Detecting Outliers in Cell Phone Data
Correcting Trajectories to Improve Traffic Modeling

Christopher Horn, Stefan Klampfl, Michael Cik, and Thomas Reiter

The use of cell phone signaling data for traffic modeling has great potential. Because of the high coverage rate of these phones, the data can be used as an addition to or even as a replacement for stationary sensors when the deployment of stationary sensors is not possible or too expensive. However, cell phone signaling data are error-prone and have to be preprocessed for use in traffic modeling. First, the positions reported by cell phone signaling data may be inaccurate. Second, because of privacy issues, additional data may be introduced to obfuscate actual movements. This study presents three filters to rectify the trajectories generated by cell phone movements. For evaluation, the filters were applied to cell phone trajectories and compared with corresponding GPS-based tracks. The evaluation data covered 4,933 tracks collected automatically and 5 tracks collected manually. The proposed filters significantly improved the estimation of speed and position compared with the raw trajectories of cell phone movements.

Currently several methods are available for traffic modeling. The primary data source for traffic state recognition is roadside detectors (e.g., loop detectors and overhead detectors), which can measure occupancy rates, volumes, and speeds at a specific position on a road. Such detectors do not allow the transportation planner to derive time-space trajectories because it is not possible to recognize a specific car at different detectors. Only measurements with automatic number plate recognition (ANPR) systems allow this recognition, but their usage for the purpose of transportation planning is often limited by high installation costs and data protection issues (1). As an alternative, the monitoring of cell phones (mobile phones) can provide floating phone data (FPD) for a large number of travelers in the network.

The use of cell phone data as traffic sensors has two main advantages. First, there is no need to buy and mount expensive traffic sensors. Because most drivers already carry a cell phone, it is reasonable to use those devices as sensors. Second, cell phone data sensors are independent of weather conditions.

But using cell phone data for this purpose also has significant disadvantages. The first problem is that, unlike GPS sensor data, cell phone events are sparse and do not occur frequently. Events are only triggered when a user initiates or receives a phone call or text message or when the user changes location area. Since location areas might cover hundreds of kilometers, it may happen that no update-event is sent from a given cell phone, although the user is driving in the area. The second problem is that cell phone events only contain the approximated geographic position of the user, which is significantly less exact than the GPS position. In urban areas with a high density of antennas, the position accuracy of cell phone devices is about 500 m; in rural and alpine regions of Austria, the position accuracy is about 2.5 km. The third problem is the phenomenon of so-called supersonic jumps (outliers), which are events that suddenly occur kilometers away (up to several hundreds of kilometers) within a short period of time. Although some of those jumps are system inherent for cell phones (they have a range of up to 35 km), some jumps might be triggered by external mechanisms. For example, arbitrary events such as temporary mobile subscriber identity (TMSI) might be injected to hamper the ability of tracing applications to use this kind of data. The TMSI is the local and limited number of a subscriber within a location area and might be used instead of the international mobile subscriber identity (IMSI) for the connection. The injection of arbitrary events was introduced to discourage the creation of movement profiles and thus protects the privacy of the participants (2).

To use cell phone data for accurate traffic modeling, these shortcomings must be considered and mechanisms need to be applied that rectify the trajectories. This paper presents three approaches (filters) to detect and remove outliers in trajectories to facilitate the usage of cell phone data for traffic modeling. The filters were applied to 4,933 trajectories and compared with a ground truth of corresponding GPS tracks. The results showed that the filters significantly improved the speed and position estimation compared with the raw trajectories.

LITERATURE REVIEW

The use of cell phone data for traffic modeling has gained huge attention in the past few years. Steenbrugge et al. provided a systematic overview of the main studies and projects that address the use of data derived from cell phone networks to obtain location and traffic estimations of individuals (3). Their overview was a starting point for further research on incident and traffic management. Sánchez et al. proposed several models to infer the number of moving vehicles on certain roads by means of cell phone data (4). They concluded that because of the error levels achieved in their methodology and the high computation costs, cell phone data should be seen as a complementary sensor when trying to accurately estimate traffic information in real time. Schlach was used FPD trajectories to analyze the route choice behavior of drivers (5). He used variable message signs that provide route recommendations. Maximum likelihood estimation was used to examine which factors influence

C. Horn and S. Klampfl, Know-Center GmbH, Inffeldgasse 13, 8010 Graz, Austria. M. Cik and T. Reiter, Institute of Highway Engineering and Transport Planning, Graz University of Technology, Rechbauerstrasse 12/2, 8010 Graz, Austria. Corresponding author: C. Horn, chorn@know-center.at.

