Poros: Configurable Proxies for Distant Interactions in VR

Henning Pohl
henning@di.ku.dk
University of Copenhagen
Copenhagen, Denmark

Klemen Lilija
lilija@di.ku.dk
University of Copenhagen
Copenhagen, Denmark

Jess McIntosh
jemc@di.ku.dk
University of Copenhagen
Copenhagen, Denmark

Kasper Hornbæk
kash@di.ku.dk
University of Copenhagen
Copenhagen, Denmark

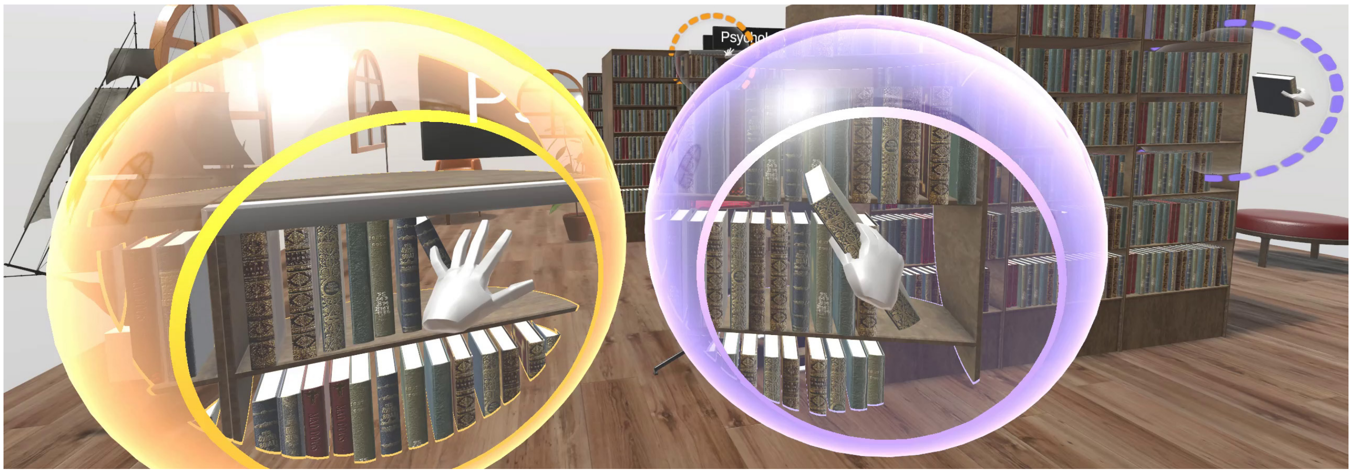

Figure 1: Poros enables users to bring portions of distant spaces closer so that they can interact with and across them. Shown here are two proxies, linked to marked spaces (shown in the same color) around two different bookshelves. The user is about to move a book from one space to the other. In addition to direct interactions through them, users can move and arrange proxies, as well as perform operations on them, such as merging and aligning.

ABSTRACT

A compelling property of virtual reality is that it allows users to interact with objects as they would in the real world. However, such interactions are limited to space within reach. We present *Poros*, a system that allows users to rearrange space. After marking a portion of space, the distant *marked space* is mirrored in a nearby *proxy*. Thereby, users can arrange what is within their reachable space, making it easy to interact with multiple distant spaces as well as nearby objects. Proxies themselves become part of the scene and can be moved, rotated, scaled, or anchored to other objects. Furthermore, they can be used in a set of higher-level interactions such as alignment and action duplication. We show how Poros enables a variety of tasks and applications and also validate its effectiveness through an expert evaluation.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; *Pointing*.

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445685

KEYWORDS

virtual reality, portals, worlds in miniature, 3D user interface

ACM Reference Format:
Henning Pohl, Klemen Lilija, Jess McIntosh, and Kasper Hornbæk. 2021. Poros: Configurable Proxies for Distant Interactions in VR. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411764.3445685

1 INTRODUCTION

Virtual reality (VR) enables high levels of immersion [24] but at a cost: immersive interaction is often not efficient interaction. For example, reaching an object at the other end of a room requires physical effort as the user first needs to walk to the object. This issue is exacerbated when users need to be at different locations in quick succession. Locomotion techniques, such as teleportation, can help but incur a cost every time the user switches position (e.g., due to disorientation [6]). An alternative approach is to enable users to directly act at a distance.

Several techniques have been proposed to enable more efficient distant interactions in VR. For example, users' reach can be extended (e.g., by extending hands [31] or raycasting) or the distant space can be brought closer (e.g., by portals [13]). Both of these approaches have unresolved problems. Extending the user's reach decreases the precision of interaction due to angular error. Bringing the distant space closer can interfere with interaction by distorting the visual space [8], creating inconsistency of hand mapping [21], making it difficult to notice the boundary between the space that is brought closer and the scene, or by occluding the scene as portals do [13]. Worlds in miniature [34] avoid some of these problems by enabling easy access to other parts of the scene while existing outside of the scene itself. However, in their current forms worlds in miniature only afford limited manipulation support, such as not allowing users to create and arrange them as they see fit.

Building on worlds in miniature and portals, we propose *Poros*, where portions of space can be marked and linked to proxies. The proxies can be brought close to the user and allow all interactions as if they were next to the marked space, effectively acting as surrogates [15]. This approach does not distort the visual space, keeps the 1:1 hand mapping, allows users to interact simultaneously with both distant spaces and nearby objects, makes the boundaries between multiple spaces visible, and allows multiple levels of indirection. As shown in Figure 1, this can, for example, be used to move objects efficiently between two distant spaces.

Proxies do not just enable direct interaction with distant spaces. They become part of the scene and can be transformed, arranged, or anchored to objects. For example, a user can create a proxy that is linked to a toolbox and anchor it to themselves. In this way, the user can always access a tool (e.g., a ruler) through the self-anchored proxy. Further interactions are possible once multiple proxies have been created. For example, users can perform operations between proxies to align them, highlight shared objects, or duplicate actions by linking multiple marks to a proxy. To demonstrate the usefulness of Poros, we show several examples of interactions with distant spaces, interactions at different scales, multiple perspectives, and multi-proxy interactions. We also validate that Poros can be easily understood and used effectively through an expert evaluation.

1.1 Contributions

Our paper makes the following contributions:

- We present a VR system, *Poros*, that enables users to create proxies to distant spaces and interact through them—all using direct hand-based manipulations.
- We show a range of operations on single, pairs, and groups of proxies enabled by having proxies as objects in the scene.
- We demonstrate that by combining operations, Poros enables many interaction techniques, including: occluded object interactions, interacting on different scales, multi-perspective manipulation, replicated interactions, space searching, and inter-space alignment.
- We validate Poros' effectiveness with an expert evaluation.

2 RELATED WORK

Our paper is informed by previous work on enabling interaction with distant targets in VR. In particular, interaction techniques for distant reaching (e.g., ray-casting), ways of warping the virtual space (e.g., Erg-O [21]), and the creation of meta-spaces (e.g., overviews). Each of these three approaches has shortcomings that our technique addresses.

2.1 Interaction Techniques for Distant Targets

There are a large number of selection techniques for VR [2]. When selecting distant targets, two classic examples are ray-casting and arm-extension [5]. An example of the latter is *go-go*, where the user's arm grows non-linearly as it moves away from their body [31]. The PRECIOUS technique solves one of the problems of ray-casting: disambiguation between close targets [19]. If a selection is ambiguous, the user is moved closer to the candidate targets, makes a selection there, and afterward moved back to the original position. Another approach to disambiguation was proposed by Pierce et al. who used hand gestures to, for example, put a frame around the desired object [27]. Similarly, gaze information can be used for selecting distant objects which can then be manipulated as if they are close [26]. Another approach for improving ray-casting is to "bend" the ray towards potential targets [33] or through user control [22], easing the selection of dense, occluded, or distant targets. Common issues with these techniques are that (1) they distort or move the user's body, potentially impacting body ownership or disorienting the user [6], and that (2) interaction no longer occurs directly through the user's hands, which has similar repercussions. With Poros, users' bodies are not altered. Instead of extending their reach, moving them, or introducing pointer-like constructs, we move parts of space, bringing objects of interest into reach.

