A Lens-Based Extension of Raycasting for Accurate Selection in Dense 3D Environments

Carole Plasson, Dominique Cunin, Yann Laurillau, Laurence Nigay

A Lens-based Extension of Raycasting for Accurate Selection in Dense 3D Environments

Carole Plasson[1], Dominique Cunin[2], Yann Laurillau[1], and Laurence Nigay[1]

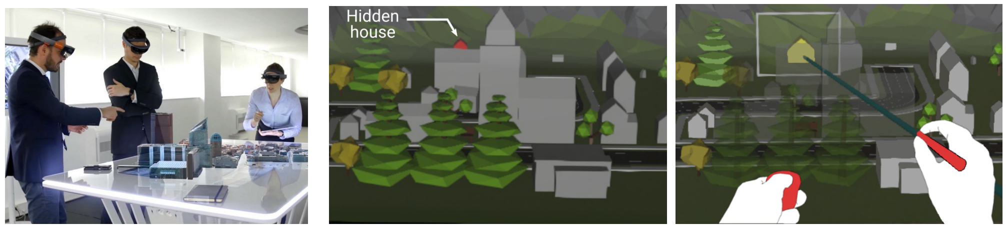
[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
firstname.lastname@univ-grenoble-alpes.fr
[2] Ecole Superieure d'Art et Design Grenoble-Valence, F-26000 Valence, France
dominique.cunin@esad-gv.fr

Abstract. In mixed environments, the selection of distant 3D objects is commonly based on raycasting. To address the limitations of raycasting for selecting small targets in dense environments, we present *RayLens* an extended raycasting technique. *RayLens* is a bimanual selection technique, which combines raycasting with a virtual 2D magnification lens that can be remotely moved in 3D space using the non-dominant hand. We experimentally compared *RayLens* with a standard raycasting technique as well as with *RaySlider* an extension of raycasting based on a target expansion mechanism whose design is akin to *RayLens*. *RayLens* is considerably more accurate and more than 1.3× faster than raycasting for selecting small targets. Furthermore, *RayLens* is more than 1.6× faster than *RaySlider* in dense environments. Qualitatively, *RayLens* is easy-to-learn and the preferred technique making it a good candidate technique for general public usage.

Keywords: Augmented Reality · HMD · Pointing technique · Lens.

1 Introduction

Tabletop Augmented Reality (AR) systems, combining a tangible physical surface and virtual objects, support a variety of applications such as architecture and urban design [14,42,43] (Fig. 1) as well as cultural heritage, visualization systems, and 3D modeling [14,46]. When interacting in tabletop AR, most objects are not directly reachable by hand due to the size of the table [6]. In 3D virtual and augmented environments, the selection of such distant objects is largely based on the raycasting metaphor. Raycasting techniques implement a ray generally held by the users to point at distant objects. However, raycasting techniques suffer from several limitations, especially in dense environments. Due to hand tremors and human pointing accuracy, the selection of small objects can be difficult and longer distances to the objects amplify this difficulty. Besides, using a standard implementation of raycasting, the first object intersecting the ray is selected, making the selection of occluded objects difficult. Dense environments with partially occluded objects are frequent in various applications of tabletop AR as illustrated in the following scenario:



(a) Context of use – Courtesy of Immersion `www.immersion.fr`.

(b) Implemented augmented map of the district of a city. The targeted house is hidden behind another building (left); This house is selected with *RayLens* (right). The pointed house turns yellow.

Fig. 1: Examples of augmented maps.

A property developer presents to two purchasers a district of a city where several houses are for sale. A 3D virtual model of this district is placed on the table (Fig. 1a) and allows the purchasers to visualize the buildings, the streets, and the surrounding shops. The property developer presents the houses for sale in this district by selecting them on the 3D virtual maps. Some houses are distant and therefore small. Moreover, as the building sizes are different, several houses are hidden behind buildings (see the house for sale in red in Fig. 1b).

This scenario illustrates the importance of overcoming the limitations of raycasting. A variety of techniques implement extensions of raycasting to improve its performance in dense environments [4,11,22,27,47]. For occluded objects, bendable rays [38] or rays with adjustable sizes [18,47,56] are two approaches. For precision limitation, pointing facilitation mechanisms initially developed for cursor-based 2D selection such as target expansion mechanisms [20] are applied to extend raycasting [4,33]. However, such pointing facilitation is impacted by the density of the environment and especially the proximity of distractors (i.e. selectable objects which are not the current target) around a targeted object.

An alternative to facilitate object selection is to use zooming techniques [25]. For instance, a magnification effect can be done to enlarge objects located in a specific area. In contrast to target expansion mechanisms, zooming is target-agnostic and does not depend on the immediate surrounding of the targeted object. Zooming is then "especially relevant on dense populations of targets" [12]. Lens-based selection techniques [53] using magnification are mainly developed for 2D environments [1,37,39,40]. Very few AR/VR studies have used them for selection tasks in 3D environments. Also, no study has considered magnification lenses as a pointing facilitation mechanism for accurate 3D selection in dense environments.

We propose a new bimanual technique *RayLens* for accurate 3D virtual object selection which combines a ray held in the dominant hand, with a magnification lens controlled by the other hand. As a standard magnification lens will enlarge the objects of an area but "lead to the same visual output" [53], the occlusion problem remains. To pass through obstacles and reach an occluded area, the lens can be remotely and freely moved in 3D space (see the metaphor of the "eyeball-in-hand" [51,52]). Once the lens is placed in the scene, the lens is occluded by the objects placed in front of it. To remove this occlusion, we apply

a transparency filter on the obstacles placed between the users and the lens. Finally, the selection of an object is performed by pointing to the 2D projection of the magnified object displayed on the distant lens using raycasting.
We compare *RayLens* with a standard raycasting technique as well as with *RaySlider*, a technique extending raycasting with the widely-used target expansion mechanism as a pointing facilitation mechanism. As *RayLens*, *RaySlider* is bimanual: users move the ray using the dominant hand and operate a physical slider to move the cursor along the ray using the non-dominant hand. To facilitate the selection, the cursor always selects the nearest object (such as *Bubble Cursor* [17]). Moreover, the same transparency filter as *RayLens* is applied between the users and the cursor to reduce the occlusion of the area of interest.

In this paper, we first review related work on extended raycasting and lens-based techniques. We then present the design rationale of *RayLens* and also describe *RaySlider* inspired by the design of *RayLens*. We then report an experiment comparing *RayLens*, *RaySlider* and the baseline technique raycasting. We conclude with a discussion of our results and directions for future work.

2 Related Work

We build on previous work on pointing facilitation mechanisms that can extend raycasting as well as on lens-based techniques in 3D environments.

2.1 Extending raycasting: pointing facilitation mechanisms

Raycasting selection techniques perform poorly with small distant objects and when the targeted object is occluded by other objects (i.e. dense environments). To overcome these limitations, a variety of techniques implement extensions of raycasting. In the following, we opposed the target-aware techniques (a priori knowledge on potential targets) from the target-agnostic ones [12].

Several target-aware techniques have been proposed for facilitating 2D selection [12] and have been applied to 3D selections. One intensively studied approach, namely target expansion, is to enlarge the effective size of the target. Guillon et al. [20] classify target expansion techniques according to their "underlying visual feedforward mechanisms" and distinguish *target-based* from *cursor-based* visual feedforward. *RayCursor* [4] extends raycasting with a target expansion mechanism: the closest object from the 3D cursor placed on the ray is highlighted. The position of the cursor along the ray is adjusted by forward-backward displacements on the touchpad of a Vive Controller. This technique provides a *target-based* visual feedforward as it always highlights the closest target from the cursor. Other target expansion techniques provide a *cursor-based* visual feedforward by displaying the activation area of the cursor [17,28]. For instance, for 2D selection *Bubble Cursor* [17] displays a bubble with an adaptive size that encompasses the nearest object. Vanacken et al. [56] proposed *3DBubble*, an extension of *Bubble Cursor* for 3D environments. *3DBubble* uses a 3D cursor moved with the virtual hand metaphor and displays a sphere around the 3D cursor that includes the closest object. All targets within 4 cm of the cursor become semi-transparent to reduce the occlusion of objects close to the cursor.

