Modeling Across Realities (MAR): Extending Desktop 3D Modeling with Augmented Reality

Mucahit Gemici, Kyle Johnsen, Mayra Donaji Barrera Machuca, Anil Ufuk Batmaz

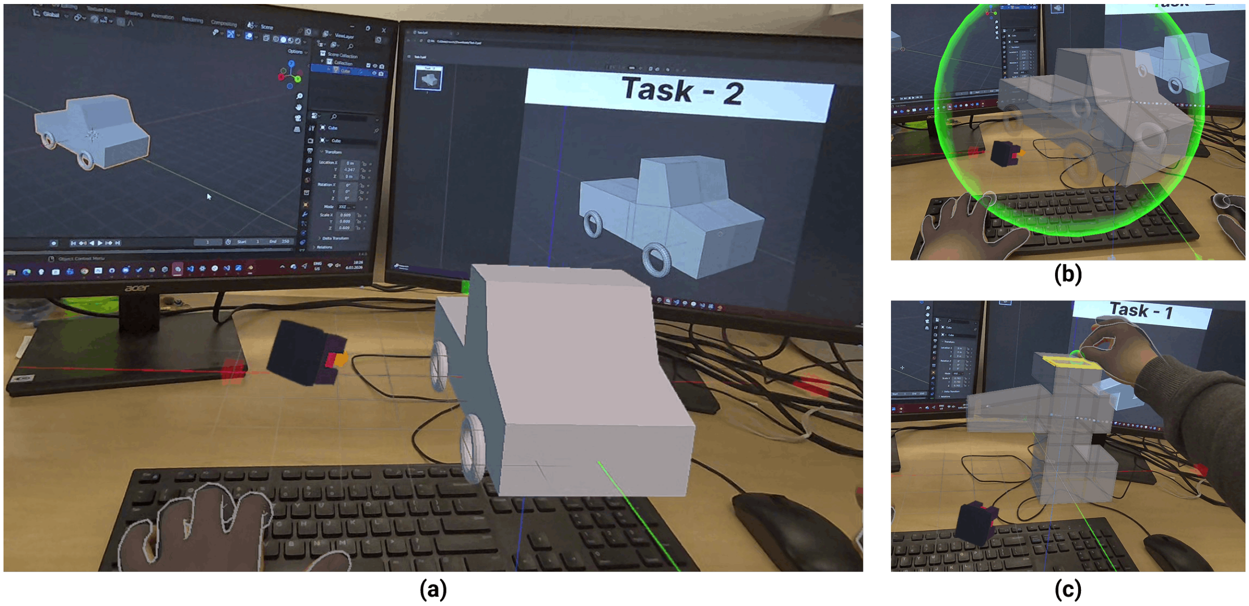


Figure 1: (a) MAR visualizes the model from Blender (3D modeling application) in immersive AR. (b) In inspection mode, MAR allows the user to manipulate the model's position, rotation, and scale for spatial inspection. (c) The user performs a face selection through a pinch gesture with the dominant hand.

Abstract— Three-dimensional modeling (3D Modeling) is an essential part of creating digital content across various industries. However, most industry-standard tools rely on conventional desktop setups with 2D screens. One possible solution is to use Augmented Reality (AR), which allows users to perceive 3D shapes in a body-centric space better, but this increased spatial understanding comes at the cost of precision. To solve this problem, we present the MAR (Modeling Across Realities) system, a cross-reality 3D modeling system that combines the precision of conventional desktop-based 3D modeling with immersive AR visualization and interaction. We systematically evaluated MAR by asking 20 participants to model an abstract shape and an everyday object. Our results show that MAR outperformed desktop-based 3D modeling with lower workload, higher usability, and user preference without compromising the model quality in the abstract shape task, and improving model quality in the everyday-object task. These outcomes indicate that MAR is a promising cross-reality approach with the potential to inform the design of future 3D modeling systems.

Index Terms—Polygonal Modeling, 3D Modeling, Augmented Reality, Cross-Reality, Gesture Interactions

1 Introduction

Three-dimensional modeling (3D Modeling) is the computer graphics process of creating a digital representation of a three-dimensional object or shape using specialized software [22, 47]. It is an essential process for industries such as film, gaming, architecture, healthcare, and manufacturing [4, 60], as it enables the virtual replication of real objects. Advancements in 3D printing technologies have also increased the demand for 3D models [51], as people need to use them as a base for the printing process. However, even with increased demand and interest, most industry-standard modeling tools remain conventional, relying on 2D desktop workflows.

Creating a 3D shape on a 2D display is found to be tricky due to the lack of dimensionality, causing less intuitive interaction [62], as well as cognitively demanding due to difficulties in viewpoint control and depth estimation [34]. For example, users may misinterpret the 3D shapes on the screen as they lack stereo perception on a 2D display, they also must frequently adjust the viewpoint to understand the spatial structure of the shape. Furthermore, when users familiar with 2D applications use 3D software, they may use 2D operations such as zooming or panning for 3D navigation, causing unintentional movement, or they may lose track of their position in 3D space, a phenomenon known as 3D amnesia [21].

Immersive technologies, such as eXtended Reality (XR) and Augmented Reality (AR), are promising alternatives for 2D displays to mitigate the spatial disorientation and cognitive load [13], that are associated with 2D displays by offering high immersion [42], high spatial awareness [29], and more realistic depth perception [3]. These technolo-

• *Mucahit Gemici is with Concordia University. E-mail: mucahit.gemici@mail.concordia.ca.*
• *Kyle Johnsen is with University of Georgia. E-mail: kjohnsen@uga.edu.*
• *Mayra Donaji Barrera Machuca is with University of Calgary. E-mail: mbarrera@ucalgary.ca.*
• *Anil Ufuk Batmaz is with Concordia University. E-mail: ufuk.batmaz@concordia.ca.*

Manuscript received xx xxx. 202x; accepted xx xxx. 202x. Date of Publication xx xxx. 202x; date of current version xx xxx. 202x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.202x.xxxxxxx

gies place users in spatially immersive workspaces [18] and support embodied interaction, changing how users perceive and interact with the digital objects [31]. By allowing users to observe and manipulate 3D objects as they would in the real world using a 6-degree-of-freedom (6DoF) device, immersive environments can mitigate the cognitive load caused by "dimensionality crossing", i.e., mental transformation from 2D images into 3D structures [13, 30, 40].

While immersive technologies have the potential to support 3D modeling by enabling spatial viewing and interaction with virtual objects [32], they do not inherently address the high precision required for detailed 3D editing [2, 70]. This limitation is particularly relevant and critical in polygonal modeling, where users construct and refine meshes by manipulating vertices, edges, and faces through structured editing operations.

Desktop mouse-and-keyboard input, by contrast, provides precision and stability needed for such fine-grained editing, but offers weaker depth and spatial cues than immersive environments [11, 12]. These complementary strengths motivate combining the two: the desktop for precise editing and immersive AR for spatial perception and understanding of 3D objects. Rather than replacing the desktop workflow, we integrate both within a single cross-reality workflow.

To explore this integration, we present MAR, a cross-reality 3D modeling system that combines the strengths of desktop and AR environments. With MAR, users perform polygonal modeling with a conventional keyboard-and-mouse interface using an industry-standard 3D modeling software, i.e., Blender, which preserves the precision, stability, and familiarity needed for detailed mesh operations. At the same time, a video see-through AR interface brings the developing shape into the users' real environment, supporting bimanual hand-based manipulation and selection, and allowing users to examine the developing shape from multiple perspectives. Beyond enhancing perspective viewing and direct manipulation, MAR can support scenarios such as 3D printing and product design, where precise desktop editing must be combined with spatial inspection of the object in its real environment before fabrication. In this study, our contributions are:

- A cross-reality 3D modeling system that enhances industry-standard desktop modeling workflows with Augmented Reality.
- A bimanual hand interaction design for object manipulation and mesh element selection (vertices, edges, and faces) in the Augmented Reality interface.
- An empirical evaluation demonstrating that the proposed cross-reality system improves user experience and system usability, reduces workload, and decreases viewport changes during 3D modeling tasks.

2 Related Work

2.1 Immersive and Hybrid 3D Modeling

Immersive environments such as VR and AR are commonly used for 3D content creation in HCI research, and this combination has been explored with multiple interaction styles. For instance, a common approach is to sketch the outline of a shape in VR to create a 3D drawing. VRSketchIn [16] investigated using a 6DoF tracked pen and tablet for mid-air sketching in VR. While they support drawing lines in mid-air, Jackson and Keefe [32] transferred 2D hand-drawn lines in VR, then lifted the digital curves by freehand sketching to create 3D surfaces. Beyond sketching, some systems explored modeling workflows such as VRCreatIn [17] integrating modeling, lighting, and texturing in VR with pen-based input. However, using such a tangible controller may be inconvenient to carry [36], and a pen-based input may not provide the precision required in 3D modeling.

