

# The Effect of Road Bumps on Touch Interaction in Cars

Sven Mayer<sup>1</sup>, Huy Viet Le<sup>1</sup>, Alessandro Nesti<sup>2</sup>, Niels Henze<sup>1,3</sup>,  
Heinrich H. Bülthoff<sup>2</sup>, Lewis L. Chuang<sup>2,4</sup>

<sup>1</sup>University of Stuttgart, Stuttgart, Germany, {sven.mayer, huy.le}@vis.uni-stuttgart.de

<sup>2</sup>Max Planck Institute for Biological Cybernetics, Tübingen, Germany,  
{alessandro.nesti, heinrich.buelthoff}@tuebingen.mpg.de

<sup>3</sup>University of Regensburg, Regensburg, Germany, niels.henze@ur.de

<sup>4</sup>LMU Munich, Munich, Germany, lewis.chuang@um.ifi.lmu.de

## ABSTRACT

Touchscreens are a common fixture in current vehicles. With autonomous driving, we can expect touch interaction with such in-vehicle media systems to exponentially increase. In spite of vehicle suspension systems, road perturbations will continue to exert forces that can render in-vehicle touch interaction challenging. Using a motion simulator, we investigate how different vehicle speeds interact with road features (i.e., speed bumps) to influence touch interaction. We determine their effect on pointing accuracy and task completion time. We show that road bumps have a significant effect on touch input and can decrease accuracy by 19%. In light of this, we developed a Random Forest (RF) model that improves touch accuracy by 32.0% on our test set and by 22.5% on our validation set. As the lightweight model uses only features that can easily be determined through inertial measurement units, this model could be easily deployed in current automobiles.

## Author Keywords

Touch accuracy; offset correction model; center console; in-vehicle touchscreens; on board entertainment system; car.

## CCS Concepts

•Human-centered computing → Graphical user interfaces; Touch screens; HCI theory, concepts and models; User studies; Ubiquitous and mobile devices;

## INTRODUCTION

Direct touch interaction is one reason for the success of mobile devices such as smartphone and tablets. Their success has since permeated into other device domains including ATMs, vending machines, and tabletops to enable intuitive input and data manipulation. The transportation domain is another area that is increasingly employing the use of touchscreens. Most commercial airplanes are already equipped with touchscreens to support interaction with their media centers. The latest generation of cars provide touchscreens for a wide range of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*AutomotiveUI '18*, September 23–25, 2018, Toronto, ON, Canada

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-5946-7/18/09...\$15.00

DOI: <https://doi.org/10.1145/3239060.3239071>

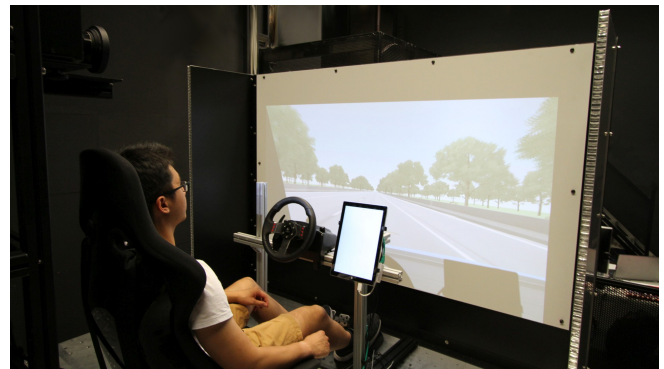


Figure 1. The study setup showing the motion simulator with the car seat mounted on and the simulated center console next to it.

functions including climate control, music playback, and route guidance. Vehicle manufacturers such as Tesla Inc. have prominently integrated touchscreens into their cockpits as a central interaction device. Tesla’s Model S offers a 17” capacitive touchscreen to provide an intuitive and extensible way to interact with the car. With the recent trend towards autonomous driving, a wide range of further use cases are envisioned for in-vehicle touchscreens [19].

Previous researchers thoroughly studied touch interaction on mobile devices. Holz and Baudisch [13] studied the user’s mental model for touch target selection and the resulting touch offset in a lab study. They developed a model that aims to explain how users select targets on stationary touchscreens. Similarly, Henze et al. [10] conducted an “in the wild” study with a large number of users to compensate for the systematic misalignment between the registered position and the center of the aimed target. Previous work also investigated the effect of contextual factors (e.g., the user’s activity) on touch input, e.g., [17]. However, previous work is focused on handheld devices; moreover, they often investigate scenarios where the user is in a stationary setting. Interaction in vehicles is fundamentally different. In cars, for example, the user is not only in forward motion but also experiences additional motion caused by uneven roads. Therefore, touch interaction in moving vehicles can be expected to pose additional challenges that have yet to be addressed by most previous work on mobile device interaction.

Salmon et al. [22] investigated the effect of vehicle motion on user interaction with in-vehicle touchscreens. They showed that driving under high motion conditions increases task completion time for touch-heavy tasks such as writing, as well as exerting a negative effect on perceived usability. To design interaction for in-vehicle touchscreens, we need to understand the influence of any additional motion on touch interaction and the extent to which users can compensate for these effects on their own. Such knowledge will contribute towards the design of interfaces appropriate scale of the user interface (UI). Additionally, it could also enable supporting users in compensating the effects of driving motion on the interaction.

