

Developing a passive GPS tracking system to study long-term travel behavior

Working Paper**Author(s):**

[Marra, Alessio Daniele](#) ; [Becker, Henrik](#); [Axhausen, Kay W.](#) ; [Corman, Francesco](#) 

Publication date:

2018-10

Permanent link:

<https://doi.org/10.3929/ethz-b-000300458>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

Arbeitsberichte Verkehrs- und Raumplanung 1393

Developing a passive GPS tracking system to study long-term travel behavior

Alessio D. Marra^{1,2} Henrik Becker¹ Kay W. Axhausen¹
Francesco Corman¹

¹Institute for Transport Planning and Systems, ETH Zürich

²e-mail: alessio.marra@ivt.baug.ethz.ch

Abstract

This paper describes development and testing of a passive GPS tracking smartphone application and corresponding data analysis methodology designed to increase the quality of travel behavior information collected in long-term travel surveys. The new approach is intended to replace the pencil-and-paper travel diaries and prompted recall methods that require more user involvement due to requirements for manual data entry and/or high battery usage. Reducing the burden placed on users enables researchers to collect data over longer periods, thus improving the quality of travel behavior research. To reduce battery use the smartphone-based application collects GPS data less frequently than other methods. Therefore, new algorithms were developed to identify trips and activities, transport mode, and even the specific vehicle used by the traveler. An important finding was the significant advantage of using users past data to improve mode detection results. The system was tested successfully in Zürich and Basel (Switzerland).

Keywords

Tracking; travel survey; public transport operations; mode detection; smartphone; GPS

1 Introduction

Until now travel diaries have been the primary source of travel behavior information on activity chains, trip patterns, mode choice and time use (Schlich and Axhausen (2003)). Unfortunately, conducting and analyzing travel diary studies is complicated and expensive. More importantly, since completing travel diaries places a significant burden on respondents, most surveys only ask respondents to report on travel during one randomly chosen day (Greene et al. (2016)) although research clearly shows day-to-day variation in travel behavior (Pas and Sundar (1995), Axhausen et al. (2000)), which limits the political and operational value of one-day data (Susilo and Axhausen (2007)).

The use of GPS data, for instance provided by smartphones, can significantly improve the efficiency of travel diary studies and the quality of information collected. Collecting GPS data from smartphones places a lower burden on respondents, offers greater spatio-temporal precision and has lower implementation costs (Vij and Shankari (2015)). The main drawback of using smartphones is their reliance on energy-intensive GPS services that quickly draw-down the smartphone battery, thus reducing the desire of travelers to use them.

The goals of this research were to develop a passive GPS tracking application that consumes very little battery power and a set of algorithms that can use this GPS data to provide de-

tailed traveler behavior data. An application meeting the low power consumption objective was developed by reducing the GPS sampling frequency. The algorithms were designed to use this low(er) quality locational data to understand all user trips over a period of several weeks. These algorithms consisted of: activity and trip identification (dividing the users' records into activities and trips); trip segmentation (grouping trips into walking or using some means of transport); and, mode detection (identifying the transport means).

The key benefits of the application and algorithms are:

- The low battery consumption and limited (essentially zero) burden on the traveler means that it is possible to collect a great deal of positional data over long time periods;
- The user's travel behavior, in terms of activities, trips and transport modes used, is derived from low-frequency GPS data;
- The mode detection algorithm can also detect the particular public transport vehicle used using public transport operations data;
- Past travel information from users are used to identify missing transfers and improve the mode detection accuracy.

In particular, identifying the specific public transport vehicle used is not possible with existing mode detection algorithms and provides helpful information for understanding user travel behavior. The application passively collected location information from users approximately every 30 seconds. No interaction was required from respondents except to install the app and complete two short questionnaires at the beginning and at the end of the study. The smartphone application and algorithms were tested in Zürich. A further dataset, collected with a different smartphone application in the city of Basel, was used to validate the proposed algorithms.

The paper is organized as follows: section 2 describes state of the art; section 3 describes the smartphone application, the survey process and the datasets; section 4 describes the data cleaning procedure; section 5 describes the trip and activity identification algorithm; section 6 describes the trip segmentation algorithm; section 7 describes the mode detection algorithm; section 8 presents the results of the data collection; section 9 describes the validation procedure; and, finally, section 10 presents conclusions.

2 State of the art

GPS tracking makes it possible to collect long-term travel diaries while placing a very low burden on respondents (Stopher et al. (2008)). Initial studies using GPS loggers for travel diary collection were promising, although they required a substantial effort to distribute the devices and to obtain additional information from respondents necessary to interpret the GPS records (Bohte and Maat (2009); Oliveira et al. (2011); Schuessler and Axhausen (2009); Montini et al. (2014)).

Today, the focus has shifted away from GPS loggers towards smartphone applications (Cottrill et al. (2013)), due to their easier administration and the development of automatic methods for detecting transport mode based on GPS data. Most of these methods are based on machine learning techniques and they often integrate GPS data with data from other smartphone sensors such as accelerometers (two related reviews are Wu et al. (2016) and Nikolic

and Bierlaire (2017)).

Most of today's smartphone-based tracking systems use a prompted recall approach. This requires respondents to manually add details such as trip purpose, mode, group size, transit fare, parking fees etc. to each trip. Although some systems use statistical learning to make suitable suggestions to reduce the burden on respondents, a substantial amount of user interaction is still required to annotate or validate trip information.

Up until to now these smartphone-based tracking methods have mostly been tested on small datasets. For example, Tsui and Shalaby (2006) is based on 60 trips; and, Stenneth et al. (2011) recorded information for three weeks of travel by six people. Research based on large datasets have used dedicated GPS devices, making data collection complicated and expensive. Examples include Zheng et al. (2010) who collected information on 65 people over a period of 10 months; and Schuessler and Axhausen (2009) who used a dataset of 4882 people (requiring multiple waves to reduce the number of devices needed).

Finally, it is important to mention Prelipcean et al. (2017), who developed MEILI, a battery-saving app to collect travel diaries from smartphones. They tested it only for one week and since the users had to manually annotate their trips, only about one third of them completed the study.

As this result shows, the problem of manually annotating trips places a significant hurdle on data collection. Therefore, a key tool for increasing the efficiency of smartphone-based travel diary data collection and analysis is automatic mode detection. Several studies have used automatic mode detection from GPS data including Schuessler and Axhausen (2009), Stopher et al. (2005), Zhu et al. (2016), Zheng et al. (2010), Zhang et al. (2011). These studies divide the problem of mode detection into the following four tasks (with small variations):

- Data cleaning
- Trip and activity identification
- Trip segmentation
- Mode detection

Also this paper follows this structure, as shown in Figure 1. Since these tasks are often considered as separate problems, a short literature review is presented for each task.

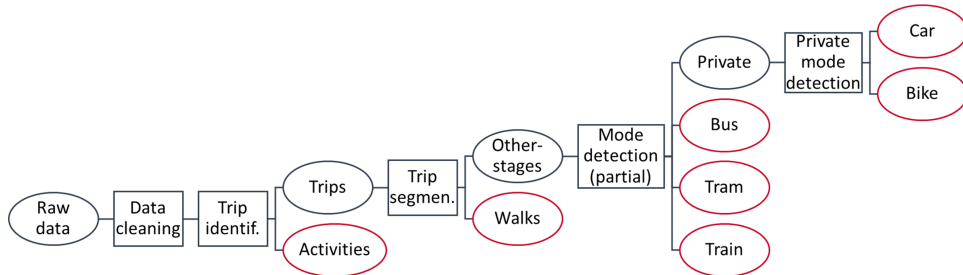
2.1 Data Cleaning

Since raw GPS data may have systematic errors, it is necessary to identify and correct GPS errors before the data can be used in the next steps (Wu et al. (2016)). The two main techniques used for data cleaning are filtering and smoothing.

Data filtering removes data that do not represent the users real position. Several methods have been used to filter data. Ogle et al. (2002) used the position dilution of precision (PDOP, an accuracy measure based on the geometry of satellites) and the number of satellites for the filtering process. Gong et al. (2012) used the horizontal dilution of precision (HDOP) and the number of satellites to discard points. Schuessler and Axhausen (2009) used altitude values and sudden jumps in position to discard points. Ansari Lari and Golroo (2015) used speed to help identify GPS points for filtering.

The smoothing process is used to reduce the random noise present in the data. Jun et al.

Figure 1: Sequence of algorithms used for mode detection. Rectangles indicate algorithms; ovals indicate data; red ovals represent the final output.



(2006) compared several smoothing techniques and found that a modified Kalman filter works best. Nitsche et al. (2014) also preprocessed the positional data using a Kalman filter. Schuessler and Axhausen (2009) used a Gauss kernel smoothing approach.

2.2 Trips and Activities Detection

The smartphone application records the user's position continuously throughout the day. Therefore, each user's data must be divided in trips and activities. An activity is shown by a sequence of points near each other, indicating that the user is in the same place for a long period. In contrast, a trip is shown by a sequence of points located apart from each other, representing the user's movement to a different place. A user's day is formed by activities alternating with trips.

Several techniques have been used to identify trips and activities. So far, to the best of our knowledge, all of them begin by first detecting activities and then defining trips. One of the most common techniques is to measure the time between two consecutive GPS points and compare it to a given threshold value. Different threshold values have been used ranging from 45 seconds (Pearson (2001)), 300 seconds (Wolf et al. (2004)) and 900 seconds (Schuessler and Axhausen (2009)).