2.2 Warping and Moving Space for Interaction

Another approach to bringing targets closer is to warp the whole space. For instance, Chae et al. presented an augmented reality (AR) technique where users can shrink a room along one axis to bring distant objects closer [8]. Also for AR, Sandor et al. built a system that distorts space to show points of interest that are out of view or occluded [32]. In VR, Mine et al. scale the world as users grab distant objects for manipulation [20]. In Elmqvist's *BalloonProbe* technique only objects are warped instead of the whole space [9]. By repelling objects away from each other, occluded ones can be accessed more easily. In general, warping or distorting the space requires users to adapt to new, likely unfamiliar, spaces. This is exacerbated in non-linearly warped spaces, where movement and interaction can be particularly difficult.

Another method to warp space is the use of portals. With portals, arbitrary locations in space can be linked—stepping through one portal instantly moves one to the linked location. This can be used to shorten distances, but also to break the spatial consistency of a virtual world [13]. Stoev and Schmalstieg named this *through-the-lens* interaction and discussed how it can be used for a range of tools [35]. Portals can also be used to avoid the use of teleporting within a virtual environment [18]. With *PhotoPortals*, Kunert et al. proposed the use of portals for easier collaboration [14]. Users can capture views of the scene, manipulate them, and share them with others. Their portals also include a view mode where a whole cuboid slice of the captured space is shown. *SpaceTime* [38] also focuses on supporting collaboration. Containers in *SpaceTime* could be seen as portals to another place as they only allow teleportation to the remote content contained in it and not direct manipulation through it. But *SpaceTime* also breaks the spatio-temporal consistency of the world, enabling objects within containers to exist as clones, or time-shifted versions of the original ones. Similar to version control software, this enables collaborators to work in parallel and resolve conflicts later. However, it is unclear if the conflicts that arise by allowing this can be easily handled by direct manipulation in VR.

We take a similar view of distant spaces as *PhotoPortals* and *SpaceTime*, but instead of collaboration we focus on distant interactions and interactions built atop the combination of multiple spaces. Furthermore, we maintain the spatio-temporal consistency of the scene as spaces and objects are only made accessible, not cloned.

2.3 Interaction with Meta-Spaces

Another alternative for extending the reach of users are meta-spaces, such as Stoakley et al.'s *Worlds in Miniature* [34]. In addition to a first-person view, users see an overview of the world in which they can interact as in the main view. Furthermore, the miniature can be used for navigation [23]; in some systems, several miniatures can be available. Instead of functioning as an additional view, the whole world can be turned into a miniature by scaling the user up, which then allows them to move faster through the scene [1]. The main world and the world in miniature can also be distributed between users, enabling collaboration across scales [30].

Worlds in miniature for large or complex spaces can be hard to use and hence Trueba et al. presented several improvements to the technique that clip and filter what is shown [36]. Another version of worlds in miniature is Bluff's *Miniature Metaworld* [4]. In contrast to the work above, his miniature is situated within the original scene. Users can manipulate objects in either the original or miniature view, but also move objects between them. An alternative to showing whole worlds in miniature is to only present distant landmarks to users and allow them to teleport to these [28].

Instead of replicating space, Pierce et al. explored how to replicate individual objects for interaction [29]. Their *Voodoo Dolls* technique enables users to grab distant objects via an image plane technique and then manipulate them in their hands. To show context, nearby objects can be placed in the other hand to, for example, allow for placing an object on a shelf.

In Poros, we also apply the worlds in miniature concept. As with metaworlds, our "miniatures" become part of the original scene. With Poros we improve upon worlds in miniature in several ways, enabling users to (1) create instances on demand and at a distance, (2) manipulate the bounds, orientation, and location of worlds in miniatures already in the scene, (3) anchor worlds in miniature to scene objects or themselves, making them dynamic, (4) merge and split worlds in miniature to replicate spaces and actions, (5) perform operations on worlds in miniature like aligning them or searching them, and (6) gradually peek into worlds in miniature, controlling how much of the view is taken over by them.

3 POROS FOR INTERACTION AT A DISTANCE

The primary goal of Poros is to enable manual interaction with distant objects. The core idea for this is to allow users to bring parts of space closer to themselves and directly interact inside and across those parts. Based on this capability, the second goal of Poros is to enable users to exploit the introduced indirection in order to make common tasks easier. More specifically, Poros allows users to:

- Mark distant spaces and bring them close in the form of *proxies*. Users can then perform **interactions through proxies**, to allow direct interaction with these distant spaces. Users can reach into proxies to manipulate their contents and peek into them to inspect a distant location.
- **Manipulate proxies** as they are first-class citizens in the scene. Users can scale, transform, minimize, align, clone, and otherwise manipulate proxies. Thereby proxies offer more interaction possibilities than worlds in miniature and portals. Users can leverage these possibilities to configure and optimize their workspace for the task at hand. Users can also anchor proxies to other objects and avatars to make these workspaces mobile.
- **Manipulate marks** to adjust what part of space a proxy shows. Marks can also be anchored to objects, which allows for tracking and manipulation of moving objects through proxies.
- Perform **abstract operations on proxies** that change them according to their or other proxies' content. This includes several alignment operations, as well as content-sensitive highlighting, and merging. Making use of these operations allows users to perform complex tasks that would otherwise require many actions or substantial movement.

Figure 2 illustrates the basic components of Poros: (1) users place *marks* in the scene to denote spaces they want to link to, which results in (2) *proxies* close to them that they then can interact with. Proxies are an exact mirror of a subset of the scene; any change to either is reflected in the other. Note that most commonly one proxy is linked to one mark, however, this association can also be one-to-many or many-to-one (as a result of merging and cloning operations, described in Section 3.6.3). In contrast to portals, proxies are not two-dimensional gates to another place, they are three-dimensional replications.

3.1 Setup

We implemented Poros using the Unity game engine. We used an HTC Vive headset for output, with a Leap Motion controller attached for hand tracking. This setup allows users to walk around to explore the scene and use their hands to interact with the scene. The use of hand tracking, instead of controllers, was motivated by three considerations: (a) to heighten the sense of immersion, (b) to allow for complex and high-dimensional control, and (c) to leverage already acquired knowledge of object manipulation from real life.

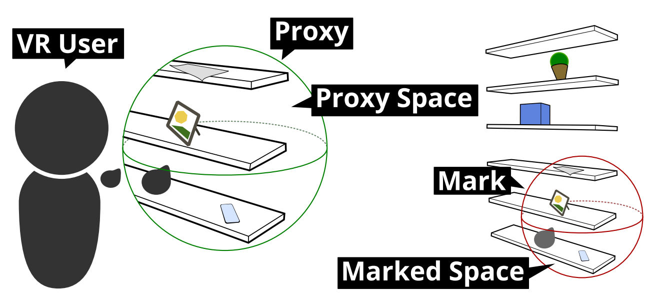

Figure 2: In Poros, users create marks around space they want to bring close and subsequently called. The enclosed space is called the marked space. Through marking, a proxy is also created next to the user. It contains the proxy space, which is an identical mirror of the marked space.

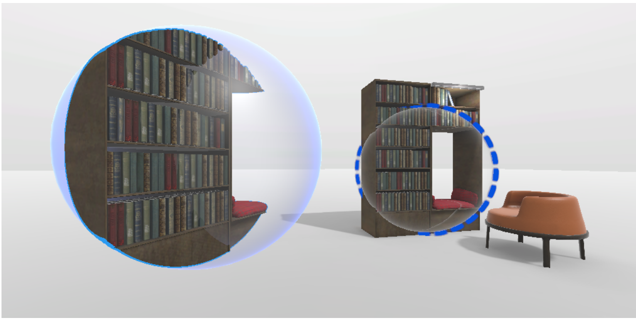

Figure 3: Proxy spaces are rendered inside colored bubbles, marked spaces are shown in a fainter style. A dashed outline with a marching ants effect around marks is colored to hint at their connected proxy space.

