Choosing a Selection Technique for a Virtual Environment

Danilo Souza[1], Paulo Dias[1,2], and Beatriz Sousa Santos[1,2]

[1] DETI/UA- Department of Electronics, Telecommunications and Informatics
[2] IEETA- Institute of Electronics and Telematics Engineering of Aveiro
University of Aveiro
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
`danilo@danilo-souza.net, {paulo.dias,bss}@ua.pt`

Abstract. Bearing in mind the difficulty required to create virtual environments, a platform for Setting-up Interactive Virtual Environments (pSIVE) was created to help non-specialists benefit from virtual applications involving virtual tours where users may interact with elements of the environment to extract contextual information. The platform allows creating virtual environments and setting up their aspects, interaction methods and hardware to be used. The construction of the world is done by loading 3D models and associating multimedia information (videos, texts or PDF documents) to them.

A central interaction task in the envisioned applications of pSIVE is the selection of objects that have associated multimedia information. Thus, a comparative study between two variants of the ray-tracing selection technique was performed. The study also demonstrates the flexibility of the platform, since it was easily adapted to serve as a test environment.

Keywords: Virtual Reality, Virtual Environments, Selection.

1 Introduction

Setting up a virtual environment is a complex process that requires a large amount of time and resources, and also specific knowledge in programming languages and computer graphics. This is true even when it is created with the aid of specialized frameworks that abstract part of the complexity, therefore excluding part of the possible benefit to be obtained from a virtual environment. Aiming to alleviate these problems, a platform for Setting up Interactive Virtual Environments (pSIVE) was developed to allow non-specialists to setup interactive virtual environments that can be applied for virtual visits in the scope of different application areas such as training, education, marketing, etc. Besides setting up the virtual environment, the platform also allows to associate additional information to 3D models in the environment. This gives the possibility to select an object and view the available information about it, such as documentation, videos, etc.

A central interaction task in the envisioned applications of pSIVE is the selection of those objects that have associated multimedia information. Because of the relevance of

R. Shumaker and S. Lackey (Eds.): VAMR 2014, Part I, LNCS 8525, pp. 215–225, 2014.

selection in this context, the fact that object selection is one of the fundamental tasks in 3D user interfaces and thus poorly designed selection techniques often have a significant negative impact on the overall user performance [1], a comparative study to evaluate the selection techniques that should be available on the platform was conducted. In what follows a brief overview of pSIVE is given, the comparative study between two variants of the ray-tracing selection technique is described, and some conclusions are drawn.

2 pSIVE Platform

2.1 Architectural Decisions

From the frameworks analyzed, VR Juggler (VRJ)1 was chosen as the basis for our system since it had all the qualities needed to fulfill the project requirements and its community is active developing new features and supporting users to solve problems from its creation back in the late 90's. The project activity was the main point that made VRJ the chosen framework, as while it had a very active community other possible frameworks such as inVRs2 had reduced activity.

The graphics engine was tied to the base framework. For VRJ, even though it supports a number of graphics engines, some are more developed, accepted and therefore, easy to work with. The decision was between OpenSceneGraph and OpenSG. Both were complete solutions to meet pSIVE's requirements. However, the project activity was again decisive and OpenSceneGraph[3] was chosen since OpenSG[4] is outdated and lacks updates and improvements.

The alternative of using a well-known game engine such as Ogre[5] and Unity[6] was also considered, yet in general they are more specialized with focus on game applications, and more generic graphics engines, as scene graphs, more easily cover a wide variety of applications, and adapt to different application types.

2.2 Platform Development

Figure 1 shows an overview of pSIVE's structure. The Virtual Environment (VE) depends on a series of settings to be configured before running using the configuration tool. While the Virtual Environment is built on top of a group of frameworks, namely VR Juggler to handle input/output devices along with VRPN[7] (that adds an extra set of supported devices to VR Juggler through network). VR Juggler also handles the window system creation and the system calls (to the operating system).

---

[1] `http://vrjuggler.org/`

[2] `http://www.invrs.org/`

[3] `http://www.openscenegraph.com/`

[4] `http://www.opensg.org/`

[5] `http://www.ogre3d.org/`

[6] `http://unity3d.com/`

[7] `http://www.cs.unc.edu/Research/vrpn/`

OpenSceneGraph is the graphic rendering framework, however most of its features are encapsulated by VR Juggler. The top four elements in Fig. 1 are the modules built using elements of both frameworks to manage the whole Virtual Environment.