Lam et al. investigated 3D model editing with vertex manipulations through hand gestures due to the inconvenience of tangible controllers, especially for mobile usage scenarios [36]. Their study shows that immersive environments are not only promising for ideation (e.g., sketching), but also for editing and manipulating 3D shapes. However, prior work reports precision limitations for immersive input modalities, such as mouse-based interaction achieving higher precision than pen-based [48], and hand-based [59,70] inputs in precise 3D positioning and manipulation tasks, partly because hand jitter and tracking inaccuracies affect fine-grained movements [10, 24, 54]. This limitation is especially important for 3D modeling, where users perform precise operations on vertices, edges, and faces through structured editing commands.

Beyond fully immersive sketching and editing, prior work has also explored 3D modeling with hybrid environments. These systems suggest that immersive 3D design does not have to rely entirely on mid-air interaction; instead, they can benefit from mixed setups that provide additional control. For instance, Mockup Builder [14] investigated 3D modeling on and above a multitouch surface in a semi-immersive environment, while SymbiosisSketch [1] combined mid-air 3D drawing with 2D surface sketching, leveraging complementary affordances of 3D and 2D interactions for 3D conceptual design. Similarly, MixFab [66] showed how mixed-reality environments can support the creation of 3D objects with hand gestures for personal fabrication.

More closely related to our work, DesignAR [51] and DualCAD [45] investigate how immersive depth cues can support 3D modeling in combination with more controlled workspaces, such as multitouch screens and desktops. DesignAR integrates a 2D interactive surface with AR to expand the screen real estate and proposes the metaphor of Augmented Displays. While their work contributes to cross-reality 3D modeling systems, they focus on simple single-object modeling scenarios and leverage AR as a display extension tool, which may limit the potential benefits of the AR. On the other hand, DualCAD combines a conventional desktop CAD environment with an AR mode, allowing users to switch between precise desktop input and mobile device-based pointing and interaction. While this demonstrates the value of combining desktop precision with immersive depth cues, it introduces an explicit transition between desktop and AR interaction and adds an extra device to the workflow (leaving the keyboard and mouse, then taking the mobile device), which may increase workflow overhead and disrupt user focus [68].

These studies show that the key challenge is not only combining immersive visualization with structured workspaces, but also enabling smooth coordination between them. Harlan et al. [27] identified two challenges between conventional desktop CAD workflows and immersive systems: the data gap, i.e., the difficulty of moving data between CAD and XR due to mesh optimizations for XR rendering, and the interface gap, i.e., the difficulty of moving between desktop and XR input. MAR presents and evaluates a cross-reality modeling system that supports continuous Blender-to-AR synchronization and bidirectional interaction while keeping Blender as the source of truth. In this way, MAR provides a concrete step forward in addressing the data and interface gaps discussed in prior hybrid modeling work.

2.2 Cross-Reality and Hybrid Spatial Interaction

Combining environments at different points of reality-virtuality continuum [44] is a long-standing strategy for balancing the strengths of conventional and immersive interfaces. When a single user works across this continuum within one task, the resulting setup is referred to as Cross-Reality (CR). Wang and Maurer [64] organized this space into a design space for single-user CR applications, and later work has examined when such combinations help and when they introduce overhead [63]. MAR follows this single-user framing: the user stays at a desktop workstation for precise mesh editing, while an AR view registered to the same physical space supports spatial inspection and interaction. This framing, however, was developed for analytics and visualization rather than for 3D modeling.

In visualization and data analysis, several systems combine a conventional 2D environment with AR. Besançon et al. [7] surveyed spatial interfaces for 3D visualization, where a recurring idea is to augment an existing, familiar display rather than replace it. Reipschläger et al. [52] extended large interactive displays with personal AR for information visualization under the concept of Augmented Displays, which was also applied to 3D modeling in DesignAR [51]. The same strategy recurs across device classes: AR has been used to extend desktop-based data analysis tools [65], to complement mobile devices in MARVIS [37], to enlarge the collaboration space around wall displays [33], and to provide an AR second screen for desktop AI-assisted authoring [23].

Related setups also span AR and VR through handheld touch input [43], and AR has additionally been surveyed as a medium for collaborative work [58]. Across these systems, the conventional display preserves precise and familiar input while AR adds spatial perception and additional viewpoints, a principle we adopt for 3D modeling, where users must perform structured operations on vertices, edges, and faces rather than explore an existing dataset.

More closely related to the interaction technique in our system, prior work has studied how to select content within and move objects across realities. Sereno et al. [57] combined touch and tangible input for hybrid spatial selection in AR. Rau et al. [49] examined techniques for transitioning 3D objects between desktop and AR, and the broader problem of moving between a workstation and an immersive view was studied early on by Kijima and Ojika [35] using a projective head-mounted display. In MAR, these mechanisms support a structured mesh editing process rather than data exploration or object handover, extending the cross-reality paradigm to polygonal modeling.

3 Modeling Across Realities (MAR)

We designed Modeling Across Realities (MAR) to extend the conventional 3D modeling workflows that rely on 2D displays with immersiveness [42], spatial awareness [29], and the depth perception [3] of the AR. Our goal is to combine the high precision of keyboard-and-mouse with the spatial understanding provided by immersive AR.

3.1 Mesh Transfer

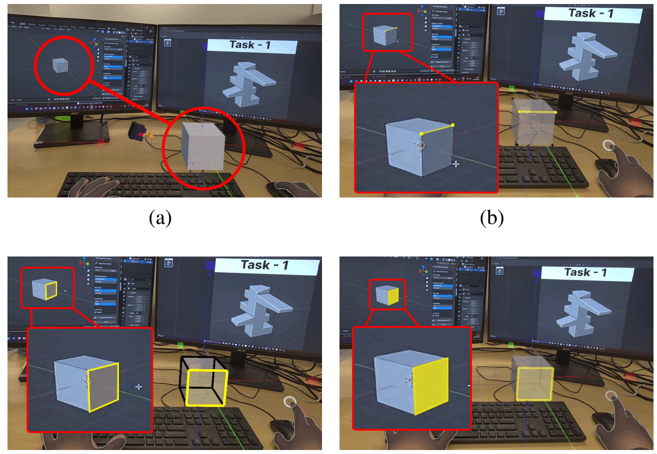
Our system, MAR, relies on mesh transfer from the 3D modeling application Blender to the AR interface in Unity. To achieve this, we developed a custom add-on in Python that runs within Blender's Python environment. During modeling, the add-on periodically checks the active mesh using the Blender Python API [9]. It assigns persistent IDs to each vertex, edge, and face, and transfers updated mesh data, including edge connectivity, whenever the topology or geometry changes. We transfer the mesh only when changes are detected to avoid overloading the communication pipeline and disrupting seamless interaction between Blender and AR.

Communication is handled through a Fast API server using JSON, and Unity reconstructs the mesh from the transmitted IDs and local vertex positions at runtime when a change is detected. This lets us reconstruct geometry from continuously streamed data rather than repeatedly exporting and importing full mesh files, which could introduce higher latency. We also visualize axes and grid lines in AR to help users establish a spatial correspondence between the object in AR and the Blender workspace (see Figure 2(a)). To prevent overloading the AR view, these are optional and can be toggled with a button click (see Figure 3(a-b)).

3.2 Hand Inspection

MAR also includes a synchronization mode for the currently modeled object. When synchronization is enabled, Blender controls the object's position, rotation, and scale, and any manipulation in Blender is reflected in AR by appending transform information to the JSON data sent from Blender. However, strictly following Blender's transforms would limit the benefits of AR, as users would be restricted to use keyboard and mouse. To address this, we designed a hand-based inspection mode that becomes available when synchronization is disabled through a digital button in the AR interface (see Figure 4(a)). The AR interface visual is shown in Figure 3(a). We positioned this interface on top of the monitors (see Figure 3(b)), allowing both direct button press and point-and-pinch.

When synchronization is disabled, the positional, rotational, and scale link between the AR object and the Blender object is temporarily disabled, giving the user full control in AR. A green outline appears around the object to indicate that it is interactable. The user grabs the object with one hand to reposition and rotate it, and with both hands to scale it, moving the hands apart to enlarge or together to shrink. This hand-based manipulation is intended for inspection. When the user re-enables synchronization, they select the same UI button again, which hides the green outline and animates the object back to the transform specified in Blender. We keep this transform synchronization one-way: Blender remains the source of truth for the object's transform, and the position, rotation, and scale changes made in AR act as temporary inspection transforms that are not written back to Blender. This lets users freely reposition, rotate, and scale the object in AR without unintentionally harming the transform needed for continued precise editing on the desktop. Only the AR-side inspection transform is decoupled; the mesh stays synchronized at all times, so any selection or edit still refers to the same geometry. Full bidirectional transform synchronization can be valuable in other cross-reality systems, for instance, to manipulate the object directly in the immersive view [67], but it does not fit our goal of preserving a stable desktop editing state during inspection.

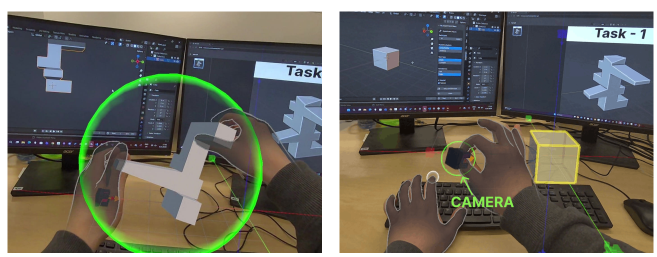


(a) (b) (c) (d)

Figure 2: Visualization of the object in Blender (a) Object mode, (b) Edit mode in vertex domain, (c) Edit mode in edge domain, and (d) Edit mode in face domain. Mesh elements are highlighted differently per domain.

