Application and Taxonomy of Through-The-Lens Techniques

Stanislav L. Stoev
Egisys AG
stanislav.stoev@egisys.de

Dieter Schmalstieg
Vienna University of Technology
dieter@cg.tuwien.ac.at

ABSTRACT

In this work, we present a set of tools based on the through-the-lens metaphor. This metaphor enables simultaneous exploration of a virtual world from two different viewpoints. The one is used to display the surrounding environment and represents the user, the other is interactively manipulated and the resulting images are displayed in a dedicated window. We discuss in detail the various different states of the two viewpoints and the two synthetic worlds, introducing taxonomy for their relationship to each other. We also elaborate on navigation with the through-the-lens concept extending the ideas behind known tools. Furthermore, we also present a new remote object manipulation technique based on the through-the-lens concept.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture/Image GenerationViewing algorithms; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and RealismVirtual reality; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

Design, Human Factors

Keywords

Virtual Environment Interaction; Virtual Reality; Interaction; Data Manipulation; Visualization Techniques; Human-Computer Interface; Interaction Techniques;

1. INTRODUCTION

The main step towards letting the participant in a virtual reality application feel the virtual surrounding as real as possible is the interaction with it. The interaction can be divided in two main categories: the navigation through the synthetic world and the object manipulation in it. The growing virtual worlds make tools for adequate navigation indispensable in today's virtual reality applications. They define the acceptance and the usability of the latter and therefore have to be easy to use, while powerful and enable accomplishment of various kinds of task-adapted navigation through the synthetic world.

On the other hand, each virtual reality application is as interactive as the supported tools for manipulation of objects in it. Regarding this manipulation, we distinguish two different categories: (a) the manipulation of objects close to the user, i.e. in its physical hand reach and (b) the remote object manipulation (ROM). The remote object manipulation is an important feature, especially for interacting with large scenes. Instead of navigating through the scene manipulating the target object and navigating back to the original position in order to examine the results, ROM techniques make it possible to directly manipulate the desired objects, while examining the result of the performed actions from the current user viewpoint.

In the remainder of this work, we elaborate on the navigation in virtual worlds and on the remote object manipulation. In particular, we present a concept for displaying the surrounding world seen from an interactively defined viewpoint in a dedicated window. This simultaneous view makes it possible to explore a copy of the surrounding world and navigate through it using the introduced window, while staying at the same location in the full size view. We first introduce the various possible configuration of the window on which the scene as seen from the additional viewpoint is displayed. Afterwards, we discuss the relationship between the full size view and the scene behind the window.

2. RELATED WORK

There have been various published contributions in the fields of navigation and remote object manipulation. In this section we will review the once relevant to our work in each of the two categories: navigation and remote object manipulation.

2.1 Navigation in VEs

Like Andries van Dam and coauthors [20], we divide the navigation into three groups of techniques:

- *Searching* is the motion to a particular location in the virtual environment.
- *Exploration* is defined as navigation without particular target.
- *Maneuvering* is the high-precision adjustment of the user position in order to perform other tasks.

Besides the application of these navigation techniques for performing particular tasks, each of them has a different application range.

Searching and exploration techniques are utilized for overcoming large distances, while maneuvering is applied rather locally.

Fortunately, people are often confronted with the counterpart of the navigation problem in every day life, which facilitates the exploration of the subject. For instance, Darken and Sibert [6] presented a toolkit for navigation applying principles from real world navigation aids (e.g. maps). They also compare the strengths and weaknesses of such aids. Stoakley et al. [17] extended this work to three-dimensional maps introducing the *World-in-Miniature* or WIM-technique.

Originally, the WIM was applied for interaction in virtual worlds, i.e. manipulating objects in space. Pausch et al. [11] extended this approach to provide a navigation tool for accomplishing searching and exploration tasks, enabling the user to directly manipulate the current viewpoint. For this, they utilize a doll representing the user in the miniaturized world. However, they also reported that despite the intuitive application of the WIM, the direct viewpoint manipulation was confusing to many users.

Another work discussing and comparing navigation tools was presented by Ware and Osborne [23]. They describe and evaluate three navigation metaphors: the "flying vehicle control", "eyeball-in-hand", and "scene-in-hand", concluding that: "None of the techniques is judged the best in all situations, rather the different metaphors each have advantages and disadvantages depending on the particular task". Similarly to the WIM-technique, the main problem with the *eyeball-in-hand* and the *scene-in-hand* techniques is that the viewpoint is directly manipulated and the resulting image immediately displayed. This, however, often leads to confusion of the user or may even cause loss of orientation.

A more detailed analysis of navigation considering its basic components: direction selection, velocity selection, and input conditions is discussed in [4]. In their work, Bowman et al. introduce a taxonomy for viewpoint motion in virtual environments. They discuss experiments showing that "pointing"-based travel techniques are advantageous compared to "gaze-directed" steering techniques. In addition, they found out that the instant user teleport is correlated with increased user disorientation. This cognition is closely related to the techniques described in the remainder of this paper. Instead of teleporting the user, we offer a sort of preview window, through which the location seen through the window can be entered. Similar techniques for entering a world through a window are discussed in [12].

