Toward Detection of Unsafe Driving with Wearables

Luyang Liu♮, Cagdas Karatas♮, Hongyu Li♮, Sheng Tan§,
Marco Gruteser♮, Jie Yang§, Yingying Chen◇, Richard P. Martin♮
♮WINLAB, Rutgers University, North Brunswick, NJ, USA
♮{luyang, cagdas, hongyuli, gruteser, rmartin}@winlab.rutgers.edu
§Florida State University, Tallahassee, FL, USA
§{tan, jie.yang}@cs.fsu.edu
◇Stevens Institute of Technology, Hoboken, NJ, USA
◇yingying.chen@stevens.edu

ABSTRACT

This paper explores the potential for wearable devices to identify driving activities and unsafe driving, without relying on information or sensors in the vehicle. In particular, we study how wrist-mounted inertial sensors such as those in smart watches and fitness trackers, can track steering wheel usage and inputs. Identifying steering wheel usage helps mobile device detect driving and reduce distractions. Tracking steering wheel turning angles can improve vehicle motion tracking by mobile devices and help identify unsafe driving. The approach relies on motion features that allow distinguishing steering from other confounding hand movements. Once steering wheel usage is detected, it also use wrist rotation measurements to infer steering wheel turning angles. Our preliminary experiments show that the technique is 98.9% accurate in detecting driving and can estimate turning angles with average error within two degrees.

Categories and Subject Descriptors

C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems; H.5.m [**Information Interfaces and Presentation(I.7)**]: Miscellaneous

Keywords

Driving Safety, Driver detection, Smart Watch, Smart Phone, IMU Sensors

1. INTRODUCTION

While the emerging ecosystem of mobile and wearable devices is often viewed as a distraction that can lead to accidents (e.g., [7, 10, 12, 14]), it also presents opportunities to prevent accidents through safety services. Mobile and wearable safety apps differ from conventional built-in automotive safety systems, which are typically constructed as stovepipe systems focusing on a specific risk and employ

dependable systems techniques such as multiple levels of redundancy, quantifiable guarantees on both the timing and paths of state transitions, and precision sensing. There is considerable interest in using mobile devices to deliver softer safety services, however, since they promise low-cost designs that reach a much larger population (e.g., [5, 6, 16]).

Existing solutions either depend on interaction with the vehicle or have struggled to achieve the high event detection accuracies necessary to provide safety services. In order to apply mobile sensing to driver behavior analysis, the devices need to recognize when their user is driving and distinguish that from being a passenger. Prior approach depend on access to audio infrastructure in the car for acoustic ranging [15], access to vehicle dynamics data [13], or NFC radios near the door [11]. Such access may not be widely available and this reduces the chance for quickly reaching a large number of users. Chu et.al. [3] attempt to detect specific movements like entry swing, seat-belt fastening, or pedal press using inertial phone sensors. The accuracy of this approach is very dependent on phone position, however. Several projects have analyzed driver behavior with mobile sensing techniques [1, 2, 4, 8]. These projects evaluate or classify driver behavior by using the embedded sensors and other auxiliary devices like OBD-II. Apps such as [16] fuses information from sensors, GPS and cameras to detect dangerous driving events. Still, they require access to the vehicle bus, remain limited in accuracy, or dependent on favorable phone placement.