Lu et al. [33] propose to display a 2D disc instead of a sphere around the cursor. The technique also includes a bendable ray to connect the ray and the closest target (the closest target being the target with the minimal angular distance with the ray). Another technique proposed by Vickers [58] uses a "sensitive cube" to enlarge the activation area of a 3D cursor manipulated by a wand: when an object enters the cube, the cursor automatically jumps to this object.

One alternative to these target-aware approaches includes multiple-step techniques by manual refinement. The first step selects a subset of objects in the 3D scene. One or several following steps are required to disambiguate the selection. We classify these approaches into two groups: (1) the techniques which select several or all objects along the ray during the first step; and (2) the techniques which use a volume instead of the ray to select objects during the first step. Grossman and Balakrishnan proposed extensions of raycasting for occluded object selections on 3D volumetric displays [18]. For instance, *DepthRay* highlights all objects intersected by the ray. A cursor on the ray allows users to select one object among the highlighted objects. Instead of moving a cursor along the ray, another approach, *FlowerRay*, displays the selected objects in a marking menu. Other techniques consider a volume (e.g., a sphere, a cone) instead of a ray. For instance, the technique *SQUAD* [2,27] uses a sphere-casting metaphor to select objects around the raycasting pointer. These objects are then rearranged into a quad-menu: a targeted object is selected by pointing repeatedly the part of the quadrant containing this object. In contrast to the progressive refinement induces by a menu, Cashion et al. [11] proposed to rearrange all selectable objects into a semi-transparent grid displayed in front of the user. With this two-step technique users partially maintain the context of the scene because they can always perceive the scene through the grid.

Our target-agnostic approach also includes two steps by considering magnifying lenses. The first step consists of positioning a magnifying lens and the second step consists of directly selecting the targeted object displayed on the lens by raycasting. In contrast to previous techniques based on space rearrangement methods [2,11,27], magnifying lenses keep the relative positions of the objects (which is essential for some usage contexts such as in augmented maps as presented in the scenario of Section 1). Also, lenses implement the concept of *focus+context* [13] by contrast to a standard zoom technique. Thus, users can simultaneously visualize the global scene and the magnified projection of the zone of interest. Magnifying lenses have been largely used for facilitating 2D selection such as *Magic Lens*, [7] *Shift* [60], *Pointing Lenses* [44], *Widgetslens* [1], *Fisheyes* [21], and *Sigma Lenses* [40]. The following section explores the lens metaphor applied to AR/VR environments.

2.2 Lenses in AR/VR environments

We review studies on lenses in AR/VR environments according to the tasks the lenses support, their shape, and how the users move the lenses. A more detailed review on lenses can be found in the survey of Tominski et al. [53].

Tasks. Applied in AR [30,34,54] and VR [36,45] environments, lenses are used

for several tasks classified by Tominski et al. [53] into 7 tasks: *Select*, *Explore*, *Reconfigure*, *Encode*, *Abstract/Elaborate*, *Filter* and *Connect*. For the last 6 tasks, the lenses can display additional information through them [16,34], remove occlusion [7,29,45,54,55] or magnify a part of the scene [30,41,49,50]. For instance, Bane et al. [5] developed an AR system to give users an X-ray vision (i.e. a virtual lens placed on a wall to display the occluded room behind this wall). Few AR/VR techniques use lenses for facilitating the selection of a 3D object (Task *Select* in [53]). The recent *Slicing-Volume* [35] includes a lens for multi-target selection in an occluded area. Looser et al. [30] also propose to use a lens as a tool for selection in tabletop AR. A raycasting metaphor can be implemented to select "objects targeted by the center of the lens" [31].
In summary, while lenses have been widely used in AR/VR environments for exploration tasks, very few studies have used them for selection tasks. And none of the lens-based techniques have considered magnifying lenses to facilitate the selection by raycasting in dense environments.

Shapes of the lenses. In 3D environments, there are 2 shapes of lenses: volumetric and flat lenses [15,59]. As defined by Schmalstieg et al. [48], the *volumetric lenses* [32,34,36] affect "every object inside the lens region" and the *flat lenses* affect "every object that has a projection falling into the area covered by the magic lens" (Viega et al. called this region *the lens frustum* [59]). The shape of the lens can be predefined but can also be adaptive to the visualized data. The lenses with content-adaptive shapes are not the focus of our work. Volumetric lenses are useful for extracting or filtering a portion of the 3D scene. For example, *Slicing-Volume* is a cube-shape lens [35] to extract and interact with occluded objects in a dense scene.
In our context of pointing facilitation techniques, we use a flat magic lens instead of a volumetric lens, to reduce the difficulty of the pointing task, from a task in 3D space to a task in the 2D space of the lens. For instance, the lens implemented by Looser et al.[30] for the selection of objects in tabletop AR is a flat lens. Moreover in [35] the content of the cube-shape volumetric lens defining a region in the 3D space is then projected on a flat lens for multiple selections.

Movement of the lenses. Lenses can be moved using tangible objects held by the users. For instance, Brown et al. propose a 2D mirror-like prop as a lens [9,10] using Head-Mounted Projective Display technology. In the same way, markers are tracked in [30,34] to display the lens over the markers. In a VR environment, Mota et al. [36] use the controller to move the lens in the 3D scene. Similarly, two controllers held in both hands are used to define the *Slicing-Volume* [35]. Moreover, a physical tablet and stylus are fixed to the controllers for haptic feedback and touch input when selecting objects. Finally, freehand gestures can also be used to activate and move the lens, the lens following the position of the users' hand [45,54]. Several of these approaches make the placement of the lens within an out-of-reach area difficult and tedious. Kluge et al. [26] proposed to move a distant lens by using a proxy placed close to the user's hand. This proxy is manipulated by the virtual hand metaphor and the 3D motion of the proxy is translated into the motion of the distant lens.

By combining design elements (i.e. lens, magnification as opposed to space rearrangement, *focus+context* as opposed to standard zooming and transparency effect) that have been applied in different interaction contexts and for different tasks, we propose *RayLens*, the first lens-based technique extending raycasting for accurate selection in dense 3D environments. The novelty of our technique relies on the synergistic combination of these concepts to overcome the limitations of raycasting. The resulting technique is thus greater than the sum of its constitutive parts. We also compare our target-agnostic extension *RayLens* to a target-aware technique *RaySlider* whose design shares elements from *RayLens* while being inspired by the large amount of research on enlarging the activation area of the 3D cursor.

3 RayLens

RayLens combines two independent components: a ray and a magnification flat lens. A bimanual version of *RayLens*, where users move the ray in one hand and move the lens with the other hand, is intrinsically a bimanual "asymmetric and dependent" [24] technique. Indeed, this design is consistent with Guiard's theory principles [19]: "the two hands have very different roles to play which depend on each other" [24]. As a consequence, we decided to design *RayLens* as a bimanual technique: users move the ray using the dominant hand and move the lens using the other hand.

The virtual ray extends a physical wand held by the users. Users select with the ray the 2D projection of the magnified object displayed on the distant lens. *RayLens* is an extension of the standard raycasting technique and the lens should be used only when needed. Nevertheless in order to compare *RayLens* with the standard raycasting in our experiment (see Section 6), an object can be selected only on the lens: a selection by standard raycasting is then not allowed with *RayLens* in the experiment. In the virtual lens, the 2D projections of objects intersected by the ray are highlighted in yellow. A selection is validated by pressing a button fixed to the physical wand.

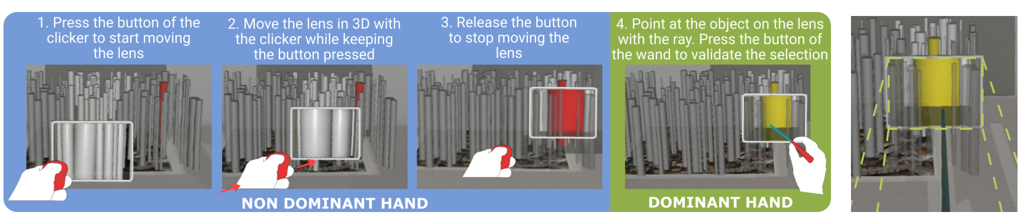


Fig. 2: *RayLens* walkthrough: (1-4) Bimanual selection of a target (left); Occlusion management of the lens (right).