2.2 Remote Object Manipulation

Remote object manipulation allows the user to work with objects not within the reach of his/her hand and to examine the virtual world as seen from the current viewing position. Many researchers have addressed the subject of remote object manipulation in virtual environments. Pierce et al. [13] presented the *Voodoo Dolls*-technique for remote object manipulation. A doll looks like a minified (or magnified) copy of an object. The user creates a doll by framing an object with her/his hand on the image plane and pinching his/her fingers together. The system then instantaneously creates a copy of the object, scales it so that the new doll reaches a comfortable working size, and moves the object to the user's hand.

Mine et al. [10] presented another approach for remote object manipulation: the scaled-world grab. The basic idea of this technique is to automatically scale objects in such a way, that their projected size remains unchanged, while bringing them close to the user. He/she can now manipulate them as if they were in the hand's reach. After the manipulation is completed and the object released, it is scaled back to its original location.

Poupyrev et al. [14] described the go-go mechanism for nonlinearly extending the arm of the user, thus, enabling manipulation of objects out of the reach of the user's physical hand. This metaphor provides the user with the traditional one-to-one mapping of the translation of the tracked device to the virtual hand within given application radius. Outside this area, the mapping extends the virtual hand applying a quadratic increase of the arm extension.

Bowman and Hodges [3] gave a brief evaluation of this and other existing techniques for grabbing and manipulating objects at remote locations. In their work, they report on a user study and compare the go-go-technique, other arm extension techniques, and a ray-casting technique [9]. The authors also propose the *HOMER* technique, which carries out a combination of the ray-casting technique for object selection and in-hand object manipulation. The paper concludes that none of the tested techniques is a clear favorite, because none of them were easy to use and efficient throughout the entire interaction consisting of grabbing, manipulating, and releasing.

Pierce et al. [12] presents a set of image plane techniques, which enable selection, manipulation, and navigation in virtual environments. Their idea is to work not with the objects, but with their projections onto the image plane.

Finally, as stated before, the WIM technique [17] can also be used to remotely manipulate objects in the space. The user can grab, manipulate and release the objects in the miniaturized world, which are linked with the full size world and its objects.

3. THROUGH-THE-LENS CONCEPT

The main idea of a *through-the-lens*-tool is to provide a viewpoint, additional to the one used to display the surrounding scene[1]. Afterwards, the scene as seen from this viewpoint is shown in a dedicated *output window* W_o (as shown in Figures 1, 2 and 3). In other words, we assume that there is a copy of the full size synthetic world, existing simultaneously with the surrounding world in the physical space. The user is surrounded by one of these worlds, called the *primary* world. He/she is represented by the *primary* viewpoint in the primary world. The *secondary world* is the copy of the primary world that can be viewed only through a sort of *magic lens* [2, 21] in the primary world (Figure 1). It displays images seen by a virtual camera in the secondary world.

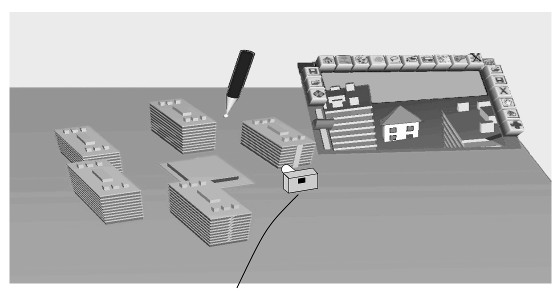

Figure 1: The primary world surrounds the user, while the secondary world can be explored only through a window in the primary world. The house visible in the secondary world exists in the primary world as well. However, it is not visible from the viewpoint in the primary world.

In contrast to [18], where only the navigation in virtual worlds based on the through-the-lens metaphor is described, here we will

[1] This term was first used in [7], who proposed a system for camera control based on the features seen through the virtual camera. The authors do not involve any interaction techniques.

provide a detailed discussion on the taxonomy and the application of TTL-tools in general. We will also show how some well-known tools can be derived from the through-the-lens concept, applying various restrictions to the viewpoint motion or the relationship between the primary and the secondary world.

Conceptually, there are two copies of the explored synthetic world and thus two windows, one in each of these worlds. The window in the primary world, through which the user views the secondary world, we call *output window* (W_o). The virtual counterpart of this window in the secondary world we call *viewing window* (W_v). Although these windows are attached to each other, for clarity we will use these two different terms throughout this work, depending on the world we refer to (see Figure 3).

3.1 Taxonomy for the States of the Two Worlds

The primary and the secondary world, as well as the two viewpoints in each of these worlds may have different relations to each other. Before we introduce the tools based on the through-the-lens concept, we will identify these various configurations and give a short example for their application.

3.1.1 W_o in the Primary World

Let us first consider the output window W_o, hence, the window in the primary world. W_o can have three different states in the primary world (as shown in Figure 2):

- (case O1) fixed in the primary world;
- (case O2) fixed in the image plane of the user;
- (case O3) mapped onto a pad, held by the user.

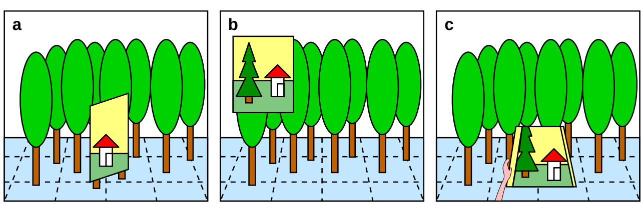

Figure 2: (a)-(c) display the states O1-O3 respectively.

In the first case O1, the window is only visible when viewed from the appropriate direction. Since it is fixed in the primary world, the user cannot move it. Changes of the user position in the primary world allow viewing the virtual world behind the window from different angles.