We therefore ask whether the emerging quantity and diversity of wearable devices can be exploited to achieve highly accurate sensing and operation that is truly independent of the vehicle. Towards this goal, this paper examines the benefits of wrist-worn inertial data, such as those from smart watches or fitness bands, when combined with phone measurements. Such data is particularly useful for tracking hand movements and can therefore provide information about the driver's vehicle operation—most notably steering wheel usage. To allow more accurate identification of driving, we first develop an inertial detection technique that can distinguish motions when handling a steering wheel from other passenger hand movements. We then extend this technique to also track the turning angle of the steering wheel. This information can allow for more precise vehicle motion tracking than gyroscope data and was previously only available through proprietary interfaces to the vehicle CAN bus. Tracking the usage of the steering wheel can help monitor driving behaviors and identify unsafe driving. For example, steering wheel
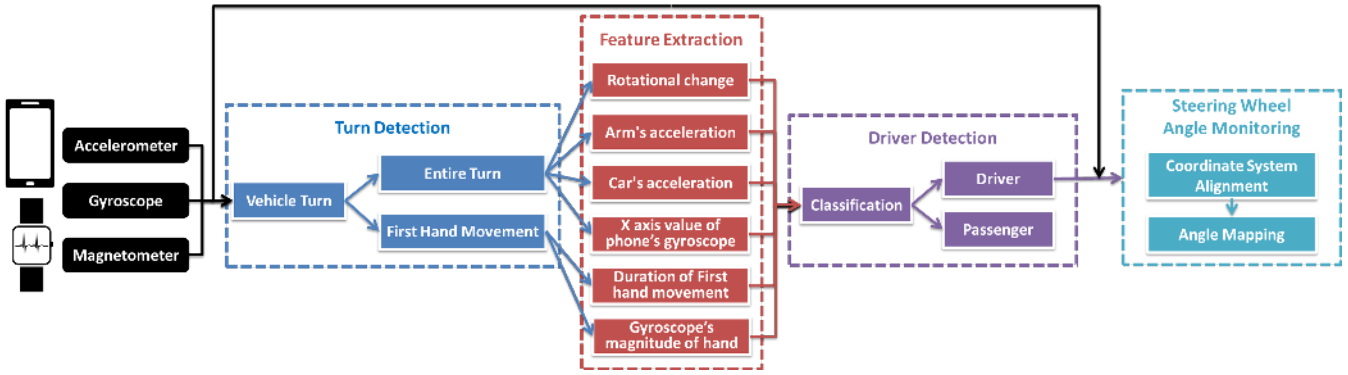
Figure 1: System overview

turning angles could be used in lane departure warning system to warn a driver when the vehicle is about to drift across the lane if the turn signal isn't on. Unsafe driving behaviors such as swerving, understeer or oversteer could be also detected based on the steering wheel turning angles and other inertial or speed measurements.

2. SYSTEM DESIGN

Towards detecting unsafe driving, we design and implement a system to detect driving, and further perform preliminary driving behavior monitoring through steering wheel angle measurement.

For driving detection, an obvious distinction between a driver and a passenger is that the driver's hand will steer the wheel when the car is making a turn. This leads to the intuition of our system being to distinguish driver and passenger through capturing the pattern of steering wheel movements. Our system overview is shown in Figure 1. It first collects data from sensors (including the accelerometer, gyroscope and magnetometer) of both a user's wrist-worn sensing device (e.g. smartwatch) and smartphone. We assume that riding in a vehicle can be detected using existing inertial or location-based techniques [3] and concentrate on differentiating between the driver and passenger. To this end, it performs turn detection using the collected data. Because the most discriminative driver hand's motion tends to happen during turns, our system conducts turn detection by recognizing three different phases of each turn. Based on the output of turn detection, we extract six features to characterize the hand movements. With this feature set, our system can detect whether the user wearing sensing device is a driver or a passenger using a Support Vector Machine (SVM) based classifier. If the user is a driver, the system further estimates the steering wheel angle through a mapping profile.

2.1 Quaternions Background

Processing the inertial sensors data requires transformations between the different watch, phone, car and world (earth) coordinate spaces through rotations. In this work, we use quaternion to represent rotations in three dimensions, a representation that is also directly available through some sensors APIs [9].

There are two important applications of quaternions in our system. The first is to represent rotation of a coordinate frame with respect to another coordinate frame.
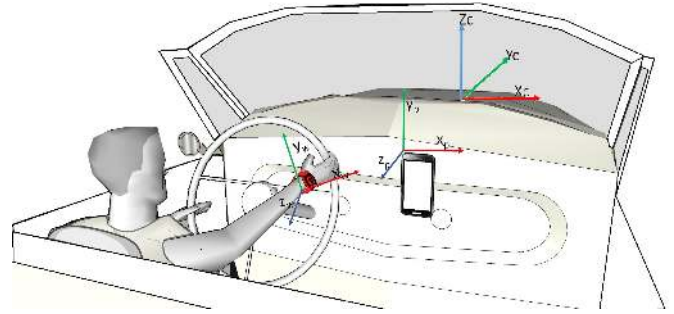

Figure 2: Phone and smart watch can be utilized to detect user and car movements.