3.1 Virtual lens

The main component of *RayLens* is the virtual lens measuring 22.5 cm×17.5 cm (equivalent to the size of a physical tablet), see Fig. 2. This virtual component acts as a physical tablet, adding an interactive virtual screen in the scene and another point of view via a camera fixed to the lens (like the "eyeball-in-hand" metaphor). Thus, the lens allows the user to add a new point of view on the scene which stays fixed regardless of the user's position (in a collaborative context, a fixed point of view allows all the participants to share the same view on the lens). When the lens does not move, the displayed content is stable making the selection of objects easier. The lens also magnifies the content placed behind it (Fig. 2): it enlarges the visual representation of the objects that fall inside the *lens frustum*. As visual feedback, all objects in this area are brighter than others.

Users can move the lens in 3D by using the clicker of the HoloLens, tracked in the 3D space. Once a click event is detected on the clicker, the device behaves as a 3D remote controller: its 3D motion is translated into the motion of the lens (1:3 ratio[3].) until the button is released. This lets users do clutching to reach more distant positions. These distant positions would be unreachable by a user holding a physical tablet without moving around the table. This motion of the lens is essential for users to be able to see and select occluded objects. To do so users move the lens through obstacles. In the current implementation, the lens cannot be rotated. The usage context implies that users are standing in front of the table and that the virtual objects are at most 20 degrees to the left and right of the users: a rotation of the lens is then unnecessary.

3.2 Occlusion of the lens

When moving the lens through the 3D scene, some distractors will appear in front of it. This causes the lens to be hidden from the users, which makes selections on the virtual lens screen difficult. To reduce this occlusion, we apply a filter on all the objects placed between the lens and the user (in contrast to the transparency effect of [56,57] which is applied only on the objects within 4 cm of the cursor). As the user is fixed, standing in front of the table, we approximate the filtering area as the area in front of the lens, see the yellow box in Fig. 2-right. This filter makes all these objects semi-transparent to minimize the "visual occlusion" of the lens. The semi-transparent objects cannot be selected by the ray anymore (the ray passing through them) to reduce the "physical occlusion" of these distractors. Thanks to this filter, users can easily reach the lens screen with the ray as illustrated in Fig 1b.

In summary, the design rationale of *RayLens* is based on two properties:
Precision. (1) Without voluntary movements of the lens by users (i.e. press on the clicker), the position of the lens is fixed. This stable magnification lens makes the selection of small targets easier. (2) As the selection of an object is done on the lens and not directly on the 3D object, the distance between the users and the target is reduced. The problem of ray stability is consequently also reduced.
Occlusion & Density. The movement of the lens in the 3D scene and the transparency effect enable the selection of occluded targets by passing through ob-

[3] With the 1:3 ratio, a 30 cm motion of the clicker allows users to move the lens from its initial position (30 cm in front of the scene) to the position of the farthest objects (90 cm from the initial position of the lens)

stacles. The lens acts as a "clipping plane" in a similar way as in [15,35]. Users can thus erase the distractors in front of a target to increase its visibility.

4 RaySlider

RaySlider is another extension of the raycasting metaphor that shares design elements from *RayLens* while implementing a target expansion mechanism (Fig. 3). *RaySlider* uses a cursor that moves along the ray to pass through distractors. As *RayLens*, this technique is bimanual: users move the ray with the dominant hand and move the cursor on the ray with the other hand. Considering a cursor attached to the ray, both hands impact the position of the cursor: a 3DOF movement of the cursor is controlled by the dominant hand (i.e. the ray movement) and a 1DOF movement along the ray is controlled by the non-dominant hand. As the dominant hand already controls the 3D position of the cursor, users do not need to additionally move the cursor on the ray when precision is required. Additionally, simultaneous control with both hands could "increase the load on the participant's motor system" and the task could "become more difficult" as observed in [3]. Thus, with such design, the precision phase should be performed by the dominant hand which holds the ray, and the 1DOF movement of the cursor along the ray should be mainly used during the ballistic phase. As a consequence, we design *RaySlider* to quickly place the cursor in the scene.

As *RayLens*, the virtual ray extends a physical wand held by the users. To obtain an easy and fast motion of the cursor along the ray, the cursor is moved thanks to a tangible 7.3 cm long slider held by users. Thanks to muscle memory, participants can quickly place the cursor close to the target. The minimum value of the slider (its bottom position) corresponds to a cursor placed at the tip of the wand. The maximum value of the slider (its top position) corresponds to a cursor at 3 meters away from the tip of the wand. In the setting, the farthest target is at 2 meters from the user. The range of the slider values [0-1023] enables the user to comfortably only use the lower part of the slider (2/3 of its length).

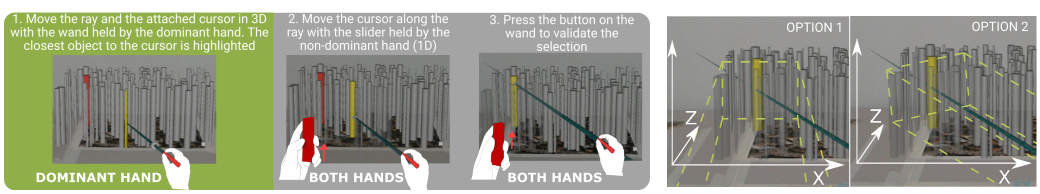


Fig. 3: *RaySlider* walkthrough: (1-3) Bimanual selection of a target (left); Two different occlusion managements of the cursor (right).

To facilitate the selection of small targets, the closest object to the cursor is always highlighted (i.e. a *target-based* visual feedforward [20]) and can be selected by pressing the button fixed to the physical wand. To facilitate the selection of occluded objects, we apply a filter making distractors semi-transparent as done with *RayLens* (see the yellow box in Fig. 3-right) to minimize occlusion around the cursor. In a pilot study, we compare 2 possible positions of the filtering box area, see Fig. 3-right: (1) Option 1. The box is placed between the cursor and the user; (2) Option 2. The box is placed along the ray. In this context of scenes with medium sizes (width=70 cm, depth=60 cm), participants found no difference between the two implementations. For the conducted comparison experiment, we chose the first option which is similar to the filter of *RayLens*.

5 Implementation

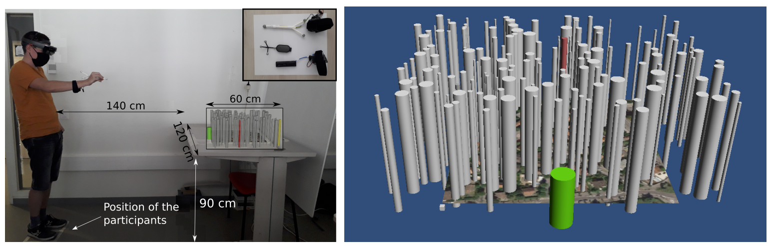


Fig. 4: Setup of the experiment (left) and example of a 3D scene (right) with a start target in green, a goal target in red and distractors in grey.

We use a Microsoft HoloLens 1 to visualize the 3D virtual scene that is created and managed by Unity. This HMD is composed of two HD see-through displays ($FoV = 30 \times 17.5$ degrees) with a frame rate of 60 fps. We also use a tracking system composed of 6 OptiTrack cameras (running at 100 fps). Reflective markers are placed on the wand to track its position and rotation in the 3D space and to be able to extend it with a virtual ray. We fixed a small Arduino button to the wand for the validation of the selection. For *RayLens*, we use the HoloLens clicker to move the virtual lens in the 3D space. We add 4 markers on this clicker to track its 3D position in real-time. For the *RaySlider* technique, users hold a tangible slider. The events of the wand button and the slider are sent via wireless communication (using Arduino radios) to the HoloLens.

To link the OptiTrack system and the Unity Engine, we place an image on the table (which is non-interactive). This image is used as an *Image Target* that Vuforia Engine can detect and track. To optimize the performance of the system, we use Vuforia only at the initialization of the application. Once this image is detected, we create and place a world anchor (see Microsoft MRToolkit) at the bottom left and stop the Vuforia detection. This anchor creates a link between the position of the real objects (e.g., the wand, the clicker, the table) and the position of the virtual objects (e.g., the ray). Fig. 4 presents the general setup.

6 Comparative Study

In this section, we present the experiment that compares *RayLens* with *RaySlider* and the well-known raycasting to study its efficiency (e.g., performances, users feedback). As several studies such as [4,11] that propose improvements of the virtual pointer metaphor, we choose a standard raycasting technique as a baseline. The *RayCasting* technique is implemented as a ray that selects the first intersected object. Without collision, this ray is infinite. As with the two other techniques, users press the button on the wand to validate a selection.