In this paper, we investigate the effect of vertical acceleration caused by road bumps on the accuracy of in-vehicle touch interaction with built-in touchscreens. We present the first study to understand how sudden changes in vertical acceleration affect touch performance. We conducted a controlled lab study in which participants performed target selection tasks while seated on a motion platform. The motion platform simulated the vertical acceleration profiles caused by road bumps while driving with different speeds. While we did not observe a significant effect on task completion time, we found that introducing changes in vertical acceleration increase the offset between the target and the touch point by 19%. Thus, driving over a bumpy road affects touch interaction up to a level where users' performances are noticeably degraded. To improve touch accuracy in cars, we developed compensatory models that reduced the predicted touch offset that was expected to have been caused by road bumps. We present a Random Forest (RF) model that uses six hand-crafted features and reduces the touch offset by 24.2% for our test set and by 31.0% for our validation set. Lastly, we contribute the dataset as well as the scripts to train the models.

## RELATED WORK

The current study investigates the effect of speed bumps on in-vehicle touch interaction. In this section, we review previous work on touch interaction in general, different in-vehicle interaction techniques, and known effects of vehicle motion on the user.

### Improving Touch Accuracy on Mobile Devices

Previous work has thoroughly investigated the inaccuracy of touch interaction, especially on mobile devices. Touch inaccuracy mainly results from a systematic offset due to the user's mental model and the approach of touchscreens that translate touches into a 2D location. Holz and Baudisch [13] investigated the offset for precise target selection and proposed the *projected center model* to reduce offsets from 4 mm to 1.6 mm. Their model takes visual features of the finger into account to match the mental model of the users. Moreover, they proposed a model that explains 67% of the touch inaccuracy that can be attributed to the fat-finger problem [12].

On a larger scale, Henze et al. [10] conducted "in the wild" studies to investigate the systematic offset of touches. Using a game published in Google's Play Store, they recorded touch behavior while players pointed at circles that appeared on the screen. They showed that touch positions are systematically

skewed and developed a compensation function to shift the users' touches to reduce the amount of errors by 7.8%. Similarly, they extended the study to touchscreen keyboards and proposed a model that decreased the typing error rate of users by 9.1% while increasing the performance by 2.2% [11].

While previous researchers have demonstrated how touch accuracy can be improved on mobile devices, they did not focus on situations in which situational movements could also affect touch interaction. Moreover, interacting with the mobile devices (e.g., smartphone or tablet) requires almost no arm movements, compared to an interaction with a car's center console. According to Newton's second law, the net force on an object is equal to the mass of the object multiplied by the acceleration of the object. Since the arm is heavier than the thumb, the net forces are different for interaction with cars' center console so that the results for mobile device interaction of previous work cannot be applied.

### Influence of Motion on Touch Accuracy

Previous work by Salmon et al. [23] has shown that driving motion significantly increases task completion time for touch-heavy tasks such as writing. Furthermore, it has a negative effect on usability of touch interfaces. To compensate the driving motion's effect on touch accuracy, Ahmad et al. [2] used a Leap Motion as a pointing gesture tracker to predict a spatial region of interest, i.e., intended GUI element on the touchscreen in a car. They used a Bayesian framework to calculate the probability for each GUI element and evaluate the course of the trajectory's velocity and finger angle to GUI element from start to end of the motion. Furthermore, they described an evaluation study on the usability and input performance on in-vehicle touchscreens. As a result, they showed the frequency of failed selections, distances by which users miss the intended target, and durations of undertaken free hand positioning gesture to accomplish the selection tasks [3]. To further improve the touch accuracy, they introduced an intent-aware display to simplify free hand pointing gestures by determining the user's intent in the early phase of the pointing gesture. The study describes the benefits of such a display by assessing workload and pointing task duration. As a result, they showed that intent-aware displays significantly reduce workload and the duration of pointing tasks by 20% [1].

As a solution to counteract touch inaccuracy, Rümelin and Butz [21] investigated different interaction methods that are less affected in a driving scenario. In particular, they compared direct touch buttons by making use of proprioception (sense of body parts to each other), haptic guidance, and touch gestures. They found that direct touch provides the best results in terms of task completion time. Haptic guidance was perceived as useful, while gestures reduced the visual attention for the interaction. Moreover, Ecker et al. [6] proposed the use of a pie menu which provides higher usability and efficiency especially due to reduced cognitive load while driving.

Closer to our work, Rahmati et al. [20] presented an anti-shake mechanism against shakes, bumps, and vibration on mobile devices. They used the accelerometer data to predict the magnitude of device displacement and adjust the position of the display area. However, they only shift the screen content

without considering correcting the touch input. Moreover, this was done on a mobile device which does not require moving the arm to perform input as it is the case with cars' center console.

### Summary

In summary, previous papers have proposed models to improve touch accuracy on mobile devices. In comparison to the interaction with cars' center console, only fingers move while arm movements are not necessary. However, the net force is larger on the arm than on the finger, so that the results cannot be applied to touch interaction on cars' center console which is a distant touchscreen that requires arm movement. While researchers also investigated this scenario, they used additional hardware such as the Leap Motion and focused on target selection as a classification problem. In contrast to their work, we develop a regression model to reduce the touch offset for general touch interaction in a driving scenario.

### METHOD

In this study, we used speed bumps as a way of understanding how road features or, in general, vertical acceleration could affect the touch accuracy of in-vehicle touchscreen interactions. To study the effect of speed bumps on touch interaction on a car center console, we conducted an experiment in which we reproduced the vertical acceleration induced by speed bumps that participants might experience during touchscreen interactions. Using a motion platform, we simulated driving over a speed bump at five different speeds. During the simulation, participants selected targets on a touchscreen located in front of them by pointing with their index fingers.