A second technique is to define activities as periods when there are very low values of speed for a specified minimum amount of time (Tsui and Shalaby (2006)).

A third technique is the density-of-points based method used by (Stopher et al. (2005), Schuessler and Axhausen (2009)). This method identifies activities where the density of points in a certain area is greater than a specific threshold. Schuessler and Axhausen (2009) defined a value of density for each GPS point by counting how many of the 30 preceding and succeeding points are within a 15 meters radius. An activity occurs when there is a sequence of points with a density higher than 15 for at least 10 points or 300 seconds.

2.3 Trip segmentation

Trip segmentation consists of dividing the users' trips into stages, which can be walk-stages or other-stages. To simplify terminology walk-stages are referred to as walks in the rest of

the paper. A walk occurs when the user is walking or is waiting for transport in a single place. An other-stage occurs when a user travels using a mode of transport (car, bus, train or other vehicle).

Trip segmentation is the most challenging part of mode detection. There is no commonly accepted solution for this task described in the literature. Solutions include using additional sensors, or they rely on high sampling frequency (≤ 10 s) for the GPS data.

All the trip segmentation algorithms found in literature were based in some way on speed. For instance, Biljecki et al. (2013) considered the user's stops as potential transition-points between modes. They identified a stop when consecutive points in an interval of 12 seconds did not have a speed higher than 2 km/h.

Shin et al. (2015) and Zheng et al. (2010) used acceleration to detect walks and stops. Since people usually walk or stop between two different transport modes Zheng et al. (2010) used a threshold of speed and of acceleration to divide the points into walks and non-walks, then they merged segments of points of the same type according to rules depending on the segment length. Zhu et al. (2016) labeled points as walk or non-walk based on speed and acceleration threshold values, then adjusted the labels based on nearby points: if at least M (a value dependent on the number of points) of the previous and posterior points have a different label, then the point's label is changed. Zhang et al. (2011) used heading change to identify stops. Liao et al. (2006) used GIS information for trip segmentation, in particular the proximity to transition locations such as a bus stop. However, this approach is less reliable in cities with a high density of bus stops. Unfortunately, none of the algorithms outlined above has been tested on a dataset with a low sampling frequency similar to the dataset collected in this study.

2.4 Mode detection

Many methods have been studied to automatically collect information about users' travel behavior from raw GPS data. One of the major problems is identification of travel mode. A recent review by Wu et al. (2016) identified two categories of methods for mode detection: machine learning methods and hybrid methods, also relying partially on machine learning, or on probabilistic models like Hidden Markov Models (Reddy et al. (2010)).

Nikolic and Bierlaire (2017) systematically reviewed the literature and found that the methodology adopted by all the studies was similar: first, some features are extracted from the sensors, then a training set is built to train a machine learning algorithm, and finally the algorithm is used to classify unseen data. For instance, Stenneth et al. (2011) compared different inference models including Bayesian Net, Decision Tree, Random Forest, Naive Bayesian and Multilayer Perceptron. They were able to classify different transport means as car, bus, train, bike, walking and stationary.

Reddy et al. (2010) built their classification system using accelerometer data in addition to GPS data and used a hybrid approach based on a decision tree and a Hidden Markov Model. Montoya et al. (2015) built a system based on a Bayesian network to infer the transport mode from smartphone data (GPS, wifi, accelerometer) and transport network information (e.g., public transport timetables). Patterson et al. (2003) presented a Bayesian model inferred in an unsupervised manner to distinguish between walk, drive or taking a bus. The research also showed that additional knowledge such as bus stop location can improve the algorithm results. Bantis and Haworth (2017) analyzed the relationship between personal and socio-demographic characteristics and travel mode choice using a Bayesian network.

Mode detection algorithms based on machine learning require manual labelling of user movements to train the model. This can be prohibitive when collecting data over a long period, since the labelling process requires significant effort from users. Another shortcoming of existing approaches is that none of the mode detection algorithms can detect the exact public transport vehicle used by the traveler, but rather the generic mode (bus, train, etc...). In this regard, it is important to mention Carrel et al. (2015) who did not perform a mode detection analysis but used automatic vehicle location (AVL) data from the public transport network to identify public transport trips.

3 Smartphone Application and Datasets

As part of this research, a smartphone app called *ETH-IVT Travel Diary* was developed to collect travel diaries over long periods of time while placing minimal burden on the users. The app was tested in a field trial with students at ETH Zürich. This section outlines app development and field testing. Analyses in the following sections are based on the records from this field trial.

3.1 App Design

The travel diary app developed in this research was designed to be as easy to use as possible by not requiring regular interaction with the respondent and not substantially affecting battery life (therefore the GPS sampling frequency must not be too high). The travel diary app was developed for the Android operating system and made available on the Google Play store.

The app's user interface consisted only of a brief study description, a field to enter the respondent's identification code and a button to start data collection. Once launched, a process runs in background collecting GPS coordinates and timestamps until the end of the study period.

The app requests location data from Android's internal location services since direct access to the device's GPS is not possible. To reach an optimal tradeoff between data quality and battery consumption, the location requests were sent with a low frequency (average sampling frequency of about 38 seconds). It is important to note that update of location information is at discretion of Android's location services. Typically, location is determined using GPS, Wi-Fi and Bluetooth. Update frequency and sensors used depend on the frequency and priority of location requests from all the smartphone apps and varies by operating system version and device.

The *ETH-IVT Travel Diary* app can also use location information collected from other apps by sending a zero-priority data request, which provides the information without triggering an update. This means the app data is more accurate when respondents simultaneously use fitness trackers or navigation apps. Finally, in contrast to prompted-recall approaches employed in earlier studies, no interface was provided for respondents to (re)view their records.

3.2 Data Collection

Students enrolled in the civil engineering program at ETH Zürich ($N = 1209$) were invited to participate in the study via e-mail in late March 2018. The study consisted of four weeks of tracking with the app. Students were paid CHF 20 to participate. No reminder e-mail

was sent.

A total of 48 students having an Android smartphone signed up to participate in the study. However, the particular smartphone operating system for 9 respondents blocked the data collection and therefore their data could not be used. The resulting data set, referred to as the “Zürich dataset” in this paper, consisted of the travel diaries of 39 students and two of the co-authors. In total, 1 053 days of travel diaries were collected, which corresponds to an average of 25.7 days per respondent.

At the end of the study, 35 respondents completed the exit survey providing feedback on the app. They rated app user-friendliness as high and 80% stated that battery consumption was acceptable. This is a good result given that the app requires respondents to have their GPS turned on at all times, which increases battery consumption.

3.3 Validation Dataset

A second dataset was used for validation and is referred to as the “validation dataset”. The validation dataset contains the ground truth of modes taken by users in addition to their GPS data.

The validation dataset was collected with a different smartphone application in the city of Basel (Switzerland) during early 2018 (following the setup described in Becker et al. (2018) and Becker et al. (2017)). It contains GPS data from 625 users, with an average of 7.4 days of travel each. The ground truth was obtained by using a proprietary automatic segmentation procedure that identifies stages when the user is not moving for a certain amount of time. Next, these stages are presented to the users through a web interface and the users are asked to manually specify the mode used for each stage. Asking users to label stages rather than individual GPS points reduces work for the users.

Interestingly, although users had the ability to correct the segmentation, they didn’t do it very often. This means the ground truth may be less accurate than in reality. For instance, if a user walks and then takes a bus, the whole trip may have been labelled as bus, then the information about the walk is not present in the dataset. In the following sections, we always refer to the Zürich dataset except for the private mode detection (Section 7.3) and the validation (Section 9), where we refer to the validation dataset.

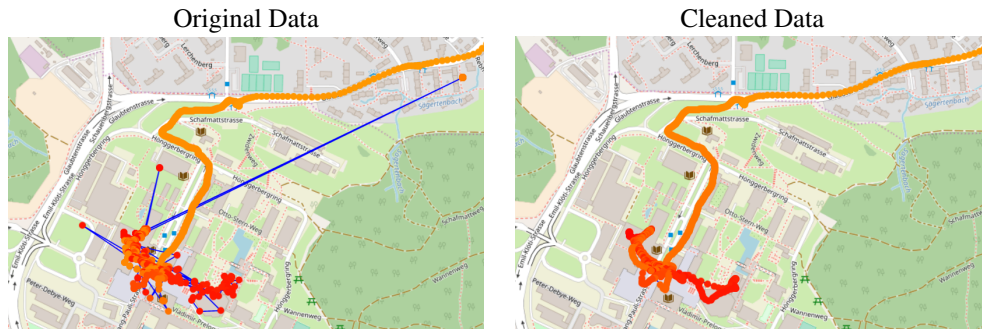
4 Data Cleaning

The data cleaning process consisted of filtering and smoothing. Two main features were used to filter erroneous GPS points: the speed and the angle between points. Since the smartphone app collected only GPS coordinates and timestamps, other features such as those described in 2.1 were not available.

The speed for each point was derived from the previous point. Points with a speed equal to zero were removed because in these cases it is likely that the smartphone merely returned the previous recorded position not the real one. Points with a speed of over 150 km/h were also removed since they were above the maximum accepted speed.