Our hypotheses are:

- *H1.* *RayLens* and *RaySlider* are faster and more accurate than *RayCasting* for selecting small targets.
- *H2.* With a low density of the environment, *RaySlider* is the fastest technique. When the distance between the target and the closest distractors is large, automatically selecting the closest object is very efficient.
- *H3.* With a high density of the environment, *RayLens* is faster and more accurate than *RaySlider*. Indeed target expansion techniques as *RaySlider* are impacted by the density of distractors [8]. And *RayLens* is little impacted by density thanks to the magnification and the transparency effects.
- *H4.* *RayCasting* and *RayLens* require respectively the lowest and the highest workload. With *RayLens*, switching between the 3D view of the scene and the 2D projected view displayed on the lens may require significant extra cognitive efforts from users.

6.1 Tasks

In this experiment, participants have to select 3D cylindrical targets in 3D virtual scenes that visually appear as resting on a table (Fig. 4). Each 3D scene is a set of cylinders, composed of one start target, one goal target, and 120 distractors (Fig. 4). All distractors are gray, the start target is green, and the goal target is red. Pointing at a cylinder changes its color to yellow. The start target is always in front of the participants and of the other cylinders. Before each trial with *RayLens*, the lens is positioned 30 cm in front of the start target. The goal target is randomly positioned but always at a fixed distance of 40 cm from the start target. All the distractors are randomly positioned under the following constraints: they must not intersect nor fully visually occlude the goal target. The diameter size is randomly set between 0.5 cm (small target size) and 3.5 cm (large target size). To ensure similar experimental conditions between the techniques and the participants, we computed a set of unique 3D scenes before the experiment which is shared between all participants and all techniques.

The participants are instructed to stand at 140 cm from the table. The exact location is marked on the ground. The first step of the task is to select the start target. Then, the start target disappears and the participants have to select the goal target. A selection is validated by pressing the button on the wand. A trial ends once the goal target has been successfully selected. The participants are instructed to select the goal target as fast as possible while minimizing the number of selection errors (i.e. number of presses before a successful validation).

6.2 Protocol

Density spacing. To control the density of the environment, we rely on a method inspired by [57]. In the 3D scene, we place four additional distractors in the immediate surrounding of the goal target: two are positioned on a line defined by the goal target and the static position of the participant (one in front of and one behind the goal target), and the other two distractors are positioned on a perpendicular line (one on each side of the goal target). To minimize visual occlusion, we set these four distractors with a small diameter size (0.5 cm) and rotate them around the goal target to easily see the target from the participant's position. As in [57], we call *density spacing* the distance between the goal target and these additional distractors. The other 120 distractors are pseudo-randomly placed around these 4 distractors so as not to visually occlude the target.

Design. We used a within-subject design with the 3 following independent variables: the technique *TECH* (*RayLens*, *RaySlider* and *RayCasting*), the density spacing *DS* (1 cm for high density, and 5 cm for low density) and the target diameter *SIZE* (0.5 cm, 1.5 cm, and 3.5 cm). To observe a possible learning or tiredness effect, we grouped trials into three blocks per technique. Within each block, the 6 combinations of *DS* $\times$ *SIZE* were repeated 3 times in a random order for a total of $3 \times 2 \times 3 = 18$ trials per block. This experimental design results in a total of 54 trials per technique. Before each *TECH* condition, the participants perform a training session to experience each of the *DS* $\times$ *SIZE* conditions. Participants can take a break after each technique. The order of the techniques was counterbalanced across participants using a Latin square design. The experiment lasted approximately 75 minutes per participant.

Participants. We recruited 12 unpaid volunteers (2 females, 10 males), ranging from ages 22 to 38 ($mean = 28.7$, $std = 4.23$). All participants were right-handed. None of them was an expert in augmented reality. We applied COVID-19 preventive sanitary measures for the experiment to take place safely.

Measures. We consider two objective measures (completion time, number of errors), and two subjective measures (workload, user preference). Completion time refers to the elapsed time between the successful selection of the start target and the successful selection of the goal target. The number of errors refers to the number of error presses (i.e. when a user presses the button while no object or a distractor is selected) before the successful selection of the goal target.
As subjective measures, the participants fill in a questionnaire for each *TECH* condition. It combines questions to rate: the perceived workload based on a shortened version of NASA-TLX (called Raw TLX[4]); the perceived performance (from 1 to 7 points); the perceived usefulness of (from 1 to 7 points): (a) the visual feedback representing the field of view of the lens (*RayLens*); and (b) the transparency filtering effect (*RaySlider* and *RayLens*). The participants also fill in a final questionnaire to determine the least mentally and physically demanding techniques, the most successful, the most accurate, and the fastest techniques. Finally, they rank the techniques in order of preference. The experiment ends with an interview. In particular, the participants were asked about the manipulated techniques, their usage, and the positioning of the lens or of the cursor.

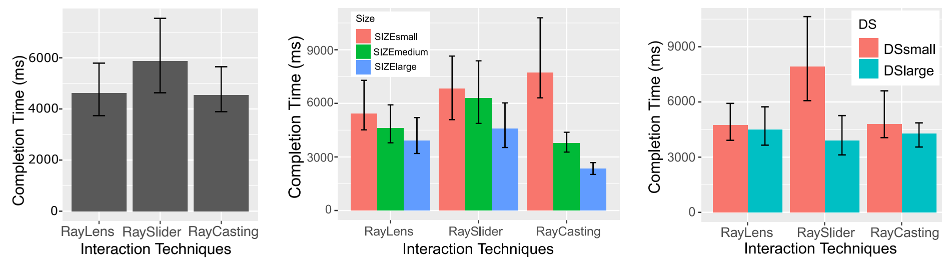
7 Results

As the completion times follow a normal distribution (Shapiro-Wilk test), we use ANOVAs and t-tests with Bonferroni adjustment for pairwise comparisons. Means (m) and 95% confidence intervals (CI) are shown in all graphs. For the

[4] According to the survey [23], NASA-TLX and Raw TLX perform equally well.

non-parametric analysis of the number of errors, we apply the aligned rank transformation (ART) [61] with Bonferroni correction for pairwise comparisons.

7.1 Objective Measures: Completion time



(a) Mean completion times. (b) Completion times according to the size of the target. (c) Completion times according to the density spacing.

Fig. 5: Completion times of the techniques.

We do not find a main effect for *TECH* on completion time ($F_{2,22} = 2.51$, $p = 0.1$), see Fig. 5a. However, we find a main effect for *SIZE* ($F_{2,22} = 103$, $p < 0.0001$, $\eta^2_G = 0.31$), and *DS* ($F_{1,11} = 94.8$, $p < 0.0001$, $\eta^2_G = 0.12$). The average completion times are 5.9s for *RaySlider*, 4.6s for *RayLens* and 4.5s for *RayCasting*. Also, by comparing performances between blocks, we do not find a learning or tiredness effect.

Interaction effect between *TECH* and *SIZE*. We observe significant *TECH* $\times$ *SIZE* interaction effects ($F_{4,44} = 26.3$, $p < 0.0001$, $\eta^2_G = 0.15$), see Fig. 5b. Results show large completion time variations across sizes for *RayCasting* ($RC_{small} = 7.6s$, $RC_{medium} = 3.7s$, $RC_{large} = 2.3s$): the technique is very fast at selecting large targets while it becomes very slow with small targets. Pairwise comparisons show significant differences between small and large targets ($p < 0.0001$) and between medium and large targets ($p = 0.004$). *RaySlider* ($RS_{small} = 6.8s$, $RS_{medium} = 6.4s$, $RS_{large} = 4.5s$) and *RayLens* ($RL_{small} = 5.5s$, $RL_{medium} = 4.6s$, $RL_{large} = 3.9s$) are slightly impacted by the size of the targets. We only find a significant difference between small and large targets ($p = 0.02$ with *RaySlider*, $p = 0.035$ with *RayLens*).

Pairwise comparisons show significant differences between the techniques across sizes. For large targets, *RayCasting* is significantly the fastest selection technique compared to *RayLens* ($p = 0.005$) and *RaySlider* ($p = 0.0003$). For medium targets, *RayCasting* is significantly faster than *RaySlider* ($p = 0.008$). We find no significant difference between *RayCasting* and *RayLens*. For small targets, *RayLens* tends to be the fastest selection technique (1.38$\times$ faster than *RayCasting* and 1.24$\times$ faster than *RaySlider*). However, we only find a significant difference between *RayLens* and *RayCasting* ($p = 0.038$).