### Design & Task

We used a repeated-measures design with SPEED (5 levels) and BUMP (2 levels: *NoBump* and *WithBump*) as independent variables. We used five car simulations that differed in terms of the simulated speed of the car. We varied the speed (SPEED) using five levels that are expected to commonly occur in daily driving, namely 10, 20, 30, 40, and 50 km/h. We measured four dependent variables: task completion time (TCT), *touch offset*, *touch down events*, and *touch move events*. TCT is the actual time span between the appearance of the visual target and the touch event. The *Offset* is the distance between a target and the location where the touchscreen registered the participant's touch location. *Touch down events* is the number of times the screen was touched in an attempt to touch the target. Lastly, *touch move events* is the number of move events of the finger on the screen. The second two dependent variables will give insights into false inputs and attempts by the participants to correct the initial touch down event.

The participants were asked to rest their arm on a provided armrest. Whenever a red crosshair appeared on the touchscreen, they had to touch the crosshair with their index finger. Following this, they placed their arm back on the armrest. We used a total of 30 different touchscreen positions for the presented crosshair, that was shown on a touchscreen that represented a center console of a car cockpit. To understand the effect of a speed bump on touch accuracy, we introduced the bump stimuli at different timings, in relation to the visual

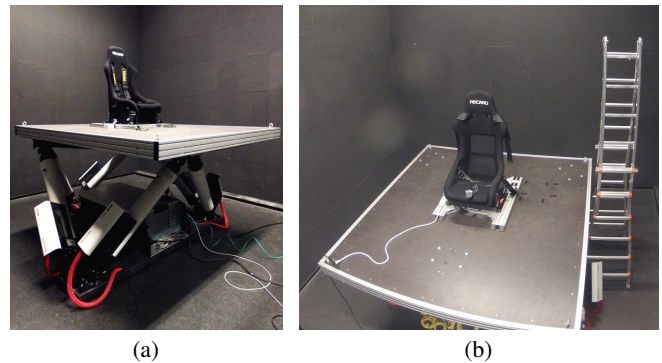


Figure 2. The motion platform without the driving simulation equipment for overview purposes.

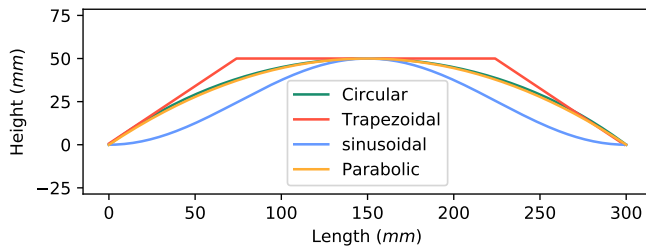
appearance of the crosshair. We used five timings to stretch the bump occurrence across the duration of the arm movement towards the target. For the five timings, we used a range from  $-1s$  to  $3s$  in  $1s$  steps in relation to the appearance to the crosshair.

### Apparatus and Stimuli

To simulate real vehicle motion, we used a BOSCH Rexroth e1500 motion system<sup>1</sup>, see Figure 2 for a setup overview before the projection screen, paddles, and steering wheel were mounted. The motion system is a six-degree-of-freedom (6DOF) Stewart platform with a fixed seat for the participant. It can reproduce horizontal linear accelerations up to  $7m/s^2$  and vertical linear accelerations up to  $10m/s^2$ , while maximum angular velocities are  $34deg/s$ ,  $37deg/s$  and  $41deg/s$  for roll, pitch, and yaw rotations respectively. To simulate the center touchscreen console of a car cockpit, we mounted a tablet to the front of the seat, with a right offset (distance to seat center  $x = 60cm$ ,  $y = 40cm$ ). To simulate the vehicle motion profiles, we used a model of a D-class hatchback vehicle that traveled at constant speed. Participants experienced 5 simulated speeds: 10, 20, 30, 40, and 50km/h. Each speed was simulated for 12min. Throughout the simulation, we systematically introduced road bumps. As is common practice in driving simulation, the platform was used to reproduce linear and angular accelerations. Therefore, while driving at constant speed, the platform did not move. The visual perception of self-motion was induced using a large screen that subtended more than  $90^\circ$  field of view of the participant, see Figure 1.

The 30 target positions were arranged in a  $5 \times 6$  grid on a Microsoft Surface Pro 3. The tablet has a touchscreen with a diagonal of 11.8in. A single target was present at a time. We used red crosshairs as targets. On average, a new target was presented every 15s. A speed bump appeared for 90% of all targets. Thus, 10% of all targets were selected without experiencing a speed bump to provide a baseline and to avoid learning effects.

<sup>1</sup><http://boschrexroth.com/en/xc/industries/machinery-applications-and-engineering/motion-simulation-technology/products-and-solutions/6dof-motion-platform/emotion-1500/index>



**Figure 3.** A set of possible bump profiles of today’s speed bumps. We found that the parabolic profile works best with our motion simulator.

### Motion Profile

Typically, the height of a speed bump is between 50 and 120mm; its length is between 0.6 and 1m and its width span the entire road width[8, 18]. Speed bump profiles commonly employed in road design are circular, parabolic, trapezoidal (or flat-topped) and sinusoidal (or raised cosine). Examples of these profiles are presented in Figure 3.

We created motion profiles to simulate speed bumps on the Bosch motion system. We found that a road bump with a parabolic profile, 5cm height, and 30cm width (see Figure 3) generated accelerations suitable for being reproduced with the motion simulator. Therefore, we chose a parabolic bump.

The accelerations of a vehicle driving over the designed bump were computed using CarSim 8.2.2 (Mechanical Simulation, Michigan, USA). First, we created a customized virtual environment consisting of a straight, flat road. In one condition, we added street bumps to introduce vertical acceleration. The model of a D-class hatchback car was used for the acceleration recordings. The data was sampled at 1000Hz.