(a) (b)

Figure 3: Digital UI (a) as a visual and (b) in the AR environment. The menu has two buttons: Axis Visualization, which toggles the axes and grid lines, and Transform Sync, which changes the synchronization mode.



(a) (b)

Figure 4: (a) The system is in inspection mode, highlighted with a green outline around the object. (b) User changes the Blender view by manipulating the camera simulation.

3.3 Viewpoint Control

We also designed a hand-gesture viewpoint mechanism to support navigation and reduce loss of orientation [21,46]. Khan et al. [34] proposed ViewCube, a corner widget that indicates and controls viewing direction in 3D CAD tools. Inspired by this idea, we designed a viewpoint simulator that shows the camera position and orientation in AR. It allows users to understand their current viewpoint and to change it by grabbing the simulated camera (see Figure 4(b)) and releasing it at a desired location around the model. Our viewpoint control mechanism also supports orthographic views through predefined transparent camera visuals placed around the model. Releasing the simulated camera onto one of these predefined views changes the Blender view to the corresponding orthographic view.

3.4 Mesh Element Selection

Many mesh operations require selecting vertices, edges, or faces before editing. Because MAR allows users to inspect the model spatially, enabling mesh element selection in the same view lets them directly act on the part they identify. To support this, we designed a bimanual hand-based selection system. In our design, the dominant hand is used for selection, and the non-dominant hand is used for deselection. We assigned asymmetric roles to the two hands following prior work on bimanual interaction in XR [41], where the non-dominant hand supports actions performed by the dominant hand. We placed adjustable spheres in front of the index fingers to indicate the selection and deselection areas, making the technique adaptable for both multi-selection and one-by-one selection. By default, these spheres were transparent gray to indicate inactivity, as shown in Figure 2(b-d). Users activated them with a pinch gesture, which changed the sphere to green for selection (Figure 6(a)) or red for deselection (Figure 6(d)). We chose pinch gesture, rather than grab, for selection because our system supports both object manipulation and mesh selection, and this distinction helps avoid unintended selections or grabs. To adjust the sphere size, the other hand pinches the sphere's surface and moves it away to enlarge it or closer to reduce it, as shown in Figure 6(b). The selection domain follows Blender's selection modes, and the system visualizes mesh elements differently depending on the current mode, by highlighting vertices in vertex mode (Figure 2(b)), thickening edges in edge mode (Figure 2(c)), and hiding vertices and making edges less visible to highlight faces in face mode (Figure 2(d)).

When a mesh element enters the selection or deselection sphere, AR (Unity) sends Blender the corresponding persistent IDs and local positions. Requests are processed through our custom add-on, and the updated visual state is sent back to AR after processing. Figure 6(c) shows a user selecting multiple vertices with an enlarged selection sphere. To maintain both visual and behavioral consistency, Blender serves as the source of truth. We intentionally avoided implementing two separate selection systems, since the additional synchronization logic could introduce visual inconsistencies, erroneous edits, or disruptions in seamless cooperation between AR and Blender. Selection works whether synchronization is enabled or disabled, allowing interactions such as rotating the object with one hand and then selecting mesh elements with the other. The high-level workflow diagram of MAR is given in Figure 5.

Pipeline latency was low: mesh updates from Blender to Unity had mean/median latencies of 33.16/31.63 ms, and selection commands from Unity to Blender 62.85/62.77 ms.

4 User Study

We designed our evaluation to examine how adding AR to a desktop modeling workflow affects user behavior and modeling outcomes as a whole, rather than to isolate the effect of each cross-reality feature.

Following this exploratory framing, we investigated the following research questions: **RQ1:** How does MAR affect modeling performance, spatial inspection, and navigation, as measured by task completion time, model quality, positional and angular view change, and post-edit inspection time? **RQ2:** How does MAR affect perceived workload, usability, user experience, and preference?

4.1 Apparatus

The user study was conducted on a desktop PC with an Intel(R) i7-12700F processor at 2.1GHz, 16GB RAM, and an NVIDIA GeForce RTX 3070 graphics card. The computer had two 27-inch Acer V277U monitors. To keep the desktop interface readable, we set the screen resolution to 1280 x 720 during the study. The keyboard was a Dell KB216, and the mouse was a Dell MS116. As the AR Head Mounted Display, we used a Meta Quest 3 video see-through AR. The AR application was implemented using Unity 6000.0.62f1 with Meta XR All-In-One SDK 83.0.1 as a PC based application run via Meta Quest Link. The Blender add-on application was implemented with Blender 3.4.0 and Python 3.13.5.

4.2 Participants

We performed a prior power analysis using G*Power software [19] with a significance criterion of $\alpha = .05$ and power = .90, we calculated the minimum sample size needed with a large effect size ($\eta^2 = 0.14$) [69] as 20 for RM ANOVA. Thus, we recruited 20 participants (10 female, 10 male) aged 20-32 (M = 25.1, SD = 2.99). One participant was left-handed, but all preferred to use their right hand for the mouse. Out of 20 participants, eight had normal vision, and twelve had corrected-to-normal vision. In terms of AR experience, ten participants reported 0-2 hours, one 2-4, two 4-6, one 6-8, one 8-10, and five more than 10 hours. We had a novice-heavy participant group, with eight participants stated no prior 3D modeling experience, seven stating that they are novices, three stating intermediate, one advanced, and one stating expert.

4.3 Tasks

We evaluated our system with two different polygonal modeling tasks, T-ABSTRACT and T-TRUCK. In the T-ABSTRACT, participants were asked to recreate an abstract shape which is used in prior work on 3D sketching studies [5, 6]. With an abstract shape, we aimed to reduce their reliance on prior knowledge, and bring their attention to geometric relationships and spatial understanding of the developing shape. In T-TRUCK, we used a familiar truck concept, which was previously used in 3D modeling studies [20], in a more complex mirrored modeling setup where they could change the shape only in negative x-axis in the viewport. Both tasks started from a cube shape, and required them to perform mesh editing operations to reach the target shape. The target shapes for both tasks can be seen in Figure 7(a-b). These tasks allowed us to evaluate the MAR under both abstract shape reconstruction and familiar-object modeling scenarios. We kept both tasks controlled and refined them through pilot testing, so the study remained feasible for our novice-heavy group while still exercising common modeling operations: extrusion, vertex, edge, and face manipulation, repeated refinement, and mirrored modeling.

4.4 Procedure

Upon arrival, participants were briefed and given a consent form to read and sign if they chose to participate. This study was approved by our institutional research ethics board with an approval number of *30023195*. The study consisted of two steps of training: Blender training and MAR system training. In Blender training, the experimenter taught them how to use the Blender interface, the operations they need to use to complete the tasks, and their shortcuts. Also, a paper containing operation names and corresponding shortcuts is given to the user. This sheet was placed next to the desktop screen and remained available throughout both training and the study, and participants could pick it up at any time. The Blender tutorial took 20 minutes on average, and participants were asked to design a simple chair model (see Figure 7(c)) to complete the Blender tutorial. Participants then performed the MAR tutorial, testing the system until they felt comfortable. The second step of the tutorial took approximately 10 minutes. Participants then performed the conditions, and filled out SUS, NASA-TLX, and UEQ questionnaires after each condition. Performing the conditions and filling out the questionnaires took 50-60 minutes on average.

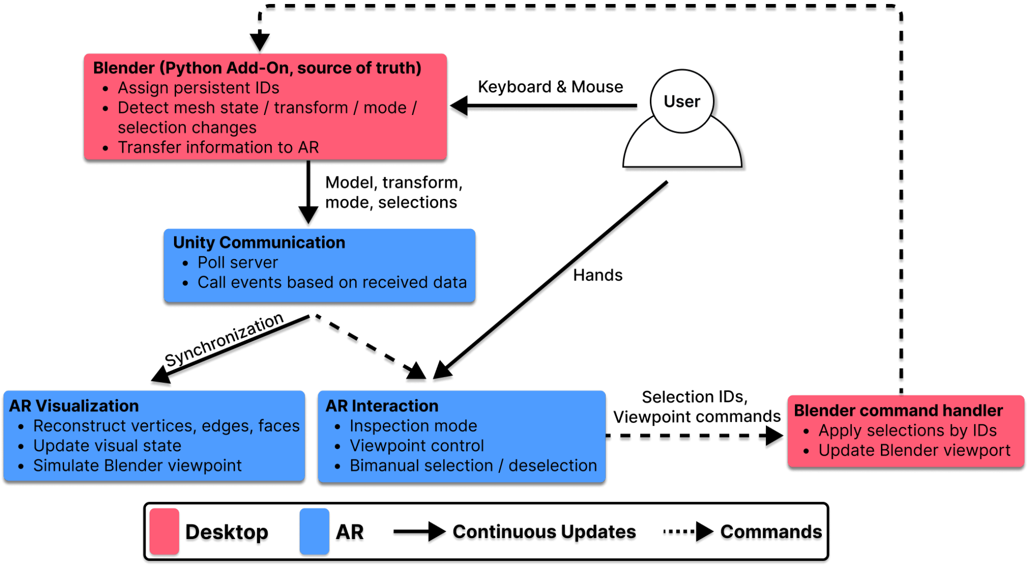
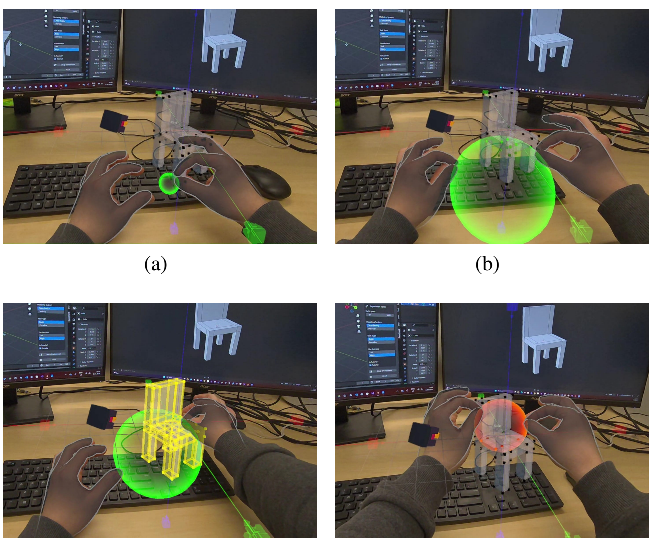


