AutoWSD: Virtual Reality Automated Driving Simulator for Rapid HCI Prototyping

Andreas Riegler
University of Applied Sciences Upper Austria
Hagenberg, Austria
andreas.riegler@fh-hagenberg.at

Andreas Riener
Technische Hochschule Ingolstadt
Ingolstadt, Germany
andreas.riener@thi.de

Clemens Holzmann
University of Applied Sciences Upper Austria
Hagenberg, Austria
clemens.holzmann@fh-hagenberg.at

Figure 1: View from the user's perspective towards the outside environment. On the windshield display, a warning message for take-over request is displayed.

ABSTRACT

Human factors research in automated driving is nowadays often conducted using either low-quality setups such as 2D monitors or highly expensive driving simulators with motion platforms. Additionally, software for automated driving scenarios is often expensive and hard to modify for different scenarios. We intend to bridge this gap by proposing a low-cost, high-fidelity immersive prototyping solution by utilizing the recent advances in development of virtual reality (VR): AutoWSD - Automated driving simulator for research on windshield displays. We showcase a hybrid software and hardware solution that is based on the popular and widely used Unity development platform. Furthermore, we demonstrate the little effort needed to create scenarios for user studies, and thereby foster discussion about potential improvements and extensions for AutoWSD, as well as the topic of trust, acceptance, user experience and simulator sickness in automation.

MuC '19, September 8–11, 2019, Hamburg, Germany

ACM ISBN 978-1-4503-7198-8/19/09.
https://doi.org/10.1145/3340764.3345366

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Interaction techniques**; *User studies*; *Scenario-based design*; *Interface design prototyping*.

KEYWORDS

virtual reality, windshield display, heads-up display, user studies, prototyping

ACM Reference Format:
Andreas Riegler, Andreas Riener, and Clemens Holzmann. 2019. AutoWSD: Virtual Reality Automated Driving Simulator for Rapid HCI Prototyping. In *Mensch und Computer 2019 (MuC ’19), September 8–11, 2019, Hamburg, Germany*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3340764.3345366

1 INTRODUCTION

Evaluating human-centered concepts and theories is often carried out with the help of user studies. In 2018’s AutoUI proceedings, 30 of 35 published full papers utilized them, whereas 27 (90%) of those implemented functional prototypes (i.e. no low-fidelity paper prototyping or similar). However, from our experience conducting user studies in automated driving research in the past, realizing them is often a time-consuming and repetitive task, in many cases having to start from scratch when designing a new scenario for a user study, especially when involving multiple devices and types of interaction and communication. Therefore, we designed and implemented a software framework in combination with external hardware for more immersive user studies. This setup allows study organizers to have full control over the use case scenarios which is often limited when using proprietary software. The presented system is already in use, and with our demo, we want to encourage discussion of possible improvements, changes and extensions for this implementation.

2 SYSTEM DESCRIPTION

AutoWSD is a software system with interfaces for hardware devices (e.g., steering wheel, pedals, android devices, microphone, webcam) which aims to simplify and speed up the design, implementation and evaluation of user studies in the context of automated driving. For this purpose, we introduce a low-cost high-fidelity driving simulator in virtual reality, created in the Unity development platform [13]. Although our aim of this simulator is the use in virtual reality using a VR headset (see fig. 2), it can also be used in a setup consisting of one or multiple monitors.

Figure 2 shows a typical setting we use when conducting user studies. The monitor is used by the study organizer to see the progress of the participant.

Figure 1 gives an overview of the main components of the presented VR driving simulator framework, as well as their purpose. We designed the framework as a modular system with loose coupling between the components. Therefore, it can easily be modified and extended with new components.

Event Manager

For the purpose of passing data (e.g., gesture input type, recognized speech commands, gaze direction, start/stop events from the study organizer) between components, we implemented an event manager. Study designers can create any events and optionally pass any kind of data to other components listening to these events.

Figure 2: Example study setting.

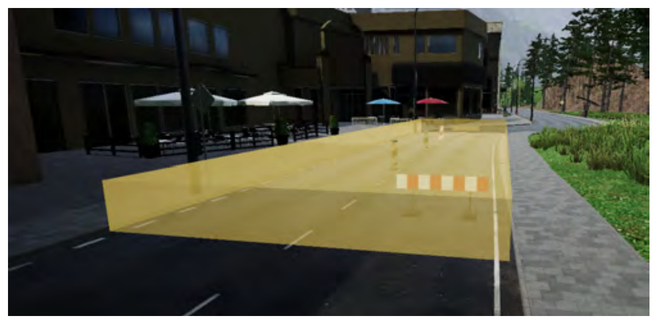

Figure 3: Take over request trigger, displayed in yellow, for triggering a manual take over because of an unexpected road situation.

Figure 3 shows a use case of the event manager. Whenever the user’s vehicle "collides" with the take over request (TOR) trigger while in automated driving mode, the vehicle switches to manual driving mode and the driver must take over the steering wheel and pedals to maneuver the car away from danger. Upon leaving the TOR area, the vehicle switches back to automatic driving mode. For demonstration puposes, we visualized the TOR trigger as a yellow cube. In user studies, the TOR area is invisible to the user (driver).