In the second case O2, the window is fixed within the image plane of the viewer. This means, that when the user moves his/her viewpoint, the window remains at the same position in the image plane.

Finally, for the realization of the last case O3, we utilized the *Personal Interaction Panel* (*PIP*) concept [19, 15]. The PIP consists of a tracked palette, on which the virtual tools are displayed is such a way, that the user sees them on the pad's surface (see Figure 1). In contrast to the first two scenarios, where the window W_o is fixed, in this case the pad, thus the window mapped on it, can be freely moved within the primary space.

3.1.2 States of the Secondary World

These were the possible states of the output window in the primary world. Regarding the additional viewpoint and the scene seen through it, there are also three conceptually different states of the *viewing window* W_v and the secondary world seen through it:

- (case V1) the secondary world is fixed in the primary world's space;
- (case V2) the secondary world is fixed with respect to the viewing window;
- (case V3) the secondary world is fixed with respect to the primary viewpoint.

In the first case V1, the coordinate systems of the two worlds are fixed with respect to each other. The window connecting them can be positioned arbitrarily in the primary space. Depending on the position of the window, different areas of the secondary world are visible.

In contrast, in the second scenario (V2), the secondary world is fixed in the windows coordinate space. This means, that independent of the position of the window, the observer views always the same location of the secondary world behind the window. Looking at the window from different viewing angles enables exploration of different areas behind it.

Finally, in the third case V3, the secondary world is fixed with respect to the primary viewpoint in the primary world. In other words, independent of the *position* and the *orientation* of the output window and the primary viewpoint in the primary world, the area of the secondary world seen through the window remains unchanged.

3.2 Combinations of O1-O3 with V1-V3

Each of the states O1-O3 can be combined with each of the states V1-V3. In this section, we will describe each of these scenarios and give to each of them a short example.

3.2.1 W_o Fixed in the Primary World

If the output window W_o is fixed in the primary space (case O1), the secondary world and the primary world are fixed with respect to each other. Thus, we cannot distinguish between the cases V1 and V2. This scenario was first described in [16], where the window is used for sewing two different virtual worlds together. Once this is done, the user can travel from one world to the other by moving the viewpoint through the provided window.

In contrast, in case (O1/V3) the secondary world moves with the user motion in the primary world. This means, that the position of the secondary viewpoint in the secondary world remains unchanged, when the primary viewpoint is moved in the primary world. Thus, when the user moves in a particular direction, different parts of the secondary world can be examined with the output window W_o, which is static in the primary world.

3.2.2 W_o Fixed in the Image Plane

When W_o is fixed in the viewing frustum of the user (case O2), two cases O2/V1 and O2/V23 are theoretically possible. In the first case (O2/V1), when the primary viewpoint is moved in the primary world, the output window and thus the viewing window move with the image plane and the user sees different parts of the secondary world. This corresponds to moving the viewpoint in both worlds simultaneously. This scenario is often applied in semi-transparent head mounted display systems, where the primary world is the physical world surrounding the user. The secondary world seen through the HMD is a virtual world, allowing for superimposing information aligned with the primary world.

The second case (O2/V23) is rarely used, since a secondary world fixed with respect to the viewing window would result in displaying always the same image independent of the viewing direction and position of the viewer in the primary world. The scenario in which this feature may be useful is when the user intends to "keep an eye" on a given location in the secondary world.