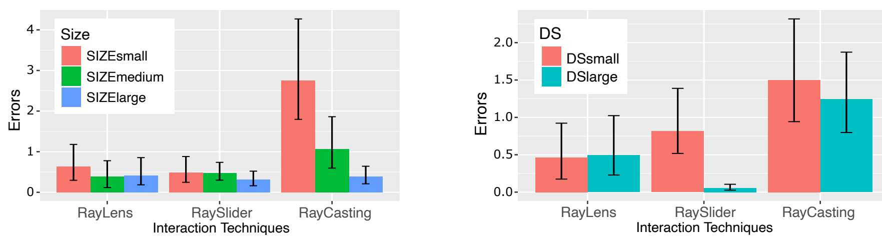
Interaction effect between *TECH* and *DS*. We observe significant *TECH* $\times$ *DS* interaction effects ($F_{2,22} = 35.4$, $p < 0.0001$, $\eta^2_G = 0.13$), see Fig. 5c. Pairwise comparisons show a strong effect of *DS* for *RaySlider* ($p = 0.0008$,

$RS_{DSsmall} = 7.9s$, $RS_{DSlarge} = 3.9s$). We find no significant effect of *DS* for *RayCasting* ($RC_{DSsmall} = 4.8s$, $RC_{DSlarge} = 4.3s$) and *RayLens* ($RL_{DSsmall} = 4.7s$, $RL_{DSlarge} = 4.5s$).

Pairwise comparisons show no significant difference between the techniques for low density (DS=large). For high density (DS=small), *RaySlider* seems to be the slowest technique (*RayLens* and *RayCasting* are 1.6× faster than *RaySlider* on average) but we do not find a significant difference between *RaySlider* and *RayLens* ($p = 0.06$), and between *RaySlider* and *RayCasting* ($p = 0.08$).

To summarize, *RayCasting* is strongly impacted by *SIZE*, *RaySlider* is strongly impacted by *DS*, and *RayLens* is relatively independent of *SIZE* and *DS*.

7.2 Objective Measures: Number of errors



(a) Numbers of errors according to the size of the target.

(b) Numbers of errors according to the density spacing.

Fig. 6: Numbers of errors of the techniques.

The number of errors is the number of presses before successful validation of the target. We find a main effect for *TECH* ($F_{2,187} = 46.12$, $p < 0.0001$), *SIZE* ($F_{2,187} = 33.51$, $p < 0.0001$) and *DS* ($F_{1,187} = 14.61$, $p = 0.0002$) on the number of errors. The average number of errors is 0.43 for *RaySlider*, 0.48 for *RayLens* and 1.35 for *RayCasting*. Pairwise comparisons show that *RayCasting* is significantly less accurate than *RaySlider* (3.1×, $p = 0.004$) and *RayLens* (2.8×, $p = 0.001$). We found no significant difference between *RayLens* and *RaySlider*.

Interaction effect between *TECH* and *SIZE*. We observe significant *TECH* × *SIZE* interaction effects ($F_{4,187} = 23.71$, $p < 0.0001$), see Fig. 6a. Results show large errors variations across sizes for *RayCasting* ($RC_{small} = 2.7$, $RC_{medium} = 1.05$, $RC_{large} = 0.38$): the number of errors with small targets is 7× higher on average than with large targets. Pairwise comparisons show significant differences between small and large targets ($p = 0.0002$), between small and medium targets ($p = 0.016$), and also between medium and large targets ($p = 0.05$). We find no significant difference between sizes for *RayLens* ($RL_{small} = 0.63$, $RL_{medium} = 0.39$, $RL_{large} = 0.40$) and *RaySlider* ($RS_{small} = 0.49$, $RS_{medium} = 0.47$, $RS_{large} = 0.31$).

For small targets, pairwise comparisons show that the number of errors with *RayCasting* is significantly higher than with *RayLens* ($p = 0.001$) and with *RaySlider* ($p = 0.0006$). For medium targets, *RayLens* is significantly more accurate than *RayCasting* ($p = 0.04$). For medium targets, we find no significant difference between *RaySlider* and the other techniques. Finally, for large targets, we find no significant difference between the techniques.

Interaction effect between *TECH* and *DS*. We observe significant *TECH* $\times$ *DS* interaction effects ($F_{2,187} = 11.39$, $p < 0.0001$), see Fig. 6b. Pairwise comparisons show a highly significant difference between densities for *RaySlider* ($RS_{DSsmall} = 0.82$, $RS_{DSlarge} = 0.05$, $p < 0.0001$). We find no significant difference between densities for *RayLens* ($RL_{DSsmall} = 0.47$, $RL_{DSlarge} = 0.49$) and for *RayCasting* ($RC_{DSsmall} = 1.48$, $RC_{DSlarge} = 1.23$).

For high density ($DS = small$), *RayLens* is significantly the most accurate technique. Pairwise comparisons show significant differences between *RayLens* and *RayCasting* ($p = 0.009$) and between *RayLens* and *RaySlider* ($p = 0.046$) for high density. We find no significant difference between *RayCasting* and *RaySlider*. For low density ($DS = large$), *RaySlider* is significantly the most accurate technique. Pairwise comparisons find a significant difference between *RaySlider* and *RayLens* ($p = 0.006$) and between *RaySlider* and *RayCasting* ($p < 0.0001$). We also find that *RayLens* is more accurate than *RayCasting* ($p = 0.009$).

In summary, our results suggest that *RayCasting* is the least accurate technique, while *RayLens* and *RaySlider* present a similar number of errors. The accuracy of *RayLens* is relatively independent of *SIZE* and *DS*. By contrast, *RaySlider* appears very impacted by *DS* and *RayCasting* by *SIZE*.

7.3 Subjective Measures: workload & participants feedback

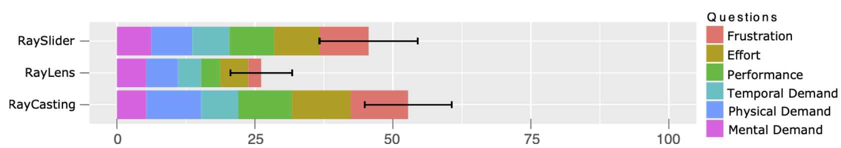


Fig. 7: [NASATLX] Total workload between 0 and 100.

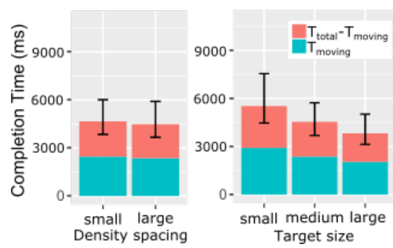
Workload. *RayLens* requires the lowest workload by far ($26.16/100 \pm 5.6$), see Fig. 7. The other techniques present similar workloads ($45.6/100 \pm 8.9$ for *RaySlider* and $52.78/100 \pm 7.9$ for *RayCasting*). Participants find that *RayLens* is much less frustrating ($\approx 4\times$ less), that it requires less effort ($\approx 2\times$ less), and that they are more successful than with the other techniques ($\approx 2\times$ more).

Participants ratings. The final questionnaire confirms the results of the RTLX questionnaire: 100 % of the participants choose *RayLens* as the least frustrating, the most accurate, and the most successful technique. We thought that moving the lens to find a target in a dense environment would be difficult for the participants. However, they report no difficulty with the motion of the lens for almost all of the trials. *RayLens* is also chosen as the least physically demanding technique (*RayLens*: 9/12 participants, *RaySlider*: 3/12 participants), and as the technique which demands the least effort (*RayLens*: 9/12 participants, *RaySlider*: 2/12 participants, *RayCasting*: 1/12 participant). Results also show that *RayCasting* is the least mentally demanding technique (*RayCasting*: 8/12 participants, *RaySlider*: 3/12 participants, *RayLens*: 1/12 participants).

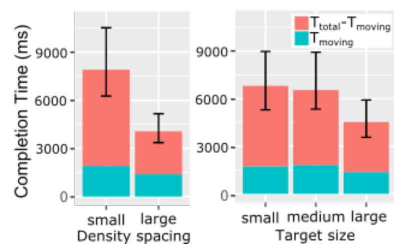
The participants unanimously prefer *RayLens* (12/12 participants). They report liking this technique because it is easy to understand, intuitive, and because the selections are easy and stable. *RaySlider* is the least preferred technique for 5/12 participants and *RayCasting* for the 7 remaining participants.