Transportation Research Record: Journal of the Transportation Research Board, No. 2405, Transportation Research Board of the National Academies, Washington, D.C., 2014, pp. 49-56. DOI: 10.3141/2405-07
route choice. Thiagarajan used three systems to estimate the trajectories of cell phone movements: cellular tower sightings, GPS data, and Wi-Fi (a trademarked wireless networking technology) localizations (6). He also found that cell phone data are highly error-prone when it comes to accurate position localization.

To address the problem of noisy sensor data, Kalman filters have been successfully applied for traffic modeling. Frühwirt proposed the application of a linear Kalman filter for track filtering, track prediction, and track smoothing (7). He also described how to use the Kalman filter and the χ² test to detect outliers on a given track. Aydos et al. compared linear and unscented Kalman filter process models for urban traffic applications (8). They concluded that a linear model performs well when the measurements are dense. Wang and Papageorgiou used an extended Kalman filter to estimate the state of traffic on stretches of a freeway (9).

Research has also been done on the detection and removal of outliers from trajectories. Thiagarajan et al. introduced VTrack, a framework that uses cell phones to estimate traffic delays (10). The data are collected via the GPS interface and Wi-Fi localization of a cell phone. A simple speed-based filter (delete all points where speed > 200 mph) is used to remove outliers from the trajectories, which are then processed with a hidden Markov model to achieve map-matching and travel time estimations. Their filter is similar to the first proposed filter in the research reported here, although their filter is not applied recursively.

Many other approaches mostly focus on detecting trajectory outliers in a set of normal (non-outlying) trajectories. Picarielli et al. used supervised classification techniques (i.e., support vector machine) to detect outlying trajectories in a given set of training trajectories and evaluated their approach with a video analysis of a road segment where they detected forbidden U-turns (11). Li et al. used discrete pattern fragments called motifs to express trajectories (12). Lee et al. proposed a framework to detect outlying subtrajectories first by partitioning the trajectories in a set of subpartitions and second by detecting the outlying subpartitions based on a set of neighboring trajectories (13). Yong et al. introduced a trajectory outlier method called TOP-EYE, which aims to detect evolving outliers at a very early stage (14). However, most of those approaches relied on a set of normal (non-outlying) trajectories in order to detect the outlier.

The research reported here did not group similar trajectories, but only considered one single, isolated trajectory that may contain outlier positions. Hence, the other approaches were not applicable to tackle the problem of removing outliers from cell phone data. Further, most of the other approaches only considered spatial information and not temporal information. Considering the sparseness of events, the use of temporal information is crucial to detect outliers in cell phone trajectories. Finally, cell phone data do not provide constant observations, but sparse events mixed with potential bursts of events. This characteristic also has an impact on outlier detection.

METHODS

For this research, the largest Austrian mobile network operator, A1 Telekom Austria, provided access to the iTraffic data stream, which issues events received by the monitoring network of its infrastructure. Information for determining the geographical position of each cell phone is based on active event types in the cellular network (e.g., voice call connection build-up, a short written message, or use of the Internet). If the device is used passively, only the change from one location area to another and a periodic location request are saved as an event. Generally the anonymous identification number (ID) of the cell phone, the time stamp, the event type (e.g., voice call or short message), and the position (longitude and latitude) are provided in the iTraffic data stream. For privacy reasons, an anonymized ID (valid for 24 h) is transmitted instead of the phone number. Thus, the iTraffic data stream enabled the research to model the road network traffic based solely on cell phone data. The GPS position of the cell phone was available only when the provided navigation solution of the network provider was used. Otherwise, the estimated position (depending on the cell tower) was saved as the actual position. The daily iTraffic data stream covers 50 to 70 gigabytes of data.

