DriveSafe Inspector: A Wearable-based Fine-grained Technique for Driver Hand Position Detection

Huangxun Chen*, Zhice Yang†, Chenyu Huang*, Qian Zhang*
E-mail: hchenay@connect.ust.hk, yangzhc@shanghaitech.edu.cn, chuangak@connect.ust.hk, qianzh@cse.ust.hk
*Department of Computer Science and Engineering, Hong Kong University of Science and Technology
†School of Information Science and Technology, ShanghaiTech University

***Abstract*—This paper presents DriveSafe Inspector, a fine-grained driver hand position monitoring system, which continuously detects a driver's hand position on the steering wheel. The steering wheel is divided into twelve 30° sectors like a clock. Our system can be applied on off-the-shelf hardware and works without extra modification to vehicles. In our system, sensor readings from both a wearable and its paired smartphone are fused to infer hand posture and turning angle between static holding states. With both static holding and dynamic turning information, our system achieves fine-grained hand position prediction in the presence of diverse road conditions and inter-individual differences. The on-road evaluation shows that our system can achieve an average 91.59% hand position detection accuracy with only static information, and can be further improved to 94.63% accuracy combined with dynamic turning information.**

I. INTRODUCTION

According to statistics, there were around 222 million drivers holding a valid driving license in the U.S in 2016. Besides, the number grows around 1 to 4 million per year [1]. To protect both drivers and passengers, it is important for the drivers to hold the steering wheel correctly [2]. Wrongly holding the steering wheel could delay a driver's response to unexpected traffic events, resulting in terrible accidents. However, drivers may not always comply due to fatigue and drowsiness after driving for a long time or simply bad driving habits. A personal monitoring system that accurately detects a driver's hand positions and warns the driver would not only arouse the driver's alertness but also enable him/her to self-improve his/her driving skills [3].

There have been many efforts to monitor drivers [4], [5], [6]. However, these designs require additional devices such as pressure sensors or inertial sensors installed on the steering wheel, which prevents wide adoption in practice. Recently, hand-worn smart devices with embedded sensors have become popular in monitoring peoples activities. Several systems [7], [8], [9], [10] have been developed to track drivers' hand movement using wearables and smartphones. They focused on detecting whether the driver's hands are on the steering wheel or off the steering wheel to use phone, eat, drink and etc. [7], [8] estimated steering turning angles relying on pre-collected angle mapping profiles, which induces inconvenience in practical use. This paper goes a step further to provide an accurate hand position profile on the steering wheel. Compared with hand on/off detection, a fine-grained hand position profile can provide more valuable indicators to infer other dangerous behaviors relevant to driving safety, e.g., overturning, underturning, and overtaking. In addition, these profiles could help driving schools study trainees' learning processes to improve teaching, which would enhance traffic safety in the long term. It is believed that the fine-grained driver hand position detection is an important building block for the emerging automotive IoT ecosystem [11].

Therefore, this paper proposes DriveSafe Inspector, a driver hand position monitoring system to provide fine-grained hand position profile. Our key idea is that when the driver's hands are in different positions on the steering wheel, the gravity would result in different accelerometer readings on the hand-worn device(see Fig.1). Although the idea sounds straightforward, there are some challenges in practice. Firstly, actual routes may cover different regions like city street or mountain path, or may consist of various terrains, such as flat and hilly, which increases the difficulty of sensor fusion. Besides, we found that driver' hands are more flexible and have more motion variance in some positions on the steering wheel, which results in more ambiguity between different hand positions and reduces the system's performance. We also need to deal with inter-individual differences. People grasp the steering wheel in their own ways due to different hand sizes and strengths, thus we should consider such differences in the design to make the system work for different drivers.

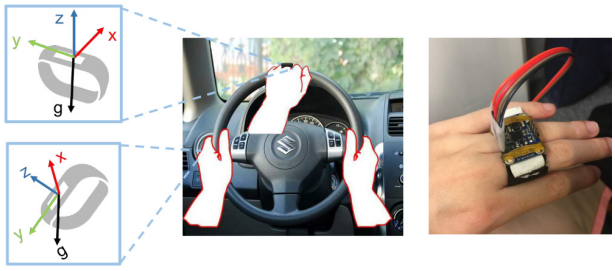


Fig. 1. The basic idea: gravitational acceleration $\vec{g}$ in wearable coordinate frame varies when a driver hand is in different positions. When a driver's hand is at the 12 o'clock position, the z-axis value is the major component of $\vec{g}$; moving to the 9 o'clock position, the y-axis value become the major component of $\vec{g}$

To handle the above challenges, our system first acquires 9-axis sensor readings from the wearable and its bluetooth-paired smartphone and conduct coordinate alignment to eliminate the various road condition's interference. Then we extract the gravity acceleration in the wearable's coordinates to detect different hand positions using personal customized classifiers. Finally, turning angles between two static holding positions are measured to mitigate the effect of inter-positional difference.