The usefulness of the visual feedback. We also ask the participants how useful the visual feedback of *RaySlider* and *RayLens* are. For *RaySlider*, the usefulness of the transparency filter applied on the objects placed in front of the cursor is rated 5/7. For *RayLens*, the usefulness of the transparency filter in front of the lens is rated 5.6/7. The visual feedback representing the field of view of the lens is also found useful by the participants (4.6/7).

7.4 Analysis of lens and cursor movement durations



(a) Times spent moving the lens according to *DS* and *SIZE*.



(b) Times spent moving the cursor according to *DS* and *SIZE*.

Fig. 8: Times spent moving the lens or the cursor (T_{moving}) over total completion times (T_{total}).

Fig. 8 shows the average time spent to move the lens and the cursor according to the target size and the density spacing. On average, the participants spent $54.9\% \pm 15\%$ of the trial durations moving the lens and only spent $34\% \pm 9\%$ of the trial durations moving the cursor.

For *RayLens*, the time spent moving the lens is slightly impacted by target size ($T_{movingLens} = 2.04s$ for large targets, $2.37s$ for medium targets, and $2.91s$ for small targets). However, we find no significant effect of *SIZE* nor of *DS* ($T_{movingLens} = 2.36s$ for $DS = large$ and $2.46s$ for $DS = small$) on this duration, see Fig. 8a. For *RaySlider*, we find no significant effect of *SIZE* ($T_{movingCursor} = 1.37s$ for large targets, $1.82s$ for medium targets, and $1.74s$ for small targets) nor of *DS* ($T_{movingCursor} = 1.4s$ for $DS = large$ and $1.91s$ for $DS = small$) on the time spent moving the cursor, see Fig. 8b.

8 DISCUSSION

8.1 Hypotheses

Two extensions faster and more accurate than *RayCasting* for selecting small targets. For the selection of large targets, *RayCasting* remains the best option: it allows users to select large targets faster and with the same accuracy as the two extensions.

H1 suggested that *RaySlider* and *RayLens* would perform better than *RayCasting* with small targets. This study highlights the limitations of *RayCasting* and clearly shows an advantage of *RayLens* and *RaySlider* in terms of precision for the selection of small targets. Our results also suggest that *RayLens* is faster than *RayCasting*. However, as *RaySlider* is only slightly faster than *RayCasting*, this partially validates *H1*. For medium size targets, *RayCasting* is as fast as *RayLens*: this might reveal the existence of a threshold size below which *RayLens* is more accurate than *RayCasting*.

***RayLens* little impacted by the density of the environment.** This study shows that *RayLens* presents similar completion times and number of errors regardless of density. On the other hand, *RaySlider* is strongly impacted by density. For the case of low environment density (large density spacing), *H2* assumed that *RaySlider* would perform better than *RayLens*. However, all three techniques show similar completion times at low density, so this invalidates *H2*. For the case of a high density of distractors, the *H3* hypothesis assumed that the performance of *RayLens* would surpass those of *RaySlider*. The results show that *RayLens* is the most precise technique in dense environments. We do not observe a statistically significant difference in terms of completion time, but the results suggest that on average *RayLens* is more than 1.6× faster than *RaySlider* at high density. *H3* is therefore partially validated.

Higher workload with *RayCasting* and *RaySlider* than with *RayLens*. *RayLens* is unanimously the most preferred technique of this study. The RTLX questionnaire reveals that *RayLens* requires the lowest workload. In contrast, *RayCasting* and *RaySlider* show similar results and a much higher workload than *RayLens*. Thus, switching between the 3D view of the scene and the 2D projected view displayed on the lens does not seem to involve additional cognitive efforts as assumed by the *H4* hypothesis: *H4* is therefore invalidated.

8.2 Control of the cursor, the lens and the ray

***RaySlider*: coupled control of cursor and ray, the slider mainly used during the ballistic phase.** The placement of the cursor in the scene with the slider held by the non-dominant hand is reported as fast by some participants (participant 8 reports an "automatic movement with the slider to bring the cursor in the scene"). This technique was designed with the rationale that only the dominant hand is used for the precision phase (the slider being used only for the ballistic phase). This behavior is confirmed by the recorded data. The ratio, *time spent without using the cursor at the end of the task* over the *total time to complete the task*, is 0.48. This ratio shows that the slider is not used during the second half of the total selection time (precision phase).

For the precision phase in dense environments, as we placed 4 distractors around the target, in front, behind, on the left and on the right, participants must be precise not only along the depth axis but also along the left-right axis. The end of the task is thus similar to a task of precisely placing a cursor in 3D, which explains the difficulty of selecting objects in dense environments with *RaySlider*.

***RayLens*: independent control of lens and ray.** *RayLens* is consistent with Guiard's theory principles [19]. As first explained in [24] "such consistency is a good starting point for identifying two-handed usage that seems natural". This is confirmed by our experimental results (Section 7, no learning effect) that show a rapid achievement of fast performance with *RayLens*. Moreover, with the bimanual design of *RayLens*, users can refine the position of the lens (e.g., to change the target magnification) and point at the target at the same time without switching between the movement of the lens and the movement of the ray. This possibility has been observed during the study. However, based on the recorded movements of the lens and of the ray we found it difficult to quantify this synergistic use of the two hands because the ray is permanently controlled by the dominant hand (i.e. no specific mode for controlling the ray). However, 5/12 participants asked during the training phase if it was possible to point at and select the target while moving/holding the lens and they reported this feature useful. The synergistic use (e.g., for a relatively easy selection of a target displayed on the lens) is as useful as the sequential use of placing the lens at a fixed position and then selecting on the lens (e.g., for a difficult selection of a small target even after magnification).

8.3 Limitations and advantages of *RayLens*

Bimanual technique. The bimanual design of *RayLens* is reported as useful and intuitive but the use of the two hands can limit its usability in some contexts. In these cases, this technique could be adapted as a unimanual technique; e.g., the lens could be moved along the ray with the dominant hand by using a touchpad placed on the wand. An alternative is to add a button to switch between the movement of the lens and the movement of the ray.

Physical objects not managed. In an augmented scene, physical objects can be placed on the table and can occlude the target. The current implementation of *RayLens* does not consider these occlusions. A possible extension is to implement an X-ray vision effect [5] of the physical objects placed in front of the lens.

Good accuracy. Results show that the precision of *RayCasting* and *RaySlider* is respectively strongly impacted by the size of the target and the density of the environment. In contrast, *RayLens* presents similar completion times and number of errors, regardless of target size or density. Overall, in dense environments, *RayLens* remains an accurate technique and is 1.7× more accurate than *RaySlider* and almost 3.2× more accurate than *RayCasting*.
Two aspects of *RayLens* can explain these results. First, magnification makes the pointing task easier, especially for small targets. For example, if we consider a lens positioned in the motor space 35 cm from a small target 0.5 cm in diameter, the width of the target projected onto the lens is 2 cm: more than 4 times its initial diameter. Besides, magnification increases in motor space and in visual space the distance between nearby distractors and the target. Finally, the transparency filter applied to the objects placed in front of the lens removes all visual and motor occlusion of the target. Therefore, targets become larger, density and occlusion are decreased, contributing to good accuracy.

A facilitated 2D selection on the lens. According to the results obtained in this study, selection tasks with *RayLens* are quick and easy in dense environments and with small targets. The design of this technique may explain these results. First, the selection of an object is done on the lens and not directly on the 3D object. This leads to a shortened pointing distance which also reduces the problem of ray stability. Second, the user points to the 2D plane defined by the lens: the 3D pointing task is facilitated as it becomes a 2D pointing task. Also, the target is enlarged due to the magnification.

An easy-to-learn/use technique. *RayLens* requires the lowest workload and is the preferred technique. As stated in Section 8.2, its bimanual design consistent with Guiard's theory principles implies a two-handed usage that seems "natural" [24] and avoids a learning phase (see Section 7).

9 CONCLUSION

We presented *RayLens*, a novel bimanual interaction technique extending raycasting for accurate selection in dense environments. It combines a ray with a virtual 2D magnification lens that can be remotely moved in the 3D space using the non-dominant hand. We compared *RayLens* with a standard raycasting and *RaySlider*, a bimanual raycasting that uses a target expansion mechanism.