There was a need to remove outliers from the cell phone trajectories in order to use cell phone data as sensors for traffic modeling. Figure 1 shows the visualization of the raw trajectories from users driving 150 km (93.2 mi) from Graz to Klagenfurt, Austria, based on the iTraffic data stream. In the figure, each color reflects a different user and the small phone icons indicate cell phone events (calls, texts, or Internet use). Most trajectories contain several outliers, which makes it almost impossible to calculate the accurate speed and position without preprocessing.

An outlier was defined as follows: Let $M$ be a set of sequentially occurring geospatial cell phone events. An event $M_i$ from this set is defined as an outlier if the calculated velocity between $M_{i-1}$ and $M_i$ is greater than $V_{\text{supersonic}}$, km/h. The term "supersonic jump" does not refer to actual supersonic speed, but describes jumps that are exceptionally fast. In the experiments, $V_{\text{supersonic}}$ was set to 250 km/h (155 mph), which is about twice the allowed driving speed. The following subsections will describe three filters that realign the trajectories of driving cell phone users by removing the outliers.

Recursive Naive Filter

The recursive naive filter simply removes any outliers from the sequential stream of cell phone events and connects the surrounding events. Algorithm 1 describes the filter in pseudocode. The input is a list of geospatial cell phone events. First, the list is sorted by time. Then, the list is iterated and the speed is calculated for all consecutive events $L_i[i-1]$ and $L_i[i]$. If this speed exceeds $V_{\text{supersonic}}$, the $L_i[i]$ is removed from the list. However, it might happen that two outliers occur consecutively and hence the second outlier is not removed. To remove such outliers as well, the filter is applied recursively: the filter is applied iteratively as long as it removes events. If the input list is equal to the output list, the filter stops.

Algorithm 1. Recursive naive filter
Input: list of location events $L$
Output: list of location events $L$ (outliers removed)

Sort $L$ by time of events
For $i = 0$ to $L$.size:
If $i > 0$:
\[ v = \text{distance}(L[i-1] \cdot \text{pos}, L[i] \cdot \text{pos})/(L[i] \cdot \text{time} - L[i-1] \cdot \text{time}) \]
If $v > V_{\text{supersonic}}$:
Remove $L[i]$
End if
End if
End for
Return $L$. 
The recursive naive filter does not tackle the following problem: Consider three consecutive events \(E_1, O_2\), and \(E_3\), in which \(E_1\) and \(E_3\) are regular events and \(O_2\) is an outlier. If there is a long time range between \(E_1\) and \(O_2\) and \(O_2\) appears closely before \(E_3\), the outlier is treated as a regular event (because the calculated speed from \(E_1\) to \(O_2\) is below the threshold \(V_{\text{supersonic}}\), while the correct event is treated as an outlier (the speed from \(O_2\) to \(E_3\) is below the threshold). Consequently, this filter erroneously removes the correct event. To overcome this issue, the recursive look-ahead filter is introduced.

### Recursive Look-Ahead Filter

The recursive look-ahead filter is a modification of the recursive naive filter. If an outlier is detected, the recursive look-ahead filter looks at the subsequent event in order to decide which of the previous two events is the outlier and which is the correct event. Consider the sequence of events \(E_1, O_2, E_3\), and \(E_4\), in which \(O_2\) appears close to \(E_3\). As described in the previous subsection, the recursive naive filter would incorrectly identify \(E_3\) as an outlying event. The recursive look-ahead filter looks at the subsequent event \(E_4\) and measures the geographic distance between \(E_3\) and \(E_4\), and \(E_2\) and \(E_3\). The filter removes the event with the larger distance to \(E_4\).

Algorithm 2 describes how the filter works. The input for the filter is a list of occurrence cell phone events, in which each event has a position (latitude and longitude) and a timestamp. First, the algorithm sorts the events by timestamp and iterates over this list. Whenever the calculated speed between two succeeding events \(L[i]\) and \(L[i-1]\) exceeds \(V_{\text{supersonic}}\), the distances from \(L[i]\) to the next event \(L[i+1]\) and from \(L[i-1]\) to \(L[i]\) are calculated. The filter then removes the event with the larger distance to \(L[i+1]\). Again, this filter is applied iteratively until the input list equals the output list.

