SATOR: Seamless 3D Teleportation to Both Ground and Mid-Air Targets

Daniel Rupp Matthias Wölwer Torsten W. Kuhlen Daniel Zielasko Tim Weissker

Fig. 1: *SATOR* consists of two modes: an air mode (a-b) where a ray cursor is used and a ground mode with either a ray (c) or an arc (d-e). a) Distance control: move the cursor along the ray by either vertically tilting the controller or rate-based using the joystick. b) Target location is automatically clipped when the cursor is moved on top of standable geometry. When the user teleports on top of standable geometry, the ground mode is activated, which is either: c) a straight ray or d) a parabolic arc. e) The arc can also be activated in the air to select targets on the current elevation level.

Abstract— Traditional target-selection-based teleportation depends on the intersection of a (curved) ray with the scene's geometry, which limits navigation to points on the ground, restricting users' navigational freedom. While previous techniques exist that permit mid-air target selection, they are not optimal for transitioning between air and ground navigation, leading to inaccurate or lengthy interaction sequences. In this paper, we introduce *SATOR*, a new 3D teleportation technique designed to enable efficient and accurate navigation to both ground and mid-air targets by combining and enhancing previous approaches. Informed by the literature, we implemented four different parametrizations of our technique and compared them to a previously published technique that also enables both ground and mid-air target selection. Our user study with 30 participants indicates that *SATOR* is more efficient, accurate, and easier to use than the baseline. As a result, *SATOR* effectively helps users get an overview of the environment, observe features at different heights, or maneuver quickly around larger obstacles.

Index Terms—Virtual Reality, 3D User Interfaces, 3D Navigation, Head-Mounted Display, Teleportation, Flying, Mid-Air Navigation.

1 Introduction

Target-selection-based teleportation, often simply referred to as *teleportation*, has emerged as a standard method for navigation in Virtual Reality (VR), allowing users to egocentrically select a target location before being automatically repositioned there [42]. Because the relocation is instantaneous, teleportation is more efficient than continuous methods, especially over large distances [10,45], and prevents cybersickness by avoiding contradictory visual cues of self-motion, making it a preferred technique over continuous navigation [8,10,42,43,45,60]. To specify the target location, traditional teleportation techniques rely on the intersection of a (parabolic) ray with the scene's geometry, therefore limiting navigation to points on the ground plane or on top of smaller adjacent platforms. While ground-based navigation is usually sufficient for tasks that replicate real-world scenarios, unrestricted three-dimensional navigation allows users to teleport to mid-air targets in order to gain an overview of the environment, observe environmental features at different heights, or maneuver quickly around larger obstacles. While previous work exists that extends the target specification process to allow a fast and accurate selection of mid-air targets, many papers only look at the process of mid-air target selection in isolation [15,33,34,37]. However, in virtual environments, we usually have geometry where users can stand on, and we therefore expect a mix in navigation sequences, requiring frequent transitions between the ground and mid-air or to stay on the same elevation level for multiple teleports in order to inspect or interact with an object. The target specification process in prior work also often relies on a straight ray, which limits performance for ground movement compared to a parabolic arc [16,19,48].

To approach these limitations, the work of Weissker et al. introduced techniques that enable users to change the elevation of a previously selected target point on the ground plane [58]. This approach preserves ground movement via a parabolic arc, but also enables users to stay on the same elevation level while in the air, which reduces visual jitter when teleporting in small intervals. However, the process of specifying the point in the air becomes slower as it is not possible to specify the mid-air point directly. Instead, users have to aim at a point on

- *Daniel Rupp, Torsten W. Kuhlen, and Tim Weissker are with the Visual Computing Institute at RWTH Aachen University. E-mail: daniel.rupp@rwth-aachen.de, kuhlen@vr.rwth-aachen.de, me@tim-weissker.de*
- *Matthias Wölwer is with the Human-Computer Interaction Group at Trier University. E-mail: woelwerm@uni-trier.de*
- *Daniel Zielasko is with the Technical University of Denmark. E-mail: daniel.zielasko@rwth-aachen.de.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

their current elevation level, which is then moved into the air, further decreasing accuracy, as the point's accuracy in the air depends on the accuracy of the point on the current elevation level.

In this paper, we present a novel technique called **SATOR**: **S**eamless **A**erial **T**eleportation **O**ptimized **R**ay-cursor that combines the accurate and fast target specification approach of straight ray target selection, while also considering a seamless transition to ground teleportation or to targets on the same elevation level. In an empirical user study with 30 participants, we compare our technique to the best-performing technique from Weissker et al. [58] to answer the **research question** of what the most efficient, accurate, and comfortable way is to specify mid-air target locations, without compromising ground or same elevation level maneuverability. Our contributions can be summarized as follows:

- *SATOR*, an integrated 3D teleportation technique for seamless navigation to both ground and mid-air targets
- The derivation and implementation of four *SATOR* parametrizations, informed by related work
- Scientific evidence from a user study with 30 participants comparing *SATOR* to a baseline from Weissker et al. [58], showing that *SATOR* is more efficient, accurate, and easier to use, and a comparison of the four parametrizations of *SATOR* with recommendations for future use

2 Related Work

Navigation is one of the most fundamental tasks in VR applications, consisting of the motor component *travel* and the cognitive component *wayfinding* [32]. While physical walking is the most straightforward and intuitive way of travel, showing several benefits such as high accuracy and presence [47,53,54,61], its capabilities are often restricted by the limited size of the physical tracking space.

In steering-based approaches, users move in a continuous motion throughout the virtual environment by specifying an input vector whose length is usually determined by a joystick or touchpad, and the direction is determined by a tracked device like the controller or the VR headset [32]. The direction vector is often projected on the ground plane to mimic real-world walking behavior. In these cases, moving to other elevations, such as inside buildings, can be done using steps [50], ramps [14], ladders [29, 50], or elevators [56]. This, however, does not allow users to navigate freely in 3D space to get a better overview of the environment, observe objects from different elevations, or move more efficiently over obstacles. To do so, one can remove the constraint and have the input vector point in any direction, which is often referred to as flying [1, 38, 46].

An alternative to steering is target-selection-based teleportation, often referred to as teleportation or Point & Teleport [6]. It has emerged as a standard for VR locomotion and is one of the highest researched locomotion techniques identified in the literature review of Martinez et al. [36]. Compared to steering, it is more efficient, especially for traversing larger distances [10, 45], and it has been shown to be less prone to cybersickness than steering-based approaches [8, 10, 42, 43, 45, 60] although teleportation compromises on spatial awareness [3, 9, 26, 30, 40]. In the classification of travel metaphors presented by LaViola et al., teleportation falls under the category of selection-based travel metaphors [32]. Although different input metaphors are used for teleportation (e.g., [5, 13, 57]), the most widespread approach uses a tracked device from which a ray is cast into the environment, and users select a target location from an egocentric perspective [6, 42]. The intersection point of that (parabolic) ray with the geometry in the scene then determines the target location. If users want to select a target location in mid-air, other approaches are necessary as the target location cannot be determined by the intersection point of the ray. For more information about teleportation-based travel, refer to Prithul et al.'s mini-review of different teleportation techniques [42].

To have users benefit from the full potential of unconstrained 3D navigation, our work in this paper explores approaches to realize efficient and user-friendly mid-air target selection with seamless ground transitions. The following sections summarize related work that addresses 3D *object* selection (Section 2.1) and 3D *target* selection (Section 2.2) to formalize the research gap our work addresses.

2.1 3D Object Selection

Not all 3D object selection techniques can be adapted for selection-based travel, as the number of selectable objects is usually restricted and informs the selection process. Therefore, most object selection techniques rely on a ray and some form of intersection mechanism with a set of pre-defined selectable objects (e.g. [12, 35, 39, 52]). In the following, we will focus on 3D selection techniques that do not rely directly on ray-object intersections as these are more versatile and, therefore, potentially applicable to target selection for teleportation as well. For a comprehensive overview of selection techniques, refer to Argelaguet et al. [2].

The *Go-Go* technique from Poupyrev et al. does not rely directly on object intersections, and could therefore be adapted for mid-air target selection [41]. Instead, the technique scales the reach of the user's arm in a non-linear fashion, allowing users to grasp objects at greater distances. The drawbacks, however, are that it can become quite tiring, especially since navigation is a fundamental task that is carried out frequently. Range is also limited by some factor of the user's reach, and due to the non-linear scaling of the user's arm, it can be difficult to maintain accuracy, especially when selecting distant targets. We therefore did not consider it for our implementation.