Figure 5: High-level workflow of MAR. Solid lines represent continuous updates and dotted lines represent user-triggered commands. Blender remains the primary polygonal editor, while our Python add-on transfers data to AR (Unity). In AR, the model is reconstructed for visualization, and interaction tools are provided, such as inspection mode, viewpoint control, and selection technique. Orange and green boxes denote Desktop and AR operations, respectively.



(a) (b) (c) (d)

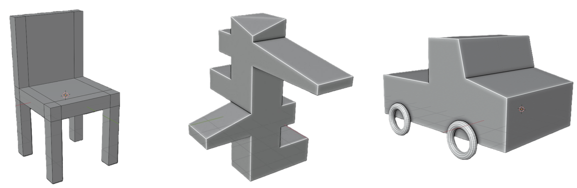
Figure 6: The vertex selection procedure: (a) Activating the selection sphere with a pinch gesture, (b) adjusting the sphere size, (c) performing multiple vertex selections, and (d) deselecting vertices.

4.5 Experimental Design

We used a within-subjects experiment design with 2 Modeling Systems (MS_2: {MAR, DESKTOP }) and 2 Tasks (T_2: {T-ABSTRACT, T-TRUCK }), resulting in 4 conditions in total. To control potential order effects, the order of conditions was counterbalanced across N=20 participants using a Latin Square Design.

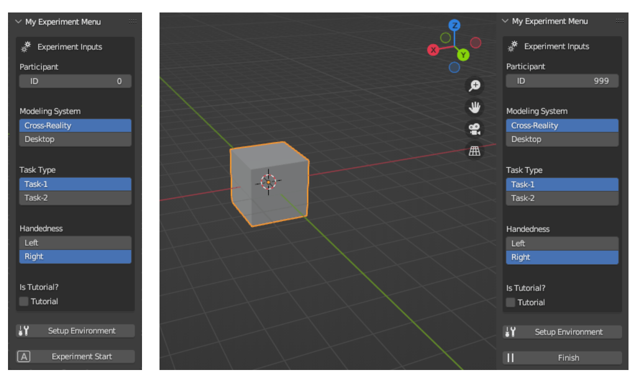
4.6 Evaluation Metrics

We have recorded the following metrics to evaluate the system. **Task Time** (s) refers to the time between users click on the "Start Experiment" and "Finish" buttons in the Blender. These buttons are shown in Figure 8(a-b). **Positional View Change** (m) refers to the amount of camera viewpoint position change in the Blender application. **Angular**



(a) (b) (c)

Figure 7: The target shapes used in the user study. Participants were asked to reproduce the shapes (a) for Blender tutorial, (b) for T-ABSTRACT, and (c) for T-TRUCK.



(a) (b)

Figure 8: (a) The Blender add-on UI for controlling the experiment (participant ID, mouse hand, task type); pressing "Experiment Start" begins it. (b) During the run, the button becomes "Finish", which ends the experiment.

View Change (°) refers to the camera viewpoint direction change in Blender. We recorded positional and angular view changes as indicators of how much participants relied on desktop viewport navigation to understand the model. **Post Edit Inspection Time** (s) refers to the

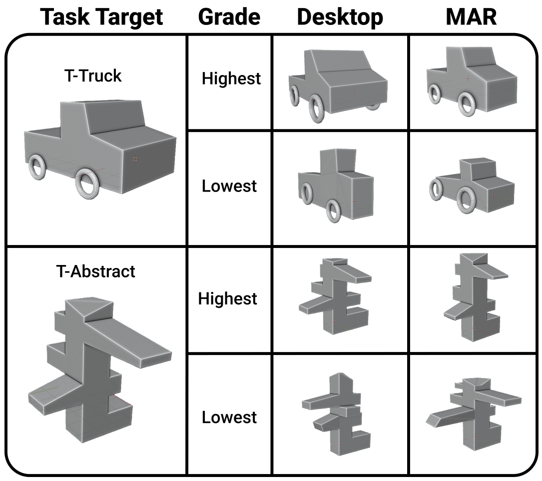


Figure 9: Example models of the highest and lowest graded models done by participants between DESKTOP and MAR.

amount of time between the users' last edit and the time they pressed the "Experiment End" button, capturing how long participants spent verifying the model after their final modeling operation. **Model Grade** (%) refers to the model quality grade determined by two evaluators on a scale from 0 to 100. Each evaluator assessed the models based on their similarity to the reference shape, and then the average of the scores determined the final grade of the model. For T-ABSTRACT, evaluation considered extrusion accuracy, edge manipulations, vertex manipulations, and overall shape similarity, with misplaced or missing extrusions reducing the score. For T-TRUCK, the evaluation considered cabin shape, wheel size, wheel position, and overall shape similarity, with size discrepancies and misaligned vertices reducing the score.

In addition to task metrics, we also asked participants to fill out System Usability Scale (SUS) [39] to measure usability of the system, NASA-TLX [28] to measure the perceived workload, and User Experience Questionnaire (UEQ) [56] to measure user experience of the system. Furthermore, we asked participants to fill out a preference question at the end of the experiment with their reasoning for the preference.

We also recorded exploratory modeling behavior metrics (deformation time and ratio, selection change frequency, topology change frequency) and computed F-score and Chamfer distance following prior work [61]. Because that work used high-poly meshes, these geometry-based measures were less informative for our relatively simple task models, so we did not use them as primary measures. Detailed figures and analyses of exploratory and alternative model quality assessment measures, together with per-subscale NASA-TLX and UEQ figures, are provided in the supplemental material. The project source code of the system is publicly available under MIT License[1].

5 RESULTS

The data were analyzed using SPSS 29. We considered it to be normally distributed if Skewness (S) and Kurtosis (K) were within ±1 [25, 26]. If not, we performed a log-transform to yield a normal distribution. We first evaluated task metrics with RM ANOVA and a Bonferroni-corrected pairwise test, and then proceeded with the assessment of the questionnaire data with Friedman test and the Wilcoxon post-hoc test.

5.1 Task Results

Although task completion time is often positively skewed, in our data, skewness and kurtosis of Task Time ($S = 0.83, K = -0.05$) were within our stated ±1 criterion, so we analyzed raw task time without a log transformation (an individual-point distribution plot for Task Time is provided in the supplemental material). Positional View Change ($S = 0.9, K = 0.19$), Angular View Change ($S = 1, K = 0.64$), and Model Grade ($S = -0.64, K = -0.4$) were normally distributed. Post Edit Inspection Time was normally distributed after a log-transform ($S = 0.7, K = 0.9$).

Task Time: RM ANOVA results showed significant differences in MODELING SYSTEM and interaction of TASK and MODELING SYSTEM on Task Time (see Table 1). According to Figure 10(a), the participants were slower with MAR in T-ABSTRACT, while observing no significant time difference in T-TRUCK.

