Bootstrapping safe IVIS development with an affordable testing suite

Leonel Mandarino [1] [0000-0003-0135-0304], Carlos Carvajal [2] [0000-0002-8130-2304], Andrés Rodriguez [3] [0000-0002-7600-6041], and Alejandro Fernandez [4] [0000-0002-7968-6871]

[1,2,3,4] LIFIA, Facultad de Informática, UNLP, Calle 50 y 120, La Plata, Argentina
[4] Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, La Plata, Argentina
[1] `leonelmandarino@gmail.com`
[2] `andres.rodriguez@lifia.info.unlp.edu.ar`
[3] `ccarvaj@gmail.com`
[4] `alejandro.fernandez@lifia.info.unlp.edu.ar`

Abstract. In-vehicle information systems (IVIS) represent a growing industry. IVIS were originally built and deployed by car manufacturers, which ensured that they complied with the safety regulations of the car industry. Nowadays, IVIS enter the vehicle in the driver's and passenger's phones. These "nomadic IVIS", which sometimes interact with the car's entertainment system, can escape important safety checks. Any software developer, without training in vehicle safety, can build and distribute IVIS. This reality calls for tools and methods that help software developers conceive safe applications. This article proposes an affordable and reliable testing suite that provides support to developers of nomadic IVIS. The suite takes the form of a simulation and data collection environment, oriented to the rapid prototyping of IVIS. It considers security requirements while maintaining a low technological and economic threshold, to provide easy access to developers compared to expensive physical environments with real vehicles.

Keywords: IVIS, Safety, Prototyping, Evaluation.

1 Introduction

The Strategic Highway Research Program (SHRP 2) [1] project obtained video and audio data from more than 3500 drivers over a 3-year period using information collected by the Naturalistic Driving Study [2]. This project was able to capture information on more than 35 million miles driven, comprising 905 automobile accidents with property and passenger damage. With this information, reports were obtained that indicate a 3.53% prevalence of accidents caused using native car apps, and a 6.4% prevalence of accidents caused by the use of cell phone apps. Therefore, it is estimated that 10 percent of the time, accidents are caused by drivers who are operating electronic devices while driving. It can also be noted that accidents are more frequent in connection with the use of mobile applications [3].

Typically, native car app developers work for large automotive companies and are trained to consider factors such as driver vision, time spent on the app, distracting elements, and colors in the interface, among others. However, there is an increase in the number of applications for smart devices that are intended as information systems for in-vehicle use[1]. For companies not specialized in the automotive industry, it is difficult and costly to tackle the development of tests on specialized tracks and with physical vehicles, especially if they want to work on multiple prototypes and new ideas. To contribute to the solution of this challenge, this article presents and evaluates an affordable and reliable IVIS testing suite.

The rest of this article is organized as follows. First, we provide a brief discussion of related work. We also introduce concepts and mechanisms used to determine the level of cognitive load of a driver. Then, we describe the proposed testing suite. Following, we present the results of a preliminary evaluation of the suite, focused on comparing two different IVIS in terms of cognitive load and driver distraction. Finally, we present the conclusions of the work done, summarize the main contributions and discuss future work.

2 Related work

The Naturalistic Driving Study (NDS) collected information on 3,362 private vehicles, all driven by citizen volunteers who were observed over a 3-year period. The NDS project was mainly based on the belief that a better understanding of driving safety issues can be gained by studying drivers, their behavior and the different factors that affect them, such as weather conditions, the driving environment, electronic devices present in the car, etc. Participants' vehicles were fitted with cameras, radar, and hidden sensors to capture data as they went about their normal activities. Information was obtained on 6,650,519 trips, totaling almost 50 million miles traveled [2]. A study [3] following the DNS project sought to answer "why" and "how" a task is distracting. The authors concluded that most dangerous visual distractions are those in which the driver is exposed to the risk of a situation with abrupt and rapid changes. This is related to the duration of the distraction, being that the longer the driver loses sight of the road, the greater the possibility of finding himself in a difficult situation to face. Among the recommendations to reduce the risk when driving, they considered important to design interfaces that minimize the need for visual interaction on the part of the driver. Another aspect to keep in mind is that the elimination of long glances (more than 2 seconds) will not eliminate distraction problems, since most accidents are the result of small distractions at inopportune moments.

Comprehensive and realistic studies like NDS are valuable and only available to a handful of organizations. They are extremely costly in terms of time and necessary resources. The suite we present in this article incorporates the lessons documented by

[1] https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/

the NDS study but aims at the individual IVIS developer or to the small development company.