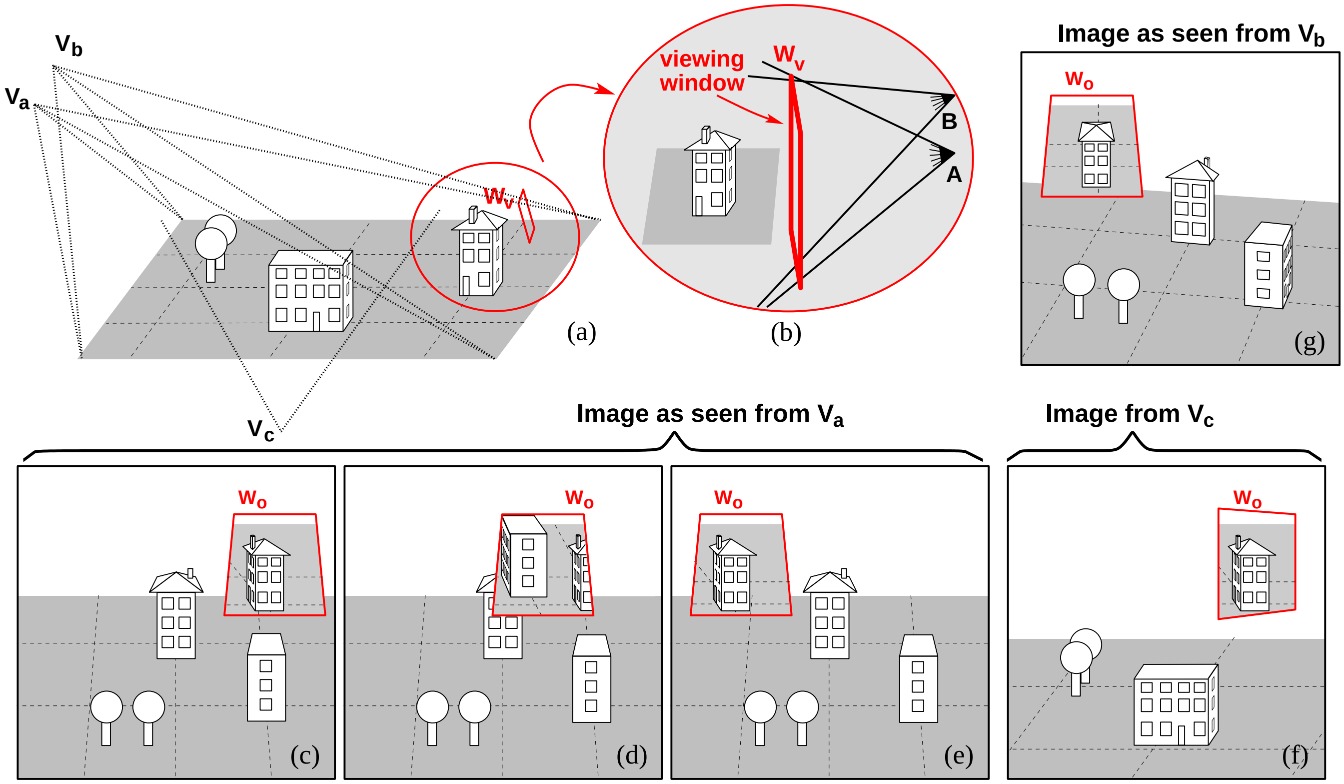

Figure 3: The position of the viewing window (W_v) is shown with respect to the scene seen through it (a). V_a and V_b are two different viewing positions. (b) shows the two viewing positions A and B, derived from the current camera positions V_a and V_b. In case the viewing window W_v is fixed in the secondary scene and the output window W_o is moved in the primary scene, the secondary scene moves with the viewing window as shown in (c) and (e). (d) illustrates the "scene fixed in space" scenario (O3/V1) (compare to (c)). Moving W_o allows viewing different parts of the scene (compare (c) and (d)). Detaching the secondary viewpoint from the primary, allows the user to travel the primary world, while staying at the same position in the secondary world (compare (c) and (f)). When the viewpoint changes (e.g. from V_a to V_b), the scene shown in W_o can be viewed from different angles as depicted in (e) and (g).

3.2.3 W_o Mapped on the Pad

The most interesting case is case O3, in which we are able to interactively position the output window in the primary world and thus the viewing window in the secondary world.

In case O3/V1, the output window W_o mapped on the pad is used to explore parts of the secondary world that is fixed in the primary world's space (see Figure 3(c) and (d)). To this category belong magic lens-like tools [2, 21]. Inspired initially by this concept, we call the proposed metaphor the *Through-The-Lens*-metaphor.

In contrast, in the second scenario (O3/V2) the window can be adjusted to show a given part of the secondary scene (location of interest), such that even if the output window W_o is moved, the virtual window W_v remains fixed in the secondary world's space (see Figure 3 (c), (e), and (g)). In this way, a target location in the secondary world can be observed independent of the user's motion in the primary space (similar to cases O2/V23). In addition, in this scenario the user still can look at the world through the window from different angles.

Finally, case O3/V3 makes it possible to travel in the primary world, without applying any changes of the primary viewpoint to the secondary viewpoint in the secondary world. This is similar to the case where the secondary world is anchored to the viewing window (case O3/V2). However, unlike in case O3/V2, moving the output window in the primary world enables exploration of different parts of the secondary world, while looking at the window from different angles does not enable exploration of different areas in the secondary world. State O3/V3 is especially useful when the user travels the primary world and wants to keep the position of the secondary viewpoint unchanged in the secondary world (see Figure 3 (c) and (f)).

4. THROUGH-THE-LENS NAVIGATION

After introducing the different states of W_o and W_v within the primary and the secondary world, here we will address the adjustment of the secondary world in such a way that a particular target location can be viewed through the output/viewing window.

Various navigation techniques belonging to one or more of the navigation types introduced in Section 2.1 are reported in the literature. The navigation tools we present in this work are inspired by the *eyeball-in-hand*, *scene-in-hand* (which we call *grab-and-drag*) [23], and WIM-techniques [11], but attempt to overcome their limitations. We combine these tools with the above *through-the-lens* concept, extending the functionality and improving the usability of the original tools. In particular, we apply the manipulation described in the original techniques to the secondary viewpoint. Hence, the effect of the manipulation is observed through the window, rather than applying direct transformation of the primary viewpoint. In this way, the presented navigation aids provide a set of flexible and powerful tools, covering all of the navigation categories introduced in Section 2.1.

4.1 TTL Scene-In-Hand

The scene-in-hand technique was presented originally in [23]. It provides a handle attached to the scene, such that the translations and rotations of the handle are applied one-to-one to the scene. This technique is easy to understand and apply even for motions extending the hand's reach: a clutch button is used to attach and release the scene to/from the virtual handle. This approach has shown to be useful for manipulating discrete objects and changing the viewpoint of the user for scene exploration [22, 8].