For the visual replication inside proxies, we extended the built-in shaders and added a custom render feature to Unity's scriptable render pipeline. That changes the pipeline to add an additional render pass to opaque as well as transparent objects. On top of the default rendering, each object is then rendered another time for each proxy in the scene. For each active proxy that entails culling, modifying transformation matrices, setting global shader clipping data, and drawing the contained objects. To replicate hands, we wrap the Leap provider to transform hand data before passing it on to the Leap interaction system.

Poros is available as open source software[1] so others can extend and try it themselves.

3.2 Basic Properties of Proxies

In Poros, proxies and marked spaces are always spherical. This reduces the complexity for the users when creating or editing them. Conceptually, however, proxies and marks could be of any shape. For example, the interaction technique from *TunnelSlice* would be suitable for marking cubic volumes [16].

Figure 3 shows how proxies and marks are rendered inside a VR scene. Both use a fresnel effect for shading as well as further highlighting where they intersect the scene. This makes for a translucent and ephemeral appearance that limits scene occlusion. A circle with a marching-ants effect circumscribes each mark. Proxies and marks are color-coded to show which ones are connected. As users approach a proxy with their hands or head, the colored shell opens up (see, e.g., Figure 4) in order to provide a clearer view of the contained space.

Proxies display all the content within their linked marked space. If an object only intersects a marked space, it is clipped to the space's boundary. A proxy's content is scaled according to the relative size of the marked and proxy space; users may change that using operations on the proxy (explained later). If a marked space is twice as large as a proxy, its content is shown at half the size inside the proxy. When entering the proxy, the user's hands always stays at the size it had outside of proxy (see Figure 4). Such 1:1 mapping is important for accurate motor control during entry and exit as well as for the manipulation of objects in the proxy. When the hands are inside of a proxy they are also rendered in each of the marked spaces the proxy links to. While the user's hands always stay at the same scale, the replicated hands in the mark are scaled. If a marked space is twice as large as a proxy, so are the replicated hands rendered in it (see Figure 4).

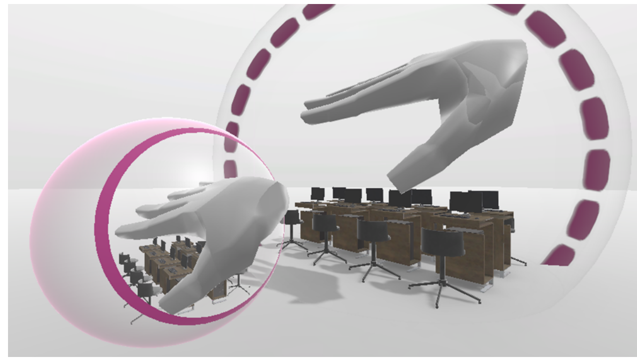

Figure 4: Proxies and marks can exist at different scales. Here the marked space is much larger and the user's hand hence scaled up inside of it.

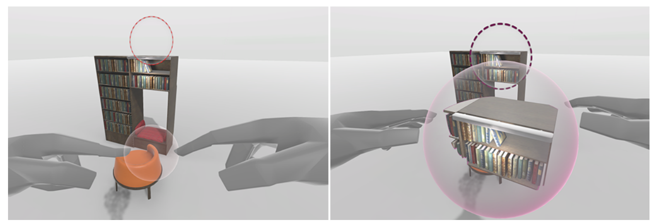

Figure 5: Proxies and marks are created using a bimanual pointing hand gesture. Users can adjust the size and position of the mark inside the scene by moving their hands.

Just as proxy spaces show the visual content of linked mark spaces, they also bring their sounds closer. If a book falls down in a different room, this would normally not be audible to the user. However, if the book hits the ground within a marked space whose proxy is close to the user, that sound is also played at the corresponding location in the proxy space. Only sounds that occur within a marked space are audible in proxies; just as we use a hard boundary for visuals, we also apply this boundary for audio sources.

3.3 Creating Marks and Proxies

In Poros, users can mark a space through manual interaction; this process simultaneously creates a proxy. As shown in Figure 5, users start this creation process by making a hand gesture where the index fingers point towards each other. This creates a proxy between the fingers, and a mark in the distance; similar to aperture-based selection methods [10].

By moving their hands, users can continuously translate and scale the proxy, which is constrained to fit between users' index fingers. This ensures that everything in the proxy (and thus also the marked space) is within reaching distance. As the proxy is sized, the marked space scales proportionally. Furthermore, the distance

[1] Available at https://github.com/henningpohl/poros

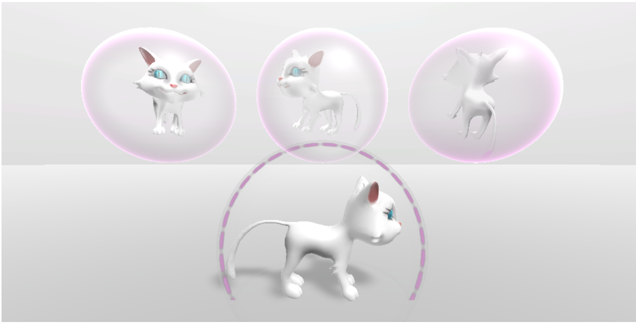

Figure 6: Proxies allow for different views on the scene. Shown here are three proxies linked to the same marked space. Each is rotated in a different way, showing a cat from multiple perspectives.

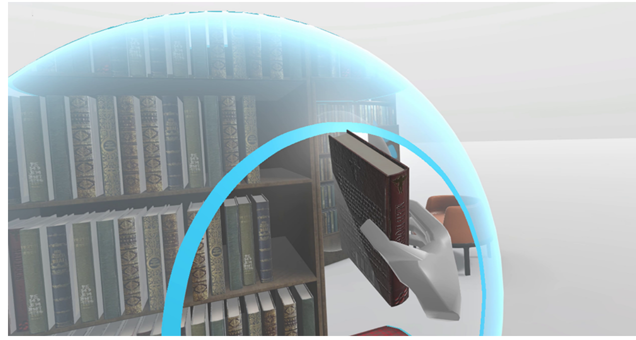

Figure 7: Users can interact with objects inside of proxies, but also move objects into and out of them, like this book. If the proxy was scaled, objects taken out of it soon afterwards shrink or grow to their actual size.

to the proxy changes proportionally to the distance between the user's body and hands. These proportions can be adjusted, but we have found that an exponentially scaling proportion allows for the greatest flexibility. Through this mechanism, users are able to create marked spaces far away from them. The mechanism means that the proxy and mark are dependent on each other, and finer adjustments should be done as a subsequent action afterward. We chose this approach as it fits the direct manipulation approach used in Poros. Alternatively, one could use techniques such as ray interaction [37] or clutching [3] to place the marked space.

3.4 Interactions Through Proxies

The main interaction afforded by proxies is direct manipulation of distant objects. Any movement or action within a proxy is handled exactly as if it occurred in the linked marked space.

3.4.1 Interacting Inside a Proxy Space. When users reach into a proxy space, their hands are effectively transported to the marked space. Consequently, they see their hands twice: within the proxy and within the scene. This extended reach allows users to grab and move objects far away from them but also to push buttons or operate other mechanisms. In Poros, we treat the proxy and the marked space as identical and thus objects are shown twice but do not exist twice. The objects only exist in the scene, that is, at the marked space. This also extends to events triggered via interaction—a button press inside a proxy only triggers one button press event.

The scale and orientation of a proxy space are independent of the rest of the scene. Hence, objects can appear much smaller or larger in the proxy space but also can be seen from other perspectives (see Figure 6). Allowing users to freely pick their desired scale and orientation enables interactions that are hard or impossible to do otherwise. For example, a marked space that encompasses an entire room can allow for an easy rearranging of furniture.

3.4.2 Moving Objects into and out of a Proxy Space. Interaction with distant objects is not limited to the inside of proxies. A common need, for example, is to use these objects elsewhere in the scene. To enable this, objects can be moved into and out of proxy spaces (see Figure 7). When objects enter a proxy they exist only in the scene location, that is, at the marked space. Similarly, when the user takes an object out of the proxy, it only exists in the user's hand and stops existing at the distant location. As the proxy is only a representation of the actual space, an object never exists in two places at once and it always exists at its actual scale.