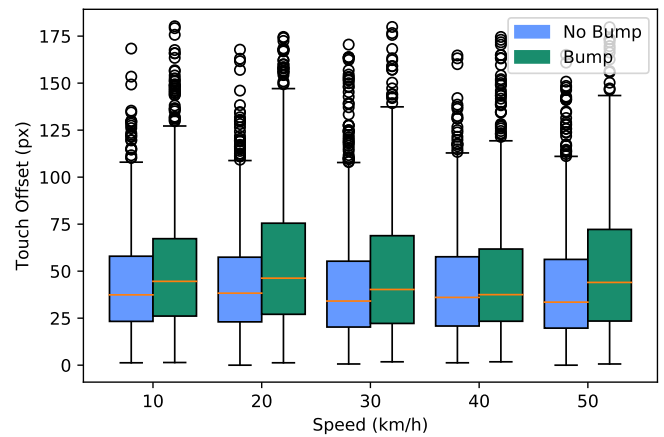
The typical sprung foam and metal seat used in cars attenuates frequencies above about 5Hz [7], while a rigid seat such as the one mounted on the employed motion simulator (see Figure 1) will transmit the accelerations unaltered. Therefore, we filtered the motion profile (the linear accelerations and angular velocities) using a bandpass filter (4<sup>th</sup> order IIR filter, passband between .3 and 5Hz).

### Procedure

Prior to testing, we explained the procedure of the study and provided a general introduction to the motion simulator with an emphasis on safety procedures. We informed participants about their right to quit the study and to immediately stop the simulator by saying “stop.” After signing the consent form, we collected their demographics and guided them on to the motion platform. We explained the target selection task in detail and ensured that participants properly fastened the seat belt. Participants had to rest their arm on an armrest until a target appear. We instructed them to touch the target with their index finger as quickly and as precisely as possible.

### Participants

We recruited 18 participants (9 females, 9 males) through the Max Planck Institute participant database. Participants were aged from 23 to 39 years ( $M = 26.67$ ,  $SD = 3.6$ ). All of them



**Figure 4.** The average touch offset for SPEED vs. BUMP.

reported normal or corrected to normal vision. No participant reported any locomotor disabilities or motion perception difficulties, and all participants were right-handed.

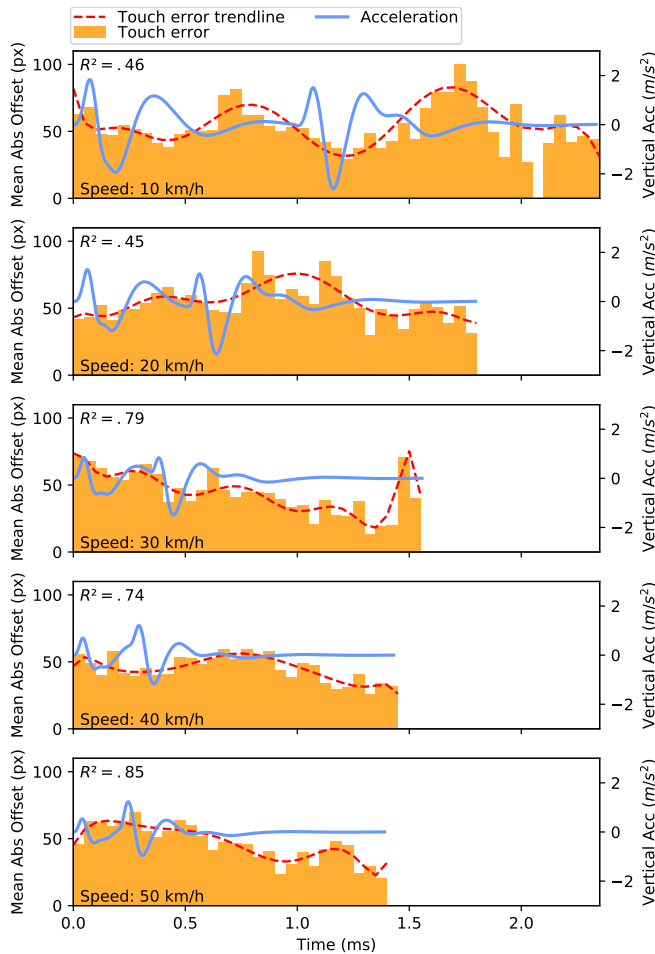
### RESULTS

Due to technical issues while writing the log files, we excluded the data from 2 participants from the analysis. Therefore, the analysis is based on 9,154 touch down, and touch up events and another 52,231 touch movement events. In the following, whenever Mauchly’s test showed that the sphericity assumption was violated in the repeated measures analysis of variance (RM-ANOVA), we report Greenhouse-Geisser (GG) or Huynh-Feldt (HF) corrected p-values.

Next, we analyzed the TCT using a two-way RM-ANOVA. There were no statistically significant main effects (BUMP:  $F_{(1,15)} = 1.679$ ,  $p = .215$ ; SPEED:  $F_{2,421,36,315} = .427$ ,  $p = .789$ ) and also no statistically significant two-way interaction effect (SPEED  $\times$  BUMP:  $F_{4,60} = .914$ ,  $p = .462$ ).

To analyze the effect of a bump on touch performance we needed to ensure that touches in the *WithBump* condition has not been performed before the bump occurred. Thus, we checked if the touch of the participant actually occurred during the movement of the platform. We found that 16.8% of the samples in the *WithBump* condition occurred before the bump. Accordingly, we moved these samples into the *NoBump* condition. Furthermore, 4.67% of the samples occurred when the vertical acceleration was less than  $.05m/s^2$  for more than half a second. While it is possible that the touch could have been influenced by the vertical acceleration, we argue that we cannot draw conclusions given this limited set of data points. Therefore, we also removed these samples from the evaluation. As moving the fastest samples skewed the TCT towards faster input in the *NoBump* condition, the following analysis purely focuses on the touch accuracy.

