Design and Evaluation of a Free-Hand VR-based Authoring Environment for Automated Vehicle Testing

Sevinc Eroglu* Frederic Stefan† Alain Chevalier† Daniel Roettger† Daniel Zielasko ‡
Torsten W. Kuhlen* Benjamin Weyers ‡

*Visual Computing Institute, RWTH Aachen University
†Ford Motor Company, Aachen, Germany
‡Human-Computer Interaction, University of Trier

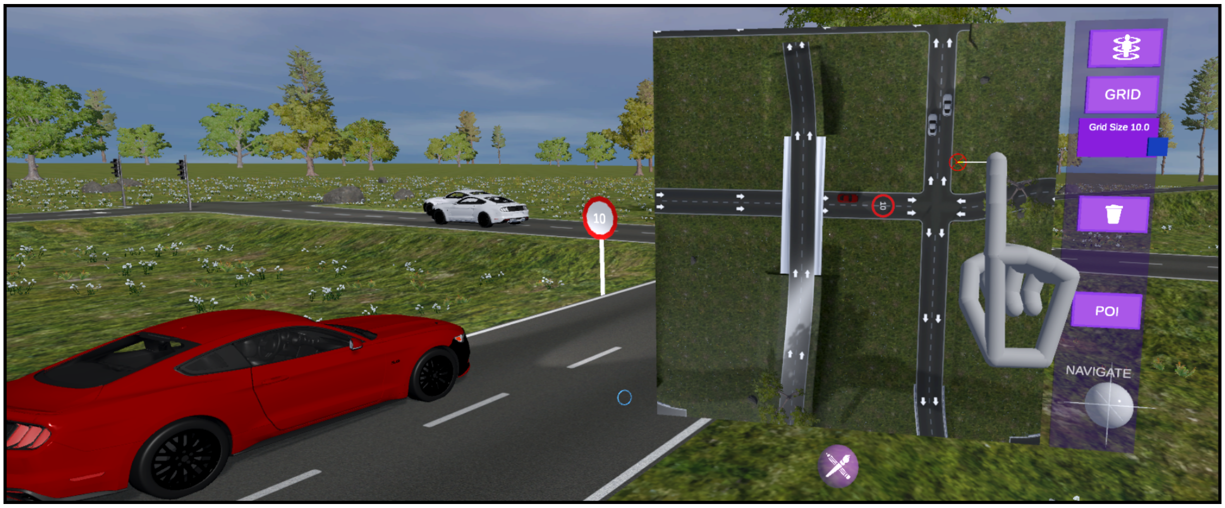

Figure 1: The user authors a virtual traffic scenario via 3D interactions on a 2D panel by using free-hand gestural inputs.

ABSTRACT

Virtual Reality is increasingly used for safe evaluation and validation of autonomous vehicles by automotive engineers. However, the design and creation of virtual testing environments is a cumbersome process. Engineers are bound to utilize desktop-based authoring tools, and a high level of expertise is necessary. By performing scene authoring entirely inside VR, faster design iterations become possible. To this end, we propose a VR authoring environment that enables engineers to design road networks and traffic scenarios for automated vehicle testing based on free-hand interaction. We present a 3D interaction technique for the efficient placement and selection of virtual objects that is employed on a 2D panel. We conducted a comparative user study in which our interaction technique outperformed existing approaches regarding precision and task completion time. Furthermore, we demonstrate the effectiveness of the system by a qualitative user study with domain experts.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Virtual Reality; Human-centered computing—Interaction design and evaluation methods—User interface design—User studies

1 INTRODUCTION

Real-life testing and validation of automated and autonomous vehicles is a challenging task regarding safety concerns and efficiency. Engineers have to consider unexpected traffic, road, and weather conditions when preparing testing scenarios. Designing and executing these scenarios may involve potential risks to the system itself, or the environment. Additional to these risks, reproducibility of the scenarios is limited, and slow development and testing cycles become expensive for automotive companies [65].

To address these limitations, original equipment manufacturers employ simulators of virtual traffic scenarios. This enables much faster development and testing cycles at reduced cost and without safety concerns. Recently, Nvidia introduced the commercial Nvidia Drive Sim and Nvidia Drive Constellation platforms to simulate testing of autonomous vehicles [13]. They collaborate with rFpro who provides photo-realistic rendering of traffic scenes. Furthermore, Siemens recently announced their in-house autonomous driving performance validation and verification solution [58].

Virtual Reality (VR) offers a number of potential benefits for examination of the simulated scenarios and to perform system validation and verification [54]. Immersive visualization enables improved spatial perception, e.g. for the realistic judgement of driving speeds and distances [43]. Furthermore, efficient navigation in the 3D environment is possible and the scenario can easily be observed from varying perspectives. For instance, users can follow traffic movement from inside a vehicle and seamlessly switch to a pedestrian's perspective standing at a crossing.

An important building block of such virtual traffic simulations is the design of specific testing scenarios, which includes the creation of road networks and behavioral properties such as speed limits and

---

*e-mail: {eroglu, kuhlen}@vr.rwth-aachen.de
†e-mail: {fstefan, achevali, droettge}@ford.com
‡e-mail: {zielasko, weyers}@uni-trier.de

traffic directions. These scenarios can be generated from real public data sources such as geographic information systems (GIS) [45], or by procedural generation [35]. In addition, user-controlled interactive design of road networks and traffic scenarios can be performed to test vehicles in custom environments [8]. User-defined scenarios can be adapted to specific requirements and provide increased flexibility in their design.

However, to create such scenarios, engineers are limited to 2D desktop-based domain-specific tools for the design of road networks or common 3D modeling environments, which require adequate knowledge of these expert tools. Even with expertise on such tools, it is inefficient to perform a full design iteration including an evaluation of the scenario in VR. If an intermediate modification of the road geometry is necessary while being immersed, the user has to leave the VR environment and perform the modifications with a desktop-based tool. The model then has to be re-imported and the user has to re-enter the validation environment. This breaks the immersion and, more importantly, slows down the design and evaluation cycle, which costs time and effort.

To this end, we propose a VR-based authoring environment that enables an iterative workflow of road network authoring and automated driving simulation inside a single immersive virtual environment (IVE). Users can create and modify road networks at run-time via free-hand gestural interactions. Free-hand gestures are employed to support natural and intuitive interactions. Design iterations with direct feedback become possible, which enables a human-in-the-loop rapid evaluation and comparison of traffic scenarios.

Road network creation is an inherently 2D interaction task. To define a road network, start and end points of each road segment need to be defined. The height of the terrain, however, is irrelevant for the resulting network topology. Therefore, we designed a control widget that enables this 2D interaction task in a 3D environment by orthographic rendering of the terrain. It is used for road network manipulation as well as an efficient means to navigate the scene. The 2D scene control panel is embedded into the 3D environment and can be placed freely relative to the user. We devised a novel 3D interaction technique for indirect placement and selection of objects on the 2D terrain surface. It is thus possible to perform manipulations of the road network, while immediately observing the result in the 3D surrounding.

To show the effectiveness of our approach, we evaluated possible design alternatives and performed a comparative user study. Existing work on free-hand road network creation in VR proposes a direct-manipulation interface that transforms the scene via two-handed gestural interactions [26]. Objects are placed directly onto a scaled-down version of the terrain, which is then scaled up again to perceive the surrounding in real-world size. In contrast to their approach, we enable users to do simultaneous manipulation and perception of the virtual environment via indirect interaction techniques by utilizing a 2D scene control panel.

