese gleicht und beinhalten für jeden Trackpoint Informationen über den Verlauf des Mapmatchings. Bestandteile einer Hypothese sind: die gematchten Links, die Projektionen auf den gematchten Links, die Downstream-Junctions der gematchten Links, die Distanzen der Projektionen von den Downstream-Junctions auf den gematchten Links, die Upstream-Junctions der gematchten Links, die Information in welchem MatchMode der Trackpoint gematcht wurde, die Information, ob MatchMode mit Sicherheit der richtige MatchMode war, die Heading-Differenzen zwischen dem Heading des Trackpoints und den Headings der gematchten Links, so, wie sie bei der Gewichtung der gematchten Links ermittelt wurden, die Anzahl der während des Mapmatchings aus den gematchten Links entfernten Links, die Information, wie oft eine Abbiege-Einschränkung missachtet wurde und ihr Gesamtgewicht, das sich aus der Summe der Gewichte der gematchten Links berechnet.

Als wesentlich neue Bestandteile des Verfahrens werden im Folgenden eingeführt: eine weitere Gewichtsfunktion, deren Existenz und Verwendung zentral für die Bearbeitung vieler der benannten Herausforderungen beim Mapmatching ist sowie eine neue Vorgehensweise beim Mapmatching, die nach einem (besseren) senkrechten Match sucht. Zum Schluss wird die Reduktion der Hypothesen erläutert, die ebenfalls ein ganz zentraler Bestandteil dieses Mapmatching-Algorithmus ist.

3.1 Hypothesenaufbau

Der Hypothesenaufbau umfasst die drei MatchModes und somit auch die Gewichtsfunktionen für Kandidaten-Links. Als neue zusätzliche Gewichtsfunktion wird eine Funktion vorgeschlagen, die einen Kandidaten-Link stärker gewichtet, wenn eine senkrechte Projektion des Trackpoints auf diesen möglich ist und dementsprechend den Kandidaten-Link schwächer gewichtet, wenn eine solche senkrechte Projektion nicht möglich ist. Dieser Vorschlag basiert auf der Beobachtung, dass eine senkrechte Projektion eines Trackpoints in vielen Fällen auch ein richtiges Mapmatching-Ergebnis darstellt.

Das initiale Mapmatching funktioniert folgendermaßen: Für jeden Kandidaten-Link, dessen Gewicht nicht unter einem gewissen Abbruchkriterium liegt, wird eine eigene Hypothese erzeugt. Damit ist sichergestellt, dass falls der Kandidaten-Link mit dem höchsten Gewicht aus irgendeinem Grund nicht der richtige Link ist, das richtige Ergebnis zunächst nicht verworfen wird.

Beim Mapmatching auf einem Link wird entgegen der Vorgehensweise in [VQB09] der aktuelle Trackpoint nicht einfach nur auf den letzten gematchten Link gematcht, sondern es wird vorher nach einem besseren senkrechten Match gesucht. Diese Maßnahme hilft zusätzlich, das zu späte Verlassen eines Links zu bemerken und sorgt dafür, dass Kurven mit realistischeren Abständen zwischen den projizierten Positionen gematcht werden können.

Das Mapmatching an einer Junction schließlich verläuft wie folgt. Für jeden Kandidaten-Link wird eine neue Hypothese erstellt, wenn dessen Gewicht nicht unter einem gewissen Abbruchkriterium liegt. Von dieser Regelung werden Kandidaten-Links ausgenommen, wenn sie ein Tunnel sind.