To sum up, this paper makes the following contributions:

- This paper proposes DriveSafe Inspector, a driver hand position monitoring system to provide fine-grained hand position profile whilst driving, which works without the extra hardware modification of vehicles.
- DriveSafe Inspector achieves a good performance under diverse road conditions and different drivers by leveraging sensor fusion and learning techniques. Besides

that, the system measures turning angles to mitigate the effects of inter-positional differences and further improves detection accuracy.
- We implement the DriveSafe Inspector prototype using off-the-shelf components and commodity smartphone. The system achieves an average of 94.63% detection accuracy in the on-road evaluations with different routes and drivers.

II. System Design

This section first gives an overview of our system, a wearable-based fine-grained driver hand position detection system, then elaborates on the detailed designs including the system preliminaries, classifier-aided hand position detection and fine-tuning prediction through measurement of turning angle.

A. System Overview

Our system basically leverages ring wearables with an in-vehicle smartphone to infer a driver's hand position on the steering wheel.

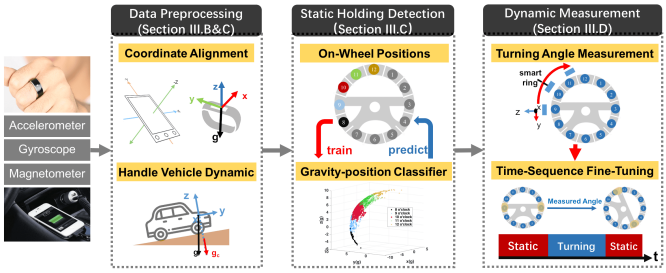


Fig. 2. System Overview.

As shown in Fig.2, our system acquires 9-axis sensor readings (i.e., accelerometer, gyroscope, and magnetometer) from the ring wearable and its bluetooth-paired smartphone. These readings then enter the data preprocessing module including coordinate alignment and dynamic vehicle handling. Both devices produce readings under their own predefined coordinate frame, thus conducting coordinate alignment to align their readings is required. Next, in-vehicle phone readings, which represent the vehicle dynamics, are utilized to eliminate the interference caused by the various terrains in real routes. The preprocessing module produces the pseudo-gravity w.r.t. wearable coordinate frame (elaborated in Section II-C), and the gyroscope reading without a vehicle turning component (i.e., the gyroscope readings correspond to the steering wheel being turned by hand). In the training phase of the system (dealing with inter-individual difference), extracted pseudo-gravity readings would be used to train the classifier for the current driver. In the testing phase, firstly, the system utilizes the classifier to predict hand position based on the extracted pseudo-gravity readings. Secondly, the system estimates turning angles based on the preprocessed gyroscope readings, and leverages the constraints between static holding states and the turning angle between them in time sequence to conduct prediction fine-tuning. In summary, our system integrates both static and dynamic turning information to provide a fine-grained hand position profile.

B. System Preliminaries

The hand positions on the steering wheel are defined as shown in Fig.3, where the steering wheel is divided into twelve 30° sectors and each position occupies one 30° sector. In our system, the position labels (blue circles in Fig.3) remains in place even if the steering wheel (gray part in Fig.3) is turned, e.g., when the steering wheel rotate 120° clockwise, the label "1" would be in the previous position of label "9" relative to the steering wheel (see the middle of Fig.3).

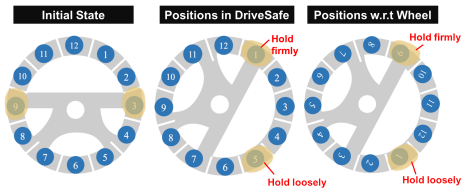


Fig. 3. Position labels in our system and positions relative to the steering wheel. The leftmost figure shows the initial 9-3 o'clock holding state. The middle figure shows the position labels in our system, which remain in place even if the steering wheel is turned, while the rightmost figure shows the positions relative to the steering wheel, which turn along with the steering wheel. When turning more than 90°, one hand would loosely hold the wheel over the driver's lap and wait to take over the wheel soon after.

We adopt this position system because it is commonly used in the handbook which instructs people on how to hold a steering wheel and master push-and-pull and rotational steering, e.g., hold the wheel at "9-and-3" when the car is moving straight; for the push-and-pull steering method, pull the steering wheel down to let the other hand take over at "5" or "7" o'clock and then push the steering wheel up to let the other hand take over at "1" or "11" o'clock; for rotational steering, turn the wheel with your top hand to "3" or "9" o'clock then bring your bottom hand up to "1" or "11" o'clock. Thus, our system can be utilized to help check whether the driver is proficient in the recommended methods.