In the study "Steering Control in a Low-Cost Driving Simulator: A Case for the Role of Virtual Vehicle Cab" [4], an analysis is made regarding the credibility of the results obtained with low-cost simulators, based on the lack of a driver's cab. Depending on the initial investment for building a simulation environment, they can be classified into three general categories:

- Low cost: they have a gaming wheel, with pedals, several screens, and no cockpit.
- Medium cost: large projection, fixed base, partial or full cockpit, etc.
- High cost: 360º view of the simulator, cockpit of a real vehicle mounted on motion sensors, etc.

The credibility of the results obtained with driving simulators, in general, has been a cause of concern and debate in the scientific community [5, 6]. Some studies have shown that the results obtained in simulators are of the same order and direction to those of real-world data [7, 8]. Other studies have compared low-cost simulators with more expensive and complex simulators to determine whether the former are sufficiently valid to be taken into account [9]. The results indicate that low-cost simulators are good indicators for studying the effects of IVIS on driver behavior.

Skyline [10] is a prototyping platform for user experience development based on a driving simulation. It was developed at Intel Labs, following principles of flexibility, integration, and customization, to enable rapid prototyping of in-vehicle user experiences. The main difference between Skyline and the platform presented in this article is the scope and focus of the simulation environment. In our case, facial recognition sensors are available, and mechanisms were implemented to measure the cognitive load of the driver. Thus, with this information, a more detailed profile of the driver and his behavior can be obtained, to develop safer interfaces oriented to his context of use.

3 Platform overview

As previously discussed, visual distractions of 2 or more seconds prior to the occurrence of an unexpected event such as deceleration and/or braking of another vehicle are significantly more dangerous than distractions of less time or when a precipitating event does not occur. We conclude that the determinants of risk are the presence of an unexpected event and the time we keep our eyes off the road. This is directly related to the level of uncertainty and the reaction time available to the driver, i.e., once he/she regains concentration on the road after a distraction.

With this background in mind, we focused the design of the test suit in the core components depicted in Fig. 1. The test suite consists of two computing nodes, one used by the driver (subject of the test) and one used by the experimenter. The driver's computing node runs the driving simulation environment, the test instruments, and a test controller service. The driving simulation environment offers a first person, 3D, immersive simulation where the test subject drives a vehicle. The gaze tracker detects when the driver looks away from the road and towards the UI of an IVIS. The Detection Response Tasks instrument is used to assess the cognitive load of using an IVIS while driving in a simulated scenario. The simulation environment and the instruments are configured and controlled via the test controller service. The test controller service is also responsible for collecting data from the simulation and the instruments and making it available to the experimenter. The experimenter's node connects to the test controller in the driver's node to plan, launch and monitor experiments. The following subsections discuss the some of these components with more detail.

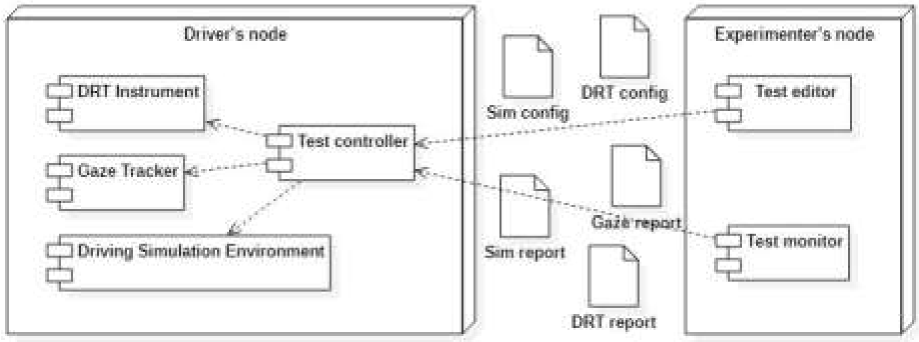

Fig. 1. Architecture overview

3.1 Driving simulation environment

The driving simulation environment executes the driving simulation, where tests take place. In addition, it is the element to which the other components subscribe to configure testing scenarios and to obtain information about the events and actions that occur during tests.