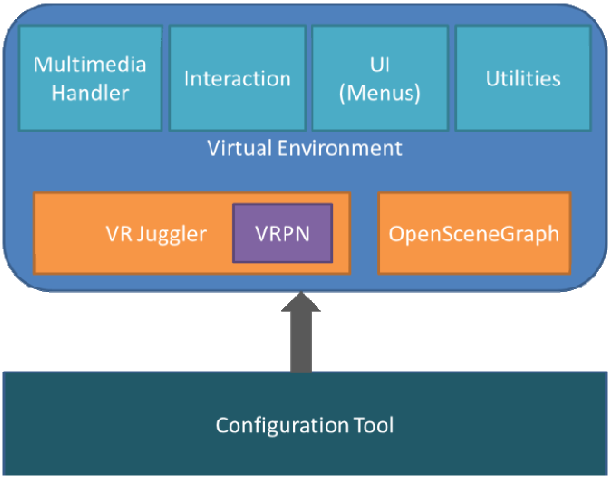


Fig. 1. Platform Overview

pSIVE must be flexible enough to allow a simple simulation (one could load a single model just to see and rotate it in a VE) or load a complex environment such as a whole factory with machinery in which the user could interact with a document describing the machines or watch a documentary on the maintenance of a part on its location. The platform must also cope with several possible devices, file formats to show as contextualized information, as well as the interactions that trigger and control all of it. A group of modules was developed to allow flexibility:

- Multimedia Module: handles the input and exhibition of multiple formats of 3D models and its multimedia contents, the multimedia module takes advantage of the plugin architecture from OpenSceneGraph which loads dynamically the plugin needed for different kinds of formats. Besides the 3D models, the currently supported formats for additional information are: PDF Files, Videos and text files. An example is shown in Fig. 2.
- Interaction module: takes care of the whole interaction allowing navigation, selection and manipulation. An example of selection is presented in Fig. 3, where the user is pointing at a dinner table. The name of the model is presented as visual feedback indicating the existence of additional information related to the model.

Fig. 2. User inspecting a PDF file while immersed in a VE created by with pSIVE

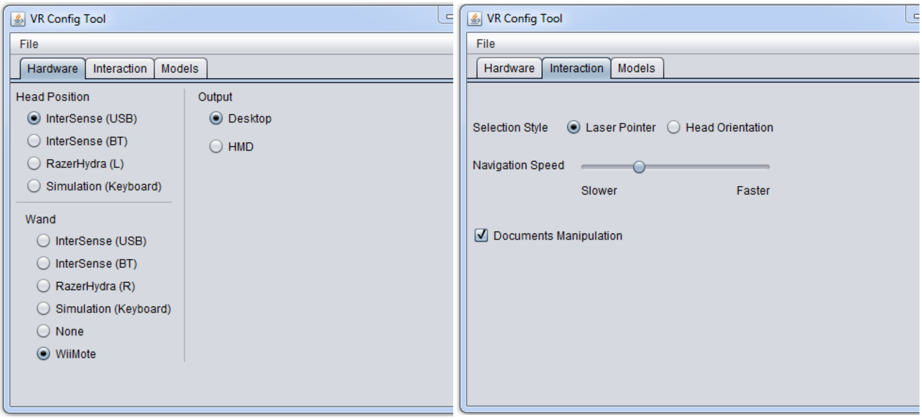
Fig. 3. User selecting an object (dinner table) by head orientation (HOS) while immersed in a VE created with pSIVE

- Menus module: provides a way for the user to access information inherent to virtual elements. The Virtual Environment presents 2D adapted menus created accordingly to the elements that are available for a certain model. In Fig. 4 a three item menu is presented indicating the existence of a video, a pdf file and text information associated to the fridge model.
- Utilities module: provides elements that aid the development and programming of pSIVE such as mathematical and text processing functions.

Fig. 4. A three item menu concerning information associated to a model in a VE created with pSIVE

2.3 pSIVE Configuration

The configuration tool is a simple JAVA application designed to generate the XML file that will control the whole application. It allows users to configure the layout and tell VR Juggler which hardware to use on the Virtual Environment. Its interface was designed to rely on tabs, each one controlling a specific aspect of the system. The hardware Tab allows the user to choose from a list (previously defined by the developer) of equipment supported by pSIVE, dividing them into three classes: Head Tracking, Hand Tracking/Controller and Output. Models tab loads the 3D files giving the possibility of adding information to each file and adjust its position/orientation.