Algorithm 2. Recursive look-ahead filter
Input: list of location events \(L\)
Output: list of location events \(L\) (outliers removed)

#### Kalman Filter

The Kalman filter uses a probabilistic approach to remove the outliers in a trajectory: the Kalman filter is an algorithm that estimates the state of an underlying linear dynamical system from a series of noisy measurements \((15)\). Given a stream of such noisy input data, it recursively updates estimates not only of the current state of the system, but also of the current accuracy of this state. For this estimation, the Kalman filter assumes process and measurement noise generated by different Gaussian distributions.

For this study, it was assumed that the positions of cell phone events were noisy observations of an underlying true trajectory of a vehicle that could be described by a linear dynamical system with noisy state transitions. The benefits of the Kalman filter for this analysis were twofold: first, the analysis obtained corrected positions from the current state estimate and second, the current estimate of the state accuracy enabled identification of outlier events and removed them from the trajectory.
Although the events occurred at irregular time intervals, the Kalman filter was simulated in constant time steps $\Delta t = 1\,\text{s}$ and incorporated observations only in those time steps in which a cell phone event occurred. The internal state in the $k$th time step was modeled as a 4-dimensional vector $x_k = [x_k, y_k, \dot{x}_k, \dot{y}_k]^T$, reflecting the estimates of the current position and the velocity of the vehicle. The state evolved according to $x_k = F \cdot x_{k-1} + w_k$, with the state transition matrix

$$
F = \begin{bmatrix}
1 & 0 & \Delta t & 0 \\
0 & 1 & 0 & \Delta t \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
$$

where $F$ is independent of $k$ and there is no control input ($B = 0$). The process noise $w_k$ is normally distributed with covariance $Q(w_k \sim N(0, Q))$, which captures the statistics of accelerations and direction changes of vehicles within $\Delta t$. The matrix $Q$ is used to update the estimate of the current state accuracy, which is maintained in the form of a four-by-four covariance matrix $P_k$.

The information available in the existing GPS tracks was used to calculate the covariance $Q$. The analysis calculated samples of acceleration vectors between temporally adjacent GPS events that were $\Delta t$ apart,

$$
\begin{bmatrix}
a_0 \\
a_1 \\
a_x \\
a_y
\end{bmatrix} = \begin{bmatrix}
\cos \varphi & \sin \varphi \\
-\sin \varphi & \cos \varphi \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
\Delta v_x \\
\Delta v_y \\
\Delta \Delta t
\end{bmatrix}
$$

where $\varphi$ is the angle of movement from the previous position to the current position, and $\Delta \Delta t$ is the speed difference in m/s. The covariance of these acceleration samples was found to be

$$
C_a = \begin{bmatrix}
0.14 \, \text{m}^2/\text{s} & 0 \\
0 & 0.11 \, \text{m}^2/\text{s}^4
\end{bmatrix}
$$

The process noise covariance was then calculated by $Q = GC_aG^T$, with

$$
G = \begin{bmatrix}
\Delta \Delta t/2 & 0 \\
0 & \Delta \Delta t/2 \\
\Delta t & 0 \\
0 & \Delta t
\end{bmatrix}
$$

It was assumed that observations in the form of cell phone events were generated by $z_k = H \cdot x_k + v_k$, where $z_k = [\xi_k, \eta_k]^T$ was the two-dimensional position of the event. Thus,

$$
H = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
$$

and $v_k \sim N(0, R)$ is an observation noise that is also normally distributed. The covariance matrix $R$ accounts for the deviations of the position of events from the true position and was also estimated with the GPS data. For each cell phone event, the corresponding true position was determined from a GPS event with the same timestamp, yielding a vector that described the positional offset $[\Delta x, \Delta y]^T$. $R$ is then given by the covariance of these offset samples:

$$
R = \begin{bmatrix}
50,611,561 \, \text{m}^2 & 9,909,674 \, \text{m}^2 \\
9,909,674 \, \text{m}^2 & 27,244,590 \, \text{m}^2
\end{bmatrix}
$$

Both $C_a$ and $R$ have higher variance in the $x$-direction, which most probably reflects the larger east-west elongation of the Austrian territory. Since all internal calculations were in meters, the GPS coordinates were converted into a relative Cartesian coordinate system with the position of the first event of a GPS track as the origin. For display purposes, the estimated positions were later converted back into GPS coordinates with the same coordinate system.

