Smartphone Placement within Vehicles

Johan Wahlström, Isaac Skog, Peter Händel, Bill Bradley, Samuel Madden, and Hari Balakrishnan

***Abstract*—Smartphone-based driver monitoring is quickly gaining ground as a feasible alternative to competing in-vehicle and aftermarket solutions. Today, the main challenges for data analysts studying smartphone-based driving data stem from the mobility of the smartphone. In this study, we use kernel-based k-means clustering to infer the placement of smartphones within vehicles. All in all, trip segments are mapped into fifteen different placement clusters. As part of the presented framework, we discuss practical considerations concerning e.g., trip segmentation, cluster initialization, and parameter selection. The proposed method is evaluated on more than 10 000 kilometers of driving data collected from approximately 200 drivers. To validate the interpretation of the clusters, we compare the data associated with different clusters and relate the results to real world knowledge of driving behavior. The clusters associated with the label "Held by hand" are shown to display high gyroscope variances, low maximum speeds, low correlations between the measurements from smartphone-embedded and vehicle-fixed accelerometers, and short segment durations.**

***Index Terms*—Telematics, inertial sensors, smartphones, kernel-based k-means clustering.**

I. Introduction

One of the primary challenges of the intelligent transportation systems (ITS) society lies in modifying current implementations so that they can be based on the sensing, computation, and connectivity capabilities of mobile devices such as smartphones. This field within ITS is referred to as smartphone-based vehicle telematics, and emerged after the birth of the modern smartphone, about a decade ago [1]. Smartphones enable end-user centric and vehicle-independent telematics solutions that complement vehicle-centric factory-installed telematics systems. Generally, smartphone-based implementations benefit from seamless software updates, intuitive user interfaces, short development cycles, and low development costs [2]–[7]. One example application where smartphones have sparked a major industry disruption is insurance telematics. In insurance telematics, driving data is used to adjust automotive insurance premiums and provide various value-added services to policyholders. By utilizing smartphone-embedded sensors rather than e.g., in-vehicle sensors, many insurers can both lower maintenance costs and increase customer engagement [8]. As of March 2018, there were sixty-nine active mobile-based insurance telematics programs [9]. This represents an increase of more than 100 % in two years.

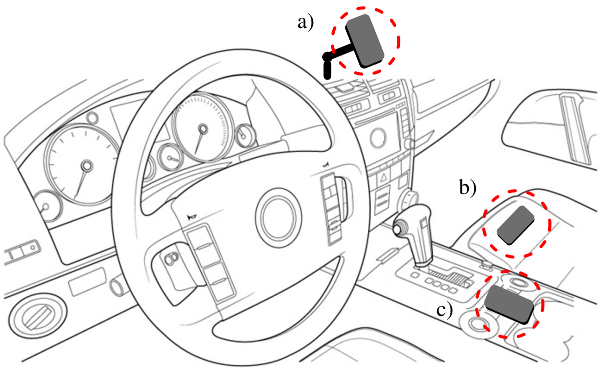


Fig. 1. We consider the problem of inferring the smartphone's placement in a vehicle, i.e., whether it is placed a) in a car mount; b) on the passenger seat; c) on the center console; d) in the driver's pants (not illustrated); or e) held by hand (not illustrated).

Over the same period, the market share of smartphone-based solutions within insurance telematics has increased from 2% to 29%. With four out of five millennials positive to sharing their recent driving data, the industry can expect further growth in the near future [10]. There are today several companies, such as Cambridge Mobile Telematics, TrueMotion, and The Floow, whose primary focus is to provide smartphone-based telematics solutions to insurance companies around the world. Smartphones may also be used for road-condition monitoring [11], real-time ridesharing [12], vehicular ad hoc networks (VANETs) [13], and value-added services such as vehicle-finder services and fuel and routing optimization.

Despite the many advantages of smartphone solutions, smartphone-based driver behavior profiling is generally challenging and does in many situations not reach the same performance as competing data collection methods. Due to the mobility of the smartphone, the data processing is fundamentally different from traditional sensor fusion using vehicle-fixed sensors. Recent studies have for example focused on how to classify data from smartphone-embedded sensors into classes associated with different transportation modes [14]–[16], as well as how to estimate the orientation [17] and the position [18], [19] of the smartphone with respect to the vehicle frame. Further, the utility and characteristics of vehicle data recorded from smartphones can be expected to depend on whether the smartphone is rigidly mounted on a cradle, placed on the passenger seat, held by hand, etc.

Despite an increasing interest in smartphone-based driver behavior profiling, there is still a big gap between academia and industry: While most academic studies assume that the smartphone is fixed to the vehicle, anyone who studies large-scale industry data collected from smartphone-embedded and orientation-dependent sensors such as accelerometers, gyroscopes, and magnetometers, will inevitably have to consider

J. Wahlström is with the Dept. of Computer Science, University of Oxford, Oxford, UK (e-mail: johan.wahlstrom@cs.ox.ac.uk).

I. Skog is with the Dept. of Electrical Engineering, Linköping University, Linköping, Sweden (e-mail: isaac.skog@liu.se).

P. Händel is with the Dept. of Information Science and Engineering, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: ph@kth.se).

B. Bradley, S. Madden, and H. Balakrishnan are with Cambridge Mobile Telematics, Cambridge, MA 02142 (email: {wfbradley, sam, hari}@cmtelematics.com).

the effects of e.g., users picking up their smartphones during trips. As illustrated in Fig. 1, this article attempts to bridge this gap by examining the problem of unsupervised inference on the smartphone's conceptual placement[1] within a vehicle. Knowledge of the smartphone's placement is important for several reasons. For instance, it can be used to assess the reliability of collected data (inertial measurements collected from smartphones held by hand are e.g., less suitable for the detection of harsh acceleration events since hand motions may interfere with the measurements). Hence, detected harsh braking events could be considered more or less reliable depending on during which smartphone placement they were detected. Moreover, it may be used for accident reconstructions (to answer e.g., "Was the driver interacting with his smartphone before the accident?") and assessments of driver distraction (drivers interacting with their mobile phone are at a significantly increased risk of being involved in an accident [20]). Recently, there has been several efforts to reduce driver distraction by tracking the amount of time that drivers spend interacting with their smartphone using smartphone-embedded sensors and then use this to encourage more attentative driving [21].

The following approach is taken in this article: Trip segments are mapped into fifteen different states using kernel-based k-means clustering. The features are computed using data from smartphone sensors and a vehicle-fixed sensor tag equipped with an accelerometer triad. Within the inference framework we make use of classical mechanics, established methods for inertial measurement unit (IMU) alignment, knowledge of the relation between the smartphone-to-vehicle orientation and the smartphone's placement in the vehicle, and behavioral statistics published by the National Highway Traffic Safety Administration (NHTSA). The results are interpreted by studying the feature distributions of the different clusters and relating this to typical driver behavior characteristics. Last, we demonstrate the effect that the smartphone's placement has on the efficiency of accelerometer-based detection of harsh accelerations.

II. TRIP SEGMENTATION

Assume that the smartphone at each time instant can be said to be in one and only one of a discrete number of states describing its conceptual placement. Instead of jointly estimating the smartphone placement at all time instants in a given trip, we will first attempt to identify the time points at which the state can change (such as when the driver pulls out the smartphone from his pocket and mounts it on the dashboard). Assuming that all true state changes have been detected, all driving segments stretching from one detected state change to the next can then be clustered by means of standard clustering methods.

A. Gyroscope-based Segmentation

Periods at which the smartphone state can change were considered to start when the Euclidean norm of the gyroscope measurements exceeded $\bar{\omega}$. Likewise, these periods were considered to end a time period $\bar{t}$ after the last sampling instance at which the gyroscope measurements exceeded the same threshold. All data within these periods was then discarded, and the data before and after was used to form two new independent trip segments. Segments that were shorter than $\bar{t}$, or where the speed did not exceed $\bar{s}$, were also discarded. The segmentation algorithm is illustrated in Fig. 2.