Head Position
● InterSense (USB)
○ InterSense (BT)
○ RazerHydra (L)
○ Simulation (Keyboard)

Wand
○ InterSense (USB)
○ InterSense (BT)
○ RazerHydra (R)
○ Simulation (Keyboard)
○ None
● WiiMote

Output
● Desktop
○ HMD

Selection Style ● Laser Pointer ○ Head Orientation

Navigation Speed
Slower Faster

☑ Documents Manipulation

Fig. 5. pSIVE configuration tool

The last tab controls the interaction styles to be used on the environment: Navigation speed, Selection style and Manipulation (of the documents). Fig. 5**.** presents some views of the configuration tool created.

The list of devices to be used is sent to pSIVE transparently for the user. The configuration tool modifies the VR Juggler files that define each device so the correct ones can be loaded by the platform.

3 Comparison of Selection Techniques

As mentioned, object selection is a fundamental interaction task in Virtual Environments in general, and particularly in the envisioned applications of pSIVE to select those objects that have associated multimedia information. Thus, during the latest stages of development of pSIVE a study was proposed to assess the effectiveness and to find out which selection technique was better applied to different situations, given the characteristics of the platform, and the several possible selection methods, two variants of the "ray-tracing" technique, a well know and very popular selection technique [2], were compared. Both techniques are considered pointing techniques [3] which require the user to perform a set of movements to place a selection tool (on this case, a ray) inside the target (the object to be selected).

The first variation used the hand to control the selection tool with its orientation and position, and to become the selection tool start point. This recalls a laser pointer located in the user's hand. On the second variant the selection tool is controlled by the head, this also is its origin point, requiring the object to be on the center of the user's viewpoint. The two methods share common aspects, however by detaching the viewpoint to the control of the selection tool, the laser pointer technique suffers from the eye–hand visibility mismatch which, according to Argelaguet et al. [2] "bias how the user perceives the difficulty of the task; some objects might seem easy to select but in practice they are not".

Implementations of "ray-tracing" exist since early 90's [4], however most studies [5–7] are specially interested in comparisons between laser pointer and gaze oriented selection (based on eye tracking) or variations of each techniques. From our literature review, we did not found a clear advantage between head orientation and laser pointer and thus we performed a direct comparison to evaluate which technique suits more our platform in different situations.

3.1 Method and Setup

From the analysis of related work and theoretical aspects, the following hypotheses were formulated about this experience: Head Oriented Selection (HOS) may produce fewer errors for distant objects since hand movements are less steady, and Laser Pointer Selection (LPS) may have higher selection times in initial selections given the need for users to place their hands in a comfortable position. We also did not expect any influence of the initial method (HOS or LPS) on users' performance, if they had to use both methods in sequence.

The selection method was the input (or independent) variable, with two levels (HOS or LPS). The output variable (or dependent variable) considered was the user performance and satisfaction with both techniques, assessing basically the number of mistakes and the time elapsed as well as difficulty and preferences. The learning effect according to the initial selection method was considered a secondary variable.

During the evaluation participants had to perform a series of selection tasks that were considered representative of what users would typically do in the applications intended for pSIVE: select a green cube (55cm size) on a grid of 3x3 cubes (Fig. 6.). The position of the green cube to select is random, and after each selection the distance to the grid increases (starting at 5 meters and increasing 5 meters each step until the final position 70 meters away from the user) in order to assess how user performance varies with distance. After performing the tasks, all participants answered a questionnaire concerning their profile and impressions on different aspects of their experience (easy to orientate, pleasant, etc…), the satisfaction rate for each selection method, the preferred method and any comment about difficulties or any other experiment related subject.

The evaluation had the participation of 16 volunteers (14 male and 2 female) with ages between 19 and 26 years. All users performed the selection with both methods, however to compensate for possible learning effects, half of the participants started with Laser Pointer (LPS) and the other half with selection by Head Orientation (HOS) in a within-groups experimental design [8]. The participants were observed while performing these tasks.

This evaluation layout was based on a previous experiment performed by Bowman et al. [9], differing on the selection methods and without manipulation of the selected object.