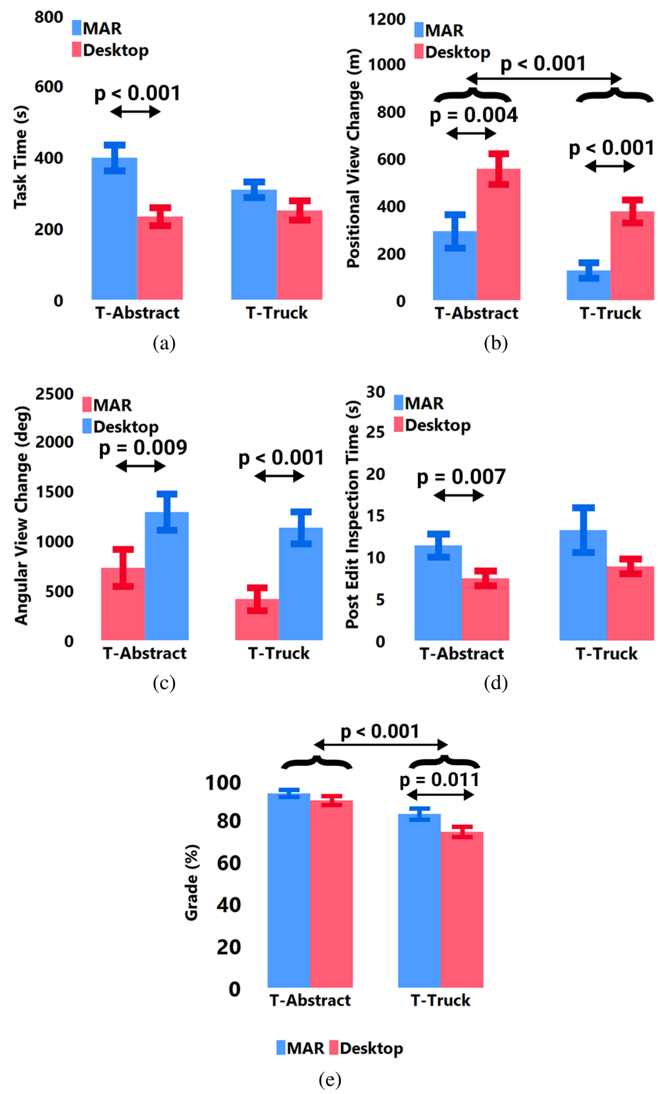


Figure 10: RM ANOVA results of the task metrics: (a) Task Time (s), (b) Positional View Change (m), (c) Angular View Change (°), (d) Post Edit Inspection Time (s), and (e) Model Grade (%).

Positional View Change: Table 1 shows that MODELING SYSTEM and TASK had a significant main effect on Positional View Change. Participants moved the camera significantly less with MAR compared to DESKTOP in both tasks, while T-ABSTRACT resulted in higher camera movement compared to T-TRUCK (Figure 10(b)). Because the viewing scale in Blender can be adjusted freely, we also recomputed a scale-normalized version of this metric (S = 0.85, K = 0.167) and found the same pattern: significant main effects of MODELING SYSTEM (F(1, 19) = 20.087, $p < 0.001$, $\eta^2 = 0.514$) and TASK (F(1, 19) = 5.012, $p = 0.037$, $\eta^2 = 0.209$), and no significant interaction (F(1, 19) = 0.276, p = 0.606, $\eta^2 = 0.014$); figure in supplemental material.

Angular View Change: According to Table 1, MODELING SYSTEM had a significant main effect on Angular View Change, as MAR

[1]https://github.com/mucahitGemici/Modeling-Across-Realities

significantly reduced the angular camera movements of the participants in both T-ABSTRACT and T-TRUCK (Figure 10(c)).

Post Edit Inspection Time: We found a significant main effect of MODELING SYSTEM on Post Edit Inspection Time (Table 1). Post-hoc revealed that participants inspected the model they created significantly longer with MAR compared to DESKTOP before completing the T-ABSTRACT (Figure 13(d)).

Model Grade: We observed a significant main effect of both MODELING SYSTEM and TASK on Model Grade (Table 1). Results show that participants got significantly higher scores in recreating the shape with MAR compared to DESKTOP in T-TRUCK. Models with the highest and lowest grades are shown in Figure 9.

Additional figures for exploratory modeling behavior metrics and the alternative geometry-based quality measures are included in the supplemental material.

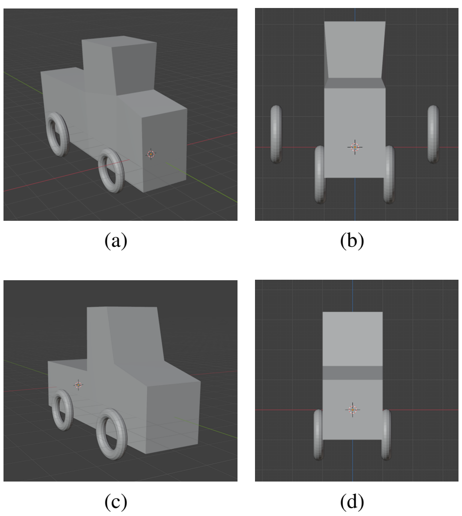


Figure 11: Resulting models of P17 in T-TRUCK. (a) Final DESKTOP model and (b) its orthographic view, revealing the positioning mistake; (c) final MAR model and (d) its orthographic view.

5.2 SUS Results

The Friedman test revealed a significant difference in the SUS Score ($\chi^2(3,N=20)=27.177$, $p<0.001$, $W=0.453$). According to Figure 12, MAR condition achieved significantly higher usability scores than DESKTOP condition in both T-ABSTRACT and T-TRUCK. Table 2 shows the average SUS scores and corresponding letter grades determined based on the Sauro-Levis grading scale [38, 55], demonstrating the superiority of MAR over DESKTOP with A and A+ grades (vs C and D, respectively).

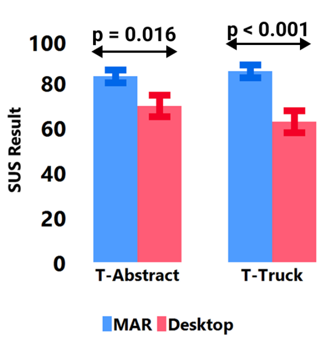


Figure 12: Graph comparing the SUS scores of MODELING SYSTEM and TASK. Pairwise comparisons are shown with the arrows.

5.3 NASA-TLX Results

We found significant differences in NASA-TLX scores for Mental Demand ($\chi^2(3,N=20)=19.406$, $p<0.001$, $W=0.323$), Temporal Demand ($\chi^2(3,N=20)=8.777$, $p=0.032$, $W=0.146$), Performance ($\chi^2(3,N=20)=18.113$, $p<0.001$, $W=0.302$), Effort ($\chi^2(3,N=20)=13.741$, $p=0.003$, $W=0.229$), Frustration ($\chi^2(3,N=20)=19.386$, $p<0.001$, $W=0.323$), and Overall Score ($\chi^2(3,N=20)=16.553$, $p<0.001$, $W=0.276$). In T-ABSTRACT, MAR resulted in significantly lower Mental Demand, Temporal Demand, and Overall Scores than DESKTOP (see Figure 13(a)). In contrast, in T-TRUCK, participants perceived significantly lower workload on all NASA-TLX subscales (except Physical Demand) for MAR compared to DESKTOP (see Figure 13(b)).

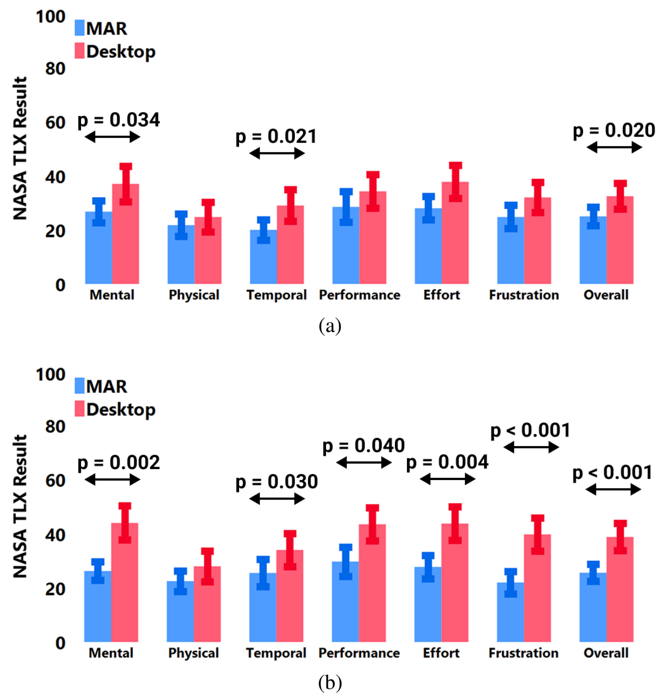


Figure 13: NASA-TLX Results comparing MAR and DESKTOP for (a) T-ABSTRACT, (b) T-TRUCK

5.4 UEQ Results

UEQ evaluation revealed significant differences in Attractiveness ($\chi^2(3,N=20)=32.952$, $p<0.001$, $W=0.549$), Dependability ($\chi^2(3,N=20)=11.445$, $p=0.010$, $W=0.191$), Efficiency ($\chi^2(3,N=20)=14.567$, $p=0.002$, $W=0.243$), Novelty ($\chi^2(3,N=20)=34.016$, $p<0.001$, $W=0.567$), Perspicuity ($\chi^2(3,N=20)=16.946$, $p<0.001$, $W=0.282$), and Stimulation ($\chi^2(3,N=20)=29.225$, $p<0.001$, $W=0.487$). MAR outperformed DESKTOP in all UEQ subscales for both T-ABSTRACT (Figure 14(a)) and T-TRUCK (Figure 14(b)).