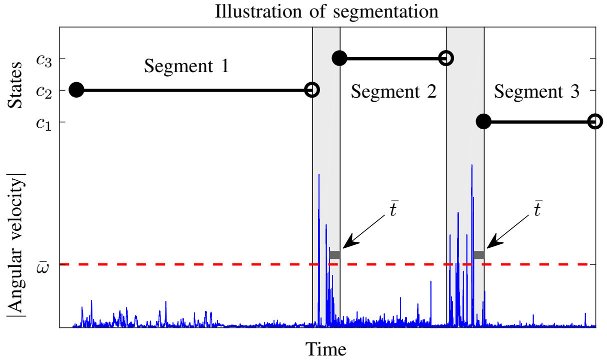


Fig. 2. Illustration of the trip segmentation described in Section II-A based on typical gyroscope measurements. The true placements over the considered trip are assumed to be c_2, c_3, and c_1.

B. Segmentation Parameter Selection

The threshold of $\bar{\omega} = 100\,[^\circ/s]$ was chosen so that it will typically not be reached during normal vehicle maneuvers[2] (when the smartphone is fixed to the vehicle), but so that it will be reached in most of the cases when the smartphone is, by hand, moved from one place to another within the vehicle. Further, the lower limits on the duration and speed of a segment of interest were set to $\bar{t} = 10\,[s]$ and $\bar{s} = 10\,[km/h]$, respectively.

C. Accelerometer-based Segmentation

For the trips where gyroscope measurements were unavailable (some older smartphones are not equipped with gyroscopes) we instead used smartphone-based accelerometer measurements to estimate the angular velocity. First, the accelerometer measurements were low-pass filtered using a Butterworth filter of order five and with a cutoff frequency of $2\,[Hz]$ (to suppress noise and make the number of trip segments similar to that obtained with gyroscope measurements). Second, instantaneous estimates of the smartphone's roll and pitch angles were computed by assuming that the filtered signals only reflect the accelerometer measurements due to gravity[3] [23]. Obviously, this approach also relies on

[1] Note that we talk of placement in a more abstract sense than simply referring to the smartphone's position with respect to the vehicle.

[2] One way to assess the range of angular velocities that are encountered in everyday driving is to consider the lateral acceleration of a vehicle driving in an horizontal circle with constant speed. Thus, assuming a minimum turning radius of $5\,[m]$ and a coefficient of friction of 1, it follows that the vehicle cannot drive with an angular velocity higher than $\sqrt{9.8/5} \cdot 180/\pi\,[^\circ/s] \approx 80\,[^\circ/s]$ without losing its grip of the road surface. Here, we use that the no-sliding condition under the stated assumptions is $\omega^2 r/g < \mu$, where ω, r, g, and μ denote the vehicle's angular velocity, the turning radius, the gravity force, and the coefficient of friction, respectively [22].

[3] The smartphone frame was defined to have its three coordinate axes pointing to the right (as seen from a user facing the display), pointing upwards along the display, and in the direction of the display.

the assumption that eventual sensor biases can be neglected. Third, we used the estimated roll and pitch angles to compute the matrix $\delta\mathbf{C}_k \triangleq \mathbf{C}_{k+1}^{\intercal}\mathbf{C}_k$ at each sampling instance k. Here, $\mathbf{C}_k$ is the rotation matrix corresponding to the roll and pitch estimates at sampling instance k and a zero yaw angle (the yaw angle can be estimated if magnetometer measurements are available). Under the small angle approximation, $\delta\mathbf{C}_k \cdot f_s$, where f_s is the sampling rate, is equal to a skew symmetric matrix from which estimates of the smartphone's angular velocity can be extracted [23]. Once these estimates have been extracted, the segmentation can be performed in the same way as when gyroscope measurements are available.

III. KERNEL-BASED K-MEANS CLUSTERING

The objective of kernel-based k-means clustering is to map the N data points $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ into the k different classes

$$\{\Omega_1, \ldots, \Omega_k\} \triangleq \underset{\Omega_1,\ldots,\Omega_k}{\arg\min} \sum_{c=1}^{k} \sum_{\mathbf{x}_i \in \Omega_c} \|\mathbf{h}(\mathbf{x}_i) - \mathbf{m}_c\|^2. \quad (1)$$

Here, $\mathbf{m}_c \triangleq \sum_{\mathbf{x}_i \in \Omega_c} \mathbf{h}(\mathbf{x}_i)/|\Omega_c|$ and $|\Omega_c| \triangleq \sum_{\mathbf{x}_i \in \Omega_c} 1$ denote the centroid and the cardinality of the cth cluster, respectively, while $\mathbf{h}(\cdot)$ is the basis function vector. Further, $\|\cdot\|$ denotes the Euclidean norm, and the kernel function is defined as

$$k(\mathbf{x}_i, \mathbf{x}_j) \triangleq \mathbf{h}(\mathbf{x}_i)^{\intercal}\mathbf{h}(\mathbf{x}_j). \quad (2)$$

Typically, one would employ iterative methods to find one or several local minimums to the optimization problem (1) [24]. The k-means algorithm was chosen due to its simplicity, computational efficiency, and familiarity to practitioners. Refer to [25] for a discussion on regularized k-means clustering and how it can be used to prevent overfitting.

A. Features

This subsection describes the features that were extracted from the segments. The features were derived from smartphone-based inertial and global navigation satellite system (GNSS) measurements, information on the smartphone's screen state, and accelerometer data from a vehicle-fixed sensor tag. The measurements from the tag were only used to compute the fifth feature, the tag-smartphone correlation.

1) Smartphone-to-vehicle orientation: Methods for estimating the smartphone-to-vehicle orientation have previously been reviewed in [1] (see also [26]). In this study, we first computed the median values (over the segment) of the accelerometer measurements in each direction. The roll and pitch angles of the smartphone-to-vehicle orientation were then estimated by assuming that these median values only reflect the accelerometer measurements due to gravity, and that the vehicle's roll and pitch angles could be approximated as zero[4] (the smartphone-to-vehicle orientation is illustrated in Fig. 3) [23]. Finally, the yaw angle was estimated by finding the direction in the horizontal plane (perpendicular to the direction of the vector with the median accelerometer measurements) that had the largest variance in the accelerometer measurements, and assuming that this was the vehicle's forward/backward direction [27]. We then separated the vehicle's forward and backward directions by assuming that the accelerometer measurements in the forward direction were positively correlated with the differentiated GNSS measurements of speed. As should be obvious, the computations rely on the assumption that while the smartphone may vibrate or be affected by minor hand movements, the smartphone-to-vehicle orientation is not subject to any significant changes during the course of a given segment (i.e., that the segmentation described in Section II has been successful).

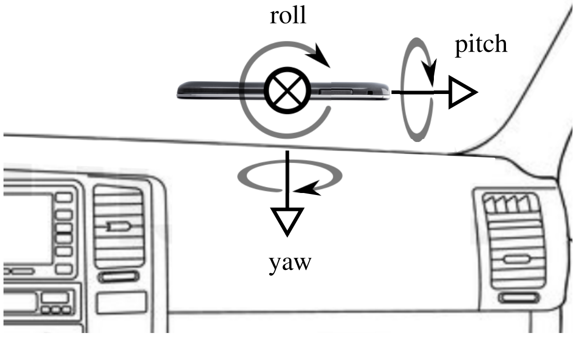


Fig. 3. Illustration of smartphone and vehicle when the roll, pitch, and yaw angles of the smartphone-to-vehicle orientation are zero.

2) Accelerometer variance: The accelerometer variance was first computed separately along each of the three spatial dimensions. We then computed the sum of the three variance measures and took the logarithm of the resulting value to compress order-of-magnitude variations.

3) Gyroscope variance: The gyroscope variance was computed in the same way as the accelerometer variance.

4) Maximum speed: The maximum speed of the vehicle during the segment as indicated by GNSS measurements.

5) Tag-smartphone correlation: The correlation between the accelerometer measurements from the external accelerometer tag and the smartphone (both rotated to the vehicle frame) was first computed separately along each spatial dimension. The feature was defined as the sum of the three correlations.

6) Segment duration: The logarithm of the temporal length of a segment.