Quaternions provide a way to encode the axis-angle representation, and to apply the corresponding rotation to four element vectors. A sensor quaternion vector determines sensor rotation relative to the world coordinate frame, which we define as the x-axis pointing east, y-axis pointing north, and the z-axis pointing up. In Figure 2 the different coordinate systems are illustrated. In our work, the smartwatch's rotation relative to the smartphone's can be found by:

$$\mathbf{q_{wp}} = \mathbf{q_{pe}q_{we}}^{-1}. \tag{1}$$

Here q_{wp} represents smartwatch quaternion relative to phone coordinate frame, q_{pe} and q_{we} represents pone and watch quaternions relative to the earth coordinate frame.

The second application of quaternions is representing a vector defined in world coordinate frame in another coordinate frame by using the Hamilton product as shown below:

$$\mathbf{p_w} = \mathbf{q_{we}p_eq_{we}^{-1}}. \tag{2}$$

Here $\mathbf{p_e}$ represents the input vector in the world coordinate frame, $\mathbf{q_{we}}$ represents the quaternion in the world coordinate frame, and $\mathbf{p_w}$ is the output vector in the watch coordinate frame.

2.2 Turn detection

We develop a *three-phase turn detection* algorithm, which segments different turning periods based on the gyroscope data collected from both smartwatch and smartphone.

The first phase is to find the *vehicle turn period*, wherein the vehicle is actually turning, by applying threshold detection to a sliding time window over the vector magnitude
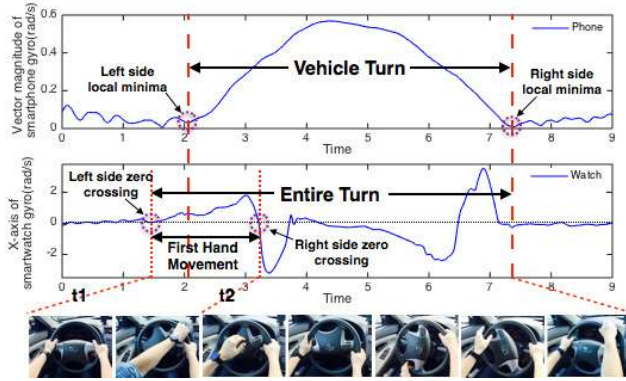
Figure 3: Illustration of the turn detection procedure.

of the smartphone gyroscope. Once all data in the window exceeds the threshold the algorithm detects a turn and expands the left and right sides of the time window to the first local minima, to capture the whole vehicle turn period, as shown in Figure 3.

However, based on our observation, the beginning part of the turning steering wheel movement might be missed in the vehicle turn period, especially at low driving speeds. The pictures at the bottom of figure 3 shows the whole turning steering wheel movement of a right turn, and the beginning part, between t1 and t2, always occurs slightly before the vehicle begins to turn. Thus, the second phase of our turn detection is to detect the first hand movement period through gyroscope x-axis data from the smart watch, which could precisely characterize this movement. The algorithm uses two pointers to search for the first zero crossing point on both the left and right sides of the starting time of the vehicle turn. The segment between left side and right side zero crossing point is identified as the *first hand movement period*.

Finally, the third phase identifies the *entire turn period* as the union of the vehicle turn period and the first hand movement period. The entire turn period now includes the entire steering hand motion. Note that this three-phase turn detection algorithm is also applied to passenger watch and smartphone data. In this case, the hand movement may not exist and the detected period is simply the vehicle turn.

2.3   Feature Extraction

After a turn is detected, our system extracts features from sensor data so that real steering movements can be distinguished from arbitrary arm movements performed by passengers. Our system consists of 6 features in total, which are described as follows and denoted by $f_{featurename}$.

Rotational Change. Since the arm and wrist typically rotate during steering movements, we select watch rotation as the first feature. Directly detecting the rotation of arm using smartwatch data is hard as the watch's rotation (relative to world coordinate frame) is affected by both the arm movement and car's change in heading. We therefore eliminate the effect car rotation during the entire turn period by calculation the rotation of the smartwatch relative to the smartphone ($q_{wp}(t)$). This is performed by applying Eq. (1).