This experiment also served to test the platform flexibility. The test environment was created using pSIVE, that was modified to record measures characterizing users' performance.

Any device supported by pSIVE could have been used, yet the Razer Hydra[8] (an electromagnetic tracker including two controls) was chosen as it is easy to operate and provides 6DOF allowing to easily emulate the natural act of pointing. Since the Hydra is composed of two controllers, one was placed on the back of each participant's head, to track its orientation, along with the VR2000[9] head mounted display since the built-in tracker is not yet supported neither by VR Juggler nor the VRPN. The other controller was held for inputting commands to start the simulation and to trigger the selection. The second controller was also used as a laser pointing, by tracking the position and orientation of the hand.

---

[8] `http://www.razerzone.com/minisite/hydra`

[9] `http://www.vrealities.com/products/head-mounted-displays/vr2000-pro-dual`

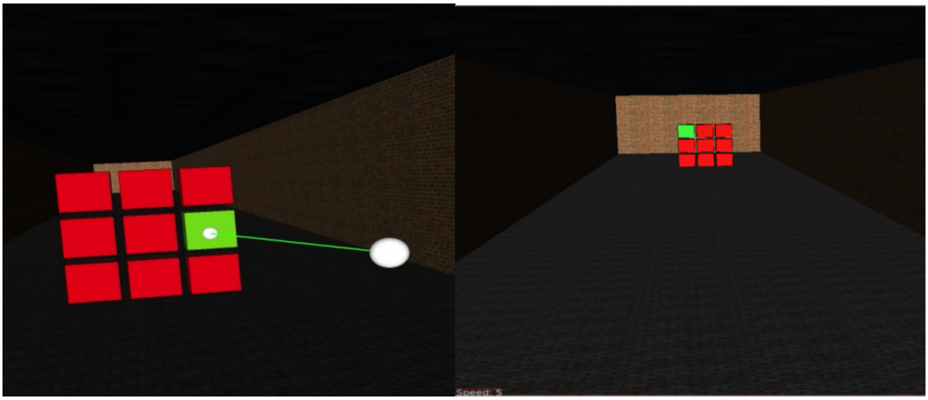


Fig. 6. Selection of a cube using the LPS method (left); next grid position with the object to be selected highlighted (right)

3.2 Results

As mentioned, several measures were automatically recorded by the platform, corresponding to the user's performance: number of errors (an error corresponded to a wrong object selection), time elapsed from the activation of the test and the selection, the distance of the grid and the position of the correct object on each step. The method that the participants began with was also recorded (HOS or LPS).

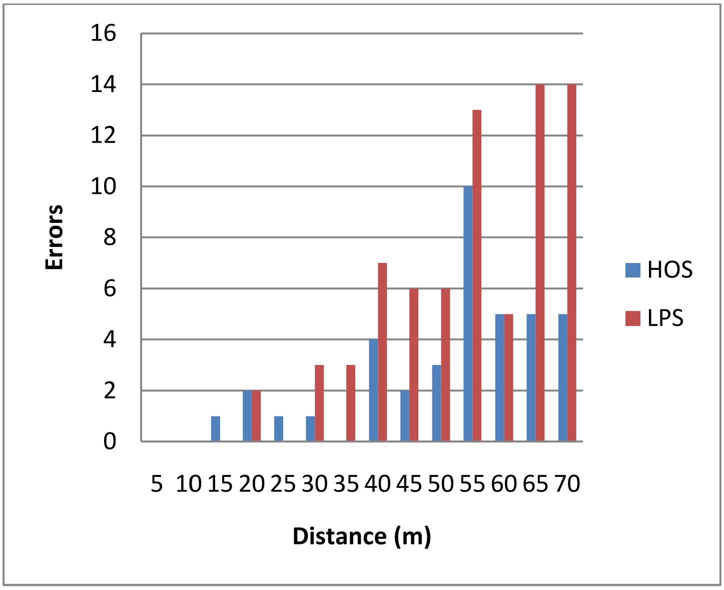


Fig. 7. Sum of errors by distance

Regarding errors, as shown on Fig. 7, users performed fewer errors (wrong selections) with HOS for almost all distances, and the number of errors obtained with both techniques increased with distance; both results were expected.