7) Screen state: The percentage of time that the screen was on during the segment.

8) Initial screen state: The percentage of time that the screen was on during the first ten seconds of the segment.

All features except the smartphone-to-vehicle orientation were normalized over all segments by means of linear rescaling, and after the normalization took on values between -1 and 1. If a feature could not be computed for a given segment due to missing data (for example, some vehicles did not have a sensor tag installed) the feature value was set to the median value (as taken over the remaining segments) of the same feature. In commercial deployments intended for a large number of users, low-dimensional features could be computed

[4] The vehicle frame was defined as a forward-right-down frame. The Euler angles describing the smartphone-to-vehicle orientation were defined to be the angles of rotations that applied to the yaw-pitch-roll axes (in that order) rotated the vehicle frame to the smartphone frame.

TABLE I
INITIALIZATION FOR K-MEANS CLUSTERING.

State c	Placement label d	Description of (likely) placement	ϕ [rad]	θ [rad]	ψ [rad]	Initial screen state
1	1) Seat	Lap or driver/passenger seat	π	0	0	off
2	1) Seat	Lap or driver/passenger seat	π	0	$\pi/2$	off
3	1) Seat	Lap or driver/passenger seat	π	0	π	off
4	1) Seat	Lap or driver/passenger seat	π	0	$3\pi/2$	off
5	2) Cup holder	Cup holder	$3\pi/2$	0	0	off
6	2) Cup holder	Cup holder or car mount	$3\pi/2$	0	$\pi/2$	off
7	2) Cup holder	Cup holder	$3\pi/2$	0	π	off
8	2) Cup holder	Cup holder	$3\pi/2$	0	$3\pi/2$	off
9	2) Cup holder	Cup holder or car mount	$3\pi/2$	0	$\pi/2$	on
10	3) Pants	Pants pocket	0	0	$\pi/2$	off
11	4) Console	Dashboard or center console	π	0	$\pi/2$	on
12	5) Held by hand	Held by hand (call, left ear)	$3\pi/2-\epsilon$	ϵ	0	on
13	5) Held by hand	Held by hand (call, right ear)	$3\pi/2-\epsilon$	$-\epsilon$	π	on
14	5) Held by hand	Held by hand (e.g., text messaging)	$5\pi/4$	0	$\pi/2$	on

Here, ϕ, θ, and ψ denote the roll, pitch, and yaw angles (of the smartphone-to-vehicle orientation), respectively. Further, ϵ denotes the angle $10 \cdot \pi/180$ [rad].

using local resources on the smartphone and then sent to the cloud for further processing.

B. Kernel

The clustering was made using the kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-(\|\mathbf{x}_i^{\setminus\nu} - \mathbf{x}_j^{\setminus\nu}\|^2 + (2\nu_{i,j}/\pi)^2)/(2\sigma^2)\big) \quad (3)$$

where $\mathbf{x}^{\setminus\nu}$ is the vector of normalized features (all except the smartphone-to-vehicle orientation) and

$$\sigma \in (0, \infty) \quad (4)$$

is the kernel width. This kernel was chosen since it was the most simple conceivable kernel that would handle the orientation feature in a suitable way. Here, $\nu_{i,j}$ is the rotation angle corresponding to the rotation matrix $\mathbf{C}_i\mathbf{C}_j^{\intercal}$ [28], and $\mathbf{C}_i$ is the rotation matrix for the smartphone-to-vehicle orientation associated with data point i. The calibration of the kernel width is described in Section III-D. The factor $2/\pi$ was included since $(2\nu/\pi)^2 \in [0, 4]$, i.e., $(2\nu/\pi)^2$ will take on values in the same range as the contribution from each individual feature to $\|\mathbf{x}_i^{\setminus\nu} - \mathbf{x}_j^{\setminus\nu}\|^2$. Note the similarity between the employed kernel and the well-known radial basis function (RBF) kernel [29].

C. Initialization

The iterative methods that are used to find local minimums to the k-means problem often make use of some initialization based on available information on the problem at hand. We made an initialization based on the features describing the smartphone-to-vehicle orientation and the initial screen state. All in all, fifteen states were constructed. The expected typical orientations and initial screen states of the fourteen first states are specified in Table I. The second column describes the aggregated and abbreviated placement labels that will be used for the experimental study in Section IV, and the third column gives more details on the placements associated with the individual states. The expected orientations for most of the states are illustrated in Fig. 4. The fifteenth state is a dummy state "Unknown" for unexpected orientations. This state will tend to attract segments that are difficult to interpret, thereby preventing such segments from having a distorting influence on the remaining states. Initial trials showed that including a dummy state made the difference between the feature distributions associated with the "Held by hand" label and those of the other labels more pronounced.

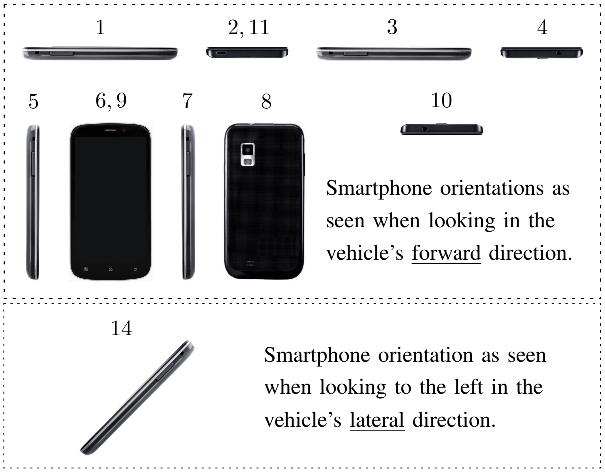

1
2, 11
3
4
5
6, 9
7
8
10
Smartphone orientations as seen when looking in the vehicle's forward direction.
14
Smartphone orientation as seen when looking to the left in the vehicle's lateral direction.

Fig. 4. The smartphone orientations that were used for the initialization. The numbers above the images refer to the states described in Table I. The orientations for states twelve and thirteen are not shown due to their more complicated nature.

We arrived at the fifteen states described in the previous paragraph by manually fine-tuning the set of states and their specification to make the clustering results comply with the behavioral statistics described in Section III-D. States that were included during initial clustering trials but then removed due to having too few associated segments included $\phi = \pi$, $\theta = \pi/4$, $\psi = 0$, with the screen on. This could be interpreted as handheld interaction with the display in landscape mode.

Each segment was initialized in the state in Table I with the nearest orientation and an initial screen state matching that of the segment. In mathematical terms, all trip segments i where the screen was on more than $50\,\%$ of the first ten seconds were initialized in the state

$$c_i = \arg\min_{c \in \Omega_{\text{on}}} \nu_{c,i} \tag{5}$$

where $\nu_{c,i}$ denotes the rotation angle corresponding to the rotation matrix $\mathbf{C}_c\mathbf{C}_i^{\mathsf{T}}$, $\mathbf{C}_c$ is the rotation matrix for the smartphone-to-vehicle orientation specified for state c in Table I, and Ω_{on} denotes the set of states for which the initial screen state is specified as "on" in Table I. This should be interpreted as choosing to initialize the segment i in the state with the "closest" expected smartphone-to-vehicle orientation, among the states for which the screen initially is expected to be on. The only exceptions were the trip segments where

$$\min_{c \in \Omega_{\text{on}}} \nu_{c,i} > \gamma \tag{6}$$

with

$$\gamma \in (0, \pi) \tag{7}$$

being a fixed threshold parameter. In these cases, the smartphone-to-vehicle orientation of segment i was not considered to be close enough to any of the orientations specified for states $c \in \Omega_{\text{on}}$ in Table I, and hence, these segments were initialized in the state "Unknown". The same approach was taken for trip segments where the screen was on less than $50\,\%$ of the first ten seconds. However, in this case, the minimization was taken over Ω_{off}, i.e., the states for which the screen state is specified as "off" in Table I. The calibration of the threshold parameter is described in Section III-D.

