Assessing the Influence of Preprocessing Methods on Raw GPS-Data for Automated Change Point Detection

Tomas Thalmann and Amin Abdalla

Abstract The automated recognition of transport modes from GPS data is a problem that has received a lot of attention from academia and industry. There is a comprehensive body of literature discussing algorithms and methods to find the right segments using mainly velocity-, acceleration- and accuracy-values. Less work is dedicated to the derivation of those variables. The goal of this chapter is to identify the most efficient way to preprocess GPS trajectory data for automated change-point (i.e., the points indicating a change in transportation mode) detection. Therefore the influence of different kernel based smoothing methods as well as an alternative velocity derivation method on the overall segmentation process is analyzed and assessed.

1 Introduction

GPS sensors have become a standard feature of modern mobile devices, such as smart phones or tablet computers. Consequently, a plethora of tracking applications have been developed to monitor user movement. The output is mostly a list of points with time-stamps, and therefore not very conclusive. Thus, deducing transportation modes from raw GPS-tracks, i.e., the determination of walking, biking, driving (car) or public transport can add considerable value to it. It can, for example, improve the availability of data for transport studies, where such information is still mainly gathered through hand written diaries. The idea of automated recognition of transport modes is not new, and has been a topic of interest in various research communities (Li et al. 2010; Reddy et al. 2010; Stopher et al. 2008).

T. Thalmann (✉) · A. Abdalla
Research Group Geoinformation, Department of Geodesy and Geoinformation,
Vienna University of Technology, Vienna, Austria
e-mail: tthalmann@gmail.com

A. Abdalla
e-mail: abdalla@geoinfo.tuwien.ac.at

J. Huerta et al. (eds.), *Connecting a Digital Europe Through Location and Place*,
Lecture Notes in Geoinformation and Cartography, DOI: 10.1007/978-3-319-03611-3_8,

A great part of the literature, though, is dedicated to the development and assessment of algorithms and methods to detect change points, i.e., those points in the list that represent changes in transportation means. Interestingly, the values used for the segmentation process remain unquestioned. Neither the derivation of speed from the raw GPS data, nor the potential smoothing of the data to reduce the influence of positional errors were subject of discussion.

The remainder of the work is structured as follows: In Sect. 2 we discuss relevant literature and methods used to deduct change points. Section 3 gives a detailed account of methods used. Sections 4 and 5 present the result of the analysis and the last section concludes with a set of recommendations derived from the findings.

2 Background and Related Literature

Liao et al. (2007) proposed a method to categorize and predict the movement and transportation behavior of individuals using GPS data. Their methodology is based on machine learning techniques and therefore require training data-sets that are specific to a user. This work will focus its attention on methods that are user independent, hence no user history is required for the process of segmentation. Such methods usually rely on a mix of GPS, acceleration, GSM and WIFI data. The main task is to assign a transportation mode for each point in a given GPS trajectory. This classification process consists of three steps:

1. Select and extract sensor measurements and metrics (features or descriptors) from the data.
2. Select a classifier and calculate the required thresholds and parameters.
3. Run tests to qualitatively evaluate the classification result.

As pointed out by Li et al. (2010), mobile phones have become multisensor-systems with great potential. Of course the features/observations from point 1 heavily depend on the chosen sensor(s). By using GPS and Accelerator data, for example, Reddy et al. (2010) achieved a 93 % classification accuracy. Unfortunately, they did not distinguish between the various modes of motorized transportation. The research of Chen and Bierlaire (2013) has focused on the usage of all sensors available on a Smart-phone and relies on a mix of GPS, Accelerator, Blueooth, GSM and WIFI data.

Ogle et al. (2002) draw attention to the fact that GPS-data is subject to systematic and random errors that coarsely depend on the geometric constellation of the receiver and the satellites in the field of view, as well as on atmospheric and clock influences (Hoffmann-Wellenhof et al. 2001). One have to take GPS accuracy into account when calculating and selecting features for transportation mode classification and derivation of other motion patterns.

Laube et al. (2007) identify four levels of analysis, respectively scale of feature calculation: *instantaneous* (local), *interval* (focal), *episodal* (zonal) and *total* (global). In consideration of the fact, that every track-point-location can contain error-components from GPS-measurements, it is assumed that interval-based feature calculation is able to heavily reduce such error influences compared to instantaneous calculation. Therefore researcher have proposed to use a moving interval to calculate basic features such as velocity or acceleration (Gómez and Valdés 2011; Cimon and Wisdom 2004).

Other approaches use statistical smoothing methods to reduce GPS error influences. E.g. Giremus et al. (2007) propose a particle filter to detect and reduce multipath errors and Wann and Chen (2002) investigate Kalman filtering with an additional post-Kalman-smoothing for velocity. Jun et al. (2006) compare Kalman filtering, least squares spline approximation, kernel-based smoothing and an adopted version of Kalman filtering in respect of velocity and acceleration profiles.

Dodge et al. (2008) categorizes the features derived from a generic trajectory as *primitive parameters* (position, time-stamp or interval), *primary derivatives* (distance, direction, duration or velocity) or *secondary derivatives* (change of direction, sinuosity or acceleration).