The second feature used in the filtering process was the angle between points (this feature has not been used in previous research to the best of our knowledge). Here, any point that forms a very small angle with the following and previous points, and which is far away from the previous point, is considered to be a false GPS point. More specifically: all points with an angle less than 15 degrees and a distance greater than 60 meters from the previous point

Figure 2: Data Cleaning: Comparison of original data with cleaned data. Colour represents the time (from orange to red).



Source: map from openstreetmap.org.

were removed. This rule was applied iteratively to all GPS points and the procedure was repeated until no more points are removed. The distance threshold was determined empirically; it was necessary because a small angle between near points can often occur if the user is walking or stopping. A limitation of this approach is that it assumes only one out of three consecutive points can be wrong, which means it cannot detect false points in case of two consecutive wrong points.

After completing the data filtering, the Kalman Filter was applied to smoothen the two space coordinates (longitude and latitude) of the GPS data. The Kalman Filter can deal with inaccurate observations and its efficacy increases with the frequency of the observations. For example, with a sampling frequency of 1 second, false points can be significantly corrected, while for a sampling frequency of 38 seconds (the average frequency of data collection in this study), only small adjustments are made to the user's trajectory. Figure 2 shows the application of data cleaning to part of a user's day, recorded with a high sampling frequency (≈ 1 second). In this example, some erroneous points are removed, because of their small angle. Furthermore, the smoothing process adjusted the trajectory of the points in the bottom-left part of the figure.

5 Trip and Activity Identification

The next step was to identify trips and activities. In this research, an activity was defined as a user remaining within 250 meters of the same point for at least 10 minutes. In turn, a trip is identified as a movement between two activities. The density-of-points method was used to identify trips and activities.

This method differs from those mentioned in Section 2.2 in that the algorithm does not rely on the number of GPS points (e.g. Schuessler and Axhausen (2009)) or on their frequency. In fact, the algorithm cannot rely on them, because the frequency of point data is too low. Therefore, in this research an activity was defined when there are at least 2 successive points within a 250-meter radius (*activityRadius*) for at least 10 minutes (*activityTime*). At least 2 points are required because with 1 point it is not known if there is an activity or if no signal has been received. The iterative algorithm is presented in Algorithm 1. The 250-meter

Algorithm 1 Trip and activity identification.

```
1: for each point P do
2:   if there are following points in a radius1 long activityRadius for at least activityTime
   then
3:     P is the starting point of the activity
4:     The last point in the radius is the ending point
5:   end if
6: end for
```

¹The center of the radius for each point Q following P is in the average point of all the points from P to Q.

radius used in this research is higher than used by others (e.g., 15 meters in Schuessler and Axhausen (2009) and 30 meters in Stopher et al. (2005)). It was made necessary due to the low precision of the GPS data. On the other hand, very short walks, starting and ending near an activity, are quite likely to really be part of the activity rather than an actual trip. When identifying activities and trips it is possible that the first and last points of an identified activity are in reality points from the previous or following trip. Therefore, two rules were added to the algorithm to better assign these points. These rules rely on a variable called center of mass. Center of mass is computed as the average of the coordinates of all the activity's points.

- If the distance from the activity starting point to the center of mass is greater than two times the average distance of all points to the center of mass, an activity is not identified and the algorithm is repeated from the next new activity starting point.
- If the distance from the activity end point to the center of mass is greater than two times the average distance of each point to the center of mass, then this point is removed from the activity and the algorithm.

These additional rules improve the algorithm's ability to detect the break points between activities and trips which helps improve mode detection.

Figure 3 illustrates an example of activity identification. As shown in the figure, the first and last points, describing the user's arrival and departure, are not included in the activity, even though they are within the 250-meter radius and they are not the farthest from the activity center of mass.

An additional rule was used to avoid the detection of false positive trips: a trip with an origin and destination very close to each other (less than 250 meters) and with a short duration (less than 5 minutes), was merged with the previous and following activity to create one single activity. This addresses the problem of erroneous GPS position recording caused by proximity to cell sites or antennas.

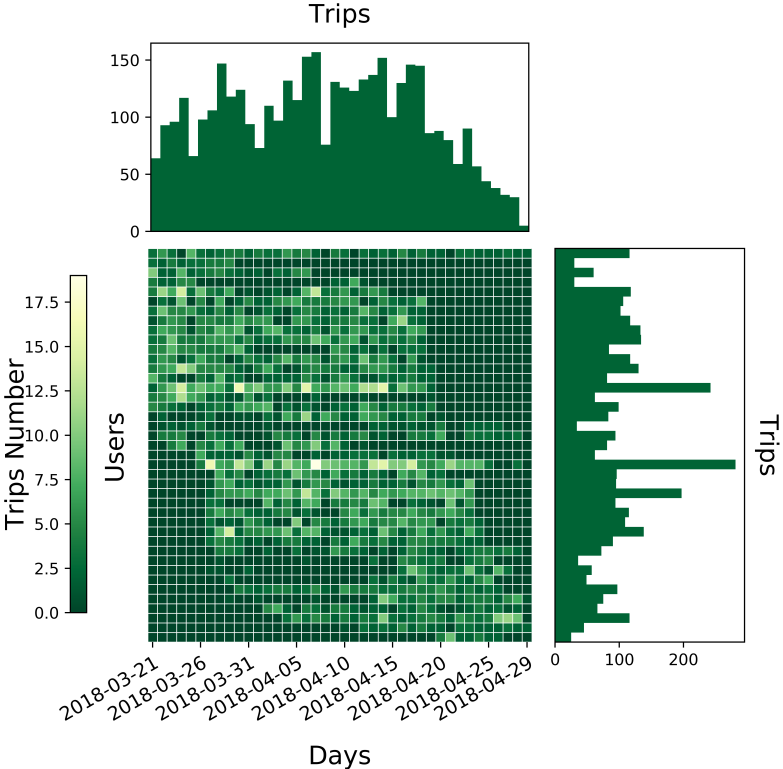
Figure 4 summarizes the data collected in the Zürich dataset. The study period consisted of 40 days (21 March until 29 April 2018). The central part shows the number of trips detected for each user for all days in the study period. The top bar chart shows there are fewer trips made on Sundays than on other days of the week. The right bar chart shows that the number of trips is different among users: there are 96.7 trips per user on average, with two users making more than 200 trips during the study period.

Figure 3: Activity identification: The red points are in the activity, the green points are the last points of the previous trip, the blue points are the first points of the following trip.



Source: map from openstreetmap.org.

Figure 4: Number of trips for each user for each day. The users are ordered by the moment they installed the app. The number of trips is aggregated by user and by day in the two bar charts.



6 Trip Segmentation

Trip segmentation consists of dividing a user’s trip into a sequence of walk and other-stages using a segmentation algorithm. In this research a walk was defined as a movement performed by walking or a transfer between different vehicles. An other-stage was defined as a movement performed using a vehicle. Therefore, a trip is formed by alternating walks and other-stages. After trip segmentation, all other-stages can be assigned a mode using a mode detection algorithm.

The GPS data recorded for this study have a low sampling frequency that varied considerably with different paths or users. Therefore, it was necessary to design a segmentation algorithm based on only GPS position and a derived speed that could work with irregular sampling frequency.

Before applying the trip segmentation algorithm, the trip data was checked to identify any trips which had an absence of signal for more than 7 minutes. If so, the trip was divided into two different trips, to avoid errors during the segmentation.

The trip segmentation algorithm consists of four steps. Step 1 is label specification; here each GPS point is marked as either a walk or an other-stage according to the speed threshold *minSpeed*. In step 2 the label of each point is adjusted based on the label of the points adjacent in time. In other words, if a single point is labelled walk in the middle of a series of other-stage points, it is changed to an other-stage. Step 3 consists of grouping the consecutive points of the same type into sequences. Step 4 consists of merging the sequences according to rules depending on duration, distance and speed.

The segmentation algorithm is presented in detail in Algorithm 2. The parameters *minSpeed*, *maxNearTime* and *scale* are used for the label specification. The parameters *minDuration*, *stageMinDuration* and *walkMinDuration* are used to merge small stages into walks and other stages. Their values were automatically tuned during the algorithm validation (explained in Section 9). In particular: *minSpeed* was set to 8.2 km/h to reduce the number of wrongly detected other-stages. This is higher than the 6.48 km/h specified by Zhu et al. (2016), but this was appropriate since in this research it was possible to correct falsely identified walks later using the mode detection algorithm. The *stageMinDuration* was set to 50 seconds because it is unlikely that a users other-stage would last less than 50 seconds. Similarly, the *walkMinDuration* was set to 70 seconds because it is unlikely that a user’s walk is shorter than 70 seconds. The values used for all parameters are shown in Table 1.

The segmentation algorithm principally relies on the speed of each point (derived from the position and the time). Information on acceleration and heading change were not used, because they are not reliable with a low sampling frequency. Therefore, there are a few cases in which the trip segmentation will fail: a fast walk can be detected as an other-stage; a rapid change of buses, with the second departing shortly after the first arrives, can be detected as

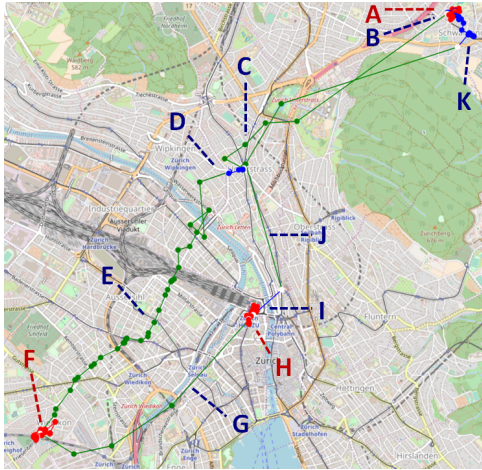
Table 1: Values of the segmentation parameters.