Besides quantitative data on precision and task completion time, we furthermore gathered qualitative feedback on our overall system design from a heterogeneous group of participants that included VR experts and control engineers. The insights we gathered from the qualitative evaluation will serve as a basis for further improvement of the system.

In summary, the contribution of this work is two-fold:

- First, we present a system for authoring road networks in VR that can serve for automated vehicle testing. We evaluate the effectiveness in a qualitative user study with VR experts and control engineers.
- Second, we propose an indirect interaction technique for object placement and selection on a 2D terrain surface. We evaluate its performance in a comparative user study towards a direct free-hand interaction technique.

2 RELATED WORK

2.1 Scene Authoring in Virtual Reality

Immersive visualization in VR enables an improved spatial perception. Regarding interactive scene authoring, this provides the user with a more realistic sense of scales and distances in the virtual environment, and a better judgment of velocities for dynamic scenes. Commercial game engines recognize this potential and provide frameworks such as the *VR Editor* by Unreal Engine [12] and *EditorXR* by Unity Engine [3] to author 3D environments while being immersed.

To author an IVE, scene objects need to be created and arranged by the user. Object creation inside VR can be achieved by different means, such as geometric modeling [17, 31, 39, 41, 51, 53], sketching [4,11,30] or procedural generation [59]. In the following, we focus on existing approaches for high-level scene authoring, i.e. placement and arrangement of objects, rather than geometric modeling.

Mine [52] proposes *ISAAC* that enables users to construct interactive virtual scenes directly in an IVE by using direct and indirect manipulation techniques. MultiGen's *SmartScene*, evolved from *PolyShop* by Mapes et al. [49], employs 2-handed interaction techniques realized via tracked data gloves to author the scene in VR. Wang et al. introduce the hybrid immersive level-editing system *DIY World Builder* that enables users to create the environment via a hybrid wand and tablet interface [67]. They provide functionalities for terrain editing, placing and texturing objects, and controlling lights in the virtual environment. Barot et. al present the *Wonderland Builder* that enables users to create and manipulate objects in VR via multimodal interaction techniques, coupling voice, and tracked hand input commands [16]. *Genesys* is a VR scene builder that provides assets from a built-in content browser in VR [28]. Users can load and manipulate these assets via free-hand gestural inputs. Another approach is presented by Ichikawa et al. [38] in *VR Safari Park*, which enables users to design a virtual environment simulation using blocks and a hierarchical world tree. The blocks represent objects such as animals and trees, and by attaching these blocks to the world tree, objects are added into the virtual environment. While these works present various approaches to the general problem of scene authoring in VR, none of the previous works specifically address road network creation.

2.2 Road Network Creation

Creation of virtual road networks has been addressed by different methods. These methods can be based on public data sources [45,66,70], procedural generation [1,24,29,34,36,62], deep learning [32,37,42] or user-defined creation. A user-driven approach is inevitable when the design has to be based on specific application requirements. While manual creation can be a cumbersome process, it yields flexibility in the road network creation. To this regard, engineers employ desktop-based tools such as *Road Generator* [9], *CarMaker* [5], *CarSim* [6] and *SUMO* [10]. However, modification of the created road networks in VR is not possible with these tools.

A recent work that is presented by Côté et al. [26] addresses this issue by enabling users to create and modify roads in VR. Like our system, users interact with the virtual environment using free-hand gestures. It enables users to create roads by placing control points directly on the terrain. The shape of the road can be modified by dragging these control points. Navigation is realized by scaling the virtual world by moving the closed fists apart or closer. Inspired by their work, we implemented direct interaction and travel techniques to compare with our indirect interaction techniques for creating roads and navigating the virtual environment (see Section 4).

2.3 2D Selection in 3D Space

Object selection is a major task for scene authoring in VR. While there is comprehensive literature on 3D selection techniques [14,46,50,69], previous work also studied 2D interactions, such as the design and evaluation of 2D widgets or menus in VR [21, 27, 71].

Regarding object selection, several techniques employ interactions with the projected 2D image plane in an IVE. The *Aperature* technique presented by Forsberg et al. [33] uses a tracked six degrees of freedom (DOF) input device, while the *Image-plane* technique by Pierce et al. uses free-hand interactions for object selection [55]. Ware and Lowther presented *One-Eyed Cursor*, in which they evaluated the effectiveness of a 2D cursor compared to a stereo cursor for 3D selection tasks [68]. Their study was later extended by Teather et al. [63] accounted for input devices with varying DOF. *EZCursorVR* is another image-plane selection technique, proposed by Ramcharitar et al. [56], uses a 2D head-coupled cursor fixed in screen space that can be employed with 2, 3 or 6 DOF input devices.

While our technique is similar to these approaches, it differs in the rendering of the projected image on the 2D plane and the way interactions are performed. Compared to other techniques, we render an orthographic projection of the scene on an upright 2D plane that can be positioned freely relative to the user. Furthermore, we utilize a ray-based selection that is perpendicular to the 2D plane. The ray originates from the tip of the finger and does not depend on the pointing direction. Therefore, stretching out the finger is not necessary, as in the *Sticky Finger* technique by Pierce et al. [55], which might cause fatigue after extended usage [40].

3 SYSTEM DESCRIPTION

To enable users to author road networks and driving scenarios, we developed an interactive VR-based authoring environment. The system will be used by novice users that might lack considerable experience with 3D interfaces. We, therefore, designed a natural user interface based on free-hand gestures to make the application as approachable and intuitive as possible. See Section 4.2 for the technical setup of the system.

For the interactive creation of road networks, various operations need to be implemented. Specific requirements were iteratively developed in close collaboration with control engineering experts of Ford Motor Company. First, the system needs to enable users to create, delete, and cut road segments, add crossings and bridges, and modify the properties of these elements, such as the curvature or slope of a road segment. The second requirement is to define parameters for the later simulation. Thus, our system enables users to change the traffic direction and add static objects, vehicles, and speed limits while creating the virtual environment as well as during the testing phase. The third requirement is that users need to efficiently navigate through the environment during scene authoring. Therefore, users can navigate via flying, teleportation, or by using a minimap technique. Further, our system enables saving and loading of the created scenes, to continue the design at a later time or export the scenario for offline editing. Moreover, to assist users to employ the system, video tutorials for each feature are provided that can be watched inside the virtual environment. In the following, our system is explained in detail.

3.1 System Menu and Control Interface

To enable novice users to interact with our system, a natural user interface based on free-hand interactions is implemented. Gestures executed by the user serve as the system input. Although gestures can be defined by the whole body of the user, we focus on using hand, finger, and palm. We find that a sitting position is the right setup for interacting with our system since authoring road networks in VR can be tiresome if the user has to stand for hours [18,72]. In the following, we describe the interaction with the UI elements and the scene control panel.

3.1.1 UI Elements

Because we aim to address a large number of features, adapted 2D menus are implemented, as suggested by Laviola et al. [46]. One of the concerns about floating menus in VR is the occlusion with the environment. However, we addressed this issue by enabling the placement of floating menus freely relative to the user by dragging a sphere handle. Upon grabbing the handle, the attached floating menu is minimized to provide an unobstructed view of the scene, and the semi-transparent frame will be visible to guide the arrangement of the menus in the environment (see Figure 2). Furthermore, the floating menus can be anchored to the user's left hand.