According to Zheng et al. (2010) the overall goal of an untrained method is to find features, which are not affected by differing traffic conditions, thus better reflect the transportation mode of a user. According to Zheng et al. (2010) these are Heading-Change-Rate, Stop-Rate and Velocity-Change-Rate. A classification algorithm then takes the descriptors (or features) and assigns one of the predefined classes.

As mentioned before the classification systems need parameters and thresholds. They can be determined either by supervised learning or by empirical examination. The current work chose an untrained method proposed by Zheng et al. (2010) that uses GPS data only. They state:

- Our method is independent of other sensor data like GSM signal and heart rate, and map information, for example, road networks and bus stops, etc. Thus, it is generic to be deployed in a broad range of Web applications.
- The model learned from the data-set of some users can be applied to infer GPS data from others; that is, it is not a user-specific model.

Similar to Stopher et al. (2008), their approach is a clear and descriptive two-step algorithm: (1) a change-point based track segmentation; (2) determination of transportation mode per segment.

Interestingly, all of the investigated methods proposed in literature start with existent velocity and acceleration data. The deriving process of those variables from a raw GPS track, or the effect of different methods for calculating speed and acceleration is not discussed. On the other hand approaches for trajectory analysis rarely deal with real data and the problems of uncertainty and sampling that come along with real data (Laube and Purves 2011). The influence of different analysis on practical applications like transportation mode for instance has not been extensively addressed in literature. This work shows that preprocessing of GPS-track data has a considerable effect on the overall performance of common classification methods. The assumption is that the results of such segmentation and classification processes can be improved by appropriate preprocessing.

3 Preprocessing of Raw GPS Data

This section discusses three approaches to preprocess raw GPS data, to reduce positional error influence on velocity estimations: (1) GPS-Point Accuracy (2) Smoothing Filters and (3) Velocity derivation.

It has to be stated, that most of the GPS-capable mobile phones (iOS and Android) already use built-in, model-based filters such as Kalman filters. So in fact the data used in this work already have been partially smoothed.

3.1 GPS Accuracy

The accuracy of the GPS position measurement mainly depends on the geometric constellation of the receiver and the available satellites. This is described by the Dilution of Precision (DOP), especially the horizontal DOP (HDOP) values. Widely spread satellites cause a more accurate position fix and a smaller DOP value. Since 4 satellites are necessary to fix a position it is considered to be advantageous to sort out the track points that have less than 4 satellites and a DOP higher than 3 (Stopher et al. 2005). The biggest advantage of this method is its simplicity, but it comes with drawbacks, such as the reduction of points in the GPS trajectory (see Sect. 4.2 for further discussion of this issue).

3.2 Velocity and Acceleration Derivation

If GPS data does not contain velocity values from Doppler-measurements the values have to be calculated somehow. A naïve approach, as done by Zheng et al. (2010) for example, is to simply calculate the velocity using distance and time difference from two consecutive points. Considering GPS-errors, the distance d_{AB} can contain large error components resulting from the positional errors of point A and B. The influence of such errors is smaller, the greater the distances get. To reduce the estimation error, Gómez and Valdés (2011) proposed the use of the n-th track point TP_{n+1} instead of the immediate follower $TP_{(n)}$ to calculate velocity value v_n (see Eq. 1).

$$v_n = \frac{d\left(TP_{(n+i)}, TP_n\right)}{\delta t\left(TP_{(n+i)}, TP_n\right)} \tag{1}$$

We will call this a point-based interval of length i. This approach works fine, as long the track follows a relatively straight line, but in reality the trajectory rarely is. Using longer intervals leads to an underestimation of the distance, because the air-line distance could be considerably shorter than the trajectory, especially in urban areas. This issue is known and one possible solution would be to matching GPS trajectories to external map data (Li et al. 2013).

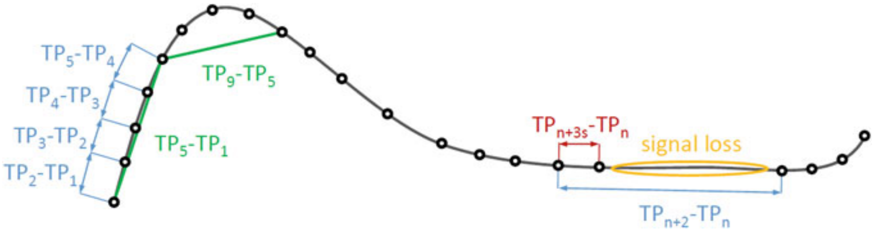


Fig. 1 Trajectory and different intervals for calculating average velocity; point based intervals (Eq. 1) in *blue* and *green*, time based interval (Eq. 2) in *red*

In addition, location observations are not equidistant, as it is assumed in Fig. 1. Observations are only delivered by the sensor in case a GPS signal is available and a position fix obtainable. So time-periods between location-fixes vary. Even if a position fix is delivered constantly every second, the distance between two trackpoints depends on the current speed. The spatial density of observations strongly varies, and so does the length of the temporal interval. Furthermore, the periods of signal loss can become very long when traveling on the subway. It is not desirable that points at the end of a subway segment skew the velocity derived by them and those before the segment. The use of wide point based-intervals increases the number of erroneous velocity calculation, e.g., if the interval length is set to 20 points, the velocity at 19 points before the gap will be skewed by the error. To avoid such problems, we propose to use time-fixed intervals rather than point-based ones. Eq. 1 becomes:

$$v_t = \frac{d\left(TP_{(t+\delta t)}, TP_t\right)}{\delta t\left(TP_{(t+\delta t)}, TP_t\right)} \tag{2}$$