Regarding the average elapsed time needed to perform the selection at each distance, Fig. 8 shows that participants took longer times while using LPS, in most instances. However, the original difference tends to decrease in the range of 10 to 25 meters. As expected, users needed higher selection times with LPS in the first selections probably due to the fact that they took some time to find a comfortable position for the hand as reported by 8 of the 16 participants.

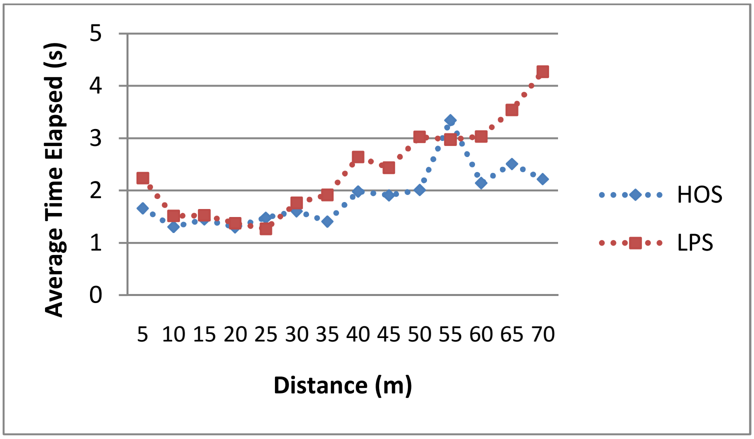
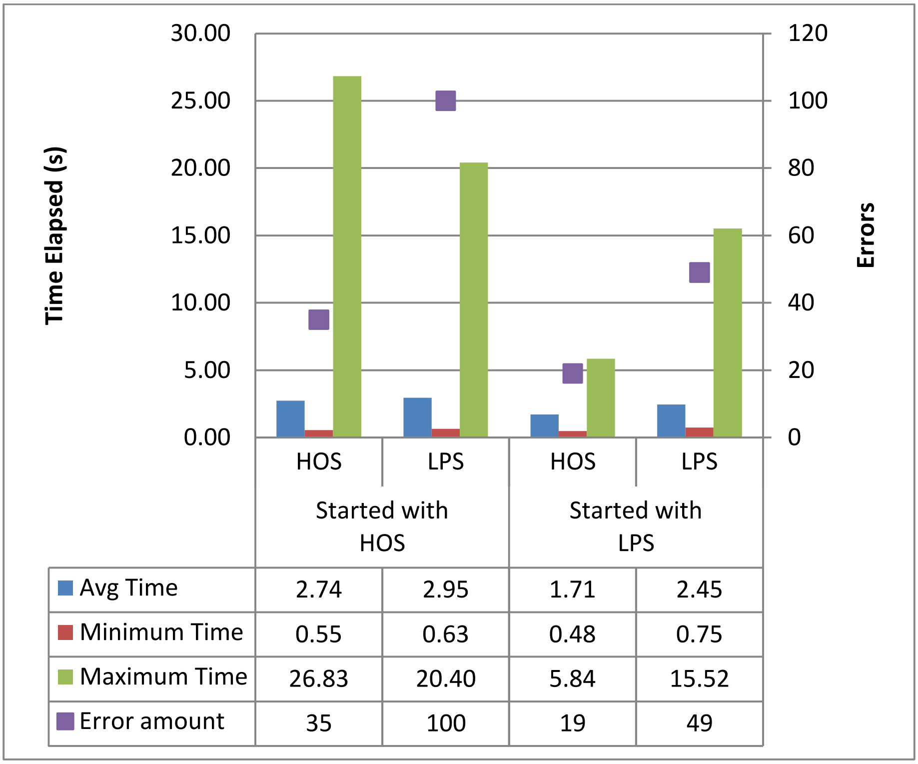


Fig. 8. Average time to select by distance

An interesting result was a noticeable increase in both selection time and number of errors for those who started using the LPS method. Fig. 9 shows clearly that the number of errors accounted for those who began with HOS and were currently using LPS was approximately two times higher than those who started with LPS and were currently using HOS. A possible explanation is the fact that both techniques require small, precise and especially steady movements, which are more easily obtained when controlling the beam with the head. The change from steady movement (with the head) for an unstable movement (with the hand) or vice versa was reflected in the learning rate of the user. It is probable that those who started with LPS were subjected to a method requiring more training to be used, as the users reported (see Table 1), and thus had to be more focused to understand how to interact with the system resulting in a performance improvement when using even HOS.