In the following, we introduce background preliminaries for sensor fusion in our system. The basic sensor fusion in our system is coordinate alignment, which mainly contains two step. Firstly, each device measures its own magnetometer and accelerometer readings to find its rotation matrix with respect to the earth coordinate frame. Secondly, given the rotation matrices calculated before, the system obtains a rotation matrix that maps wearable sensors readings to in-vehicle smartphone's coordinate frame. The well-known quaternion is used to represent object orientation and rotations in three dimensions. A sensor quaternion vector determines sensor rotation relative to the earth's coordinate frame (i.e., the x-axis pointing east, y-axis pointing north, and the z-axis pointing up), which can be calculated given magnetometer and accelerometer readings [12]. Then the wearable rotation relative to the vehicle (or smartphone) coordinate frame can be figured out: $q_{wv} = q_{we}q_{ve}^{-1}$,where q_{wv} represents wearable rotation relative to the vehicle coordinate frame, q_{ve} and q_{we} represents vehicle and wearable rotation relative to the earth coordinate frame. Given the quaternion, a vector in a vehicle coordinate frame can be transformed into a wearable coordinate frame as follows: $p_w = q_{wv}p_vq_{wv}^{-1}$, where p_v represents a vector in a vehicle coordinate frame, and p_w is the transformed vector in a wearable coordinate frame.

C. Classifier-Aided Hand Position Detection

As shown in Fig.1, our system utilizes different gravity acceleration in a wearable coordinate frame to differentiate various hand positions, i.e., when the driver holds various positions on the steering wheel, gravity poses different effect on three axes of the wearable coordinate system since the wearable coordinate system has various postures w.r.t. vehicle coordinate system. However, it is not accurate to adopt original earth gravity acceleration. Fig.4 shows a car on the flat and a slope respectively, where vehicle coordinate frame is defined the same as an in-vehicle phone's coordinate frame. When on the flat, the direction of gravitational acceleration, denoted as $\vec{g}$, is parallel to the z axis of vehicle coordinate frame, while on the slope, there is an angle between $\vec{g}$ and the z axis, which means even if the driver holds the same position in two situations, the mapped $\vec{g}$ in wearable coordinate frames are still different.

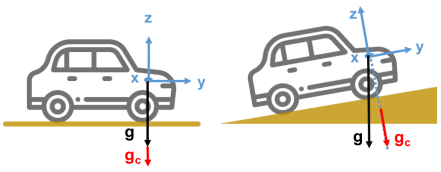


Fig. 4. Vehicle on different terrains: flat and slopes. $\vec{g}$ denotes the gravitational acceleration, while $\vec{g}_c$ denotes the pseudo-gravitational acceleration, which is a unit-length vector aligned with z axis of the vehicle coordinate frame.

What meets our system's requirement is $\vec{g}_c$ in Fig.4, which is a unit-length vector aligned with z axis of vehicle coordinate frame. Since the rotation matrix between the wearable and vehicle is obtained through coordinate alignment, $\vec{g}_c$ can be transformed into the wearable's coordinate frame. Now on the flat or slope, if the driver holds the same position of the steering wheel, transformed $\vec{g}_c$ in the wearable's coordinate frame are the same, while if the driver holds different positions, transformed $\vec{g}_c$ are different. Compared with original gravity $\vec{g}$, pseudo-gravity $\vec{g}_c$ in the wearable coordinate frame could be used to distinguish different hand positions without interference induced by terrains.