We start with two aligned viewpoints, which correspond to two aligned (primary/secondary) synthetic worlds. The user is able to manipulate the scene seen from the secondary viewpoint by grabbing a point in it (an object or even the air) and dragging it in the desired direction (Figure 4). Note that the secondary scene remains

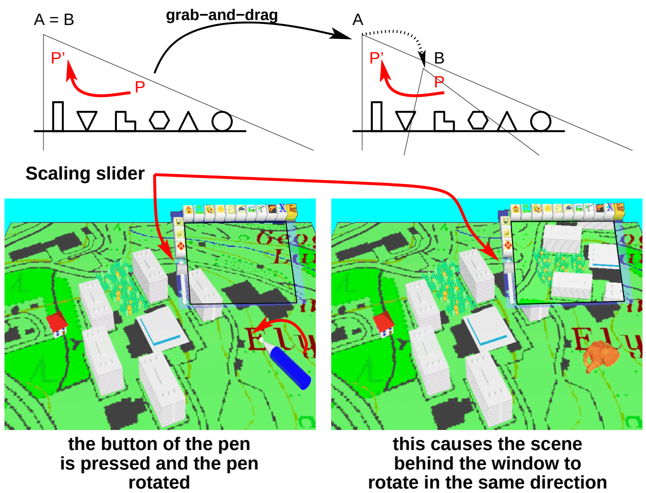

Figure 4: Initially, both viewpoints are aligned as shown on the left. Grabbing the scene at point P and dragging to point P' corresponds to a translation combined with rotation of viewpoint A to viewpoint B. (This figure is reproduced in color on page 216.)

always fixed with respect to the primary world (case O3/V1), except when it is grabbed. In this scenario, the secondary scene can be grabbed at any arbitrary location, using the second button of the interaction pen. In contrast to the original implementation [23], we did not fix the center of rotation to the center of the scene, since, as the authors point out, rotations are difficult to perform when the viewpoint is far from the fixed center of rotation. In this case, the translation and the rotation are mapped one-to-one to the secondary world.

This approach has some similarities with the scaled-world grab locomotion metaphor described by Mine et al. [10]. They propose a technique for grabbing distant objects using a form of image-plane interaction. Thus, the user can pull him/herself towards any visible object. Unlike the scaled-world grab where the authors apply the motion immediately, we (a) provide a preview window, (b) use a direct technique for grabbing and dragging the *secondary world*, and (c) do not require the user to grab an object, but allow any point in the space to be grabbed. The scene seen through the output window is now manipulated applying a simple grab-and-drag handle. Thus, the viewed part of the scene can be chosen very precisely.

In addition to the grab-and-drag mechanism, the user can also scale the secondary scene if needed (see slider in Figure 4), hence, making this tool especially suitable for final high-precision adjustment. Furthermore, the scaling facilitates the traveling of large distances. When the user intends to view a distant location he/she can scale down to secondary world, place the target location underneath the center of the output window using the grab-and-drag mechanism and scale the secondary world up again.

Nevertheless, when applied for viewing very distant locations, the proposed technique may be circumstantial. This drawback can be overcome by combining our through-the-lens technique with other techniques for remote object grabbing (e.g. go-go [14], image plane [12], or scaled-world [10] techniques).

4.2 TTL World-In-Miniature

Originally, the WIM metaphor was applied for remote object manipulation [17]. The miniaturized copy of the world is mapped onto a hand-held device. Pausch et al. [11] extended this concept to traveling in immersive environments. They found out, that the direct mapping of the manipulated user viewpoint icon in the miniaturized world to the full-scale virtual world causes disorientation.

In contrast to the original WIM tool, with the TTL-WIM we do not map the miniature copy of the virtual world *on top* of the pad. Instead, we display the latter underneath the pad's surface. In this way, we create the impression of looking into the miniaturized virtual world through a window defined by the pad on top of an imaginary box. Instead of explicitly defining the final position of the user in the miniaturized world, the user interactively selects a region of interest dragging a box around it. The selection is made on the top of the bounding box of the virtual world. The miniaturized world is scaled up in such a way, that the selection fills up the viewing window and the top of the bounding box is still aligned with the surface of the interaction pad (as shown in Figure 5).

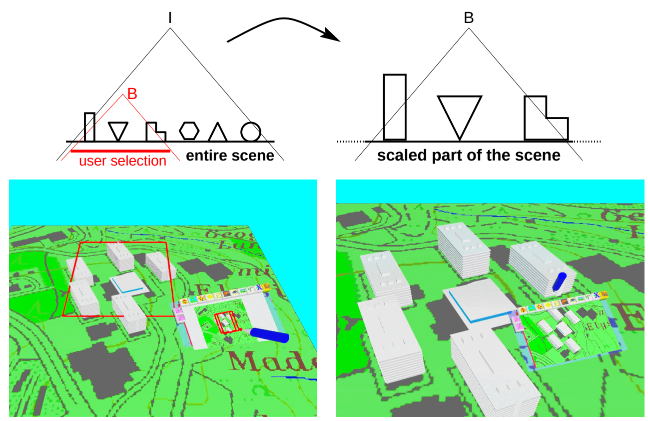

Figure 5: Initially, a miniaturized copy of the entire scene as seen from viewpoint I is displayed on the interaction pad. During the interactive selection of a region of interest, the selection is shown in the primary world as well. After completing the selection, the viewpoint is moved to B, such that only the selected region is visible through the pad. The lower right image shows the transformation applied to the current viewpoint in the primary world. (This figure is reproduced in color on page 216.)

The selected part can be examined not only on the pad, but also in the virtual world surrounding the user, as shown on the left of Figure 5. In this scenario, the viewing window W_v is always fixed in the secondary scene. Thus, it corresponds to case V2.