In most aspects, this behaves exactly as when objects are moved in the rest of the scene. However, with proxy spaces, a few situations arise that require special handling. One is dropping an object in a proxy that only links to an empty volume of space. If a user drops an object in such a proxy it will fall through the bottom of the marked space and come to rest at the distant location.

Another situation that needs special handling is moving objects to and from the proxy spaces of different scales. Because proxy spaces can be magnified or shrunk in comparison to the rest of the scene, objects also appear larger or smaller respectively, even though their actual size is unchanged. For example, when a marked space spans an entire room, the furniture appears much smaller inside a proxy space than it actually is. This brings benefits, such as being able to grasp objects that would otherwise be too large to grasp (e.g., shelves). However, it also requires transitioning to the object's actual scale when they are taken in and out of a proxy or moved between proxies. We handle this by keeping object sizes visually constant while they are moved to and from proxies. Once the user lets go of an object, it snaps back to its actual size. For example, if the furniture is taken out of the proxy described above, it appears small at first, but grows to full size as it is placed down. The same happens when objects are being moved between proxies.

In addition to objects, users can also move *themselves* inside a proxy space (see Figure 8). For Poros, we restricted users to sticking their head into proxies, in order to "peek" at a distant space. When peeking, the proxy space expands, giving the user a wider view of the marked space. This allows for inspection of a distant space, with easy transition back to a normal perspective (i.e., by taking the head out of the proxy). When the user's head is fully inside a proxy, the whole view is identical to the one a user would have at the mark's location in the room—not clipped anymore as when looking at the proxy space from outside. Furthermore, if the proxy is visible from the mark's location then the user can see themselves within it.

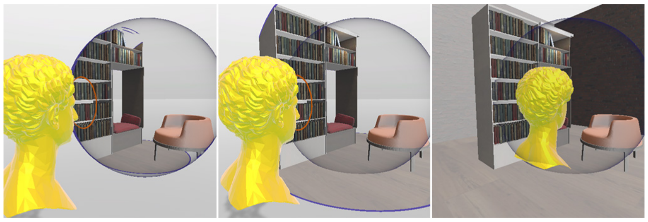

Figure 8: Proxy spaces initially only show what is contained inside. As the user moves into a proxy space, the shown volume expands exponentially until the user is finally immersed in the linked marked space.

Figure 9: By resting their hand on the outside of a proxy, users can activate a manipulation mode. The proxy then turns more opaque and allows users to translate (shown here), scale, and rotate it with their hands.

3.5 Manipulation of Proxies and Marks

As proxies become part of the scene, they also become available for interaction. Users can manipulate proxies, as well as the marked spaces linked to them with manual interactions. This allows users to, for example, arrange proxies, adjust mark locations, or take proxies along as they move around.

3.5.1 Proxy Manipulation. Just like other objects in the scene, proxies can be manipulated. In Poros we allow for translation, scaling, and rotation of proxies (see Figure 9 for an example of translation). These manipulations can be performed by grabbing the shell of the proxy with one or two hands respectively.

However, to prevent accidental changes to a proxy, we require a mode switch to enable manipulation. For this, users have to briefly rest their hand on a proxy's shell. Upon switching to manipulation mode, proxies no longer open up for the user, and appear more opaque. When users move away or stop interacting, proxies return to the default mode for easy access to their contents.

Sometimes a static arrangement of proxies is not sufficient. For example, users might want to take a proxy along when walking. To address this, we allow users to *anchor* proxies to other elements of the scene, including to themselves. To trigger anchoring the user rests one of their hands on a proxy while grabbing and pulling with the other. While the proxy stays in place, a tether emerges and follows the pulling hand. Users can then drag this tether to other objects for anchoring. To anchor a proxy to themselves, users drop the tether at a target that appears near their waist. An anchor can be disengaged by starting the anchoring process again, then releasing in open space.

Figure 10: Users can minimize proxies, which then attach to their wrist. This allows users to take them along. Holding the wrist up and touching a minimized proxy returns it to its original size.

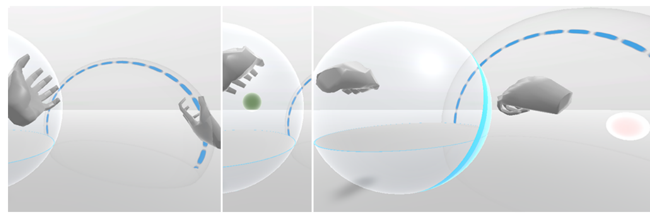

Figure 11: Resting a hand in the inside of a proxy activates mark manipulation. A handle appears that can be used to translate and scale the mark. Shown here is how grabbing it with the left hand and dragging outwards expanded the marked space.

While anchoring enables users to take proxies with them, proxies are not always needed and could obstruct other parts of the scene if always close. For this reason, we also provide a way for users to temporarily store away and bring along proxies (see Figure 10). When scaling a proxy below a certain size, it is instead minimized and flies to a storage location on the user's left forearm. Where anchoring to oneself keeps proxies in view, this alternative storage option allows users to move around without being obstructed by a proxy. Once users need to get access to a proxy again, they can restore it by lifting their forearm and selecting the desired proxy on it. The proxy then returns to its original size and appears in front of the user.

3.5.2 Mark Manipulation. In addition to proxies themselves, we enable users to change the boundaries of the marked spaces they are linked to (see Figure 11). Marked spaces can be moved and resized, changing what part of the scene they encompass, but not the scene itself. As with proxies, mark manipulation needs to be activated. Users have to place their hand on the inside boundary of a proxy (i.e., when their hand is already in the marked space). This makes a handle appear which acts as a joystick for the mark. When grabbing it with their left hand, users can translate the marked space, while grabbing with their right hand controls its scale.

Marked spaces can be anchored to objects (see Figure 12) and users just the same as proxies. This can be useful, for example, if a distant object is moving. Users can anchor a marked space by pinching on the manipulation handle, dragging an emerging tether to an object, and releasing the pinch on it. Self-anchoring and detaching work as in proxy anchoring.

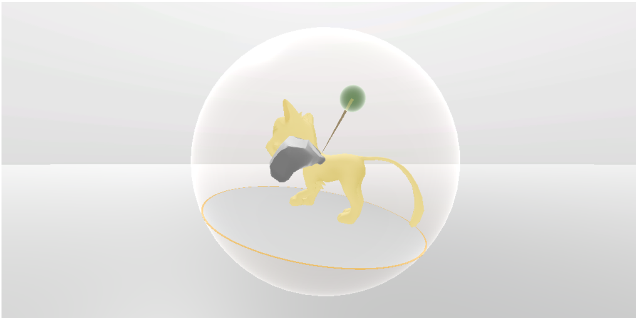

Figure 12: Users can anchor marked spaces to objects within, like this cat. When in manipulation mode, pinching on the handle reveals a tether that can be dragged onto the object to anchor to. As the cat walks around, the marked space will now follow it.

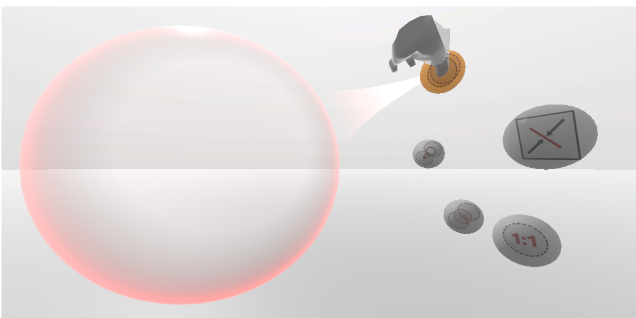

Figure 13: Pinching the surface of a proxy and dragging away reveals a crossing menu with several operations to chose from. Menu items and submenus are selected by dragging through them. For operations with a target location or object, users continue dragging and release on them.

