

Automatische multimodale Verkehrsmoduserkennung und Situationserfassung mit Hilfe von Fuzzy-Regeln

István Pál und Anke Sauerländer-Biebl

*Institut für Verkehrssystemtechnik, DLR, Rutherfordstraße 2, 12489
Berlin (Tel: +49-30 / 67055-7972, e-mail: Istvan.Pal@dlr.de ; Tel:
+49-30 / 67055-605; e-mail: Anke.Sauerlaender-Biebl@dlr.de).*

Multi-Mode-Detektierung, multimodale Verkehrsmoduserkennung,
ModeDetection, Situationserfassung, Fuzzy-Regel, GPS,
Accelerator-Sensor

Abstract: Im Verkehrsmanagement ist die Erfassung von verschiedenen Bewegungsinformationen sowohl bei Personen, z.B. anhand der Geschwindigkeit aus Smartphone-Sensoren oder aus digitalen Bildsequenzen von Außenkameras, als auch bei Fahrzeugen und anderen Objekten von großer Bedeutung. Dadurch können verschiedene Verkehrssituationen besser analysiert und/oder gefährliche Verkehrssituation vorausgesagt werden. Außerdem können verkehrsmittelangepasste Informationen und Applikationen angeboten werden. Diese Aufgabe ist mit technischen beziehungsweise kognitiv-ähnlichen Erkennungsprozessen verbunden. Die vollautomatische Entscheidung, ob eine Datensequenz von mobilen Geräten wie GPS-Positionen oder G-Sensor-Daten zu einem Fußgänger, zu einem Fahrzeug- oder zu einem Bus-, Bahnpassagier gehören, ist nicht immer eindeutig. Die Realisierung anhand des Geschwindigkeitsprofils ohne weitere Informationen (z.B. Karteninformationen) ist immer noch eine technische Herausforderung. Eine Multi-Mode-Erkennung ist noch schwieriger und wurde noch nicht realisiert. Dabei wechselt eine Person seinen Verkehrsmodus (Fußgänger → Bahn → Fahrrad) innerhalb eines Datensatzes. Die vorliegende Arbeit konzentriert sich in erster Linie auf die Untersuchung von Smartphone-Sensordaten und macht Vorschläge, wie verschiedene Bewegungsmodi mit Fuzzy-Regel als Bewegungsmodus-Klassifikatoren erkannt werden können. Abschließend gibt sie einen Ausblick auf mögliche zukünftige Applikationen.

1. HINTERGRUND DER ARBEIT

Im Verkehrsmanagement ist die Erfassung von verschiedenen Bewegungsinformationen sowohl bei Personen als auch bei Fahrzeugen und anderen Objekten von großer Bedeutung. Anhand der Messdaten aus Smartphone-Sensoren oder aus digitalen Bildsequenzen von Außenkameras können verschiedene Verkehrssituationen und Verkehrsteilnehmer besser analysiert und/oder gefährliche Verkehrssituation vorhergesagt werden. Auch können verkehrsmittelangepasste Informationen und Applikationen angeboten werden. Diese Aufgabe ist mit technischen bzw. kognitiv-ähnlichen Erkennungsprozessen verbunden. Die Arbeit fokussiert die Erkennung des Verkehrsmodus anhand von Smartphone-Sensordaten. Die vollautomatische Entscheidung, ob eine Datensequenz von GPS-Positionen oder Accelerator-Sensordaten (Erschütterungen) zu einem Radfahrer, einem Fußgänger, einem PKW-, einem Bus- oder zu einem Bahnpassagier gehören, ist nicht immer eindeutig. Zum Beispiel kann sich ein PKW in der Stadt genau so langsam bewegen wie ein Fahrradfahrer. Die Realisierung anhand des Geschwindigkeitsprofils ohne weitere Zusatzinformationen, wie z.B. ein digitales Straßennetz oder Echtzeitinformationen von ÖPNV-Anbietern, ist immer noch eine technische Herausforderung. Erschwerend kom-

men ungenaue GPS-Ortungen und damit unzuverlässige Geschwindigkeitsermittlungen hinzu. Eine multimodale Verkehrsmodus-Erkennung, das heißt, die sich fortbewegende Person wechselt ihren Verkehrsmodus (Fußgänger → Bahn → Fahrrad), ist noch schwieriger. Ebenso ist die Unterscheidung, bzw. Klassifizierung von Objekten auf den digitalen Bildsequenzen, wie Fahrräder oder Fußgänger nicht einfach. Oft müssen zusätzlich weitere Sensoren (Laser, Radar) samt Fusionierung der Daten mitverwendet werden, die aber nicht immer zur Verfügung stehen. Bei Videodaten ist die Geschwindigkeit der Objekte nicht einmal bekannt und muss mit verschiedenen Bewegungsmodellen geschätzt, bzw. ermittelt werden, die die Abbildungsperspektive und Kameraposition berücksichtigen. Weiterhin müssen Merkmale von Form und Größe betrachtet werden, um z.B. eine gefährliche Verkehrssituation zu erfassen bzw. exakt vorherzusagen zu können. In der vorliegenden Arbeit, die sich in erster Linie auf die Untersuchung von Smartphone-Sensordaten konzentriert, werden Vorschläge gemacht, wie die verschiedenen Verkehrsmodi mit Fuzzy-Regeln als Verkehrsmodus-Klassifikatoren erkannt werden können. Dabei werden die Ausgangsdaten der Fuzzy-Regelung auf die Bewegungsarten (Verkehrsmodi) WALK, BICYCLE, CAR,

TRAIN und BUS abgebildet. Anschließend wird ein Ausblick in die möglichen, zukünftigen Applikationen gegeben.