With Eq. 2 we are able to retain the benefits of the point based-intervals and address the problem of data gaps to reduce the negative impact of signal loss. This can lead to considerable improvement for the analysis of GPS-tracks, especially in cities with a subway-system. The result of Eq. 2 is illustrated in Fig. 2 by the gray curve. It can be seen that the maximum of this curve moved to the left, so that it is now at the same time-stamp as the acceleration began in the original data in blue. By using the adjusted Eq. 3 unwanted time-shifts caused by Eq. 2 are eliminated.

$$v_t = \frac{d\left(TP_{(t-\delta t/2)}, TP_{(t+\delta t/2)}\right)}{\delta t\left(TP_{(t-\delta t/2)}, TP_{(t+\delta t/2)}\right).} \tag{3}$$

3.3 Smoothing Filters

The third possibility to reduce the influence of positional errors on a velocity profile is statistical smoothing. The goal is to smooth a signal, respectively applying a low-pass-filter. The operation applied is a kernel-weighted average on a sliding

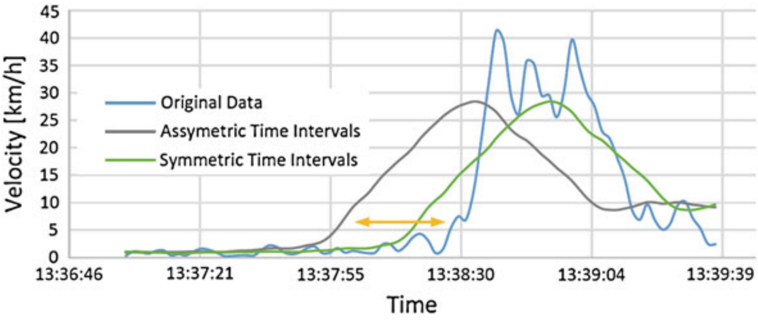


Fig. 2 Asymmetric time intervals of Eq. 2 show an unwanted time-shift, which can be eliminated with symmetric time intervals of Eq. 3

window of size h. Following a similar consideration which lead to time-based intervals in Sect. 3.2, we take a constant bandwidth, e.g. $h = 3\,\mathrm{s}$, instead of a K-Nearest-Neighbor-Bandwidth.

The Nadaraya-Watson kernel smoothing algorithm defined in Eqs. 4 and 5 (Hastie et al. 2009) is used:

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_h(x_0, x_i) y_i}{\sum_{i=1}^{N} K_h(x_0, x_i)} \tag{4}$$

$$K_h(x_0, x) = D\left(\frac{|x - x_0|}{h(x_0)}\right) \tag{5}$$

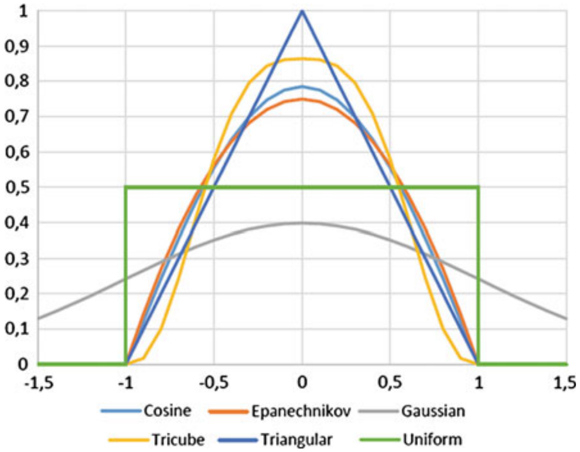
$D(t)$ determines the shape of the weighting function. The most popular kernels can be seen in Fig. 3. After visual investigation of differing weighting functions a Tricube-kernel seemed to bear the best results. It is the kernel with the steepest shape except from the Uniform-kernel, and would therefore maintain walk-stop-changes or generally short-time changes of velocity more precisely.

It is defined by:

$$D(t) = \begin{cases} 70/81\left(1 - |t|^3\right)^3, & \text{if } t \leq 1 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

There are mainly two possibilities to apply a kernel smoothing method on GPS-Tracks, namely on the velocity values itself, or on the positions prior to the velocity calculation. Preceding experiments showed that the velocity-smoothing does not bring any notable benefit, so in the coming experiments only position smoothing is applied to the data. Thus, the kernel smoothing is applied on the locations only, respectively on the Latitude/Longitude pairs of the track, which leads to a translation of every track point depending on the surrounding points. Afterwards the velocity is calculated as described in Sect. 3.2.

Fig. 3 Most popular kernels



4 Assessing the Preprocessing Methods

In this section a visual analysis of the effect of the three discussed preprocessing methods is used to discuss their effectiveness. It shows that presumably a combination of two will achieve the best results.

4.1 Test Data

The experiments and results in this chapter are based on a data set consisting of 14,260 track points that were tracked over 8 h and 26 min. A total of 40 Change-Points, i.e., the points indicating a change in transportation mode, were manually determined and the track was subsequently segmented into transport types of Walk, Bike, Tram, Bus, Car, Train and Subway. The tracks have been recorded by 4 persons with 3 different Android phones (Sony Xperia Z, HTC One X and Motorola Defy) and the free App GPSLogger.[1] The tracking took place between 7 am and 11 pm on varying days in June 2013 and are spread over the urban area of the city of Vienna, Austria (Fig. 4).