Among the different options available to use as the basis for the simulation environment, it was decided to use CARLA [11]. This simulator, developed at Intel Labs, has the advantage of being a recognized and constantly growing project. The project is Open Source, which made it very easy to integrate it into the suite and then exploit it for the development of new functionalities. In addition to the simulation engine, CARLA provides a vast catalog of digital resources such as urban designs, buildings, vehicles, and pedestrians that enrich the experiments and make them more attractive compared to other simulators. CARLA uses Unreal Engine for the execution and layout of the simulation. The control and configuration of the simulation is given through an API. CARLA allows you to control the amount of Non-Playable Characters (NPCs) distributed around the map; these are cars and pedestrians which CARLA internally calls "actors". Actors are important for the simulation because they bring life, dynamism and a higher degree of difficulty when walking around the map and they can be used to induce risks in the tests.

The simulation environment offers several sensors that allow it to react to events and to know the characteristics of the driving environment. The collision sensor records an event every time the vehicle collides with something in the simulated world. The GNSS (Global Navigation Satellite System) sensor tells the exact location of the vehicle. The lane encroachment sensor records an event every time the vehicle crosses a given street or surface. The different surfaces are defined at the time of map construction. Finally, the RGB cameras stream and/or record the images of the different scenes during the simulation.

3.2 Detection Response Tasks instrument

Detection Response Task (DRT), standardized by ISO 17488:2016 [12], is a method to assess the potential for cognitive distraction introduced by a secondary task performed while driving. DRT consists of a visual stimulus that drivers must respond to while simultaneously driving and performing a secondary task. To assess the cognitive load of the secondary task, one compares the response to the visual stimulus (e.g., response time, hits, and misses) while driving with and without performing the secondary task.

The proposed suite implements the DRT visual stimulus as a small circle of a high contrast color, in the screen, simulating a light on the windshield. The driver must respond to the stimulus by pressing a button on the driving wheel, which causes the stimulus to disappear. The tool records the time it took the driver to respond to the stimulus. Failing to respond while the stimulus is active, will cause the tool to record it as miss. Responding when there are no active stimuli causes the tool to record an error.

The DRT component is configurable. The options to configure are:

- The radius of the circle determined in pixels.
- The color expressed in hexadecimal.
- The time in seconds that the light is on before it turns off (and a miss is recorded).
- The execution mode: this can be manual if you want to turn the light on and off at will, or random if you want the light to appear after a randomly selected number of seconds. An interval can be set for time selection.
- The location of the light; you can choose a fixed location on the screen determined by coordinates on the Cartesian axis. As an alternative to this option a random position on the screen can be chosen. To avoid obstructing the driver's view while moving through the world, you can choose quadrants of the screen where the light can appear.

3.3 Gaze tracker

DRT helps isolate the impact of a secondary task on the driver's attention. The various sensors offered by the simulation (e.g., the crash sensor) help assess the impact of a secondary task on driving. However, none of them can help us tell where the driver was looking at. The Gaze tracker is a low-cost instrument that can record this information. It monitors the user's gaze using a camera. This is very valuable since it complements the objective of the DRT tasks and helps to identify the driver's attention during the tests.

For the development of the Gaze tracker we used the OpenCV library, which provides a great amount of functionalities for the detection of elements and people in real time. Thus, using the mechanism of the Haar Cascade Classifier, it is possible to identify objects in images or videos, regardless of their location or size. In addition, this algorithm is very fast, which makes it possible to detect elements in real time. Thus, with a pre-trained Haar Cascade model and using the functionalities provided by OpenCV, it is possible to detect the driver's movements in real time.

The gaze tracker attempts to continuously detect the face and eyes of the driver (see Fig. 2), and records as loss of attention the moments in which detection is not possible. This can occur, for example, when the driver is distracted by checking an application on an external screen and takes his eyes off the windshield.

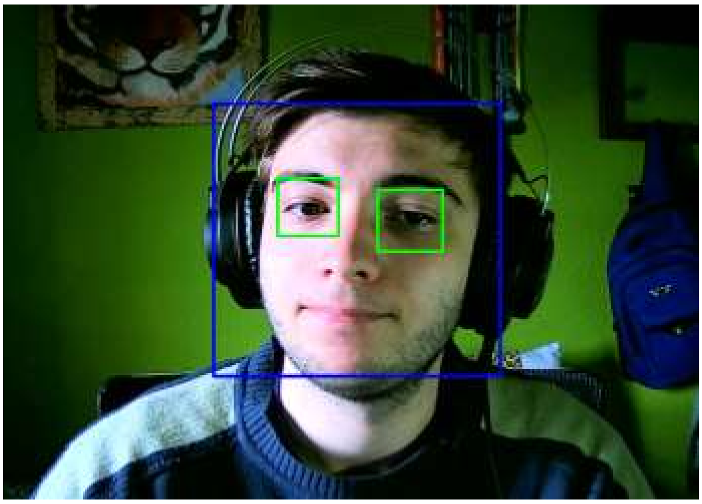

Fig. 2. Gaze recognition in real time

One of the goals of this work is to achieve a rapid and inexpensive prototyping and testing environment, so that it is available to as many developers as possible. It should be noted that it is not necessary to have a high-quality camera for proper operation. It is even possible, by means of mobile applications, to establish a connection between the camera of a cell phone and the computer, thus effectively turning it into a webcam.