Mit der Einführung der neuen Gewichtsfunktion, die die Orthogonalität der Projektion eines Trackpoints auf einen Link bewertet, wird es möglich ein neues Verfahren zum Mapmatching zu verwenden. Dieses Verfahren sucht einen senkrechten Match, also einen Kandidaten-Link, auf den ein Trackpoint senkrecht projiziert werden kann. Dazu wird eine Breitensuche auf den Links des Straßennetzes durchgeführt, deren Anfang der aktuelle Kandidaten-Link ist. Während der Durchführung dieser Breitensuche gilt die Annahme, dass der letzte Link einer Unterhypothese immer komplett, das heißt von der Upstream-Junction bis zur Downstream-Junction durchfahren wird, solange bis der beste senkrechte Match gefunden wird. Das Gewicht des aktuellen Kandidaten-Links wird als Basisgewicht verwendet, das von dem (besseren) senkrechten Match übertroffen werden muss. War der aktuelle Kandidaten-Link kein senkrechter Match, dann wird sein Basisgewicht so verwendet, als sei ein senkrechter Match auf diesem möglich gewesen. Das liegt darin begründet, dass auch ein nicht senkrechter Match ein korrekter Match sein kann, der keineswegs durch einen schlechteren senkrechten Match verdrängt werden sollte. Dieser Algorithmus erstellt Unterhypothesen, die einen einzigartigen zusammenhängenden Pfad im Straßennetz darstellen. Für diese wird ein Gesamtgewicht errechnet, das sich aus der Summe der Gewichte der enthaltenen Links zusammensetzt. Ein senkrechter Match ist gefunden, wenn eine senkrechte Projektion des Trackpoints auf den Kandidaten-Link möglich ist. Das Gewicht dieses Kandidaten-Links wird das Gewicht des aktuell besten senkrechten Matches. Ein besserer senkrechter Match ist gefunden, wenn die senkrechte Projektion des Trackpoints auf einen Kandidaten-Link möglich ist, dessen Gewicht größer ist als das Gewicht des aktuell besten senkrechten Matches. Der Algorithmus sucht so lange nach einem besseren senkrechten Match, bis für alle Unterhypothesen entweder der beste senkrechte Match gefunden wurde, eine Fahrt im Kreis entdeckt wurde oder ein Abbruchkriterium erreicht wird. Der beste senkrechte Match ist gefunden, wenn ein senkrechter Match gefunden wird, dessen Gewicht kleiner ist, als das Gewicht des aktuell besten Matches. In diesem Fall wird davon ausgegangen, dass der aktuell beste senkrechte Match auch der insgesamt beste senkrechte Match für diese Unterhypothese ist. Eine Fahrt im Kreis wird gefunden, wenn während der Breitensuche ein Link gefunden wurde, der bereits Teil der aktuellen Unterhypothese ist. Das Abbruchkriterium ist dasselbe, das auch für das „normale“ Mapmatching verwendet wird. Es wird erreicht, wenn alle Kandidaten-Links (mit Ausnahme von Tunnel-Links) einer Unterhypothese ein Gewicht aufweisen, das unter dem Abbruchkriterium liegt. Das Ergebnis des Algorithmus ist eine Anzahl von Unterhypothesen, die einen besten senkrechten Match enthalten. Diese Unterhypothesen müssen zunächst noch gefiltert werden. Es muss sichergestellt sein, dass der letzte gematchte Link des letzten gematchten Trackpoints mit dem ersten Link einer Unterhypothese nicht übereinstimmt, da davon ausgegangen wird, dass der letzte gematchte Link – unabhängig vom MatchMode – in jedem Fall verlassen wurde, wenn ein besserer senkrechter Match gefunden wurde. Anschließend wird jede Unterhypothese daraufhin überprüft, ob sie den letzten gematchten Link des letzten Trackpoints an irgendeiner Stelle enthält. Ist das der Fall, so liegt offensichtlich eine Kreisfahrt vor und die entsprechen-