The App delivers .GPX-Files of track points containing Locations, time-stamp, HDOP, Number of satellites and Velocity from Doppler-measurements. The files were imported to a SQL-database for the experiments corresponding to the formal definition of a geo-spatial lifeline from Hornsby and Egenhofer (2002): A list of ordered GPS-track point corresponds to a list of space-time observations of the form `<ID, Location, Time>`, where `ID` is a unique identifier, `Location` is a spatial coordinate pair and `Time` is a sequential time-stamp. Other measurements like HDOP or Doppler-velocity is linked to a track point via the `ID`.

[1] https://play.google.com/store/apps/details?id=com.mendhak.gpslogger

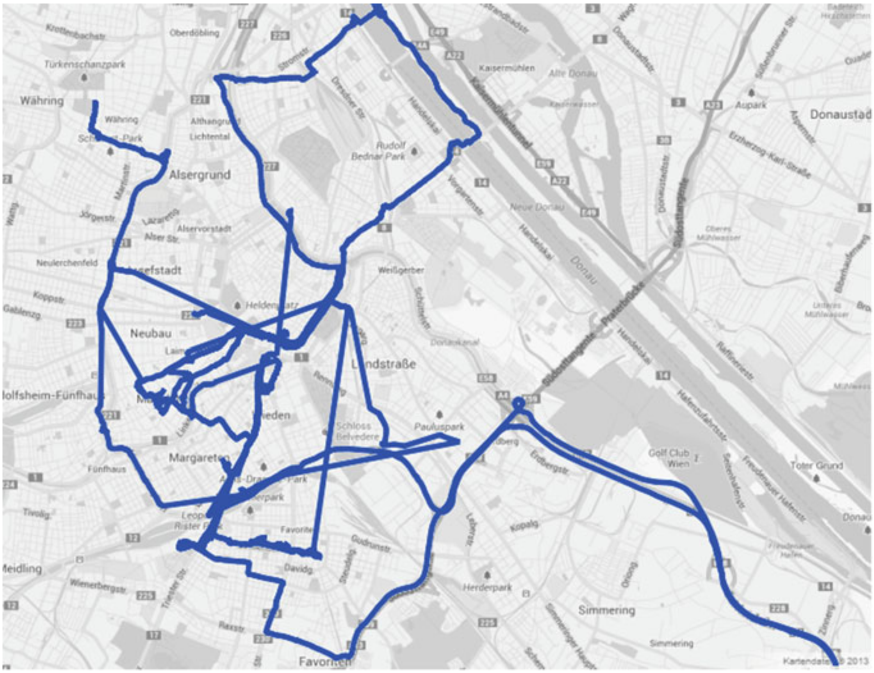


Fig. 4 Distribution of the testdata in the urban area of Vienna

4.2 Visual Investigation

To investigate the effect of the methods, we apply each of it to the raw GPS data separately, from which we then derive velocity based on point-based intervals from Eq. 2. First, we look at filtering out points based on the HDOP and satellite count, as discussed in Sect. 3.1. Figure 5 shows that some of the peaks in velocity, caused by inaccurate GPS-fixes were sorted out. On the other hand it expands some of the periods without positional fixes. Such gaps in the track can make change point-detection more difficult or in worst case impossible.

The effect of the smoothing filter (Sect. 3.3) can be seen in Fig. 6. The velocity is calculated from the smoothed positions with an interval of 1 point. The reader should note that the unfiltered data is still very noisy ($i = 1\text{point},\ h = 0\,\text{s}$). With higher smoothing parameters the data gets smoother. Subsequently, the difference between walk and stop segment becomes less recognizable. On the other hand the peak at around 13:36:05, a result of GPS-position-error, is partly removed from the graph by using a 20 s smoothing parameter. While removing the mentioned peak is desirable, it is also crucial to maintain the distinguishing properties of walk and stop segments. Finally, Fig. 7 illustrates the improvement that can be achieved by using the temporal interval based velocity derivation in Eq. 3.