Given a sequence of cell phone events, the Kalman filtering procedure works as follows: the initial state is set to $x_0 = 0$, and the initial accuracy is set to $P_0 = \text{diag}(10^4 \, \text{m}^2, 10^4 \, \text{m}^2, 10 \, \text{m}^2/\text{s}^2, 10 \, \text{m}^2/\text{s}^2)$. The Kalman filter is simulated from the time of the first event to the time of the last event of the sequence, discretized into steps of $\Delta t$. In each time step, the state $x_k$ and its accuracy $P_k$ are updated by using $F$ and $Q$; in the absence of observations (cell phone events), the accuracy will decrease. If there is an event in this time step, the Kalman filter is updated on the basis of its position, taking into account $H$, $R$, and $P_k$.

The output of the Kalman filtering process is a modified sequence of cell phone events, for which the position of each event is replaced by the current position estimate of the Kalman filter (the first two dimensions of the state vector). In addition, an event is removed if it is unlikely to be generated from the current state of the Kalman filter, which is the case if its position $\tilde{z}_k$ is too far from the current state $x_k$ in terms of the accuracy $P_k$. Such an event was removed if

$$
\frac{N(\tilde{z}_k | H x_k, P_H)}{N(\tilde{z}_k | H x_k, P_H)} < 0.1
$$

where $N(x | \mu, \Sigma)$ is the value of the probability density function of the normal distribution with mean $\mu$ and covariance $\Sigma$ at point $x$.

The algorithmic implementation of the Kalman filter is described in Algorithm 3. The input is a list of location events $L$, which are then sorted by time. Afterward, the Kalman filter is initialized with the parameters described earlier. Then, it iterates in steps of 1 s from the time of the first event to the time of the last event. If the input list contains an event at the given time, the Kalman filter is corrected with the converted coordinates of the event. If the observed position does not deviate too much from the internal state position (confidence $\geq 90\%$), the position of the internal state is added to the output list. On every time step, the predict-method of the Kalman filter is called, causing it to update its internal state.

Algorithm 3. Kalman filter

Input: list of location events $L$
Output: corrected list $M$ of location events (outliers removed)

Sort $L$ by time of events
Initialize Kalman filter $K$ with $x_0$, $P_0$, $H$, $Q$, and $R$
Initialize new list $M$

origin = $L[0]$ · position;

For time $= L[0]$ · time to $L[|L| \cdot \text{size} - 1]$ · time with Interval 1 second:

If $L$ contains event at time:

position = convertToCartesianCoordinates (event · position, origin)

$K$ · correct(position)
If \( K \cdot \text{confidence(position)} < 0.1 \):
\[
\text{estimatedPosition} = \text{convertToGPSCoordinates}(K \cdot \text{estimated Position, origin})
\]
\[
M \cdot \text{add(Event(estimatedPosition, time))}
\]
End if
End if
\[
K \cdot \text{predict()}
\]
End for
Return \( M \)

EVALUATION

To evaluate the proposed filters, three different approaches were taken. First, the filters were evaluated visually: a tool was created to draw the trajectories of moving phone users on a map and then apply the proposed filters. After the visual evaluation, an automatic evaluation based on provided GPS tracks was carried out. And last, a manual evaluation was done by equipping participants with GPS trackers and validating the extracted cell phone routes against the manually collected GPS tracks.

Visual Evaluation

Figure 2 shows a visualization of eight users driving from Graz (upper right) to Klagenfurt (lower left) on December 1, 2012. The travel distance was about 150 km (93.2 mi). In Figure 2a, no filters were applied. The data contained multiple outliers, rendering them almost useless for any traffic modeling. In Figure 2b, the recursive naive filter was applied. The results improved; however, the data still contained various outliers. Most of the outliers were removed with the recursive look-ahead filter in Figure 2c. The Kalman filter was applied in Figure 2d. All outliers were removed, although some trajectories were not as close to the actual road as with the recursive look-ahead filter. This difference was caused by the fact that the Kalman filter actually updated the location of the cell phone event with the current location of the internal state.