de Unterhypothese wird entfernt. Abschließend muss, falls der Aufruf aus „Mapmatching auf einem Link erfolgte“, sichergestellt werden, dass nur jeweils eine Unterhypothese mit demselben letzten gematchten Link existiert. Existieren mehrere solcher Unterhypothesen, die verschieden sein müssen, da Unterhypothesen einzigartig sind, wurden unterschiedliche Pfade zum besten senkrechten Match gefunden. Da ein Fahrzeug jedoch nur einen Weg gefahren sein kann, wird davon ausgegangen, dass in diesem Fall nur die kürzeste Unterhypothese, also diejenige, die die wenigsten gematchten Links enthält, die Richtige sein kann. Diese Annahme wird dadurch untermauert, dass das Fahrzeug aufgrund des MatchModes „Mapmatching auf einem Link“ den letzten gematchten Link des letzten Trackpoints eigentlich nicht verlassen hätte, wäre die Suche nach einem besseren senkrechten Match nicht aufgerufen worden, woraus geschlossen wird, dass das Fahrzeug nicht so weit gekommen sein kann. Alle Unterhypothesen, die auf demselben letzten gematchten Link enden und nicht die minimale Anzahl an gematchten Links beinhalten, werden daher in diesem Fall entfernt. Für jede der verbleibenden Unterhypothesen wird schließlich unabhängig vom MatchMode, aus dem die Suche nach einem besseren senkrechten Match aufgerufen wurde, eine neue Hypothese erstellt.

3.2 Hypothesenreduktion

Da an jeder Junction für jeden Kandidaten-Link eine neue Hypothese erzeugt wird, ist klar, dass bis zum Ende des Mapmatchings nicht alle Hypothesen behalten werden können, da irgendwann die Grenzen des Arbeitsspeichers erreicht werden. Aus diesem Grund wird es notwendig, die Anzahl der Hypothesen anhand einer Strategie zu reduzieren. Schüssler et al. zum Beispiel setzen hier ein festes Limit von N_{max} Hypothesen ein. Existieren mehr Hypothesen, so werden nur die besten N_{max} Hypothesen behalten [SA09]. Dieses Vorgehen ist offensichtlich gefährlich, da zu keinem Zeitpunkt garantiert ist, dass die richtige Hypothese nicht verworfen werden kann.

Eine andere wichtige Frage ist, wann diese Reduktion durchgeführt werden soll. Schüssler et al. führen ihre Reduktion nach dem Mapmatching jedes Trackpoints durch, wobei alle Hypothesen davon betroffen sein können [SA09]. Dieses Vorgehen hat nicht nur den Nachteil, dass es einen hohen Berechnungs-Overhead erzeugt, sondern auch den Nachteil, dass mit etwas Pech eine Hypothese, die den letzten Link korrekt gematcht hatte, verschwindet. Eventuell hätte sich jedoch zu einem späteren Zeitpunkt eine bessere Bewertung ergeben.

Aus den beiden vorgenannten Gründen wird vorgeschlagen, die Reduktion der Hypothesen nur noch dann durchzuführen, wenn sie tatsächlich nötig wird. Weiterhin wird ein Mechanismus vorgeschlagen, der zuverlässig falsche Hypothesen verwirft. Die dynamische Reduktion der Hypothesen basiert auf folgenden Annahmen:

1. Eine Hypothese, die keinen Konkurrenten hat, braucht nicht für eine Reduktion in Betracht gezogen zu werden. Der Konkurrent einer Hypothese ist eine Hypothese, die auf dem gleichen Link endet und in dieselbe Richtung auf diesem Link gerichtet ist (d.h., dass die Upstream- und Downstream-Junctions übereinstimmen).

2. Es kann mehrere Hypothesen zwischen einem Start- und einem Ziel-Link geben. Ein Fahrzeug kann jedoch nur immer einen Weg genommen haben. Das heißt, dass es auch nur eine richtige Hypothese zwischen einem Start- und einem Ziel-Link geben kann.