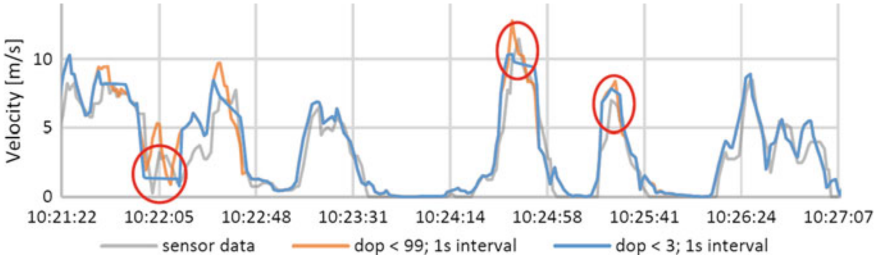


Fig. 5 Effects of a DOP-filter on the *velocity curve*

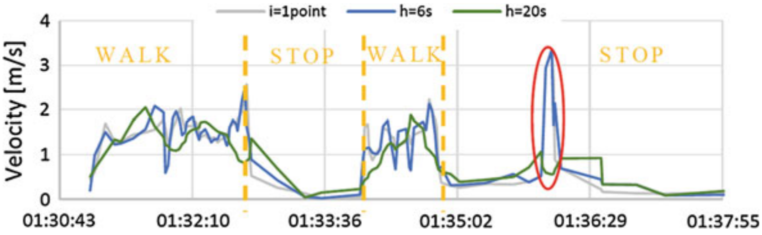


Fig. 6 Different smoothing parameters h of kernel smoothing on positions

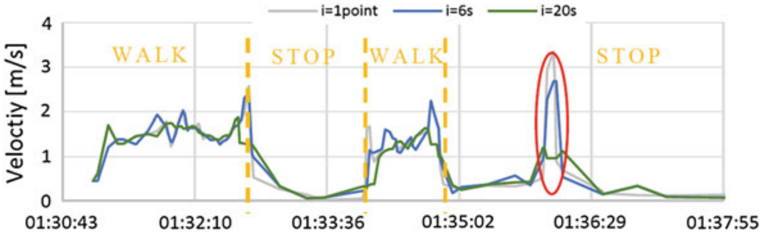


Fig. 7 Different temporal intervals i

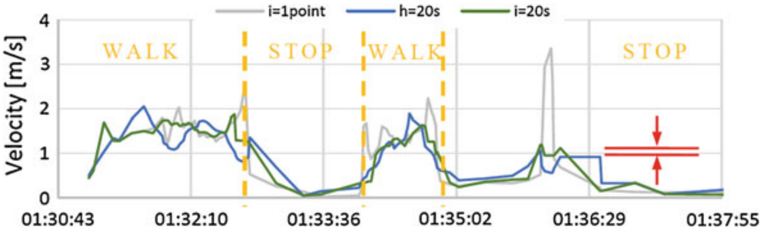


Fig. 8 Comparison of standalone temporal intervals and standalone kernel smoothing

The influence of a longer interval i is similar to that of a higher smoothing parameter h. With the interval-method the difference between walk and stop segment is better preserved than with the kernel smoothing. On the other hand the GPS-error is not considerably lowered (see Fig. 8).

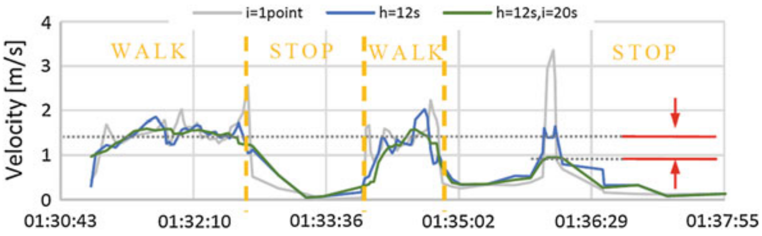


Fig. 9 Reduction of GPS-errors through the combined preprocessing methods

Each of the introduced methods has its pros and cons, but based on visual analysis, we infer that the best results might be achieved by a combination of more than one. Thus, in the following we investigate the effect of using combinations of methods, i.e., the temporal interval-method (see Sect. 3.3) applied to position-smoothed data using a TriCube-kernel. The goal is to reduce the amplitude of short-time peaks usually resulting from positional inaccuracies. In other words, sudden or unfeasible increases in speed should be filtered, such that a classification algorithm does not split a segment at that point. With a combined approach of both methods the maximum of the short-time peak of the GPS error can be downshifted below the average velocity of walk segments. This is necessary for the algorithm to be able to correctly detect walk segments and ignore errors (see Fig. 9).

In conclusion, it is assumed that the best performance can be achieved by a combination of a temporal interval-based-calculation and an applied kernel filter.

5 Change Point Detection and Segmentation Assessment

This section will test the performance of a well-known segmentation-algorithm proposed by Zheng et al. (2010) on various preprocessed GPS data sets. We will show that, as asserted in the previous section, a combination of temporal interval based velocity derivation and a kernel filter results in improved performance of the exemplary algorithm.

5.1 Changepoint Detection

The algorithm used for this study is taken from Zheng et al. (2010) and searches for track points, at which the testimonial changed transportation mode. Such points are called change-points and partition the track into segments. The classification of transportation mode per segment is conducted in a second step.

The algorithm is based on the fact that around 99 % of transportation mode changes happen with an interjacent walk segment. So at first the track is split up into alternating walk and non-walk segments, which greatly reduces the complexity of the segmentation.

- *Step 1*: Using a loose upper bound of velocity (Vt) and that of acceleration (at) to distinguish all possible Walk Points from Non-Walk Points.
- *Step 2*: If the distance or time span of a segment composed by consecutive Walk Points or Non-Walk Points less than a threshold, merge the segment into its backward segment.
- *Step 3*: If the length of a segment exceeds a certain threshold, the segment is regarded as a certain segment. Otherwise it is deemed as an uncertain segment. If the number of consecutive uncertain segments exceeds a certain threshold, these uncertain segments will be merged into one Non-Walk segment.
- *Step 4*: The start point and end point of each Walk segment are potential change points, which are used to partition a trip.