This technique is primarily used for coarse selection of the viewed area in very large virtual worlds. Once the user has adjusted the desired part of the scene to be seen through the output window on the pad, there are two ways of entering the new location: In the first scenario, the primary world is automatically scaled, whereas the view orientation in the primary world remains unchanged relative to the surrounding world (see Figure 5). In the second scenario, the secondary scene behind the window is released from the window, thus fixed in the space (case V1), and can be further adjusted applying another through-the-lens technique, or directly entered (see Section 4.4).

4.3 TTL Eyeball-In-Hand

This technique has been introduced and explored by several researchers [1, 5, 23]. The eyeball-in-hand originally uses a tracked device as a virtual camera that can be moved about the virtual scene. Thus, the participant sees on the screen what the camera sees through its lens.

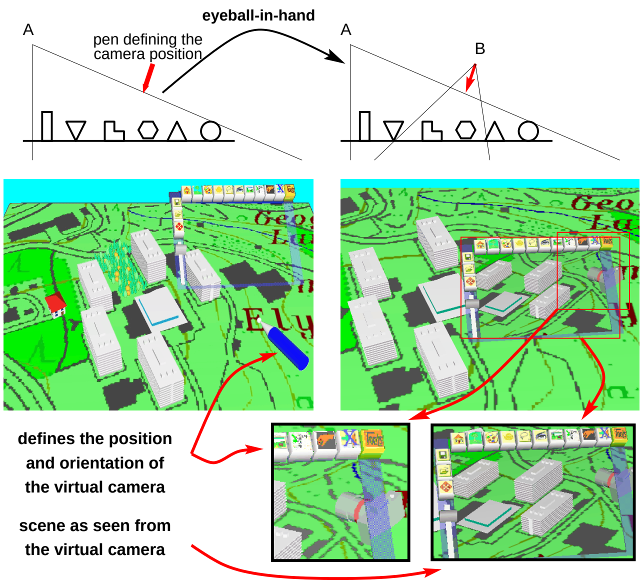

Figure 6: Applying the eyeball-in-hand tool, the secondary viewpoint can be positioned explicitely by defining a position and orientation of the virtual camera B. (This figure is reproduced in color on page 216.)

Despite the intuitive mental model applied with this metaphor, the main problem is the often-caused disorientation. Moreover, the one-to-one mapping of the hand to the virtual viewpoint makes precise adjustment of the virtual camera very hard. Even though, the eyeball-in-hand metaphor is simple to understand and requires a simple mental model of the scene, the above limitations make it unsuitable as a sole navigation technique. In order to circumvent these limitations, while still supporting the features of this metaphor, we introduced a preview window to the eyeball-in-hand technique. This makes it possible to view the scene from various viewing positions (in the hand's reach) without changing the current viewpoint of the user in the primary world, thus, reducing confusion and disorientation.

In our implementation, the pen held in the dominant hand is used to define the secondary viewpoint in the surrounding virtual environment (see Figure 6). The scene, seen from this viewpoint, is displayed in the output window, which is mapped on the interaction pad. Since the user sees the position of the virtual camera in the primary world (surrounding environment) and the scene as seen by the positioned camera *simultaneously*, the virtual camera can be positioned very precisely. In this way, our tool overcomes the limitations of the original eyeball-in-hand metaphor, while still supporting its features.

4.4 Entering the Secondary World

Once the adjustment of the additional viewpoint in the secondary world is accomplished, the new location can be entered, thus providing navigation capabilities. In order to enter the secondary world as seen from the additional viewpoint, the user has to move the pad towards her/his face until the window on the pad completely covers the viewing area[2]. Once this is done, the system automatically detects this action and sets the secondary viewpoint v_r to be the current viewpoint $v_p \leftarrow v_r$.

5. REMOTE OBJECT MANIPULATION

In general the remote object manipulation can be realized in two different ways, considering the underlying concept of the technique:

- (UR) the manipulated object (or an icon of it) is brought into the reach of the user's hand (User Reach - techniques);
- (ER) the manipulation tool is extended to reach the remote object (Extended Reach).

To the first set of techniques count the Voodoo Dolls [13], the WIM [11, 17], the scaled-world grab [10], and the image plane interaction [12]. Within this set, the techniques can be divided in two main categories:

- (a) projection plane techniques;
- (b) manipulation of copy of the target object.

The first category consists of techniques that make use of the projection of the object being manipulated. The second provides an appropriately scaled copy of the target object. This copy (icon) is linked with the original, in such a way, that actions performed on the icon are immediately applied to the original object.

The idea of the second category (ER) is to extend the physically limited reach of the user's hand. To this set count techniques like the go-go [14], the HOMER [3], and ray-casting [9] techniques.

The common feature of the techniques in the first category (UR) is that all of them support interaction with objects in the local environment. In contrast, with the second set of metaphors (ER), the manipulation is performed in the remote location. Unfortunately, none of the referenced techniques allows for spontaneous combination of both. This capability would make it possible to exploit the best features of both remote manipulation concepts simultaneously.

5.1 Direct TTL Manipulation of Remote Objects

What we would like to have is a tool, which allows working with the remote objects in their natural environment at a freely chosen scale. The through-the-lens remote object manipulation is an improvement allowing both modes, ER and UR, to be arbitrarily combined. The basic idea is to allow reaching through the window and manipulating the objects seen through it.

We have shown in Section 4 how the secondary world viewed through the output window can be adjusted such that a target location is viewed through it. In this way, a kind of preview window to

[2]Note, that the output window is moved and that the secondary world is fixed in space, thus the viewpoint "flies" through the window!

a remote location is provided, similar to a wormhole known from science fiction.