4 Evaluation

The goal of the evaluation was to assess whether the proposed test suite helps developers understand and interpret driver behavior in different automotive environments while using an IVIS application. To do so, we used the testing suite to assess the impact of operating two different mobile applications while driving. This was done under the hypothesis that more complex interfaces require a higher level of attention and engagement from the driver, thus using the test suite should help us reach such conclusions.

The experiment was run with two different applications, with different levels of complexity: Facebook and Google maps. The Facebook mobile app was chosen under the hypothesis that it would result in a high cognitive load and source of distractions (as it was not designed to be used while driving). In contrast, Google maps was expected to be less demanding and suitable for in-car use. The working hypothesis is that using Facebook would yield considerably higher distraction time and more DRT misses.

4.1 Evaluation setting

Five subjects with an age range of 21 to 56 years (median=24) participated in the experiments. All subjects had a valid driver's license, and none had visual or hearing impairments.

Subjects seated in an ergonomic office seat in front of three adjacent 23-inch screens, which served as car windshields. They controlled the simulation using a steering wheel and pedals.

A smartphone was placed at the height of the steering wheel so as not to obstruct the driver's view of the windshield, in a position that is common to many drivers. The application under test was running on the phone.

The DRT light was displayed (turned on) for 2 seconds, in a random position, at random intervals (between 1 and 3 seconds). The driver had to press the "R2" button on the steering wheel as soon as he saw the light. In case of not pressing the button in time, the light automatically disappeared, and a miss was recorded. In case the "R2" button was pressed when the light was off, an error was recorded.

4.2 Tasks

Participants were asked to make a short tour of the city for 5 minutes, trying to respect all traffic rules (stop at red lights, avoid colliding with other cars, always maintain control of the car and stay in the corresponding lane). In addition, they were instructed on the tasks to be performed with a mobile application while driving through the city. In addition, while driving and using the app, they should respond to the DRT stimulus.

Participants were given a set of tasks to complete with each app. In the case of the use of Facebook, participants should read and like 8 publications, and access a person's profile. In the use of Google Maps, the trip to a destination must be configured, an intermediate stop must be added after starting the trip and finally an alternative route must be selected.

Each participant performed 3 simulation rounds of 5 minutes each with each of the applications (6 simulations in total). To avoid learning bias, the order in which the different applications are presented was randomized.

All tests were conducted on the same day with a half-hour separation between users. Before starting the tests, each driver was given 2 minutes to get used to operating the controls in the simulation. The series of 6 rounds was conducted with a 5-minute separation between each round, totaling a test load of 55 minutes (30 driving, 25 rest) per user session. Instructions were repeated prior to each start of running an application.

At the end of each session, a survey was conducted based on the Driver Activity Load Index (DALI), which is a method for measuring subjective user workload. The DALI questionnaire consists of a 6-item Likert scale (0 to 5, low to high) on: global attentional demand, visual, auditory, tactile demand, stress level, temporal pressure, degree of secondary task interference over driving.

4.3 Results and discussion

Fig. 3 summarizes the observed results. The orange bar (upper bar) corresponds to Google maps, whereas the blue bar (lower bat) corresponds to Facebook. The row labeled "DRT miss" indicates the number of DRT stimuli that were not attended in time and disappeared. The row labeled "DRT Error" reports the number of times the "R2" button was pressed and there was no DRT stimulus (this is counted to prevent the driver from constantly pressing the DRT button while the notification is not on the screen). The row labeled "Distraction time" indicates the average time (in seconds) that the driver kept his/her eyes off the simulation. For the calculation of this variable, the average of each driver was obtained and then the results of all participants were averaged.