The choice of using multiple separate states for a single conceptual placement (all the four first states in Table I are, for example, associated with the smartphone being placed on the "lap or driver/passenger seat") is natural as we expect the set of smartphone-to-vehicle orientations and initial screen states associated with some given placements to be non-convex[5]. This is clearly understood when studying e.g., the smartphone-to-vehicle orientations for states twelve, thirteen, and fourteen, all associated with the placement "Held by hand" (see Table I). The use of multiple separate states for a single conceptual placement is further motivated by the fact that different orientations can be expected to be more common than others. For example, we would expect the clusters of states six and eight (both associated with the smartphone being placed in a cup holder) to have different characteristics since the smartphone display will more often be facing the driver or the passengers (state six) than not (state eight). We do not expect the clusters associated with a given placement to be non-convex in any of the features other than the smartphone-to-vehicle orientation and the two screen state features.

D. Clustering Parameter Selection

It now remains to tune the kernel width σ and the threshold parameter γ from equations (4) and (7), respectively. Methods for parameter selection in cluster analysis include manual tuning, optimization of internal evaluation measures quantifying some chosen clustering objective, optimization of external evaluation measures using ground truth data, optimization of robustness measures using cross-validation [31], or choosing parameter values based on the density of feature values [32]. We compared the obtained clustering distribution with previously published behavioral statistics on smartphone placements (unfortunately, data was only available for a subset of the locations we considered). The kernel width and the threshold were chosen as the solution to the least-squares problem

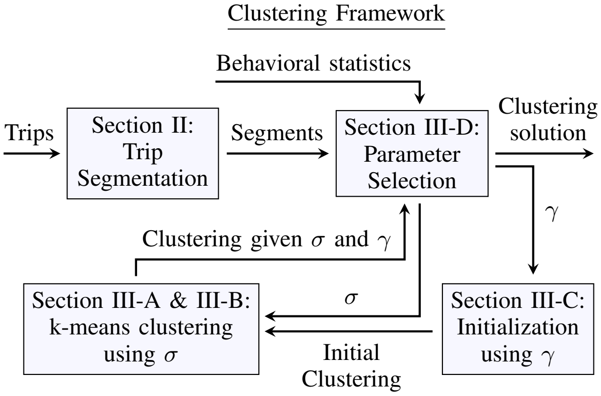

Clustering Framework
Behavioral statistics
Trips
Section II: Trip Segmentation
Segments
Section III-D: Parameter Selection
Clustering solution
γ
Clustering given σ and γ
Section III-A & III-B: k-means clustering using σ
σ
Initial Clustering
Section III-C: Initialization using γ

Fig. 5. Block diagram illustrating the clustering framework described in Sections II and III.

$$\arg\min_{\sigma,\gamma} \sum_{d=1}^{L} (\hat{y}_d(\sigma,\gamma) - y_d)^2. \tag{8}$$

Here, $\hat{y}_d(\sigma,\gamma)$ denotes the percentage of segments that were clustered into one of the states associated with placement label d given the parameters σ and γ. Similarly, y_d denotes the percentage of "reaching instances" where a given set of drivers reached for their smartphone from placement d as reported in a study published by the NHTSA [33]. The parameter d traverses the first four placement labels in Table I, giving $L = 4$ with $y_1 = 48.9\,[\%]$, $y_2 = 42.0\,[\%]$, $y_3 = 5.7\,[\%]$, and $y_4 = 3.4\,[\%]$. Since the NHTSA study did not report on the percentage of segments where the smartphone remained held by hand for a substantial period of time, and, in addition, included some less common reaching locations such as "purse", both $\hat{y}_d(\sigma,\gamma)$ and y_d were normalized before the optimization so that $\sum_{d=1}^{L} \hat{y}_d(\sigma,\gamma) = 1$ and $\sum_{d=1}^{L} y_d = 1$.

An overview of the clustering framework presented in Sections II and III is given in Fig. 5.

IV. Experimental Results

The studied data consists of 1000 trips collected from 194 drivers in South Africa (with left-hand traffic) using the DriveWell app in May 2016 as part of a partnership between *Cambridge Mobile Telematics* and *Discovery Insurance*[6]. The mean trip duration and the mean trip length were about 20 minutes and 11 kilometers, respectively. All signals were sampled at $15\,[Hz]$, with the exception of the GNSS data

[5] Standard k-means clustering can only find clusters that are convex in the basis function vector [30].

[6] One of the outcomes of the partnership was a contest where drivers earned points for safe driving. Over time, the contest had the effect of reducing the participants' speeding, hard braking, hard cornering, and phone usage while driving [34].

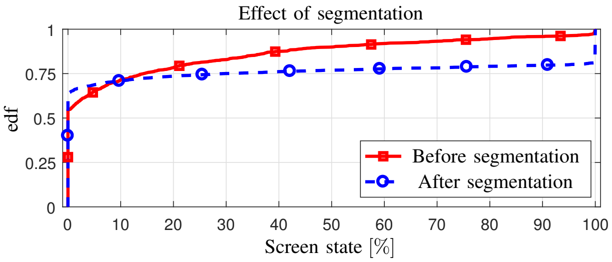


Fig. 6. The empirical distribution functions of the screen state feature before and after the segmentation.

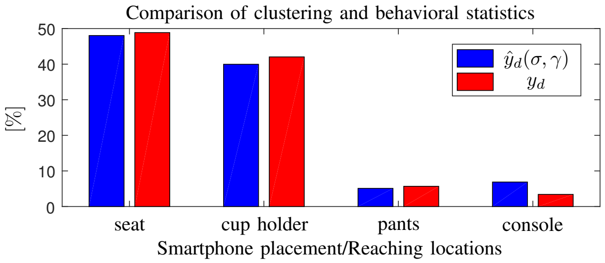


Fig. 7. Distribution of the "reaching locations" as estimated by the clustering algorithm ($\hat{y}_d(\sigma,\gamma)$) and as specified in a report from NHTSA (y_d) [33].

that was sampled at $1\,[Hz]$ (this is the standard sampling rate of GNSS receivers in current smartphones). Accelerometer, GNSS, and screen state measurements were available from all trips, while gyroscope and tag measurements were available for all but 64 and 137 trips, respectively.

A. Trip Segmentation

The trips were segmented as described in Section II, thereby dividing the 1000 trips into 3218 segments. We chose to implicitly evaluate the segmentation using the screen state feature, which is a feature that is independent of the inertial measurements that are used to perform the segmentation. Fig. 6 illustrates the segmentation by displaying the empirical distribution function (edf; sometimes denoted ecdf) of the screen state feature, contrasting the distributions across entire trips versus trip segments. As can be seen from Fig. 6, the percentage of trips or segments where the screen state feature takes on values far from $0\,[\%]$ and $100\,[\%]$ decreases after the segmentation. This is intuitive since a change in the instantaneous screen state (i.e., the screen is turned on or off) in the middle of a trip implies that the placement of the smartphone is likely to have changed over the trip. In other words, the edf shows that the segmentation ”polarizes” phone use — typical usage over a trip segment is mostly near $0\,[\%]$ or $100\,[\%]$. This polarization suggests that the segmentation successfully divides the trips into segments with distinct smartphone placements.

B. Clustering Parameter Selection

The parameter selection was performed as described in Section III-D, and the optimal values $\sigma = 5 \cdot 10^{-3}$ and $\gamma = 2.1$ were found by means of a grid search with step sizes $2 \cdot 10^{-4}$ and 0.05 in the directions of σ and γ, respectively. The distributions of $\hat{y}_d(\sigma,\gamma)$ and y_d are illustrated in Fig. 7.