We next need to detect the direction of the rotational movement to distinguish turning steering wheel from other rotational movements, such as eating snacks. Since, the phone can be in an arbitrary rotation in the car, the calculated q_{wp} does not provide information about the direction of arm movement. We can find the rotation of the watch's movement during entire turn relative to watch's rotation at the beginning of the car's turn, *initial coordinate frame*, by using Eq. (1) with q_{wp} and initial quaternion (q_{w0}), the quaternion at the beginning of the turn, as summarized below:

$$q_w(t) = q_{w0}(q_{pe}(t)q_{we}^{-1}(t))^{-1}$$
$$f_{rotChange} = max(abs(\mathcal{T}\{q_w(t)\})) \tag{3}$$

Finally, we use the maximum of the absolute values of the time series of Euler angles in the individual axes as the rotational change feature vector $f_{rotChange}$. Here, $\mathcal{T}\{\}$ refers to the transformation from quaternion time series to Euler angles time series. We use the absolute values, since we are not interested in the direction of rotation (steering left or right) and select the maximum rotation in each axis over the time window of the entire turn period.

Arm Acceleration. Although rotational change feature provides information about the direction and angle of the turn, it does not provide any information about the radius of rotational movement. As a consequence two hand movements can produce same rotational change feature as long as hand is rotated along the same axis and same degrees. To the best of our knowledge, there is no direct way of finding the radius of rotational movement from inertial sensors. Fortunately, acceleration vectors relative to a fixed coordinate frame will be larger when the radius of the rotational movement is large and will be smaller when the radius is small. Thus, we can use acceleration as a feature. However, acceleration vectors are expressed relative to arm's rotation and it will be changing when the arm is rotating. Therefore, these measurements need to be represented in the same coordinate frame before doing any vector operation on them or using them as a feature. We chose arm's initial rotation as the fixed coordinate frame for the sake of simplicity of calculations. Our method calculates watch's acceleration relative to this coordinate frame during entire turn period by first calculating quaternion vectors relative to initial coordinate frame ($q_{winit}(t)$) and then taking Hamilton product (Eq. (2)) of these quaternion vectors with accelerometer vectors ($a_{we}(t)$). Finally, the maximum of calculated acceleration is chosen as the feature ($f_{armaccel}$).

$$q_{winit}(t) = q_{w0}q_{we}^{-1}(t)$$
$$f_{armaccel} = max(q_{winit}(t)a_{we}(t)q_{winit}^{-1}(t)) \tag{4}$$

Car Acceleration. Since the watch's acceleration measurements are largely affected by the car's acceleration, we also need to provide the car's maximum acceleration relative to the watch's initial coordinate frame as a feature ($f_{caraccel}$). The car's acceleration will be equal to the phone's acceleration when the phone is fixed. Therefore, a similar method as in the $f_{armaccel}$ calculation is carried out for coordinate frame transformations with the phone's acceleration

measurements during entire turn period.

$$\mathbf{q_{pinit}}(t) = \mathbf{q_{w0}}\mathbf{q_{pe}}^{-1}(t)$$
$$\mathbf{f_{caraccel}} = max(\mathbf{q_{pinit}}(t)\mathbf{a_{pe}}(t)\mathbf{q_{pinit}^{-1}}(t)) \tag{5}$$

First Hand Movement. As introduced in section 2.2, we also need to extract some features from the first hand movement. The maximum of watch's gyroscope x-axis value is selected to express the maximum rotation velocity during this movement. Besides, the time duration of the movement is also selected, since passenger's hand movements are usually shorter than drivers.

Other Features. To further improve our feature set, we experimented with several additional features like mean or maximum of devices' gyroscope data. By utilizing feature selection, we abandon some of them and choose the watch's gyroscope magnitude mean value of the entire turn period, which can improve the accuracy of our classification model.

2.4 Classification

With the identified feature set, the next step of our system is to distinguish whether a user of the smart watch is a driver. There are multiple methods that could be used for such a binary classification task and here we choose SVM as our classifier—not only because it is one of the most robust ones but also because we empirical found that it achieves the better performance in our task than several alternatives.

During the classification process, we first normalize the feature set by converting it into unit domain. With this normalized feature input, the classifier determines whether the data fits driver or passenger.

2.5 Steering Wheel Angle Monitoring

Once driving is detected, our system uses the gyroscope readings of the smart watch to monitor the steering wheel turning angles. When making a turn, the smart watch on the driver's wrist rotates together with the steering wheel. The steering wheel turning angles thus can be inferred from the rotations of the smart watch.