We first analyzed how often the participants touched the screen for each attempt to touch the target. We conducted a two-way RM-ANOVA to determine if SPEED and BUMP have a significant effect on the number of *touch down events*. The analysis revealed a significant effect of BUMP on the number of *touch down events* ( $F_{1,15} = 14.116$ ,  $p < .002$ ). How-

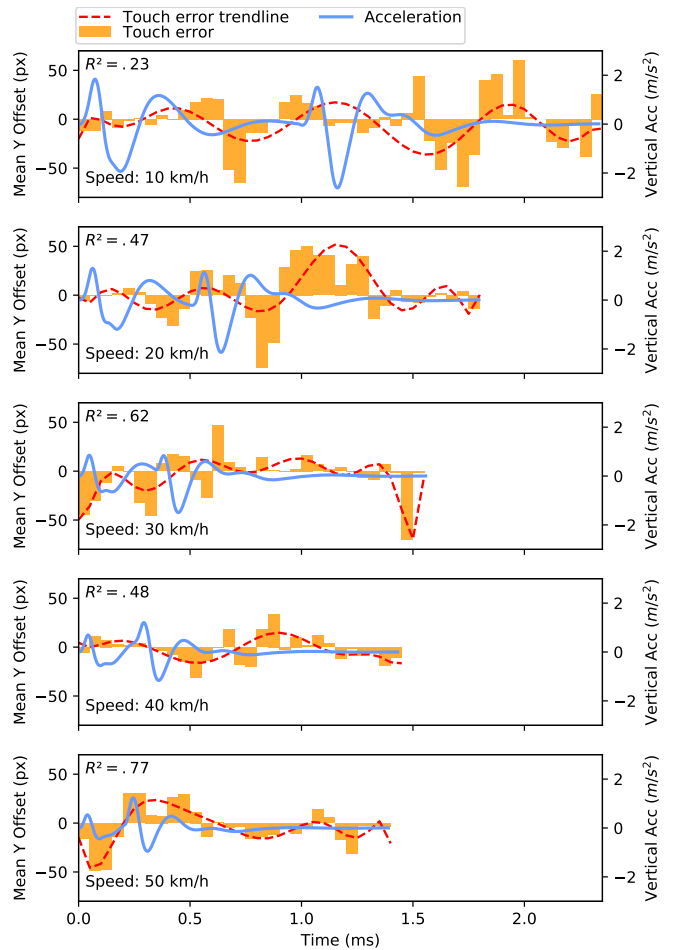


**Figure 5.** The absolute offset of the touch input for the release touch event in the five SPEED conditions. Showing only samples where a bump was present to the participant. The red line is showing the trendline of the touch offset over time.

ever, we found no statistically significant effect for SPEED ( $F_{1,365,20.479} = .727, p = .445, \epsilon = .445$ ) and no statistically significant two-way SPEED  $\times$  BUMP interactions effect ( $F_{4,60} = 2.313, p = .068$ ). Thus, the analysis revealed that participants needed to touch the screen significantly more often in the *WithBump* conditions ( $M = 1.012, SD = .043$ ) compared to the *NoBump* conditions ( $M = 1.028, SD = .045$ ).

Extending from the number of *touch down events*, we analyzed the number of *touch move events* between touch down and touch up. We conducted a two-way RM-ANOVA to determine whether SPEED and BUMP have a significant effect on the number of *touch move events*. Again, the analysis revealed a statistically significant effect of BUMP on the number of *touch move events* ( $F_{1,15} = 8.114, p = .012$ ) but no significant effect of SPEED and also no statistically significant two-way interaction effect ( $F_{1,933,28.993} = 2.375; p = .113, \epsilon = .483$  and  $F_{1,529,22.939} = 1.187, p = .311, \epsilon = .382$ , respectively).

Next, we used a two-way RM-ANOVA to determine whether SPEED and BUMP have a significant effect on the *touch offset* – the distance between the position selected by the participant and the target, see Figure 5. Our analysis revealed a significant



**Figure 6.** The average offset of the touch input in y direction, for the release touch event in the five SPEED conditions. Showing only samples where a bump was present to the participant.

effect of BUMP on *touch offset* ( $F_{1,15} = 68.441, p < .001$ ) (*NoBump*:  $M = 41.8px, SD = 15.7$ ; *WithBump*  $M = 51.9px, SD = 17.9$ ) and a significant two-way SPEED  $\times$  BUMP interactions effect ( $F_{4,60} = 6.043, p < .001$ ), see Figure 4. However, we found no significant effect for SPEED ( $F_{1,933,28.993} = 1.495, p = .236, \epsilon = .614$ ).

## DISCUSSION

We analyzed if the task completion time (TCT) is influenced by speed or a bump that could have given rise to a rapid change in vertical acceleration. Our results show no significant influence on TCT by either speed or by introducing a bump. However, we show that in the bump condition, participants had more touched events on the touchscreen. Thus, participants touch by accident and then corrected their touch input immediately. Moreover, we support this by a second ANOVA which showed that participants also made more movement on the screen when being subject to a bump. We argue that the participants hold on to the screen while making adjustments before releasing their finger from this screen. Thus, we suggest that whenever implementing touch interaction in cars, the action should always be triggered by touch release and not on the first contact. Furthermore, we can also extend

from single touch input to gesture input. As we showed more touch contacts in the bump condition, we argue that gesture recognizers need to consider “idle input times” where the user is just holding to overcome an external force on the car, as well as jitter in the gesture movement due to change in acceleration.