Parameter	Value	Parameter	Value
<i>minSpeed</i>	8.2 <i>Km/h</i>	<i>minDuration</i>	30 <i>seconds</i>
<i>maxNearTime</i>	30 <i>seconds</i>	<i>stageMinDuration</i>	50 <i>seconds</i>
<i>Scale</i>	0.8	<i>walkMinDuration</i>	70 <i>seconds</i>

Algorithm 2 Trip segmentation algorithm: Label specification (lines 2:20), merging rules (lines 22:25, 26:29, 30:33).

```
1: procedure SEGMENTATION(trip, minSpeed, maxNearTime, scale, minDuration,  
   stageMinDuration, walkMinDuration)  
2:   for each point  $\in$  trip do  
3:     if point.speed  $<$  minSpeed then  
4:       point.type  $\leftarrow$  walk  
5:     else  
6:       point.type  $\leftarrow$  other-stage  
7:     end if  
8:   end for  
9:   repeat  
10:    for each point  $\in$  trip do  
11:      adjacentPoints  $\leftarrow$  previous and next points in maxNearTime  
12:       $M \leftarrow \text{size}(\text{adjacentPoints}) * \text{scale}$   
13:      if more than  $M$  points in adjacentPoints are walk then  
14:        point.type  $\leftarrow$  walk  
15:      end if  
16:      if more than  $M$  points in adjacentPoints are other-stage then  
17:        point.type  $\leftarrow$  other-stage  
18:      end if  
19:    end for  
20:    until no changes  
21:    stages  $\leftarrow$  group all sequence of points of the same type  
22:    repeat  
23:      merge each stage with duration  $D < \text{minDuration}$  between  
24:      previous and next stage if their duration  $> D$   
25:    until no changes  
26:    repeat  
27:      merge a other-stage between two walks if its duration  $< \text{stageMinDuration}$   
28:    until no changes  
29:    repeat  
30:      merge a walk between two other-stages if its average speed  $> \text{minSpeed}$   
31:      or its duration  $< \text{walkMinDuration}$   
32:    until no changes  
33:    return stages  
34: end procedure
```

Figure 5: Trip segmentation of a user’s day (from A to K): activities (red), other-stages (green), walks (blue). The table reports the average time between two points (sampling interval) for the stages and the activities.



Source: map from openstreetmap.org.

	Sampling Interval (s)
A	76
B	85
C	56
D	29
E	21
F	25
G	124
H	53
I	312
J	264
K	53

a single other-stage; a vehicle stuck in traffic for a long time can be detected as a walk. To overcome these problems, information obtained from the mode detection algorithm applied in the next step of the process was used to improve the trip segmentation. This is explained in Section 7.

Figure 5 presents an example of trip segmentation. The user’s real path starts from the activity A where the user had a short walk (B) to take a tram (C). Then the user waited for a bus in D, took the bus (E) and arrived at F to stay there a while. Later the user took a train (G), stayed at the Zürich Main Station (H), walked to a stop (I), took a tram (J) and walked to home (K). Although the sampling frequency is different throughout the day, the segmentation algorithm is able to divide each trip correctly. More specifically, the figure shows that the frequency is lower when the user is on a train (G) or in a tunnel (the upper-right part of C and J). It is also lower on I, because part of the main station is underground.

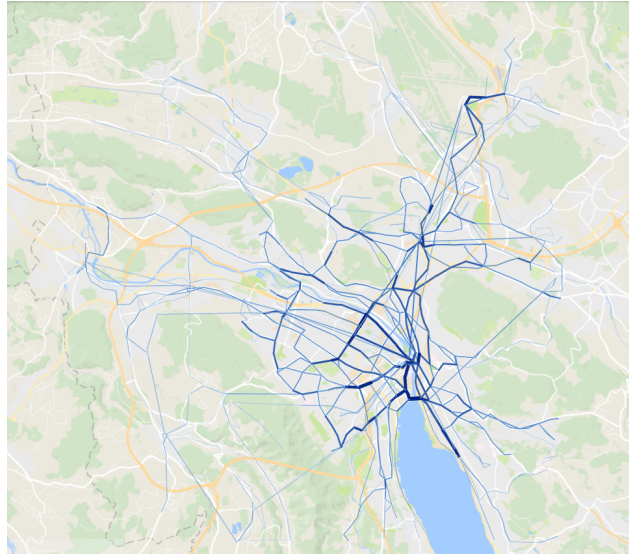
7 Mode Detection

Mode detection consists of assigning a specific mode of transport to the stages identified as other-stages in the trip segmentation. As outlined in Section 2.4, all the studies except Patterson et al. (2003), rely on inference models using a training data set to perform mode detection. Creating a training data set requires manually labelling of the transport modes, which limits the amount of data easily available.

In this study GPS data sampling frequency is a crucial parameter for the mode detection algorithm, because of the smartphone application’s low sampling frequency. Unfortunately, most published mode detection studies lack information about sampling frequency.

A specific mode detection algorithm was developed in this research to use this low sampling frequency GPS data. The mode detection algorithm is unsupervised, and it does not use any

Figure 6: Public transport traffic in Zürich: darker and larger lines represent more transport vehicles travelling on a working day (26-03-2018).



Source: map from google.com.

statistical inference model. Instead, it uses actual public transport operations data. Only Stenneth et al. (2011) used this type of data, but only to build features for their inference model.

Figure 6 illustrates Zürich's extensive public transport traffic network of buses, trams and trains. The system is efficient and well used, in fact the city's modal split is 32% public transport, 33% walking, 21% motorized private transport, 12% bicycle and 2% other (Städtevergleich Mobilität 2015 (2018)). Since the Zürich dataset was collected mainly from students, its modal split could be different from the citywide figures. The actual public transport operational data consisting of planned and actual arrival and departure times for all vehicles are available for all stops in Zürich (SBB Opendata (2018)).

The mode detection algorithm uses this operational data to label an other-stage as being carried out by bus, tram, train or otherwise a private vehicle. An addition to this algorithm described in Section 7.3 shows how to distinguish between cars and bikes for private vehicle stages. Moreover, an original contribution of the algorithm developed in this research is its ability to detect the exact public transport vehicle taken by the user.

The algorithm works as follows: given an other-stage, identify all the public transport stops in a radius *detectionRadius* near the starting point of the other-stage, next select all the vehicles stopping at one of these stops in \pm *detectionTime* from the starting point. Repeat the same process for the end point of the other-stage. Next, find the intersection of the vehicles in the two groups (this is a list of all the vehicles passing near the user at the beginning of the other stage and at the end). Finally, apply a likelihood function to each element of the vehicle list to identify the most probable vehicle taken by the user. If the list is empty, that means a private vehicle was used for the trip. In this research the parameters were set to

$detectionTime = 300 [sec.]$ and $detectionRadius = \max(250, \min(accuracy, 400)) [m]$ where accuracy is a value in meters provided by the smartphone for each GPS point.

To overcome the incorrect identification of walks during trip segmentation (e.g., the algorithm identifies a walk when, in fact, a vehicle is stuck in traffic), a further rule is applied: if a trip has the pattern “other-stage, walk, other-stage” and the two other-stages are not assigned to a means of transport, then the three stages are considered as a single other-stage, and the mode detection algorithm is computed again on the new other-stage. To overcome the problem of undetected walks, the user’s past data are used (described in Section 7.2).

7.1 Likelihood function

This research used a likelihood function to determine which vehicle out of a set of possible vehicles best matches the users other-stage, computing the degrees of similarity between the users path and the paths of the vehicles. Other studies have also used probabilistic functions or a rule-based system for the mode detection although in a different manner. For instance, Schuessler and Axhausen (2009) and Tsui and Shalaby (2006) used fuzzy logic approaches with rules based on speed and acceleration of the GPS records.

The likelihood function to determine vehicle used, $L(v,s)$, compares the path of a user’s other-stage s with the path of a vehicle v . It is the product of a function of space/time likelihood between the paths, $L'(v,s)$; and a scaling factor $T(v,s)$, which takes into account how much the same time/distance path from a given vehicle can explain the user’s entire trip. The mathematical formulation of $L(v,s)$ is shown below:

$$L'(v, s) = \lambda * TimeDifference(v, s) + (1 - \lambda) * PathDistance(v, s) \quad (1)$$

$$T(v, s) = \text{times } v \text{ is a candidate vehicle for a other-stage in the same trip of } s \quad (2)$$

$$L(v, s) = L'(v, s) * T(v, s) \quad (3)$$

The function $L'(v,s)$ is also a combination of two functions: *TimeDifference* and *PathDistance* which compare the user and vehicle paths in time and space.

TimeDifference is the sum of the difference between the vehicle and user departure times and the difference between the vehicle and user arrival times. *PathDistance* is the average Euclidean distance between the user’s coordinates and the vehicle coordinates. The vehicle coordinates were calculated at the same timestamps as the user’s coordinates (GPS points), by interpolating from the arrival time and the coordinates of each stop.