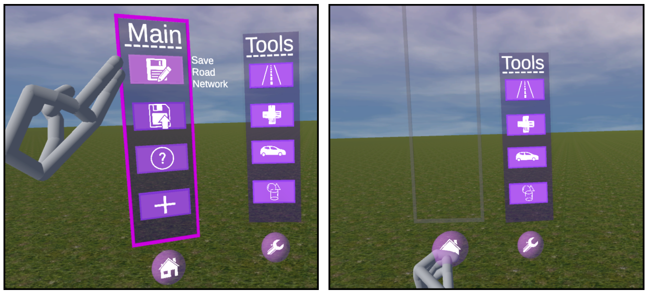

Figure 2: Adapted 2D menus. Left: Hovering over a menu button. Right: Grabbing the menu handle.

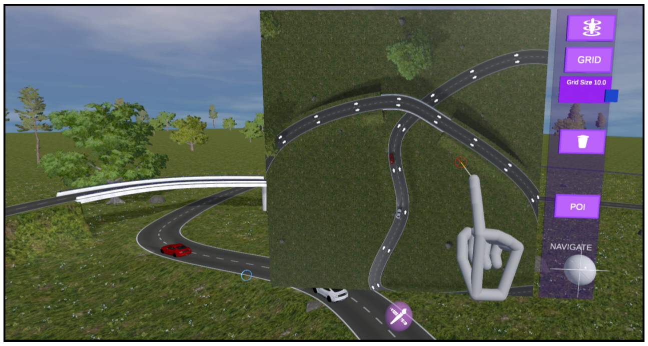

Figure 3: Interaction with the 2D scene control panel.

Upon rotating the left palm towards the eyes, two menu items become visible. These items can be detached by pulling them away from their anchor. To enable users to interact in a fast manner, these handles snap to their anchor location when they are in a defined range. This is visualized by interpolating the color of the handle from blue to magenta. Furthermore, multi-modal cues, visual and auditory, are provided to substitute for the lack of haptic feedback during interaction with the menu buttons [20].

3.1.2 Scene Control Panel

The planning of real-world road networks by civil engineers typically starts from a 2D blueprint. To match this common workflow, we designed a 2D interface that enables users to create road networks in VR (see Figure 3). The *scene control panel* consists of a minimap, a navigation panel, and a tools menu that can be placed freely in the virtual environment after detaching from its anchor. Via the navigation panel, the user can activate teleportation, zoom in/out, and panning on the minimap. The minimap, a 2D World in Miniature [61], shows an orthographic projection of the scene, and users can add, select, and delete objects by interacting with it. Furthermore, altering the properties of objects is possible via the property menus that can be enabled by a left pinch gesture. The property menu spawns in front of the minimap and is drawn semi-transparently to enable users to observe the changes during interaction.

3.2 Travel

To address our requirement of efficiently navigating the scene, we implemented three travel techniques that are in the category of steering-based and selection-based techniques [46].

Figure 4: Left: The ray, which is segmented into 3 regions (red, yellow and white represent add, select and idle modes respectively), is drawn perpendicular to the 2D panel. Right: Upon selection of an object.

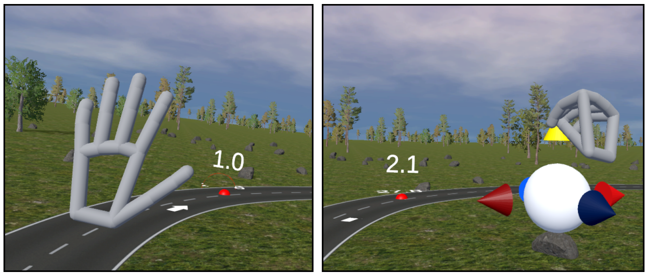

Figure 5: The flashlight pointing selection (Right) and manipulation of control points via the widget (Left) in the system.

Flying [57] is realized via a multi-modal interaction technique to improve the user experience [46]. It is enabled and disabled by giving "fly" and "stop" voice commands, respectively. After enabling the technique, the position of the user is continuously translated into the pointing direction of the stretched right hand. Furthermore, flying speed can be manipulated by changing the distance between the knuckle of the index finger and the thumb of the right hand.

Teleportation [22] is realized by instantly changing the location of the user to a position that is selected on the minimap. Furthermore, the view of the minimap can be changed via the *navigation widget* that is located on the bottom right of the scene control panel (see Figure 3). By dragging the sphere handle from its resting position, zooming in/out and sideways panning can be performed.

3.3 Object Selection and Manipulation

To address our first requirement, a user interface for the selection and manipulation of objects in VR has to be provided. Therefore, we designed an interaction technique for selection and manipulation of objects via the minimap on the scene control panel. It provides an efficient tool for 2D selection and placement of objects on the terrain. The interaction is based on free-hand gestural input and illustrated in Figure 4. For a better spatial perception of the manipulated objects, we furthermore enable users to perform interactions in 3D space.

The selection and placement of objects is performed based on motion of the right index finger towards the minimap. The specific action that is triggered depends on the distance of the finger tip to the 2D panel. A line that represents the pointing ray is drawn perpendicular to the panel, and indicates the active range for the different actions to the user. It is drawn as 3 colored segments and has a cursor, drawn as a red cross (see Figure 4 Left). The colored segments represent the different actions that are triggered when the tip of the index finger meets with the respective segment. White, yellow, and red-colored regions on the line represent the *idle*, *select*, and *add* modes, respectively.

An object is added to the virtual environment when the tip of the index finger reaches the red segment of the line and touches the cursor. After an add operation, the indicator line becomes invisible, and the user needs to pull back his index finger to the distance at which the idle mode is defined to perform further interactions. This *push-pull gesture* is used to avoid accidental add operations.

To select an object, the user moves the index finger into the yellow segment of the indicator line (see Figure 4 Right). Selected objects are highlighted in a red color. Different actions can then be performed on the selected objects via the control panel, such as deleting, splitting or merging of road segments.

Apart from the indirect interactions via the scene control panel, all objects in the virtual environment can furthermore be transformed via 3D selection and manipulation. For 3D object selection, we use the flashlight pointing technique, which uses a conic selection volume to select small and distant objects [47]. In our system, the apex of the cone is located at the palm of the left hand, such that objects can be selected by pointing the left palm towards an object.

To perform accurate manipulation of an object's position, we employ a transformation widget instead of a direct mapping to the user's hand motion (see Figure 5). By dragging the widget via the axis arrows, the position of the selected object can be manipulated.

3.4 Road Network Authoring

Our system enables users to author road networks and traffic scenarios via placement and manipulation of objects in the virtual environment. This addresses our primary requirement for interactive road network creation. Objects can be roads, crossings, bridges, speed limit signs, static scene objects and vehicles.

Regarding road geometry, the default shape of a road segment is a Catmull-Rom spline, which is defined by adding control points onto the terrain. The user can add a continuous sequence of control points to create connected road segments. By adding the last control point close to the starting point of the road, closed-loop road tracks can be created. The creation of a new, separate road segment can be activated via the tools menu. To enable users to create road segments with zero curvature, an underlying grid is implemented onto which the added control points will snap. The size of the grid can be altered and its visibility can be toggled via the scene control panel.

Our system enables users to interact with the road network control points by means of translating, deleting, adding onto the road segment, or changing the type of control point via the properties menu. Altering the control type has an effect on the shape of the road segment which is generated between the selected markers and the next one. Besides splines, arc segments with constant curvature, as well as straight-line segments can be generated.

To define a road network, the creation of different types of crossings is needed. In our system, the user can add crossings and bridges onto the terrain or onto a road segment. When added onto a road segment, the road will divide into two separate segments that are connected to this crossing. If the added crossing object is close to the beginning or end of an existing road segment, it will snap to the control point and get connected automatically. Road segments that originate from the crossings can be generated. To achieve this, the user first selects the crossing and then adds the road control points. The default crossing shape is an "X" crossing to which 4 separate roads can be connected. It can be switched to a "T" crossing via the crossing properties menu.

