

Efficiently Navigating Data Sets Using the Hierarchy Browser

Andrea Bönsch, Sebastian Pick, Bernd Hentschel, Torsten Kuhlen

Virtual Reality Group
Seffenter Weg 23
Tel.: +49 (0)241 80 24922
52074 Aachen

E-Mail: boensch@vr.rwth-aachen.de

Abstract: A major challenge in Virtual Reality is to enable users to efficiently explore virtual environments, regardless of prior knowledge. This is particularly true for complex virtual scenes containing a huge amount of potential areas of interest. Providing the user convenient access to these areas is of prime importance, just like supporting her to orient herself in the virtual scene. There exist techniques for either aspect, but combining these techniques into one holistic system is not trivial.

To address this issue, we present the Hierarchy Browser. It supports the user in creating a mental image of the scene. This is done by offering a well-arranged, hierarchical visual representation of the scene structure as well as interaction techniques to browse it. Additional interaction allows to trigger a scene manipulation, e.g. an automated travel to a desired area of interest. We evaluate the Hierarchy Browser by means of an expert walkthrough.

Keywords: interaction techniques, user interfaces, reuseable software, user study

1 Introduction

Virtual Reality (VR) systems are used to interactively and immersively explore virtual environments (VEs), e.g. architectural models, medical scans, or numerical simulation results. But the more complex those VEs become, the more difficulties occur to orient oneself. This is caused by a huge abundance of information in the virtual scene, that has to be accessed and remembered. Occlusions and cluttering make it hard and sometimes impossible to access the desired data. Thus, it becomes important to efficiently present relevant areas to the users. On the one hand, techniques to reveal structures by e.g. fading out objects which occlude the areas of interest are needed. On the other hand, the user needs to be supported during different tasks. First of all, she needs aid to quickly build up a mental, hierarchical structure, called the *cognitive map* [BKLP04]. This map, used to understand the scene structure, includes only the relevant information and their relationship as *treelike structures* [BKLP04]. Furthermore, the user needs assistance to keep all the important positions of the virtual scene in mind. Besides this, a technique is needed to enable an efficient

exploration of the scene with the requirement of quickly reaching the points of interest. This situation leads to two major issues:

1. Search Problem - Where can relevant information be found?
2. Accessibility Problem - How can all the relevant information be efficiently accessed?

Both aspects are addressed in the area of route-planning respectively guided tours. There, the aim is to define a tour through the VE along which the user is automatically moved. The tour path is defined by a set of locations in the scene from which relevant information can be accessed. Those locations are the key points of the user's cognitive map. Instead of guiding the user to all positions in a certain, predefined order, our focus is on semi-automatic route planning. Here, the next destination is chosen during exploration depending on what the user needs to access next.

We assume a set of locations, called Areas-of-Interest (AoIs), to be predefined by an expert, consisting of the relevant structures of one VE addressing a specific scenario. The problem to be solved is how to present those AoIs to the user to support her during the aforementioned tasks.

Various requirements make it complicated to find a solution for a suitable presentation of the collection. Probably the most important aspect is that users have to intuitively understand the relationship between the AoIs to be able to orient themselves in the virtual scene. Another aspect is that nowadays users from varying backgrounds and experience use to work with VR. On the one hand there are knowledgeable users, who only need little support in exploration of the scene, on the other hand there are also novice users who are often not familiar to VR techniques. Nevertheless, there should only be one single technique to represent the set that is used by all users, enables to work efficiently, and supports the learning process.

Because of the complexity of most virtual scenes, we also have to deal with occlusions and clutter. So beside the aspects of understanding the scene structure and pure guided navigation, the representation should also be used as a tool to systematically manipulate the scene. Those manipulations may be, e.g., fading in or out scene objects or additional information like annotations.