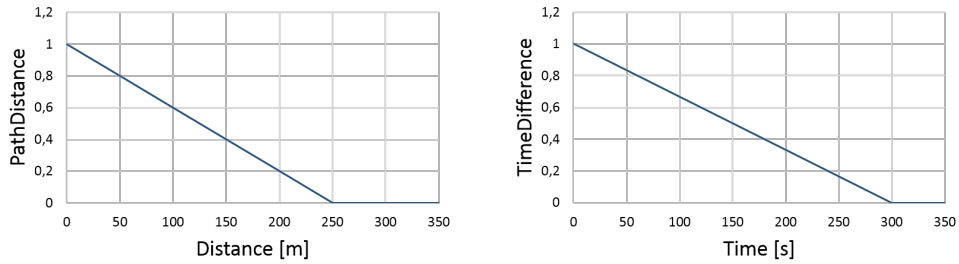
The *TimeDifference* and *PathDistance* values were scaled to be comparable as shown in Figure 7, considering a maximum value of *TimeDifference* = *detectionTime* (300 seconds) and a maximum value of *PathDistance* = *maxPathDistance* (250 meters). A value of 0.5 was chosen for λ because using only the *TimeDifference* can result in a false positive of matching a user with a vehicle that had a different path; meanwhile the *PathDistance* is not reliable if there are too few points from the user. Vehicles with *PathDistance* = 0 were discarded, except for trains since train path is hard to describe using only stop point data.

The scaling factor, $T(v,s)$, is the number of times that the vehicle v appeared in one of the lists of candidate vehicles for other-stages of the user’s in the same trip of s . This reflects the idea that if a single vehicle is a candidate match for different other-stages of the same trip, it is probable that the user took only one vehicle for all the other-stages.

To compute the likelihood that the user travelled on the given vehicle $L(v,s)$, the function $L'(v,s)$ is multiplied by the scaling factor $T(v,s)$. It is important to note that the value of L is

not comparable for different other-stages, because it is dependent on the quality of the user’s path data. With low quality data, L tends to be low for the taken means; in contrast, with high quality data L approaches 1. Furthermore, vehicle path data is always of high quality, because they are based on the position of the stops (known) and on the actual arrival times (provided by the operator and which are subject to smaller errors than user smartphone GPS data).

Figure 7: Mode detection likelihood: Scaling of the two sub-functions *PathDistance* and *TimeDifference*.



7.2 Using past data to identify transfers

The mode detection algorithm may miss transfers between two vehicles, especially when the transfer is performed quickly. Therefore, this research developed a new method for identifying missed transfers by applying the user’s past data to improve trip segmentation. This new method uses a personalized map of the places visited by each user from their travel history. The visited places consist of each activity (its center of mass) and the starting and ending points of each other-stage. For close together places (within 250 *meters*), only the center of mass is considered, since they represent the same location.

After assigning modes to other-stages, the mode detection algorithm tries to detect missing transfers based on the user’s visited places map. The places near the user’s path (distance less than 400 *meters* from any point on the path) are considered possible transfer points. Next, the algorithm tries to detect if a transfer was feasible for each of these points. The mode detection is (re)computed from the starting point to the potential transfer point and from the potential transfer point to the ending point. If the mode detection algorithm can identify two public transport means that could support a transfer at this location, then it is assumed the transfer was made.

The process is based on the assumption that the user’s travel behavior is recurring, especially during weekdays and for commuters. Therefore, places where the user has already been have a higher probability for the user to perform a transfer, especially if the user performed a transfer there on previous trips. This technique represents the first attempt, to our knowledge, of using the user’s past data for a in a mode detection algorithm. This technique is particularly useful (and possible) for user datasets spanning multiple weeks with a large quantity of data.

7.3 Private mode detection

The main objectives of this study were to develop a smartphone application and mode detection algorithm to obtain travel behavior information with minimal impact on users. The methodology described in the previous sections is able to distinguish the used mode among walk, bus, tram, trains and private vehicles. Then, in this section, it is shown that is possible to integrate an additional module to distinguish between bicycles and cars for private stages. This module is not described in detail, since it is similar to classical mode detection algorithms (described in Wu et al. (2016) and Nikolic and Bierlaire (2017)), although the classification is performed only between two modes (bicycle and car).

The private mode detection used machine learning to identify modes. This required a ground truth and therefore the validation dataset was used to train and evaluate the private mode detection model. All the stages marked in the validation data as performed by bicycle or car were selected, then a set of features were extracted to represent each stage by a vector of features. The selected features were: number of points; length of the stage (*meters*); duration of the stage (*seconds*); average distance between two consecutive points; maximum speed; average speed; median speed; maximum acceleration; average acceleration; median acceleration; average angle formed by a point with the previous; median angle.

The validation dataset was divided in 70% for the training set and 30% for the test set. Then, several classification algorithms were tested: logistic regression, svm, decision tree and random forest. The one with the greatest accuracy, defined as the percentage of correct detection, was selected for use in this study. The results of this procedure are described in Section 9.4.

8 Assessment of Algorithm Results using Zürich Dataset

Two methods were used to assess the algorithm and methods developed in this research. First the algorithm was applied to the Zürich dataset to test the overall ability of the smartphone application and algorithm to understand users movements. Second, the algorithm was applied to the Basel dataset (which included the ground truth of modes actually taken by the users) to determine how well the algorithm performed. This section describes testing with the Zürich dataset, section 9 describes testing with the Basel dataset.

Table 2 presents the results of the mode detection algorithm applied to the Zürich dataset. The top portion summarizes the number of activities, trips, walks and other-stages detected by the algorithm and the bottom portion the mode detection results. As shown in the top of Table 2 the algorithm found that users performed 3.8 activities per day and made 3.8 trips (based on a total of 1,053 days of valid data). These trips consisted of 8.1 walk stages and 5.3 other-stages. These results are reasonable for the study participants (university students).

The lower part of Table 2 summarizes how the algorithm classified other-stages. As shown, the other-stages are divided into three groups: *detected stages* (34%), *not assigned stages* (16%), and *ignored stages* (50%). Stages are marked as *detected* if the mode detection algorithm was able to identify a mode. Stages are marked as *not assigned* if the algorithm could not identify a mode. Stages outside Zürich were ignored because this study relied on data from the city of Zürich.

The *not assigned* stages principally consist of other-stages performed with a private vehicle but could also include public transport stages that were not detected due to low GPS quality

Table 2: Zürich dataset activities, trips and stage data obtained by the mode detection algorithm.

	Quantity	Per user day
Activities	3975	3.8
Trips	3965	3.8
Walks	8564	8.1
Other-stages	5548	5.3
Detected stages	1906 (34%)	1.8
Not assigned stages	863 (16%)	0.8
Ignored stages (outside of Zürich)	2779 (50%)	2.6
Past Data Detected stages	64	0.1

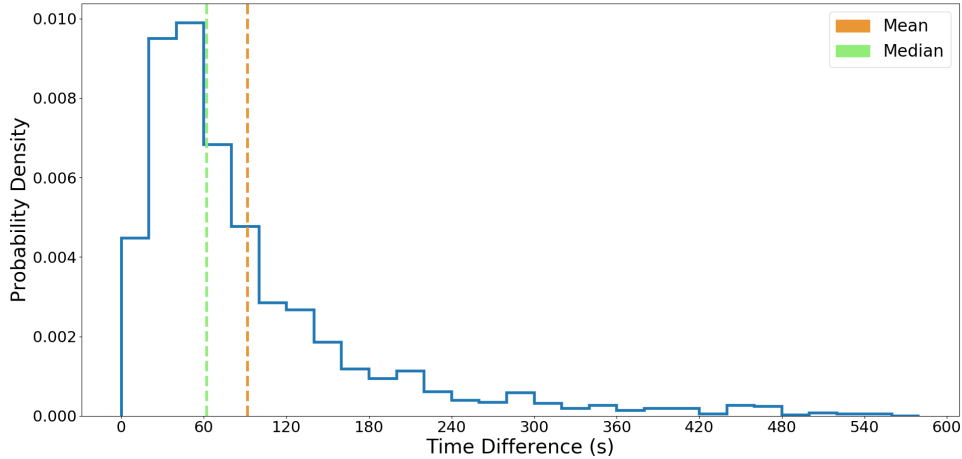
or problems in the activity identification or trip segmentation steps. For instance, the *not assigned* stages could include false positives such as if the user is travelling in a car or on a bike directly behind a bus and the algorithm identifies bus as the transport mode. However, this case is considered rare because a car can overtake or has a different lane and a bicycle normally has a higher travel time.

The 64 stages detected using past data (representing 3.4% of the detected stages) would have been labelled as *not assigned* if the past data had not been used, with an increase of the *not assigned* group of the 7.4%. This indicates the importance of using information about the user’s past travel behavior for mode detection. Its impact will be better analyzed in Section 9.3.

To measure the quality of the mode detection, the *TimeDifference* function (described in Section 7.1) was used. This function represents the sum of the difference of the departure times and the difference of the arrival times of the user and the detected vehicle. A low value indicates that the detection is correct, because the user and the vehicle were in the same places at the same times. The *PathDistance* function is not as good an indicator because the GPS data describing a user’s path can be quite noisy.