Next, as we now understand that more variation in acceleration decreases touch accuracy, our goal is to build a model which counteracts the touch offset. Therefore, we model the offset based on a set of features to reduce the touch offset. First, we used visual inspection of the average offset behavior for the five SPEEDS to better understand the offset before hand-crafting suitable prediction features, see Figure 5. However, we found that the absolute offset is not enough to understand the behavior. Therefore we also inspected the mean y-offset, see Figure 6. Here, we found that the direction of the offset is actually correlated with the direction of the vertical acceleration.

### OFFSET MODELING

Related work used various approaches to improve touch accuracy on mobile devices. Weir et al. [24] initially used Gaussian processes (GPs) to improve touch accuracy on mobile devices. Buschek et al. [5] used SVMs to improve input accuracy for Back-of-Device (BoD) keyboard on tablets.

Since related work has shown numerous ways to improve touch accuracy without external influences to the touch surface, we will focus on the touch inputs with bump occurrence only. Furthermore, as detecting a bump using an inertial measurement unit (IMU) is trivial, our correction model can easily be applied only to cases in which a bump occurred. Thus, the following models will only be trained on the data where there was an actual bump.

### Model Input Features

For our model to predict the *touch offset* we used the following six hand-crafted features: 1) the speed of the car, 2) the acceleration of the car at the time of the touch up event, 3) the time delta to the last local extrema of the vertical acceleration, 4) the acceleration peak of the last local extrema, and 5+6) the x and y position the point. We included the position of the touch due to the findings by Avrahami et al. [4], who found that the offset is influenced by the bezels of the touchscreen.

### Modeling

We used common machine learning models established in previous work and performed a grid search to find the best hyperparameters for:  $k$ -nearest neighbor ( $k$ NN), Random Forest (RF), Support Vector Machine (SVM), Gaussian process (GP), and Decision Tree (DT). The search for the best hyperparameters was optimized by minimizing RMSE. We used a 64% : 18% : 18% participant-split for training, test and validation set. Resulting in 10 participants for training and three participants for each test, and validation; respectively 1807, 814, and 735 samples. In the following, we report the best hyperparameters based on the test set. All hyperparameters which are not reported are those not changed in respect to

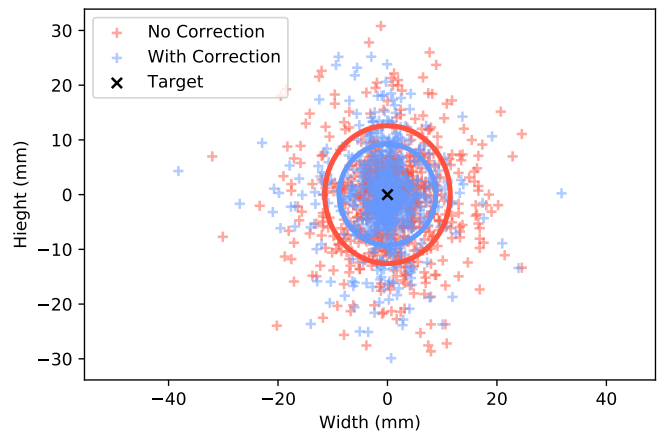
scikit-learn<sup>2</sup> version 0.19.1 together with python 3.6. To prevent overfitting on the grid search we used for the targets, we did not train with the touch points on the middle horizontal grid line and vertical middle grid line.

Table 1 shows the results of the best performing models. Overall, we found that RF performed best in correcting for touch offset during vehicle motion. In detail, the RF improved the x-direction of the touch by 38.6% in absolute mean error and the model to correct the y-offset improved by 23.3%. By combining both models, we improved touch accuracy by 32.%. The effect of the model on the distribution of the touch inputs is shown in Figure 7.

In fact,  $k$ NN performed worse than the baseline of no correction. For  $k$ NN,  $k = 21$  for the x-correction, and  $k = 24$  for the y-correction performs the best. All others performed better on test performance. GP reduced the touch offset by only 1.2% when using a y normalization. SVM performed the same with a reduction of also 1.2% when using an “RBF” kernel for both models and  $C = 100$  for the x-correction and  $C = 1000$  for the y-correction. Next up is the DT with a reduction in offset of 17.5%. For the x-correction, we found that a MIN SAMPLES SPLIT of 5 is optimal with a MAX DEPTH of 8 and for the y-correction we used 20 and 2 respectively. Finally, we used an RF with 19 estimator and maximal 3 features for the x-correction and respectively 26 and 2 for the y-correction. With that, we achieved the best performance and improved touch accuracy by 32.%.

To conclude our model development, we test the model quality using the validation set containing three participants which have not been used for training or testing of the RF model. Here, we found an RMSE of 5.3mm ( $M = 3.3$ ,  $SD = 4.1$ ) for the x-correction and an RMSE of 7.9mm ( $M = 5.6$ ,  $SD = 5.6$ ) for the y-correction. This results in an overall MAE of 6.9mm ( $SD = 5.7$ ) of the validation set, which therefore reduced the touch offset on the validation set by 22.5%.