5.2 Assessment Methodology

As mentioned in the last section, the algorithm requires a velocity upper bound u, an acceleration upper bound a, a minimal segment length d and a minimal certain segment length (in this work fixed as 1.5-times minimal segment length) as input. We call such a set of input parameters (the configuration of the algorithm): $p_i \in P = U \times A \times D$.

For our assessment we first apply the differing preprocessing methods described in Sect. 3 (with varying values for time interval $i\ elem\ I$, smoothing parameter $h \in H$, and upper DOP-bound $o \in O$). We then have a set of outputs defined by their input $v \in V$ with $V = I \times H \times O$. Then the differing parameter configurations $p_1 \ldots p_n \in P$ are applied to every output produced by the preprocessing parameters $v \in V$. Finally, for each of the following quality measures a matrix was generated:

- Correct CPs
- Missed CPs
- Wrong detected CPs

A CP is deemed *correct*, if it lies within a distance-threshold of 100 m; *missed* if a real-world-CP is not detected and *wrong* if a detected CP does not exist in real world.

To calculate an overall measure for CP recognition for each (p_i, v_j) pair, the number of correct and missed CPs have to be taken into account as well as the number of wrong CPs. For this purpose the concepts of recall and precision, as done in other work (Olson and Delen 2008), are facilitated:

$$Recall = \frac{(correct\ CPs)}{(total\ real\ world\ CPs)} \tag{7}$$

$$Precision = \frac{(correct\ CPs)}{(correct\ CPs + wrong\ CPs)} \tag{8}$$

For the evaluation of change-point detection, respectively segmentation the recall is viewed to be more important since false positive (wrong) CPs can be corrected

Table 1 Effect of an HDOP filter

	Recall	Precision
$o = 999$	0.89	0.240
$o = 3$	0.875	0.288

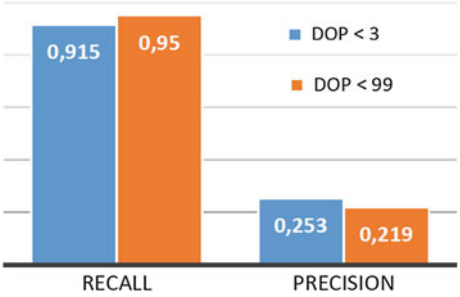


Fig. 10 Overall results of an HDOP filter

afterwards, i.e., if the two adjacent segments are of the same transportation mode. A output-version is calculated by a pair of temporal interval length i and a smoothing parameter h for a given parameter setting p out of the parameter set P. So, the value of a cell is the maximum recall of $v \in V$, which has been achieved by the specified $p \in P$. A matrix is generated for recall and precision.

5.3 Comparison and Assessment

At first we have applied our set P on the original data calculated from the point based-interval of length 1 (v_0) and searched for the parameter $p_{(0,optimal)} = (u = 2.3\,\mathrm{m/s}, a = 0.2\,\mathrm{m/s}^2, d = 15\,\mathrm{m})$ achieving the best result.

The same parameter configuration $p_{(0,optimal)}$ is then applied on a reduced data set, where all track points with a HDOP higher than 3 are sorted out ($o = 3$), using the point based velocity derivation of interval 1 (Table 1):

The HDOP filter causes a reduction in terms of recall, while the precision improved. Figure 10 shows the average recall and precision for versions with HDOP filter and the average recall and precision for versions without HDOP filter. It suggests that HDOP based filtering decreases recall ($-3.5\,\%$) but positively affects the precision ($+3.5\,\%$) of change-point detection. We suppose that this effect is caused by the increasing number and length of gaps in the tracks. Since we are interested in high recalls the set of preprocessing, parameter versions V degrades to $V = I \times H$, in other words we do not filter out points of high uncertainty. Table 2 shows the recalls of the configuration $p_{(0,optimal)}$ for varying point count-based intervals compared to the recalls of $p_{(0,optimal)}$ for varying duration-based interval data (The interval length is in points for the point based intervals and in seconds for the temporal intervals).

Table 2 Recalls from point-count intervals and temporal intervals with varying interval length

Length	Point-count	Temporal
$i = 1$	0.89	0.85
$i = 6$	0.89	0.90
$i = 10$	0.84	0.925
$i = 16$	0.815	0.875
$i = 22$	0.815	0.90

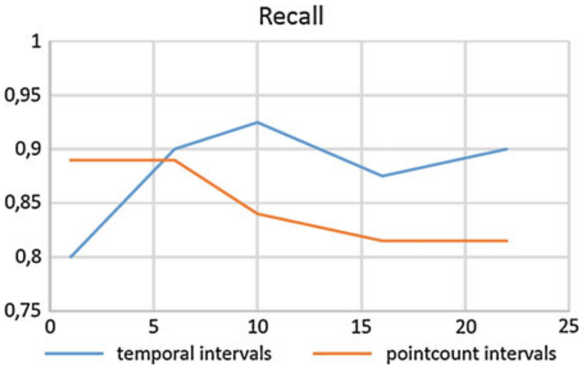


Fig. 11 Recalls from point-count intervals and temporal intervals with varying interval length

Table 3 Recalls of the combined preprocessing methods with the parameter set $p_{(0,optimal)}$