2. ZIEL DER ARBEIT

Ziel dieser Arbeit ist es, anhand von Smartphone-Sensordaten den Bewegungsmodus/Verkehrsmodus einer Person zu erkennen. Dabei kann die Person ihren Verkehrsmodus beliebig ändern. Die Erkennung beruht hauptsächlich auf Positions- und Acceleratorprofildaten. Die Positionsdaten werden per GPS, aus bekannten WLAN-Standorten oder der Position der aktuellen Funkzelle ermittelt. Der Acceleratorsensor, der auf den meisten gängigen Smartphones vorhanden ist, misst die Beschleunigung in m/s^2 in drei, senkrecht aufeinander stehenden Achsen, die sich am Smartphone ausrichten, siehe Fig. 1:

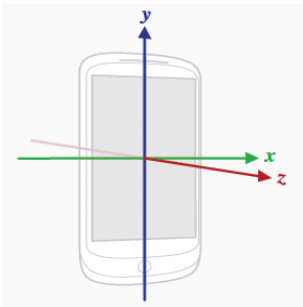


Fig. 1. Achsen des Accelerator-Sensors in einem Smartphone, Quelle: [14]

Um die Beschleunigungen bei unterschiedlichen Positionen des Smartphones vergleichen zu können, müssen sie auf ein bezüglich zur Erde, festes Koordinatensystem transformiert werden, siehe Fig. 2. Mit Hilfe des ebenfalls auf fast allen mobilen Geräten vorhandenen Magnetfeldsensors und des Gravitationsensors können die Acceleratorwerte auf die drei Erdkoordinaten y = magnetisch Nord (horizontal zur Erdoberfläche), z = senkrecht zum Himmel und x , dem orthogonalen Vektorprodukt dieser beiden Achsen, also ungefähr Osten (horizontal zur Erdoberfläche), transformiert werden.

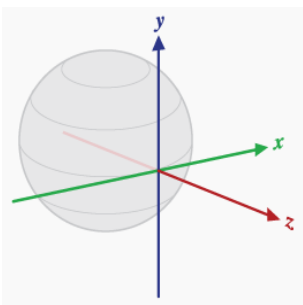


Fig. 2. Ziel-Erdkoordinaten, Quelle: [14]

Zur vorgestellten Verkehrsmoduserkennung sind keine weiteren Informationen zu dem Besitzer des Smartphones bekannt, d.h. keine Information über eine gerade verwendete App, wie z.B. Fahrradrouting, kein Straßennetz und auch keine Informationen darüber, ob die Person ein Auto besitzt oder nicht. Ziel ist die unterschiedlichen Verkehrsmodi ohne Zuhilfenahme von Kartenmaterial oder ÖPNV-Haltestellen-Informationen zu bestimmen, um

regionsunabhängig zu sein. Es gibt andere Ansätze, wie z.B. Stenneth et.al. [3] mit Verwendung von zusätzlichen GIS-Informationen oder Gonzalez et.al. [1] unter Verwendung von Neuronalen-Netzen. Fuzzy-Methoden wurden in der Arbeiten von Malazi et.al. [4] vorgeschlagen und Rasmussen et.al [5] kombiniert die Fuzzy-Logik mit GIS-Informationen. Upadhyay et.al. [6] verwendet zwar Fuzzy-Logik, realisiert jedoch keine multimodale Erkennung.

In dieser Arbeit sollen folgende Verkehrsmodi ermittelt werden können:

- WALK
- BICYCLE
- CAR
- TRAIN
- BUS

wobei unter WALK auch Laufen und Stehen/Verharren zusammengefasst sind, sowie unter TRAIN die Verkehrsmittel Tram, S-Bahn, U-Bahn, Regional- und Fernzug zusammengefasst sind. Kann der momentane Verkehrsmodus eines Smartphonebesitzers festgestellt werden, so kann das Gerät diesen nutzen und z.B. Informationen über Störungen im ÖPNV liefern, vor Staus warnen, Wetterhinweise liefern. Als weiteren Nutzen kann das aufnehmende Gerät geeignete Positionsdaten anonymisiert an Dritte zur weiteren Verarbeitung weiterleiten, z.B. an Verkehrsinfrastrukturanlagen: Ampeln empfangen die Daten und reagieren darauf. Erklären sich die Benutzer darüberhinaus bereit, die Daten auch an einen zentralen Server zu übertragen, dann können daraus auch Informationen für einen weiteren Nutzerkreis geschaffen werden, indem die Daten prozessiert und fusioniert werden und als Ergebnis z.B. aktuelle Verkehrslagen, Informationen über Stauverlaufstendenzen oder einen verbesserten Routingservice für den MIV (motorisierter Individualverkehr) liefern.

Die technisch zu lösenden Herausforderungen der multimodalen Verkehrsmoduserkennung sind vor allem

- die nicht eindeutige Klassenzugehörigkeit,
- Übergänge beim Wechsel von Verkehrsmitteln (Verkehrsmodusänderungen),
- nicht vollständige Daten, Lücken im Datensatz
- unterschiedlich skalierte Acceleratoren in den Smartphones

Hierzu wurden verschiedene Methoden untersucht, angepasst und kombiniert, um die Problem zu lösen.

3. ANGEWANDTE MATERIALIEN UND METHODEN

Die notwendigen Bewegungsdaten werden durch die Sensoren eines Smartphones erfasst. Dieses sind Positionsdaten durch GPS-Ortung und Beschleunigungs-Sensordaten, sogenannte Acceleratordaten. Da die GPS-Signale oft nicht zur Verfügung stehen oder sehr ungenau sind, werden letztere zusätzlich benötigt. Sie liefern zusätzliche Informationen über die Bewegung, wie z.B. die vertikale Beschleunigung oder das allgemeine "Gewackel" des Aufnahmegerätes während der Bewegung. Ursache für fehlende GPS-Positionsdaten können unterschiedlicher Natur sein:

- GPS-Signalempfang nicht möglich (Abschottung durch Häuserschluchten) oder sehr ungenau

- Bewegung findet in einem Gebäude oder unterirdisch statt (U-Bahn)

Die durch das Smartphone erfassten Daten werden zu einem Server gesendet. Auf dem Server sind die Daten in einer Datenbank gespeichert, worauf die Prozessierung, im Weiteren *ModeDetection* genannt, zugreifen kann. Zusätzlich kann die *ModeDetection* auch lokal auf Android-Geräten arbeiten und die Verkehrsmodi auf dem Gerät ermitteln und anzeigen. Die während dieser Arbeit aufgenommenen Daten stammen alle von Android-Smartphones, allerdings von unterschiedlichen Geräteherstellern und mit unterschiedlichen Android-Versionen (2.3 bis 4.1).