5.5 Preference & Qualitative Feedback

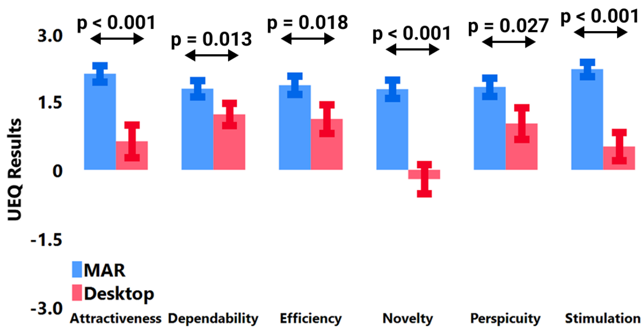
We also asked participants which modeling system they would prefer overall. Eighteen participants preferred MAR, one preferred DESKTOP, and one indicated their preference depended on the task. When asked for the reason, participants most frequently stated improved spatial understanding from viewing the model in their environment with MAR (P1, P3, P4, P7, P8, P14, P16, P17, P18, P20). For example, P4 noted that "AR made visualizing the object much more intuitive", and P17 stated that "With AR, I can see the depth very clearly and I do not need to image in my mind." Participants also reported that MAR reduced the need to repeatedly adjust the desktop viewpoint (P8, P9,

Table 1: RM ANOVA results of the task metrics. The significant results are highlighted with bold text.

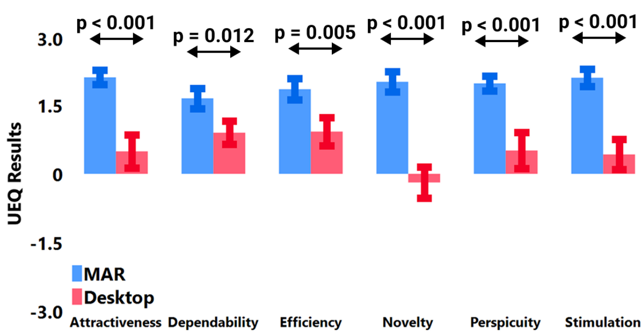
	Modeling System	Task	Modeling System x Task
Task Time	**F(1, 19) = 19.897, p** $< \mathbf{0.001}$**,** $\eta^2 = 0.512$	F(1, 19) = 2.678, p = 0.118, $\eta^2 = 0.124$	**F(1, 19) = 6.729, p = 0.018,** $\eta^2 = 0.262$
Positional View Change	**F(1, 19) = 25.986, p** $< \mathbf{0.001}$**,** $\eta^2 = 0.578$	**F(1, 19) = 13.954, p** $< \mathbf{0.001}$**,** $\eta^2 = 0.423$	F(1, 19) = 0.028, p = 0.870, $\eta^2 = 0.001$
Angular View Change	**F(1, 19) = 22.014, p** $< \mathbf{0.001}$**,** $\eta^2 = 0.537$	F(1, 19) = 3.745, p = 0.068, $\eta^2 = 0.165$	F(1, 19) = 0.559, p = 0.464, $\eta^2 = 0.110$
Post Edit Inspection Time	**F(1, 19) = 7.158, p = 0.015,** $\eta^2 = 0.274$	F(1, 19) = 1.206, p = 0.286, $\eta^2 = 0.060$	F(1, 19) = 0.376, p = 0.547, $\eta^2 = 0.019$
Model Grade	**F(1, 19) = 9.810, p = 0.005,** $\eta^2 = 0.340$	**F(1, 19) = 35.786, p** $< \mathbf{0.001}$**,** $\eta^2 = 0.653$	F(1, 19) = 1.991, p = 0.174, $\eta^2 = 0.095$

Table 2: SUS Scores and corresponding grades based on the Sauro-Levis grading scale [38, 55].

	MAR	Desktop
T-Abstract	82.875 (A)	69.625 (C)
T-Truck	85.125 (A+)	62.625 (D)



(a)



(b)

Figure 14: UEQ Results for (a) T-Abstract, (b) T-Truck

P13, P17); for instance, P8 highlighted that they could "visualize the model better and make changes rather than fixing the camera position every time in Blender". The two non-majority responses reported technical limitations of AR (P6) and that preference may depend on the task and the convenience of keeping hands on mouse/keyboard (P16). All participant comments are included in the supplemental material.

6 Discussion

We evaluated MAR against desktop modeling on an abstract shape (T-Abstract) and an everyday object (T-Truck). We observed significantly lower positional and rotational view changes with MAR, indicating that participants preferred inspection mode to understand the shape and select mesh elements with less reliance on viewport adjustment in Blender. Participants were free to use any available tool rather than being required to use AR input. Thus, our results suggest that participants preferred using inspection mode to understand the shape rather than navigating in the desktop interface. In Desktop, participants had to change the view and navigate more to obtain an overview of the model, and prior work suggests that repeated viewpoint changes may make it harder to understand how different views relate to one another and can introduce disorientation in 3D space [21, 34]. This may also explain why MAR achieved better SUS and NASA-TLX results. This pattern is also reflected in UEQ results, where MAR achieved higher scores in all subscales, demonstrating that MAR improved not only pragmatic qualities such as efficiency and dependability, but also hedonic qualities such as attractiveness, stimulation, and novelty. Lower positional and rotational view changes with MAR offer a way to reduce the navigation burden that was previously studied by Safe 3D Navigation [21] and ViewCube [34]. Our system may have reduced this navigation burden by moving spatial understanding from repeated desktop view changes into embodied AR inspection.

On the other hand, we found that participants were significantly slower with MAR in T-Abstract, but not in T-Truck. Prior work suggests that switching between interfaces or views can increase task time [12, 68]; however, MAR still produced lower workload and higher usability, indicating that the slower times were not simply due to poor usability or system design. Participants often continued examining the model with MAR before ending the task, sometimes expressing satisfaction with the result. Post Edit Inspection Time was significantly longer with MAR in T-Abstract, indicating that participants used inspection mode to verify the shape. MAR may therefore have helped users with verifying the correctness of what they created, even when it did not always speed up the design process. Furthermore, T-Abstract involved an abstract shape that participants had no prior familiarity with, which may have increased the need for repeated checks. In contrast, T-Truck involved a more familiar object, which may have allowed users to complete the shape faster using their current knowledge. We therefore believe that the combination of a greater need for observation in T-Abstract, the ease of observing the shape in MAR, and participants' desire to ensure that the final shape was satisfactory may have contributed to longer task times with MAR in T-Abstract. Our model grade results also showed very high grades in T-Abstract, showing that participants were successful in completing the shape, although doing so required greater viewpoint change in Desktop.

Moreover, task completion time may not fully capture the value of a 3D modeling system, since prior work suggests that evaluation should also consider output quality [53], creativity-related outcomes [15], and spatial understanding [8]. Our system resulted in comparable model quality in T-Abstract, and significantly higher model quality in T-Truck, while also providing a better user experience and lower workload than Desktop.

We also observed that MAR yielded different results based on the Task. In T-Truck, participants achieved higher model grades with MAR and reported better usability and user experience and lower workload. In T-Abstract, MAR improved usability and user experience and reduced workload, but did not significantly affect the model grade. In terms of SUS, MAR obtained an *A+* in T-Truck, while receiving an *A* in T-Abstract. Moreover, in addition to the Mental, Temporal, and Overall NASA-TLX subscales that were significantly different in T-Abstract, the Performance, Effort, and Frustration ratings were also significant in T-Truck. These findings may also support a higher value of MAR as the spatial reasoning demand of the task increases. This demand was higher in T-Truck, where participants often had difficulty positioning the wheels of the truck. A common pattern was that they positioned the wheels from the same view (see Figure 11(a)), but placed them incorrectly because of the limited depth cues on a 2D screen (see Figure 11(b)). In contrast, they noticed the positioning error more with MAR (see Figure 11(c-d)). The higher model grade with MAR in T-TRUCK is also in line with prior work showing that immersive environments can improve spatial perception when reviewing 3D CAD models, especially as model complexity increases [29]. Our wheel-positioning observations suggest that MAR was especially beneficial when users had to judge spatial fit and depth rather than only execute editing commands. Taken together, the two tasks reveal a task-dependent trade-off: in T-ABSTRACT, AR inspection supported repeated geometric checking of an unfamiliar shape but increased completion time without a clear model-quality gain, whereas in T-TRUCK, AR inspection helped participants detect spatial errors such as wheel misplacement and improve model quality without a significant time cost.

These findings also locate MAR on the spectrum between purely desktop-based and fully immersive modeling: keeping structured editing operations such as extrusion and precise vertex, edge, and face manipulation on the desktop, while leveraging AR for the parts of the workflow that rely on spatial perception, such as inspecting the developing shape, understanding the current viewpoint, and selecting the relevant mesh elements. In this sense, MAR does not replace desktop-based modeling but combines it with immersive interaction where spatial understanding is needed. Prior work pointed to complementary strengths between immersive environments and desktop input [13,48]; our results show how they can be combined in a seamless cross-reality workflow. Recent cross-reality systems follow a similar strategy, extending a display-based workspace with AR rather than replacing it, an approach surveyed under the concept of augmented displays [50]. Unlike prior immersive modeling [36] that relies heavily on direct hand-based editing, MAR uses hand interaction for spatial inspection and interaction rather than replacing the full editing workflow. Therefore, the contribution of MAR is not simply the addition of hand interaction to 3D modeling; instead, leveraging hand interaction where it is needed without replacing high editing capabilities of desktop input.