Figure 8 presents the distribution of the *TimeDifference* for all detected stages in the Zürich dataset. This value depends on two main factors: the trip segmentation and the sampling frequency. In particular, an erroneous trip segmentation can identify the beginning or the end of the other-stage at a point before or after the real beginning or end point. Importantly, due to the low sampling frequency there are often no points in the dataset representing the exact time the user boarded the transport vehicle. For this reason, the distribution’s mean value of 91 s and a median value of 62 s for the *TimeDifference* can be considered good values and a strong indication of correct matching. Instead, with higher values, such as more than 300 s, the probability of a wrong detection increases.

Figure 8: Distribution of *TimeDifference* for all the *detected* stages in the Zürich dataset (grouped each 20 s)



9 Assessment of Algorithm Results using Validation Dataset

Showing that the average time difference between the users' paths and the detected public transport means is low demonstrates the validity of the proposed algorithm and shows good overall results but it does not show clearly the accuracy of mode detection. Therefore, the mode detection algorithm was also evaluated using the validation dataset collected in Basel, described in Section 3.3, which contains the ground truth data about user modes.

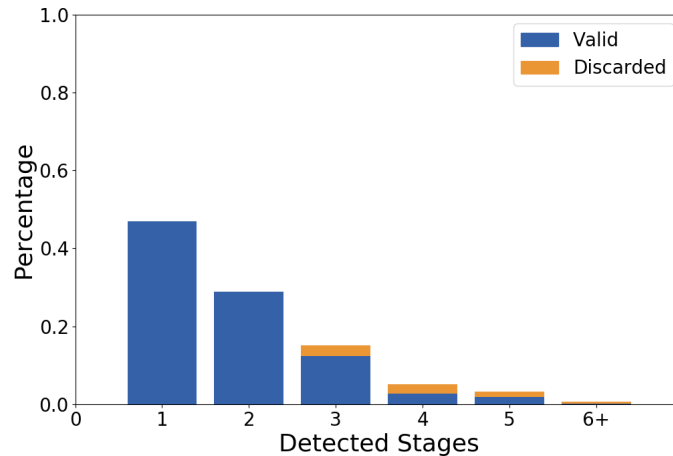
9.1 Preparing the validation dataset

The first step in using the Basel dataset was preparing the data. More specifically determining which labelled stages of the validation dataset can be considered valid for use in determining the accuracy of mode choice detection results. This step is needed because participants may have erroneously labeled a sequence of stages as a single stage, due to imprecise framing of the labelling question at the time of data collection or laziness in reporting. These cases are characterized by having fewer stages in the validation dataset compared to actual.

The process began by comparing each stage available in the validation dataset (labelled stage) with all the stages identified by the mode detection algorithm in the same time interval. If there was at least one stage with the same mode, the detection was considered correct. Figure 9 presents the distribution of number of stages detected by the algorithm for each labelled stage from the validation dataset. As shown, almost 50% of the stages perfectly match with one stage; 30% with two stages, which always include a walk stage; and very few with more than two stages, showing that the validation methodology provides a reasonable upper bound.

If there is only one stage detected for one labelled stage, then the labelled stage can be considered valid, since the two stages can be easily compared. If there are two detected stages

Figure 9: Distribution of the number of stages detected by the mode detection algorithm with an interval of time in common for each labelled stage from the validation dataset. In orange, stages not considered for the validation, since they are detected as performed by both Bus/Tram and a private vehicle.



for one labelled stage, then the labelled stage can be considered valid because one of the detected stages can be assumed to be a walk and every real trip includes a walk.

In cases where the algorithm identifies more than two stages there is some ambiguity as to whether the validation data has been correctly reported. In these cases if a labelled stage is detected by the algorithm as performed by both a bus/tram and a private vehicle, the stage is discarded as not valid, and not considered further for the validation, as it is impossible to associate a ground truth to it. This case is highlighted in orange on Figure 9; it occurs for only 7.4% of the stages. Further details on the combinations considered valid are available in the Appendix.

There were several shortcomings in the Basel dataset. First, actual public transport data was only available for half of the buses and trams in the network. For the other half, the algorithm used planned timetable data (two operators work in Basel, BVB and BLT, and the realized data are provided only by BVB). Therefore, the algorithm cannot identify buses or trams when they are delayed by more than *detectionTime*. Second, there are on average only 7.4 days of data recorded per user, meaning that the user’s travel history has only limited value in identifying possible transfers. Finally, to avoid errors in data collection, data from outside the city of Basel or that had stages with no signal for over 7 minutes were not used in the validation, to avoid errors due to the data collection.

9.2 Tuning mode detection algorithm parameters

A subset of the validation data consisting of approximately 400 days of tracking data (about 8%) was used to tune the parameters in the mode detection algorithm. Since the algorithm contains several parameters for each step (cleaning, activity identification, trip segmentation and mode detection), it was prohibitive to analyze all possible parameters. Therefore,

Table 3: Different values for the validation parameters. The selected best values are bold.

Parameter	Values
maxNearTime (s)	{ 30 , 60, 90}
minDuration (s)	{10, 20, 30 }
minSpeed (m/s)	{6.8, 7.5, 8.2 }
detectionTime (s)	{150, 200, 250, 300 }
maxPathDistance (m)	{150, 250 , 300, 450}

Table 4: Mode detection accuracy, express as percentage of correct detection.

Mode	Total	Detected	Accuracy
All Stages	5659	4875	86.14 %
Walk	1520	1435	94.41 %
Bus/Tram	888	716	80.63 %
Train	123	84	68.30 %
Private	3128	2640	84.40 %

only the five most relevant parameters were selected for tuning: three for segmentation (*maxNearTime*, *minSpeed* and *minDuration*) and two for mode detection (*detectionTime* and *maxPathDistance*). The other parameters were set as described in previous sections, except the *activityRadius*; this was set at 100 meters, to better align with data from a different smartphone application used in the Basel study.

Table 3 presents the different values considered for the five parameters. The set of evaluated combinations is the Cartesian product of those parameters. The final configuration was selected in order to have a balanced accuracy, in terms of percentage of correct detection, among the different modes. Low values of *detectionTime* and *maxPathDistance* lead to fewer trips being assigned to public transport mode and more trips being assigned to private mode. When lower values are used the algorithm tries to match the user’s path only with the public transport means closer to the user during the trip.

Theoretically, with a correct segmentation and good quality of the data, the mode detection algorithm should work correctly with low values of these parameters. However, when the quality of GPS data is poor, and the segmentation is incorrect, higher values are needed to guarantee the matching of user trips to public transport data. Therefore, improving the quality of GPS data and segmentation makes it possible to decrease *detectionTime* and *maxPathDistance*, and improve the quality of mode detection algorithm by reducing the number of false positives.

9.3 Accuracy of Mode Detection Algorithm

After the mode detection algorithm was tuned it was used to detect modes from the rest of the validation dataset. Table 4 presents results of that analysis showing the accuracy of the mode detection algorithm in terms of percentage of correct detections. Since the validation

Table 5: Mode detection accuracy without using past data.

Mode	Acc. with past	Acc. without past
All Stages	86.14 %	86.15 %
Walk	94.41 %	94.28 %
Bus/Tram	80.63 %	76.53 %
Train	68.30 %	66.93 %
Private	84.40 %	85.68 %

dataset contains only one label for bus and tram, the two modes were considered as one. The highest rate of correct detection was for walking. This shows the high accuracy of the segmentation algorithm. On the other hand, trains have the lowest rate of correct detection since the quality of user GPS data is lower inside stations and trains, and therefore it is harder to identify the exact GPS point when the train is departing (or arriving). The Bus/Tram stages were moderately well detected; the main problems for detecting them were the low quality of GPS data; segmentation errors; lack of user past data; and mainly the fact that actual public transport operations data were not available for all lines. Finally, the quality of private mode identification depended on the parameters *detectionTime* and *maxPathDistance*, chosen in order to have a balanced detection of public and private modes.

In conclusion, the average accuracy of the mode detection algorithm was found to be comparable with the state of the art (the average accuracy of the works reviewed by Nikolic and Bierlaire (2017) is between 75% and 98%). Furthermore, these results can be significantly improved with more past data from users and actual public transport operations data for the entire network. Table 5 presents results from the mode detection algorithm with and without using the past user data. These results show the importance of users' past data, even though there are only 7.4 trips per user on average. In particular, the use of past data increases the accuracy of Bus/Tram stage detection by 4.1% and of train stages by 1.4%, by detecting missed transfer points. On the other hand, the slight decrease of accuracy for private mode stages is due to the detection of false transfer points. Table 6 compares the accuracy of the algorithm on the validation dataset using the realized public transport data (for the trains and the bus/trams with associated realized data) versus using only planned timetable data. As shown, the accuracy of public transport mode detection using planned schedule data is sharply lower. On the other hand, the accuracy for walks and private modes is almost the same (it is slightly different because their detection also depends on the public transport detection). These results confirm that the accuracy of the mode detection algorithm would have been greater if realized data had been available for all the public transport lines.

9.4 Results of the private mode detection

A private mode detection algorithm, described in Section 7.3, was used to distinguish private trips between bicycles and cars. The research tested several classification algorithms and selected the random forest algorithm with a maximum depth = 5 and 50 trees (since it performed best as defined by the highest percentage of correct detection). The training dataset for private mode detection needed to take into account that the Basel dataset con-

Table 6: Impact of realized data of public transport.