Als Basis für die multimodale *ModeDetection* dienen die Positionsdaten und die damit verbundenen Geschwindigkeiten. Eine Verkehrsmoduserkennung ausschließlich auf Basis der Acceleratordaten wurde nicht durchgeführt. Aus den Positionsdaten eines Smartphones wird zunächst ein Geschwindigkeitsprofil einer Bewegung erstellt (Beispiel eines Verkehrsmoduswechsel, siehe Fig. 4). Dabei werden die Daten in nicht zwingend äquidistante Intervalle zerlegt (I_1 , bis I_n). Dazu wird der Geschwindigkeitsverlauf auf lokale Minima und Maxima untersucht. Ein Intervall I_j beinhaltet den Verlauf von einem lokalen Minimum bis zum Nächsten. Diese Intervalle werden im Folgenden Beschleunigungsintervalle genannt.

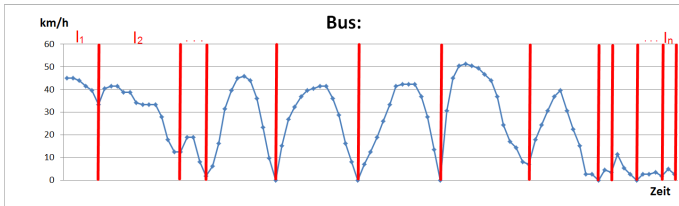


Fig. 3. Beschleunigungsintervalle einer Busbewegung

Um eine multimodale Bewegung zu erkennen, sind ganz besonders die Punkte interessant, an denen ein Wechsel des Verkehrsmittel stattfinden könnte. Das kann in der Regel nur dann passieren, wenn die Geschwindigkeit Null oder zumindest sehr niedrig ist. Natürlich muss an diesen Stellen nicht notwendigerweise ein Wechsel des Verkehrsmittels vorliegen: Ein Autofahrer kann vor der roten Ampel warten oder die S-Bahn hält an einer Haltestelle. Um einen Wechsel des Verkehrsmodus zu erkennen, müssen also noch weitere Kenngrößen ermittelt werden, z.B.:

- Standzeiten
- Richtungswechsel
- Anzahl der Halte
- Länge der oben beschriebenen Beschleunigungsintervalle

In den folgenden Diagrammen werden die zeitgleich gemessenen Geschwindigkeits- und Acceleratorwerte eines Bewegungsausschnitts gegenübergestellt (siehe Fig. 4 und Fig. 5)

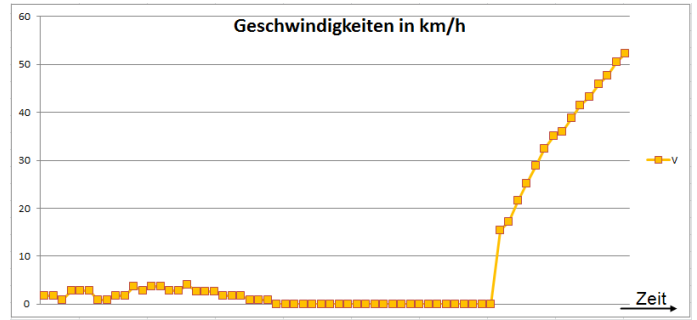


Fig. 4. Geschwindigkeitsprofil (km/h) eines Wechsel von WALK auf TRAIN

Dazu die zeitgleich gemessenen Acceleratorwerte im m/s^2 :

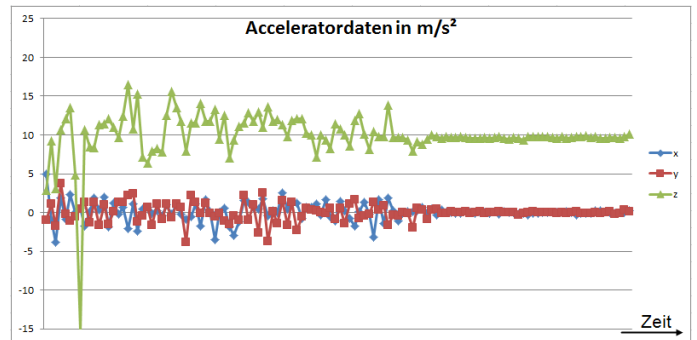


Fig. 5. Acceleratorprofil eines Wechsel von WALK auf TRAIN

Die niedrigen Geschwindigkeiten zu Beginn der Bewegung stellen das Gehen zur Bahnstation dar, dann sitzt die Person im Zug (Geschwindigkeit = 0) und wenige Sekunden später fährt der Zug los, so dass die Geschwindigkeit deutlich ansteigt. Anders die Acceleratordaten: Sie schwingen beim Gehen und Sitzplatzsuchen in der Bahn deutlich, während sie bei der Abfahrt des Zuges deutlich geringer sind und weniger schwanken.

3.1 Vorverarbeitung von Bewegungskenngrößen

Als Eingangsgrößen für die Fuzzy-Regelung werden die Daten vorverarbeitet. Das ist unverzichtbar, da sowohl die GPS-Daten, z.B. in Folge von schlechtem Empfang, als auch die Acceleratordaten eine sehr hohe Varianz haben können. Es wird angestrebt, die Bewegungsdaten und bei der Merkmalgewinnung bzw. der Ableitung der Bewegungskenngrößen eine gleichzeitige Filterung zu realisieren. So kann z.B. später eine Datenanalyse auch auf einem mobilen Gerät schneller durchgeführt werden. Es bietet sich der Median, der in der Signal- und Bildverarbeitung ein oft angewandter, nichtlinearer Rangordnungsfilter ist, an, um 'Salz-und-Pfeffer' Rauschen zu eliminieren [11, 12]. So hat der Median den Vorteil, im Gegensatz zum Mittelwert, dass er die großen Ausreißer in seiner Umgebung unterdrückt (siehe Gl. (1)). Das heißt, dass der Median die Varianz der Messwerte minimiert:

$$\sum_{i=0}^n (x_i - \tilde{x})^2 \leq \sum_{i=0}^n (x_i - \bar{x})^2, \quad (1)$$

wobei durch \tilde{x} der Median und durch \bar{x} der Mittelwert gekennzeichnet ist.