For further editing of the created road, the system enables users to cut the road into two separate segments via the minimap. Moreover, separated tracks can be connected by first selecting one of the tracks and then adding a control point close to another road segment. Possible connections are indicated by a floating "Connect" text and by highlighting the target control point. Thereby, the user will have an understanding of his actions. Connections can be created between two separate road tracks, two crossings, or between a crossing or bridge and a road track.

The user furthermore can add speed limit signs at a desired world

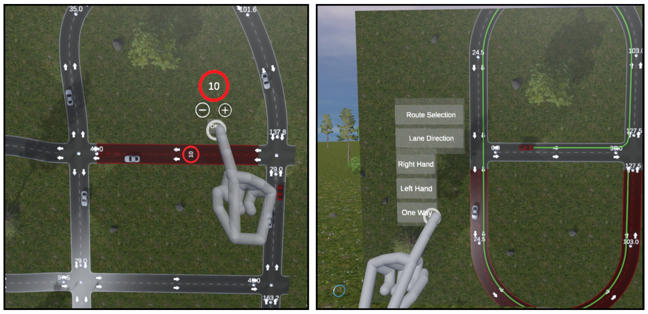

Figure 6: Left: The speed limit user interface. Right: Traffic direction setting via the property menu.

position via the minimap. Upon adding a speed sign, a small widget will spawn to set the limit (see Figure 6). Via this widget, the user can increase or decrease the limit in steps of ten. After confirming the limit, a speed sign with the chosen limit becomes visible on both sides of the road. Undesired speed signs can be selected and deleted via the control panel.

Our system enables users to change the traffic direction of a generated road segment while authoring the environment, or even during the testing phase, which is not possible with existing simulation tools [2, 6]. The traffic direction is visualized as an arrow on each lane of the road and can be altered between right-hand, left-hand, and one-way traffic. Currently, the user can create roads with two lanes, and the direction of these can be changed via the road properties menu (see Figure 6).

In the simulation, the system needs to identify dead ends and grant a right or left turn at intersections. Therefore, the underlying road network topology is represented as a directed multi-graph [19]. Crossings are defined as vertices and each lane of the road is expressed as a directed edge. By using this underlying multi-graph representation, an automated vehicle can perform route planning during the simulation.

Vehicles can be placed onto a lane via the minimap. The orientation of the vehicle is adapted based on the traffic direction. Furthermore, the user can add static scene objects such as rocks onto the scene. This enables testing scenarios in which the vehicle encounters an unexpected object on the road while the simulation is running.

4 USER STUDY

Since our goal is to provide a tool that can be used by domain experts in professional environment, we seek to provide an interface for road network authoring that is fast and efficient. Furthermore, the precise placement of objects and road network control points is required for the creation of testing environments for automated driving. At the same time, the system needs to be approachable by novice users and provide a good user experience. Therefore, we consider *speed*, *accuracy*, and *usability* as the three major relevant aspects. To this end, we compared our indirect interaction technique to an existing approach based on direct free-hand interaction. In addition, we are interested in general feedback on the usability of our system. Therefore, we conducted an empirical evaluation in three parts.

In the first two parts, we focused on the comparison of the two techniques, where the direct method is highly inspired by the VR tool described by Côté et. al [26]. This approach enables users to create roads by adding control points into the virtual environment by directly touching the terrain. The comparative study aims to investigate whether our indirect approach improves the user performance.

Therefore, we investigate the following hypotheses:

H1: Regarding the task completion time, indirect interaction will be faster than direct interaction.

H2: Regarding the precision, indirect target selection will be more accurate than direct selection.

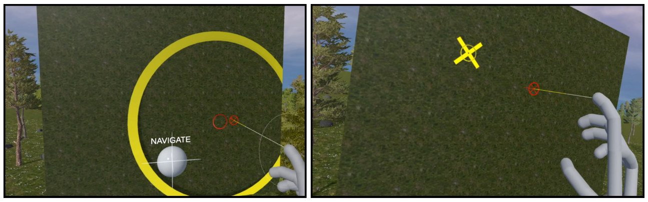

Figure 7: Task completion time (Left) and precision tasks (Right) for the **Indirect** condition.

H3: Users will have a higher preference to use the 2D panel for road network creation in VR.

H3.1: A top-down view alongside the 3D scene rendering improves the spatial understanding of the scenario.

Considering previous findings in the original work and based on our preliminary discussion, we assume that the direct method is more suitable for adding objects to the environment. Thus, we further hypothesize:

H4: Direct interaction will be more preferable by the users to add objects into the VR environment.

In the third part of the study, we gathered qualitative feedback on the overall functionality of our system. This part serves as a guideline for further improvements of the interface concepts, refinement of the usability, and the implementation of additional features.

4.1 Study Design and Tasks

To test **H1** to **H4**, we created a one-factorial within-subject design where our two conditions (**Indirect** and **Direct**) were implemented. For both conditions, two scenarios were designed. In the first scenario, we measure task completion, selection, and navigation time, while the second scenario measures precision only.

The experimental conditions were carefully designed to be as comparable as possible. The order of the **Indirect** and **Direct** conditions was counterbalanced across the participants, while the specific target sequence was kept identical. Regarding navigation, we applied the same scaling factors to match the effective distances the user's hands need to travel. Thereby, the measured effects, which are described below, are due to the interaction technique, and the influence of other variables is minimized.

Regarding task completion time (see **H1**), the task is to add a sequence of control points onto predefined target locations, while being as fast as possible. To achieve this, both selection and navigation have to be performed. To make the user employ the travel technique for zooming in on a target, we disabled target placement as long as the view is zoomed out too far. Participants first have to navigate to the target location that is indicated with 2 circles with different color and size (see Figure 7). While the size of the outer circle is animated in a repeated manner, the inner circle has a static size. The participants could only add a control point to the target location as soon as the inner circle became visible based on the zoom or scale level. Task completion time is counted after the first control point has been added to the target location.

Regarding the precision tasks (see **H2**), participants were instructed to be as precise as possible while adding the sequence of control points without time restriction. We disabled the zoom in/out features so that the target sizes and distances are identical for both conditions, and the precision values are not affected by any other factors. The target location is rendered with an animated circle and an "X" icon. The participants were instructed to add a control point onto the center of the "X" icon.

In the condition **Indirect**, the participants employed the minimap on the 2D scene control panel to add control points, and the navigation widget to navigate on the minimap (see Section 3.2). In the training phase, we instructed the participants to position the navigation widget

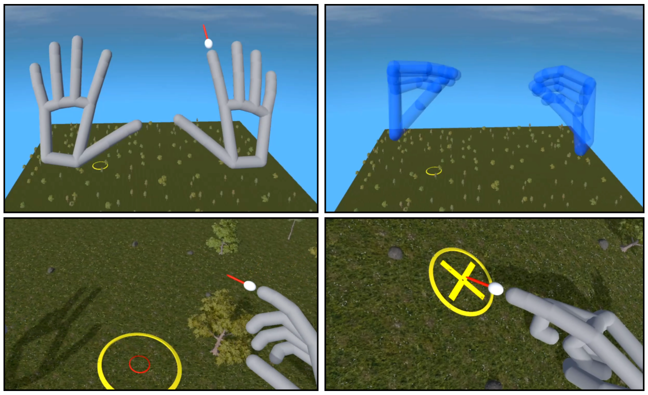

Figure 8: **Direct** interaction hands for selection (Top-Left) and navigation (Top-Right).Task completion time (Bottom-Left) and precision tasks (Bottom-Right) for the **Direct** condition.

wherever it feels comfortable by performing a pinch gesture at the desired location. This location then stays fixed throughout the task.