Mode	Accuracy Realized	Accuracy Timetable
All Stages	86.14 %	82.22 %
Walk	94.41 %	93.69 %
Bus/Tram	80.63 %	55.51 %
Train	68.30 %	60.33 %
Private	84.40 %	84.44 %

tained many more bicycle trips than car trips. Therefore, each sample has a weight inversely proportional to the class frequencies, to equally train the classifier.

The confusion matrix is shown in Table 7. As shown the private mode detection algorithm had an overall accuracy of 86.75%. This good result is in spite of the fact that the segmentation procedure for the validation dataset was not perfect, as described in Section 3.3, and that no data were available from an accelerometer or other sources that could help distinguish between cycling and automobile travel.

Table 7: Confusion matrix and results of the private mode detection.

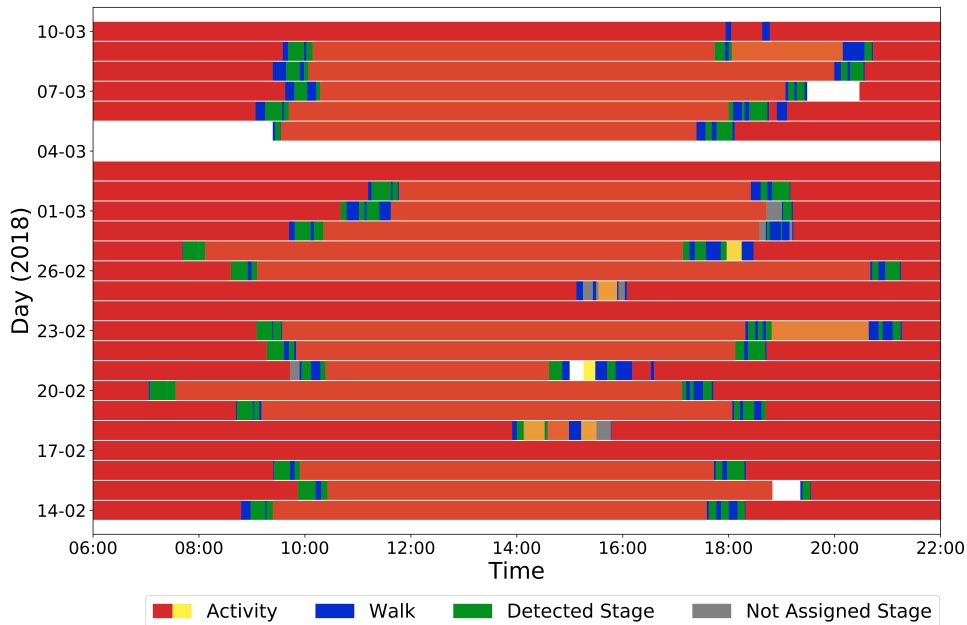
		Detected			Accuracy
		Bike	Car	Total	
Real	Bike	915	92	1007	91 %
	Car	119	466	585	80 %
Correct stages		915	466	1592	87%

10 Conclusions and Future Work

Results of this research confirm that GPS data collected from smartphones are a powerful means for understanding travel behavior and have several advantages over traditional survey methods. The research also clearly demonstrated that it is possible to develop a smartphone GPS tracking application that overcomes two of the main problems with earlier tracking technology by placing very low demand on the smartphone battery and requiring almost no work by the user. This ease of use makes it possible to easily track a large number of travelers for long periods of time, thereby significantly increasing the amount of data available for analyzing travel behavior.

Furthermore, the research shows that it is possible to understand user travel behavior based on only low-frequency and low-precision GPS data by designing and testing specific algorithms for activity detection, trip segmentation and mode detection for use with this type of data. The mode detection algorithm used in this research is an improvement over other methods because it only needs user GPS data and public transport network data. Most existing work described in the literature is based on supervised learning, requiring significant efforts to manually label data.

Figure 10: Continuous tracking of a single user for different days. Activities in the same place have the same color, which goes from red to yellow according to the time spent doing the activity. A white space indicates absence of signal.



In addition to distinguishing between public transport and private modes, the algorithm also is able to detect the exact public transport vehicle used by the traveler. The method represents an original attempt to use actual operations data for a travel survey purpose. The algorithm also includes a method for exploiting the users past data to detect transfers and thereby improve the quality of mode detection. The research results show clearly that this extra information helps improve the quality of results.

The research points out several paths for future work. First, the ability to unobtrusively collect data from users over long periods of time means it is possible to obtain much more interesting travel behavior data. For example, Figure 10 illustrates all the movements of one study participant over a 25-day period. Assuming that the participant's home and workplace are easily recognizable, there is a clear pattern in the morning trips from home to work. Although it is less regular, there is a similar pattern in the opposite direction in the evening. This illustrates a first step in discovering travel patterns and traveler choices that cannot be derived from a small dataset.

Another path for future work is detection of the exact public transport vehicle. This is powerful information which cannot be obtained with a traditional mode detection algorithm. This precise data will make it possible to better analyze why people make specific travel choices (e.g., choosing a certain line), and to understand their main criteria for these choices. Lastly, the authors plan to use the methodology to collect a large-scale dataset, in terms of both users and days of tracking, which will support a wide variety of travel behavior research.

Acknowledgements

The authors wish to thank all study participants and the Swiss Federal Railways (SBB) for the open data.

Funding: This work was partly supported by projects funded by the Swiss Federal Road Office (project ASTRA 2017/006) and the Swiss Innovation Agency (Innosuisse, SCCER Mobility).

Declarations of interest: none.

Appendix

Valid combinations of detected stages

Table 8 shows the valid combinations of detected stages for each type of labelled stage, i.e. the combinations that can be used in evaluating the mode detection algorithm's accuracy.

The first four rows of Table 8 are cases when one stage is detected. In other words, the mode detection algorithm identifies one mode for a case where the validation dataset shows one mode. In the case of a one to one correspondence, the detected mode must be the same to be classified as correct (this is the case of the first column in Figure 9).

The next three rows are cases where the mode detection algorithm detects two stages, although the validation dataset shows only one stage. In these cases, the algorithm has detected a walk. When the algorithm detects two stages one of them is always a walk, since the trip segmentation algorithm divides a trip into alternating walks and other-stages. In other terms, a combination {Bus,Private} would be algorithmically infeasible, since there would be a small walk in between the two. Furthermore, in the real world it is common to walk before or after taking a vehicle. Since it is possible the user labelled the stage with only one mode (the one considered most relevant by the user), the detection is considered valid and correct either for a walk or for the other detected mode.

The next two rows are cases where the mode detection algorithm has detected three different stages for a single stage in the validation dataset. When the algorithm detects a labelled stage as performed by a train; either a bus or a private transport; and a walk (representing the transfer), the combination is considered valid, since it is common to take a bus or car to reach a train station.

The bottom two rows describe cases where the mode detection algorithm has identified combinations that include a bus and a private mode. A combination of this type is considered implausible and these data are discarded (this case is highlighted in orange in Figure 9).

The reason for discarding this data is that it is not possible to determine whether the algorithm made a mistake, or the user made a mistake when labelling the data. For instance, assume a user has performed a trip with 5 stages consisting of: a bus stage, a walk, a private stage, a walk, and again a bus stage (that would fit the scheme {Bus, Walk, Private} in Table 8). The user labels this as a single stage, associated to private mode. The algorithm correctly identifies the 5 stages but is unable to determine if this match is correct or incorrect.

Another possibility is that the algorithm is incorrect. For example, assume the user performs a trip by car, and the car drives so close to a bus that the algorithm assigns the first part of the trip to a bus stage, and something similar happens on the last part of the trip. The algorithm would detect the same 5 stages listed above, although the user (correctly) would label this as a private trip.

Table 8: Valid combination of detected stages for each type of labelled stage. The correct are marked. The last two combination are discarded from the validation. The order of the modes is not relevant. Repetition of the same mode are not considered in the table. Combination with at least 2 stages and without a walk are not algorithmically feasible.

Labelled stage \ Detected stages	Bus	Walk	Private	Train
{Bus}	X			
{Walk}		X		
{Private}			X	
{Train}				X
{Bus, Walk}	X	X		
{Private, Walk}		X	X	
{Train, Walk}		X		X
{Bus, Walk, Train}	X	X		X
{Private, Walk, Train}		X	X	X
{Bus, Walk, Private}	discarded			
{Bus, Walk, Private, Train}	discarded			

Previous analysis has shown that user labelling in the original dataset was prone to misidentification of successive stages. For instance, users systematically identified a sequence of stages “walk, private, walk” as a single private stage. Therefore, it is not possible to exclude either of the two previously described possibilities. It is not clear whether it is the fault of the user, or the fault of the algorithm, and therefore we discard this labelled stage when determining the correctness and accuracy of the algorithm.

Finally, a labelled stage detected by the algorithm as an activity, like a slow walk near the same place, has been discarded for the validation, since it represents an activity and not a stage, according to our definition.