Die relevanten Bewegungskenngößen, die als Input für die Fuzzy-Regelung genutzt werden, werden hauptsächlich anhand ihrer statistischen Eigenschaften beschrieben. So wird aus den Messdatensatz eines Beschleunigungsintervalles ein Geschwindigkeitshistogramm berechnet. Für die Accelerator-Sensordaten wurden entsprechend die Histogramme für x,y- und z-Beschleunigungen gebildet. Daraus können die gängigen statistischen Kenngrößen wie Mittelwert, Varianz, Median und die α -Perzentile/Quartile der Geschwindigkeits-/Beschleunigungshäufigkeiten aus dem Histogramm $h(x)$ bestimmt werden. Der Mittelwert und die Varianz wurden aus $h(x)$ folgendermaßen bestimmt:

$$\bar{x} = \sum_{i=0}^n x_i \cdot h(x_i), \quad (2)$$

$$s^2 = \sum_{i=0}^n (x_i - \bar{x})^2 \cdot h(x_i), \quad (3)$$

Der Medianwert wird aus $h(x)$ wie folgt berechnet:

$$\tilde{x} = \sum_{i=0}^{n_j} h(i) \leq 1/2, \quad (4)$$

und die α -Perzentile sind

$$P_\alpha = \sum_{i=0}^{n_\alpha} h(i) \leq \alpha, \quad (5)$$

mit $0 < \alpha < 100$. Die Perzentile P_{05} , P_{95} werden für die Berechnung der lokalen Minimum- und Maximum-Werte des Beschleunigungsintervalles verwendet. Datenabschnitte mit einer Null-Geschwindigkeit sind dementsprechend die, die unter P_{05} liegen.

Die Histogrammberechnung der physikalischen Kenngrößen kann dabei durch das folgenden, einfachen C/Java-Codebeispiel realisiert werden:

```
// Histogramm
for ( int i = 0 ; i < n; i++ )
    h[data_array[i]]++;

// Normierung des Histogrammes
for ( int i = 0 ; i < max_h; i++ )
    h[i] /= (float) max_h;
```

Die Implementierung des Medians kann mit Hilfe des obigen Histogrammbeispiels realisiert werden oder, wo die Datenstruktur als verkettete Liste vorliegt, mit der Collection-Sortierfunktion von Java gelöst und entsprechend der folgenden Formel berechnet werden:

$$\tilde{x} = \begin{cases} \frac{1}{2}(h(x_{n_k}) + h(x_{n_k+1})) & \text{falls } n \text{ gerade ist} \\ h(x_{n_k}) & \text{sonst} \end{cases} \quad (6)$$

3.2 Erkennung des Verkehrsmodus

Zunächst wird für jedes wie oben beschriebene Beschleunigungsintervall I_j ein Verkehrsmodus rein anhand der Geschwindigkeit geschätzt. Dazu werden der Median

der Geschwindigkeit und die maximale Geschwindigkeit herangezogen. Durch den Median der Geschwindigkeit werden Ausreißer im Datensatz unterdrückt. Allein an diesen Kenngrößen lassen sich jedoch noch nicht ähnlich schnelle Fortbewegungsmittel unterscheiden. Beispiele dafür sind:

- Bus und Auto im Berufsverkehr
- Bahn und Auto auf einer Schnellstraße
- Tram und Bus
- Tram und schneller Radfahrer

Aus den Positionsdaten werden weitere Kenngrößen berechnet, wobei auch die zusätzlich aufgezeichneten Accelerator-daten herangezogen werden:

- Stillstandzeiten
- Startbeschleunigung nach Stillstand,
- Durchschnittliche Beschleunigungszeiten
- Medianwerte der horizontalen Beschleunigung aus den Acceleratordaten
- Medianwerte aus der vertikalen Beschleunigung aus den Acceleratordaten
- Medianwerte der "Amplituden" aus den Acceleratordaten (jeweils für horizontale und vertikale Beschleunigung)
- Varianzen zu den sogenannten Accelerator-"Amplituden"

Die Auswertung der Acceleratordaten gestaltete sich als problematischer als erwartet, da Smartphones unterschiedlicher Hersteller in gleichen Situationen unterschiedlich große Beschleunigungen messen. Zwar ist die Tendenz gleich, die absoluten Werte unterscheiden sich aber so sehr, dass typische Wertebereiche eines Gerätes zu einem Bewegungsmodus in den typischen Bereich eines anderen Gerätes zu einem anderen Verkehrsmodus fallen. Deshalb können nur signifikante Unterschiede zwischen den einzelnen Modi verglichen werden, was aber auch schon dazu beiträgt, einen Radfahrer von einem langsamen Auto unterscheiden zu können.

An dieser Stelle kommt die Fuzzifizierung zum Einsatz. Diesmal wird die *ModeDetection* nicht für jedes einzelne der Beschleunigungsintervalle I_j durchgeführt, sondern für Bewegungsabschnitte zwischen zwei Punkten mit Geschwindigkeiten gleich oder annähernd Null, also zwischen möglichen Verkehrsmoduswechseln. Das Konzept von Fuzzy-Mengen mit zugehörigen Algorithmen wurde erstmals von L. Zadeh [7, 8] vorgestellt. Zadeh hat eine mathematisch präzise Theorie der sogenannten linguistischen Ausdrücke/Terme geschaffen, die im täglichen Sprachgebrauch nicht exakte, nicht eindeutige, bzw. unscharfe Bedeutungen haben, wie "hoch", "mittel", "gering" etc.. Ähnliche linguistische Befehle werden auch bei der Steuerung eines Systems (z.B. bei Schifffahrt) mündlich gegeben [10]. Diese unscharfe Problematik trifft praktisch auch auf verschiedene technische Regelungs- und Klassifikationssystemen zu. Im Gegensatz zu Neuronalen-Netzwerk-Lösungen für die Verkehrsmoduserkennung [1] erlauben es Fuzzy-Techniken, ohne langwierige Trainingsphase physikalische, schon bekannte Kenngrößen und Erfahrungswerte direkt in die Zugehörigkeitsfunktionen und in die Wissens-/Regelbasis des Fuzzy-Systems einbauen zu können. Auch bei statistischen Bayes-Methoden könnte man bei der Ermittlung von a-priori- und bedingten Wahrscheinlichkeiten

Schwierigkeiten haben, da die Sensordaten (GPS, Accelerator) nicht unbedingt einer Gaußverteilung gehorchen. Aus den obigen Überlegungen heraus wird die Fuzzy-Regelmethode präferiert. Weitere formale Beschreibungen zu Fuzzy-Techniken findet man unter anderem in [9, 10].