For the application of the remote object manipulation, we assume that the secondary world is fixed in the space, as discussed before. In this way, the pad becomes a magic lens revealing the remote location.

Once the secondary world is adjusted as desired and fixed in the space of the primary world, the window can be detached from the surface of the interaction pad. Decoupling the window from the pad's surface allows projecting interaction tools on the latter and applying them as usual. This scenario corresponds to case V1/O1, namely, the secondary world and the output window are fixed with respect to the primary world's space.

On the other hand, if the window is not detached from the pad, the pad can be used to "browse" different areas at the remote location. If the aim of the remote object manipulation is adjustment of the position and orientation of an object, this scenario may be even preferred compared to detaching the window from the interaction pad.

After accomplishing the adjustment of the viewpoint in the secondary world, the tracked stylus is used to interact with the remote objects. The user can manipulate remote objects by reaching with the stylus into the frustum volume defined by the lens and the current viewpoint (see Figure 7). If the stylus is outside this volume, it

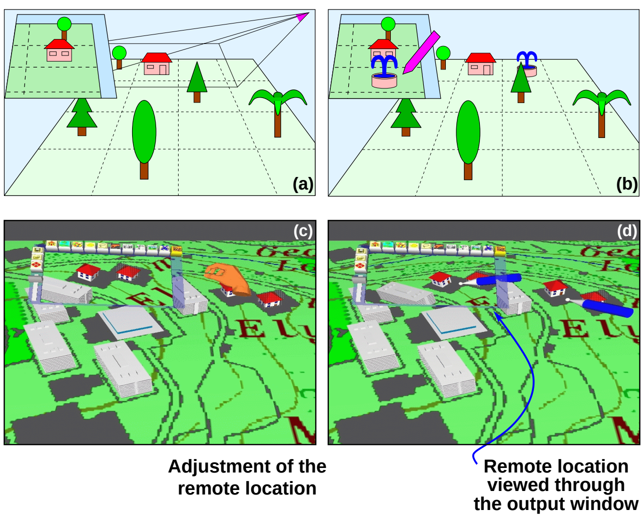

Figure 7: (a) and (b) show a sketch of the remote object manipulation. The left sketch shows the output window and the remote location in the primary world, while the right shows an object (fountain) added through the lens at the remote location. The snapshots (c) and (d) show the proposed technique in action. After defining a window to the secondary world, the user can move objects at the remote location. In (d), the pen is visible in both worlds. (This figure is reproduced in color on page 216.)

acts in the local environment in the normal way. Moving the stylus from the remote volume to the local volume and vice versa instantly changes the context of interaction (see Figure 7 (d)).

This scenario enables the user to select an object at the remote location and change its properties. Furthermore, the proposed tool can be used to rotate and translate the object at its *original* position. Since the secondary world can be viewed at an arbitrary scale, the remote objects can be moved with high precision at any desired scale size.

5.2 TTL Remote Drag-And-Drop

The change of context applied when the stylus is moved can be exploited to teleport objects between locations by drag-and-drop operations between volumes. As soon as the interaction pen and an object picked with it leave the view volume described above, the object is dragged to the primary world (the test is performed for the tip of the stylus). When the manipulation is completed, it may be put back to its original location.

In a slightly more complex scenario, objects can be even transferred between *multiple* remote locations with drag and drop operations. In this way, the user can assemble a complex scene with arbitrary fine details without having to change his/her position in the primary world, while still having a tool for examining the scene from different viewing positions. Thus, our approach provides a solution to the problem of changing and examining the scene from the current viewpoint, while manipulating objects in distant locations of the virtual world.

6. USABILITY

Event though we have not performed detailed quantitative usability studies yet, preliminary qualitative evaluation of interactive sessions with a virtual world assembly application have shown that the TTL grab-and-drag and the TTL WIM tools are intuitive and do not require training time in order to apply them appropriately. In contrast, the eyeball-in-hand tool turned out to be confusing for many users due to the 6DOF manipulation (see Table 1).

Considering the TTL-remote object manipulation, we also found out that once the secondary world is adjusted appropriately, the manipulation at the remote location is easy to perform. This is due to the fact that the applied tools behave like in the surrounding environment.

One of our future research directions will be the proof of usability of the proposed techniques, in which many users are envolved. Furthermore, they will have to compare different interaction and remote object manipulation techniques and judge about their applicabilaty when performing different tasks.

7. CONCLUSIONS

Although each of the proposed techniques has some limitations, the combination of all of them provides a powerful toolkit for exploring distant locations in a virtual world, as well as navigating in virtual environments. The set of all proposed techniques allows for covering all navigation categories addressed in the introduction.

The application of the through-the-lens concept for navigation in virtual environments provides a powerful mechanism for implementing preview-enriched navigation tools. It allows viewing locations of interest, while still being at the same location in the primary virtual world. This main contribution of this work enables the enhancement of existing navigation aids and development of new tools exploiting the through-the-lens concept.

Additionally, the proposed through-the-lens technique was also applied for manipulating distant objects, while still at their original location. It provides an universal technique for working with objects out of the user's physical reach and proved to be a valuable tool for assembling virtual worlds, circumventing some of the disadvantages of other known remote manipulation metaphors. In this way, the user is not required to navigate to the remote location in order to manipulate objects, but can stay at the current location and examine the result of the remotely performed actions.