While the *DepthRay* technique from Vanacken et al. [55] or the *RayCursor* technique from Baloup et al. [4] use a ray, it does not rely on intersecting with an object. In both techniques, the user can manually move a cursor along the ray to specify the depth of the selection point. The closest object to that point is then selected. This approach is beneficial for selecting objects in densely populated environments. Selecting a target location during teleportation presents a similar problem, as freely selecting a target location can be seen as selecting infinite, densely packed objects. Therefore, this type of cursor lends itself well to specifying target locations during teleportation, which is why we decided to focus on this approach.

2.2 3D Target Selection

Most prior research papers that explicitly focused on mid-air target selection for subsequent teleportation relied on the ray cursor principle, similar to the object selection technique from Baloup et al. discussed in the last section [4]. The proposed approach involves a straight ray that is emitted in the controller's forward direction, and some form of distance control mechanism that lets users move a cursor along the ray.

Drogemüller et al. suggested a ray cursor where the cursor's distance is controlled using the controller's touchpad [15]. In their *SkyPort* technique, Matviienko et al. revisit this idea and also suggest a second variant of the ray cursor, where the cursor moves along a parabolic arc emanating into the sky [37]. They also suggest that the target location should be specified in relation to the controller's position. Results showed that the straight ray outperformed the parabolic arc in speed and accuracy. However, as the experiment was conducted in mid-air only "to avoid confusion" it did not consider any form of air-to-ground or ground-to-air transitions [37]. To improve this in future work, they suggest that in hybrid spaces, where a transition between air and ground is necessary, a parabolic curve could be activated manually via a button press or automatically based on the controller's horizontal angle.

In the ray cursor implementation from Lim et al. [34], the distance is fixed at 20 *m* if the ray does not intersect with an object, or the intersection point minus a safety margin, so users are not stuck inside a wall or obstacle otherwise. Lee et al. also used a ray cursor and suggested that the distance can be automatically controlled by the system based on a signed distance field [33]. Weissker et al. used a different approach [58]. Their implementation considers transitions between ground and air and is based on a parabolic arc rather than a straight ray. Instead of directly specifying the target location, they suggest first selecting the target point on the current elevation level before moving it up or down vertically. The addition of the mid-air selection step as an additional degree of freedom of the target specification process was implemented in a *simultaneous* (while aiming, the target point can be moved vertically with the touchpad), *two-step* (the target point is fixed on the current elevation first and tilting the controller up or down moves the point vertically accordingly), or *separate* (users teleport horizontally first,

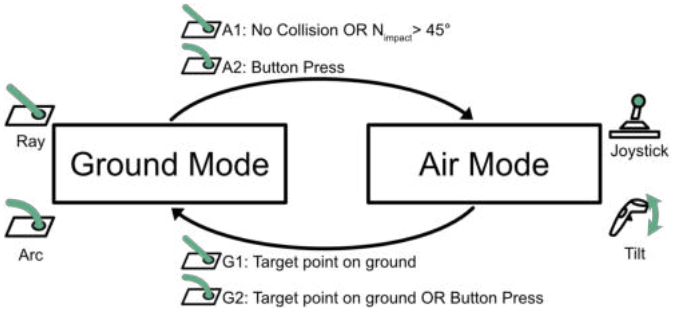

Fig. 2: The *SATOR* technique follows a state machine with two main modes: a ground and an air mode. In ground mode, we implemented and compared a (straight) ray and a (parabolic) arc for ground navigation. In air mode, a ray cursor is used, where the depth of the cursor along the ray can be moved by using the joystick or by tilting the controller. The transitions between both states depend on the tool used for ground navigation.

and then move vertically afterwards separately) approach. In the end, while both the two-step and simultaneous approaches performed well, the two-step approach was faster, had less task load, and was preferred by more users.

Another approach was presented by Kim et al. that uses both hands [28]. Their implementations use two straight rays, with the target location determined by where both rays intersect. They compared their approach to the simultaneous technique from Weissker et al. [58]. While the results show advantages of their crossing rays approach, the evaluation was, however, focused on target selection only, and users were not actually teleported. It is therefore not clear how well the technique works for (rapid) sequential teleportations.

2.3 Summary

Most existing mid-air selection techniques use a form of ray cursor, in which the distance is controlled by the controller's touchpad [15, 37] or is automatically determined by the system [33, 34]. However, none of them consider navigational requirements like staying at the same elevation level, ground-based movement, or transitions between ground and air or vice versa. Therefore, in the case of ground-based movement, they rely on the user to freely select appropriate positions in the vicinity of the ground with the ray cursor. The approach from Weissker et al. works similarly to traditional Point & Teleport, where a parabolic arc and the intersection point of a projectile following that arc with the scene's geometry determines the target location, which can then be moved up and down in addition [58]. While this technique does consider staying on the same elevation level and transitions between ground and air, first specifying the location on the current elevation level before moving the point vertically reduces efficiency and accuracy, as the mid-air target cannot be specified directly, and the accuracy of the point in the air depends on the accuracy of the point on the current elevation level. To combine the advantages of all previously presented interaction paradigms, our work in this paper proposes a combination of the fast and efficient approach of the ray-cursor method with the consideration for ground maneuverability, similar to the technique proposed by Weissker et al. [58]. Combining these aspects, our technique differs from related work by using a direct mid-air target selection approach, while still considering same level maneuverability and enabling seamless air-to-ground and ground-to-air transitions.

3 System Design

Based on the drawbacks and suggestions from previous research, we implemented our *SATOR* technique with the goal of improving efficiency, user experience, and navigational flexibility. Two allow seamless ground transitions, our proposed technique has two main modes, a *ground* and an *air mode*. These two main states are visualized in Figure 2, along with their transitions. In this section, we will explain both modes and present four parametrizations of the *SATOR* technique. Our implementation was designed to work with the Meta Quest 3 controller and therefore assumes a joystick and a continuous trigger button with no physical click. However, the system can be easily adapted to work with other input devices as well.

3.1 Ground Mode

Ground mode is active by default when users enter the virtual environment on the ground level. If the user is in the air and teleports on top of standable geometry (impact normal $< 45°$), the system also automatically switches to the ground mode (see transition G1 and G2 in Figure 2), which is indicated on the user's controller and on the shirt of the preview avatar by a footsteps icon (see Figure 1 c-e). Based on the implementation of common ground-based teleportation techniques in the literature [42], we implemented two common modalities: A (parabolic) arc and a (straight) ray. The behavior on the ground and the transition back into the air differ for each *ground tool* and will be explained in more detail in the following paragraphs.

3.1.1 Ground Tool: Ray

With the ray as *ground tool*, when users press the trigger button, the controller emits a ray in the forward direction, and the intersection point of that ray with the floor determines the target location, similar to Point & Teleport from Bozgeyikli et al. [6]. The ray has a fixed maximal length of 21.52 *m*, which is identical to the reach of the parabolic arc, described in the next section. If the ray does not intersect with geometry on the ground, because it intersects with a wall or points into the air, the system will switch to the air mode (see transition A1 in Figure 2). Releasing the trigger button results in an instantaneous transition to the selected target location. For the ray *ground tool*, the transition from air to ground and vice versa is straightforward and seamless, as in both cases a straight ray is used. However, a straight ray for ground navigation is not ideal, as users cannot easily teleport over obstacles, on top of small buildings, and are less accurate due to jitter. Therefore, as an alternative, we also propose using an arc for ground movement, which is also commonly used for ground-based teleportation [42].

3.1.2 Ground Tool: Arc

With the arc as *ground tool*, when users press the trigger button, the controller emits a (parabolic) arc in the forward direction. The arc visualizes a projectile that is emitted from the controller with a velocity of 14 $\frac{m}{s}$ based on best practices from Rupp et al., allowing selections up to a distance of 21.52 *m* [48]. The intersection point of that projectile with the scene's geometry then determines the target location. While it is trivial for the ray tool when the user wants to move into the air, with the arc, it is not clear, as users cannot simply tilt it up into the air, as the projectile would simply fall down. We tested the proposed approach of Matviienko et al. [37], where the switch would happen automatically once the user tilts the controller above 45°. However, in testing, we found that this limits the usability of the parabolic arc during ground navigation, as it is no longer possible to teleport on top of larger obstacles, which often requires higher tilt angles, so the arc's vertex point is higher. Additionally, it limits the initial specification of mid-air targets, as it would not be possible to have a low angle of ascent. We therefore opted for a simple button press to switch between air and ground mode explicitly (see transition A2 and G2 in Figure 2). This also adds the benefit of switching to ground mode while in the air. Doing so will activate an invisible ground plane on the user's current elevation level, allow them to select targets on the current elevation level (see Figure 1 e), improving visual stability during rapid teleportations.