Konkret heißt das, es werden keine scharfen Wenn-Dann-Regel aufgestellt, wie z.B.

Wenn $\text{MaxGeschwindigkeit} > 50$, dann $\text{Verkehrsmodus} = \text{Auto}$,

sondern sogenannte weiche Regeln. Weiche Regeln bedeuten, dass der Wenn-Teil nicht zu 100 Prozent erfüllt sein muss, sondern z.B. nur zu 70 Prozent. Beispiel: Wenn die mittlere Geschwindigkeit 28 km/h ist, kann das zu 70 Prozent eine Tram sein und zu 20 Prozent ein Auto. Mit anderen Worten: Ein Verkehrsmodus kann mehr oder weniger wahrscheinlich sein. Die Zuordnung der Wahrscheinlichkeiten geschieht über sogenannte Zugehörigkeitsfunktionen. Sie liefern Werte aus dem abgeschlossenen Intervall [0,1].

Der Wenn-Teil kann aus mehreren logischen Verknüpfungen (Und-, Oder- oder Nicht-Verknüpfungen) zusammengesetzt sein, z.B.:

Wenn $\text{MittlereGeschwindigkeit} = \text{fahrradtypisch}$
und $\text{HorizontaleAcceleratorAmplitude} = \text{groß}$,
dann $\text{Verkehrsmodus} = \text{BICYCLE}$

Für die vorliegende *ModeDetection* haben wir ein Regelwerk von sogenannten linguistischen Variablen aufgestellt.

Es folgt ein Beispiel für die linguistische Variable *Mittlere Geschwindigkeit*: Anhand von Messdaten wurde die Verteilung der mittleren Geschwindigkeiten der einzelnen Verkehrsmittel festgestellt. Als Zugehörigkeitsfunktionen wurden Trapezoide gewählt, deren Maximum (Wert = 1) die Spanne der am häufigsten festgestellten mittleren Geschwindigkeiten umfasst plus Bereiche davor und dahinter, die weniger häufig gemessen wurden aber vorkommen. Die linguistische Variable *Mittlere Geschwindigkeit* mit ihren Zugehörigkeitsfunktionen wird wie skizziert definiert:

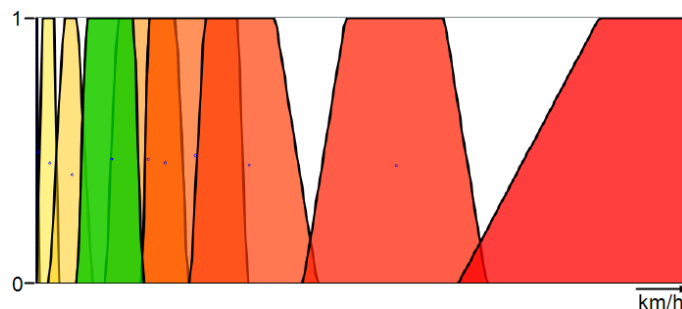


Fig. 6. Skizze trapezoide Zugehörigkeitsfunktionen von mittleren Geschwindigkeiten

Die einzelnen Trapezoide stellen die Zugehörigkeitsfunktionen

- walk: Trapezoid (1.000, 2.000, 7.000, 8.000)
- bicycle: Trapezoid (5.000, 10.000, 15.000, 20.000)
- tram: Trapezoid (15.000, 18.000, 35.000, 38.000)
- bus: Trapezoid (25.000, 29.000, 50.000, 53.000)
- city_car: Trapezoid (38.000, 40.000, 50.000, 55.000)

- fast_city_train: Trapezoid (38.000, 40.000, 72.000, 75.000)
- car_country: Trapezoid (55.000, 60.000, 85.000, 100.000)
- car_motorway: Trapezoid (95.000, 110.000, 145.000, 160.000)
- very_fast: Trapezoid (150.000, 200.000, 500.000, 500.000)

dar.

Es ist zu erkennen, dass es Überschneidungen gibt und weitere linguistische Variablen notwendig sind, um einen Verkehrsmodus erkennen zu können. Eine mittlere Geschwindigkeit von 45 km/h bedeuten sowohl mit einer Wahrscheinlichkeit von 100 Prozent bus, city_car als auch fast_city_train.

Als eine weitere markante Kenngröße hat sich die sogenannte durchschnittliche Beschleunigungsintervallzeit, gemessen in Sekunden, herauskristallisiert. Damit ist die Zeit gemeint, die vom Beginn einer Beschleunigung über Halten der Geschwindigkeit bis zum Abbremsen und erneutem Beginn einer neuen Beschleunigung vergeht. Für sie wurden einige markante Bereiche als *short*, *moderate* und *long* klassifiziert:

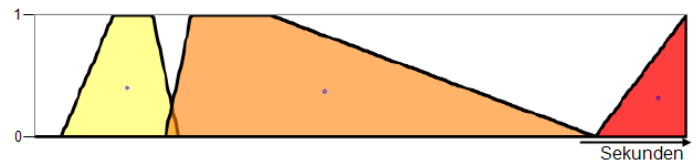


Fig. 7. Skizze Zugehörigkeitsfunktionen Beschleunigungsintervallzeit

Alle oben genannten Kenngrößen wurden als linguistische Variablen definiert und als Und-, Oder- oder Not-Bedingungen zu linguistischen Regeln (Interferenzen) verknüpft. Beispiele hierfür sind:

- if MeanVelocity is tram_like and MaxVelocity is tram_like and AvgAccIntervallDuration is not short then TrafficMode is TRAIN
- if MeanVelocity is tram_like and MaxVelocity is tram_like and AvgAccIntervallDuration is short then TrafficMode is CAR
- if MeanVelocity is tram_like and MaxVelocity is bus_like and StartAcc is tram_like_acc then TrafficMode is TRAIN

Die Defuzzifizierung geschieht durch Aufsummierung der Wahrscheinlichkeiten je Verkehrsmodus (Fuzzy-Output-Variablen) und Ermittlung des Maximums. Das Ergebnis ist ein Verkehrsmodus aus der Menge {UNKNOWN, WALK, BICYCLE, BUS, CAR, TRAIN}.