3.6 Abstract Operations on Proxies

Poros also supports abstract operations on one proxy as well as multiple proxies. These can speed up simple manipulations (e.g., aligning of proxies) or enable interactions that would otherwise not be possible (e.g., highlighting of overlapping objects).

3.6.1 Triggering Operations. Operations are triggered by a hierarchical crossing menu, accessible through pinching. When pinching the surface of a proxy and dragging away, the menu shown in Figure 13 appears in the dragging direction. Moving through a menu item selects that item, or triggers the next menu within the hierarchy. The release position of the pinch can be used as parameter for single proxy operations, and to pick the second proxy in multi-proxy operations.

3.6.2 Single Proxy Operations. Poros supports four operations on single proxies: cloning, splitting, aligning to scene, and resetting.

Cloning allows users to create another proxy space that is linked to the same marked space as the proxy the operation was used on. This can be useful to, for example, create multiple views on the same space in order to see an object from different angles.

Splitting is an operation only available on proxies that have previously been merged (see below). Such proxies are linked to multiple marked spaces. With this operation, the proxy splits up into separate proxies—one per linked marked space.

Aligning to the scene rotates a proxy to make it best fit in with the scene around it. For example, a proxy space containing a cabinet, when close to other furniture, would align with it. This operation makes use of the prevalent orientations within a proxy space, as well as the scene around it. We derive orientations from each object's representation in the physics simulation (e.g., as a box). Resulting from this are two sets of world-space direction vectors. We then use the Kabsch algorithm [11] to find the rotation matrix that best aligns the proxy space's content with the surroundings.

Resetting, reverts any orientation and scale differences between a proxy space and the linked marked space. Afterwards, the scale within the proxy is equal to the rest of the scene and everything within is facing the same direction it does in the scene.

3.6.3 Multi-Proxy Operations. Four additional operations work on multiple proxies at once: aligning, convenience aligning, merging, and highlighting. Users activate them with the same menu as above and, after command selection, then drag to the other proxy to include in the operation.

Aligning and **convenience aligning** both use the mechanism for *aligning to the scene* described above. However, instead of aligning to the scene, the former makes groups of proxies face in the same direction (e.g., having a row of cabinets all face forward). The latter takes into account the position of the user to make transfer of objects from one proxy to the other more convenient. In the case of two bookcases, for example, moving books between them is easier if both are tilted towards the user, instead of just facing forward. As object data can be ambiguous with respect to an object's main orientation, we currently manually annotate them with a preferred orientation (e.g., the forward direction for a bookcase).

Merging results in one proxy that is linked to multiple marked spaces. The spaces overlap and are all shown concurrently. While this can, at times, be visually confusing, it does allow for a set of advanced interactions. The user's hands in the proxy are mirrored in each marked space, all moving in unison and all with the ability to interact with the scene. Hence, interaction in a proxy that is linked to multiple marked spaces is replicated across them. For example, pushing down on a button in one also results in pushes in all other spaces (if the buttons are aligned). This feature can be used to move multiple objects at once or trigger multiple actions at once. Such replication has the most potential where a scene contains multiple instances of the same interactive object. For example, consider turning on multiple machines, opening a row of windows, or playing with several slot machines at once.

Highlighting helps users search proxy spaces. Whatever is contained in one proxy defines a lens for another one. For example, one proxy encompassing just one egg can be used to find all the eggs in a second, much larger, proxy. In our current implementation, objects are tagged manually and search is then performed within these annotations. When highlighting, floating exclamation marks are temporarily attached to every matching instance. Users can then see the highlighting within the searched proxy, but also as part of the scene.

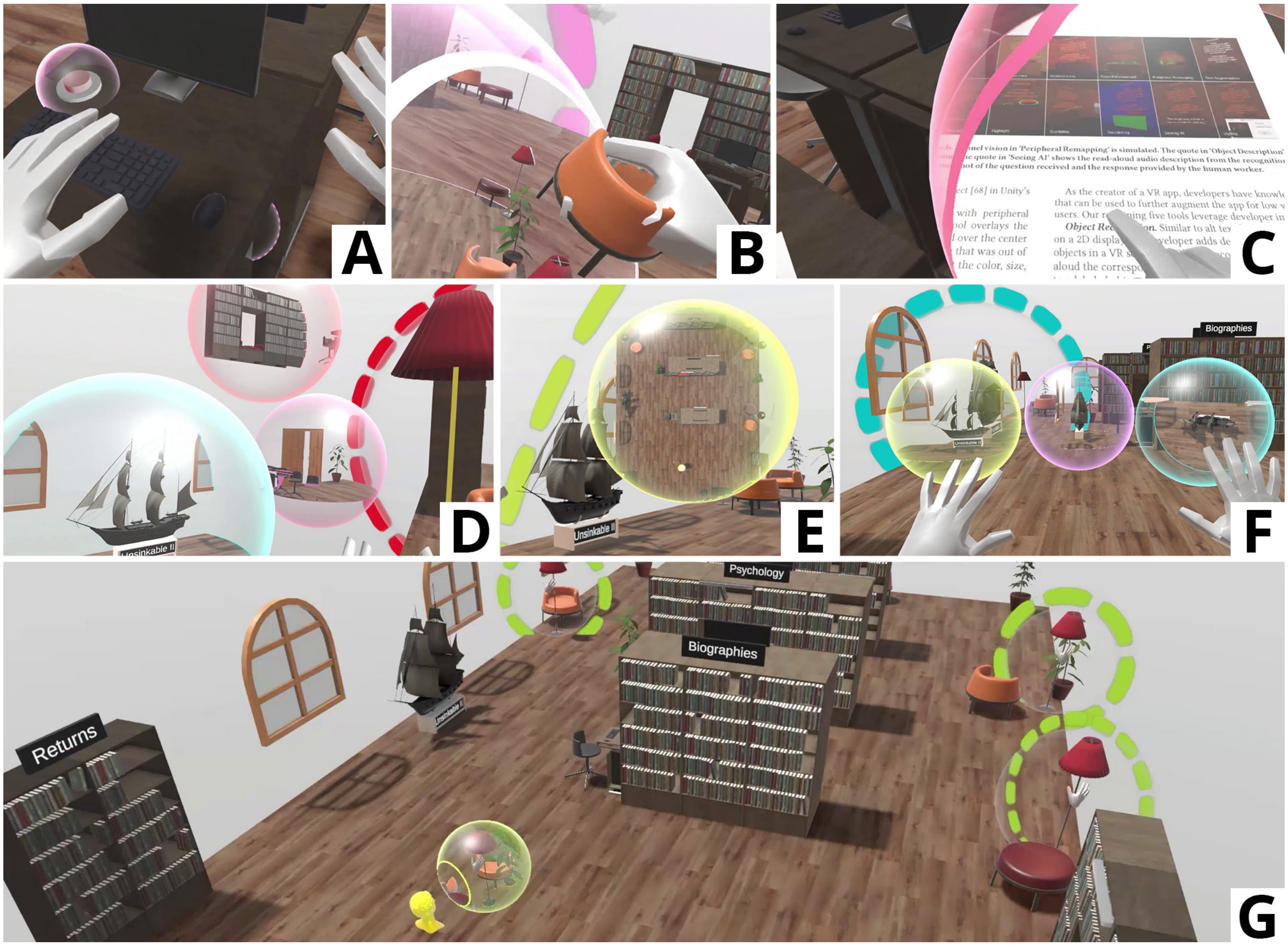

Figure 14: We build a range of examples with Poros. Shown here are (A) re-configuration of space so the PC's power button is on the table, (B) moving furniture, (C) reading small print on books, (D) monitoring different parts of a room, (E) setting up a mini-map that includes oneself, (F) inspecting a sailing ship from multiple perspectives, and (G) replicating an action across multiple spaces to turn off several lamps at once.

4 EXAMPLE APPLICATIONS

Poros enables novel applications but also allows for easy implementation of a range of existing interactions. In this section, we describe several examples of such applications and uses. See Figure 14 for an overview of these examples.