In this paper we present the *Hierarchy Browser (HiBro)*, a map that respects all mentioned requirements. It consists of three sub-structures. One part gives an overview over all relevant scene structures and their relationships. The second section presents a subset of those structures in more detail, and the last part shows a close-up of one single structure. While developing this technique, our design was influenced by two main goals. The first aspect was an easy and intuitive handling, because novice and knowledgeable users should both be able to work with the *HiBro*. Secondly, this map should have a footprint that is as small as possible, so large occlusions between it and the virtual scene are avoided, underlining the support function of the map.

The rest of this paper is structured as follows: Section 2 presents relevant related work. In section 3 we explain the *HiBro*'s structure. Section 4 covers an expert walkthrough that

we conducted in order to create an initial evaluation of the interaction technique. Finally, we give a brief conclusion and an outlook on future work in section 5.

2 Related work

Navigation is an important aspect of VR as virtual environments (VEs) readily lend themselves to interactive exploration [BKLP04]. Therefore one needs to be able to *travel*, meaning to change one’s position and field of view within the VE. But traveling is just one part of navigation. The second one is *wayfinding*. It describes the mental process of identifying a route from the current position to the desired destination. Wayfinding requires spatial awareness, which is divided into spatial knowledge and spatial orientation. Both aspects are supported by the so called *cognitive map* [Kit94], a mental construct that automatically builds up in a user’s head while exploring unknown areas. It describes the structure of the world known so far. Humans memorize landmarks in an area and the relationship between them, e.g., their distance to each other. The more they get to know the area, the more information is added to the cognitive map, which keeps its hierarchical structure. With the help of the cognitive map, the user is then able to find a way to another location.

To support the user to quickly build a cognitive map for a specific VE, different techniques have been developed. One possibility is to leave the task of traveling to the user but assist her at wayfinding by providing a map on which important areas have been marked [BKLP04]. The advantage of explicitly travelling is that it supports building the cognitive map, as the user is able to independently explore the VE while moving to her desired destination [WZZ09]. Another possibility are guided tours that lead a user along automatically generated paths to locations of interest within the scene. To make this automated movement comfortable for the user, one has to respect certain requirements, like collision-free travel paths or smooth transitions between the state of no motion and motion [AVF04].

While exploring and interacting with the scene, users may often have to travel between different locations. Once they established a sufficiently detailed cognitive map, they can quickly identify a route to the desired location and travel there. Nevertheless, when navigating to the same locations over and over again, it would be more desirable for the user to only indicate the location of interest and then be taken there automatically. One possibility are teleportations. The user chooses a destination and is instantly taken there. Unfortunately, it was shown that teleporations should not be used in VR, because they lead to disorientation and cyber-sickness [BKH97]. Instead, guided tours can be used here to address these issues.

To give the user further assistance in understanding the structure of a scene and building up her own cognitive map of it, the Hierarchically-Structured Map (HiSMap) was developed [BBP09]. Having been created specifically for exploration of multi-scaled VEs, it depicts a scale-based hierarchy of the objects in the scene, setting them in spatial relation to each other. The hierarchies of a scene were manually created by experts with the purpose of communicating their knowledge about the scene to novice users. The HiSMap was also integrated with guided tours: by selecting entries from the HiSMap, a user could initiate an

automated navigation that brought her to the selected object.

3 Hierarchy Browser

In this section we explain the *HiBro*'s structure and how users can interact with the map.

Our initial situation is a set of locations of a given VE, predefined by an expert of this special scene. The process of extracting those AoIs is not relevant for the concept of our *Hierarchy Browser*, so it will not be discussed here. For each AoI in the set, a position and an orientation in the VE are given. They are used as destination for the guided navigation later on. In addition, thumbnails showing the structure at each AoI are assumed to be available. Those images address the *environmental cognition* [Kit94] of the user and are used as preview of the set in our design. The collection now has to be presented in a way that users

1. are able to build up a cognitive map out of the presented information,
2. need not keep all AoIs in mind and
3. are able to find a desired AoI and select it in order to be automatically moved there.