Hierbei wurde die Fuzzy-Regelung also für die Erkennung der Bewegungsmodus verwendet (siehe Fig. 8). Die bekannten physikalischen Kenngrößen der Bewegungen wie Geschwindigkeit, Beschleunigung, usw. wurden für die Erkennung der verschiedenen Bewegungsmodi als Input-Variablen für die Fuzzifizierung verwendet. Die Festlegung der Fuzzy-Regeln bzw. die Bestimmung der Interferenzen, also die Übertragung der fuzzyfizierten Kenngrößen auf die entsprechenden Verkehrsmodi, die auch als Regelbasis betrachtet werden kann, basiert auf empirischen, physikalischen Erfahrungen. Die Defuzzifizierung, die die

Zuordnung der Ausgänge auf die Verkehrsmodi-Menge abgebildet, liefert als Output eine Klassifikation, die in diesem Fall ohne weitere Lernmechanismen auskommt.

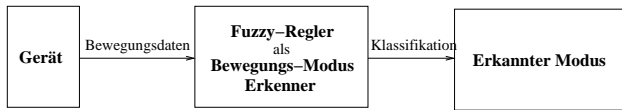


Fig. 8. Die sematische Darstellung der Erkennung von Bewegungsmodi

Für die Fuzzyfizierung wird die OpenSource-Software Fuzzylite [Juan Rada-Vilela. fuzzylite: a fuzzy logic control library, 2014. URL <http://www.fuzzylite.com>] eingesetzt. Sie hat den Vorteil relativ schlank zu sein und kann auch auf Android-Smartphones eingesetzt werden.

Als dritter, abschließender Schritt werden einige weitere logische Korrekturen an den ermittelten Verkehrsmodi einer Bewegungssequenz vorgenommen. Dies sind Zusammenhänge, die sich nicht als Fuzzy-Regeln darstellen lassen:

- Korrektur der Bewegungsmodi WALK oder BICYCLE, wenn auf TRAIN folgend und kein Richtungswechsel stattfindet.
Es wird angenommen, dass die Person in einem haltenden oder langsam fahrenden Zug sitzt.
- Ein WALK-Abschnitt von weniger als 30 Sekunden zwischen zwei gleichen, schnellen Verkehrsmodus (CAR, TRAIN oder BUS) wird überschrieben mit diesem, weil die Fahrt wahrscheinlich nicht unterbrochen wurde.
- Bei direkt aufeinanderfolgenden verschiedenen, schnellen Verkehrsmodi (TRAIN, CAR, BUS), ohne dass diese durch eine WALK-Sequenz getrennt sind, wird ermittelt, welcher Modus am häufigsten durch die Fuzzylogik festgestellt wurde. Ist das Ergebnis eindeutig, dann wird für alle diese so zusammenhängenden Sequenzen der häufigste Modus angenommen.

4. KURZE DARSTELLUNG DER ERGEBNISSE

In der vorliegenden Arbeit wurde eine automatische multimodale Verkehrsmoduserkennung realisiert. Als Problem wurde die Bus ↔ Car Trennung im Stadtverkehr festgestellt. Hier sind zumeist Zusatzinformationen wie ÖPNV-Haltestellenposition nötig.

In der folgenden Tabelle sind Prozentangaben zu den Anteilen der korrekt ermittelten Verkehrsmodi angegeben. Das bei einer multimodalen Bewegungssequenz der Wechsel eines Verkehrsmodus nicht erkannt wurde, kam nicht vor, stattdessen wurden zuviele Wechsel erkannt oder der neue Verkehrsmodus falsch erkannt. Deshalb sind die Fehlerquoten auf die einzelnen Verkehrsmodi aufgeschlüsselt und nicht auf eine gesamte multimodale Route. Auch wären diese schwer zu vergleichen.

Bei allen Testbewegungen wurde das aufnehmende Smartphone nicht zusätzlich durch Nutzung dieses bewegt. Es wurde möglichst so positioniert, dass es die Bewegung unter realen Bedingungen aufnehmen kann: Z.B. im Auto auf dem Beifahrersitz in einer Tasche, in der Bahn in der Jackentasche und/oder Handtasche, beim Radfahren

oder Gehen in der Jackentasche oder in einer zusätzlichen Tasche. Tests mit gleichzeitiger, aktiver Nutzung des Smartphones durch den Benutzer während der ModeDetection stehen noch aus.

Die Auswertung der Testaufnahmen wurden erst im Nachhinein durchgeführt. Die Daten wurden in ein Testmodul eingelesen, die Verkehrsmodi bestimmt und dann die Erfolgsquote ermittelt. Bei einer Auswertung auf Richtigkeit des jeweils aktuellen Verkehrsmodus während der Aufnahme, werden die Ergebnisse schlechter, da die *ModeDetection* besser wird, je mehr Informationen sie hat, sprich je länger die aufgezeichnete Strecke ist.

Um schon online im Verkehr eine Information über den ermittelten Verkehrsmodus zu haben, wurde eine Android-App erstellt, die mittels Open-Street-Map-Karten die aktuelle Position anzeigt und auf Wunsch die ermittelten Verkehrsmodi der zurückgelegten Strecke anzeigt.