Statistical Evaluation Based on GPS Data

In addition to the visual evaluation, a statistical evaluation was performed on the basis of GPS data. The network provider offers a navigation solution for cell phones. When enabled, it sends the current GPS position every 30 s to the provider. At the same time, ordinary cell phone events are sent when a user sends or receives a call or changes location area. A random subset of these data was extracted that contained 4,933 users driving through Austria over a period of 4 months. For a user to be included in the evaluation data set, the following criteria must have been fulfilled: (a) each track was at least 10 km (6.2 mi) long; (b) each track contained at least 20 GPS events; (c) there were five non-GPS events; and (d) each non-GPS event occurred within the time range of the first and last

FIGURE 2  Visual validation of proposed filters: (a) no filter, (b) recursive naive filter, (c) recursive look-ahead filter, and (d) Kalman filter (raw trajectories in (a) contain multiple outliers, which have been removed with the filters in (b), (c), and (d)).
GPS events (+/− 20 s). Once a user was included, the GPS track served as the ground truth against which the cell phone trajectory was evaluated. Three methods were used to compare the two tracks:

1. Speed-based evaluation. On the basis of the timestamp of every GPS event, the two surrounding cell phone events were selected, and the travel speed between those events was calculated (distance/time) and compared with the current speed reported by the GPS device. Then the mean difference for every user was calculated. These steps were repeated for every filter.

2. Distance to calculated position. Identical to the speed evaluation, the two surrounding cell phone events for every GPS event were selected. On the basis of timestamp of the GPS event, the estimated position was interpolated between the two mobile phone events. Then the distance from this estimated position to the actual GPS position was calculated.

3. Distance from cell phone event to GPS position. For every cell phone event, the distance to the time-corresponding GPS position was calculated.

Figure 3 presents a visualization of the results of the three evaluations for the 4,933 routes. The average route length was 86.7 km (53.9 mi) and contained 5,367 GPS and 68 non-GPS events. The speed-based evaluation is shown in Figure 3a. All filters improved the result significantly compared with non-filtered data (Wilcoxon signed-rank test: \( p < .001 \) for all filters). The recursive look-ahead filter [mean = −16.3 km/h, standard deviation (SD) = 15.2 km/h] had a lower variance than the recursive naive filter (mean = −12.0 km/h, SD = 18.4 km/h) and the Kalman filter (mean = −20.2 km/h, SD = 21.5 km/h). Without filtering, large deviations occurred because of outliers (mean = −0.6 km/h, SD = 329.9 km/h). The reason why all the filters estimated lower speeds than the actual speed can be explained by the fact that the linear distance between two cell phone events was shorter than the driving distance between those locations. Thus, the speed calculation (distance/time) tended to be lower.

Figure 3b shows the result for the second approach, the mean deviation from the actual GPS position. Although the recursive naive filter (mean = 3.35 km, SD = 8.56 km) and the recursive look-ahead filter (mean = 3.15 km, SD = 10.47 km) improved the result significantly (Wilcoxon signed-rank test: \( p < .001 \)), the Kalman filter (mean = 4.34 km, SD = 8.12 km) worsened the result significantly (\( p < .001 \)) compared with no filtering (mean = 3.90 km, SD = 8.59 km). This difference can be explained by the fact that the Kalman filter actually rewrote the route. The route produced by the Kalman filter was slightly delayed compared with the actual route; thus, the calculated distance from the actual to the estimated position was higher. This phenomenon also explains the poorer performance of the Kalman filter compared with the other filters in the third evaluation method.

Figure 3c provides a visualization of the third evaluation approach. All three proposed filters, the recursive naive filter (mean = 1.45 km, SD = 5.77 km), the recursive look-ahead filter (mean = 1.25 km, SD = 8.12 km), and the Kalman filter (mean = 1.60 km, SD = 5.42 km) significantly improved the estimation of travel speed and position (Wilcoxon signed-rank test: \( p < .001 \)) compared with no event filtering (mean = 1.62 km, SD = 4.50 km).