One problem to be solved is that it is not sufficient to just show the thumbnails of the structures themselves. In particular, the first aspect requires a representation with additional information about the AoIs' relationship. Therefore, a logical grouping of the AoIs has to be done. Bacim et al. proposed one layout for multiscale VE [BBP09]. They developed a 2D map, called Hierarchically-Structured Map (HiSMap), in which the whole collection is displayed as a tree. The AoIs are grouped with respect to scale and then presented in a way that each level of the map's tree represents a level of scale of the VE. Each node represents one AoI and is mapped to the appropriate thumbnail. Applying this idea to all different VEs, other metrics for grouping might be distance, color, hierarchical composition and so on. Which criterion is used for a special virtual scene, is not relevant for our design. The representation as a tree, suggested by Bacim et al., supports building up the cognitive map, as this mental construct also includes all relevant information as "treelike structures" [BKLP04]. Although Bacim et al. scale their nodes up while focusing them, the unscaled nodes have to be presented in a size so that users can recognize what is shown on the thumbnails. This approach results in a big map, which is disadvantageous, because huge parts of the scene will be occluded by it. Additionally, in our opinion, the tree presents too much information especially for novice users, which leads to mental overload. Our design is an extension to the HiSMap, specifically improving these aspects.

3.1 Structure

As proposed in [BBP09], we also use a rectangular, 2-dimensional map to present the predefined and grouped AoIs. In contrast to the HiSMap, we paid attention to create a space-saving map, so that only small portions of the scene will be covered by it. Overall, we divided

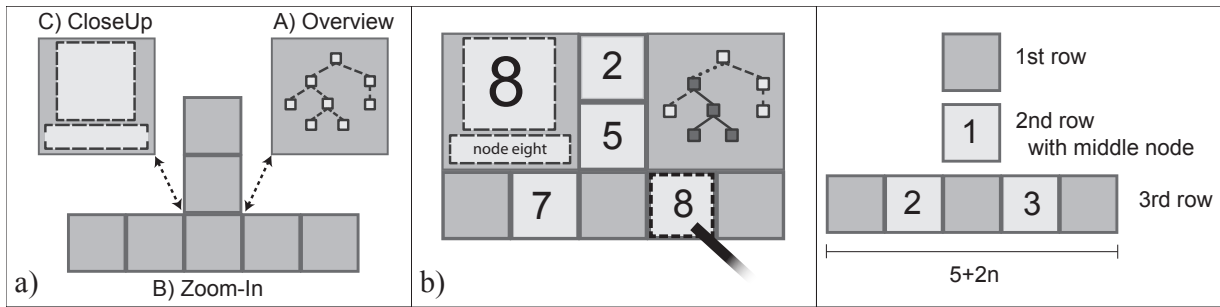


Figure 1: (a) Trisection of the *HiBro*: the *Overview* (A), the *Zoom-In* (B) and the *CloseUp* (C). (b) Relationship between *Overview* and *Zoom-In*: All nodes in the tree, colored dark grey, are shown enlarged in the *Zoom-In*, building a subtree. Those nodes are connected by solid edges. The edges connecting the root and the top node of a subtree are shown as dotted lines. For all other edges dashed lines are used. (c) The *Zoom-In* is made up of three rows, which represent a subtree of the AoI hierarchy that is defined by the *center node*.

the map into three parts with different functionality and different levels of detail as shown in figure 1(a). With this trisection, all requirements enumerated in section 1 can be fulfilled as will be explained next.

3.1.1 Overview

The *Overview* is positioned in the upper right corner of the *HiBro* (cf. figure 1(a)). Here the relevant scene structure, established as a set of AoIs, is shown as a tree. The tree is computed in a way that subtrees are positioned symmetrically beneath their parents in order to quickly understand the (node) relationship and, from it, gain insights into the scene structure. As proposed in [BBP09], every node represents an AoI. In contrast to [BBP09], we do not show the thumbnails here, because a plain tree presentation requires less space. The images will instead be used in the other two parts of the map.