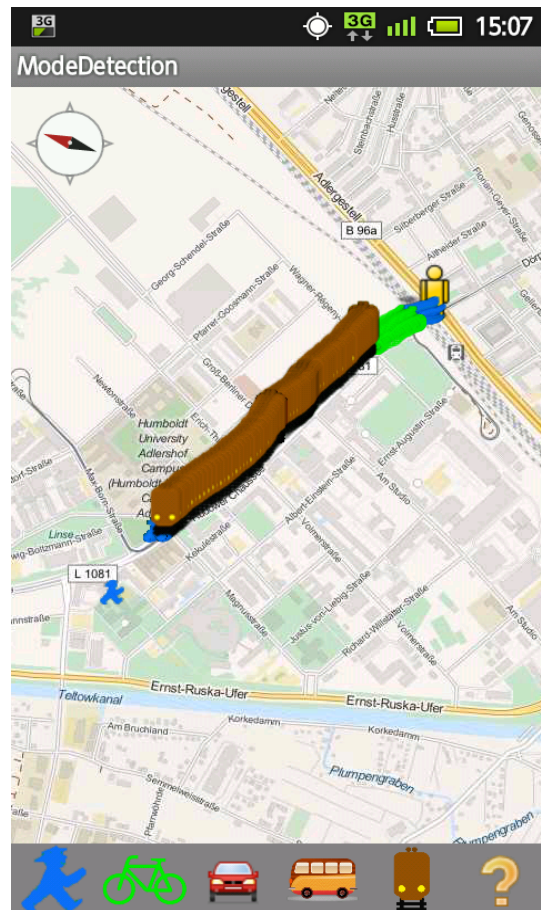


Fig. 9. App zum Anzeigen der ermittelten Verkehrsmodi

Fig. 9 zeigt ein Beispiel für eine WALK-TRAIN-WALK-Bewegung, wobei TRAIN eine Trambahn ist und die WALK-Strecken davor und dahinter sehr kurz. Die letzten Meter der Tram-Bewegung wurden nicht als TRAIN erkannt, sondern als BICYCLE. In diesem Fall lag es an der langsamen Geschwindigkeit der Tram und dem erhöhten "Gewackel" der Acceleratordaten durch das Aufstehen im Zug.

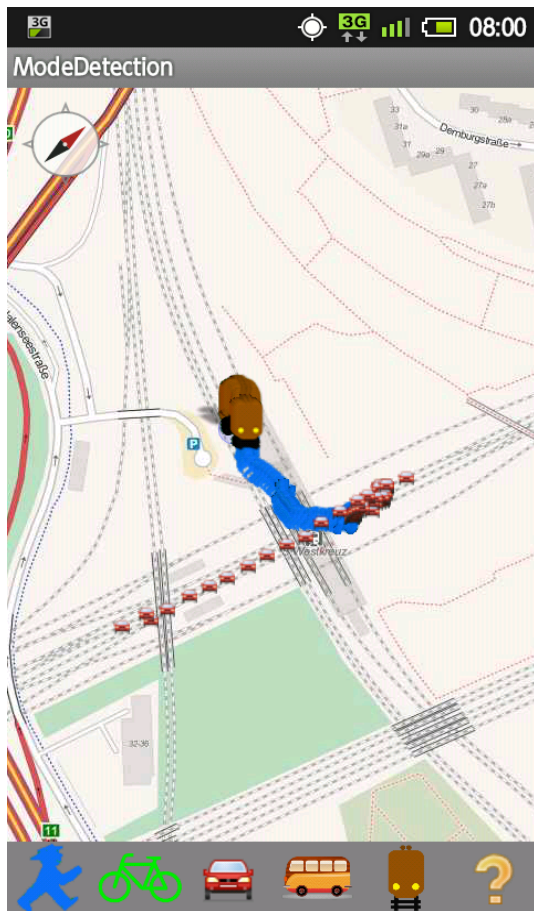


Fig. 10. Ermittelte Verkehrsmodi beim Umsteigen

Fig. 10 zeigt eine Umsteigesituation mit Bahnsteigwechsel an. Fälschlicherweise wurde das erste S-Bahn-Stück als Autofahrt erkannt. Der blau dargestellte Fußweg und die anschließende Weiterfahrt mit der S-Bahn wurden jedoch wieder korrekt erkannt.

Wie aus Tabelle 1 erkennbar ist, gibt es noch größere Probleme bei der Unterscheidung zwischen einem Auto im Stadtverkehr und Busfahrten.

5. KURZES FAZIT UND PRAKTISCHE IMPLIKATIONEN

Durch die Anwendung der vorgestellten Methoden ist die multimodale Verkehrsmoduserkennung möglich. Im Gegensatz zu neuronalen Netzwerken können die Daten ohne eine Trainings-Phase zu den verschiedenen Modi zugeordnet werden (siehe Tabelle 1). Die Analyse der Verkehrs-Modi ermöglicht eine automatisierte, bewegungsspezifische Applikationsanpassung. Anhand der festgestellten Verkehrsmodi können die Bewegungsdaten für unterschiedliche Datenverarbeitungszwecke genutzt werden. Davon kann der Einzelne oder aber auch einer größeren Zielgruppe im Verkehr profitieren.

Beispiele:

- Modus = Car: Stauwarnungen, Weiterleitung der Daten zur Ermittlung der Verkehrslage
- Modus = Train: Anzeige von aktuellen Verspätungsinformationen

Tab. 1. Die Ergebnisse der Auswertungen

Auswertung Fehlerraten der detektierten Verkehrsmodi		
Modus	Erkannt	Bemerkungen
WALK	94%	Es wurden nur Fußstrecken zw. 5 und 10 Minuten gemessen
BICYCLE	90%	Auch Hügel und Schnee dabei
CAR	85% 96% 100%	Stadtverkehr Landstraße Autobahn (nur 3 Fahrten)
BUS	34%	Wurde meist als CAR erkannt
TRAIN	99% 87% 95%	Tram Berlin + Potsdam S-Bahn Berlin U-Bahn Berlin* Fernbahn: Noch keine Tests

*U-Bahn wurde nur getestet mit vorher (oberirdisch) aktiviertem GPS, so dass die eigentliche U-Bahnfahrt daran erkannt werden kann, dass kein GPS mehr vorliegt, wohl aber Acceleratordaten, nach einiger Zeit aber wieder GPS-Positionen gemeldet werden und dass eine gewisse Distanz in einer gewissen Geschwindigkeit zurückgelegt wurde. Dabei wird berücksichtigt, dass es unterirdische Wartezeiten gibt (Fußweg zum Gleis, Warten auf die Bahn, Umsteigezeiten). Auch Autos, die durch einen längeren Tunnel fahren, verlieren das GPS-Signal, legen die Entfernung aber deutlich schneller zurück. Ein Stau im Straßentunnel kann aber dazu führen, dass fälschlicherweise eine U-Bahnfahrt angenommen wird.