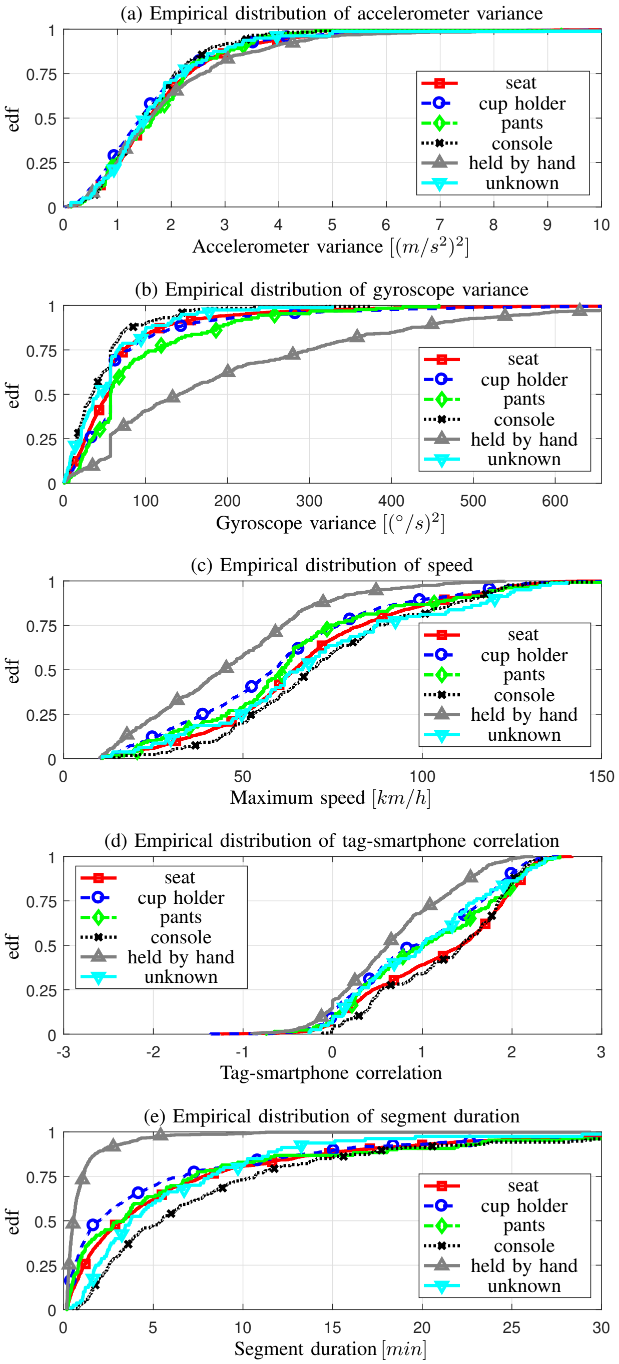


Fig. 8. The distributions of the features (a) accelerometer variance, (b) gyroscope variance, (c) maximum speed, (d) tag-smartphone correlation, and (e) segment duration.

Although the distributions agree quite well, one should be careful not to overinterpret the results. For example, note that the NHTSA data and the clustered data were collected in the US and South Africa, respectively. As a result, the underlying distributions may be different, which could mean that we are, to some extent, distorting the clustering results when trying to fit the distribution of the clustered data to that of the NHTSA data. In addition, selection biases may be present in the studied populations (the NHTSA e.g., only used data from drivers who reported using their cell phone at least once per day while driving), and we should also expect the clustering algorithm

TABLE II
THE CLUSTERING DISTRIBUTION.

State c	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
# of trips	192	385	357	280	205	306	196	189	114	129	174	170	125	316	80
pct. of trips [%]	6.0	12.0	11.1	8.7	6.4	9.5	6.1	5.9	3.5	4.0	5.4	5.3	3.9	9.8	2.5
Placement label d	1				2					3	4	5			6
# of trips	1214				1010					129	174	611			80
pct. of trips [%]	37.7				31.4					4.0	5.4	19.0			2.5

The clustering distribution over the fifteen states described in Table I, and over the aggregated placement labels 1) Seat, 2) Cup holder, 3) Pants, 4) Console, 5) Held by hand, and 6) Unknown.

to misinterpret some segments. The clustering distribution is detailed in Table II.

C. Feature Comparison

Since there is no ground truth available, the results of the clustering will be analyzed by studying the edfs of the different features that were used for the clustering. The edfs are computed separately for the aggregated placements labels specified in the second column of Table I. Figs. 8 and 9 display the distributions of all features expect the smartphone-to-vehicle orientation. The figures show that the most distinct distributions originate from the clusters associated with the smartphone being held by hand. For example, Figs. 8 (b) and (d) illustrate that segments where the smartphone is held by hand generally have higher gyroscope variance and lower tag-smartphone correlation. This is, of course, expected since the smartphone in these segments often will be subject to movements originating from hand motions. Similarly, segments where the smartphone is held by hand tend to be associated with lower speeds (Fig. 8 (c)) and shorter segment durations (Fig. 8 (e)). Most likely, this reflects that users tend to interact more with their smartphones in urban areas when they e.g., briefly need to use navigation apps or call someone who they will meet. Alternatively, the speed difference may be caused by increased phone use when drivers are in slow traffic. Finally, we note that Fig. 8 (a) indicates that the accelerometer variance is not significantly affected by the smartphone being held by hand. Presumably, this is because most of the variance stems from vehicle dynamics rather than from hand movements.

As can be seen from Fig. 9 (b), the clustering is, to a large extent, driven by the initial screen state. By comparing Figs. 9 (a) and (b), it can also be seen that there are many segments where the screen is on only during the initial part of the segment. This could happen when e.g., the smartphone goes to sleep mode due to inactivity (console), or when the screen is shut off during a call (held by hand). Moreover, Fig. 9 (b) shows that the initial screen state was on for about 10 [%] of the segments mapped to the label "Cup holder". Obviously, these segments primarily derive from state 9, which had the initial screen state "on" in Tab I.

D. Classification

The presented clustering framework can easily be modified to enable classification of new segments. The simplest way to do this is to construct a k-nearest neighbor (k-NN) classifier based on the presented kernel and the collected data [29]. To illustrate this, we first extracted the feature values (other than the roll and pitch angles) from a randomly chosen data point that was mapped to state fourteen (held by hand; e.g., text messaging) during the clustering. A classifier was then constructed by clustering all data except the data point that was used to set the feature values, and then assuming that the mappings from the clustering were correct. Fig. 10 (a) shows the resulting classification margin of a k-NN classifier with $k = 10$ (for simplicity, k was chosen by visual inspection to achieve a sensible trade-off between smoothness of the decision boundaries and responsiveness to changes in features) as a function of the roll and pitch angles describing the smartphone-to-vehicle orientation. Here, the classification margin is defined as the score (the number of votes among the k-NN) of the true state minus the largest score among the false states. The classification margin can be seen to reach its maximum value in an area that is roughly centered around $\phi = 5\pi/4$ and $\theta = 0$, i.e., the roll and pitch angles that were specified for state fourteen in Table I. The exact values of the classification margin will obviously depend heavily on other features than the roll and pitch angles.

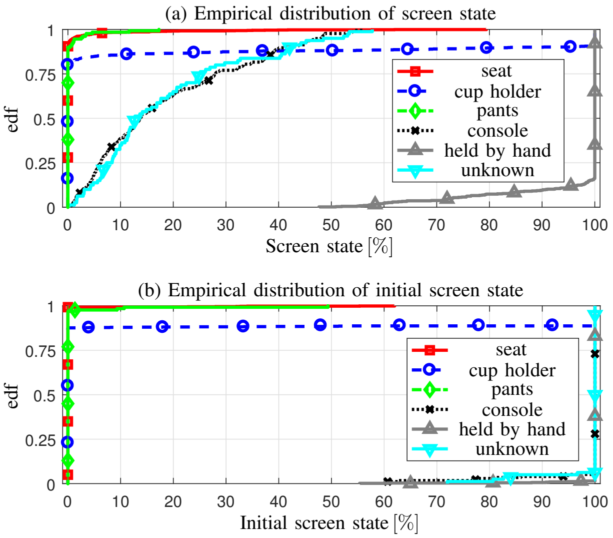


Fig. 9. The distributions of the features (a) screen state and (b) initial screen state.