Required Hardware

Apart from a VR-ready PC running Windows 10, which runs the Unity application, additional hardware is required in order to fully benefit from the system’s capabilities (see Table 2 for purpose of the hardware and examples).

Component	Modules	Purpose
Vehicle	-	Defines automated and manual driving, 3D model of the vehicles and driver, engine sounds, side view mirrors, animated steering wheel (in automated driving mode).
Interaction	Gaze Gesture Speech Steering	Defines what kinds of interactions can be performed by the user and their triggered actions. Examples: Recognition of a specified keyword, gazing at a button in the WSD user interface, performing a swipe-left gesture.
WSD	-	Defines the user interface of the windshield display, such as the menu structure, single or multiple application windows, content types, their positions ...
Data Management	Input Output Messaging Sound Timing UserStudy	Defines what kind of data is processed an how the data is processed. The Messaging module contains the *EventManager* which is responsible for passing data between components, based on triggers, timers or other events defined by the study organizer. Examples: text data displayed on the WSD, or which sound effects are placed upon a button click, or study data such as scenario and participant identifiers.
Traffic	WaypointCircuit WaypointRoute Crossing* Traffic Circle*	Vehicles (user's vehicle with the study participant, and traffic vehicles) driving around in the scene. Vehicles follow a predefined path using waypoints.
Pedestrian	WaypointCircuit WaypointRoute Crosswalk*	Pedestrians walking on the sidewalk and across streets, interacting with each other and the vehicles.

Table 1: Overview of the framework's components and modules. Entries denoted by * are not fully implemented yet.

Hardware	Purpose	Example
Virtual Reality Headset	Experiencing the immersive VR environment	HTC Vive Pro
Virtual Reality Headset Tracker(s)	Tracking the head movements of the user for correctly updating the VR environment	HTC Base Station
Hand/Finger Motion Sensor	Representation of the user's hands; Gestural interaction	Leap Motion
Steering Wheel	Steering the vehicle in manual mode	Logitech G25 Racing Wheel
Pedals	Accelerating/breaking the vehicle in manual mode	Logitech G25 Pedals
Microphone	Speech input	Samson Meteor Mic
Speakers	Audio output	HTC Vive Pro (integrated speakers)

Table 2: Required hardware.

3 VIEW MANAGEMENT

The view management of the AutoWSD VR driving simulator is based on windshield displays (WSDs). These displays utilize the entire windshield to display content on a larger scale than head-up displays (HUDs). This results in many advantages such as physical and visual clutter in the center console [9], a single interface for all in-vehicle infotainment systems [4], and the visualization of potential dangers directly in the driver's field of view [6]. Figure 1 displays an example of such a WSD application, that displays a take-over request (red box). Additionally, the WSD shows text projected on the front vehicle using augmented reality, thereby focussing the driver's attention to that vehicle.

4 INTERACTION MANAGEMENT

When it comes to managing interactions between the driver and the vehicle, we use a multimodal approach for automated driving scenarios (cf., [11]). For this purpose, we implemented a variety of interaction types that can be used either separately or combined. In the following, we present and discuss our used interaction modalities.

Gestural Interaction

In-vehicle usage of gesture interaction has many applications, such as controlling infotainment systems (e.g., climate controls, music players) [1] and maneuver-based interventions in automated vehicles [14]. Since the driver can observe the outside environment (e.g., road, pedestrians) instead of glancing at in-vehicle controls, gestures provide the benefits of increased road safety [5] and natural interaction [9]. So far, we implemented multiple gestures: *Swiping* left/right, up/down, forward/backward, *Palm* facing up/down, *Finger postures* (thumb up/down, index finger forward) and *Fist*. In the context of a view management concept, a potential use of the fist gesture would be to show the main menu, while a swipe down gesture could trigger a minimize-all-windows operation. Swiping left and right could trigger previous and next applications, similar to the application switcher on tablets.

Gaze-based Interaction

The use of gaze as input modality has been researched (e.g., [8], [2]). The human attention focus is closely related to the viewing direction [3]. Therefore, we also consider gaze an integral part of interaction design for windshield displays. We conducted a user study to determine the preferred dwell time and visual feedback type for (purely) gaze-based interactions. While we did not integrate an eye tracking device, we used the head rotation (and therefore VR headset rotation) as gaze. With the release of the HTC Vive Pro Eye, with integrated Tobii eye trackers, we intend to further validate the use of eye tracking in VR environments. The combination of gaze-based and gestural interactions has been found to show great promise [10].

Speech Interaction

In recent years, voice user interfaces have gained more and more popularity. Smart home devices such as Amazon Echo, Google Home or Apple Homepod require the user to interact with their voice, with little or no visual feedback. Auditory user interfaces in the context of automated driving provide tremendous benefits, for example for warning or infotainment purposes (e.g., [12]). We utilize the Microsoft Speech API (SAPI) to process spoken (key)words by the user and transform them into text, which we further analyze based on the given scenarios. Recognized words or phrases are passed via the *EventManager* to the registered objects. For example, spoken keywords *Yes* or *No* in the context of verifying settings can be used instead of eye gaze or gestural inputs. Since we access the offline version of SAPI, a constant internet connection is not required.