In order to enable an easy interaction, we decided to use the widget-based concept introduced in [TPHW⁺09]. All nodes are widget-handles, which means they can be focused with a pointer and afterwards selected. As focus strategy we use the *IntenSelect* technique [dHKP05]. The edges are only used to show the relationship between the AoIs.

As will be shown in 3.1.2, users can define a small subtree. To indicate which subtree is selected, we use a color coding in the *Overview*, shown with stipple patterns and grey colors in figure 1(b). All nodes belonging to the subtree, here nodes 2, 5, 7 and 8, are colored dark grey and are connected with solid edges. If the subtree does not contain the root of the overall tree, the edges connecting the tree and the subtree root are marked with dotted lines, here shown between node 1 and 2. In this way users are able to directly see the path they have covered so far. All other edges have dashed lines.

Because the whole relevant structure is presented at once, which supports short-cuts, the *Overview* can be used by scene experts, which use the *HiBro* just for guided navigation.

3.1.2 Zoom-In

In the second part of the *HiBro*, called the *Zoom-In*, a subtree is shown enlarged (cf. figure 1(a)). The subtree is defined by a so called *center node*, its direct predecessor and successors. This leads to a subtree consisting of three layers. There is only one exception; If the subtree is defined by the root of the tree, the first layer remains empty.

The *Zoom-In* is shaped as an inverted T, consisting of three rows, one for each layer (cf. figure 1(c)). Row one and two, numbered from the upper part of the map, contain zero or one node each. The *center node* selected by the user, as will be explained in 3.2, is placed in the second row. The predecessor of the *center node*, if available, is positioned to the first row. The third row contains all direct successors of the *center node*. It has places for $5 + 2n$ nodes in total with $n \in \mathbb{N}_0$, corresponding to the maximal number of direct successors in the map. We decided to always enforce an uneven number here, to be able to symmetrically present the subtree. Rows one and two are positioned above the middle slot of row three and the successors are ordered in a way, that the row is filled symmetrically with respect to the rows above. If we, e.g., have an even number of successors, the middle slot of the third row is left empty (cf. figure 1(c)).

In the *Zoom-In*, we match the thumbnails to the corresponding nodes. Although we have to scale the nodes to enable users to recognize the images, this part of the map is still small, because we just present an extract of the whole scene structure. To be able to distinguish between the nodes in the *Zoom-In* and the *Overview*, we name the former *Zoom-In-Tiles*.

As described in 3.1.1, every *Zoom-In-Tile* is a widget-handle. Both, the nodes and the *Zoom-In-Tiles*, will be highlighted while focusing. This is exemplary shown in figure 1(b) for *Zoom-In-Tile* number 8.

Because we just show an excerpt of the full structure with more detail, this part will be used by novice users to learn the scene structure step by step to reduce the mental overload.

3.1.3 Close-Up

The third part of the map, positioned in the upper left corner, is called *Close-Up* (cf. figure 1(a)). This area is for feedback only and will be used by all users. As mentioned in 3.1.1 and 3.1.2, the nodes and *Zoom-In-Tiles* can be focused. In this case, the thumbnail of the current focus is shown enlarged in the *Close-Up*, exemplary shown for *Zoom-In-Tile* number 8 in figure 1(b). In addition, a text label with up to two rows, depending on the text length, is presented as further explanation for the scene structure.