Die Umsetzung der ersten Annahme führt dazu, dass nur noch eine Reduktion für Hypothesen in Betracht gezogen wird, die auf demselben Link enden und in die gleiche Richtung verlaufen. Dadurch wird sichergestellt, dass Hypothesen, die im Moment schlecht sind, nicht sofort verworfen werden, sondern frühestens dann, wenn sie mit einer Hypothese zusammenlaufen, die besser ist. In Kombination mit der zweiten Annahme folgt, dass die Hypothesen gruppiert reduziert werden müssen. Zwei Hypothesen, deren letzte gematchte Links und Upstream- bzw. Downstream-Junctions nicht identisch sind, brauchen nicht gegeneinander evaluiert zu werden, da sie keine Konkurrenten sind.

Die zuverlässige Erkennung falscher Hypothesen ist nicht trivial. Es gibt in Bezug auf das Gesamtgewicht einer Hypothese mehrere Phänomene, die beim Mapmatching auftreten können: Erstens kann es passieren, dass eine Hypothese, die falsch ist, ein höheres Gewicht bekommt als die richtige Hypothese, wenn die richtige Hypothese durch einen Tunnel läuft, wo das Gesamtgewicht abnimmt, da die Links aus denen der Tunnel besteht, negative Gewichte bekommen, während die falsche Hypothese einfach abbricht und nach dem Tunnel mit initialem Mapmatching fortfährt. Zweitens kann es passieren, dass eine Hypothese, die falsch ist, ein höheres Gewicht bekommt als die richtige Hypothese, wenn ein sogenannter „falscher“ U-Turn auftritt. Drittens kann es passieren, dass eine Hypothese, die falsch ist, ein höheres Gewicht bekommt als die richtige Hypothese, wenn während des Mapmatchings eine Abbiege-Einschränkung missachtet wird. Viertens kann es aufgrund der entwickelten Technik der Suche nach einem senkrechten Match passieren, dass eine Hypothese, die falsch ist, ein höheres Gewicht bekommt als die richtige Hypothese. Fünftens kann es passieren, dass eine Hypothese, die falsch ist, ein höheres Gewicht bekommt als die richtige Hypothese, wenn in einer Offroad-Situation bei einer Hypothese weitergematcht wird, während andere Hypothesen abbrechen.

Beim ersten Phänomen wird offensichtlich der Zusammenhang einer Hypothese verletzt. An wie vielen Stellen eine Hypothese unzusammenhängend ist, kann leicht ermittelt werden, indem für den gesamten bisher gematchten Track geprüft wird, ob die Downstream-Junction eines gematchten Links L_i mit der Upstream-Junction seines gematchten Nachfolgers L_{i+1} übereinstimmt. Eine Hypothese ist besser als ein Konkurrent, wenn sie zusammenhängender als dieser ist. Alle Hypothesen, die weniger zusammenhängend sind als ihre Konkurrenten können deswegen entfernt werden.

Ein „falscher“ U-Turn ist ein U-Turn, der real nicht stattgefunden hat. Dieses Phänomen tritt besonders häufig an Kurven auf. Hat der Mapmatching-Algorithmus an einer Junction mehrere Kandidaten-Links zur Verfügung, so wird für jeden dieser Links eine Hypothese erstellt. Liegen die Trackpoints nun ungünstig, kann es passieren, dass die Hypothese, die zunächst falsch abbiegt dann umdreht und schließlich richtig fährt, ein deutlich besseres Gewicht erhält als die richtige Hypothese. Ob ein U-Turn ein falscher U-Turn ist, ist nicht direkt erkennbar. Ein U-Turn jedoch ist erkennbar, indem die Downstream-Junction zum Zeitpunkt des Betretens eines Links mit der Downstream-Junction zum Zeitpunkt

des Verlassens eines Links verglichen wird. Unterscheiden sie sich, hat ein U-Turn auf dem Link stattgefunden. Diese Information lässt sich nutzen, indem über den gesamten bisher gematchten Verlauf des Tracks nach U-Turns gesucht wird. Aufgrund der Annahme, dass von mehreren Hypothesen, die denselben Start- und Ziel-Link und auf dem Ziel-Link dieselbe Upstream- bzw. Downstream-Junction haben, nur eine richtig sein kann und aufgrund der Tatsache, dass eine Hypothese einzigartig ist, kann man sagen, dass eine Hypothese, die weniger U-Turns enthält als die anderen, die bessere ist. Auf diese Weise ist es möglich, alle Hypothesen zu eliminieren, die falsche U-Turns enthalten.