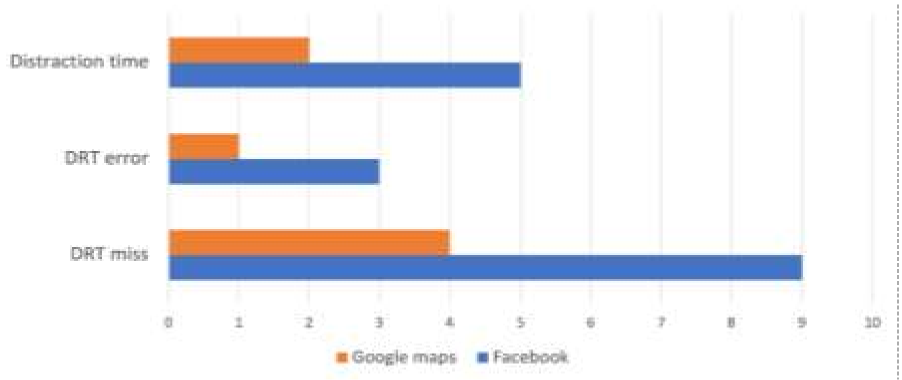

Fig. 3. Average observations for DRT errors, DRT misses, and total distraction time

As expected, distraction time was twice as much for Facebook than for Google maps. Similarly, DRT errors and misses were significantly more while using Facebook than while using Google maps.

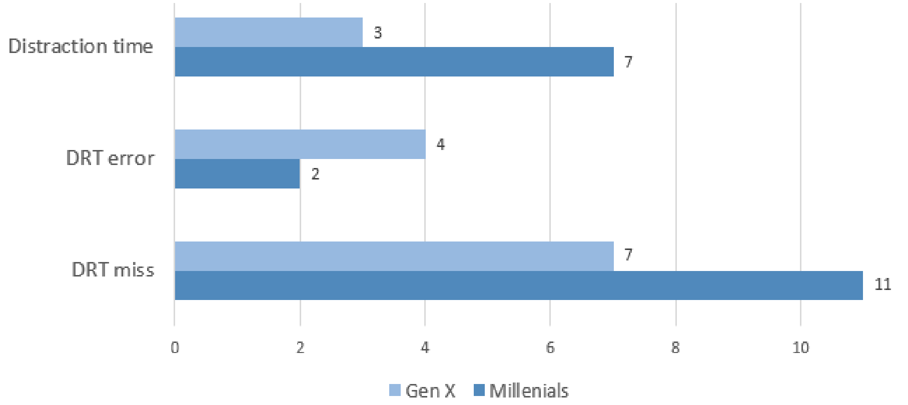

Fig. 4. Average DRT errors, DRT misses, and TDT by age group, while using Facebook

We wondered whether the suite could help us find any differences between millennial participants (that grew up using mobile apps) and Gen X participants (who adopted mobile apps during adulthood). At the risk of a reduced validity of the results (given the size of the sample), we partitioned results in two age groups: millennials (10 to 41 years old, to include both Gen Z and millennials) and Gen X (42 to 57 years old). We present two graphs, one for Facebook (Fig. 4) and one for Google maps (Fig. 5). In both graphs the lighter bar (upper bar) represents generation X, and the darker bar (lower bar) represents millennials (including Gen Z).

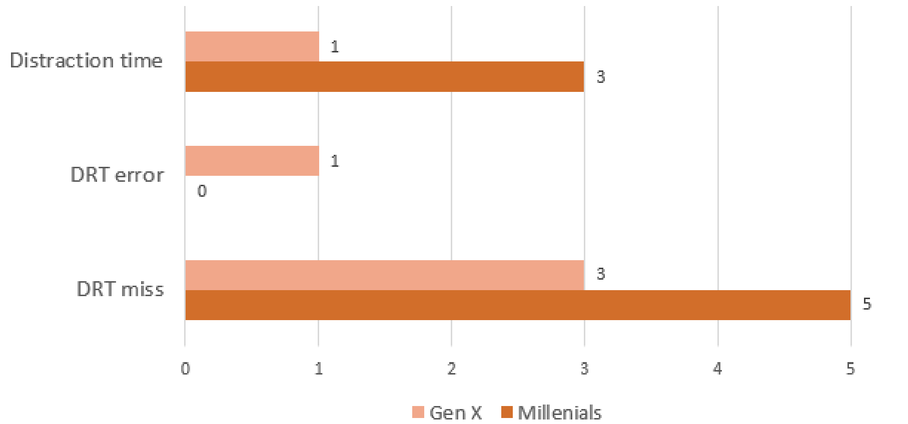

Fig. 5. Average DRT errors, DRT misses, and TDT by age group, while using Google Maps

It is notable that millennials have longer total distraction time and more missed DRTs but make fewer errors for both applications. Perhaps one explanation is related to the persuasive power of applications in adolescents. This preliminary result may offer a hint of an interesting question worth of further analysis with a larger user base.