3.2 Interaction

The *HiBro* provides three kinds of interaction to enable an efficient usage. First the basic functions are opening and closing the map whenever wanted in any place and with any alignment. In addition the *HiBro* can be moved through the VE while being open to reduce situations where important scene structures are occluded by the map. Second, we enable

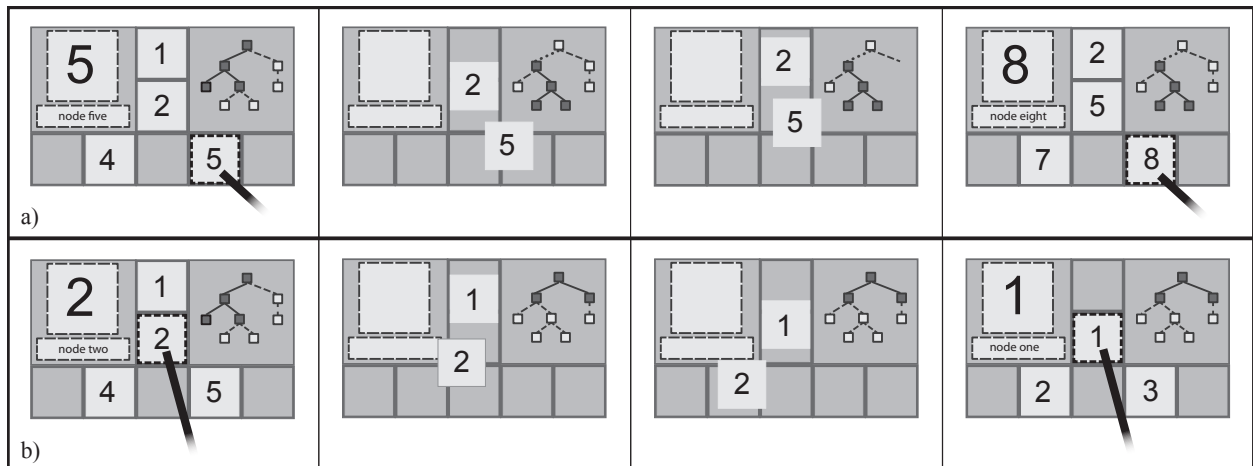


Figure 2: Users can use the *Zoom-In* to navigate step by step through the hierarchy. To illustrate the movement, the tiles, shown in both subtrees, are moved by a linear animation to their new respective positions. (a) If the user chooses an inner node from the third row, she climbs down. (b) If she chooses a node out of the first two rows, she climbs up.

the user to navigate through the predefined scene hierarchy. Manipulating the scene and starting the guided navigation is the third interaction.

3.2.1 Navigating through the scene hierarchy

To be able to build up a cognitive map or to recover a desired structure in the representation, users have to be able to access every level of the hierarchy. In the *Overview* this is supported without additional effort. The user can focus a node with the pointer; the *Close-Up* is subsequently updated. As visual feedback, the focused node is highlighted. If the user selects this node, she defines a subtree, which is immediately displayed in the *Zoom-In*. If the selected node is an inner node or the root, the subtree is defined as described in 3.1.2. Otherwise if the user chooses a leaf, the parent will be used as *center node* of the subtree.

In the *Zoom-In*, we need additional effort to allow access to all hierarchy levels. We enable a stepwise navigation depending on the selection of the user. A *Zoom-In-Tile* can be focused using the pointer. The focused tile is highlighted and the *Close-Up* is updated accordingly, shown on the left side of figure 2. If this *Zoom-In-Tile* is an inner node of the overall tree, it's position in the inverted T is important for the initialized navigation through the hierarchy. If the tile is in row three, the user can climb down one step, as shown in figure 2(a). Otherwise, she reaches a level one step higher in the hierarchy, which is shown in figure 2(b). In case the selected *Zoom-In-Tile* is a leaf, nothing will happen. To clarify which navigation is initialized by the selection, we implemented an algorithm that first deletes all tiles not contained in the new subtree. Afterwards, the remaining tiles are smoothly moved to their new respective positions by linearly interpolating between their old and new positions. Finally, the new tiles completing the subtree are faded in. With this algorithm, all levels of the hierarchy can be accessed by the user.