Um dem dritten Phänomen entgegenzutreten, werden alle Hypothesen auf die Anzahl ihrer verletzten Abbiege-Einschränkungen hin untersucht. Nur die Konkurrenten mit den wenigsten Verletzungen werden behalten, da davon ausgegangen werden muss, dass sich der Fahrer eines Fahrzeugs an die Straßenverkehrsordnung hält.

Für die weitere Reduktion werden die zu reduzierenden Hypothesen anhand von fünf zusätzlichen Kriterien bewertet: der minimalen Anzahl der gematchten Links im letzten Trackpoint, der minimalen gemittelten Distanz der Trackpoints von ihren auf die gematchten Links projizierten Positionen, der minimalen gemittelten Heading-Differenz zwischen den Headings der Trackpoints und den Headings der zugehörigen gematchten Links, der maximalen Anzahl an senkrecht gematchten Links und der Orthogonalität des letzten Trackpoints auf dessen letzten gematchten Link. Für Details sei auf [Hei11] verwiesen.

Sind die Hypothesen nach diesen Kriterien reduziert worden, kann es immer noch sein, dass Hypothesen existieren, die noch Konkurrenten besitzen. In diesem Fall wird nur jeweils die am höchsten gewichtete Hypothese jeder Gruppe behalten, so dass am Ende der Reduktion keine Hypothese mehr einen Konkurrenten hat.

Der Ablauf der Hypothesen-Reduktion ist in Abbildung 1 bildlich dargestellt.

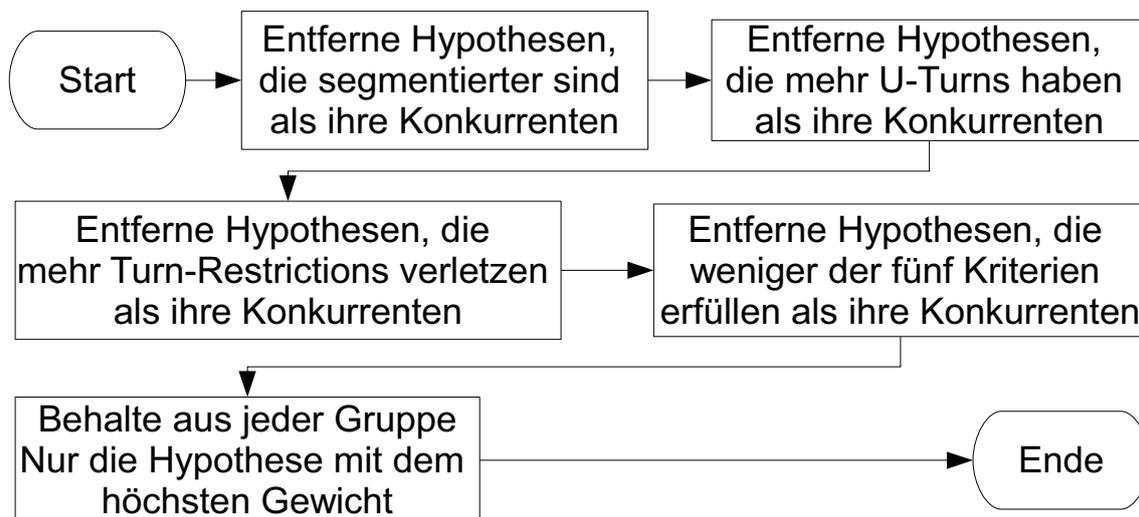


Abbildung 1: Ablauf der Hypothesen-Reduktion

4 Evaluation

Für die Evaluation werden im Folgenden zunächst die Kriterien festgelegt, nach denen der Algorithmus evaluiert werden soll. Danach werden sowohl der Mapmatching-Algorithmus aus [VQB09], der MHT-basierte Mapmatching-Algorithmus aus [SA09] und der erweiterte topologische Mapmatching-Algorithmus aus [Hei11] sowie der in dieser Arbeit vorgestellte MHT-Mapmatching-Algorithmus anhand dieser Kriterien begutachtet. Zum Schluss werden die Ergebnisse ausgewertet.