3.2 Air Mode

In air mode, the basic concept is a ray cursor displayed while the user holds the controller's trigger button. To design the input modality to move the cursor along the ray, we followed the design space introduced by Weissker et al. [58] that suggests three approaches when adding additional degrees of freedom to the target selection process of teleportation: *simultaneous*, *two-step*, and *separate*. While the *separate* approach was the least preferred by participants when specifying height and scale as additional parameters, both *simultaneous* and *two-step*

have been shown to have their own individual benefits [58, 59]. We therefore propose two approaches to control the distance of the cursor along the ray: 1) a *simultaneous* approach where the joystick of the controller is used to move the cursor while aiming, similar to previous techniques (cf. [15, 37]) described in Section 3.2.1; and 2) a *two-step* approach where the user first fixes the direction and tilting the controller up- or downwards moves the cursor along the ray, described in Section 3.2.2. The ray always has a fixed maximum length, based on the reach of the parabolic arc in ground mode, which is 21.52 m. Whenever the ray intersects with an object in the scene that the user cannot stand on (impact normal $> 45°$), the maximal distance is set to the intersection point minus a safety margin of 5% of that distance, so users will not get stuck inside objects after teleportation similar to the ray cursor presented by Lim et al. [34].

During conventional ground-restricted teleportation, the user's feet are teleported to the specified target location, resulting in the user standing on the ground plane. For mid-air teleportation, this would result in the user's feet being positioned higher when the ray is aimed horizontally, making it harder to maintain the same elevation level. To address this, we recommend choosing the target location relative to the user's controller, as also suggested by Matviienko et al. [37]. While aiming, the ray is constantly visualized, together with a preview avatar at the currently selected target location. Following the suggestion of Medeiros et al., when the user is in the air, a small solid platform is shown below the user to mitigate cybersickness, fear of heights, and balance issues [38]. A similar, however, transparent platform to improve visibility is also shown below the preview avatar (see Figure 1 a). The preview avatar also fades out when moved very close to the user, so it does not obstruct the view. If the selected teleportation point is below the controller height on top of standable geometry (intersection normal $< 45°$), the user would be stuck inside the floor. Therefore, the location is clamped accordingly, so the user would "land" on the ground after teleporting, triggering a transition to the ground mode (see transition G1 and G2 in Figure 2). This is also visually indicated, as the ray is still visualized in the controller's forward direction, but the preview avatar remains positioned above the ground (see Figure 1 b). For simplicity, the position of the cursor is always initialized 0.5 m in front of the user, allowing for small adjustments without requiring to move the cursor. When the user releases the trigger button, they are instantly teleported to the specified location. If the new target location is on the ground, the behavior would automatically transition to ground mode (see transition G1 and G2 in Figure 2). Additionally, the current target specification process can be canceled at any time by pressing the grab button on the controller. The following two sections provide more details about the two different distance control mechanisms used to move the cursor along the ray while in the air.

3.2.1 Simultaneous Distance Control: Joystick

This approach is similar to the ray cursor implemented by Matviienko et al. [37]. While the trigger is held, users can use the joystick of the VR controller to move the cursor along the ray. The speed of the cursor is scaled according to the length of the ray, which in our case results in an initial speed of $0.47\frac{m}{s}$, that linearly increases over the course of 2.5 s to $11.62\frac{m}{s}$, based on internal testing. Releasing the stick in between resets the speed again. After releasing the trigger, users are instantly teleported to the specified location.

3.2.2 Two-Step Distance Control: Tilt

In tilt mode, the target specification follows a two-step process. First, the user has to aim in the intended direction by starting to press the trigger button. When the user holds the trigger button fully pressed, a slight vibration is played, and the ray geometry remains fixed based on the direction when the trigger was pressed. Subsequently, the cursor can be moved along the ray by tilting the controller up to move it away from the user or down to move it towards the user. Based on internal testing, we limited rotation to $\pm 40°$ as lower values would reduce accuracy and higher values would become uncomfortable. If the user then releases the trigger, they will be instantly teleported to the specified location.

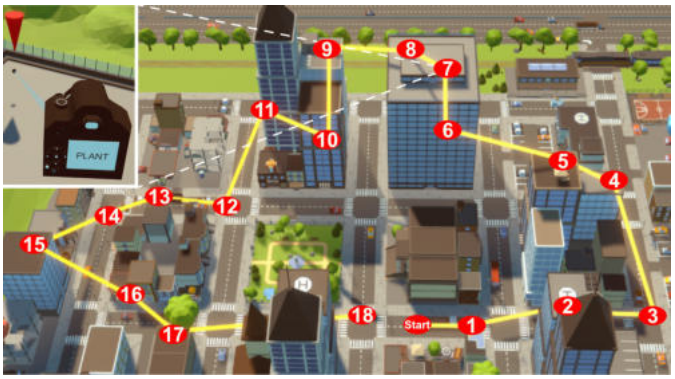

Fig. 3: Route layout used during the user study, where each number indicates a waypoint that users need to traverse to using all five conditions. The top left shows waypoint seven up close. A waypoint is represented by a camera, and the camera's display shows a word that users need to read out loud during the study. A red cone on top of the next camera, and a ray from the current to the next camera, help users to find the next waypoint quickly.

3.3 Summary

The *SATOR* technique considers target specification on the ground and in the air by offering a dedicated mode for each aspect. In ground mode we implemented two tools: 1) a (straight) ray that allows for a seamless transition from ground to air without changing the visualization, as in the air users also have a straight ray; 2) a (parabolic) arc, allowing easier and more accurate ground maneuverability and allowing same level maneuverability for visual stability. Users can transition to the air by either pointing in the air in case of the ray *ground tool* or by pressing a dedicated button in case of the arc *ground tool*. In air mode, users have a ray cursor to be able to efficiently specify a target location. To move the cursor along the ray, we implemented two *distance control* mechanisms, tilt and joystick, based on the design space suggested in previous work. To transition back to ground mode, users simply teleport on top of standable geometry or activate it manually, in the case of the arc *ground tool*. Our parametrizations of *SATOR* therefore follows a two by two design: *ground tool* (ray, arc) $\times$ *distance control* (joystick, tilt), the combination of which leads to four conditions that we compare to a baseline in our user study.

4 User Study

To get scientific insights about the efficiency and usability of our technique, we conducted a within-subjects user study, comparing our *SATOR* technique to a baseline.

4.1 Task Design

The virtual environment consisted of a small city with a variety of buildings of various sizes to foster vertical navigation. Similar to the task presented by Weissker et al. [58], we decided to present users with a primed search task [51], where the route consisted of a series of waypoints on different elevation levels, where each waypoint was visualized by a camera. Users were required to navigate accurately on two levels: 1) they needed to navigate close to the cameras to be able to touch each camera's display; 2) upon touching the display, a word would show up on the display, and they needed to be able to read it out loud for the experimenter. Doing so successfully would then trigger the next camera. The next camera to go to was always marked by a big red cone floating on top of it, together with a line from the current camera to the next one (see Figure 3 top left). The route stayed the same in all conditions to ensure comparability. In total, there were 18 cameras, labeled from **1** to **18** in the order they appeared along the route. Each camera was positioned to fall into a specific category: $\mathbb{C} =$ (from [ground, air], to [ground, air], elevation [same, higher, lower]), denoting the properties of the camera placement along the route. For example, the camera $\mathbb{C}_{\mathbf{4}\ air,ground,higher}$ denotes the fourth camera in

From	To	Elevation		
		Same	Higher	Lower
Ground	Ground	**1**, **13**, **14**	**5**, **15**	**9**, **18**
Ground	Air	**10**	**2**, **6**	**16**
Air	Ground	**17**	**4**	**8**, **12**
Air	Air	**3**	**7**	**11**

Table 1: Categories of cameras placed during our route-following task, where each number corresponds to a camera in a certain category, named after the order they appeared along the route.

the route that the user teleports to from a lower position in the air to a higher position on the ground (e.g., landing on top of a large building). These categories are used to test the different design considerations of our technique by fostering multiple ground-to-air and air-to-ground transitions as well as navigating to waypoints on the same elevation level. In total, there were 12 $(2 \times 2 \times 3)$ types of cameras and each type was used at least once. Table 1 shows the distribution of the cameras in each category, and Figure 3 gives an overview of the camera placements in the scene. The whole route had a total length of 436 *m*.

4.2 Independent Variables

In total, we have one independent variable with five levels or conditions: the four variants of our *SATOR* technique and the two-step technique from Weissker et al. [58], referred to as *Elevator* as a baseline.

4.2.1 SATOR

Based on the different *ground tool* and *distance control* mechanisms, for our *SATOR* technique we implemented four variations, as described in Section 3: arc/joystick, arc/tilt, ray/joystick, ray/tilt.