On the other hand, users generally obtained worse results while using LPS independently of the initial method. These results are reflected on the opinions provided by the participants. On a 5 level Likert-like scale, where 1 means strongly disagree and 5 strongly agree: it can be seen on Table 1 that despite the fact both techniques are pleasant, LPS presents irritant features and requires more training than HOS.



	Started with HOS		Started with LPS	
	HOS	LPS	HOS	LPS
🟦 Avg Time	2.74	2.95	1.71	2.45
🟥 Minimum Time	0.55	0.63	0.48	0.75
🟩 Maximum Time	26.83	20.40	5.84	15.52
🟪 Error amount	35	100	19	49

Fig. 9. Average elapsed time and errors according to the starting selection method

Table 1. Questionnaire results (median of each index in a scale of 1-disagree to 5- agree)

	HOS	LPS
Ease of Orientation	4	4
Is Pleasant	4	4
Has Annoying features	2	4
Is Intuitive	4	4
Requires Training	3	4
Is Useful for near Obj.	5	4
Is Useful for Distant Obj.	4	3
Satisfaction Rate	4	3

From the results, it is clear that that HOS was the technique preferred by most users (Table 1) and along with the results concerning performances, HOS was better for these tasks regardless of the position and distance of the object.

4 Conclusions and Future Work

This paper presents pSIVE, a platform that abstracts the usage of several frameworks to setup Virtual Environments, allowing even laymen to work with it. Albeit several frameworks already ease the creation of Virtual Environments none addressed in a satisfactory way the need to interact directly with code or configuration files, difficult to non-experts. Results suggest pSIVE is flexible enough to allow extending it for domain specific applications, as it was the case for the user study performed. The results of the tests indicate the selection performed with the head orientation allow users to have better overall performance, when directly compared to the results for the selections performed with the laser pointer. However the laser pointer analogy was more intuitive and delivered a better learning effect so that those who started with it could finish the tasks with lower times and fewer errors than those who started with the head orientation.

The current version of pSIVE is a prototype and still requires further refinement concerning its structure and modules. Furthermore, it is still limited and does not have the elements to provide a better and more natural usage of the Virtual Environment, for instance allowing new selection methods supported by the interactions with different types of equipment.

Acknowledgements. This work was partially funded by FEDER through the Operational Program Competitiveness Factors – COMPETE – by National Funds through Foundation for Science and Technology – FCT (references FCOMP-01-0124-FEDER-022682 and PEst-C/EEI/UI0127/2011) and by Produtech PTI - New Processes and Innovative Technologies for Manufacturing Technologies (reference QREN 13851).

References

1. Bowman, Kruijff, E., LaViola, J., Poupyrev, I.: 3D User Interfaces: Theory and Practice. Addison-Wesley Professional (2004)
2. Argelaguet, F., Andujar, C.: A survey of 3D object selection techniques for virtual environments. Comput Graph. 37, 121–136 (2013), doi:10.1016/j.cag.2012.12.003
3. Mine, M.: Virtual environment interaction techniques. UNC Chapel Hill Comput. Sci. Tech. Rep. (1995)
4. Liang, J., Green, M.: JDCAD: A highly interactive 3D modeling system. Comput Graph 18, 499–506 (1994)
5. Jimenez, J., Gutierrez, D., Latorre, P.: Gaze-based Interaction for Virtual Environments. J. Univers Comput. Sci. 14, 3085–3098 (2008)
6. Cournia, N., Smith, J.D., Duchowski, A.T.: Gaze- vs. hand-based pointing in virtual environments. In: CHI 2003 Ext. Abstr. Hum. factors Comput. Syst. - CHI 2003, p. 772. ACM Press, New York (2003)
7. Sanz, F.A.: Pointing facilitation techniques for 3d object selection on virtual environments (2011)
8. Dix, A., Finlay, J.E., Abowd, G.D., Beale, R.: Human-Computer Interaction, 3rd edn. Prentice-Hall (2003)
9. Bowman, D.A., Johnson, D.B., Hodges, L.F.: Testbed evaluation of virtual environment interaction techniques. In: Proc. ACM Symp. Virtual Real. Softw. Technol. - VRST 1999, pp. 26–33. ACM Press, New York (1999)