We would expect that the smartphone display is facing upwards in the vehicle and has its up direction pointing upwards in the vehicle when a user interacts with his smartphone with the screen orientation set to portrait (as presumably is the case

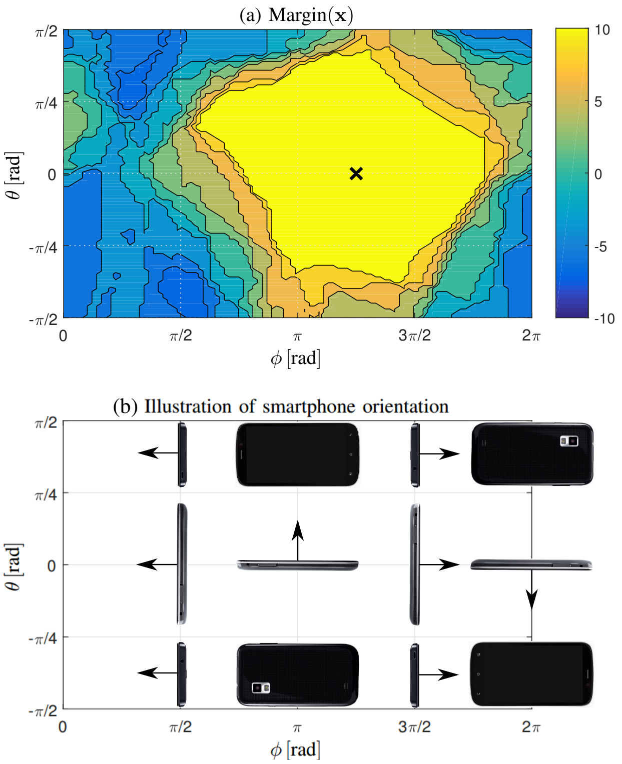


Fig. 10. (a) Example of how the classification margin can depend on the roll and pitch angles of the smartphone-to-vehicle orientation. The true state was state fourteen in Table I. The point $\phi = 5\pi/4$ and $\theta = 0$, associated with state fourteen during the initialization, is marked with a black cross. (b) Smartphone orientations (up to yaw rotations) as seen from a passenger or driver looking at the smartphone in the vehicle's forward direction.

for many segments mapped to state fourteen). To demonstrate this, Fig. 10 (b) displays the smartphone orientations at different points in the roll-pitch plane when the yaw axis is equal to zero. We note that the smartphone display is facing upwards in the vehicle when $[\mathbf{C}_s^b\mathbf{e}_3]_3 = \cos(\phi)\cos(\theta) < 0$, or equivalently, when $\pi/2 < \phi < 3\pi/2$ and $-\pi/2 < \theta < \pi/2$. Similarly, the smartphone display has its up direction pointing upwards in the vehicle when $[\mathbf{C}_s^b\mathbf{e}_2]_3 = \sin(\phi)\cos(\theta) < 0$, or equivalently, when $\pi < \phi < 2\pi$ and $-\pi/2 < \theta < \pi/2$. Here, we have used $\mathbf{e}_i$ and $[\mathbf{e}]_i$ to denote the ith three-dimensional unit vector and the ith element of the vector $\mathbf{e}$, respectively. As can be seen from Fig. 10, the studied data point is indeed the most likely to be classified into state fourteen in the interval $\pi < \phi < 3\pi/2$, i.e., when the smartphone display is facing upwards in the vehicle and has its up direction pointing upwards in the vehicle.

E. Smartphone-based Detection of Harsh Accelerations

Accelerometer-based detection of harsh accelerations has been the focus of a wide range of academic studies on smartphone-based driver safety classification, and is considered to be an important element in many smartphone-based telematics programs [1] (the rate of harsh accelerations is highly correlated with accident rates [35]). However, the effect that the smartphone's placement in the vehicle has on the detection of harsh accelerations still remains to be investigated. To remedy this, accelerometer data from both the tag and the smartphone (from each segment where tag data was available) were processed as follows. To begin with, the accelerometer measurements were rotated to the vehicle frame as described in Section III-A. Then, all sampling instances where 1) the absolute value of the accelerometer measurements in the vehicle's forward direction exceeded the threshold $a^{\text{thres}} = 2\,[m/s^2]$ (while there is no industry standard, similar thresholds have previously been considered in [36]–[38]); and 2) the previously detected braking event from the same sensor occurred more than $\delta t \triangleq 5\,[s]$ ago, were noted as detected acceleration events[7]. The second condition was checked by traversing all samples from the first to the last. This produced sets of time points for detected acceleration events $\{t_n^{\text{tag},i}\}$ and $\{t_n^{\text{smph},i}\}$ for each segment i. Next, treating the events detected using the tag data as the ground truth, the number of false alarms associated with a given placement label d was defined as the sum of events $t_n^{\text{smph},i}$ for which

$$\min_m |t_n^{\text{smph},i} - t_m^{\text{tag},i}| > \delta t \tag{9}$$

where the minimization is taken over all events $\{t_m^{\text{tag},i}\}$ detected using the tag data from segment i, and where the summation is taken over all events $\{t_n^{\text{smph},i}\}$ from segments i that were mapped to placement label d during the clustering. Expressed in words, each detected event $t_n^{\text{smph},i}$ for which no event $t_m^{\text{tag},i}$ that satisfies $|t_n^{\text{smph},i} - t_m^{\text{tag},i}| \leq \delta t$ exists, is counted as a false alarm associated with the placement label of segment i. Similarly, the number of missed detections and true positives associated with placement label d were defined as the sum of events $t_n^{\text{tag},i}$ for which

$$\min_m |t_m^{\text{smph},i} - t_n^{\text{tag},i}| > \delta t \tag{10}$$

and

$$\min_m |t_m^{\text{smph},i} - t_n^{\text{tag},i}| \leq \delta t, \tag{11}$$

respectively, with the same approach to the minimization and summation as for the false alarms. The sensitivity (recall) and precision for a given label d can now be defined as Sensitivity $\triangleq$ TP/(TP + MD) and Precision $\triangleq$ TP/(TP + FA), were MD, FA, and TP denote the associated number of missed detections, false alarms, and true positives, respectively.

As suggested from Fig. 8 (d) (and as should be expected from intuition), Table III shows that segments where the smartphone is believed to be held by hand have a lower detection accuracy than segments of most of the other labels. The only exception is the dummy state "Unknown". Several factors may have influenced the results for this state. For example, we note that when removing the four segments assigned to "unknown" with the largest number of false alarms and missed detections from the computations, the sensitivity and precision take on the values $83.2\,[\%]$ and $47.2\,[\%]$, respectively. In other words, only four segments need to be removed from the computations to make the detection accuracy of the dummy state higher than that of the handheld state. Since the total number of segments

[7] The two major paradigms within driver profiling are rule-based methods (for example defining a harsh braking to have occurred when the absolute acceleration exceeds some threshold) and learning-based methods (defining a harsh braking to have occurred when the sensor measurements are similar to templates of typical measurements during harsh brakings).

TABLE III
ACCURACY OF DETECTION OF HARSH ACCELERATION.

Placement label d	Sensitivity [%]	Precision [%]
Seat	80.6	59.2
Cup holder/Car mount	80.8	51.0
Pants	73.8	66.5
Console	77.3	47.3
Held by hand	73.8	46.3
Unknown	70.1	28.6

assigned to "unknown" is 80 (see Table II), this corresponds to 5 [%] of the segments in this cluster. Given the size of the studied data set, we cannot exclude the possibility that this is caused by inadequate trip segmentations or other problems related to anomalous data. Since the dummy state is designed to collect segments with unexpected smartphone-to-vehicle orientations, segments that are subject to e.g., inadequate trip segmentation (which can lead to deficient IMU alignments) are likely to be mapped to this state.

Last, we emphasize that the results in Table III only illustrate one way in which smartphone interaction interferes with the detection of harsh accelerations. An additional source of interference is related to the removal of data during the trip segmentation described in Section II: As illustrated in Fig. 8 (e), the typical segment duration is much shorter when the smartphone is held by hand. As a consequence, drivers who frequently hold their smartphone in their hand while driving will generally have more periods where the high dynamics of their hand movements make the inertial measurements from their smartphone useless for purposes of analyzing their driving behavior.