4.2.2 Baseline Technique: Elevator

As a baseline technique, we adapted the two-step technique from Weissker et al. [58], based on the source code that was made available to us by the authors. We will refer to it as *Elevator* from now on. We made only slight changes to the original implementation, such as adjusting the colors of the preview avatar and selection ray to align with *SATOR*. For the same reason, we also changed the visuals of the platform below the user when in the air. The original technique features a preview window that automatically appears when the target location is at a certain threshold above or below the user. While we believe this is a helpful addition, we also suggest adding it to our ray cursor techniques in a final version. However, in our user study, we excluded it because it could potentially be a confounding factor, as we decided to focus on an isolated comparison of target specification methods without introducing confounding variables by different forms of auxiliary information for spatial orientation.

4.3 Dependent Variables

During the task, we logged the time it takes users to navigate to all the cameras. The timer started after the first camera was activated and ended after the last one was activated. As a form of efficiency measure, we introduce the *overshoot* value, which is calculated as the difference between the length of the route and the total teleportation distance. A value close to zero would therefore mean that users teleported very efficiently towards the cameras, without any unnecessary teleportations. If users were less accurate, they might teleport too far and overshoot, requiring them to backtrack, which increases the overall teleportation distance. In addition to these aggregated measures, we logged the number of teleports and the average specification time per teleport, measured without idle time, from the start of the selection process until they are teleported.

We also used a set of questionnaires to quantify established measures of usability and user experience. The raw NASA task load score (TLX) was used to quantify task load on a scale from 0 to 100 [21, 22]. Furthermore, the User Experience Questionnaire (UEQ) [31] was used, which results in six scores between -3 and 3 representing the perceived *attractiveness*, *perspicuity*, *efficiency*, *dependability*, *stimulation*, and *novelty* of the technique. For the conditions that use the arc as a *ground tool* users needed to press an extra button to activate it, which could disrupt the flow. We therefore also used the flow short scale that features ten flow items that are evaluated on a 7-point Likert scale and summed up to give an overall flow score between 10 and 70 [44].

4.4 Hypotheses

While we generally follow an exploratory approach in our analysis, we have also formulated a few general hypotheses to test formally. Our hypotheses and evaluation will be two-fold. First, we will not consider the individual variants of *SATOR* and compare the technique as a whole to the baseline *Elevator* technique. By averaging the results of the four *SATOR* variants, we want to show that the direct target specification approach of *SATOR*, which is present in all variants, is more efficient than the two-step approach implemented in *Elevator*. We therefore formulated the following hypothesis for the analysis of *SATOR* vs. *Elevator*:

H_1: *SATOR* will be more efficient than *Elevator*.

In the second part of our analysis, we will then focus on the four *SATOR* variants to figure out the best parametrization and formulated the following hypotheses:

H_2: Conditions with *ray* as *ground tool* will have a higher flow rating than *arc*.
H_3: Conditions with *tilt* as *Depth Control* will be faster than *joystick*.
H_4: Conditions with *arc* as *ground tool* will be faster when teleporting to targets on the same elevation level.

For the ray *ground tool*, no manual mode switch will be necessary; therefore, we hypothesize that the flow rating will be higher. Because *tilt* uses a direct mapping of the controller's tilt angle to *distance control*, we expect it to be faster as a *distance control* mechanism than *joystick*, which uses a rate-based approach. Since the *arc* is a standard technique for ground navigation, it provides greater accuracy and flexibility to jump over obstacles. We, therefore, hypothesized benefits for ground navigation, as well as in general when navigating to targets at the same elevation level, with the possibility of activating the arc in the air as well.

4.5 Apparatus

The application was developed using the Unity game engine (Version 6000.0.40). For data logging during the user study, the *Unity Experiment Framework* was used [7]. The application ran on a desktop PC displayed on a Meta Quest 3 headset connected via the official link cable with the included controllers. An aftermarket head strap was added to allow users to adjust the headset better. The physical space was approximately 3 $m \times 3$ m. However, participants were instructed to remain stationary in that space. The application ran at the standard resolution of the headset ($2,064 \times 2,208$ pixels per eye) at a fixed refresh rate of 72 frames per second. To improve the reproducibility, we made the application publicly available [49].

4.6 Participants

In order to determine the required number of participants, we performed a power analysis to determine the minimum sample size using the tool G*Power [17]. We wanted to do two types of tests: 1) a 2x2 factorial repeated measures ANOVA targeting medium effect sizes, an alpha value of 5%, and a power of 80%, which required 24 participants; 2) a t-test with the same parameters, requiring 27 participants. More information is given in Section 5. For the user study, we therefore recruited 30 participants (17 male, 13 female), who were mainly computer science

students between 21 and 33 years old ($M = 25.83, \sigma = 2.82$) from two different universities. 28 participants had prior VR experience, out of which nine do not use VR regularly. Out of the remaining 19 participants who use VR regularly, six did so for less than one year, six for one to two years, four for two to four years, and three for more than four years. We also asked the 28 participants who had already used VR before to rate their skill level with using VR technology (beginner: 8, advanced: 4, competent: 9, proficient: 4, expert: 3) and what interface they have used before (teleportation: 26, steering: 16, object-selection via straight ray: 26, it was possible to select multiple options).

4.7 Procedure

The user study was conducted at two different universities following the same procedure to recruit a larger and more diverse participant pool. Upon arriving at the laboratory, participants read the task description, filled out a consent form, and drew a random number to which the logged data and the questionnaires were linked. Afterward, they put on the headset, entered their unique number, and selected their handedness. Then a short video was played that introduced the route-following task and how to find and activate the cameras. If they had no further questions, they were prompted by the application to press a button to start with the first condition.

For each condition, a short video was played that explained the controls and specifics on how to use the current technique. Then they had to navigate along a short practice route consisting of 10 cameras. Once they were done, they had the option to experiment freely with the technique or continue to the main task. In the main task, they had to follow a route of 18 cameras. Once the last camera was activated a prompt was displayed telling them to put down the headset to fill out a questionnaire on the PC. On the PC, users then filled out the Fast Motion Sickness Scale (FMS) [27], which lets users self-assess their sickness on a scale from 0 to 20, the flow short scale [44], the raw TLX [22], and the UEQ [31] questionnaire. This process was repeated for all five techniques. Starting with the second condition, an additional question was added to the beginning of the questionnaire, asking users if they understood the differences between the current and the previous technique. It was not possible to proceed when selecting no; in that case, the experimenter shortly explained the differences.

In order to mitigate learning effects, the order of the conditions was counterbalanced using a *Balanced Latin Square* design, resulting in ten different orders, with each order being completed by three participants. After the user had completed all conditions, a final questionnaire had to be filled out, where participants had to rank the different conditions from best to worst and had the option to leave any additional feedback in written form. The study concluded with the user filling out a demographics' questionnaire. The user study took between 50 and 80 minutes to complete. All participants successfully completed the user study and were rewarded with around 10 dollars worth of local currency as compensation. The study procedure, including data acquisition and handling, was approved by the ethics board of Trier University (no. 10/2023).

5 Results and Discussion

As mentioned in Section 4.4, we first compare *SATOR* as a whole against the baseline *Elevator* technique in Section 5.1 before looking at the four *SATOR* variants in Section 5.3, with separate discussions in between. For a general overview of our results, we calculated the mean and standard deviations of the dependent variables over all participants. To visualize our results, we created plots using Python and Matplotlib [24]. We used the statistics tool Jamovi [25] for further statistical analysis of our dependent variables with a significance level of 5%. We assumed a normal sampling distribution of our data based on our sample size of $N = 30$ in the context of the central limit theorem [18, pp. 170–172]. Self reported **motion sickness** was generally very low during all conditions, based on the results of the FMS questionnaire, with a mean value of 1.25 and a standard deviation of 1.95 on a scale of 0 to 20.

				Elevator		*SATOR*	
UEQ Subscores	t(29)	p-value	Cohen's d	M	σ	M	σ
Attractiveness	1.16	0.127	0.212	1.15	1.50	1.43	0.78
Perspicuity	1.21	0.118	0.221	1.43	1.33	1.69	0.72
Efficiency	2.77	**0.005**	0.507	0.84	1.33	1.46	0.75
Dependability	0.94	0.178	0.171	1.26	1.40	1.48	0.61
Stimulation	0.90	0.188	0.164	1.28	1.39	1.51	0.78
Novelty	-1.64	0.944	-0.300	1.19	1.21	0.80	0.90

Table 2: UEQ results of the t-test comparing *Elevator* and *SATOR* split by each subscale. Bold values indicate significant differences.