<sup>2</sup><http://scikit-learn.org/>



**Figure 7. Resulting inputs when applying the Random Forest (RF) model to correct the offset. The figure shows all targets collapsed to a single target. Additionally, to highlight the distribution of the touches, the two ellipses present the area that 95% of the touches fall into. With and without correction conditions are shown in blue and red, respectively.**

Method	X offset in px			Y offset in px			Offset in px	
	RMSE	MAE	SD	RMSE	MAE	SD	M	SD
Baseline (no correction)	38.23	28.64	25.33	50.45	38.62	25.33	53.11	34.46
<i>k</i> -nearest neighbor ( <i>k</i> NN)	35.85	26.20	24.50	53.94	42.36	33.42	54.39	35.20
Gaussian process (GP)	37.55	28.14	24.88	50.08	38.31	32.28	52.49	34.12
Support Vector Machine (SVM)	37.55	28.14	24.88	50.08	38.3	32.28	52.49	34.12
Decision Tree (DT)	22.67	13.71	18.07	48.83	37.60	31.17	43.82	31.30
Random Forest (RF)	27.74	17.59	21.46	41.82	29.62	29.54	38.88	31.75

Table 1. Touch offset errors for the test data with baseline and correction models. All values are reported in pixel,  $10\text{ px} \triangleq 1.9\text{mm}$ .

## OFFSET MODEL DISCUSSION

We showed that the touch performance is significantly influenced by the influence of speed bumps during a typical driving scenario, but not by vehicle speed per se. Therefore, we used visual inspection to understand the parameters that influenced touch performance and derived six features to predictively model touch offset. We performed a grid search on six machine learning algorithms to find the best hyperparameters. We achieved the best test accuracy when using a Random Forest (RF). By using two RF models to reduce x and y offset, we could improve the touch accuracy by 32%. However, when looking at the improvement for x and y independently, we see that overall the correction of the x (38.6% offset reduction) offset is better than y (23.3% offset reduction). We argue that this is not surprising as the speed bump in our setup mainly changed in acceleration y direction.

We showed that DT can outperform RF but only in the x direction. Using DT for the x correction and RF for the y correction separately could, theoretically, improve touch accuracy further. However, as DT performs worse in the y correction where the bump has the most impact on the touch accuracy, we argue that DT cannot model the effect of the bumps properly. As in more realistic scenarios curves and lane changes affect the x direction, we hypothesize that DT would also perform worse than RF in the x direction. Therefore, we suggest not using a combined model using DT and RF.

We modeled five different speeds and one bump profile. We showed that speed did not affect the touch performance. Thus, we argue that when modeling touch behavior in cars it is not important to study different speeds. Moreover, while we initially modeled one specific road bump, the actual acceleration on the participant is indeed more important than both speed and the bump profile. We argue that as there was no difference in the five speed conditions, we choose a bump profile which is sufficient to understand how a range of different bumps would interfere with the touch accuracy.

While the age of our participants ranged from 23 to 39, we acknowledge that for older drivers it can be beneficial to develop a new model. However, we expect that older drivers perform touches with more error. Thus the current model might not compensate for large error, however would also not overcompensate. Thus, we argue that the current model might not be perfect for older drivers but still improves the touch accuracy when using onboard entertainment systems.

Finally, we contribute our offset correction model using RF. The data set collected in this paper is freely available under the MIT license and available on GitHub<sup>3</sup>. The models are using six hand-crafted features, therefore, we provide the scripts for preprocessing, train and validation. These models can be used by researchers in laboratory experiments as well as by car manufacturers in order to improve touch accuracy.

## CONCLUSION

In this paper, we investigated the effect of road bumps on in-vehicle touch interaction. While we found no significant effect of the car’s speed or bump on TCT, our analysis revealed a significant effect of speed bumps on touch accuracy. We show that touch accuracy is significantly lower when users experience a road bump while selecting a target. However, we found no significant effect of the car’s speed on touch accuracy. To improve users’ accuracy, we show that by using a Random Forest (RF) model we can reduce touch offset by 32%. The model corrects the offset based on six hand-crafted features. Finally, the model can be used by manufacturers to increase the accuracy of touch input in cars.

In the future, we aim to improve the accuracy of our offset model based on a more comprehensive data set which can be collected in the wild using real car consoles. This enables the use of deep learning which was shown to be superior in previous work for touch interaction [9, 14, 15, 16]. Further, we aim to extend this research to understand how different changes in acceleration affect the touch performance. We envision this work to inspire touch correction for other vehicles. Similar to interacting with cars’ touchscreens, users interacting with onboard entertainment systems for trains and airplanes can also experience rapid changes in vertical acceleration. We expect that touch interaction in vehicles is generally affected by a vehicle’s motion and should be supported by models that consider this motion to improve touch accuracy.

## ACKNOWLEDGMENTS

This work is financially supported by the German Research Foundation (DFG) within Cluster of Excellence in Simulation Technology (EXC 310/2) at the University of Stuttgart and the MWK Baden-Württemberg within the Juniorprofessuren-Programm. Further, this work was supported by funds from the Max Planck Society and by the DFG within project C03 and C04 of SFB/Transregio 161.

<sup>3</sup><https://github.com/interactionlab/Touch-Interaction-with-Road-Bumps>