3.2.2 Manipulating the scene

For the third interaction, we assign a scene state to each AoI and this state can be recalled by selecting the respective node. The scene state may include e.g. *user position and orientation* or *visibility of objects* and so on. Once selected, state changes may be animated to keep users oriented throughout the process. In this way, we e.g. construct a guided navigation to a desired AoI. We explicitly choose an automated travelling approach over a manual one as we want to focus on travel efficiency. At the same time, a smooth transition between current location and destination should enable the user to gain an overview over the scene, similar to the *Automated Guide* technique presented in [WZZ09], and thus support the user in creating a cognitive map. In addition to the automated transition, we fade out certain objects to resolve occlusions.

3.2.3 Timer Interface

We provide our map with different numbers of interaction buttons, which will be described in chapter 4. Because different interaction techniques are often combined in a VE, the number of buttons for one single technique has to be reduced. We establish two configurations where two functionalities are initiated by the same button. Therefore, a criterion is needed to decide between both commands. We decided to use a timer interface here, similar to touchpad devices. To avoid any difficulties, we use a visual feedback: a small sphere, positioned in the middle of the selected node or *Zoom-In-Tile* fading from red to green over time. On button press, the sphere is red, which means the first interaction will be initiated. When the sphere has become green, the second interaction will be triggered.

4 System evaluation

For an initial evaluation of the *HiBro* we chose a rather small-scale expert walkthrough over an extensive user study with intermediate and novice users. The reason for this is that we wanted to focus on evaluating the general relevance of our extensions and on identifying the main design issues before analyzing the *HiBro*'s usability in greater depth.

4.1 Scene description

Our test scene consisted of the geometric model of a human nasal cavity that was explored within the VRhino II project [HKB05]. A hierarchy was created from it by segmenting the model according to its main anatomical components. First, the entire model was divided into the right and left nasal cavity. Next, each cavity was further segmented into its septum, the flow channel, and the middle and lower conchae. Selecting an entry from the resulting hierarchy using the *HiBro* initiated a circular camera motion around the nose's Center of Gravity towards the desired structure. This continuous guided navigation helped to preserve



Figure 3: On the left side the *HiBro* for a scene containing the geometric model of a human nasal cavity is shown. The right shows a user interacting with the browser.

context when changing from one point of view to another. Finally, as the most important structures could not be seen from outside the nose, structures that occluded the AoI were faded out until becoming completely transparent during the camera motion.

4.2 Expert walkthrough

We conducted the walkthrough using a 2-sided LBench. Each screen measures 2.5×1.4 meters and has a display resolution of 1920×1080 pixels. In total, ten experts (average age 28.5 years) participated in the walkthrough. All of them were male and worked in the field of Virtual Reality. Feedback about the interaction technique was collected via a think-aloud protocol during the test and via a questionnaire right after it. At the beginning of each test session, the subject was told about the motivation of the interaction technique. Next, she was introduced to the trisection of the *HiBro*'s visual representation.

The first task was then to test three different button configurations and to choose one favorite. We randomly chose the order in which the configurations were tested by every participant in an effort to prevent it from having an influence on the test outcome. The three configurations were:

- **C1** - three buttons, one for each functionality
- **C2** - two buttons, one for opening/closing and moving the browser, a short click on the second one for navigation through the hierarchy, and a long click on the same button for scene manipulation
- **C3** - same as C2, but swapping the functions assigned to the short and long clicks

Before testing the configurations with two buttons, we told every subject that there would be visual feedback as support to trigger the right action via the time-based interface. We didn't explain how the feedback works in order to find out whether it is intuitive. If it became evident that a participant didn't understand how the feedback worked after some time, we explained it to her and extended the test time for the configuration. After the participant had chosen her favorite configuration, we used it for the rest of the walkthrough.