All participants reported high levels of DALI for Facebook, particularly in terms of degree of interference on driving. In addition, subjects in Gen X commented on the difficulty of reading the application and loading interface elements.

These results indicate that the data collected with the three core elements of the testing suite (the simulation environment, the DRT instrument, and the Gaze Tracker) offers an indication of the impact of the IVIS app on the driver's attention. Moreover, as expected, the observed results match the expected results in the case of the two applications under study. However, comprehensive studies (specially including more subjects) are required to better assess the usefulness, and precision of the test suite.

5 Conclusions and future work

As development and distribution of IVIS become possible for any software developer, the need for affordable methods and tools to test how these systems impact the driver's attention becomes more important. Testing tools based on simulated environments have shown to be effective and low cost. We have built a low-cost testing suite that combines a simulated driving environment, a DRT instrument, and a Gaze Tracker. Preliminary experiments with the tool and with two applications whose impact on attention can be easily assessed, show that the data obtained with the test suite matches what is expected. The evaluation reported in this article is based on a small user sample and aimed only to assess the direction and magnitude of the results. Future work will improve evaluation with a larger user base and comparing against a more adequate and more detailed baseline. Moreover, future evaluation will also cover the perspective of the experimenter as well. This additional evaluation perspective will tell us whether the suite is usable and useful for application developers.

References

1. Antin, Jonathan F., Suzie Lee, Miguel A. Perez, Thomas A. Dingus, Jonathan M. Hankey, and Ann Brach. "Second Strategic Highway Research Program Naturalistic Driving Study Methods." Safety Science 119 (November 1, 2019): 2–10. https://doi.org/10.1016/j.ssci.2019.01.016.
2. National Academies of Sciences, Engineering, and Medicine. 2014. Naturalistic Driving Study: Field Data Collection. Washington, DC: The National Academies Press. https://doi.org/10.17226/22367.
3. National Academies of Sciences, Engineering, and Medicine. 2014. Analysis of Naturalistic Driving Study Data: Safer Glances, Driver Inattention, and Crash Risk. Washington, DC: The National Academies Press. https://doi.org/10.17226/22297.
4. S. Mecheri and R. Lobjois. Steering control in a low-cost driving simulator: a case for the role of virtual vehicle cab. Human factors, vol. 60, no. 5, pp. 719–734, 2018. DOI: 10.1177/0018720818769253
5. D. de Waard, M. van der Hulst, M. Hoedemaeker, and K. A. Brookhuis, "Driver behavior in an emergency situation in the automated highway system," Transportation human factors, vol. 1, no. 1, pp. 67–82, 1999. https://doi.org/10.1207/sthf0101_7

6. E. I. Farber, “Comments on driver behavior in an emergency situation in the automated highway system”, Transportation Human Factors, vol. 1, no. 1, pp. 83– 85, 1999. https://doi.org/10.1207/sthf0101_8
7. F. Bella, “Driving simulator for speed research on two-lane rural roads,” Accident Analysis & Prevention, vol. 40, no. 3, pp. 1078–1087, 2008. https://doi.org/10.1016/j.aap.2007.10.015
8. S. T. Godley, T. J. Triggs, and B. N. Fildes, “Driving simulator validation for speed research,” Accident analysis & prevention, vol. 34, no. 5, pp. 589–600, 2002. https://doi.org/10.1016/S0001-4575(01)00056-2
9. S. L. Jamson and A. H. Jamson, “The validity of a low-cost simulator for the assessment of the effects of in-vehicle information systems”, Safety Science, vol. 48, no. 10, pp. 1477–1483, 2010. https://doi.org/10.1016/j.ssci.2010.07.008
10. I. Alvarez, L. Rumbel, and R. Adams, “Skyline: a rapid prototyping driving simulator for user experience”, in Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, pp. 101–108, 2015. https://doi.org/10.1145/2799250.2799290
11. A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in Conference on robot learning, pp. 1–16, PMLR, 2017.
12. I. O. for Standardization, “Road vehicles — transport information and control systems — detection-response task (DRT) for assessing attentional effects of cognitive load in driving,” Tech. Rep. ISO 17488:2016, International Organization for Standardization, 2016.