In the condition **Direct**, adding objects is achieved by touching the indicator, attached to the right index finger tip, onto the terrain (see Figure 8). Navigation is performed by using two-handed gestural input. It is enabled when the user performs a grab gesture with both hands. This is indicated by changing the color of the hands to blue. Panning in the virtual environment is realized by dragging the virtual world into the desired direction. Moving the grabbed hands apart executes zoom-in and moving towards each other zooms out.

Regarding **H3** and **H4**, a subjective questionnaire was designed to be ranked by the participants after each of the **Indirect** and **Direct** conditions was completed.

Finally, for the qualitative study, we designed a set of guided tasks in which the participants had to create and manipulate a road network, along with performing dynamic changes in the scene while the simulation is running. To validate our hypothesis **H3.1**, another subjective questionnaire (see Table 1) was prepared to be ranked after the qualitative study.

4.2 Apparatus

The study took place in our lab and was conducted in a seated position. The HTC Vive and HTC Vive Pro HMDs were employed, and both of them tracked with two tripod-mounted Lighthouse 1.0 base stations. To track the user's hand, we employed the Leap Motion Controller (LMC) using the 4.0.0 version of the Leap Motion SDK. The LMC was mounted onto the VR headset. The experimental platform was developed in Unity 2018.2.14f1, running Windows 10 on a 3.50GHz Intel Xeon E5-1650 with NVIDIA GeForce GTX 1080 GPU.

4.3 Procedure

We conducted our study during the Covid-19 pandemic and prepared the study with health considerations by following the guideline presented by Steed et al. [60]. Since varying tracking stations and frame rates would cause inconsistency in our measured data, we used a uniform tracking station and a desktop PC. Furthermore, since the disinfection of headsets using wipes is not a safe solution, each participant used a headset that is not shared by anyone else on the same day. Therefore, we waited 72 hours, as suggested by the current study [64], to use the respective headset again for the next participant besides cleaning it with disinfectant wipes.

The rest of the study procedure is described in detail in the following. After the participants signed a consent form, they had to answer pre-study questionnaires regarding demographics, prior experience with 3DUIs, video games, VR, and free-hand interaction in AR/VR, as well as a simulator sickness questionnaire (SSQ) [44].

Table 1: Subjective questionnaire for the overall system.

Q1: "I would use this system to create a road network in VR."
Q2: "The 2D scene control panel with free-hand interaction allows me to create a road network in VR fast and efficiently."
Q3: "The 2D scene control panel with free-hand interaction allows me to create a road network in VR with high precision."
Q4: "The 2D scene control panel with free-hand interaction allows me to create a road network in VR with low effort."
Q5: "I find panning and zooming the 2D scene control panel using the navigation widget efficient."
Q6: "I find panning and zooming the 2D scene control panel using the navigation widget intuitive."
Q7: "Having a top-down view alongside the 3D scene improves my understanding of the scenario."
Q8: "Dynamic changes to the traffic scenario (tempo limits, lane directions) while the simulation is running enables me to evaluate different scenarios more efficiently."
Q9: "I find changing the slope of the road using the manipulation widget convenient."

At the beginning of both **Indirect** and **Direct** conditions, instruction videos were shown that explain the steps and how to interact with the environment. Afterwards, a training phase was performed. This phase was unrestricted in time and trials and lasted until the participant felt comfortable with the respective technique. Then, participants executed 4 runs of task completion time and 4 runs of precision tasks. For each run, 5 target locations are defined for the completion time task, and 4 for the precision task, respectively.

To avoid fatigue, each condition was designed to last under 10 minutes and took about 8 minutes on average. After each condition, the participants took off the HMD and answered a system usability score (SUS) questionnaire [23] and a 7-point Likert scale questionnaire to give subjective feedback on the used technique. Afterwards, the participants performed a set of guided tasks to create a road network and evaluate the functionality of our system. These tasks consisted of adding and connecting crossings, bridges, and road segments, adding a vehicle onto a road, running the simulation, and adding speed limits while the simulation is running.

After the guided tasks, the participants could try out the functionality of the system freely. In the guided and free exploration phase, the participants were instructed to think-aloud and comment freely on the system so that the observer could transcribe the comments. Lastly, the participants were instructed to answer a 7-point Likert scale questionnaire (see Table 1) related to the overall system and a second SSQ. Overall, the study took about 60 min.

4.4 Participants

16 subjects (1 female and 15 male, mean age = 31.3 years old, SD=6.45) voluntarily participated in the study. Among the participants, 4 were professional control engineers, 8 were VR experts and 4 were students. All participants were right-handed and reported normal or corrected-to-normal vision. Regarding their prior experience with 3D user interfaces as well as VR, 10 participants reported regular usage and 6 used it at least once before. For the video gaming experience, this distribution was 12 to 4. Furthermore, concerning the experience on the free-handed interaction in VR/AR, 3 participants reported regular usage, 9 used it at least once and 4 had no experience at all. The SSQ reported an average score of 11.2 (SD=14.5) after the experiment and 1.1 (SD=10.3) as a difference score between before and after the experiment.

5 RESULTS

We analyzed the scaled measures using a paired samples t-test since our study has a repeated-measures design. To ensure the effectiveness of the presented result of the t-test, we also reported the effect size

	Task Completion Time [s]	Navigation Time [s]	Selection Time [s]	Precision [m]	SUS
Indirect	30.49 (±7.43)	26.37 (±6.01)	4.12 (±1.87)	0.53 (±0.31)	80.31 (±11.72)
Direct	47.90 (±19.18)	42.99 (±18.53)	4.91 (±1.96)	0.93 (±0.45)	57.96 (±27.78)

Table 2: Mean and standard deviation of task completion time, navigation time, selection time, precision and the SUS score are presented for **Indirect** and **Direct** conditions.

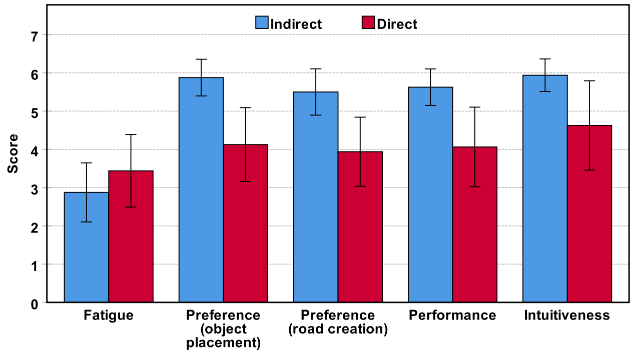

Figure 9: Subjective feedback that is ranked for **Indirect** and **Direct** conditions. Error bars represent the standard error of the mean.

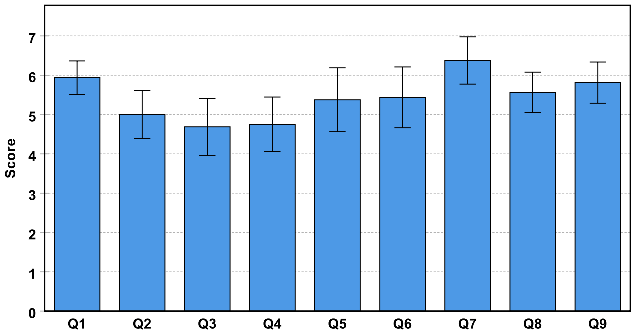

Figure 10: Subjective feedback that is ranked for the system. Error bars represent the standard error of the mean.

by calculating Cohen's d [25]. Regarding the effect size d, 0.2, 0.5 and 0.8 are considered small, medium and large respectively. The subjective questionnaires for both conditions (**Indirect** and **Direct**) were analyzed using a Wilcoxon Signed-Rank test. Due to a wrong assumption for evaluating error selections, our results do not include error rates on the selection tasks. For all tests, we assume a significance level of .05.