We first perform coordinate system alignment to convert the gyroscope readings of the smart watch to the steering wheel's coordinate system. Specifically, we first find the coordinate system of the smart watch with respect to the car's by using both the gravity of the smart watch and the acceleration/deceleration of the car when driving straight. Detailed steps can be found in [13]. Since the steering wheel's coordinate system is fixed with respect to the car's, we can easily measure it by using either an angle finder or a smartphone that is aligned with the steering wheel. We next calculate the difference between these two coordinate systems (i.e., quaternions). The gyroscope readings of the smart watch then can be converted to the steering wheel's plane by applying the quaternion.

We experimentally found that the rotation of the smart watch is not the same as that of the steering wheel. This is because the driver bends his wrist in different ways when holding the steering wheel under different turning angles. As shown in Figure 4, before making a turn, the wrist has a large angle to the steering wheel plane. The projected location of smart watch on the steering wheel is thus close to the holding position. After a 90° turn, the wrist angle to the steering wheel plane becomes smaller, which leads to the projected location farther away from the holding position. The rotation of the smart watch is thus smaller than that

Figure 4: Holding position and the projected position of smart watch on steering wheel.

of steering wheel. To address this issue, we collect training data to create a mapping (i.e., angle mapping) between the rotations of the smart watch and the steering wheel under different steering turning angles. This is based on the observation that people usually consistently bend their wrists when turning. The steering wheel turning angles thus can be inferred based on the rotation of the smart watch and the created angle mapping.

3. PERFORMANCE EVALUATION

We evaluate the system in terms of how accurately it detects turns and recognizes the user as a driver for each turn, how long it takes to detect that a user is a driver, and how accurately it estimates the steering angles.

3.1 Driver Detection Accuracy

We conduct our experiments with an Invensense MPU-9150 9-axis motion sensor, which is a prototyping alternative to a wearable device. Each sensor contains accelerometer, gyroscope and magnetometer. We collect data with three such sensors, one is worn on the driver's wrist, one is attached to the passenger's wrist, and the last one is put in the middle cup holder of the car. The sample rate is set as 100Hz in the experiments.

The experiments are conducted under two scenarios. For the first scenario, the driver drives in a controlled area and the passenger always keeps both wrists stationary (now arm movement). For the second scenario, the driver drives on a real road, and the passenger moves (playing with the phone and eating snacks). We collect data for 239 turns in the first scenario and 41 turns in the second scenario, both with passenger and driver inside the car. Our three-phase turn

detection algorithm correctly detects all of these 280 turns resulting in 100% accuracy.

Figure 5 presents the driver detection accuracy by using 10-fold cross-validation when different sets of data are used. Overall, our system achieves 98.9% accuracy by mixing the data collected in both scenarios (i.e., all data). If we only include the passenger and driver data in the first scenarios for training and testing, the accuracy goes to 100% as no wrist movement of passenger interferes with the driver detection. In the second scenario, playing with the phone still yields 100% accuracy; it can apparently easily be distinguished by the classifier. The more challenge scenario is when the passenger is eating. When using data when passenger is eating, our system results in 96.7% accuracy due to some movements of eating produce similar features as steering movements.
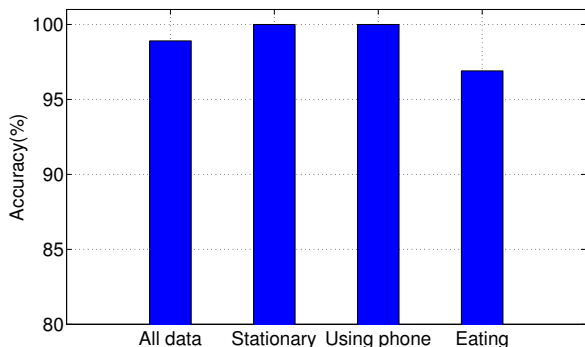

Figure 5: Detection accuracy in different cases

3.2 Detection Delay

Detection delay indicates how much time our system needs to make a decision after starting to drive. Often the first turn occurs when leaving the parking space and our system can detect the driver within the first seconds of the drive. To understand the performance under less ideal conditions, we consider here the case where no steering is necessary to leave the parking space. Since we perform driver detection based on a single turn, the detection delay thus can be defined as the time from starting a vehicle to complete the first turn. The whole detection delay (d_{all}) could be further divided into two parts: the time that starting a vehicle from stationary to the beginning of the first turn (d_s) and the time to complete the first turn (d_t).