Recall	$h = 0\,\mathrm{s}$	$h = 5\,\mathrm{s}$	$h = 8\,\mathrm{s}$	$h = 11\,\mathrm{s}$	$h = 14\,\mathrm{s}$	
$i = 1$	0.85	0.85	0.875	0.925	0.9	0.88
$i = 6$	0.9	0.875	0.925	0.925	0.9	0.905
$i = 10$	0.925	0.875	0.95	0.925	0.9	0.915
$i = 16$	0.875	0.85	0.925	0.9	0.9	0.89
$i = 22$	0.9	0.9	0.85	0.875	0.875	0.88
	0.89	0.87	0.905	0.91	0.895	Avg

The point-based interval recall values decline with increasing interval length, due to the lack of appropriate signal loss handling. The algorithm is unable to detect subway-segments because of the effect described in Sect. 3.2. On the other hand the temporal intervals are able to handle those subway segments and reduce the influence of GPS errors. Thus the results improve with increasing interval length (see Fig. 11).

The results prove that the proposed method of duration-based intervals improve change-point detection of tracks where there are periods of sustained signal loss. As stated earlier, the assumption is that a combined method of position smoothing and temporal intervals will perform best. To test the assumption, the various versions of preprocessed data, along with the optimal configuration $p_{(0,optimal)}$ found for the raw data set, is used as an input for the algorithm. The results are shown in Table 3.

Table 4 Recalls of the combined preprocessing methods with the optimized parameter set $p_{(i,optimal)}$

Recall	$h = 0\,\mathrm{s}$	$h = 5\,\mathrm{s}$	$h = 8\,\mathrm{s}$	$h = 11\,\mathrm{s}$	$h = 14\,\mathrm{s}$	
$i = 1$	0.95	0.975	1.0	1.0	1.0	0.985
$i = 6$	0.975	0.975	1.0	1.0	1.0	0.99
$i = 10$	0.95	0.975	1.0	1.0	1.0	0.985
$i = 16$	0.95	0.95	0.95	0.925	0.925	0.94
$i = 22$	0.925	0.9	0.95	0.9	0.875	0.91
	0.95	0.955	0.98	0.965	0.96	Avg

Table 5 Precisions of the combined preprocessing methods with the optimized parameter set $p_{(i,optimal)}$

Precision	$h = 0\,\mathrm{s}$	$h = 5\,\mathrm{s}$	$h = 8\,\mathrm{s}$	$h = 11\,\mathrm{s}$	$h = 14\,\mathrm{s}$	
$i = 1$	0.257	0.201	0.204	0.200	0.206	0.214
$i = 6$	0.192	0.202	0.225	0.247	0.211	0.215
$i = 10$	0.307	0.227	0.25	0.263	0.244	0.258
$i = 16$	0.288	0.222	0.221	0.242	0.266	0.248
$i = 22$	0.291	0.356	0.240	0.234	0.313	0.287
	0.267	0.242	0.228	0.237	0.248	Avg

The first column ($h = 0\,\mathrm{s}$) contains the standalone interval results, hence no position smoothing filter was applied beforehand. It can be observed, that the additional application of a kernel smoother leads to improvement in recall. Comparing the results to the point based recall values presented in Table 3, a maximum increase in recall by 6 % can be found ($i = 10\,\mathrm{s}$, $h = 8\,\mathrm{s}$).

So far we have tested all the versions with the parameter set $p_{(0,optimal)}$ found from the original point based interval data. Since the velocity curves from the different preprocessing versions V have altered characteristics, it seems likely that the optimal parameter set $p_{(i,optimal)}$ of a version v_i is not the same as $p_{(0,optimal)}$. Therefore the whole set P has been applied on all produced version using the preprocessing parameters V, to find the optimal algorithm parameter setting for the preprocessed data. Tables 4 and 5 show the results for recall and precision with the optimal $p_{(i,optimal)}$ for the preprocessed data.

The observable trend in precision shows an improvement with longer intervals and decrease with the additional kernel smoothing. The recall of the combined method shows improvements over the stand-alone interval method in column 1 ($h = 0\,\mathrm{s}$) as well as over the stand-alone kernel-smoothing method in row 1 ($i = 1\,\mathrm{s}$). The system was able to correctly detect all change-points. By using the same parameter set $p_{(0,optimal)}$ the original results from the point based-intervals were enhanced by the proposed preprocessing methods by 6 % (Table 6).

Table 6 Improvement by preprocessing with the same parameter set $p_{(0,optimal)}$

	$(v_0, p_{(0,optimal)})$	$(v_{13}, p_{(0,optimal)})$	Improvement
Recall	0.89	0.95	+6 %
Precision	0.240	0.316	+7.6 %

Table 7 Improvement by preprocessing with the optimal parameter set $p_{(i,optimal)}$

	$(v_0, p_{(0,optimal)})$	$(v_{14}, p_{(14,optimal)})$	Improvement
Recall	0.89	1.0	+11 %
Precision	0.240	0.263	+2.3%

Optimizing the parameter setting for various preprocessed versions of the data set resulted in a maximum of 11 % increase in recall, effectively resulting in recall values of 100 % for some versions (Table 7).

6 Conclusion and Recommendations

Our results showed that the preprocessing of raw GPS-data to obtain velocity data plays a considerable role in the process of automated transport mode derivation. Literature about algorithms and methods has, to our knowledge, not addressed the issue sufficiently. The chapter introduces and discusses various methods that can potentially improve the velocity derivation process from raw GPS-data. It proved their efficiency by comparing results of a well-known untrained method applied to a data set before and after it was preprocessed by the proposed methods.