5.1 Time and Precision

For the condition **Indirect** and **Direct**, we recorded task completion time, navigation time, selection time and precision per participants. The results are shown in Table 2. A paired t-test shows that **Indirect** significantly outperformed **Direct** regarding Task Completion Time ($T(15) = 4.519$, $d = 1.13$, $p < .001$), Navigation Time ($T(15) = 4.330, d = 1.08, p < .001$), and Precision ($T(15) = 4.065, d = 1.01, p = .001$). However, regarding Selection Time, results show that there is no significant difference between the techniques ($T(15) = 1.109, d = 0.27, p = .285$).

5.2 Subjective Feedback

For the **Indirect** and **Direct** conditions, the participants were asked to answer a SUS and a subjective questionnaire with 7-point Likert scale items. The SUS score for each condition is reported in Table 2. The 7-point Likert scale items are ranged from very strongly disagree (1) to very strongly agree (7). The results are shown in Figure 9. A Wilcoxon Signed-Rank test was conducted and shows a significant effect of the technique on Preference (object placement) ($Z = -2.523$, $p = .012$), Preference (road creation) ($Z = -2.760, p = .006$), Performance ($Z = -2.452, p = .014$), and Intuitiveness ($Z = -1.965, p = .049$). However, no significant effect was found on Fatigue ($Z = -0.988, p = .323$).

Concerning the guided task-based condition and free-exploration phase in the main application, the participants were asked to score another subjective questionnaire with 7-point Likert scale items (see Table 1). The results can be seen in Figure 10.

5.3 Qualitative Results

Overall, most of the participants liked the system. One of the participants stated, "It is nice and fun" and one control engineer stated, "I think I can spend hours here, it is really interesting". Furthermore, three control engineers stated that configuration of the virtual environment is very fast and easy compared to *CarMaker* [5], a commercial software that is commonly used by automotive manufacturers. However, they also stated that *CarMaker* provides considerably more overall functionality compared to our system. Furthermore, being able to save the scene in XML format, which can be loaded into the system again, was appreciated by the control engineers, since offline editing is possible.

Regarding the interaction with the 2D scene control panel, many participants appreciated to be able to immediately see the creation in the virtual environment. One of them stated, "Being able to manipulate in 2D and seeing it directly in the 3D environment gives me good feedback, and you have a better understanding of the 3D environment". Furthermore, some of the participants found the interaction with the 2D panel "intuitive".

Concerning the 2D scene control panel, four participants liked that the panel spawns upright and facing towards the user. Furthermore, one control engineer appreciated that he can position the menu items and the 2D panel freely in the environment. However, being able to tilt the panel was requested by two participants.

Regarding menu button interactions, six participants stated that toggle functionality should be employed for objects that are expected to be added multiple times into the virtual environment.

Connecting crossings functionality was grasped easily and was intuitively performed by most participants. Although our system provides textual feedback about possible connections, three VR experts reported that the snapping range was not evident and proposed a visual indicator. However, after enabling the visibility of the underlying grid, they reported that this gives more adequate feedback and helps to comprehend the functionality easily. Furthermore, one VR expert suggested visualizing this zone with a radius indicator.

While the translation of objects via the manipulation widget was liked by the participants, six of them reported that grabbing and moving the manipulation ball instead of dragging the axis arrows was expected to interact with the widget.

Besides the comments given for the existing features, participants also suggested additional features concerning the overall system. One control engineer requested a feature to enter digits for the curvature of the selected road since it would be important to be precise on the curvature values for testing vehicles. Another control engineer stated that it would be nice to see the current velocity of the vehicle and the coordinate information of the objects in the scene. One participant asked for a feature to create tunnels and sidewalks.

Furthermore, another participant stated that it would be nice to have an interactive walkthrough to guide novice users in the application.

6 DISCUSSION

As stated in our first hypothesis **H1**, we expected that the participants would complete the tasks faster by employing the indirect interaction techniques, compared to the direct techniques. Based on the results, we found a significant difference in the task completion time that shows **Indirect** outperformed **Direct** in terms of speed, thus we can accept **H1**. To complete the tasks, participants had to utilize selection as well as travel techniques. Therefore, we recorded the time that they spent on selection and navigation individually. The results show no significant difference between the selection techniques, while the travel technique in the **Indirect** condition significantly outperforms the technique in the **Direct** condition. This result indicates that the overall task completion time is affected by the navigation rather than the selection technique.

One possible reason why the navigation was significantly less efficient in the **Direct** condition is the larger travel distance of the two hands compared to the single-handed **Indirect** navigation. Another reason could be the varying degree of experience with 3D spatial transformations of our participants. During the study, we observed that some of the participants had difficulties to perform the tasks in the **Direct** condition while others grasped the technique quickly.

To investigate this further, we analyzed the relation between the expertise (usage of free-hand interaction techniques in VR/AR) and the task completion time in the **Direct** condition with a Pearson test. The result shows a strong correlation between the expertise and task completion time ($r(16) = -.606, p = .013$), and the participants who employ free-hand interaction techniques frequently completed the tasks faster than the others. As stated earlier, this could be due to a better spatial awareness inside the virtual environment, which enables these participants to grasp a newly introduced interaction and employ it with ease. Regarding the **Indirect** condition, we did not find a significant correlation ($r(16) = -.395, p = .130$). An explanation might be the widespread experience with 2D interfaces that are commonly used in everyday life. This indicates that our technique can be employed comfortably by users with a varied range of expertise on free-hand interaction techniques in VR/AR.

Orthographic projection is commonly used by engineers to create accurate drawings of models. Thus, as stated by our hypothesis **H2**, we expected that the participants would be more precise on the selection tasks using our indirect interaction on the 2D panel, compared to the direct selection. The result of our study supports this hypothesis by showing that **Indirect** significantly outperformed **Direct** in the precision tasks. One possible reason for this could be the posture of the arm. For the direct selection, the arm has to be extended further away from the body, while the indirect selection allows to rest the elbow comfortably close to the body. Another reason could be related to the DOFs of the interaction techniques. The indirect selection uses fewer degrees of freedom compared to the direct technique, for which previous work has shown higher performance in a Fitts' law task [15, 48, 63].

Since road network design is a 2D interaction task by nature, our hypothesis **H3** stated that the participants would prefer to employ the 2D panel over direct interactions. The results show a significant difference between the techniques, and **Indirect** is preferred for road creation which confirms our hypothesis. Regarding **H3.1**, we were able to support our hypothesis based on the score of the subjective question (Q7), which is ranked quite high by the participants (see Section 5.2). During the guided task phase, one participant stated that he had a better mental model of the scene while employing the 2D panel view alongside the 3D rendering of the scene.

Regarding **H4**, we expected that the participants would prefer to employ direct interactions to add objects into the virtual environment. However, our results show that the participants preferred the **Indirect** technique rather than **Direct** (see Section 5.2). Thus, we can not confirm **H4**. The reason behind this might be the navigation aspect of the direct interaction. Some users found traveling the scene via zooming in and out with both hands to be less ergonomic than traveling via the minimap. One participant stated that applying the technique "feels like doing a workout". In the next iteration, we plan to do further evaluation by omitting the travel techniques and evaluate this hypothesis based on the selection techniques only.