Route Time [s]	M	251	214	230	201	211
	σ	90.4	46.8	49.1	58.4	60.9
Overshoot [m]	M	71.1	32.1	26.0	29.8	27.6
	σ	64.2	27.7	19.8	31.1	27.3
Number of Teleports	M	64.0	51.4	57.4	52.2	55.3
	σ	24.7	14.3	13.4	16.1	18.5
Average Specification Time [s]	M	2.09	2.34	1.94	2.16	2.01
	σ	1.16	0.89	0.89	0.79	1.06
Task Load	M	23.1	19.8	21.7	15.7	19.0
	σ	17.3	14.2	13.2	12.3	14.7
Flow	M	55.0	57.1	54.6	58.4	56.9
	σ	9.17	6.39	8.28	6.52	8.46

Table 3: Mean and standard deviations for route time, overshoot (the difference between the total teleportation distance and the length of the route), number of teleports, specification time, task load and flow.

5.1 Results: SATOR vs. Elevator

To compare the dependent parameters between the averaged results of the *SATOR* conditions and *Elevator*, we used paired t-tests and calculated Cohen's d effect sizes with the threshold values of 0.2, 0.5, and 0.8 representing small, medium, and large effects [11, pp. 24–26].

Figure 4 shows the distribution of the route time, overshoot, number of teleports, average specification time, task load, and flow scores for each condition. The average **route time** was significantly higher in *Elevator* ($M = 251\ s,\ \sigma = 90.4$) than in *SATOR* $M = 214\ s,\ \sigma = 44.9$ with $t(29) = 3.57,\ p < 0.001,\ d = 0.651$ (medium effect).

The average teleport **overshoot** was significantly higher in *Elevator* ($M = 71.1\ m,\ \sigma = 64.2$) than in *SATOR* ($M = 28.8\ m,\ \sigma = 19.4$) with $t(29) = 3.78,\ p < 0.001,\ d = 0.690$ (medium effect).

The average **number of teleports** was significantly higher in *Elevator* ($M = 64.0,\ \sigma = 24.7$) than in *SATOR* ($M = 54.1,\ \sigma = 13.4$) with $t(29) = 3.68,\ p < 0.001,\ d = 0.671$ (medium effect).

For the **average specification time**, there was no significant difference between *Elevator* ($M = 2.23\ s,\ \sigma = 1.16$) and *SATOR* ($M = 2.26\ s,\ \sigma = 0.81$) with $t(29) = 0.213,\ p < 0.584,\ d = 0.039$.

The average **task load** was significantly higher in *Elevator* ($M = 23.1,\ \sigma = 17.3$) than in *SATOR* ($M = 19.0,\ \sigma = 11.9$) with $t(29) = 2.05,\ p = 0.025,\ d = 0.375$ (small effect).

Results for **user experience** can be seen in Table 2. The results show that *SATOR* has a significantly better efficiency rating (medium effect) than *Elevator*.

Figure 5 shows the results of the **preference rating**. Thirteen participants ranked *Elevator* as the worst technique, while half ranked it between second and fourth best. Only two participants ranked *Elevator* as best.

5.2 Discussion: SATOR vs. Elevator

Our results show that users were able to navigate along the route significantly faster with *SATOR* than with *Elevator* (medium effect). While *Elevator* showed similar specification times, the number of teleports was significantly higher (medium effect), contributing to the increase

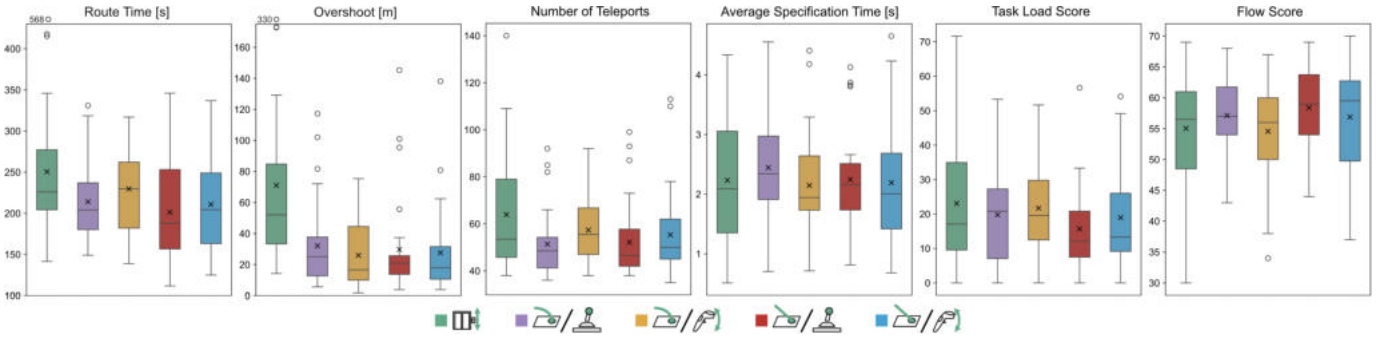

Fig. 4: Boxplots for route time, overshoot (the difference between the total teleportation distance and the length of the route), number of teleports, specification time, task load, and flow. Note that there is an outlier for route time and overshoot that is shown above the box.

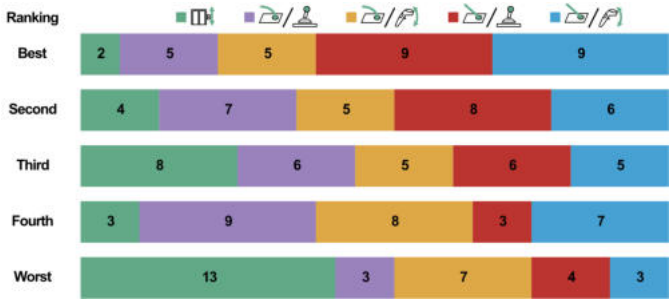

Fig. 5: Preference rating done by users at the end of the study. Numbers indicate how often a technique was chosen for each rank.

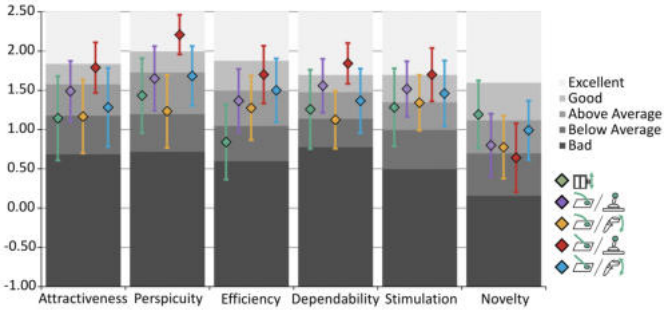

Fig. 6: Benchmark scores of the individual items of the User Experience Questionnaire with confidence intervals (confidence level 95%). Diamonds indicate mean values for each condition.

in route time. The faster speed did, however, not come at the cost of accuracy, as the teleport overshoot was significantly lower for *SATOR* compared to *Elevator* (medium effect). With *SATOR* being quicker and less prone to overshoots, therefore providing better accuracy, we can confirm H_1, stating that *SATOR*, in general, is more efficient than *Elevator*.

Figure 6 shows the results of the UEQ for all conditions individually. We can see that both *Elevator* and *SATOR* perform above average in almost all subscales, with at least one *SATOR* variant outperforming *Elevator* across all scales except *novelty*, where *Elevator* is rated best. We did not find any significant differences, except for the *efficiency* score, where all *SATOR* variants performed better. This is further underlined by the written comments made by participants, where eight participants noted that with *Elevator*, they had trouble estimating the target point in the air based on the ground point, also subjectively confirming H_1.

Regarding task load, with *SATOR* users perceived a significantly lower task load than with *Elevator* (small effect). Both scores can be considered as low when compared to established meta-analyses. In particular, the mean task load of both techniques is considerably smaller than the mean of the 72 VR conditions surveyed by Hertzum ($M = 41$, $\sigma = 15$) [23] and situated in the smallest decile of observed scores in the more general meta-analysis of Grier ($P_{10\%} = 26.08$) [20].

Overall, the results of our first analysis show that both techniques have a low task load and high usability ratings, with *SATOR* offering higher efficiency, while *Elevator* was less preferred in subjective user ratings. In our remaining analysis, we look at *SATOR* in more detail by comparing its different parametrizations with respect to our dependent variables.

5.3 Results: SATOR Variants

To analyze the four *SATOR* parameterizations, we calculated mean and standard deviations for our dependent variables (see Table 3) and created boxplots to visualize our results (see Figure 4). The results of the UEQ are visualized in Figure 6. We also added the results of the *Elevator* condition for comparison. Based on the design space of our technique, we performed a 2x2 factorial repeated-measures ANOVA to test the effect of *ground tool* (ray, arc) and *distance control* (joystick, tilt) on our dependent variables. For all ANOVAs, we also computed the effect size η_p^2 and interpreted it using the threshold values of 0.01, 0.06, and 0.14 representing small, medium, and large effects, respectively [11, pp. 285–287].