V. DISCUSSION AND CONCLUDING REMARKS

Smartphone-based collection of driving data is continuing to gain in popularity, and has been found to reduce claims losses, casualties, congestion, and emissions. Many of the distinguishing features of data collected using smartphone sensors derive from the mobility of the smartphone. On the one hand, this mobility is one of the big impediments to utilizing smartphone-based data in the same way as data from vehicle-fixed sensors. On the other hand, it also makes it possible to extract information on e.g., driver distraction, directly from smartphone sensors. In this article, we examined the problem of inferring the placement of a smartphone within a vehicle.

The considered trips were first segmented at time points when the smartphone is expected to have been picked up by a user. Kernel-based k-means clustering was used to group trip segments into fifteen different clusters, and the results indicated good agreement with behavioral statistics published by the NHTSA. Given that the NHTSA data was collected by manual labeling of video recordings, the clustering method presented here can be seen as an alternative solution with more favorable scalability properties. The most distinctive feature distributions were associated with clusters of segments where the smartphone was held by hand. These segments generally displayed high gyroscope variances, low maximum speeds, low tag-smartphone correlations, and short segment durations. Moreover, we evaluated the extent to which the smartphone being held by hand impairs the ability to detect harsh accelerations, and illustrated how the clustering framework can be reformulated to enable classification.

Several extensions can be made to attempt to increase the accuracy of the presented clustering method. Since a driver's behavior can be expected to be heavily correlated with his or her position, one natural idea would be to include one or several features using GNSS position measurements (this was not done in this study due to privacy issues). Similarly, by using data collected over a long time period it may be possible to profile the behavior of each driver and thereby extract individual patterns that relate the smartphone's placement to driving characteristics, the time of day or week, and the vehicle's position. Although one could try to estimate the position of the smartphone with respect to the vehicle and include it as a feature, it is the authors' belief that only smartphone-to-vehicle positioning methods that rely on pre-installed vehicle infrastructure would be accurate enough to provide any useful information in this context [18]. Obviously, it would also be possible to incorporate labeled trips into the presented framework.

To summarize, we have demonstrated the feasibility of inferring the placement of smartphones in vehicles using information from built-in smartphone sensors. This capability could e.g., be used to assess the utility of collected inertial measurements for the detection of harsh accelerations, for accident reconstructions, or to profile drivers in terms of distracted driving.

REFERENCES

[1] J. Wahlström, I. Skog, and P. Händel, "Smartphone-based vehicle telematics: A ten-year anniversary," *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 10, pp. 2802–2825, Oct. 2017.

[2] J. Wahlström, I. Skog, and P. Händel, "Driving behavior analysis for smartphone-based insurance telematics," in *Proc. 2nd Workshop on Physical Analytics*, Florence, Italy, May 2015, pp. 19–24.

[3] P. Händel, J. Ohlsson, M. Ohlsson, I. Skog, and E. Nygren, "Smartphone-based measurement systems for road vehicle traffic monitoring and usage-based insurance," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1238–1248, Dec. 2014.

[4] S. Tao, V. Manolopoulos, S. Rodriguez, and A. Rusu, "Real-time urban traffic state estimation with A-GPS mobile phones as probes," *J. Transport. Technol.*, vol. 2, no. 1, pp. 22–31, Nov. 2012.

[5] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Smartphone-based adaptive driving maneuver detection: A large-scale evaluation study," *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 9, pp. 2330–2339, Sep. 2017.

[6] C. Zhou, H. Jia, Z. Juan, X. Fu, and G. Xiao, "A data-driven method for trip ends identification using large-scale smartphone-based GPS tracking data," *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 8, pp. 2096–2110, Aug. 2017.

[7] K. S. Huang, P. J. Chiu, H. M. Tsai, C. C. Kuo, H. Y. Lee, and Y. C. F. Wang, "RedEye: Preventing collisions caused by red-light running scooters with smartphones," *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 5, pp. 1243–1257, May 2016.

[8] P. Händel, I. Skog, J. Wahlström, F. Bonawiede, R. Welch, J. Ohlsson, and M. Ohlsson, "Insurance telematics: Opportunities and challenges with the smartphone solution," *IEEE Intell. Transport. Syst. Mag.*, vol. 6, no. 4, pp. 57–70, Oct. 2014.

[9] "The state of usage based insurance today," Mar. 2018, Ptolemus - Consulting Group.

[10] "Infographic: How ready are consumers for connected cars and usage-based car insurance?" May 2017, Willis Towers Watson.

[11] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proc. 6th Int. Conf. Mobile Syst. Appl. Services*, Breckenridge, CO, Jun. 2008, pp. 29–39.

[12] S. Ma, Y. Zheng, and O. Wolfson, "Real-time city-scale taxi ridesharing," *IEEE Trans. Knowledge and Data Eng.*, vol. 27, no. 7, pp. 1782–1795, Jul. 2015.

[13] S. Busanelli, F. Rebecchi, M. Picone, N. Iotti, and G. Ferrari, "Cross-network information dissemination in vehicular ad hoc networks (VANETs): Experimental results from a smartphone-based testbed," *Future Internet*, vol. 5, no. 3, pp. 398–428, Aug. 2013.

[14] S. Gessulat, "A review of real-time models for transportation mode detection," Free University of Berlin, Tech. Rep., Mar. 2013.

[15] X. Su, H. Caceres, H. Tong, and Q. He, "Online travel mode identification using smartphones with battery saving considerations," *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 10, pp. 2921–2934, Oct. 2016.

[16] B. Assemi, H. Safi, M. Mesbah, and L. Ferreira, "Developing and validating a statistical model for travel mode identification on smartphones," *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 7, pp. 1920–1931, Jul. 2016.

[17] J. Wahlström, I. Skog, and P. Händel, "IMU alignment for smartphone-based automotive navigation," in *Proc. 18th IEEE Int. Conf. Inf. Fusion*, Washington, DC, Jul. 2015, pp. 1437–1443.

[18] J. Wahlström, I. Skog, P. Händel, and A. Nehorai, "IMU-based smartphone-to-vehicle positioning," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 2, pp. 139–147, Jun. 2016.

[19] J. Wahlström, I. Skog, R. L. Nordström, and P. Händel, "Fusion of OBD and GNSS measurements of speed," *IEEE Trans. Instrun. Meas.*, vol. 67, no. 7, pp. 1659–1667, Jul. 2018.

[20] S. P. McEvoy, M. R. Stevenson, A. T. McCartt, M. Woodward, C. Haworth, P. Palamara, and R. Cercarelli, "Role of mobile phones in motor vehicle crashes resulting in hospital attendance: a case-crossover study," *BMJ*, vol. 331, no. 7514, pp. 428–430, Aug. 2005.

[21] J. H. L. Hansen, C. Busso, Y. Zheng, and A. Sathyanarayana, "Driver modeling for detection and assessment of driver distraction: Examples from the UTDrive test bed," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 130–142, Jul. 2017.

[22] J. Wahlström, I. Skog, and P. Händel, "Detection of dangerous cornering in GNSS-data-driven insurance telematics," *IEEE Trans. Intell. Transport. Syst.*, vol. 16, no. 6, pp. 3073–3083, Dec. 2015.

[23] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 1st ed. Artech House, 2008.

[24] I. Dhillon, Y. Guan, and B. Kulis, "A unified view of kernel k-means, spectral clustering and graph cuts," University of Texas at Austin, Tech. Rep., Feb. 2005.

[25] W. Sun, J. Wang, and Y. Fang, "Regularized k-means clustering of high-dimensional data and its asymptotic consistency," *Electron. J. Statist.*, vol. 6, pp. 148–167, 2012.

[26] L. Girod, H. Balakrishnan, and S. Madden, "Inference of vehicular trajectory characteristics with personal mobile devices," Sep. 2014, US Patent App. 13/832,456.

[27] R. Larsdotter and D. Jaller, "Automatic calibration and virtual alignment of MEMS-sensor placed in vehicle for use in road condition determination system," Master's thesis, Chalmers University of Technol., Jan. 2014.

[28] D. Q. Huynh, "Metrics for 3D rotations: Comparison and analysis," *J. Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, Oct. 2009.

[29] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2008.