5 DEMO SETUP

We will host a live and interactive demonstration of the AutoWSD system. The setup features examples and scenarios which were already used in user studies. Additionally, we make all source code and project files visible at the booth. Visitors experience the automated driving simulator in an immersive virtual reality environment. To this end, visitors are encouraged to implement their own scenarios. We will give instructions on how to design scenarios in the Unity Editor, such as creating streets, waypoints for the vehicles to follow, windshield display user interfaces, take-over request triggers etc.

6 CONTRIBUTION

In the future, we plan to make the presented system available as open-source framework. While nowadays high-fidelity driving simulators are in use already (e.g., Lungaro et al. [7] demonstrated a high-fidelity driving simulator based on GTA V), we believe that the increasing immersion and expansion of VR will help the scientific and industrial UX community in rapidly prototyping and testing automotive applications and scenarios. By providing an interactive demo setup, we intend to foster discussion and gather feedback from the research community about further improvements and extensions. Additionally, participants can try out our interactive prototype with various scenarios already used in scientific user studies. These scenarios will provide the basis for further discussions, such as the use of windshield displays, interaction modalities, tackling user experience issues, simulator sickness, and trust in vehicle automation.

ACKNOWLEDGMENTS

This work was supported by the University of Applied Sciences PhD program of the government of Upper Austria, the program was executed by the FFG.

REFERENCES

[1] Kenneth Majlund Bach, Mads Gregers Jæger, Mikael B. Skov, and Nils Gram Thomassen. 2008. You Can Touch, but You Can'T Look: Interacting with In-vehicle Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1139–1148. https://doi.org/10.1145/1357054.1357233

[2] Stanislavs Bardins, Tony Poitschke, and Stefan Kohlbecher. 2008. Gaze-based interaction in various environments. In *Proceedings of the 1st ACM workshop on Vision networks for behavior analysis*. ACM, 47–54.

[3] Ishan Chatterjee, Robert Xiao, and Chris Harrison. 2015. Gaze+Gesture: Expressive, Precise and Targeted Free-Space Interactions. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICMI '15)*. ACM, New York, NY, USA, 131–138. https://doi.org/10.1145/2818346.2820752

[4] Joseph L Gabbard, Gregory M Fitch, and Hyungil Kim. 2014. Behind the Glass: Driver challenges and opportunities for AR automotive applications. *Proc. IEEE* 102, 2 (2014), 124–136.

[5] Renate Häuslschmid, Benjamin Menrad, and Andreas Butz. 2015. Freehand vs. micro gestures in the car: Driving performance and user experience. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 159–160.

[6] Hao Li and Fawzi Nashashibi. 2011. Multi-vehicle cooperative perception and augmented reality for driver assistance: A possibility to 'see'through front vehicle. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 242–247.

[7] Pietro Lungaro, Konrad Tollmar, Firdose Saeik, Conrado Mateu Gisbert, and Gaël Dubus. 2018. Demonstration of a Low-cost Hyper-realistic Testbed for Designing Future Onboard Experiences. In *Adjunct Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '18)*. ACM, New York, NY, USA, 235–238. https://doi.org/10.1145/3239092.3267850

[8] T. Poitschke, F. Laquai, S. Stamboliev, and G. Rigoll. 2011. Gaze-based interaction on multiple displays in an automotive environment. In *2011 IEEE International Conference on Systems, Man, and Cybernetics*. 543–548. https://doi.org/10.1109/ICSMC.2011.6083740

[9] Andreas Riener. 2012. Gestural interaction in vehicular applications. *Computer* 45, 4 (2012), 42–47.

[10] Florian Roider and Tom Gross. 2018. I See Your Point: Integrating Gaze to Enhance Pointing Gesture Accuracy While Driving. In *Proceedings of the 10th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '18)*. ACM, New York, NY, USA, 351–358. https://doi.org/10.1145/3239060.3239084

[11] Florian Roider, Sonja Rümelin, Bastian Pfleging, and Tom Gross. 2017. The Effects of Situational Demands on Gaze, Speech and Gesture Input in the Vehicle. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '17)*. ACM, New York, NY, USA, 94–102. https://doi.org/10.1145/3122986.3122999

[12] Ivan Tashev, Mike Seltzer, Yun-Cheng Ju, Ye-Yi Wang, and Alex Acero. 2009. Commute UX: Voice enabled in-car infotainment system. (2009).

[13] Unity Technologies. 2019. Unity Mixed Reality Solutions. https://unity.com/solutions/xr. Accessed: 2019-05-22.

[14] Hermann Winner and Stephan Hakuli. 2006. Conduct-by-wire–following a new paradigm for driving into the future. In *Proceedings of FISITA world automotive congress*, Vol. 22. Citeseer, 27.