

A Testbed for Rapid Design and Evaluation of VR Navigation Techniques for Industrial Applications

Jendrik Bulk¹ and Volker Paelke²

Hochschule Bremen, University of Applied Science, Flughafenallee 10,
28199 Bremen, Germany

¹jendrik.bulk@hs-bremen.de

²volker.paelke@hs-bremen.de

Abstract. VR is making inroads into industrial applications, especially in product design and presentation, training and simulation use cases. A focus of these applications is on the visualization, exploiting 3D immersive graphics as a central feature of VR. While navigation is often required in these applications to select a suitable location for observing the visualization or interacting with the simulation content it is not in the focus of the designers, often leading to the selection of techniques that are convenient during development because they are available in the implementation toolkit. However, these may be less than ideal for the industrial application context. A large variety of VR techniques for navigation have been proposed in the literature and implemented, both for motion control and guidance, but there is currently little established knowledge on their respective benefits and shortcoming, especially in non-gaming applications. To enable designers of industrial VR environments to make informed choices of navigation techniques we present a testbed that enables quick prototyping and comparative evaluation in the specific application context and with the intended target audience, making a user centered selection of navigation techniques viable.

Keywords: Virtual Reality, VR, navigation, evaluation, testing.

1 Related Work

Navigation is a common term that signifies how a person moves purposefully from one place to another. In virtual environments navigation is a common and ubiquitous activity as it provides the main mechanism for content selection and is often a prerequisite for interaction with objects of interest that are spatially distributed in a virtual environment. Research into navigation techniques for virtual reality applications has a long history. Bowman et al. [1] introduced a useful and widely adopted definition that defines navigation as the combination of wayfinding (the cognitive component of planning the route and making navigation decisions), and travel (the motor component that determines the current direction and speed during the movement phase). Early research like Bowman et al. [2], Jul and Furnas [3] provides a useful conceptual foundation for the design and evaluation of navigation techniques for virtual environments, by

identifying central factors of influence and mapping out the design space for techniques to support both wayfinding and travel activities. Nabiyouni et al. [4] and Boletsis [5] provide an overview of more recent research that aims to classify and evaluate navigation techniques in virtual environments (VE). The selection of suitable navigation techniques for a virtual reality application is highly dependent on the intended users, the application domain and the available hardware. Navigation techniques for games can make use of a wide spectrum of travel techniques ranging from “magic” techniques like teleporting and “magic carpet rides” to more realistic appropriations of normal techniques like walking or skiing (e.g. Paris et al. [6]) and can also design or adjust the virtual space to control the difficulty of navigation, e.g. by creating maze-like structures to purposefully make navigation more challenging or by providing clearly structured environments with landmarks and spatial constraints to simplify navigation. In industrial applications of virtual reality navigation is typically an essential but conceptually secondary activity that should be unobtrusive, intuitive, and easily controlled to support the user in the primary task (e.g. learning about the industrial environment, analyzing information or performing activities by manipulating content) and the spatial structure of the environment is often dictated by the corresponding real world setting. Therefore, many game inspired techniques are not applicable. “Walking” as an analogue of the corresponding real-world activity might seem like the obvious solution, but as the overview by Nilson et al. [7] shows, there are still serious challenges with many of the “natural” walking techniques that have been proposed for virtual environments.

Disorientation and poorly readable user interfaces within a VE have been found to lead to the user breaking off the experience and generally not continuing. The larger the VE, the more complex the requirements for navigation. Therefore, a comprehensible form of navigation and an adaptation to the VE are required [8]. In complex environments users often find it difficult to understand the structure of virtual areas and navigate them successfully [9]. In contrast to entertainment applications VEs for serious use cases are often bound to mimic an existing, potentially complex, spatial layout. The solution to navigation problems in these cases is therefore limited to the provision of suitable navigation aids, e.g. compass displays that show the direction towards the destination or digital maps of the surroundings on which the destination and position of the user are marked (e.g. a minimap) that allow the user to concentrate on the actual task in the application. Sabok et al. [10] have compared two different navigation and different interaction options. They have found that surrounding objects should have the ability to collide with the avatar. The user is prevented from moving through walls, which otherwise quickly leads to disorientation. Top down views and the use of gravity were also found to help test users to orient themselves more quickly.

Santos et al. [11] created a test environment to compare the usability of VR devices. Their results emphasize that different navigation components may be necessary for different environments. This means that even with good navigation solutions, they cannot be used universally for every scenario. Personal preferences and the experience of the respective user were also found to play a role when evaluating an interface or control. Their research also showed that the “obvious” solution of providing flexibility by offering different options should be balanced so that the user is not burdened by too many options.