Our comparative study first confirms that raycasting is fast for the selection of large objects and also highlights its limitations with smaller objects. In the latter case, the two extensions show much higher accuracy than raycasting. However, the performance of *RayLens* and *RaySlider* differs in dense environments. The performance of *RaySlider* is highly impacted by density, while *RayLens* offers very stable performance (both with respect to density and target size) thanks to the magnification effect and the stability of its lens in the 3D space. Qualitative results also show that *RayLens* is the preferred technique, it is easy to understand and requires the lowest workload.

According to our results, we suggest using a 2D magnification lens as an aid for accurate distant selection. While raycasting is controlled with the dominant hand, the *RayLens* extension on the other hand should be used only when needed (i.e. small targets, dense environments). Besides, we evaluate this technique in the context of tabletop AR but *RayLens* is relevant in all the AR and VR applications relying on the widely-used raycasting technique, when the two hands are free. As it is easy-to-learn and intuitive, we also believe that *RayLens* can be used by novice users, for instance in public applications where the users have to rapidly master the technique (e.g., to explore a 3D map of a site to visit).

As future work, we plan to perform an in-depth study of "the temporal overlap in the performance of the two sub-tasks" [24] involved in *RayLens* (lens and ray movements). We also plan to design an adaptive size of the lens for very distant selections (e.g., with the size of the lens increasing proportionally with its distance to the user).

10 Acknowledgements

We gratefully acknowledge the support of the AP2 project ANR-15-CE23-0001.

References

1. Agarwal, B., Stuerzlinger, W.: Widgetlens: A system for adaptive content magnification of widgets. In: Proceedings of the 27th International BCS Human Computer Interaction Conference. BCS-HCI '13, Swindon, GBR (2013)
2. Bacim, F., Kopper, R., Bowman, D.A.: Design and evaluation of 3d selection techniques based on progressive refinement. Int. J. Hum.-Comput. Stud. **71**(7–8), 785–802 (Jul 2013), `https://doi.org/10.1016/j.ijhcs.2013.03.003`
3. Balakrishnan, R., Kurtenbach, G.: Exploring bimanual camera control and object manipulation in 3d graphics interfaces. In: Proceedings of CHI '99. p. 56–62. New York, NY, USA (1999), `https://doi.org/10.1145/302979.302991`
4. Baloup, M., Pietrzak, T., Casiez, G.: Raycursor: A 3d pointing facilitation technique based on raycasting. In: Proceedings of CHI '19. p. 1–12. New York, NY, USA (2019), `https://doi.org/10.1145/3290605.3300331`
5. Bane, R., Hollerer, T.: Interactive tools for virtual x-ray vision in mobile augmented reality. In: Proceedings of ISMAR '04. p. 231–239. USA (2004), `https://doi.org/10.1109/ISMAR.2004.36`
6. Banerjee, A., Burstyn, J., Girouard, A., Vertegaal, R.: Pointable: An in-air pointing technique to manipulate out-of-reach targets on tabletops. In: Proceedings of ITS' 11. p. 11–20. NY, USA (2011), `https://doi.org/10.1145/2076354.2076357`
7. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D.: Toolglass and magic lenses: The see-through interface. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques. p. 73–80. SIGGRAPH '93, New York, NY, USA (1993), `https://doi.org/10.1145/166117.166126`
8. Blanch, R., Ortega, M.: Benchmarking pointing techniques with distractors: Adding a density factor to fitts' pointing paradigm. In: Proceedings of CHI '11. p. 1629–1638 (2011), `https://doi.org/10.1145/1978942.1979180`
9. Brown, L.D., Hua, H.: Magic lenses for augmented virtual environments. IEEE Comput. Graph. Appl. p. 64–73 (2006), `https://doi.org/10.1109/MCG.2006.84`
10. Brown, L.D., Hua, H., Gao, C.: A widget framework for augmented interaction in scape. In: Proceedings of UIST '03. p. 1–10. New York, NY, USA (2003), `https://doi.org/10.1145/964696.964697`
11. Cashion, J., Wingrave, C., LaViola Jr., J.J.: Dense and dynamic 3d selection for game-based virtual environments. IEEE TVCG **18**(4), 634–642 (Apr 2012), `https://doi.org/10.1109/TVCG.2012.40`
12. Chapuis, O., Dragicevic, P.: Effects of motor scale, visual scale, and quantization on small target acquisition difficulty. ACM Trans. Comput.-Hum. Interact. **18**(3) (Aug 2011), `https://doi.org/10.1145/1993060.1993063`
13. Cockburn, A., Karlson, A., Bederson, B.B.: A review of overview+detail, zooming, and focus+context interfaces. ACM Comput. Surv. **41**(1) (Jan 2009), `https://doi.org/10.1145/1456650.1456652`
14. Dedual, N.J., Oda, O., Feiner, S.K.: Creating hybrid user interfaces with a 2d multi-touch tabletop and a 3d see-through head-worn display. In: Proceedings of ISMAR '11. p. 231–232. USA (2011), `https://doi.org/10.1109/ISMAR.2011.6092391`
15. Fuhrmann, A., Gröller, E.: Real-time techniques for 3d flow visualization. In: Proceedings of VIS '98. p. 305–312. Washington, DC, USA (1998), `https://doi.org/10.5555/288216.288296`
16. Gasteiger, R., Neugebauer, M., Beuing, O., Preim, B.: The flowlens: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms. IEEE TVCG p. 2183–2192 (2011), `https://doi.org/10.1109/TVCG.2011.243`

17. Grossman, T., Balakrishnan, R.: The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In: Proceedings of CHI '05. p. 281–290. NY, USA (2005), https://doi.org/10.1145/1054972.1055012
18. Grossman, T., Balakrishnan, R.: The design and evaluation of selection techniques for 3d volumetric displays. In: Proceedings of UIST '06. p. 3–12. New York, NY, USA (2006), https://doi.org/10.1145/1166253.1166257
19. Guiard, Y.: Asymmetric division of labor in human skilled bimanual action. Journal of Motor Behavior **19**(4), 486–517 (1987), https://doi.org/10.1080/00222895.1987.10735426, pMID: 15136274
20. Guillon, M., Leitner, F., Nigay, L.: Investigating visual feedforward for target expansion techniques. In: Proceedings of CHI '15. p. 2777–2786. New York, NY, USA (2015), https://doi.org/10.1145/2702123.2702375
21. Gutwin, C.: Improving focus targeting in interactive fisheye views. In: Proceedings of CHI '02. p. 267–274. New York, NY, USA (2002), https://doi.org/10.1145/503376.503424
22. de Haan, G., Koutek, M., Post, F.H.: Intenselect: Using dynamic object rating for assisting 3d object selection. In: Proceedings of the 11th Eurographics Conference on Virtual Environments. p. 201–209. EGVE'05, Goslar, DEU (2005), https://doi.org/10.5555/2385984.2386013
23. Hart, S.G.: 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), https://doi.org/10.1177/154193120605000909
24. Kabbash, P., Buxton, W., Sellen, A.: Two-handed input in a compound task. In: Proceedings of CHI '94. p. 417–423. New York, NY, USA (1994), https://doi.org/10.1145/191666.191808
25. Käser, D.P., Agrawala, M., Pauly, M.: Fingerglass: Efficient multiscale interaction on multitouch screens. In: Proceedings of CHI '11. p. 1601–1610. New York, NY, USA (2011), https://doi.org/10.1145/1978942.1979175
26. Kluge, S., Gladisch, S., Freiherr von Lukas, U., Staadt, O., Tominski, C.: Virtual lenses as embodied tools for immersive analytics. In: GI VR / AR Workshop (2020), https://doi.org/10.18420/vrar2020_8
27. Kopper, R., Bacim, F., Bowman, D.A.: Rapid and accurate 3d selection by progressive refinement. In: Proceedings of 3DUI '11. p. 67–74. USA (2011), https://doi.org/10.5555/2013881.2014213
28. Laukkanen, J., Isokoski, P., Räihä, K.J.: The cone and the lazy bubble: Two efficient alternatives between the point cursor and the bubble cursor. In: Proceedings of CHI '08. p. 309–312. New York, NY, USA (2008), https://doi.org/10.1145/1357054.1357107
29. Lee, J.J., Park, J.M.: 3d mirrored object selection for occluded objects in virtual environments. IEEE Access **8**, 200259–200274 (2020), https://doi.org/10.1109/ACCESS.2020.3035376
30. Looser, J., Billinghurst, M., Cockburn, A.: Through the looking glass: The use of lenses as an interface tool for augmented reality interfaces. In: Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia. p. 204–211. GRAPHITE '04, New York, NY, USA (2004), https://doi.org/10.1145/988834.988870
31. Looser, J., Billinghurst, M., Grasset, R., Cockburn, A.: An evaluation of virtual lenses for object selection in augmented reality. In: Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia

and Southeast Asia. p. 203–210. GRAPHITE '07, New York, NY, USA (2007), https://doi.org/10.1145/1321261.1321297
32. Looser, J., Grasset, R., Billinghurst, M.: A 3d flexible and tangible magic lens in augmented reality. In: Proceedings of ISMAR '07. p. 1–4. USA (2007), https://doi.org/10.1109/ISMAR.2007.4538825
33. Lu, Y., Yu, C., Shi, Y.: Investigating bubble mechanism for ray-casting to improve 3d target acquisition in virtual reality. In: Conference on VR and 3D User Interfaces. pp. 35–43. USA (2020), https://doi.org/10.1109/VR46266.2020.00021
34. Mendez, E., Kalkofen, D., Schmalstieg, D.: Interactive context-driven visualization tools for augmented reality. In: Proceedings of ISMAR '06. pp. 209–218 (2006), https://doi.org/10.1109/ISMAR.2006.297816
35. Montano, R., Nguyen, C., Kazi, R., Subramanian, S., DiVerdi, S., Martinez Plasencia, D.: Slicing-volume: Hybrid 3d/2d multi-target selection technique for dense virtual environments. pp. 53–62 (03 2020), https://doi.org/10.1109/VR46266.2020.1581198507712
36. Mota, R.C.R., Rocha, A., Silva, J.D., Alim, U., Sharlin, E.: 3de interactive lenses for visualization in virtual environments. In: 2018 IEEE Scientific Visualization Conference (SciVis). pp. 21–25 (2018), https://doi.org/10.1109/SciVis.2018.8823618
37. Mott, M.E., Wobbrock, J.O.: Beating the bubble: Using kinematic triggering in the bubble lens for acquiring small, dense targets. In: Proceedings of CHI '14. p. 733–742. NY, USA (2014), https://doi.org/10.1145/2556288.2557410
38. Olwal, A., Feiner, S.: The flexible pointer: An interaction technique for augmented and virtual reality (03 2012)
39. Payne, A.R., Plimmer, B., McDaid, A., Luxton-Reilly, A., Davies, T.C.: Expansion cursor: A zoom lens that can be voluntarily activated by the user at every individual click. In: Proceedings of the 28th Australian Conference on Computer-Human Interaction. p. 81–90. OzCHI '16, New York, NY, USA (2016), https://doi.org/10.1145/3010915.3010942
40. Pietriga, E., Appert, C.: Sigma lenses: Focus-context transitions combining space, time and translucence. In: Proceedings of CHI '08. p. 1343–1352. New York, NY, USA (2008), https://doi.org/10.1145/1357054.1357264
41. Pindat, C., Pietriga, E., Chapuis, O., Puech, C.: Drilling into complex 3d models with gimlenses. In: Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology. p. 223–230. VRST '13, New York, NY, USA (2013), https://doi.org/10.1145/2503713.2503714
42. Plasson, C., Cunin, D., Laurillau, Y., Nigay, L.: Tabletop ar with hmd and tablet: A comparative study for 3d selection. In: Proceedings of ISS '19. p. 409–414 (2019), https://doi.org/10.1145/3343055.3360760
43. Plasson, C., Cunin, D., Laurillau, Y., Nigay, L.: 3d tabletop ar: A comparison of mid-air, touch and touch+mid-air interaction. In: Proceedings of AVI '20 (2020), https://doi.org/10.1145/3399715.3399836
44. Ramos, G., Cockburn, A., Balakrishnan, R., Beaudouin-Lafon, M.: Pointing lenses: Facilitating stylus input through visual-and motor-space magnification. In: Proceedings of CHI '07. p. 757–766. New York, NY, USA (2007), https://doi.org/10.1145/1240624.1240741
45. Ramos Mota, R.C., Cartwright, S., Sharlin, E., Hamdi, H., Costa Sousa, M., Chen, Z.: Exploring immersive interfaces for well placement optimization in reservoir models. In: Proceedings of SUI '16. p. 121–130 (2016), https://doi.org/10.1145/2983310.2985762

46. Reipschläger, P., Dachselt, R.: Designar: Immersive 3d-modeling combining augmented reality with interactive displays. In: Proceedings of ISS '19. p. 29–41. New York, NY, USA (2019), https://doi.org/10.1145/3343055.3359718
47. Ro, H., Chae, S., Kim, I., Byun, J., Yang, Y., Park, Y., Han, T.: A dynamic depth-variable ray-casting interface for object manipulation in ar environments. In: IEEE International Conference on Systems, Man, and Cybernetics. pp. 2873–2878 (2017). https://doi.org/10.1109/SMC.2017.8123063
48. Schmalstieg, D., Höllerer, T.: Augmented reality: Principles and practice. In: 2017 IEEE Virtual Reality (VR). pp. 425–426 (March 2017). https://doi.org/10.1109/VR.2017.7892358
49. Spindler, M., Dachselt, R.: Paperlens: Advanced magic lens interaction above the tabletop. In: Proceedings of ITS '09. New York, NY, USA (2009), https://doi.org/10.1145/1731903.1731948
50. Spindler, M., Tominski, C., Schumann, H., Dachselt, R.: Tangible views for information visualization. In: Proceedings of ITS '10. p. 157–166. New York, NY, USA (2010), https://doi.org/10.1145/1936652.1936684
51. Stoev, S., Schmalstieg, D., Straer, W.: Two-handed through-the-lens-techniques for navigation in virtual environments (08 2001), https://doi.org/10.2312/EGVE/EGVE01/051-060
52. Stoev, S., Schmalstieg, D., Straßer, W.: The through-the-lens metaphor: Taxonomy and application. pp. 285–286 (03 2002), https://doi.org/10.1109/VR.2002.996541
53. Tominski, C., Gladisch, S., Kister, U., Dachselt, R., Schumann, H.: Interactive lenses for visualization: An extended survey. Comput. Graph. Forum **36**(6), 173–200 (Sep 2017), https://doi.org/10.1111/cgf.12871
54. Tong, X., Li, C., Shen, H.W.: Glyphlens: View-dependent occlusion management in the interactive glyph visualization. IEEE TVCG **23**(1), 891–900 (Jan 2017), https://doi.org/10.1109/TVCG.2016.2599049
55. Traoré, M., Hurter, C., Telea, A.: Interactive obstruction-free lensing for volumetric data visualization. IEEE TVCG **25**(1), 1029–1039 (Jan 2019), https://doi.org/10.1109/TVCG.2018.2864690
56. Vanacken, L., Grossman, T., Coninx, K.: Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In: Proceedings of 3DUI '07 (2007), https://doi.org/10.1109/3DUI.2007.340783
57. Vanacken, L., Grossman, T., Coninx, K.: Multimodal selection techniques for dense and occluded 3d virtual environments. International Journal of Human-Computer Studies p. 237–255 (2009), https://doi.org/10.1016/j.ijhcs.2008.09.001
58. Vickers, D.L.: Sorcerer's Apprentice: Head-Mounted Display and Wand. Ph.D. thesis (1972), https://doi.org/10.5555/906408, aAI7310165
59. Viega, J., Conway, M.J., Williams, G., Pausch, R.: 3d magic lenses. In: Proceedings of UIST '96. p. 51–58. New York, NY, USA (1996), https://doi.org/10.1145/237091.237098
60. Vogel, D., Baudisch, P.: Shift: A technique for operating pen-based interfaces using touch. In: Proceedings of CHI '07. p. 657–666. New York, NY, USA (2007), https://doi.org/10.1145/1240624.1240727
61. Wobbrock, J.O., Findlater, L., Gergle, D., Higgins, J.J.: The aligned rank transform for nonparametric factorial analyses using only anova procedures. In: Proceedings of CHI '11. p. 143–146 (2011), https://doi.org/10.1145/1978942.1978963