4.1 Evaluationskriterien

In der Literatur werden typischerweise die beiden folgenden Maße zur Bestimmung der Güte eines Mapmatching-Ergebnisses verwendet: das prozentuale Verhältnis zwischen den korrekt gematchten Trackpoints und der Gesamtzahl der Trackpoints sowie der horizontale Fehler zwischen der realen Position des GPS-Empfängers zu einem diskreten Zeitpunkt und der gematchten Position. Der horizontale Fehler macht im Falle eines Fahrzeuges jedoch keinen Sinn, da ein korrekt gematchtes Ergebnis per definitionem keinen Fehler haben kann, da angenommen wird, dass das Fahrzeug sich auf den Straßen der Karte bewegt. Um die Vergleichbarkeit der Ergebnisse der implementierten Mapmatching-Algorithmen aus [VQB09, SA09, Hei11] mit dem entwickelten Mapmatching-Algorithmus sicherzustellen, werden stattdessen die folgenden drei Maßzahlen hinzugenommen, um die Verlässlichkeit des Ergebnisses zu dokumentieren:

- das Verhältnis zwischen korrekt gematchten Links und tatsächlich befahrenen Links,
- die Zahl falsch gematchter Links und
- die Anzahl der Zusammenhangskomponenten des Ergebnisses.

4.2 Evaluationskonfiguration

Als Testset werden drei Tracks verwendet, deren genauer Verlauf bekannt ist. Alle drei Tracks wurden in München aufgezeichnet. Es wurde jeweils ein Track mit innerstädtischer, suburbaner und ländlicher Straßencharakteristik aufgenommen. Die Samplingrate des GPS-Empfängers betrug 1Hz. Die genauen Daten der Tracks befinden sich in Tabelle 1 auf der nächsten Seite.

Die zur Berechnung verwendeten digitalen Karten stammen von NAVTEQ. Es handelt sich um das GDF-Release Q3 2010 für Deutschland.

Im Folgenden wird die zur Evaluation jedes Mapmatching-Algorithmus verwendete Konfiguration angegeben, sofern sie nicht bereits in dieser Arbeit beschrieben ist, von dieser oder von der Originalkonfiguration (betrifft die Algorithmen aus [VQB09, SA09, Hei11]) abweicht.

Track	Trackpoints	vollständig befahrene Links
Urban	470	19
Suburban	2456	308
Rural	759	90

Tabelle 1: Eckdaten der Evaluationstracks

Der topologische Mapmatching-Algorithmus nach [VQB09] wird mit der folgenden Konfiguration evaluiert.

- Der Radius für die Suche nach Kandidaten-Links wird auf 20 Meter festgesetzt. Dies entspricht ungefähr der maximalen Positionsungenauigkeit des verwendeten GPS-Empfängers.
- Die Gewichtskoeffizienten werden nach [VQB09] wie in Tabelle 2 dargestellt für den jeweiligen Evaluationstrack angepasst.

Gewichtskoeffizient	Urban	Suburban	Rural
H_w	39.99	46.24	44.48
D_w	8.13	44.99	53.52
C_w	36.40	4.46	1
T_w	15.48	4.31	1

Tabelle 2: Parameterkonfiguration für den topologischen Mapmatching-Algorithmus nach [VQB09]

Der MHT-Mapmatching-Algorithmus nach [SA09] wird wie vom Autor vorgeschlagen evaluiert. Die maximale Anzahl an Hypothesen wird auf 40 gesetzt, da der Autor behauptet, dass diese Zahl den besten Kompromiss zwischen Berechnungsdauer und korrektem Mapmatching-Ergebnis ergibt.