Even with the naive approach, a solid improvement can be achieved. Among the proposed filters, the recursive look-ahead filter yielded the best overall performance. The Kalman filter performed worse than the other filters.

**Manual Evaluation Based on GPS Data**

In the statistical evaluation, the filters were evaluated against provided GPS events. Since the data were from the same source as the cell phone events, the data might be biased. The plausibility of the GPS events was validated by drawing them on a map and checking whether they correlated correctly with a street. An external verification was also performed in which five individuals were equipped with a GPS tracker and a cell phone and asked to switch both devices on when driving on the Austrian motorway A2 from Graz to Klagenfurt. The corresponding cell phone events were extracted from the data, according to the timestamp and the approximate position, and the filters were again compared.

Table 1 shows the values for the speed calculation. Although the outliers heavily distorted the calculation when no filtering was applied, the values were more reasonable when the filters were applied. Basically, the results were in accordance with the findings in the statistical evaluation. In addition, the calculated speed was consistently lower than the actual speed, which can again be explained by shorter trajectories compared with the actual routes. (The linear

---

**FIGURE 3** Evaluation of filters against provided GPS data: (a) deviation of calculated speed from actual speed, (b) deviation from actual GPS position to estimated position, and (c) deviation from cell phone events to actual GPS position.
distance was shorter than the actual driving distance.) Table 2 and Table 3 show the second and third evaluation approaches, respectively. The recursive naive filter and the recursive look-ahead filter performed best, again in accordance with the previous findings.

**DISCUSSION AND CONCLUSION**

This study proposed filters to remove outliers in cell phone trajectories. Two of the filters, namely the recursive naive filter and the recursive look-ahead filter, significantly improved the speed and position estimations as compared with no filtering. The Kalman filter improved the speed estimation as well, but did not perform as well as the other filters regarding the estimation of actual position. One explanation might be that because of the sparseness of the cell phone events, the updates to the filter did not occur frequently enough to achieve good performance. If there was no update, for example, for 50 km, the previous events must have reflected the route well; otherwise, the internal state of the Kalman filter would deviate too much from the actual position for a suitable approximation. Another explanation could be that the standard variant of the Kalman filter that was used was linear. A nonlinear Kalman filter, such as the extended Kalman filter or the unscented Kalman filter, might have achieved better results.

Given the sparseness and inaccuracy of cell events, exact determination of the travel speed and position remains a difficult problem. Yet, the proposed filters facilitate the approximation of the actual values. Even with the relatively simple recursive naive filter, a solid improvement can be achieved. In a real-time setting, both the recursive naive filter and the Kalman filter would be preferable to the recursive look-ahead filter, since the latter relies on an additional (future) event to decide whether the current event is an outlier. In the application field of cell phone data, it has not been determined when this future event will arrive (or if it even arrives). In contrast, the recursive naive filter and the Kalman filter can decide immediately whether the current event is an outlier. On the basis of the experiments described here, the recursive look-ahead filter would be the best choice in an offline setting.

The use of cell phone signaling data for traffic modeling relies on active cell phone usage by drivers, which might pose an issue in some countries where cell phone usage is prohibited during driving. But with the increasing equipment in cars with hands-free devices, people will certainly continue to use cell phones in vehicles.

The removal of outliers from cell phone trajectories has a positive impact on different types of applications. For example, it facilitates the mapping of single or clustered trajectories to actual roads (map-matching). A second application would be the clustering of trajectories to distinguish different transportation modes (road versus train). Clusters can be separated more easily as trajectories better reflect the actual routes. In addition, travel times can be estimated more accurately when outliers are removed from cell phone data.

**ACKNOWLEDGMENTS**

This research was partly supported by A1 Telekom Austria, which provided cell phone monitoring data sets. The Know-Center is funded within the Austrian COMET Program under the auspices of the Austrian Ministry of Transport, Innovation, and Technology; the Austrian Ministry of Economics and Labor; and Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

**REFERENCES**


The Travel Survey Methods Committee peer-reviewed this paper.