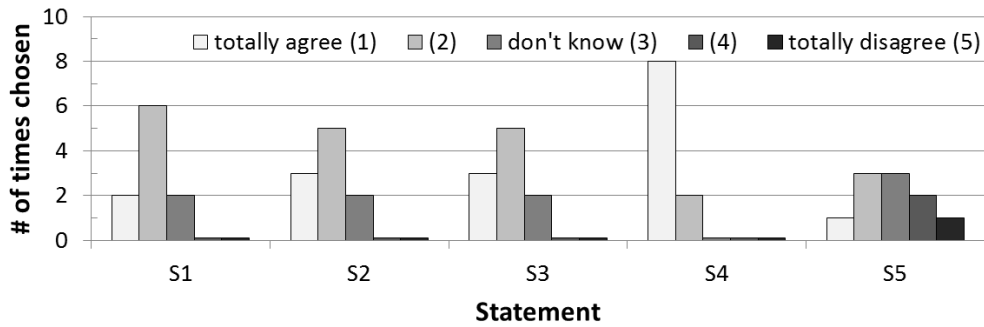


Figure 4: Statements of the user study: **S1** - The trisection supports the interaction with the *HiBro*. **S2** - The tree representation of the *Overview* supports the comprehension of the scene structure. **S3** - The tile animation of the *Zoom-In* supports the hierarchy navigation. **S4** - There should be a possibility to integrate more information than only a context label. **S5** - The usability of the time interface is supported by the visual feedback.

The next task was concerned with the link between *Overview* and *Zoom-In*. When a scene manipulation is initiated by selecting a node from the *Overview*, this node becomes marked in the *Overview*. The *Zoom-In* can react to this change in one of two ways. Either it synchronizes itself with the *Overview* and shows the subtree that is implied by the newly marked node, or it remains in its current state. To evaluate which of the two setups is more intuitive, the participants were asked to try them both and compare them.

The introduction and test took about 15 to 25 minutes depending on how detailed the comments of the respective participant were during the test. Afterwards all subjects filled out a questionnaire, which was divided into three parts. This took about 15 minutes on average. Most questions were formulated as statements and the participants were asked to either express their degree of agreement or just to agree or disagree. The first part covered the general design of the *HiBro* and included statements about the possibility to freely move the browser around, the legibility of the content labels, the coloring of the tree in the *Overview* and other design aspects. Part two focused on the aspects that were specifically addressed by the tasks during the test. Finally, part three was concerned with application scenarios as well as suggestions for further extensions.

4.3 Walkthrough results

Figure 4 shows some result graphs for the expert walkthrough. Participants stated, that the trisection supports the interaction with the *HiBro* (S1, median 2). Comments indicated that the interaction metaphor is easily understandable with only a few explanations and that novice users would probably be able to learn the interaction quickly. The tree representation of the *Overview* was perceived as supporting the comprehension of the scene structure (S2, median 2). Four participants commented that the *Zoom-In*'s bottom row of tiles should be extended by markers that indicate whether a node has successors or not as this would make it possible to see which navigation paths are possible. Nevertheless, the tile animation of

the Zoom-In was recognized as supporting the hierarchy navigation (S3, median 2). Almost all participants indicated the wish to be able to add more content to the *HiBro* than only a text label (S4, median 1). Examples included meta data about the scene (e.g. simulation values) or the hierarchy itself (e.g. author, time of creation) but also zoom-ins on details of a scene object or marking the scene object that corresponds to the selected hierarchy node. Another four participants expressed that if a node was focused either in the *Overview* or the *Zoom-In* it should also be highlighted in the respective other part of the *HiBro*. By doing so, the connection between *Zoom-In* and *Overview* would become even clearer.

The different button configurations C1 - C3 were not rated uniformly. Five subjects preferred the interaction with three buttons. Four participants specifically stated that they didn't directly understand how the time-interface worked. Overall the opinions on the time-interface widely differed (S5, median 3). Being asked which two-button configuration (C2 and C3) they prefer there was no clear answer. On the one hand, four participants preferred a short click for a navigation through the hierarchy, as they think a short click is more intuitive for a widget intern interaction. On the other hand, seven probands stated that a short click is good for scene manipulation, because they perceived this as the main function of the *HiBro*.