4.3 Ergebnisse und Auswertung

Es folgt in diesem Abschnitt die tabellarische Auflistung der Evaluationsergebnisse und deren Auswertung. Die Abkürzungen in den Tabellenköpfen haben folgende Bedeutungen:

- l_k bezeichnet die korrekt befahrenen Links
- l_f bezeichnet die fälschlicherweise befahrenen Links
- k_z bezeichnet die zusammenhängenden Ketten befahrener Links
- tp_k bezeichnet die korrekt gematchten Trackpoints

Die Ergebnisse des Mapmatchings mit dem Algorithmus aus [VQB09] sind in Tabelle 3 zusammengefasst, die Ergebnisse des Mapmatchings mit dem Algorithmus aus [SA09] sind in Tabelle 4 zusammengefasst, die Ergebnisse des Mapmatchings mit dem erweiterten topologischen Mapmatching-Algorithmus aus [Hei11] sind in Tabelle 5 zusammengefasst und die Ergebnisse des Mapmatchings mit dem erweiterten MHT-Mapmatching-Algorithmus aus dieser Arbeit sind in Tabelle 6 zusammengefasst.

Track	l_k (Verhältnis)	l_f	k_z	tp_k (Verhältnis)
Urban	19 (100%)	0	3	447 (95,11%)
Suburban	244 (79,22%)	16	13	2349 (94,49%)
Rural	64 (71,11%)	3	6	694 (91,44%)

Tabelle 3: Evaluationsergebnis: Topologischer Mapmatching-Algorithmus nach [VQB09]

Track	l_k (Verhältnis)	l_f	k_z	tp_k (Verhältnis)
Urban	19 (100%)	0	4	425 (90,43%)
Suburban	251 (81,49%)	49	18	2198 (88,42%)
Rural	29 (32,22%)	2	5	326 (42,95%)

Tabelle 4: Evaluationsergebnis: MHT-Mapmatching-Algorithmus nach [SA09]

Track	l_k (Verhältnis)	l_f	k_z	tp_k (Verhältnis)
Urban	19 (100%)	0	1	452 (96,17%)
Suburban	282 (91,56%)	36	1	2406 (96,78%)
Rural	90 (100%)	0	1	746 (98,29%)

Tabelle 5: Evaluationsergebnis: Erweiterter topologischer Mapmatching-Algorithmus nach [Hei11]

Track	l_k (Verhältnis)	l_f	k_z	tp_k (Verhältnis)
Urban	19 (100%)	0	1	470 (100%)
Suburban	308 (100%)	0	1	2486 (100%)
Rural	90 (100%)	0	1	759 (100%)

Tabelle 6: Evaluationsergebnis: Erweiterter MHT-Mapmatching-Algorithmus

Betrachtet man sich die Ergebnisse der Mapmatching-Algorithmen von [VQB09, SA09, Hei11], stellt man fest, dass die topologischen Mapmatching-Algorithmen aus [VQB09, Hei11] insgesamt deutlich besser abschneiden als der MHT-Mapmatching-Algorithmus aus [SA09]. Daraus kann man jedoch nicht schließen, dass die MHT-Technik generell schlechter beim Mapmatching ist als ein topologischer Ansatz, wie die Ergebnisse des in dieser Arbeit konzeptionierten Mapmatching-Algorithmus zeigen. Dieser ist in allen Punkten gleich gut oder besser als die topologischen Mapmatching-Algorithmen und schneidet insgesamt am besten ab.