References

- Ansari Lari Z, Golroo A, 2015. Automated Transportation Mode Detection Using Smart Phone Applications via Machine Learning: Case Study Mega City of Tehran. In: Transportation Research Board 94th Annual Meeting
- Axhausen KW, König A, Schlich R, 2000. Deskriptive Darstellung der Befragungsergebnisse des Projektes Mobidrive. Report, ETH Zurich, DOI 10.3929/ethz-a-004234136
- Bantis T, Haworth J, 2017. Who you are is how you travel: A framework for transportation mode detection using individual and environmental characteristics. Transportation Research Part C: Emerging Technologies 80:286–309, DOI 10.1016/j.trc.2017.05.003
- Becker H, Ciari F, Axhausen KW, 2017. Modeling free-floating car-sharing use in Switzer-

- land: A spatial regression and conditional logit approach. *Transportation Research Part C: Emerging Technologies* 81:286 – 299, DOI <https://doi.org/10.1016/j.trc.2017.06.008>
- Becker H, Ciari F, Axhausen KW, 2018. Measuring the car ownership impact of free-floating car-sharing - A case study in Basel, Switzerland. *Transportation Research Part D: Transport and Environment* 65:51 – 62, DOI <https://doi.org/10.1016/j.trd.2018.08.003>
- Biljecki F, Ledoux H, van Oosterom P, 2013. Transportation Mode-based Segmentation and Classification of Movement Trajectories. *International Journal of Geographical Information Science* 27(2):385–407, DOI 10.1080/13658816.2012.692791
- Bohte W, Maat K, 2009. Deriving and validating trip purposes and travel modes for multi-day GPS-based travel surveys: A large-scale application in the Netherlands. *Transportation Research Part C: Emerging Technologies* 17(3):285–297, DOI 10.1016/j.trc.2008.11.004
- Carrel A, Lau PSC, Mishalani RG, Sengupta R, Walker JL, 2015. Quantifying transit travel experiences from the users? perspective with high-resolution smartphone and vehicle location data: Methodologies, validation, and example analyses. *Transportation Research Part C: Emerging Technologies* 58:224–239, DOI 10.1016/j.trc.2015.03.021
- Cottrill C, Pereira F, Zhao F, Dias I, Lim H, Ben-Akiva M, Zengras P, 2013. Future Mobility Survey: Experience in developing a smartphone-based travel survey in singapore. *Transportation Research Record: Journal of the Transportation Research Board* 2354:59–67, DOI 10.3141/2354-07
- Gong H, Chen C, Bialostozky E, T Lawson C, 2012. A GPS/GIS method for travel mode detection in New York City. *Computers, Environment and Urban Systems* 36:131–139, DOI 10.1016/j.compenvurbsys.2011.05.003
- Greene E, Flake L, Hathaway K, Geilich M, 2016. A Seven-Day Smartphone-Based GPS Household Travel Survey in Indiana. In: *Transportation Research Board 95th Annual Meeting*
- Jun J, Guensler R, Ogle J, 2006. Smoothing Methods to Minimize Impact of Global Positioning System Random Error on Travel Distance, Speed, and Acceleration Profile Estimates. *Transportation Research Record: Journal of the Transportation Research Board* 1972:141–150, DOI 10.3141/1972-19
- Liao L, Patterson DJ, Fox D, Kautz H, 2006. Building personal maps from GPS data. *Annals of the New York Academy of Sciences* 1093:249–265, DOI 10.1196/annals.1382.017
- Montini L, Rieser-Schüssler N, Horni A, Axhausen KW, 2014. Trip Purpose Identification from GPS Tracks. *Transportation Research Record: Journal of the Transportation Research Board* 2405:16–23, DOI 10.3141/2405-03
- Montoya D, Abiteboul S, Senellart P, 2015. Hup-me: inferring and reconciling a timeline of user activity from rich smartphone data. In: *ACM SIGSPATIAL, International Conference on Advances in Geographic Information Systems*, DOI 10.1145/2820783.2820852
- Nikolic M, Bierlaire M, 2017. Review of transportation mode detection approaches based on smartphone data. In: *17th Swiss Transport Research Conference, Ascona*

- Nitsche P, Widhalm P, Breuss S, Brändle N, Maurer P, 2014. Supporting large-scale travel surveys with smartphones - a practical approach. *Transportation Research Part C: Emerging Technologies* 43:212–221, DOI 10.1016/j.trc.2013.11.005
- Ogle J, Guensler R, Bachman W, Koutsak M, Wolf J, 2002. Accuracy of Global Positioning System for Determining Driver Performance Parameters. *Transportation Research Record: Journal of the Transportation Research Board* 1818:12–24, DOI 10.3141/1818-03
- Oliveira M, Vovsha P, Wolf J, Birotker Y, Givon D, Paasche J, 2011. Global Positioning System-Assisted Prompted Recall Household Travel Survey to Support Development of Advanced Travel Model in Jerusalem, Israel. *Transportation Research Record: Journal of the Transportation Research Board* 2246:16–23, DOI 10.3141/2246-03
- Pas EI, Sundar S, 1995. Intrapersonal variability in daily urban travel behavior: Some additional evidence. *Transportation* 22(2):135–150, DOI 10.1007/BF01099436
- Patterson DJ, Liao L, Fox D, Kautz H, 2003. Inferring High-Level Behavior from Low-Level Sensors. In: *UbiComp 2003: Ubiquitous Computing*, Springer, Berlin, DOI 10.1007/978-3-540-39653-6_6
- Pearson D, 2001. Global Positioning System (GPS) and travel surveys: Results from the 1997 Austin household survey. In: *8th Conference on the Application of Transportation Planning Methods*, Corpus Christi
- Prelicpean AC, Susilo YO, Gidofalvi G, 2017. A series of three case studies on the semi-automation of activity travel diary generation using smartphones. In: *DIVA*
- Reddy S, Mun M, Burke J, Estrin D, Hansen M, Srivastava M, 2010. Using Mobile Phones to Determine Transportation Modes. *ACM Transactions on Sensor Networks* 6(2):13:1–13:27, DOI 10.1145/1689239.1689243
- SBB Opendata, 2018. <https://opentransportdata.swiss/en/dataset/istdaten>
- Schlich R, Axhausen KW, 2003. Habitual travel behaviour: Evidence from a six-week travel diary. *Transportation* 30(1):13–36, DOI 10.1023/A:1021230507071
- Schuessler N, Axhausen KW, 2009. Processing Raw Data from Global Positioning Systems Without Additional Information. *Transportation Research Record: Journal of the Transportation Research Board* 2105:28–36, DOI 10.3141/2105-04
- Shin D, Aliaga D, Tunçer B, Arisona SM, Kim S, Zünd D, Schmitt G, 2015. Urban sensing: Using smartphones for transportation mode classification. *Computers, Environment and Urban Systems* 53:76–86, DOI 10.1016/j.compenvurbsys.2014.07.011
- Städtevergleich Mobilität 2015, 2018. https://www.stadt-zuerich.ch/ted/de/index/taz/publikationen_u_broschueren/staedtevergleich_mobilitaet_2015.html
- Stenneth L, Wolfson O, Yu PS, Xu B, 2011. Transportation Mode Detection Using Mobile Phones and GIS Information. In: *19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, DOI 10.1145/2093973.2093982

- Stopher P, Jiang Q, FitzGerald C, 2005. Processing GPS data from travel surveys. In: 28th Australasian Transport Research Forum, ATRF 05, Sydney, New South Wales, Australia
- Stopher P, Kockelman K, Greaves S, Clifford E, 2008. Reducing Burden and Sample Sizes in Multiday Household Travel Surveys. *Transportation Research Record: Journal of the Transportation Research Board* 2064:12–18, DOI 10.3141/2064-03
- Susilo Y, Axhausen KW, 2007. How firm are you? A study of the stability of individual activity-travel-location pattern using Herfindahl Index. In: 11th World Conference of Transport Research, Berkely, CA, USA
- Tsui S, Shalaby A, 2006. Enhanced System for Link and Mode Identification for Personal Travel Surveys Based on Global Positioning Systems. *Transportation Research Record: Journal of the Transportation Research Board* 1972:38–45, DOI 10.3141/1972-07
- Vij A, Shankari K, 2015. When is big data big enough? Implications of using GPS-based surveys for travel demand analysis. *Transportation Research Part C: Emerging Technologies* 56:446–462, DOI 10.1016/j.trc.2015.04.025
- Wolf J, Schoenfelder S, Samaga U, Oliveira M, Axhausen KW, 2004. Eighty Weeks of Global Positioning System Traces: Approaches to Enriching Trip Information. *Transportation Research Record: Journal of the Transportation Research Board* 1870:46–54, DOI 10.3141/1870-06
- Wu L, Yang B, Jing P, 2016. Travel Mode Detection Based on GPS Raw Data Collected by Smartphones: A Systematic Review of the Existing Methodologies. *Information* 7(4):67, DOI 10.3390/info7040067
- Zhang L, Dalyot S, Eggert D, Sester M, 2011. Multi-Stage Approach to Travel-Mode Segmentation and Classification of GPS Traces. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 3825:87–93, DOI 10.5194/isprsarchives-XXXVIII-4-W25-87-2011
- Zheng Y, Chen Y, Li Q, Xie X, Ma WY, 2010. Understanding Transportation Modes Based on GPS Data for Web Applications. *ACM Trans Web* 4:1–36, DOI 10.1145/1658373.1658374
- Zhu Q, Zhu M, Li M, Fu M, Huang Z, Gan Q, Zhou Z, 2016. Identifying Transportation Modes from Raw GPS Data. In: *Social Computing, Springer, Singapore, Communications in Computer and Information Science*, pp 395–409, DOI 10.1007/978-981-10-2053-7_35