Our detailed approach for classifier-aided hand position detection works as shown in Procedure 1. The principle to obtain transformed pseudo-gravity(Line 1-3) is elaborated in Section II-B. AHRS refers to the standard quaternion-based algorithm for attitude and heading reference system. Given inertial readings, we can obtain the rotation matrix of the vehicle coordinate relative to the wearable coordinate R_{vw}. Vector $\vec{g}_c$ in vehicle coordinate frame, i.e., [0,0,1], can be transformed into the wearable coordinate by multiplying R_{vw}. Next, the system identifies static durations of the driver's hand(Line 4-9). As shown in Fig.5, the driver either holds the steering wheel or rotates it in the hands-on duration. Thus we firstly use the variance filter to identify static durations. There are two parameters for the variance filter, window size T_w and variance threshold V_{th}. A T_w duration with the variance of transformed pseudo-gravity smaller than threshold V_{th} would be identified as one static holding state. These two parameters control the system sensitivity to detect static states. If choosing too small a window size T_w or too high a variance threshold V_{th}, some slow turning duration may be regarded as static holding states by mistake; while if choosing too large a window size T_w or too low a variance threshold V_{th}, some static holding durations may not be identified correctly. Our system chooses a 0.5s window size and a 0.01 variance threshold by experiment to handle such a trade-off. For the hands-off period, we borrow the hand on/off detection technique proposed by Karatas el al. [7], and then exclude all hands-off durations so that our approach can focus on hands-on duration analysis. In the training phase(Line 10-11), we train the random-forest classifier using the partial (25% in this paper) driving profile of each driver. Ground truth collection is done by using a smartphone fixed on the car seat to record videos of the driver's hand positions and motion over his/her shoulder, which would be synchronized with the sensor readings offline. Our method only needs video recordings in the short-time initial training phase so that avoid violating the user's privacy in a long-time video recording. In the testing phase(Line 12-13), the system uses the classifier to predict hand positions. Besides, random-forest classifier also outputs the probability of prediction, i.e., the confidence of classifier for this prediction, which is used for fine-tuning prediction in the next section.

Procedure 1 Classifier-aided Hand Position Detection

Input: Wearable readings: $\{Acc_w, Gyr_w, Mag_w\}$, In-vehicle phone readings: $\{Acc_v, Gyr_v, Mag_v\}$

{I. Obtain transformed pseudo-gravity Gvt_w}

1: R_{we} = AHRS(Acc_w, Gyr_w, Mag_w)
2: R_{ve} = AHRS(Acc_v, Gyr_v, Mag_v)
3: $R_{vw} = R_{ve} \cdot R_{we}^{-1}$
4: $Gvt_w = R_{vw} \cdot [0, 0, 1]^T$

{II. Identify static states $State_w$ in time sequence Gvt_w}

5: Set $variance\_filter$: window = T_w, threshold = V_{th}
6: $Gvt_w = \{(x_i, y_i, z_i)(i = 1, 2, \cdots)\}$
7: Denote $\{x_i, \cdots x_{i+T_w-1}\}^T$ as x_{tw}, $\{y_i, \cdots y_{i+T_w-1}\}^T$ as y_{tw}, $\{z_i, \cdots z_{i+T_w-1}\}^T$ as z_{tw}
8: **if** Variance(x_{tw}, y_{tw}, z_{tw}) $\leq V_{th}$ **then**
9: $State_w$.append(Average($\{x_{tw}, y_{tw}, z_{tw}\}$))
10: **end if**

{Traing Phase: Construct Random-Forest classifier RF}

11: Training Set S_{train}: 25% of $State_w$
12: RF = RF.train(S_{train}, hand position labels from videos)

{Testing Phase: Predict hand positon with RF}

13: Testing Set S_{test}: Remaining 75% of $State_w$
14: L', $Proba$ = RF.predict(S_{test})
15: **return** Predicted Label L', Confidence Level $Proba$

D. Fine-tuning Prediction through Turning Angle Measurement

In the on-road evaluation, it is found driver' hands are more flexible and have more motion variance in some of the steering-wheel positions, which results in more ambiguity between different positions and reduces the system performance. Besides, it is noticed that drivers sometimes hold positions located close to the border of two position sectors, where the classifier has unsatisfying performance.

Based on the driving behavior analysis (Fig.5), we discover an opportunity to improve detection accuracy. Drivers generally hold the steering wheel, then turn it to a certain angle and may keep it there for a short time, then return to the original position. The turning angle and its two adjacent holding states should satisfy a certain condition in the time-domain sequence, e.g. 9 o'clock plus the 60 ° clockwise angle moves to 11 o'clock. The detailed approach of fine-tuning prediction works as illustrated in Procedure 2.

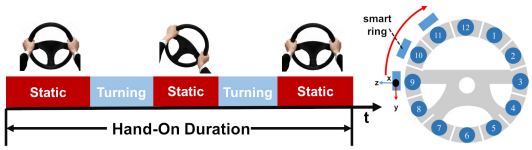


Fig. 5. In hand-on duration, the driver either hold the steering wheel or rotate it. One turning angle and its two adjacent holding positions in time sequence should satisfy a certain conditions.