To quantify the enhancement the recall of change-points, i.e., the points that mark a change in transportation mode was used as a measure. With the application of kernel smoothing on track point positions we were able to increase the recall of the investigated algorithm by a few percent. By using temporal interval based velocity derivation in addition to the position smoothing, an improvement of 6 % in recall was achieved. By optimizing the algorithm's parameters a maximum of 11 % improvement was reached, effectively resulting in a 100 % recall value in some cases. Thus, the application of each of the methods alone is not sufficient; rather a combination of them is necessary to achieve optimal results. The applied methods, on the other hand, caused a decrease in precision, i.e., increase of false change-points in addition to the correct ones. Since false positive change-points can be filtered out afterwards, e.g., if the two adjacent segments are of the same transportation mode, we deemed the recall to be a more important quality measure.

The chapter showed that the input parameters and the interval lengths of the smoothing window and velocity derivation, i.e., preprocessing, plays a role for the results of the algorithm. Further optimization of the algorithm's input parameters was only possible because of the available ground-truth data set, hence the model was fostered to that data set. Future research will need to investigate whether the parameters found to work best in this model are generally applicable for other data sets or, if not, how they can be estimated.

In general, we assume that the proposed preprocessing methods will have similar enhancement effects on other algorithms that make use of velocity as a primary classification determinant. While the work has focused on change-point detection there was no thorough testing of the overall improvement for the actual transportation mode classification, something open to future analysis.

References

Chen J, Bierlaire M, Flötteröd G (2011) Probabilistic multi-modal map matching with rich smartphone data. In: Proceedings of the Swiss Transport Research Conference (STRC), Switzerland, 11–13 May 2011

Cimon NJ Wisdom MJ (2004) Accurate velocity estimates from inaccurate GPS data. In: Proceedings of the tenth forest service remote sensing applications conference

Dodge S, Weibel R, Lautenschütz AK (2008) Towards a taxonomy of movement patterns. Inf Visual 7(3–4):240–252

Giremus A, Tourneret JY, Calmettes V (2007) A particle filtering approach for joint detection/estimation of multipath effects on gps measurements. IEEE Trans Signal Proc 55(4):1275–1285

Gómez-Torres NR, Valdés-Díaz DM, (2011) GPS capable mobile phones to gather traffic data. In: Ninth LACCEI Latin American and Caribbean conference (LACCEI, (2011) engineering for a smart planet, innovation, information technology and computational tools for sustainable development, medelln, Colombia

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning. Springer, New York

Hoffmann-Wellenhof B, Lichtenegger H, Collins J (2001) GPS: theory and practice. Springer, New York

Hornsby K, Egenhofer MJ (2002) Modeling moving objects over multiple granularities. Ann Math Artif Intell 36(1–2):177–194

Jun J, Guensler R, Ogle JH (2006) Smoothing methods to minimize impact of global positioning system random error on travel distance, speed, and acceleration profile estimates. Transp Res Rec J Transp Res Board 1972(1):141–150

Laube P, Dennis T, Forer P, Walker M (2007) Movement beyond the snapshot—dynamic analysis of geospatial lifelines. Comput Environ Urban Syst 31(5):481–501

Laube P, Purves RS (2011) How fast is a cow? cross-scale analysis of movement data. Trans GIS 15(3):401–418

Li X, Ortiz PJ, Browne J, Franklin D, Oliver JY, Geyer R, Chong FT (2010) Smartphone evolution and reuse: establishing a more sustainable model. In: IEEE 39th international conference on parallel processing workshops (ICPPW) 2010, pp 476–484

Li Y, Huang Q, Kerber M, Zhang L, Guibas L (2013) Large-scale joint map matching of GPS traces. In: Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems, ACM, pp 214–223

Liao L, Patterson DJ, Fox D, Kautz H (2007) Learning and inferring transportation routines. Artif Intell 171(5):311–331

Ogle J, Guensler R, Bachman W, Koutsak M, Wolf J (2002) Accuracy of global positioning system for determining driver performance parameters. Transp Res Rec J Transp Res Board 1818(1): 12–24

Olson DL, Delen D (2008) Advanced data mining techniques [electronic resource]. Springer, Berlin

Reddy S, Mun M, Burke J, Estrin D, Hansen M, Srivastava M (2010) Using mobile phones to determine transportation modes. ACM Trans Sens Netw (TOSN) 6(2):13

Stopher PR, Clifford E, Zhang J, FitzGerald C (2008) Deducing mode and purpose from GPS data. Institute of Transport and Logistics Studies

Stopher PR, Jiang Q, FitzGerald C (2005) Processing GPS data from travel surveys. In: 2nd international colloqium on the behavioural foundations of integrated land-use and transportation models: frameworks, models and applications, Toronto

Wann CD, and Chen YM (2002) Position tracking and velocity estimation for mobile positioning systems. In: The IEEE 5th international symposium on wireless personal multimedia communications, vol 1. pp 310–314

Zheng Y, Chen Y, Li Q, Xie X, Ma WY (2010) Understanding transportation modes based on gps data for web applications. ACM Trans Web (TWEB) 4(1):1