Because our participant group was novice-heavy, the overall positive results for MAR may indicate potential for lowering entry barriers to 3D modeling. Despite their limited prior 3D modeling experience, participants reported higher SUS and UEQ scores, and lower NASA-TLX ratings with MAR, and they also preferred MAR as the modeling system (18 out of 20 participants). Participants also achieved significantly higher model grades with MAR in T-TRUCK compared to DESKTOP.

Our system, MAR, contributes to not only a new 3D modeling interface, but also a workflow strategy distributing complementary tasks across desktop and AR: precise editing remains on the desktop, while AR supports spatial visualization, spatial inspection, and mesh element selection. The results highlight that immersive technologies do not need to replace established expert tools to be valuable; instead, they can extend them by supporting the parts of the workflow where spatial understanding and embodied inspection may be valuable.

In addition, MAR uses one-way transform synchronization and performs all editing on the desktop. Extending the system with bidirectional transform synchronization and distributed editing across desktop and AR could give users more direct control within the immersive view, and is a promising direction for future cross-reality modeling systems.

7 Limitations & Future Work

In this study, we focused on polygonal modeling with a single material and a uniform gray color. Future work could add material, color, and texture editing toward a more complete system. We also limited the number of vertices to 1000 to maintain stable AR-Blender communication; because our task models stayed under this limit, it did not restrict performance. This reflects a challenge identified in prior hybrid CAD/XR work [27], the trade-off between accurate mesh transfer and XR rendering optimization, and more complex models could support higher vertex counts through further communication optimizations. Future work should also evaluate MAR with expert users and more advanced modeling workflows, such as Boolean operations, retopology, curve and surface creation, and advanced mesh refinement, which may pose different precision and spatial demands than our tasks. We also used a spherical area-based selection technique; future work could explore alternatives offering greater precision or scalability. Our study also evaluated mesh element selection as part of the overall modeling workflow rather than in isolation, since participants were free to select either in Blender or in AR. As a result, our metrics capture the combined effect of cross-reality selection within the task rather than the performance of the AR selection technique on its own. Evaluating AR-based mesh element selection in a controlled, isolated setting is a useful direction for future work. Further limitations include not recording gaze data; so we could not determine how participants distributed attention between the desktop and the AR view, or whether they relied more on one modality for specific tasks. Fatigue may also matter in longer sessions, although our sessions did not require continuous mid-air hand use and NASA-TLX physical demand did not differ significantly between conditions. Finally, passthrough monitor readability and AR-object occlusion can affect the experience, which future designs could address through display-aware placement or anchoring.

8 Conclusion

In this paper, we presented MAR, a cross-reality 3D polygonal modeling system that combines the complementary strengths of both environments: it keeps precise mesh editing on the desktop with keyboard-and-mouse input, while using immersive AR for spatial inspection, viewpoint control, and mesh element selection. We compared MAR with desktop-based modeling through two model replication tasks: an abstract shape and an everyday object. Our results show that MAR reduces positional and angular viewpoint changes during modeling, lowers workload, and improves usability and overall user experience. Furthermore, the effect of MAR on model quality varied by task, suggesting that the system becomes particularly valuable when higher spatial understanding is required. Together with strong participant preference for MAR (18 out of 20), the findings demonstrate its potential to extend conventional desktop-based 3D modeling tools with immersive technologies.

Acknowledgments

We thank all participants for their time and valuable contributions to this study.

References

[1] R. Arora, R. Habib Kazi, T. Grossman, G. Fitzmaurice, and K. Singh. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2018. 2

[2] R. Arora, R. H. Kazi, F. Anderson, T. Grossman, K. Singh, and G. W. Fitzmaurice. Experimental evaluation of sketching on surfaces in vr. In *CHI*, vol. 17, pp. 5643–5654, 2017. 2

[3] E. Aygar, C. Ware, and D. Rogers. The contribution of stereoscopic and motion depth cues to the perception of structures in 3d point clouds. *ACM Transactions on Applied Perception (TAP)*, 15(2):1–13, 2018. 1, 3

[4] O. Barkovska, O. Shulinus, D. Rosinskiy, Y. Lebodkin, and V. Serdechnyi. Research on model rendering performance in blender 3d using massively parallel systems. In *2023 IEEE East-West Design & Test Symposium (EWDTS)*, pp. 1–5. IEEE, 2023. 1

[5] M. D. Barrera Machuca, W. Stuerzlinger, and P. Asente. The effect of spatial ability on immersive 3d drawing. In *Proceedings of the 2019 Conference on Creativity and Cognition*, pp. 173–186, 2019. 4

[6] M. D. Barrera Machuca, W. Stuerzlinger, and P. Asente. Smart3dguides: Making unconstrained immersive 3d drawing more accurate. In *Proceedings of the 25th ACM symposium on virtual reality software and technology*, pp. 1–13, 2019. 4

[7] L. Besançon, A. Ynnerman, D. F. Keefe, L. Yu, and T. Isenberg. The state of the art of spatial interfaces for 3d visualization. In *Computer Graphics Forum*, vol. 40, pp. 293–326. Wiley Online Library, 2021. 2

[8] S. Bhaduri, Q. L. Biddy, J. Bush, A. Suresh, and T. Sumner. 3dnst: A framework towards understanding children's interaction with tinkercad and enhancing spatial thinking skills. In *Proceedings of the 20th annual ACM interaction design and children conference*, pp. 257–267, 2021. 8

[9] Blender Foundation. Blender python api. 3

[10] D. Borzelli, V. Boarini, and A. Casile. A quantitative assessment of the hand kinematic features estimated by the oculus quest 2. *Scientific Reports*, 15(1):8842, 2025. 2

[11] R. Cools, M. Gottsacker, A. Simeone, G. Bruder, G. Welch, and S. Feiner. Towards a desktop-ar prototyping framework: Prototyping cross-reality between desktops and augmented reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 175–182. IEEE, 2022. 2

[12] R. Cools, I. Maerevoet, M. Gottsacker, and A. L. Simeone. Comparison of cross-reality transition techniques between 3d and 2d display spaces in desktop–ar systems. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 2, 8

[13] J. C. Cremer, C. Lausch, J. Peters, and E. Ragan. Empirical study of virtual reality and desktop systems for qualitative editing of 3d meshes: Impacts of expertise and context. In *Proceedings of the 2025 31st ACM Symposium on Virtual Reality Software and Technology*, pp. 1–10, 2025. 1, 2, 9

[14] B. R. De Araùjo, G. Casiez, J. A. Jorge, and M. Hachet. Mockup builder: 3d modeling on and above the surface. *Computers & Graphics*, 37(3):165–178, 2013. 2

[15] K. Dorst and N. Cross. Creativity in the design process: co-evolution of problem–solution. *Design studies*, 22(5):425–437, 2001. 8

[16] T. Drey, J. Gugenheimer, J. Karlbauer, M. Milo, and E. Rukzio. Vrsketchin: Exploring the design space of pen and tablet interaction for 3d sketching in virtual reality. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pp. 1–14, 2020. 2

[17] T. Drey, N. Rixen, J. Karlbauer, and E. Rukzio. Vrcreatin: Taking in-situ pen and tablet interaction beyond ideation to 3d modeling lighting and texturing. In *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia*, pp. 24–35, 2024. 2

[18] B. Ens, M. Cordeil, C. North, T. Dwyer, L. Besançon, A. Prouzeau et al. Immersive analytics 2.0: Spatial and embodied sensemaking. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pp. 1–7, 2022. 2

[19] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner. G* power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods*, 39(2):175–191, 2007. 4

[20] S. M. Feeman, L. B. Wright, and J. L. Salmon. Exploration and evaluation of cad modeling in virtual reality. *Computer-Aided Design and Applications*, 15(6):892–904, 2018. 4

[21] G. Fitzmaurice, J. Matejka, I. Mordatch, A. Khan, and G. Kurtenbach. Safe 3d navigation. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pp. 7–15, 2008. 1, 4, 8

[22] D. F. García-Molina, R. Carmona-Ávila, J. M. Moltalvo-Gil, and M. Á. Rubio-Paramio. Comprehensive design of the 3d printing process for architectural models. a case study: the medieval walled enclosure of priego de cordoba (e1: 200). *Multimedia Tools and Applications*, 84(8):4881–4905, 2025. 1

[23] M. Gemici, M. R. Bashar, W. Stuerzlinger, V. Phadnis, and A. U. Batmaz. Crossing realities has a cost: Evaluating AR second screens for AI-assisted desktop authoring. In *Proceedings of the ACM Symposium on Spatial User Interaction (SUI '26)*. Association for Computing Machinery, 2026. To appear. doi: 10.1145/3822518.3830046 2

[24] M. Gemici, V. Phadnis, and A. U. Batmaz. Before hands disappear: Effect of early warning visual feedback method for hand tracking failures in virtual reality. *Plos one*, 20(6):e0323796, 2025. 2