- Modus = Walk: Anzeige Wetterinformationen
- Bei einigen Verkehrsmodi macht es Sinn, die anonymisierten Positions- und Geschwindigkeitsdaten an die Verkehrsinfrastruktur weiterzuleiten: Ampeln wissen dann vorher schon wieviel Fahrzeuge, Radfahrer oder auch Fußgänger auf sie zufahren und können gegebenenfalls ihre Steuerung anpassen.

Zukünftig soll die gesamte Datenanalyse auf einem mobilen Datenerfassungsgerät/Smartphone performanter gemacht werden und einer größeren Menge von Nutzern zur Verfügung gestellt werden.

Eine Verbesserungsmöglichkeit des Systems und der vorgestellten Methodik, für eine bestmögliche Erkennung der Bewegungsmodi, könnte durch eine optimierte Fuzzifizierung der Input-Größen und auch durch eine optimierte Fuzzy-Regelung durch z.B. Neuro-Fuzzy-Systemen erreicht werden. Dadurch würde das System lernfähig. Hierdurch wäre es möglich, das System individuell auf eigene Bewegungs- und Fahrstile, bzw. weitere physikalische Eigenschaften des Fahrzeuges und/oder der Fahrgäste einzustellen und zu personalisieren.

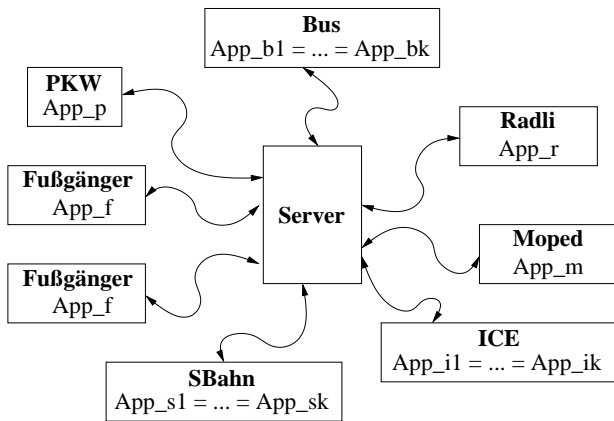


Fig. 11. Client Server Verarbeitung der Bewegungsinformationen

Fig. 11 stellt eine mögliche Anwendung dar. Die erfassten Bewegungsdaten werden zu einem Server gesendet, wo aus den Daten die Bewegungskenngrößen berechnet werden. Durch das Fuzzy-Regel-System werden die Modi bestimmt bzw. erkannt. Dann sollen z.B. allen, in einem Bus fahrenden Fahrgästen, die gleichen Modi zugeordnet werden. Die Informationen können dann für weitere für das Verkehrsmanagement relevante Operationen angewandt werden oder entsprechend des Verkehrsmodus Applikationen und/oder Verkehrsinformationen zu den Fahrgästen zurückgesandt werden.

ACKNOWLEDGEMENTS

Wir bedanken uns bei den Kollegen des Instituts für Verkehrssystemtechnik in Berlin für das Aufnehmen von Testdaten und für die hilfreichen Korrekturvorschläge für dieses Paper.

REFERENCES

- [1] P.A. Gonzalez, J.S. Weinstein, S.J. Barbeau, M.A. Labrador, Ph.L. Winters, N. Labib Georggi, R. Perez: *Automating Mode Detection using Neural Networks and Assisted GPS Data Collected using GPS-Enabled Mobile Phones*.
- [2] Tao Feng, Harry J.P. Timmermans *Transportation mode recognition using GPS and accelerometer data*, Transportation Research Part C 37 (2013), pp. 118-130, 2013.
- [3] Leon Stenneth, Philip Yu, Ouri Wolfson, Bo Xu: *Transportation Mode Detection using Mobile Devices and GIS Information*, GIS '11 Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, November 14, 2011. Chicago, IL, USA. pp. 54-63, ISBN: 978-1-4503-1031-4, 2011
- [4] HadiTabatabaee Malazi, Kamran Zamanifar and Stefan Dulman: *FED: FUZZY EVENT DETECTION MODEL FOR WIRELESS SENSOR NETWORKS*. International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 6, December 2011.
- [5] Thomas Kjær Rasmussen, Jesper Blåfoss Ingvarsson, Katrín Halldórsdóttir, Otto Anker Nielsen: *Using wearable GPS devices in travel surveys: A case study in the Greater Copenhagen Area*. (Proceedings from the

Annual Transport Conference at Aalborg University) ISSN 1603-9696. 2013.

- [6] D. Upadhyay, N. Schüssler, K.W. Axhausen, M. Flamm, V. Kaufmann: *Optimal parameter values for mode detection in GPS post-processing: An experiment*. Arbeitsbericht Verkehrs- und Raumplanung 507. July 2008.
- [7] L.A. Zadeh: *Fuzzy Sets*. Inform. and Control. Vol. 8, pp.338–353, 1965.
- [8] L.A. Zadeh: *Fuzzy Algorithms*. Inform. and Control. Vol. 11, pp.323–339, 1969.
- [9] Detlef Nauck, Frank Klawonn und Rudolf Kruse: *Neuronale Netze und Fuzzy-Systeme: Grundlagen des Konnektionismus, Neuronaler Fuzzy-Systeme und der Kopplung mit wissensbasierten Methoden*. Vieweg Verlag, ISBN 3-528-05265-1, 1994.
- [10] Rudolf Kruse, Jörg Gebhardt und Frank Klawonn: *Fuzzy-Systeme*. BG. Teubner Stuttgart, 1995.
- [11] Heinrich Niemann: *Pattern Analysis and Understanding*, Springer-Verlag, ISBN 3-540-51378-7, 1990.
- [12] Oppenheim, Alan V. Schafer, Ronald W. with John R. Buck: *Discrete-Time Signal Processing*, second edition, Englewood Cliffs, N.J., Prentice-Hall, ISBN 0-13-083443-2, 1999.
- [13] Juan Rada-Vilela: *fuzzylite: a fuzzy logic control library*. <http://www.fuzzylite.com>, 2014
- [14] Google: *Google-Android-Developer-Site*, <http://developer.android.com/reference/android/hardware/SensorEvent.html>