Procedure 2 Framework for Prediction Fine-tuning

Input: Testing set S_{test} with prediction L' and confidence level $Proba$

{I. Extract gyroscope readings for hand rotation Gyr_h}

1: $Gyr_{vw} = R_{vw} \cdot Gyr_v^T$, $Gyr_h = Gyr_w - Gyr_{vw}$

{II. Trigger condition check based on $Proba$}

2: Denote l_i as predicted label x'th entry in S_{test} with confidence level p_i

3: confidence threshold: C_{th}, off-center threshold: O_{th}

4: **for** l_i in S_{test} **do**

5: **if** $p_i \leq C_{th}$ **then**

6: Find the nearest l_j with high-confidence $p_j > C_{th}$

7: In-between angle $A_{turn} = \text{Integrate}(Gyr_h^{i \to j})$

8: **if** $A_{turn} \notin [30 \cdot (l_i - l_j) - O_{th}, 30 \cdot (l_i - l_j) + O_{th}]$ **then**

9: $l_i = (l_j + \frac{A_{turn}}{|A_{turn}|} \lfloor (|A_{turn}| + O_{th})/30 \rfloor) \bmod 12$

10: **end if**

11: **end if**

12: **end for**

13: **return** S_{test} with fine-tuned prediction L_{tune}

To leverage the constraints mentioned above, the turning angle between two static holding states first should be inferred. As shown in Fig.5, when the driver turns the steering wheel, the ring wearable also experiences the same rotation. Thus, gyroscope readings from the wearable are a combination of the angular velocities of hand turning and car turning. To obtain the component of hand turning, our system measures the vehicle's angular velocity by an in-vehicle smartphone and subtracts it from the wearable readings after coordinate alignment (Line 1).

The small blue rectangles in Fig.5 denote the wearable and show its approximate moving trajectory when the driver turns the steering wheel 60° clockwise from the 9 o'clock position. The rotation axis of the wearable is constant, in the center of the steering wheel and orthogonal to the wheel plane. A vector named simultaneous orthogonal rotations angle (SORA) [13] can be calculated through integration on the preprocessed gyroscope reading, whose components are the angles of three simultaneous rotations around coordinate system axes. For the rotation with a constant axis, the rotation angle is equal to the magnitude of the SORA vector. Given previous predictions and estimated turning angles, When the condition for the turning angle and its two adjacent holding states are found unsatisfied, the prediction with low confidence would be tuned to meet the conditions(Line 2-12).

It is found that most prediction errors occur on the prediction with low confidence. Thus, to improve computing efficiency, our system sets a confidence threshold C_{th} to control the trigger timing for the condition check, i.e., the prediction with confidence lower than threshold C_{th} would trigger the angle calculation relative to the previous prediction with high-confidence and adjusts it if conditions are not satisfied. Furthermore, predictions with high-confidence are generally located in the middle of every 30° sector as shown in Fig.3, while predictions with low-confidence are mostly located close to the border. Thus, through experiments, an off-center threshold O_{th} is set to 10 degrees for the condition check, e.g., given a prediction for 9 o'clock position with high-confidence, if its subsequent prediction with low-confidence have clockwise 20° to 40° rotation, it should be at the 10 o'clock position. Through prediction fine-tuning, our system could increase the hand position detection accuracy from 91.59% to 94.63% on average.

III. Evaluation

This section first introduces our experiment setting, then elaborates the detailed evaluation results covering classifier performance and the gain obtained by fine-tuning prediction through turning angle measurement.

A. Experiment Setting

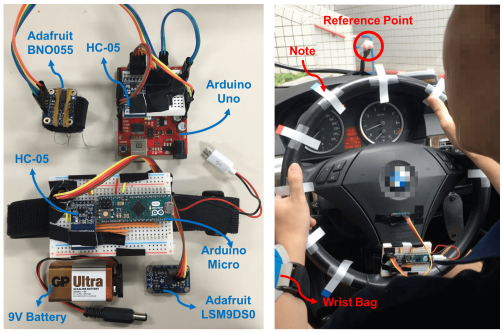


Fig. 6. Evaluation setting.

Our evaluation is conducted on the prototypes as shown in the left of Fig.6. Adafruit BNO055(MEMS 3-axis accelerometer, gyroscope and magnetometer) serves as the ring wearables fixed on a magic tape. The volunteers wear it on their left middle fingers. Sensor readings on wearables are transmitted to the paired smartphone, Samsung Galaxy S5, through an Arduino Uno Board equipped with Bluetooth module HC-05. Arduino Uno, HC-05 chip and 9V battery are put in a wrist bag to be easily worn by the volunteers. This prototype is designed to validate the feasibility of our system. All off-the-shelf components can be manufactured into ring wearables like some commercial products [14]. We place the smartphone in the middle cup holder and let its coordinate system align with that of the vehicle in Fig.4. Another 9-axis motion sensor, Adafruit LSM9DS0 with Arduino Micro and HC-05 module, is attached to the steering wheel for collecting the ground truth of the steering wheel rotation angles, which has been adopted and proved usable as the ground truth in prior work [7] The sampling rates are set to 60Hz on motion sensors and smartphones, which is supported by most commodity devices. A smartphone fixed on the car seat is used to record the ground truth of the driver's hand position and motion over his/her shoulder. The image on the right of Fig.6 is one frame captured while driving. Colorized stick notes divide steering wheel to twelve 30° sectors. we also install a reference point right above the 12 o'clock position to facilitate ground truth labeling.