4.1 Occluded Object Interactions

Sometimes what we want to interact with is hidden behind other objects or otherwise hard to get to. For example, imagine a scenario where the user desires to turn on a PC that is under a table (e.g., as in [17]). Since the PC is hard to reach, the user can mark the power button and create a proxy at a more suitable location. For example, the user can place the proxy containing the power button on the table, while anchoring the mark to the PC (Figure 14-A). After rearranging the space the user can now easily access the PC's power button when at the table, even if the PC is moved.

4.2 Large and small scale interaction

Direct manipulation in VR breaks down if objects are much larger or smaller than the user. Poros allows the users to interact at a comfortable scale. By adjusting the size of the proxy and mark, users can make the objects within the proxy appear smaller or larger. For example, the users can easily rearrange furniture and other large objects by creating a mark spanning an entire room and linking it to a small proxy in front of them (see Figure 14-B). This functionality is similar to worlds in miniature [4, 34] and the *Voodoo Dolls* technique [28], while giving the user additional freedom over the scale and perspective. Alternatively, if the user is interacting with a tiny object then the mark and the proxy can be adjusted so that the objects within it appear larger. For example, the user can magnify a part of a book to read the small print (see Figure 14-C). This functionality supports low vision users similar to the tools proposed in *SeeingVR* [39], while keeping the magnified space fully interactive. When reading the book with small print, for example, users can flip pages and continue reading on the next one.

Users can also poke their head into such scaled proxies. This expands the proxy and allows the users to have a wider look at a distant marked space—effectively being teleported there while keeping the scale as it was in the proxy. This allows the user to feel like a giant or a dwarf within the scene and allowing perspectives and manipulations that are not possible at a normal scale [38, 39].

4.3 Observing Spaces and Oneself

Users often need to monitor or observe out of sight places. For example, consider a security guard monitoring a building. Such tasks commonly involve more than one space to be monitored, such as when several rooms are watched at once. Poros enables users to easily build ensembles of proxies to watch several distant spaces simultaneously. Figure 14-D shows a proxy configuration that allows the user to monitor different parts of a library. As the space within the proxy is fully interactive the user can quickly act on the remote space when needed (e.g., to close an open window). Users can also anchor the proxies to themselves to take them along as they move around the scene.

The ability to anchor proxies and marks also enables monitoring of moving objects in the scene. For example, when anchoring a mark to a cat, this allows the user to constantly monitor what the cat is doing and to manipulate its surrounding (e.g., cleaning up a mess the cat made). To observe themselves the users can anchor a mark to themselves For example, if the user anchors a room-sized mark and a top-down view proxy to themselves, they essentially create a mini-map centered around them (see Figure 14-E). Such anchoring could also be used to show users what is behind them all the time.

4.4 Multi-Perspective Object Manipulation

When manipulating objects, only seeing them from one perspective can hide important details and limit interaction. To help with this, many VR painting and modeling application enable change of perspective to more easily draw or manipulate objects [2][3]. Furthermore, modeling software like Blender[4] allows multiple-perspectives of the same object. With Poros, users can easily achieve both. To change the perspective they can transform the proxy, and to create a multi-perspective view on the object they can duplicate proxies and manipulate those to achieve the desired arrangement of perspectives. Figure 14-F shows a multi-perspective view of a sailing ship, allowing the user to quickly interact from multiple perspectives.

4.5 Replicated Interactions

Some tasks require repetition, such as filling the bowls of several pets or opening all windows in a room. To facilitate such tasks, Poros allows for the replication of interactions. After merging proxies, one proxy space is connected to multiple marked spaces and hence any user action is replicated across them. As an example, Figure 14-G shows how one proxy is linked to several marks around a series of lamps. Users can reach into the proxy and pull on the cord switch to turn on all lamps at once. For this to work, alignment of the different cords is necessary and thus users could first trigger an *align within* operation on the proxy. In aligned spaces, switching on one lamp is no different than switching on any number of lamps. Similarly, users could move multiple objects at once, or fill the bowls mentioned above.

4.6 Searching Spaces

To manipulate objects, users first need to find them, which can be time-consuming in complex scenes. For example, consider a kitchen, library, or archive, which all contain many similar objects. The highlighting operation in Poros can help users find the objects they want to interact with. Proxies here act similar to Perlin and Fox's *portal filters* [25]. In Poros, however, the view is not just changed inside a proxy, but instead the scene itself shows the highlighting. This allows users to also see results in the larger context of the scene.

4.7 Organizing Shelves

In scenes that contain many items, organizing those can be an important task. For example, a user could desire to move all shirts to the other side of a store. When many objects need to be moved between places far apart, this necessitates a substantial amount of locomotion from the user. With Poros, users can create a workspace that is optimized for this task. For example, a user would create two proxies, linked to two different shelves. The shelves can be far apart, but the proxies can be arranged next to each other. By making use of the *aligning for convenience* operation, users can then also have the two proxies rotate to make moving items from one to the other easier. This results in a setup where no locomotion is required and users can move objects with little effort. Furthermore, this could be combined with anchoring to take a proxy along or place items on a moving target.

5 EXPERT EVALUATION

Poros enables a variety of interactions, therefore there is no single performance metric that would have covered them all. Furthermore, there are distinct trade-offs for different tasks, making comparison across the wide set of scenarios problematic. Thus, rather than conducting an empirical evaluation comparing a component of Poros to another technique, we opted for an expert evaluation focused on conceptual understanding, breadth, and overall experience. Hence, a limitation of this study is that we are not able to draw any conclusions about comparisons to possible alternative techniques in each scenario. However, the experts do hint towards comparable techniques which could be useful for future empirical evaluations.

We invited nine experienced VR developers and researchers (with at least 2 and on average 5.7 years of VR experience) to our lab. They watched a video introducing Poros beforehand and then each had 30 minutes to work with the system in four scenarios: (1) large and (2) small scale interactions (per Section 4.2), (3) proxy and mark creation and manipulation (as in the first part of Section 4.7), and (4) proxy duplication (Section 4.4). The scenarios were identical to the examples described earlier and exposed the experts to a wide range of uses.

[2]Tilt Brush, https://www.tiltbrush.com/
[3]Blocks, https://arvr.google.com/blocks/
[4]https://www.blender.org/

Throughout the evaluation, we asked the participants to talk about their thoughts and actions. At the end, we also interviewed them using open-ended questions. We audio recorded each evaluation and transcribed the recordings. For the analysis we group participant quotes by themes and report on each theme below.

5.1 Concept

Generally, most users "got the concept pretty quickly" (P1). As P2 noted, "when [the proxy] is placed there, and you can just sort of put your head in it, then I think it is very easy to understand what is going on. Like that aspect of it is very good. I didn't really need an introduction" (P2). Some likened it to other concepts: "was it Super Mario or Crash Bandicoot that has spheres and then you go to those portals, like the old old PlayStation games" (P7).

The participants could also conceptually distinguish Poros from teleportation: "So I see it as like an alternative to teleporting around" (P2). They also saw clear advantages of Poros over teleportation "In that way [in reference to a task which requires moving between places frequently], if you teleport around, it's a lot of teleportation" (P1). Furthermore, P8 identified another difference to locomotion techniques: "and you can move large objects, which you wouldn't be able to, if you use locomotion to get to that thing".

From a short demonstration, participants were able to easily understand the concept visually, and also see the differences and advantages when compared to more traditional locomotion techniques such as teleportation.

5.2 Learnability and Consistency Across Interactions

Some experts noted that the "controls require a fair amount of memorization" (P1), but that "once you get familiar with it, I think it's very nice to go from one space to another space" (P5).

Some of the interactions were perceived to be natural: "it's so one to one what is going on so you don't have to learn it" (P2). On the other hand, experts felt that the interactions were inconsistent. For instance, P6 stated that "the biggest thing [usability flaw] was when you create the sphere you do this, but then you want to change its size afterwards you have to do something else".

P3 suggested manipulating the two spaces in the same way (by grabbing them with fists) especially for smaller marked spaces: "So I think for small scale things, that would be super useful to be able to just pick it up, and then stand and look and move it around. If I'm inspecting something very small, instead of having the controls inside". Hence, by making the interactions more consistent, learnability could be improved.