Table 1: Detection delay of each drive

Turn No.	1	2	3	4	5	6	7	8
$d_s(s)$	18.55	12.45	18.45	13.90	9.50	8.75	23.20	11.35
$d_t(s)$	6.10	5.40	5.45	7.65	7.60	9.40	5.85	5.40
$d_{all}(s)$	24.65	17.85	23.90	21.55	17.10	18.15	29.05	16.75

For the previous experiments, we have made 8 trips. The detection delay for these 8 cases are shown in table 1. The results show that the detection delay (d_{all}) is very small: ranging from 16.75 seconds to 29.05 seconds, with an average of 21.13 seconds. Moreover, a vehicle usually makes multiple turns to pull out of a parking lot at the beginning of each trip. Our system can thus be expected to have even smaller detection delay in such cases.

3.3 Steering Wheel Turning Angle Estimation

We use two experimental setups to evaluate how accurate the smart watch can be used to estimate the steering wheel turning angles. We use a Samsung Gear Live smart watch, which is equipped with MPU 6515 gyroscope sensor. The sampling rate is set to 200Hz. The car is a Honda Civic Coupe LX. In both experimental setups, we use the Bosch DAF220K Miter angle finder to log the ground truth of the steering wheel turning angles.

In both setups, when the driver is steering the wheel, the car is stationary. Note that when the car is driving on the road, the rotation of the smart watch is a combination of the rotation corresponding to the steering wheel rotation and the actual turn made by the car. Since the turn made by the car can be inferred from the smartphone gyroscope readings, we can subtract the turn made by the car from the overall rotation of the smart watch to get the rotation corresponding to the steering wheel. We leave this evaluation for future work.

In the first experimental setup, we fix the smart watch on the steering wheel to evaluate how accurately the derived quaternion can convert the gyroscope readings of the smart watch to the steering wheel turning angle.

In particular, we turn the steering wheel right for 30° 60° and 90°, respectively. Each turning angle is repeated 10 times. The quaternion is calculated based on the method introduced in the previous section.

Figure 6(a) shows the mean and standard deviation of the angle estimation error. We observe that the average angle error is relatively small. It is less than 1° for the 30° turn, less than 1.5° for the 60° turn, and less than 2° for the 90° turn. Moreover, we find that the average error increases when increasing the steering wheel turning angle. The relationship between the error and the true angle is approximately linear due to the larger turn involving more time which leads to a larger accumulated error. These results show that the quaternion derived in real scenarios can be used to convert smart watch rotation to the steering turning angle with a small error.

In the second experimental setup, the smart watch is worn on the driver's wrist when making turns. We turn the steering wheel both right and left from 5° to 90° with a step size of 5°. For each turning angle, we repeat the experiment 10 times and use half of them as training data and the other half as testing data.

Figure 6(b) shows the histogram of the angle estimation error. We find that in about 70% of cases the angle error is within 2.5°, in about 25% of cases the error is in between 2.5° and 5°, and that the largest error range is from 5° to 7.5° and occurs in only 5% of the cases. Comparing these errors to the case where the watch is fixed on the steering wheel, the error with the wrist-worn watch is larger—presumably due to variations in the drivers wrist postion across turns.

4. DISCUSSION AND CONCLUSION

In this work, we took a step towards enabling activity recognition of unsafe driving using wearable devices. Our work illustrates that strategies which combine motion sensing, machine learning models and classifiers can lead to accurate classification of driving activities. We showed that using a smart watch, we can distinguish steering movements from other movement, such as eating and using a cell phone.
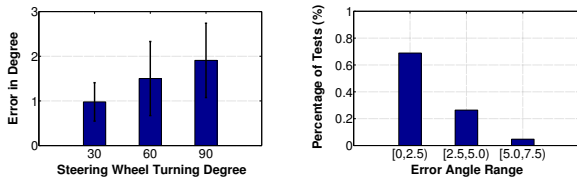
(a) Watch fixed: angle error. (b) Wear watch: range error.

Figure 6: Error of angle estimation when fix the watch on the steering wheel and wear the watch.