5.3.1 Overall Analysis

For the **route time**, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29) = 0.231$, $p = 0.634$, $\eta_p^2 = 0.008$). A significant main effect was shown for *ground tool* ($F(1,29) = 4.995$, $p = 0.033$, $\eta_p^2 = 0.147$ (large effect)) with ray ($M = 206\,s$) being faster than arc ($M = 222\,s$). A significant main effect was also shown for *distance control* ($F(1,29) = 4.908$, $p = 0.035$, $\eta_p^2 = 0.145$ (large effect)) with joystick ($M = 208\,s$) being faster than tilt ($M = 220\,s$).

For the teleport **overshoot**, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29) = 0.527$, $p = 0.474$, $\eta_p^2 = 0.018$). No significant main effects were found for *ground tool* ($F(1,29) = 0.006$, $p = 0.939$, $\eta_p^2 < 0.001$) and *distance control* ($F(1,29) = 1.030$, $p = 0.319$, $\eta_p^2 = 0.034$).

For **number of teleports**, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29) = 0.963$, $p = 0.335$, $\eta_p^2 = 0.032$). No significant main effect was found for *ground tool* ($F(1,29) = 0.114$, $p = 0.738$, $\eta_p^2 = 0.004$). A significant main effect was shown for *distance control* ($F(1,29) = 6.092$, $p = 0.020$, $\eta_p^2 = 0.174$ (large effect)) with joystick ($M = 51.8$) requiring less teleports than tilt ($M = 56.4$).

For **average specification time**, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29) = 1.900$, $p = 0.179$, $\eta_p^2 = 0.061$). No significant main effects were found for *ground tool* ($F(1,29) = 1.524$, $p = 0.227$, $\eta_p^2 = 0.050$) and *distance*

control ($F(1,29)= 2.711,\ p = 0.110,\ \eta_p^2 = 0.085$).

For the **task load**, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29)= 0.315,\ p = 0.579,\ \eta_p^2 = 0.011$). A significant main effect was shown for *ground tool* ($F(1,29)= 6.839,\ p = 0.014,\ \eta_p^2 = 0.191$ (large effect)) with ray ($M = 17.33$) having a lower task load than arc ($M = 20.76$). No significant differences were shown for *distance control* ($F(1,29)= 2.419,\ p = 0.131,\ \eta_p^2 = 0.077$).

For the results of the **flow** questionnaire, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29)= 0.260,\ p = 0.614,\ \eta_p^2 = 0.009$). No significant main effects were found for *ground tool* ($F(1,29)= 3.485,\ p = 0.072,\ \eta_p^2 = 0.107$) and *distance control* ($F(1,29)= 3.128,\ p = 0.087,\ \eta_p^2 = 0.097$).

For **user experience**, due to the large number of subscores and resulting tests, we only report significant results. The ANOVA revealed no significant interaction or main effects for *attractiveness*, *efficiency*, *stimulation*, and *novelty*.

For *perspicuity*, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29)= 0.130,\ p = 0.721,\ \eta_p^2 = 0.004$). A significant main effect was shown for *ground tool* ($F(1,29)= 10.160,\ p = 0.003,\ \eta_p^2 = 0.259$ (large effect)) with ray ($M = 1.95$) showing higher perspicuity than arc ($M = 1.44$). A significant main effect was also shown for *distance control* ($F(1,29)= 5.982,\ p = 0.021,\ \eta_p^2 = 0.171$ (large effect)) with joystick ($M = 1.93$) showing higher perspicuity than tilt ($M = 1.46$).

For *dependability*, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29)= 0.024,\ p = 0.878,\ \eta_p^2 = 0.001$). No significant main effect was shown for *ground tool* ($F(1,29)= 3.405,\ p = 0.075,\ \eta_p^2 = 0.105$). A significant main effect was found for *distance control* ($F(1,29)= 5.320,\ p = 0.028,\ \eta_p^2 = 0.155$ (large effect)) with joystick ($M = 1.70$) showing higher dependability than tilt ($M = 1.25$).

Figure 5 shows the results of the **preference rating**. In general, conditions with ray as *ground tool* were preferred over arc, with a small advantage for ray/joystick over ray/tilt.

5.3.2 Analysis of Teleports on the Same Elevation

To answer H_4, we analyzed only the same elevation teleportations (cameras 1, 3, 10, 13, 14, and 17, see Table 1). We also performed a 2x2 factorial repeated-measures ANOVA. For **route time**, the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29)= 0.128,\ p = 0.723,\ \eta_p^2 = 0.004$). No significant main effects were found for *ground tool* ($F(1,29)= 3.720,\ p = 0.064,\ \eta_p^2 = 0.114$) and *distance control* ($F(1,29)= 3.015,\ p = 0.093,\ \eta_p^2 = 0.094$).

We also looked at teleport **overshoot** and the ANOVA revealed no significant interaction between *ground tool* and *distance control* ($F(1,29)= 0.045,\ p = 0.833,\ \eta_p^2 = 0.002$). A significant main effect was shown for *ground tool* ($F(1,29)= 4.305,\ p = 0.047,\ \eta_p^2 = 0.129$ (medium effect)) with arc ($M = 8.92\ m$) showing less overshoot than ray ($M = 14.01\ m$). No significant main effect was found for *distance control* ($F(1,29)= 0.037,\ p = 0.848,\ \eta_p^2 = 0.001$).

5.4 Discussion: SATOR Variants

5.4.1 Ground Tool: Arc vs. Ray

Regarding H_2, we expected that the additional button to switch between ground and air mode would result in a lower flow rating. Results of the flow short scale, however, did not confirm any significant differences in that regard, and we therefore cannot confirm H_2 based on our data. In the comment box at the end of the user study, participants noted that they liked the arc conditions for allowing them to move over obstacles (2x), and their ability to stay on the same elevation level (6x), while others noted that the extra button or mode switch was difficult to use (8x). In addition, in the preference rating, only ten users rated arc conditions as best, in contrast to 18 for conditions with ray as *ground tool*. Quantitative results show a similar picture, where ray performs typically better than arc.

ANOVA results showed that arc conditions were slower (large effect), had a higher task load (large effect), and a worse perspicuity score than ray conditions (large effect). For overshoot, no significant differences were found when looking at the whole route. However, if we look at the small sub-analysis only considering waypoints where users had to stay on the same elevation level, our results showed that arc had less overshoot when navigating to targets on the same elevation level compared to ray (medium effect). We also expected arc conditions to be faster in this case (H_4); however, the results did not show significant effects for route time, and we therefore cannot confirm H_4. In the end, based on the more seamless transition, as no mode switch is required, and the advantages in objective results and subjective preference ratings, we recommend the use of ray as a *ground tool* for our *SATOR* technique. However, the manual mode switch to an arc could still be added as an optional alternative, allowing users to stay on the same elevation and potentially providing more accuracy when navigating to targets on the same elevation level.

5.4.2 Distance Control: Tilt vs. Joystick

When comparing both *distance control* mechanisms, for route time we expected tilt to be faster than joystick and therefore have a lower route time (H_3). The results, however, show that the opposite is true, while there were no significant differences for the average specification time, tilt had a higher overall route time (large effect) and required more teleports (large effect) than the joystick. We therefore cannot confirm H_3. In addition, for user experience, our results showed that joystick had a better perspicuity (large effect) and dependability score than tilt (large effect). Other than that, no significant differences were shown between the two. In the preference rating, there is also no clear winner as ray/joystick and ray/tilt were both ranked nine times as the best technique, although ray/joystick was mentioned slightly more in second and third place. If we look at user comments, participants also seemed to be divided. While joystick was mentioned as being slow (4x) and uncomfortable to use (3x), users also reported that it was easy to use and straightforward (6x) and felt more precise (2x). The same holds for comments about tilt, which was considered faster (3x), more intuitive (2x), but also less precise (2x) and difficult to use because it requires holding the trigger button before fully pressing it (2x). Although there is a small indication towards the joystick being faster and offering better ratings in two UEQ subscales, subjective opinions are rather divided between the two approaches, and we would therefore suggest offering both options and letting users decide which mode they prefer.