Evaluations are performed with two volunteers, one male and one female on three different routes as shown in Fig.7,

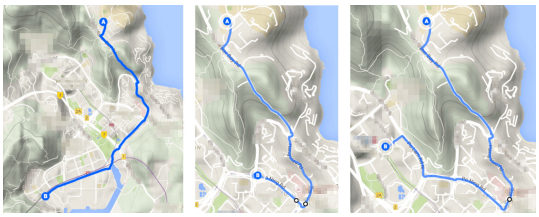
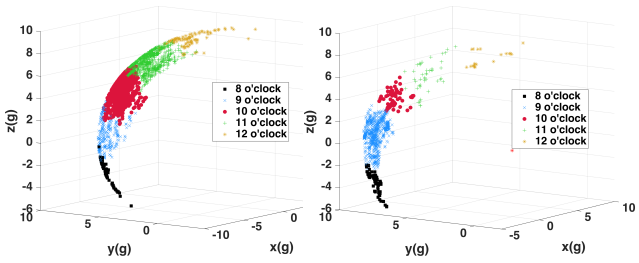


Fig. 7. Maps of three routes that cover different regions (city street or mountain path) and consist of various terrain(flat and slope).

comparative with previous works [8], [7]. The volunteers drive the car in the way they normally would. To ensure diversity, three routes are chosen to cover different regions (city street or mountain path) and consist of various terrains (flat and sloped), which involve smooth curves and also sharp turns. Due to limited experiment resources, the vehicle adopted conforms to the manufacturing standard of most general-use cars. The system performance on a large truck or bus, where the steering wheel may be flatter or larger requires future investigation. We collect data for six trips over half a month on these three routes, and obtain over 3700 samples (discrete hand position predictions) in total for evaluation.

B. Pseudo-gravity Classifier Performance

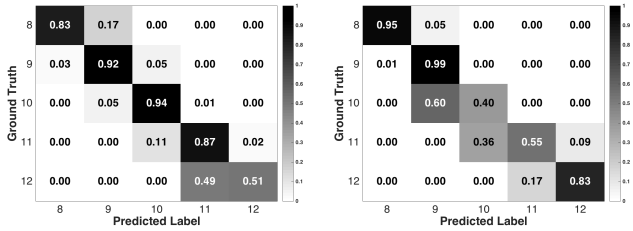
We evaluate the pseudo-gravity classifier performance on two driving profiles respectively.



(a) No.1 Driver Profile (b) No.2 Driver Profile

Fig. 8. Distribution for different hand position samples(denoted by different colors), which contains the 3-axis values of pseudo-gravity in the wearable coordinate frame.

As show in Fig.8, the 3-axis values of pseudo-gravity in the wearable coordinate frame present a distribution along a partial spherical surface for different hand position samples denoted by different colors. Where the same position samples gather together and adjacent position samples are distributed similar to the positions on the steering wheel, reveals the possibility of classification. Moreover, two observed driving profiles shows differences to each other. More comprehensive measurements in [15] also validated such inter-individual differences. They utilized driving a simulator equipped with a 3D motion capture system to record the hand positions of 20 volunteers in 6 driving scenarios involving 7 road geometries. The results showed that most drivers maintained a nearly identical strategy for all road geometries, while inter-individual differences were much larger than intra-individual differences. All data from 8 o'clock to 12 o'clock are used for evaluation, while data of other positions are so scarce(no sample or less than 5 samples) that it makes little sense to build the classifiers with such small samples. A lack of samples on such positions is mainly due to the wearables being worn on the left hand, so that it comes very naturally that most samples collected were located on the left part of the steering wheel while driving on actual routes. However, there is no essential difference for hand detection between positions '8' to '12' and others. The training and testing sets are from different driving trips to ensure the effectiveness of the evaluation.



(a) No.1 Driver (b) No.2 Driver

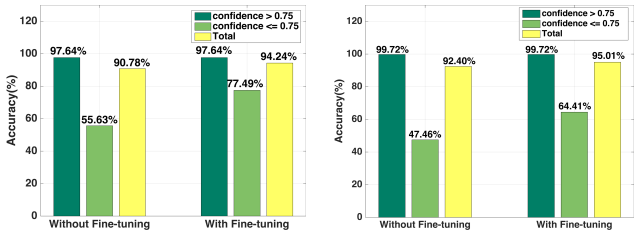
Fig. 9. Confusion Matrices of pseudo-gravity classifiers. Most errors occur within the diagonal alignment.