5.3 Additional Feedback

Participants noted that they "miss extra feedback. Like some sort of a haptic or sound when you do something" (P5). P8 also mentioned haptic feedback that could ease the difficulty in knowing when the hands are in contact with objects: "One thing I'm missing is like haptic feedback when you touch the spheres, so when you know that, okay, now it's touching outside — or inside". Future versions of Poros should include more feedback to let the user know when actions are performed or which mode they are in. Ideally, haptics would help the user to know if they are touching the spheres.

5.4 Design Choices

In this implementation, the hand size from the users perspective is kept constant. One user noted that, while this is sometimes beneficial, "it's nice to have a large hand if you want to grab a couch, for instance" (P1) this may make some interactions difficult and proposes to have "the option to scale it because if I read a paper, for instance, it's nice to you know, grab the paper". Further exploration is necessary to understand the situations where certain hand sizes are beneficial and also to allow user control over this.

Some found it difficult to position the marks to exactly where they needed to be during creation. P5 mentioned "I think the marking goes too fast. Can it be more precise, a little bit slower?" P2 described an alternative to the gesture driven movement, but then suggested he "struggles with moving it and scaling it and all that was more a matter of hand tracking." Creation and manipulation seems difficult in certain situations and is difficult to cater for all, but there may be ways to let the user control parameters of this, or to change it automatically based on context.

5.5 Utility and Applications

Several experts commented on potential applications of Poros. One expert mentioned that it could be used as a "very detailed inventory system" (P2) that could replace traditional menu based inventories with icons, with spatial references to objects. This suggestion was echoed by P4: "most training apps use menus that you have in a hand that kind of gives you the power to do things or, or tools that are ideal for this specific use case". They specifically recalled a training app for fire investigations which could be fitting: "where you can be multiple people and you can walk around in a scanned, burned out building and place evidence, numbers and take photos. It's kind of like those tools".

P3 found parallels to real world interactions with the small-scale effect: "you can do fine grained manipulation, but with larger body movements, so I think it would be useful for that. . . . in that sense, it's not that different from what they do with like, robotic surgery work".

P1 mentioned how it could be useful to clean up a room and return items quickly to where they belong: " I clean it all up and that takes forever because everything is scattered and they go to different places. So I can definitely see like if I had it [Poros], it's a bit like putting six boxes in front of me books in this one, Lego in that one, dirty clothes in that one. . . ". Therefore, they used a series of boxes as a real life analogy of Poros, for moving objects to pre-designated spaces.

Although we did not investigate collaborate tasks, several experts commented on this: "You probably also use it collaboratively so that you're able to all of you share the perspective together, so I think especially for any challenge where we collaborate on something very, very small" (P2). P7 recalled a project where this could facilitate it either collaboratively/socially: "some people in Minecraft are trying to build the whole world. And if people are going to visit different parts of the world, different cities, then they can have a sphere and then they can just have a kind of multi tasking window in that virtual 3d space there and then see what other people are doing in another part of the world".

We finally had comments about VR replacing desktop interfaces: "As we move further towards virtual reality becoming more and more used . . . I can see this working as a really good workplace, maybe just as a desktop" (P9). There, Poros could be helpful "like a 3D room as some kind of menu and then being able to interact easily with everything in the room".

5.6 Summary

The expert evaluation showed that the concept was easy to grasp, clearly different from locomotion techniques, and perceived useful across a range of applications. Having established that the system was conceptually understood well and that these tasks can be completed, we can now directly compare Poros empirically (or variations of it) to other techniques for interacting with distant objects. Regarding usability, experts mentioned a need for higher consistency between actions and additional feedback for interactions. There is further exploration with respect to either intelligently adjusting parameters of hand size, creation and manipulation based on context, or to enable user control of these.

6 DISCUSSION

Poros tackles an inherent limitation of spatial interfaces: Interaction at a distance is cumbersome. Users commonly need to traverse space and can only be in one location at once. With Poros, users can rearrange space for a given task, ensuring that what they need is within reach, wherever it is. They can glance at one or more distant locations and move objects between them. Furthermore, proxies can provide more powerful tools than just access to distant space. What they encapsulate can be operated on—used to filter, scale, or align spaces. As we have shown, this enables a range of applications that are complicated or impossible with existing technologies.

Whether setting up the kind of workspace enabled by Poros is beneficial depends on the nature of the user's activity. For one-off interactions, or when users want to be at another location for a longer time, setting up a collection of proxies might be too costly. While there is a setup cost when using Poros, subsequent interactions with distant objects are essentially "free". This is in contrast to common locomotion techniques, where no setup is necessary, but each movement incurs a cost (e.g., due to a need to reorient). Just as selection/manipulation and travel techniques are generally considered distinct [7], we consider Poros and the latter complementary—each suited to distinct situations.

Poros is designed for VR, but also builds upon work in 2D user interfaces where surrogates [15] are commonly employed to modify "distant" objects. For example, with wall-sized interfaces users also need to walk to interact and hence techniques like *Frisbee* [12] have been developed. Conceptually, this is the 2D analogy to Poros: a local "telescope" through which one can see and interact with elements at a remote destination. One of the benefits of surrogates is that they allow for non-destructive re-configuration of space. For example, when creating a proxy to a bookshelf, the original scene is still kept intact for later interactions and other users.

Enabling users to configure their workspace to fit a task, Poros brings to VR what is common in other forms of interactive computing. As we have proxies that can be arranged to make moving books easier, desktop user interfaces make it easy to place windows next to each other, in order to allow for convenient drag and drop operations, and context switching. Poros hence shows that VR space can be made just as malleable as other interactive spaces.

6.1 Future Work

There are a range of possible extensions to Poros that could be explored. For example, we only allow users to observe themselves in marked spaces, yet not to manipulate their avatar (e.g., picking themselves up to move to a different location). While we enable users to replicate an action in several linked spaces, this currently requires good alignment. Instead, a future version could adjust the hands in each linked space to their specific context. For example, opening a window in one marked space could snap the hand to window nearby window handles in all the other spaces, alleviating the need for good alignment. In Poros, we have also decided to prevent object duplication. However, conceptually, a proxy space linked to multiple marked spaces would offer an opportunity to do just that. Placing a book into such a proxy could result in a different copy coming into existence in each linked marked space.

In this implementation, we have chosen to define spaces spherically. A more complex system could explore using shapes that adapt intelligently to content, either based on the outline of objects or perhaps based on context. Alternatively, users could be given finer control over the shape of marks and proxies.

7 CONCLUSION

We have described Poros, where users can manipulate the space around them through proxies and also perform a variety of interactions on these. Through combining these interactions, we enable many applications with varying complexities. These range from interactions that other systems have enabled individually, to new interactions such as replicating actions across multiple spaces, aligning spaces, and interacting through different perspectives. We have shown several examples of how Poros can be used for interactions that are hard to do or impossible with existing techniques. The source code for Poros is available and we hope to inspire follow-up techniques. With more work moving into VR, techniques like Poros will be important to allow users to reconfigure their workspaces get the most out of their use of VR.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement 648785).

REFERENCES

[1] Parastoo Abtahi, Mar Gonzalez-Franco, Eyal Ofek, and Anthony Steed. 2019. I'm a Giant: Walking in Large Virtual Environments at High Speed Gains. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300752

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

[3] Hrvoje Benko and Steven Feiner. 2007. Balloon selection: A multi-finger technique for accurate low-fatigue 3d selection. In *2007 IEEE symposium on 3D user interfaces*. IEEE.

[4] Andrew Bluff. 2019. Don't Panic: Recursive Interactions in a Miniature Metaworld. In *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry* (Brisbane, QLD, Australia) *(VRCAI '19)*. Association for Computing Machinery, New York, NY, USA, Article 33, 9 pages. https://doi.org/10.1145/3359997.3365682

[5] Doug A. Bowman and Larry F. Hodges. 1997. An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (Providence, Rhode Island, USA) *(I3D '97)*. ACM, New York, NY, USA, 35–ff. https://doi.org/10.1145/253284.253301

[6] Doug A. Bowman, David Koller, and Larry F. Hodges. 1997. Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. In *Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97) (VRAIS '97)*. IEEE Computer Society, USA, 45.