6 Limitations

When designing *SATOR*, we had to choose various parameters, like the maximal tilt angle or the speed of the joystick. The selection of these parameters was based on internal testing, which leaves room for further optimizations, and potential shortcomings might influence the results of the respective features. In addition, we did not consider rotation specification, which is also a common feature of teleportation-based navigation techniques [6, 42]. Because of the larger number of conditions in our user study, we only considered a route-following task, which only offers one perspective on the use of the tested techniques. Although we made sure to cover a vast amount of navigation scenarios by carefully placing the waypoints, a more open task design, like an uninformed search task (as in [58, 59]), would make our results more generalizable. While we considered the *two-step* technique from Weissker et al. as our baseline, as it was the best performing technique, it would have been interesting to also compare *SATOR* against the *simultaneous* or *separate* approaches, which we excluded based on the results reported in their paper [58]. Finally, in our evaluation, we did not consider finer-grained measures like movement trajectories or behavioral data.

7 Conclusion and Future Work

In this paper, we proposed a novel teleportation-based navigation technique called *SATOR* that allows the selection of both ground and mid-air targets to facilitate the free exploration of large 3D virtual environments. We implemented four parametrizations of *SATOR* based on previous

research with the goal of providing a fast target specification and a seamless transition between air and ground navigation. In a user study with 30 participants, we compared the four *SATOR* variants against a baseline. Our results showed that *SATOR* in general improves over the baseline in terms of efficiency, usability, and task load. As a result, we suggest offering a variation of *SATOR* when users should be presented with the option to gain an overview of the environment, observe environmental features at different heights, or maneuver quickly around larger obstacles. Based on our results, we recommend using a ray as *ground tool* and either joystick or tilt for *distance control* when in mid-air.

For better comparability and simplicity, we decided that the ray cursor always starts close to the user; in a future version, we could set the initial cursor based on a distance field, similar to the ray cursor approach from [33], minimizing cursor adjustments and therefore improving efficiency even further. As users generally liked the ability to stay on the current elevation level the parabola offers, we could either add the mode switch as an optional addition to the ray *ground tool* or add some form of snapping mechanism to the ray cursor when in the air that makes it easier to point the ray perfectly horizontal. Restricting the length of the ray cursor was mainly done for comparability between arc-based conditions, but could also be increased in a future version. However, it might be hard to see the target location at a certain distance.

In the future, we hope that teleportation-based navigation techniques can become as expressive as steering-based techniques, while still offering higher efficiency and inducing less cybersickness, to keep VR accessible to a large audience.

Acknowledgments

This work has received funding from the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under grant 528403131 (Project “Put me There”).

References

[1] M. Al Zayer, P. MacNeilage, and E. Folmer. Virtual Locomotion: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 26(6):2315–2334, 2020. doi: 10.1109/tvcg.2018.2887379 2

[2] F. Argelaguet and C. Andujar. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, May 2013. doi: 10.1016/j.cag.2012.12.003 2

[3] N. H. Bakker, P. O. Passenier, and P. J. Werkhoven. Effects of Head-Slaved Navigation and the Use of Teleports on Spatial Orientation in Virtual Environments. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 45(1):160–169, 2003. doi: 10.1518/hfes.45.1.160.27234 2

[4] M. Baloup, T. Pietrzak, and G. Casiez. RayCursor: A 3D Pointing Facilitation Technique based on Raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Chi ’19, pp. 1–12. Association for Computing Machinery, New York, NY, USA, May 2019. doi: 10.1145/3290605.3300331 2

[5] B. Bolte, F. Steinicke, and G. Bruder. The jumper metaphor: an effective navigation technique for immersive display setups. In *Proceedings of virtual reality international conference*, vol. 1, 2011. 2

[6] 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*, Chi Play ’16, p. 205–216. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2967934.2968105 2, 3, 8

[7] J. Brookes, M. Warburton, M. Alghadier, M. Mon-Williams, and F. Mushtaq. Studying human behavior with virtual reality: The Unity Experiment Framework. *Behavior Research Methods*, 52(2):455–463, Apr. 2020. doi: 10.3758/s13428-019-01242-0 5

[8] F. Buttussi and L. Chittaro. Locomotion in Place in Virtual Reality: A Comparative Evaluation of Joystick, Teleport, and Leaning. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):125–136, 2021. doi: 10.1109/tvcg.2019.2928304 1, 2

[9] L. A. Cherep, A. F. Lim, J. W. Kelly, D. Acharya, A. Velasco, E. Bustamante, A. G. Ostrander, and S. B. Gilbert. Spatial cognitive implications of teleporting through virtual environments. *Journal of Experimental Psychology: Applied*, 26(3):480–492, 2020. doi: 10.1037/xap0000263 2

[10] C. G. Christou and P. Aristidou. Steering Versus Teleport Locomotion for Head Mounted Displays. In L. T. De Paolis, P. Bourdot, and A. Mongelli, eds., *Augmented Reality, Virtual Reality, and Computer Graphics*, pp. 431–446. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-60928-7_37 1, 2

[11] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Routledge, New York, 2 ed., May 2013. doi: 10.4324/9780203771587 6, 7

[12] G. De Haan, M. Koutek, and F. H. Post. IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *Eurographics Symposiumon Virtual Environments*, pp. 201–209, 2005. doi: 10.2312/egve/ipt_egve2005/201-209 2

[13] G. D. Domenico, F. D. C. Becker, B. B. Schirmer, N. L. P. Berwaldt, G. M. d. Freitas, A. C. Neto, C. T. Pozzer, L. M. Fontoura, and R. C. Nunes. Evaluation of Navigation in Immersive Large-Scale VR Environments Using Minimap-assisted Teleportation. In *Proceedings of the 26th Symposium on Virtual and Augmented Reality*, pp. 270–274, 2024. doi: 10.1145/3691573.3691608 2

[14] J. L. Dorado and P. A. Figueroa. Ramps are better than stairs to reduce cybersickness in applications based on a HMD and a Gamepad. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 47–50, Mar. 2014. doi: 10.1109/3dui.2014.6798841 2

[15] A. Drogemuller, A. Cunningham, J. Walsh, M. Cordeil, W. Ross, and B. Thomas. Evaluating Navigation Techniques for 3D Graph Visualizations in Virtual Reality. In *2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA)*, pp. 1–10, Oct. 2018. doi: 10.1109/bdva.2018.8533895 1, 2, 3, 4

[16] F. Dylong and P. Nickel. Virtual Reality Locomotion Techniques and Mental Workload: An Empirical Investigation of Point-Teleport and Directed-Steering During Virtual Task Performance. *Virtual Reality*, Dec. 2025. doi: 10.1007/s10055-025-01290-2 1

[17] 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. doi: 10.3758/BF03193146 5

[18] A. Field. *Discovering statistics using IBM SPSS statistics*. Sage publications limited, 4th ed., 2013. 6

[19] M. Funk, F. Müller, M. Fendrich, M. Shene, M. Kolvenbach, N. Dobbertin, S. Günther, and M. Mühlhäuser. Assessing the Accuracy of Point & Teleport Locomotion with Orientation Indication for Virtual Reality using Curved Trajectories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Chi ’19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300377 1

[20] R. A. Grier. How High is High? A Meta-Analysis of NASA-TLX Global Workload Scores. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 59(1):1727–1731, Sept. 2015. doi: 10.1177/1541931215591373 7

[21] S. G. Hart. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, Oct. 2006. doi: 10.1177/154193120605000909 5

[22] S. G. Hart and L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*, vol. 52, pp. 139–183. Elsevier, 1988. doi: 10.1016/s0166-4115(08)62386-9 5, 6

[23] M. Hertzum. Reference values and subscale patterns for the task load index (TLX): a meta-analytic review. *Ergonomics*, 64(7):869–878, July 2021. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00140139.2021.1876927. doi: 10.1080/00140139.2021.1876927 7

[24] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/mcse.2007.55 6

[25] jamovi. The jamovi project, 2024. (Version 2.6) [Computer Software]. Retrieved from `https://www.jamovi.org`. 6

[26] J. W. Kelly, A. G. Ostrander, A. F. Lim, L. A. Cherep, and S. B. Gilbert. Teleporting through virtual environments: Effects of path scale and environment scale on spatial updating. *IEEE Transactions on Visualization and Computer Graphics*, 26(5):1841–1850, 2020. Conference Name: IEEE Transactions on Visualization and Computer Graphics. doi: 10.1109/tvcg.2020.2973051 2

[27] B. Keshavarz and H. Hecht. Validating an Efficient Method to Quantify Motion Sickness. *Human Factors*, 53(4):415–426, Aug. 2011. Publisher: SAGE Publications Inc. doi: 10.1177/0018720811403736 6

[28] D. Kim, D. Han, S. Bak, and I. Cho. Crossing Rays: Evaluation of Bimanual Mid-air Selection Techniques in an Immersive Environment. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 329–338, Oct. 2024. Issn: 2473-0726. doi: 10.1109/ismar62088.2024.00047 3