The proposed technique has shown in informal trials with experienced and novice users that it is very intuitive and easy to use. Although it does not have a counterpart in real life, we achieved

Technique	Features	Limitations
TTL grab-and-drag	• suitable for searching tasks, and precise final adjustment tasks • intuitive viewpoint manipulation	• circumstantial for distant objects and locations
TTL WIM	• suitable for exploration and searching tasks • supports multiple scale levels	• scene cannot be entered until not "fixed in space" • improper for fine manipulations
TTL eyeball-in-hand	• requires very simple mental model • easy to use for fine precision camera adjustment	• unsuitable for exploration • may be confusing (too many degrees of freedom)

Table 1: Comparison of the proposed navigation tools.

convincing performance results applying this remote manipulation concept.

8. REFERENCES

[1] N. I. Badler, K. H. Manoochehri, and D. Baraff. Multi-dimensional input techniques and articulated figure positioning by multiple constraints. *1986 ACM Workshop on Interactive 3D Graphics*, pages 151–170, 1986.

[2] Eric A. Bier, Maureen C. Stone, Ken Pier, Ken Fishkin, Thomas Baudel, Matthew J. Conway, William Buxton, and Tony D. DeRose. Toolglass and magic lenses: The see-through interface. *ACM CHI'94 Conference on Human Factors in Computing Systems*, pages 445–446, 1994.

[3] Doug A. Bowman and Larry F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. *1997 Symposium on Interactive 3D Graphics*, pages 35–38., April 1997.

[4] Doug A. Bowman, D. Koller, and Larry F. Hodges. Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. In *IEEE Proceedings of VRAIS'97*, pages 45–52, 1997.

[5] Frederick P. Brooks, Jr. Grasping reality through illusion-interactive graphics serving science. In *Proceedings of ACM CHI 88 Conference on Human Factors in Computing Systems*, pages 1–11, 1988.

[6] Rudolph P. Darken and John L. Sibert. A toolset for navigation in virtual environments. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Virtual Reality, pages 157–165, 1993.

[7] Michael Gleicher and Andrew Witkin. Through-the-lens camera control. In Edwin E. Catmull, editor, *SIGGRAPH 92 Conference Proceedings*, volume 26, pages 331–340, July 1992.

[8] D. Mapes and J. Moshell. A two-handed interface for object manipulation in virtual environments. *Presence*, 4(4):403–416, 1995.

[9] Mark Raymond Mine. Virtual environment interaction techniques. Technical Report TR95-018, Department of Computer Science, University of North Carolina - Chapel Hill, May 4 1995.

[10] Mark Raymond Mine, Frederick P. Brooks, Jr., and Carlo H. Séquin. Moving objects in space: Exploiting proprioception in virtual-environment interaction. *SIGGRAPH 97 Conference Proceedings*, pages 19–26. August 1997.

[11] Randy Pausch, Tommy Burnette, Dan Brockway, and Michael E. Weiblen. Navigation and locomotion in virtual worlds via flight into Hand-Held miniatures. *SIGGRAPH 95 Conference Proceedings*, pages 399–400. August 1995.

[12] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark Raymond Mine. Image plane interaction techniques in 3D immersive environments. *1997 Symposium on Interactive 3D Graphics*, pages 39–44., April 1997.

[13] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. Voodoo dolls: Seamless interaction at multiple scales in virtual environments. *1999 Symposium on Interactive 3D Graphics*, pages 141–145, April 1999.

[14] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: Non-linear mapping for direct manipulation in VR. *User Interface Software and Technology*, pages 79–80, 1996.

[15] Dieter Schmalstieg, L. Miguel Encarnação, and Zsolt Szalavári. Using transparent props for interaction with the virtual table (color plate S. 232). *1999 Symposium on Interactive 3D Graphics*, pages 147–154, April 1999.

[16] Dieter Schmalstieg and Gernot Schaufler. Sewing worlds together with SEAMS: A mechanism to construct complex virtual environments. *Presence - Teleoperators and Virtual Environments*, 8(4):449–461, August 1999.

[17] Richard Stoakley, Matthew J. Conway, and Randy Pausch. Virtual reality on a WIM: Interactive worlds in miniature. *ACM CHI'95 Conference on Human Factors in Computing Systems*, pages 265–272, 1995.

[18] Stanislav L. Stoev, Dieter Schmalstieg, and Wolfgang Straßer. Two-Handed Through-The-Lens-Techniques for Navigation in Virtual Environments. *Eurographics Workshop on Virtual Environments*, 16-18 May 2001.

[19] Zs. Szalavári and M. Gervautz. The personal interaction panel - A two handed interface for augmented reality. *Computer Graphics Forum (Proceedings of EUROGRAPHICS'97)*, 16(3):335–346, 1997.

[20] Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, Jr., and Rosemary M. Simpson. Immersive VR for scientific visualization: A progress report. *IEEE Computer Graphics and Applications*, 20(6):26–52, November/December 2000.

[21] John Viega, Matthew J. Conway, George Williams, and Randy Pausch. 3D magic lenses. *User Interface Software and Technology*, pages 51–58, 1996.

[22] Colin Ware and Danny R. Jessome. Using the BAT: a six-dimensional mouse for object placement. *IEEE Computer Graphics and Applications*, 8(6):65–70, November 1988.

[23] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. *1990 Symposium on Interactive 3D Graphics, Vol. 24*, pages 175–183, 1990.