Based on the comparative and qualitative results, our indirect free-hand selection and manipulation technique has proven to be well-suited for precise and fast placement of objects onto a terrain surface. We believe that our interaction technique performed particularly well due to its design that addresses the inherently 2D nature of the road network problem. Thus, our system provides an adequate tool for the free-hand authoring of road networks and traffic scenarios that was preferred by the participants over existing scene authoring techniques.

7 LIMITATIONS & FUTURE WORK

Our system is currently limited to create roads with two lanes. In future work, we plan to enable users to create more complex road structures, such as roads with varying number of lanes, widths, and the possibility to merge roads with a different number of lanes.

Concerning traffic scenario creation, we will support saving and loading of the *OpenSCENARIO* [7] file format, which is commonly used by automotive companies to describe driving and traffic scenarios. To this end, we are investigating interaction techniques that enable users to define the behavior of vehicles based on a rule-based system that describes their relation to pedestrians and other vehicles.

Regarding our study, one limitation could be using different headsets. However, we tried to minimize the potential confound by using the same tracking system (see Section 4.2). Another possible limitation is that the results might be not generalizable to a larger population since most participants were male. These limitations should be considered in future iterations.

For further improvements on the system, the numerous valuable feedback we gathered from our participants will be addressed in future work. Furthermore, regarding our selection technique, future studies will explore the performance of our ray-based selection technique when combined with two-handed interactions and other input modalities in a Fitts' law task.

8 CONCLUSION

We presented a system for VR-based authoring of virtual environments for the testing and validation of automated vehicles. Our system enables users to create road networks and traffic scenarios, and modify them at run-time based on free-hand gestures. It provides an efficient way to perform design, testing, and validation cycles since all steps are executed inside the application without a break of immersion.

Further, we devised a novel interaction technique that enables the user to add and select objects on the terrain surface by using a 2D control panel that can be placed freely in the 3D environment. 3D objects in the scene get projected onto the 2D panel, and with our interaction technique, the user can employ multiple actions, e.g. selection and addition of objects, based on the distance of the index finger to the control panel.

To evaluate the performance of our interaction technique, we conducted a user study in which we compared against an existing free-hand interaction method. Furthermore, we gathered feedback for our system by means of a qualitative user study with domain experts. The results show that our indirect technique outperforms direct interaction methods in terms of speed and precision, as well as usability and intuitiveness.

ACKNOWLEDGMENTS

This work was partly supported by the Ministry of Culture and Science of the State of North Rhine-Westphalia under the funding line "Digital Tools for University Teaching DH-NRW".

REFERENCES

[1] Advanced 3D city design software — arcgis cityengine. https://www.esri.com/en-us/arcgis/products/arcgis-cityengine/overview?rmedium=www_esri_com_EtoF&rsource=/en-us/arcgis/products/esri-cityengine/overview. (Accessed on 11/03/2020).

[2] CarMaker — IPG automotive. https://ipg-automotive.com/products-services/simulation-software/carmaker/. (Accessed on 08/17/2018).

[3] Create immersive authoring tools in Unity EditorXR. https://unity.com/editorxr. (Accessed on 11/05/2020).

[4] Gravity Sketch — 3D design and modelling software. https://www.gravitysketch.com/. (Accessed on 11/10/2020).

[5] IPG automotive GmbH — everything about virtual test driving. https://ipg-automotive.com/. (Accessed on 10/28/2020).

[6] Mechanical simulation. https://www.carsim.com/. (Accessed on 10/28/2020).

[7] OpenSCENARIO - home. http://www.openscenario.org/. (Accessed on 06/15/2020).

[8] Press release — press by IPG Automotive GmbH. https://press.ipg-automotive.com/press-release/article/scenario-editor-efficient-generation-of-scenarios/. (Accessed on 11/10/2020).

[9] Road Generator - dSPACE. https://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models/produkte_asm/modeldesk/roadgenerator.cfm. (Accessed on 10/28/2020).

[10] SUMO-GUI - SUMO documentation. https://sumo.dlr.de/docs/sumo-gui.html. (Accessed on 10/28/2020).

[11] Tilt brush by google. https://www.tiltbrush.com/. (Accessed on 11/10/2020).

[12] Unreal Engine VR Editor — Unreal Engine Documentation. https://docs.unrealengine.com/en-US/Engine/Editor/VR/WelcomePDF/index.html. (Accessed on 11/05/2020).

[13] Virtual-based safety testing for self-driving cars — nvidia drive. https://www.nvidia.com/en-us/self-driving-cars/drive-constellation/. (Accessed on 10/30/2020).

[14] F. Argelaguet and C. Andujar. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, 2013.

[15] R. Arsenault and C. Ware. The importance of stereo and eye-coupled perspective for eye-hand coordination in fish tank VR. *Presence: Teleoperators & Virtual Environments*, 13(5):549–559, 2004.

[16] C. Barot, K. Carpentier, M. Collet, A. Cuella-Martin, V. Lanquepin, M. Muller, E. Pasquier, L. Picavet, A. Van Ceulen, and K. Wagrez. The wonderland builder: Using storytelling to guide dream-like interaction. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 201–202. IEEE, 2013.

[17] L. Beever, S. Pop, and N. W. John. LevelEd VR: A virtual reality level editor and workflow for virtual reality level design. In *2020 IEEE Conference on Games (CoG)*, pp. 136–143. IEEE, 2020.

[18] M. Bellgardt, S. Pick, D. Zielasko, T. Vierjahn, B. Weyers, and T. W. Kuhlen. Utilizing Immersive Virtual Reality in Everyday Work. In *2017 IEEE 3rd Workshop on Everyday Virtual Reality (WEVR)*, pp. 1–4. IEEE, 2017.

[19] B. Bollobás. *Modern graph theory*, vol. 184. Springer Science & Business Media, 2013.

[20] D. A. Bowman, E. Kruijff, J. J. LaViola Jr, and I. Poupyrev. An introduction to 3-D user interface design. *Presence: Teleoperators & Virtual Environments*, 10(1):96–108, 2001.

[21] D. A. Bowman and C. A. Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *Proceedings IEEE Virtual Reality 2001*, pp. 149–156. IEEE, 2001.

[22] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey. Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pp. 205–216. ACM, 2016.

[23] J. Brooke. SUS: A quick and dirty usability scale. *Usability evaluation in industry*, p. 189, 1996.

[24] G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang. Interactive procedural street modeling. In *ACM SIGGRAPH 2008 papers*, pp. 1–10. 2008.

[25] J. Cohn. Statistical power analysis for the behavioral sciences. *Hillsdale, NJ: Lawrence Earlbam Associates*, 1988.

[26] S. Côté and O. Beaulieu. VR Road and Construction Site Safety Conceptual Modeling Based on Hand Gestures. *Frontiers in Robotics and AI*, 6:15, 2019.

[27] K. Das and C. W. Borst. An evaluation of menu properties and pointing techniques in a projection-based VR environment. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 47–50. IEEE, 2010.

[28] J. D. O. De Leon, R. P. Tavas, R. A. Aranzanso, and R. O. Atienza. Genesys: A virtual reality scene builder. In *2016 IEEE Region 10 Conference (TENCON)*, pp. 3708–3711. IEEE, 2016.

[29] A. Emilien, U. Vimont, M.-P. Cani, P. Poulin, and B. Benes. Worldbrush: Interactive example-based synthesis of procedural virtual worlds. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015.