[29] C. Lai, R. P. McMahan, and J. Hall. March-and-Reach: A Realistic Ladder Climbing Technique. In *2015 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 15–18, Mar. 2015. doi: 10.1109/3dui.2015.7131719 2

[30] E. Langbehn, P. Lubos, and F. Steinicke. Evaluation of Locomotion Techniques for Room-Scale VR: Joystick, Teleportation, and Redirected Walking. In *Proceedings of the Virtual Reality International Conference - Laval Virtual*, pp. 1–9. Acm, 2018. doi: 10.1145/3234253.3234291 2

[31] B. Laugwitz, T. Held, and M. Schrepp. Construction and Evaluation of a User Experience Questionnaire. In *4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society (USAB) 2008*, vol. 5298, pp. 63–76, 11 2008. doi: 10.1007/978-3-540-89350-9_6 5, 6

[32] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, 2 ed., Mar. 2017. 2

[33] J.-I. Lee, P. Asente, B. Kim, Y. Kim, and W. Stuerzlinger. Evaluating Automatic Parameter Control Methods for Locomotion in Multiscale Virtual Environments. In *26th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–10. Acm, Virtual Event Canada, Nov. 2020. doi: 10.1145/3385956.3418961 1, 2, 3, 9

[34] D. Lim, S. Shirai, J. Orlosky, P. Ratsamee, Y. Uranishi, and H. Takemura. Evaluation of User Interfaces for Three-Dimensional Locomotion in Virtual Reality. In *Proceedings of the 2022 ACM Symposium on Spatial User Interaction*, Sui '22, pp. 1–9. Association for Computing Machinery, New York, NY, USA, Dec. 2022. doi: 10.1145/3565970.3567693 1, 2, 3, 4

[35] Y. Lu, C. Yu, and Y. Shi. Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 35–43, Mar. 2020. Issn: 2642-5254. doi: 10.1109/vr46266.2020.00021 2

[36] E. S. Martinez, A. S. Wu, and R. P. McMahan. Research Trends in Virtual Reality Locomotion Techniques. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 270–280, 2022. doi: 10.1109/vr51125.2022.00046 2

[37] A. Matviienko, F. Müller, M. Schmitz, M. Fendrich, and M. Mühlhäuser. SkyPort: Investigating 3D Teleportation Methods in Virtual Environments. In *CHI Conference on Human Factors in Computing Systems*, pp. 1–11. Acm, New Orleans LA USA, Apr. 2022. doi: 10.1145/3491102.3501983 1, 2, 3, 4

[38] D. Medeiros, M. Sousa, A. Raposo, and J. Jorge. Magic Carpet: Interaction Fidelity for Flying in VR. *IEEE Transactions on Visualization and Computer Graphics*, 26(9):2793–2804, 2020. doi: 10.1109/tvcg.2019.2905200 2, 4

[39] M. R. Mine. Virtual Environment Interaction Techniques. *UNC Chapel Hill Computer Science Technical Report TR95-018*, 1995. 2

[40] R. Paris, J. Klag, P. Rajan, L. Buck, T. P. McNamara, and B. Bodenheimer. How Video Game Locomotion Methods Affect Navigation in Virtual Environments. In *ACM Symposium on Applied Perception 2019*, pp. 1–7. Acm, 2019. doi: 10.1145/3343036.3343131 2

[41] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, Uist '96, pp. 79–80. Association for Computing Machinery, New York, NY, USA, Nov. 1996. doi: 10.1145/237091.237102 2

[42] A. Prithul, I. B. Adhanom, and E. Folmer. Teleportation in Virtual Reality; A Mini-Review. *Frontiers in Virtual Reality*, 2, 2021. doi: 10.3389/frvir.2021.730792 1, 2, 3, 8

[43] K. Rahimi, C. Banigan, and E. D. Ragan. Scene Transitions and Teleportation in Virtual Reality and the Implications for Spatial Awareness and Sickness. *IEEE Transactions on Visualization and Computer Graphics*, 26(6):2273–2287, 2020. doi: 10.1109/tvcg.2018.2884468 1, 2

[44] F. Rheinberg, R. Vollmeyer, and S. Engeser. FKS - Flow-Kurzskala, Dec. 2019. doi: 10.23668/psycharchives.4488 5, 6

[45] B. E. Riecke and D. Zielasko. Continuous vs. Discontinuous (Teleport) Locomotion in VR: How Implications can Provide both Benefits and Disadvantages. *Proc. of IEEE VR Abstracts and Workshops*, pp. 373–374, 2021. doi: 10.1109/vrw52623.2021.00075 1, 2

[46] 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. doi: 10.1145/147156.147201 2

[47] R. A. Ruddle and S. Lessels. The Benefits of Using a Walking Interface to Navigate Virtual Environments. *ACM Trans. Comput.-Hum. Interact.*, 16(1):5:1–5:18, Apr. 2009. doi: 10.1145/1502800.1502805 2

[48] D. Rupp, T. Weissker, M. Wölwer, T. W. Kuhlen, and D. Zielasko. How Far is Too Far? The Trade-Off Between Selection Distance and Accuracy During Teleportation in Immersive Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2025. doi: 10.1109/tvcg.2025.3632345 1, 3

[49] D. Rupp, M. Wölwer, T. Kuhlen, D. Zielasko, and T. Weissker. [Application] SATOR: Seamless 3D Teleportation to Both Ground and Mid-Air Targets , Jan. 2026. doi: 10.5281/zenodo.18174532 5

[50] M. Slater, M. Usoh, and A. Steed. Steps and ladders in virtual reality. In *Virtual Reality Software and Technology*, pp. 45–54. World Scientific, Aug. 1994. doi: 10.1142/9789814350938_0005 2

[51] R. Stebbins. *Exploratory Research in the Social Sciences*. SAGE Publications, Inc., Thousand Oaks, California, Aug. 2025. doi: 10.4135/9781412984249 4

[52] F. Steinicke, T. Ropinski, and K. Hinrichs. Object Selection in Virtual Environments using an Improved Virtual Pointer Metaphor. In K. Wojciechowski, B. Smolka, H. Palus, R. Kozera, W. Skarbek, and L. Noakes, eds., *Computer Vision and Graphics: International Conference, ICCVG 2004, Warsaw, Poland, September 2004, Proceedings*, pp. 320–326. Springer Netherlands, Dordrecht, 2006. doi: 10.1007/1-4020-4179-9_46 2

[53] E. A. Suma, S. Babu, and L. F. Hodges. Comparison of Travel Techniques in a Complex, Multi-Level 3D Environment. In *2007 IEEE Symposium on 3D User Interfaces*, Mar. 2007. doi: 10.1109/3dui.2007.340788 2

[54] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks. Walking > Walking-in-Place > Flying, in Virtual Environments. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99*, pp. 359–364. ACM Press, Not Known, 1999. doi: 10.1145/311535.311589 2

[55] L. Vanacken, T. Grossman, and K. Coninx. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces*, Mar. 2007. doi: 10.1109/3dui.2007.340783 2

[56] K. Vasylevska and H. Kaufmann. Influence of Vertical Navigation Metaphors on Presence. In *Challenging Presence - Proceedings of 15th International Conference on Presence (ISPR 2014)*, pp. 205–212, 2014. Accepted: 2022-08-04T12:26:35Z. doi: 20.500.12708/55149 2

[57] J. Von Willich, M. Schmitz, F. Müller, D. Schmitt, and M. Mühlhäuser. Podoportation: Foot-based locomotion in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2020. doi: 10.1145/3313831.3376626 2

[58] T. Weissker, P. Bimberg, A. S. Gokhale, T. Kuhlen, and B. Froehlich. Gaining the High Ground: Teleportation to Mid-Air Targets in Immersive Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2467–2477, 2023. doi: 10.1109/tvcg.2023.3247114 1, 2, 3, 4, 5, 8

[59] T. Weissker, M. Franzgrote, and T. Kuhlen. Try This for Size: Multi-Scale Teleportation in Immersive Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 30(5):2298–2308, 2024. doi: 10.1109/tvcg.2024.3372043 4, 8

[60] T. Weißker, A. Kunert, B. Fröhlich, and A. Kulik. Spatial Updating and Simulator Sickness During Steering and Jumping in Immersive Virtual Environments. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 97–104, 2018. doi: 10.1109/vr.2018.8446620 1, 2

[61] C. Zanbaka, B. Lok, S. Babu, A. Ulinski, and L. Hodges. Comparison of Path Visualizations and Cognitive Measures Relative to Travel Technique in a Virtual Environment. *IEEE Transactions on Visualization and Computer Graphics*, 11(6):694–705, Nov. 2005. doi: 10.1109/tvcg.2005.92 2