[25] D. George and P. Mallery. *SPSS® for Windows® step by step: A simple guide and reference.* Allyn & Bacon, 1999. 6

[26] J. F. Hair Jr, W. C. Black, B. J. Babin, and R. E. Anderson. Multivariate data analysis. In *Multivariate data analysis*, pp. 785–785. 2010. 6

[27] J. Harlan, S. Goetz, and S. Wartzack. Use cases for a hybrid augmented reality computer workstation in cad workflows. *Proceedings of the Design Society*, 3:3731–3740, 2023. 2, 9

[28] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, vol. 50, pp. 904–908. Sage publications Sage CA: Los Angeles, CA, 2006. 6

[29] N. Horvat, S. Škec, T. Martinec, F. Lukačević, and M. M. Perišić. Comparing virtual reality and desktop interface for reviewing 3d cad models. In *Proceedings of the design society: international conference on engineering design*, vol. 1, pp. 1923–1932. Cambridge University Press, 2019. 1, 3, 9

[30] H. H. Huang, H. Pfister, and Y. Yang. Is embodied interaction beneficial? a study on navigating network visualizations. *Information Visualization*, 22(3):169–185, 2023. 2

[31] C. L. Hughes and W. Karwowski. Evaluating interaction design and user experience in augmented reality: a systematic review. *Frontiers in Virtual Reality*, 6:1710161, 2025. 2

[32] 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. 2

[33] R. James, A. Bezerianos, and O. Chapuis. Evaluating the extension of wall displays with ar for collaborative work. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pp. 1–17, 2023. 2

[34] A. Khan, I. Mordatch, G. Fitzmaurice, J. Matejka, and G. Kurtenbach. Viewcube: a 3d orientation indicator and controller. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pp. 17–25, 2008. 1, 4, 8

[35] R. Kijima and T. Ojika. Transition between virtual environment and workstation environment with projective head mounted display. In *Proceedings of IEEE 1997 Annual International Symposium on Virtual Reality*, pp. 130–137. IEEE, 1997. 3

[36] K. Y. Lam, L.-H. Lee, and P. Hui. 3deformr: freehand 3d model editing in virtual environments considering head movements on mobile headsets. In *Proceedings of the 13th ACM Multimedia Systems Conference*, pp. 52–61, 2022. 2, 9

[37] R. Langner, M. Satkowski, W. Büschel, and R. Dachselt. Marvis: Combining mobile devices and augmented reality for visual data analysis. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2021. 2

[38] J. R. Lewis. Measuring perceived usability: The csuq, sus, and umux. *International Journal of Human–Computer Interaction*, 34(12):1148–1156, 2018. 7, 8

[39] J. R. Lewis. The system usability scale: Past, present, and future. *International Journal of Human–Computer Interaction*, 34(7):577–590, 2018. 6

[40] I. Lochhead, N. Hedley, A. Çöltekin, and B. Fisher. The immersive mental rotations test: Evaluating spatial ability in virtual reality. *Frontiers in Virtual Reality*, 3:820237, 2022. 2

[41] M. N. Lystbæk, T. Mikkelsen, R. Krisztandl, E. J. Gonzalez, M. Gonzalez-Franco, H. Gellersen et al. Hands-on, hands-off: gaze-assisted bimanual 3d interaction. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–12, 2024. 4

[42] K. Marriott, J. Chen, M. Hlawatsch, T. Itoh, M. A. Nacenta, G. Reina et al. Immersive analytics: Time to reconsider the value of 3d for information visualisation. In *Immersive analytics*, pp. 25–55. Springer, 2018. 1, 3

[43] J. Mayer, S. Auer, J. Friedl, and C. Anthes. Volumetric data interaction in ar and vr using a handheld touch-sensitive device. 2021. 3

[44] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies*, vol. 2351, pp. 282–292. Spie, 1995. 2

[45] A. Millette and M. J. McGuffin. Dualcad: integrating augmented reality with a desktop gui and smartphone interaction. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 21–26. IEEE, 2016. 2

[46] V. G. Nanjundaswamy, A. Kulkarni, Z. Chen, P. Jaiswal, S. S. S, A. Verma et al. Intuitive 3d computer-aided design (cad) system with multimodal interfaces. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 55850, p. V02AT02A037. American Society of Mechanical Engineers, 2013. 4

[47] L. Quan. *Image-based modeling*. Springer Science & Business Media, 2010. 1

[48] H.-R. Rantamaa, J. Kangas, S. K. Kumar, H. Mehtonen, J. Järnstedt, and R. Raisamo. Comparison of a vr stylus with a controller, hand tracking, and a mouse for object manipulation and medical marking tasks in virtual reality. *Applied Sciences*, 13(4):2251, 2023. 2, 9

[49] T. Rau, T. Isenberg, A. Koehn, M. Sedlmair, and B. Lee. Traversing dual realities: Investigating techniques for transitioning 3d objects between desktop and augmented reality environments. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2025. 3

[50] M. Raza, V. Ketsoi, and D. Reilly. A scoping survey on augmented display systems. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pp. 1–26, 2026. 9

[51] P. Reipschläger and R. Dachselt. Designar: Immersive 3d-modeling combining augmented reality with interactive displays. In *Proceedings*

of the 2019 ACM international conference on interactive surfaces and spaces, pp. 29–41, 2019. 1, 2

[52] P. Reipschlager, T. Flemisch, and R. Dachselt. Personal augmented reality for information visualization on large interactive displays. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1182–1192, 2020. 2

[53] S. Rhys Cox, H. Bøjer Djernæs, and N. van Berkel. Beyond productivity: Rethinking the impact of creativity support tools. In *Proceedings of the 2025 Conference on Creativity and Cognition*, pp. 735–749, 2025. 8

[54] S. A. Salinas, K. Grolinger, M.-E. LeBel, and A. L. Trejos. Evaluating mixed reality technology for tracking hand motion for shoulder rehabilitation assessment. In *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pp. 1049–1056. IEEE, 2024. 2

[55] J. Sauro and J. R. Lewis. *Quantifying the user experience: Practical statistics for user research*. Morgan Kaufmann, 2016. 7, 8

[56] M. Schrepp, A. Hinderks, and J. Thomaschewski. Construction of a benchmark for the user experience questionnaire (ueq). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4):40–44, 2017. 6

[57] M. Sereno, S. Gosset, L. Besançon, and T. Isenberg. Hybrid touch/tangible spatial selection in augmented reality. In *Computer Graphics Forum*, vol. 41, pp. 403–415. Wiley Online Library, 2022. 3

[58] M. Sereno, X. Wang, L. Besançon, M. J. McGuffin, and T. Isenberg. Collaborative work in augmented reality: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 28(6):2530–2549, 2022. doi: 10.1109/TVCG.2020.3032761 3

[59] J. Sun, W. Stuerzlinger, and B. E. Riecke. Comparing input methods and cursors for 3d positioning with head-mounted displays. In *Proceedings of the 15th ACM Symposium on Applied Perception*, pp. 1–8, 2018. 2

[60] E. Ülker, A. Turanboy, İ. Uysal, and G. R. Arik. Development of a web-based 3d visualization tool that can be used in geoscience studies. *Mining Revue/Revista Minelor*, 31(2), 2025. 1

[61] J. Van Brummelen, L. P. Urwin, O. J. Johnston, M. Sayed, and G. Brostow. Don't look now: audio/haptic guidance for 3d scanning of landmarks. In *Proceedings of the 2024 CHI conference on human factors in computing systems*, pp. 1–20, 2024. 6

[62] K. J. Varnum. *Beyond reality: Augmented, virtual, and mixed reality in the library*. American Library Association, 2019. 1

[63] N. Wang, S.-W. Chan, D. Aigner, O. Addam, C. Anthes, and F. Maurer. Serious cross reality-using cr to enhance analytics workflow. In *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 44–49. IEEE, 2023. 2

[64] N. Wang and F. Maurer. A design space for single-user cross-reality applications. In *Proceedings of the 2022 international conference on advanced visual interfaces*, pp. 1–5, 2022. 2

[65] X. Wang, L. Besançon, D. Rousseau, M. Sereno, M. Ammi, and T. Isenberg. Towards an understanding of augmented reality extensions for existing 3d data analysis tools. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020. 2

[66] C. Weichel, M. Lau, D. Kim, N. Villar, and H. W. Gellersen. Mixfab: a mixed-reality environment for personal fabrication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3855–3864, 2014. 2

[67] B. Wells, N. Wang, S. Wiebe, C. B. Josephson, F. Sinaei, and F. Maurer. Neurocr: Integrating mixed reality with desktop medical imaging software for pre-surgical evaluations of epilepsy patients. In *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 1236–1237. IEEE, 2025. 3

[68] J. Wentzel, F. Anderson, G. Fitzmaurice, T. Grossman, and D. Vogel. Switchspace: Understanding context-aware peeking between vr and desktop interfaces. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2024. 2, 8

[69] K. Yatani. Effect sizes and power analysis in hci. In *Modern statistical methods for hci*, pp. 87–110. Springer, 2016. 4

[70] Q. Zhou, G. Fitzmaurice, and F. Anderson. In-depth mouse: Integrating desktop mouse into virtual reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2022. 2