[30] S. Eroglu, S. Gebhardt, P. Schmitz, D. Rausch, and T. W. Kuhlen. Fluid Sketching—Immersive Sketching Based on Fluid Flow. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 475–482. IEEE, 2018.

[31] S. Eroglu, P. Schmitz, C. A. Martinez, J. Rusch, L. Kobbelt, and T. W. Kuhlen. Rilievo: Artistic Scene Authoring via Interactive Height Map Extrusion in VR. *Leonardo*, 53(4):438–441, 2020.

[32] Z. Fang, T. Yang, and Y. Jin. DeepStreet: A deep learning powered urban street network generation module. *arXiv preprint arXiv:2010.04365*, 2020.

[33] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pp. 95–96. ACM, 1996.

[34] E. Galin, A. Peytavie, E. Guérin, and B. Beneš. Authoring hierarchical road networks. In *Computer Graphics Forum*, vol. 30, pp. 2021–2030. Wiley Online Library, 2011.

[35] E. Galin, A. Peytavie, N. Maréchal, and E. Guérin. Procedural generation of roads. In *Computer Graphics Forum*, vol. 29, pp. 429–438. Wiley Online Library, 2010.

[36] A. Gambi, M. Mueller, and G. Fraser. Automatically testing self-driving cars with search-based procedural content generation. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 318–328, 2019.

[37] S. Hartmann, M. Weinmann, R. Wessel, and R. Klein. Streetgan: Towards road network synthesis with generative adversarial networks. 2017.

[38] S. Ichikawa, K. Takashima, A. Tang, and Y. Kitamura. VR safari park: a concept-based world building interface using blocks and world tree. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–5, 2018.

[39] B. Jackson and D. F. Keefe. Lift-off: Using reference imagery and freehand sketching to create 3D models in VR. *IEEE transactions on visualization and computer graphics*, 22(4):1442–1451, 2016.

[40] S. Jang, W. Stuerzlinger, S. Ambike, and K. Ramani. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 3328–3339, 2017.

[41] J. Jerald, P. Mlyniec, A. Yoganandan, A. Rubin, D. Paullus, and S. Solotko. MakeVR: A 3D world-building interface. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 197–198. IEEE, 2013.

[42] L. Z. Kelvin and B. Anand. Procedural generation of roads with conditional generative adversarial networks. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pp. 277–281. IEEE, 2020.

[43] A. Kemeny and F. Panerai. Evaluating perception in driving simulation experiments. *Trends in cognitive sciences*, 7(1):31–37, 2003.

[44] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[45] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO-Simulation of Urban MObility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

[46] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P.

Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[47] J. LIANG and M. GREEN. JDCAD: a highly interactive 3D modeling system. *Computers & graphics*, 18(4):499–506, 1994.

[48] S. Luo, R. J. Teather, and V. McArthur. Camera-based selection with cardboard head-mounted displays. In *International Conference on Human-Computer Interaction*, pp. 383–402. Springer, 2020.

[49] D. P. Mapes and J. M. Moshell. A two-handed interface for object manipulation in virtual environments. *Presence: Teleoperators & Virtual Environments*, 4(4):403–416, 1995.

[50] D. Mendes, F. M. Caputo, A. Giachetti, A. Ferreira, and J. Jorge. A survey on 3D virtual object manipulation: From the desktop to immersive virtual environments. In *Computer graphics forum*, vol. 38, pp. 21–45. Wiley Online Library, 2019.

[51] D. Mendes, D. Medeiros, M. Sousa, R. Ferreira, A. Raposo, A. Ferreira, and J. Jorge. Mid-air modeling with boolean operations in VR. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 154–157. IEEE, 2017.

[52] M. Mine. ISAAC: A virtual environment tool for the interactive construction of virtual worlds.

[53] M. Mine, A. Yoganandan, and D. Coffey. Making VR work: building a real-world immersive modeling application in the virtual world. In *Proceedings of the 2nd ACM symposium on Spatial user interaction*, pp. 80–89, 2014.

[54] T. S. Mujber, T. Szecsi, and M. S. Hashmi. Virtual reality applications in manufacturing process simulation. *Journal of materials processing technology*, 155:1834–1838, 2004.

[55] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3D immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pp. 39–ff. ACM, 1997.

[56] A. Ramcharitar and R. J. Teather. EZCursorVR: 2D selection with virtual reality head-mounted displays. In *Proceedings of the 44th Graphics Interface Conference*, pp. 123–130. Canadian Human-Computer Communications Society, 2018.

[57] W. Robinett and R. Holloway. Implementation of flying, scaling and grabbing in virtual worlds. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pp. 189–192, 1992.

[58] Siemens. Accelerate vehicle performance verification for autonomous driving systems. `https://www.plm.automation.siemens.com/global/en/webinar/vehicle-performance-engineering-autonomous-driving-simulation-testing/55387`. (Accessed on 11/03/2020).

[59] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes. A survey on procedural modelling for virtual worlds. In *Computer Graphics Forum*, vol. 33, pp. 31–50. Wiley Online Library, 2014.

[60] A. Steed, F. R. Ortega, A. S. Williams, E. Kruijff, W. Stuerzlinger, A. U. Batmaz, A. S. Won, E. S. Rosenberg, A. L. Simeone, and A. Hayes. Evaluating immersive experiences during covid-19 and beyond. *Interactions*, 27(4):62–67, 2020.

[61] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 265–272, 1995.

[62] J. Sun, X. Yu, G. Baciu, and M. Green. Template-based generation of road networks for virtual city modeling. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pp. 33–40, 2002.

[63] R. J. Teather and W. Stuerzlinger. Pointing at 3D target projections with one-eyed and stereo cursors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 159–168, 2013.

[64] N. Van Doremalen, T. Bushmaker, D. H. Morris, M. G. Holbrook, A. Gamble, B. N. Williamson, A. Tamin, J. L. Harcourt, N. J. Thornburg, S. I. Gerber, et al. Aerosol and surface stability of SARS-CoV-2 as compared with SARS-CoV-1. *New England Journal of Medicine*, 382(16):1564–1567, 2020.

[65] W. Wachenfeld and H. Winner. The release of autonomous vehicles. In *Autonomous driving*, pp. 425–449. Springer, 2016.

[66] J. Wang, G. Lawson, and Y. Shen. Automatic high-fidelity 3D road network modeling based on 2D GIS data. *Advances in Engineering Software*, 76:86–98, 2014.

[67] J. Wang, O. Leach, and R. W. Lindeman. DIY World Builder: an immersive level-editing system. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 195–196. IEEE, 2013.

[68] C. Ware and K. Lowther. Selection using a one-eyed cursor in a fish tank VR environment. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 4(4):309–322, 1997.

[69] M. Weise, R. Zender, and U. Lucke. A Comprehensive Classification of 3D Selection and Manipulation Techniques. In *Proceedings of Mensch und Computer 2019*, pp. 321–332. 2019.

[70] D. Wilkie, J. Sewall, and M. C. Lin. Transforming GIS data into functional road models for large-scale traffic simulation. *IEEE transactions on visualization and computer graphics*, 18(6):890–901, 2011.

[71] D. Zielasko, M. Krüger, B. Weyers, and T. W. Kuhlen. Passive Haptic Menus for Desk-Based and HMD-Projected Virtual Reality. In *2019 IEEE 5th Workshop on Everyday Virtual Reality (WEVR)*, pp. 1–6. IEEE, 2019.

[72] D. Zielasko and B. E. Riecke. Sitting vs. Standing in VR: Towards a Systematic Classification of Challenges and (Dis) Advantages. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 297–298. IEEE Computer Society, 2020.