[30] D. Mihai and M. Mocanu, "Statistical considerations on the k-means algorithm," *Ann. University of Craiova, Mathematics Comput. Sci. Series*, vol. 42, no. 2, pp. 365–373, Dec. 2015.

[31] N. Milosavljevic and D. Okanovic, "Automatic parameter selection for clustering of correlated high-dimensional datasets," in *Proc. 2nd IEEE Int. Conf. Inf. Sci., Technol.*, Wuhan, China, Mar. 2012.

[32] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," *Advances in Neural Inf. Process. Syst.*, vol. 17, pp. 1601–1608, Dec. 2004.

[33] G. M. Fitch, S. A. Soccolich, F. Guo, J. McClafferty, Y. Fang, R. L. Olson, M. A. Perez, R. J. Hanowski, J. M. Hankey, and T. A. Dingus, "The impact of hand-held and hands-free cell phone use on driving performance and safety-critical event risk," National Highway Traffic Safety Administration, Tech. Rep., Apr. 2013.

[34] R. Matheson, "Could this app make you a better driver?" Jan. 2016, MIT News.

[35] T. Osafune, T. Takahashi, N. Kiyama, T. Sobue, H. Yamaguchi, and T. Higashino, "Analysis of accident risks from driving behaviors," *Int. J. Intell. Transport. Syst. Research*, Sep. 2016.

[36] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. Gonzalez, "Safe driving using mobile phones," *IEEE Trans. Intell. Transport. Syst.*, vol. 13, no. 3, pp. 1462–1468, Sep. 2012.

[37] R. Vaiana, T. Iuele, V. Astarita, M. V. Caruso, A. Tassitani, C. Zaffino, and V. P. Giofrè, "Driving behavior and traffic safety: An acceleration-based safety evaluation procedure for smartphones," *Modern Appl. Sci.*, vol. 8, no. 1, pp. 88–96, Jan. 2014.

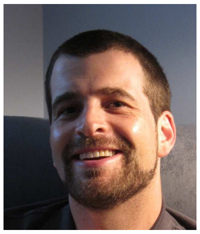
[38] L. Bergasa, D. Almeria, J. Almazan, J. Yebes, and R. Arroyo, "DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors," in *Proc. IEEE Intell. Veh. Symp.*, Dearborn, MI, Jun. 2014, pp. 240–245.

Johan Wahlström received his BSc, MSc, and PhD at KTH Royal Institute of Technology, Stockholm, Sweden in 2013, 2014, and 2017, respectively. He is a postdoc researcher at Oxford University since January 2018. His main PhD research topic was smartphone-based vehicle telematics. In 2015, he spent one month at University of Porto and six months at Washington University in St. Louis as a visiting PhD student. In 2016, he spent two months at the MIT-startup Cambridge Mobile Telematics. He was accepted into the program of excellence in electrical engineering at KTH in 2014, and in 2015 was the youngest recipient of the Sweden-America foundation's research scholarship.

Isaac Skog (S'09-M'10) received the BSc and MSc degrees in Electrical Engineering from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2003 and 2005, respectively. In 2010, he received the Ph.D. degree in Signal Processing with a thesis on low-cost navigation systems. In 2009, he spent 5 months at the Mobile Multi-Sensor System research team, University of Calgary, Canada, as a visiting scholar. In 2011 he spent 4 months at the Indian Institute of Science (IISc), Bangalore, India, as a visiting scholar. He was a recipient of a Best Survey Paper Award by the IEEE Intelligent Transportation Systems Society for the paper In-Car Positioning and Navigation Technologies - A Survey in 2013. In 2015 he received the degree of Docent in Signal Processing from the KTH Royal Institute of Technology. During the fall of 2017 he joined the Automatic Control group at Linköping University.

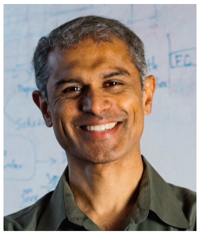
Peter Händel (S'88-M'94-SM'98) received a PhD. degree from Uppsala University, Uppsala, Sweden, in 1993. From 1987 to 1993, he was with Uppsala University. From 1993 to 1997, he was with Ericsson AB, Kista, Sweden. From 1996 to 1997, he was a Visiting Scholar with the Tampere University of Technology, Tampere, Finland. Since 1997, he has been with the KTH Royal Institute of Technology, Stockholm, Sweden, where he is currently a Professor of Signal Processing. From 2000 to 2006, he held an adjunct position at the Swedish Defence Research Agency. He has been a Guest Professor at the Indian Institute of Science (IISc), Bangalore, India, and at the University of Gävle, Sweden. In 2010, he co-founded Movelo AB, a service and technology provider for smartphone-based insurance telematics. Dr. Händel has served as an associate editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He was a recipient of a Best Survey Paper Award by the IEEE Intelligent Transportation Systems Society in 2013.

Bill Bradley received his BSc in Mathematics, Magna Cum Laude, in 1995 from Harvard University, and received his Tripos, with distinction, in Mathematics from the University of Cambridge in 1996. In 2002, he received his PhD in Mathematics from MIT. He has held various positions at Akamai Technologies, the Institute for Defense Analyses: Center for Communications research, Tapir Mathematical Consulting, Lyric Semiconductor, Analog Devices, and is since 2014 the Principal Data Scientist at Cambridge Mobile Telematics.



Samuel Madden received his PhD in 2003 from the University of California, Berkeley. Before enrolling at MIT and while an undergraduate student there, Madden wrote printer driver software for Palomar Software, a San Diego-area Macintosh software company. Professor Madden is also a co-founder of Vertica Systems and Cambridge Mobile Telematics. He has been involved in various database research projects, including TinyDB, TelegraphCQ, Aurora/Borealis, C-Store, and H-Store. In 2005, at the age of 29 he was named to the TR35 as one of the Top 35 Innovators Under 35 by MIT Technology Review magazine. Recent projects include DataHub - a ”github for data” platform that provides hosted database storage, versioning, ingest, search, and visualization (commercialized as Instabase), CarTel - a distributed wireless platform that monitors traffic and on-board diagnostic conditions in order to generate road surface reports, and Relational Cloud - a project investigating research issues in building a database-as-a-service.

Sam is a Professor of Computer Science at MIT and the Director of BigData@CSAIL, an industry-university collaboration to explore issues related to managing data that is too big, too fast, or too hard for existing data processing systems to handle. He is known for contributions to the field of database systems, including widely cited papers on topics related to managing sensor data, column-oriented databases, and databases-as-a-service.



Hari Balakrishnan received his PhD in 1998 from UC Berkeley and a BTech in 1993 from IIT Madras, which named him a distinguished alumnus in 2013. Balakrishnan is a Professor of Computer Science at MIT. His research is in networked computer systems, with current interests in networking, data management, and sensing for a world of ”truly mobile” devices connected to cloud services running in large datacenters. His previous work includes the RON overlay network, the Chord DHT, the Cricket location system, the CarTel mobile sensing system, the CryptDB secure database system, computer-generated congestion control, verifiable Internet routing, the congestion manager, and wireless TCP.

He is an ACM Fellow (2008), a Sloan Fellow (2002), an ACM dissertation award winner (1998), and has received several best-paper awards including the IEEE Bennett prize (2004) and the ACM SIGCOMM ”test of time” award (2011). He has also received a few awards for excellence in teaching and research at MIT: the Harold Edgerton faculty achievement award (2003), and the Jamieson (2012), Junior Bose (2002), and Spira (2001) teaching awards. He was elected to the National Academy of Engineering in 2015.

In 2010, Balakrishnan co-founded Cambridge Mobile Telematics, a company that develops mobile sensing, inferencing, and data analytics to change driver behavior and make roads safer around the world. He was an advisor to Meraki from its inception in 2006 to its acquisition by Cisco in 2012. In 2003, Balakrishnan co-founded StreamBase Systems (acquired by TIBCO), the first high-performance commercial stream processing (aka complex event processing) engine. Between 2000 and 2003, he helped devise the key algorithms for Sandburst Corporation’s (acquired by Broadcom) high-speed network QoS chipset.