The pseudo-gravity classifier (Procedure 1) achieves 90.78% overall classification accuracy for the No.1 driver and 92.4% accuracy for the No.2 driver, and the detailed confusion matrices are shown in Fig.9, where x-axis denotes the predicted label and y-axis denotes the ground truth label. It is noticed that most errors occur within the diagonal alignment, which is in accordance with intuition since adjacent positions are more likely to be misclassified with each other. Besides, some positions, such as the 12 o'clock position for the No.1 driver, the 10 and 11 o'clock positions for the No.2 driver, have low accuracy, i.e., they are easily identified as adjacent positions by mistake. The possible reason is that the collected sample number for these position are relatively small compared to other positions. The No.1 driver is accustomed to holding the 10 and 11 o'clock positions, while the No.2 driver is used to holding the 9 o'clock one. Since volunteers are asked to driver the car in their own natural way, it is hard to collect more samples on these positions. Some resample techniques, such as boosting may be leveraged to tackle this problem, which is one of our future works.

C. Fine-tuning Prediction Performance

As introduced in Section II-D, turning angles are measured to enhance the accuracy of prediction from the pseudo-gravity classifier. To illustrate the detailed improvement, confidence threshold C_{th} is set to 0.75 by default. In this setting, there are 15.6% testing data with confidence value lower than C_{th}. The fine-tuning technique improves the prediction on this fraction of data to boost the system performance. The following paragraphs show the improvement in per-person training results.

As shown in Fig.10, the prediction accuracy among samples with a confidence higher than 0.75 are 96.74% and 99.72% respectively, however, the accuracy among samples with confidence lower than 0.75 are 55.63% and 47.46% respectively. By setting the confidence threshold C_{th} to 0.75, the high-confidence prediction part remains unchanged, while the low-confidence part can be improved to 77.49% and 64.41% respectively so that the overall classification accuracy would be raised from 90.78% and 92.4% to 94.24% and 95.01% respectively. Thus, the purpose of the confidence threshold C_{th} is that, on the one hand, the advantages of high-confidence prediction can be preserved, on the other hand, fine-tuning

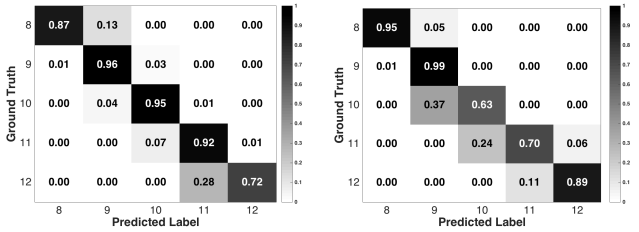


(a) No.1 Driver (b) No.2 Driver

Fig. 10. Detailed improvement through fine-tuning when setting the confidence threshold to 0.75. Prediction fine-tuning achieve 3.46% and 2.61% gain in overall accuracy for two drivers respectively.

efforts can be made focusing on the error-prone part so that unnecessary turning angle computing is avoided.

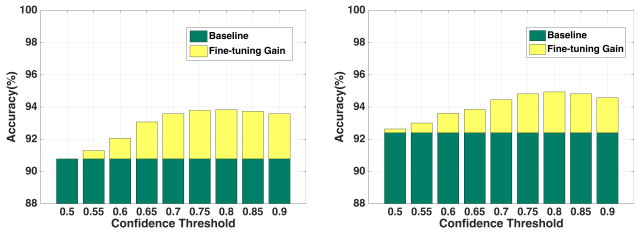
After fine-tuning , the detailed confusion matrices are shown in Fig.11, where x-axis denotes the predicted label and y-axis denotes the ground truth label. Compared with matrices in Fig.9, it is seen that the previous low-accuracy parts are mitigated, which validates the effectiveness of fine-tuning.



(a) No.1 Driver (b) No.2 Driver

Fig. 11. Confusion Matrices after fine-tuning on the previous prediction from pseudo-gravity classifier with confidence threshold $C_{th} = 0.75$

We also evaluate the fine-tuning gain with different confidence thresholds as shown in Fig.12. It is noticed that if the confidence threshold is set too low, the fine-tuning gain becomes tiny since the candidate set for fine-tuning is small. However, it is not recommended to set too high a confidence threshold, which induces unnecessary turning angle computing and checking, and the fine-tuning gain has become saturated or even decreased a little. Fig.12 shows that the best threshold may vary from person to person, and it is generally acceptable to set the confidence threshold between 0.7 to 0.8.