Er matcht alle tatsächlich befahrenen Links richtig und matcht keinen nicht befahrenen Link. Er segmentiert keinen der Testtracks und projiziert alle Trackpoints korrekt auf die befahrenen Links. Im Vergleich zu den anderen getesteten Algorithmen müssen bei diesem am wenigsten Parameter konfiguriert werden. Zu konfigurieren ist im Prinzip nur noch der Grenzwert für die Berechnung des Nähe-Gewichts. Bei diesem liefert der voreingestellte Wert von 20 für GPS-Positionsmessungen, die einen „normalen“ maximalen Fehler von 20 Metern haben, hervorragende Ergebnisse und muss in der Theorie nur verändert werden, wenn mit einem deutlich höheren Fehler bei den Positionsmessungen gerechnet werden muss. Prinzipiell existiert ein zweiter zu konfigurierender Parameter, das Abbruchkriterium, das standardmäßig auf 0 gesetzt ist. Dieses umzukonfigurieren ist jedoch in der Realität nicht nötig, da eine Wechselwirkung mit dem Grenzwert zur Berechnung des Nähe-Gewichts existiert. Dieser Algorithmus eignet sich aufgrund dieser Evaluation am besten für ein robustes Mapmatching hoch aufgelöster, fahrzeugbasierter GPS-Tracks.

Literatur

- [B⁺05] Sotiris Brakatsoulas et al. On Map-Matching Vehicle Tracking Data. In *Proceedings of the 31st International Conference on Very Large Data Bases, VLDB '05*, Seiten 853–864. VLDB Endowment, 2005.
- [Gre02] Joshua S. Greenfeld. Matching GPS Observations to Locations on a Digital Map. Bericht, New Jersey Institute of Technology, 2002.
- [Hei11] Wolf A. Heidrich. Mapmatching von GPS-Tracks zur automatisierten Qualitätsanalyse von Verkehrsinformationen. Diplomarbeit, Ludwig-Maximilians-Universität München, Juni 2011.
- [L⁺09] Yin Lou et al. Map-Matching for Low-Sampling-Rate GPS Trajectories. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09*, Seiten 352–361, New York, NY, USA, 2009. ACM.
- [OQN03] W. Y. Ochieng, M. Quddus und R. B. Noland. Map-Matching in Complex Urban Road Networks. *Brasilianische Zeitschrift der Kartographie (Revista Brasileira de Cartografia)*, 55(2):1–18, Dezember 2003.
- [PH08] Oliver Pink und Britta Hummel. A statistical approach to map matching using road network geometry, topology and vehicular motion constraints. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, Seiten 862–867, Oktober 2008.
- [QON07] Mohammed A. Quddus, Washington Y. Ochieng und Robert B. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15:312–328, Oktober 2007.
- [Rei79] Donald B. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24:843–854, 1979.
- [SA09] Nadine Schüssler und Kay W. Axhausen. Map-matching of GPS traces on high-resolution navigation networks using the Multiple Hypothesis Technique (MHT). Bericht, Institut für Verkehrsplanung und Transportsysteme, Zürich, Oktober 2009.

- [SC04] S. Syed und M. E. Cannon. Fuzzy Logic Based-Map Matching Algorithm for Vehicle Navigation System in Urban Canyons. In *ION National Technical Meeting*, Seiten 1–12, Januar 2004.
- [VQB09] Nagendra R. Velaga, Mohammed A. Quddus und Abigail L. Bristow. Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. *Transportation Research Part C: Emerging Technologies*, 17(6):672–683, May 2009.
- [W⁺07] Dongdong Wu et al. A Heuristic Map-Matching Algorithm by Using Vector-Based Recognition. In *Proceedings of the International Multi-Conference on Computing in the Global Information Technology*, Seite 18ff, Washington, DC, USA, 2007. IEEE Computer Society.
- [WSP06] Carola Wenk, Randall Salas und Dieter Pfoser. Addressing the Need for Map-Matching Speed: Localizing Global Curve-Matching Algorithms. In *18th International Conference on Scientific and Statistical Database Management*, Seiten 379–388, Juli 2006.
- [Zha97] Yilin Zhao. *Vehicle Location and Navigation Systems*. Artech House Inc, 1997.