In the preliminary evaluation, our prototype achieves 98.9% driver detection accuracy for individual turns, which corresponds to an average driver detection latency of 21.13 seconds after starting to drive. The latency would reduce significantly if one also considers the steering movements necessary to leave most parking spaces. In spite of these promising results, however, many challenges remain towards realizing the vision of automatic unsafe activity detection.

A limitation of our approach is its statistical nature for safety applications. Detection accuracy decreases if the driver steers only with the watch-less hand. A misclassification rate of 1 in 10000, for example, is not acceptable if the consequence was the loss of human life. However, there may be a range of human behavior modification and attention direction uses for which statistical approaches are appropriate. A second major challenge is grounded on our use of machine learning classifiers. The large number of classification strategies and little experience with priors in the field make model construction, training data collection, and model validation a difficult art. Reducing the space of possible techniques will require experience and experimentation with more drivers and vehicles.

Acknowledgment

5. REFERENCES

[1] Rui Araújo, A Igreja, Ricardo de Castro, and Rui Esteves Araujo. Driving coach: A smartphone application to evaluate driving efficient patterns. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 1005–1010. IEEE, 2012.

[2] German Castignani, Raphaël Frank, and Thomas Engel. Driver behavior profiling using smartphones. In *16th International IEEE Conference on Intelligent Transportation Systems*, 2013.

[3] Hon Lung Chu, Vijay Raman, Jeffrey Shen, Aman Kansal, Victor Bahl, and Romit Roy Choudhury. I am a smartphone and i know my user is driving. In *COMSNETS*, pages 1–8, 2014.

[4] Haluk Eren, Semiha Makinist, Erhan Akin, and Alper Yilmaz. Estimating driving behavior by a smartphone. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 234–239. IEEE, 2012.

[5] Mohamed Fazeen, Brandon Gozick, Ram Dantu, Moiz Bhukhiya, and Marta C González. Safe driving using mobile phones. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1462–1468, 2012.

[6] Derick A Johnson and Mohan M Trivedi. Driving style recognition using a smartphone as a sensor platform. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1609–1615. IEEE, 2011.

[7] Sheila G Klauer, Feng Guo, Bruce G Simons-Morton, Marie Claude Ouimet, Suzanne E Lee, and Thomas A Dingus. Distracted driving and risk of road crashes among novice and experienced drivers. *New England journal of medicine*, 370(1):54–59, 2014.

[8] Kun Li, Man Lu, Fenglong Lu, Qin Lv, Li Shang, and Dragan Maksimovic. Personalized driving behavior monitoring and analysis for emerging hybrid vehicles. In *Pervasive Computing*, pages 1–19. Springer, 2012.

[9] Abhinav Parate, Meng-Chieh Chiu, Chaniel Chadowitz, Deepak Ganesan, and Evangelos Kalogerakis. Risq: recognizing smoking gestures with inertial sensors on a wristband. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 149–161. ACM, 2014.

[10] Santokh Singh. Distracted driving and driver, roadway, and environmental factors. Technical report, 2010.

[11] Rainer Steffen, Jorg Preissinger, Tobias Schöllermann, Armin Müller, and Ingo Schnabel. Near field communication (nfc) in an automotive environment. In *International Workshop on Near Field Communication*, pages 15–20, 2010.

[12] Julie Tison, Neil Chaudhary, and Linda Cosgrove. National phone survey on distracted driving attitudes and behaviors. Technical report, 2011.

[13] Yan Wang, Jie Yang, Hongbo Liu, Yingying Chen, Marco Gruteser, and Richard P Martin. Sensing vehicle dynamics for determining driver phone use. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 41–54. ACM, 2013.

[14] Fernando A Wilson and Jim P Stimpson. Trends in fatalities from distracted driving in the united states, 1999 to 2008. *American journal of public health*, 100(11):2213–2219, 2010.

[15] Jie Yang, Simon Sidhom, Gayathri Chandrasekaran, Tam Vu, Hongbo Liu, Nicolae Cecan, Yingying Chen, Marco Gruteser, and Richard P Martin. Detecting driver phone use leveraging car speakers. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 97–108. ACM, 2011.

[16] Chuang-Wen You, Nicholas D Lane, Fanglin Chen, Rui Wang, Zhenyu Chen, Thomas J Bao, Martha Montes-de Oca, Yuting Cheng, Mu Lin, Lorenzo Torresani, et al. Carsafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 13–26. ACM, 2013.