All subjects preferred the *Zoom-In* to automatically synchronize with the *Overview* when the latter one was used to initiate a scene manipulation. Nevertheless, it was stated twice, that it would be helpful to decide at runtime whether the automated synchronization is performed or not. As a reason, they indicated that maybe users would like to compare the subtree to some scene structure. Without the synchronization, it would be possible to first select a subtree and then to manipulate the scene to compare both.

All in all the participants evaluated the *HiBro* positively, even though a couple of points of improvement could be identified. Eight participants even declared that they would like to use the *HiBro* in their own projects. Some specific extensions they would like to see when thinking about integration into their projects included a different representation of the scene structure in the *Overview* (4 times), e.g. a mind-map-like representation, and the possibility to bookmark nodes that are selected more frequently during exploration (4 times).

5 Discussion and future work

We presented the *HiBro*, an interaction technique that assists users in building a cognitive map of a given VE and helps them to efficiently navigate it. For this, the scene's structure and details about it are presented in a well-arranged and clear manner. Based on the presented information, users can trigger a scene manipulation, e.g. an automated travel technique that allows them to efficiently navigate to AoIs. Special care was taken to help novice users, who are new to a given VE, to effectively learn and navigate it, but also to allow advanced users to use their knowledge to more efficiently interact with the *HiBro*. The general relevance of the extensions integrated into the *HiBro* and its design were evaluated by means of an expert walkthrough.

While the results of the expert walkthrough widely vindicated the *HiBro*'s design and confirmed its usability, a couple of issues could be identified. These will be subject to future work. This includes adding feedback to specific points of the *HiBro* but also linking its individual structural components more tightly to each other. Besides addressing the issues from the walkthrough, we also plan to add extensions. First of all we would like to explore other visual representations of a scene structure. It also became evident during the expert walkthrough, that users would like to bookmark individual AoIs from a scene structure to increase interaction efficiency. Finally, while currently the scene structure is defined by an expert in a separate pre-processing step, it would be desirable to be able to create or extend it during exploration, e.g. to address dynamically changing scenes. To enable this, additional interaction techniques would have to be combined with the *HiBro*.

References

- [AVF04] C. Andújar, P. Vázquez, and M. Fairén. Way-Finder: Guided Tours Through Complex Walkthrough Models. *Computer Graphics Forum*, 23:499–508, 2004.
- [BBP09] F. Bacim, D. Bowman, and M. Pinho. Wayfinding Techniques for MultiScale Virtual Environments. In *Proceedings of the IEEE Symposium on 3D User Interfaces*, pages 67–74, 2009.
- [BKH97] D. Bowman, D. Koller, and L. Hodges. Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. In *Proceedings of the Virtual Reality Annual International Symposium*, pages 45–52, 1997.
- [BKLP04] D. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, 2004.
- [dHKP05] G. de Haan, M. Koutek, and F. Post. IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *Proceedings of the 9th International Immersive Projection Technology Workshop and 11th Eurographics Workshop on Virtual Environment*, pages 201–209, 2005.
- [HKB05] B. Hentschel, T. Kuhlen, and C. Bischof. VRhino II: Flow Field Visualization inside the Human Nasal Cavity. In *Proceedings of the IEEE Virtual Reality*, pages 233–236, March 12-16 2005.
- [Kit94] R. Kitchin. Cognitive maps: What are they and why study them? *Journal of Environmental Psychology*, 14:1–19, 1994.
- [TPHW⁺09] I. Tedjo-Palczynski, B. Hentschel, M. Wolter, T. Beer, and T. Kuhlen. Toolkit-Independent Interaction Specification for VR-based Visualization. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, pages 263–264, 2009.
- [WZZ09] A. Wu, W. Zhang, and X. Zhang. Evaluation of Wayfinding Aids in Virtual Environment. *International Journal of Human-Computer Interaction*, 25(1):1–21, 2009.