[7] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. 2001. An Introduction to 3-D User Interface Design. *Presence: Teleoperators and Virtual Environments* 10, 1 (2001), 96–108. https://doi.org/10.1162/105474601750182342

[8] Han Joo Chae, Jeong-in Hwang, and Jinwook Seo. 2018. Wall-based Space Manipulation Technique for Efficient Placement of Distant Objects in Augmented Reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. ACM, New York, NY, USA, 45–52. https://doi.org/10.1145/3242587.3242631

[9] Niklas Elmqvist. 2005. BalloonProbe: Reducing Occlusion in 3D Using Interactive Space Distortion. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (Monterey, CA, USA) *(VRST '05)*. Association for Computing Machinery, New York, NY, USA, 134–137. https://doi.org/10.1145/1101616.1101643

[10] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. 1996. Aperture based selection for immersive virtual environments. In *ACM Symposium on User Interface Software and Technology*. Citeseer, 95–96.

[11] Wolfgang Kabsch. 1976. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A* 32, 5 (Sep 1976), 922–923. https://doi.org/10.1107/S0567739476001873

[12] Azam Khan, George Fitzmaurice, Don Almeida, Nicolas Burtnyk, and Gordon Kurtenbach. 2004. A Remote Control Interface for Large Displays. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (Santa Fe, NM, USA) *(UIST '04)*. Association for Computing Machinery, New York, NY, USA, 127–136. https://doi.org/10.1145/1029632.1029655

[13] Ioannis Kotziampasis, Nathan Sidwell, and Alan Chalmers. 2003. Portals: Increasing Visibility in Virtual Worlds. In *Proceedings of the 19th Spring Conference on Computer Graphics* (Budmerice, Slovakia) *(SCCG '03)*. Association for Computing Machinery, New York, NY, USA, 257–261. https://doi.org/10.1145/984952.984995

[14] André Kunert, Alexander Kulik, Stephan Beck, and Bernd Froehlich. 2014. Photoportals: Shared References in Space and Time. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Baltimore, Maryland, USA) *(CSCW '14)*. Association for Computing Machinery, New York, NY, USA, 1388–1399. https://doi.org/10.1145/2531602.2531727

[15] Bum chul Kwon, Waqas Javed, Niklas Elmqvist, and Ji Soo Yi. 2011. Direct Manipulation through Surrogate Objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 627–636. https://doi.org/10.1145/1978942.1979033

[16] H. Lee, S. Noh, and W. Woo. 2017. TunnelSlice: Freehand Subspace Acquisition Using an Egocentric Tunnel for Wearable Augmented Reality. *IEEE Transactions on Human-Machine Systems* 47, 1 (Feb 2017), 128–139. https://doi.org/10.1109/THMS.2016.2611821

[17] Klemen Lilija, Henning Pohl, Sebastian Boring, and Kasper Hornbæk. 2019. Augmented Reality Views for Occluded Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 446, 12 pages. https://doi.org/10.1145/3290605.3300676

[18] James Liu, Hirav Parekh, Majed Al-Zayer, and Eelke Folmer. 2018. Increasing Walking in VR Using Redirected Teleportation. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 521–529. https://doi.org/10.1145/3242587.3242601

[19] Daniel Mendes, Daniel Medeiros, Eduardo Cordeiro, Maurício Sousa, Alfredo Ferreira, and Joaquim Jorge. 2017. PRECIOUS! Out-of-reach selection using iterative refinement in VR. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 237–238.

[20] Mark R. Mine, Frederick P. Brooks, and Carlo H. Sequin. 1997. Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 19–26. https://doi.org/10.1145/258734.258747

[21] Roberto A. Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. 2017. Erg-O: Ergonomic Optimization of Immersive Virtual Environments. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 759–771. https://doi.org/10.1145/3126594.3126605

[22] Alex Olwal and Steven Feiner. 2003. The Flexible Pointer: An Interaction Technique for Selection in Augmented and Virtual Reality. In *Proc. UIST'03*. 81–82.

[23] Randy Pausch, Tommy Burnette, Dan Brockway, and Michael E. Weiblen. 1995. Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 399–400. https://doi.org/10.1145/218380.218495

[24] Randy Pausch, Dennis Proffitt, and George Williams. 1997. Quantifying Immersion in Virtual Reality. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 13–18. https://doi.org/10.1145/258734.258744

[25] Ken Perlin and David Fox. 1993. Pad: An Alternative Approach to the Computer Interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA) *(SIGGRAPH '93)*. Association for Computing Machinery, New York, NY, USA, 57–64. https://doi.org/10.1145/166117.166125

[26] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze + Pinch Interaction in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton, United Kingdom) *(SUI '17)*. Association for Computing Machinery, New York, NY, USA, 99–108. https://doi.org/10.1145/3131277.3132180

[27] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. 1997. Image plane interaction techniques in 3D immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*. ACM, 39–ff.

[28] J. S. Pierce and R. Pausch. 2004. Navigation with place representations and visible landmarks. In *IEEE Virtual Reality 2004*. 173–288. https://doi.org/10.1109/VR.2004.1310071

[29] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. 1999. Voodoo Dolls: Seamless Interaction at Multiple Scales in Virtual Environments. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics* (Atlanta, Georgia, USA) *(I3D '99)*. Association for Computing Machinery, New York, NY, USA, 141–145. https://doi.org/10.1145/300523.300540

[30] Thammathip Piumsomboon, Gun A. Lee, Andrew Irlitti, Barrett Ens, Bruce H. Thomas, and Mark Billinghurst. 2019. On the Shoulder of the Giant: A Multi-Scale Mixed Reality Collaboration with 360 Video Sharing and Tangible Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–17. https://doi.org/10.1145/3290605.3300458

[31] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *ACM Symposium on User Interface Software and Technology*. Citeseer, 79–80.

[32] Christian Sandor, Arindam Dey, Andrew Cunningham, Sebastien Barbier, Ulrich Eck, Donald Urquhart, Michael R. Marner, Graeme Jarvis, and Sang Rhee. 2010. Egocentric space-distorting visualizations for rapid environment exploration in mobile mixed reality. In *2010 IEEE Virtual Reality Conference (VR)*. 47–50. https://doi.org/10.1109/VR.2010.5444815

[33] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. 2006. Object selection in virtual environments using an improved virtual pointer metaphor. In *Computer Vision and Graphics*. Springer, 320–326.

[34] Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual reality on a WIM: interactive worlds in miniature. In *CHI*, Vol. 95. 265–272.

[35] Stanislav L. Stoev and Dieter Schmalstieg. 2002. Application and Taxonomy of Through-the-Lens Techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (Hong Kong, China) *(VRST '02)*. Association for Computing Machinery, New York, NY, USA, 57–64. https://doi.org/10.1145/585740.585751

[36] R. Trueba, C. Andujar, and F. Argelaguet. 2009. Multi-Scale Manipulation in Indoor Scenes with the World in Miniature Metaphor. In *Proceedings of the 15th Joint Virtual Reality Eurographics Conference on Virtual Environments* (Lyon, France) *(JVRC'09)*. Eurographics Association, Goslar, DEU, 93–100.

[37] Hans Peter Wyss, Roland Blach, and Matthias Bues. 2006. iSith-Intersection-based spatial interaction for two hands. In *3D User Interfaces (3DUI'06)*. IEEE, 59–61.

[38] Haijun Xia, Sebastian Herscher, Ken Perlin, and Daniel Wigdor. 2018. Spacetime: Enabling Fluid Individual and Collaborative Editing in Virtual Reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 853–866. https://doi.org/10.1145/3242587.3242597

[39] Yuhang Zhao, Edward Cutrell, Christian Holz, Meredith Ringel Morris, Eyal Ofek, and Andrew D. Wilson. 2019. SeeingVR: A Set of Tools to Make Virtual Reality More Accessible to People with Low Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 111, 14 pages. https://doi.org/10.1145/3290605.3300341