(a) No.1 Driver (b) No.2 Driver

Fig. 12. Fine-tuning gain with different confidence thresholds.

With a trained classifier, the process including prediction using static holding information and fine-tuning using dynamic turning information can be completed within 1 second on the Samsung Galaxy S5, which indicates the system is capable of providing immediate feedback to drivers in real situations.

IV. Conclusion

This paper presents DriveSafe Inspector, a fine-grained driver hand position monitoring system, which continuously detects a driver hand's position on the steering wheel. Our system can be applied on off-the-shelf hardware and works without extra modification to vehicles. To achieve our design goal, sensor readings from both wearable and its paired smartphone are fused to infer the hand position and turning angle between static holding positions. With both static holding and dynamic turning information, our system achieves fine-grained hand position prediction in the presence of diverse road conditions and inter-individual differences. The on-road evaluation shows that our system can achieve , on average, 91.59% hand position detection accuracy with only static holding information, and can be further improved to 94.63% accuracy when combined with dynamic turning information. There is still room for further investigation, e.g., exploring the correlation between a driver's hands position profile and the road layout (i.e., straight road, or turns) to help improve driving skills; exploring features representing unique habitual steering methods to facilitate driver identification.

References

[1] "Number of licensed drivers in the United States from 1990 to 2016." https://www.statista.com/statistics/191653/number-of-licensed-drivers-in-the-us-since-1988/.

[2] "The American Automobile Association. Avoiding Crashes and Emergency Maneuvers(2012)." https://seniordriving.aaa.com/improve-your-driving-skills/handle-unexpected-situations/avoiding-crashes-emergency.

[3] A. Ben-Yaacov, M. Maltz, and D. Shinar, "Effects of an in-vehicle collision avoidance warning system on short-and long-term driving performance," *Human Factors*, vol. 44, no. 2, pp. 335–342, 2002.

[4] M. Klausner and W. Grimm, "Method for detecting the position of hands on a steering wheel," Mar. 28 2006. US Patent 7,019,623.

[5] T. J. Evarts, P. E. Balcom, J. R. Evarts, E. A. Balcom, K. J. Balcom, B. A. Evarts, *et al.*, "Steering wheel hand position sensing device," Oct. 22 2013. US Patent 8,564,424.

[6] L. Liu, H. Li, J. Liu, C. Karatas, Y. Wang, M. Gruteser, Y. Chen, and R. P. Martin, "Bigroad: Scaling road data acquisition for dependable self-driving," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 371–384, ACM, 2017.

[7] C. Karatas, L. Liu, H. Li, J. Liu, Y. Wang, S. Tan, J. Yang, Y. Chen, M. Gruteser, and R. Martin, "Leveraging wearables for steering and driver tracking," in *Proceedings of the 21th Annual International Conference on Computer Communications*, IEEE, 2016.

[8] L. Liu, C. Karatas, H. Li, S. Tan, M. Gruteser, J. Yang, Y. Chen, and R. P. Martin, "Toward detection of unsafe driving with wearables," in *Proceedings of the 2015 workshop on Wearable Systems and Applications*, pp. 27–32, ACM, 2015.

[9] C. Bi, J. Huang, G. Xing, L. Jiang, X. Liu, and M. Chen, "Safe-watch: A wearable hand motion tracking system for improving driving safety," in *Internet-of-Things Design and Implementation (IoTDI), 2017 IEEE/ACM Second International Conference on*, pp. 223–232, IEEE, 2017.

[10] L. Jiang, X. Lin, X. Liu, C. Bi, and G. Xing, "Safedrive: Detecting distracted driving behaviors using wrist-worn devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, p. 144, 2018.

[11] R. Kirk, "Cars of the future: the internet of things in the automotive industry," *Network Security*, vol. 2015, no. 9, pp. 16–18, 2015.

[12] "Open source IMU and AHRS algorithms." http://www.x-io.co.uk/open-source-imu-and-ahrs-algorithms/.

[13] S. Stančin and S. Tomažič, "Angle estimation of simultaneous orthogonal rotations from 3d gyroscope measurements," *Sensors*, vol. 11, no. 9, pp. 8536–8549, 2011.

[14] "NFC Ring." http://nfcring.com/.

[15] J. Schiro, P. Loslever, F. Gabrielli, and P. Pudlo, "Inter and intra-individual differences in steering wheel hand positions during a simulated driving task," *Ergonomics*, vol. 58, no. 3, pp. 394–410, 2015.