## REFERENCES

1. Bashar I. Ahmad, Simon J. Godsill, Lee Skrypchuk, Patrick M. Langdon, and Robert Hardy. 2015. Intelligent In-vehicle Touchscreen Aware of the User Intent for Reducing Distractions: A Pilot Study. In *Adjunct Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '15)*. ACM, New York, NY, USA, 2–7. DOI : <http://dx.doi.org/10.1145/2809730.2809743>
2. Bashar I. Ahmad, Patrick M. Langdon, Simon J. Godsill, Robert Hardy, Eduardo Dias, and Lee Skrypchuk. 2014. Interactive Displays in Vehicles: Improving Usability with a Pointing Gesture Tracker and Bayesian Intent Predictors. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '14)*. ACM, New York, NY, USA, Article 30, 8 pages. DOI : <http://dx.doi.org/10.1145/2667317.2667413>
3. Bashar I. Ahmad, Patrick M. Langdon, Simon J. Godsill, Robert Hardy, Lee Skrypchuk, and Richard Donkor. 2015. Touchscreen Usability and Input Performance in Vehicles Under Different Road Conditions: An Evaluative Study. In *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '15)*. ACM, New York, NY, USA, 47–54. DOI : <http://dx.doi.org/10.1145/2799250.2799284>
4. Daniel Avrahami. 2015. The Effect of Edge Targets on Touch Performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1837–1846. DOI : <http://dx.doi.org/10.1145/2702123.2702439>
5. Daniel Buschek, Oliver Schoenleben, and Antti Oulasvirta. 2014. Improving Accuracy in Back-of-device Multitouch Typing: A Clustering-based Approach to Keyboard Updating. In *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI '14)*. ACM, New York, NY, USA, 57–66. DOI : <http://dx.doi.org/10.1145/2557500.2557501>
6. Ronald Ecker, Verena Broy, Andreas Butz, and Alexander De Luca. 2009. pieTouch: A Direct Touch Gesture Interface for Interacting with In-vehicle Information Systems. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, Article 22, 10 pages. DOI : <http://dx.doi.org/10.1145/1613858.1613887>
7. M. J. Griffin. 2004. Seating Dynamics. In *Handbook of Human Vibration*. Elsevier Academic Press.
8. Ashaish Gupta. 2014. *Study on speed profile across speed bumps*. Ph.D. Dissertation.
9. Niels Henze, Sven Mayer, Huy Viet Le, and Valentin Schwind. 2017. Improving Software-reduced Touchscreen Latency. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA, Article 107, 8 pages. DOI : <http://dx.doi.org/10.1145/3098279.3122150>
10. Niels Henze, Enrico Rukzio, and Susanne Boll. 2011. 100,000,000 Taps: Analysis and Improvement of Touch Performance in the Large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 133–142. DOI : <http://dx.doi.org/10.1145/2037373.2037395>
11. Niels Henze, Enrico Rukzio, and Susanne Boll. 2012. Observational and Experimental Investigation of Typing Behaviour Using Virtual Keyboards for Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2659–2668. DOI : <http://dx.doi.org/10.1145/2207676.2208658>
12. Christian Holz and Patrick Baudisch. 2010. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 581–590. DOI : <http://dx.doi.org/10.1145/1753326.1753413>
13. Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2501–2510. DOI : <http://dx.doi.org/10.1145/1978942.1979308>
14. Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm As an Additional Input Modality on Commodity Smartphones. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 360, 13 pages. DOI : <http://dx.doi.org/10.1145/3173574.3173934>
15. Huy Viet Le, Valentin Schwind, Philipp Göttlich, and Niels Henze. 2017. PredicTouch: A System to Reduce Touchscreen Latency Using Neural Networks and Inertial Measurement Units. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 230–239. DOI : <http://dx.doi.org/10.1145/3132272.3134138>
16. Sven Mayer, Huy Viet Le, and Niels Henze. 2017. Estimating the Finger Orientation on Capacitive Touchscreens Using Convolutional Neural Networks. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 220–229. DOI : <http://dx.doi.org/10.1145/3132272.3134130>



17. Josip Musić, Daryl Weir, Roderick Murray-Smith, and Simon Rogers. 2016. Modelling and correcting for the impact of the gait cycle on touch screen typing accuracy. *mUX: The Journal of Mobile User Experience* 5, 1 (19 Apr 2016), 1. DOI: <http://dx.doi.org/10.1186/s13678-016-0002-3>
18. Margaret Parkhill, Rudolph Sooklall, and Geni Bahar. 2007. Updated guidelines for the design and application of speed humps. In *ITE 2007 Annual Meeting and Exhibit*. Pittsburgh: Institute of Transportation Engineers.
19. Bastian Pfleging, Maurice Rang, and Nora Broy. 2016. Investigating User Needs for Non-driving-related Activities During Automated Driving. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM '16)*. ACM, New York, NY, USA, 91–99. DOI: <http://dx.doi.org/10.1145/3012709.3012735>
20. Ahmad Rahmati, Clayton Shepard, and Lin Zhong. 2009. NoShake: Content stabilization for shaking screens of mobile devices. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE, 1–6.
21. Sonja Rümelin and Andreas Butz. 2013. How to Make Large Touch Screens Usable While Driving. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '13)*. ACM, New York, NY, USA, 48–55. DOI: <http://dx.doi.org/10.1145/2516540.2516557>
22. Paul M. Salmon, Michael G. Lenné, Tom Triggs, Natassia Goode, Miranda Cornelissen, and Victor Demczuk. 2011a. The effects of motion on in-vehicle touch screen system operation: A battle management system case study. *Transportation Research Part F: Traffic Psychology and Behaviour* 14, 6 (2011), 494 – 503. DOI: <http://dx.doi.org/10.1016/j.trf.2011.08.002> Special Issue: Driving Simulation in Traffic Psychology.
23. Paul M Salmon, Michael G Lenné, Tom Triggs, Natassia Goode, Miranda Cornelissen, and Victor Demczuk. 2011b. The effects of motion on in-vehicle touch screen system operation: A battle management system case study. *Transportation research part F: traffic psychology and behaviour* 14, 6 (2011), 494–503.
24. Daryl Weir, Simon Rogers, Roderick Murray-Smith, and Markus Löchtefeld. 2012. A User-specific Machine Learning Approach for Improving Touch Accuracy on Mobile Devices. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 465–476. DOI: <http://dx.doi.org/10.1145/2380116.2380175>