The designer of an industrial virtual reality application therefore has to choose navigation techniques based on his specific requirements and validate in user tests that they actually perform as intended. The goal of this research is therefore not to identify an optimal navigation technique, but to provide designers with a toolset that enables them to quickly create and evaluate application specific navigation techniques for their specific project needs.

2 Requirements

As the discussion of the related work shows there are many potential approaches to design navigation solutions for an VR environment and the application specific requirements in the industrial domain are often best addressed by adapting and customizing existing techniques or even developing new approaches. In all cases a user centered approach is required in which techniques can be evaluated easily, rapidly and frequently with users in a scenario that is representative for the specific application context.

To make such a user centered approach viable it needs to be supported by appropriate tools to limit the time and effort required to create the navigation techniques, to create and conduct user tests, to evaluate the test results and to adapt the techniques based on the test results. We established a list of requirements based on experience in previous projects and refined these in discussions with three VR experts. The resulting requirements are:

- A. To provide a set of established navigation and locomotion techniques as a baseline for comparison and as a foundation for customized adaptation of these techniques
- B. To provide the necessary programming environment to create and modify navigation and locomotion techniques for a wide variety of VR hardware
- C. To provide a set of building blocks for navigation tasks that are representative of the application domain
- D. To provide tools for composing virtual test environments of varying complexity to rapidly construct environments that are suitable for testing the navigation and locomotion techniques of interest
- E. To provide support for conducting user tests in the test environments and provide integrated means of data collection
- F. The system and especially the navigation and locomotion techniques should be implemented using widely used technology to ensure that developers don't need to learn new technologies and ideally allow to directly apply the tested navigation and locomotion techniques to the final application.

3 The Testbed Concept

To address these requirements, we have developed a testbed concept that uses a modular approach to both the construction of navigation and locomotion techniques and virtual (test) environments. The design of the concept builds on the modular structures

provided in the Unity game engine. This enables developers familiar with Unity to directly use the testbed and also provides a useful starting point for less experienced developers that can refer to the extensive documentation available for the Unity for the general information and then use a template provided to modify and add functionality using connecting points.

To address requirement (A) the testbed provides a set of ready-to-use navigation and locomotion techniques, e.g. common navigation solutions for applications on 2D-displays like Minimaps, as well as 3D solutions for directional disclosures, e.g. a 3D compass. This set of techniques can be used in two ways: First as a baseline reference for user tests against which the performance of other techniques can be compared. Since the creation of such components is a time-consuming task there is often a temptation to just implement rudimentary versions for a test which then can result in suboptimal conditions for the user which can in turn distort the test results [11]. By providing ready-to-use techniques that have already been tested and validated as a baseline, developers can focus on the implementation of new techniques and functionality, while also receiving more meaningful test results. A second use is the use as a basis for the development and adaptation of new techniques. Because the techniques are provided as standard Unity prefabs they can be easily modified and extended, simplifying the development of application specific adaptations. Over time developers can create and expand the library of “proven” baseline techniques by adding their new techniques to the library.

Requirements (B) and (F) are addressed by using the Unity game engine that offers common functionality like collision detection, physics simulation and rendering, so that the developer can concentrate on the more important task like the controls and navigation for the user. Unity is widely used in the development and prototyping of virtual environments, and therefore familiar to many developers and well supported with integration of most VR hardware available today. Using the Unity prefab mechanism, a template for a new navigation technique can be created by pushing one single button. A new navigation technique can thus be constructed in Unity by using either the menu or inside the inspector. The resulting prefab then has to be customized by adding application specific displaying methods. The script created for the prefab already contains commonly required data such as distance and direction data in relation to the destination. If the final application for which the techniques are developed is also implemented in Unity the navigation and locomotion techniques can be directly reused as prefabs in the target application (Fig. 1). If another base technology is used the techniques need to be ported but the implementation concepts of other game engines or VE toolkits are usually similar enough.

Once a new or customized navigation or locomotion technique has been created it needs to be tested. Requirements (C) and (D) address the need of creating virtual environments that are suitable to test different kinds of navigation systems. One possible solution is to test the techniques directly in the virtual environment where they will be used – however, this is usually only an option if a version of this environment already exists. Frequently the environment itself only becomes available later in the development process necessitating the use of a placeholder and even if the environment is

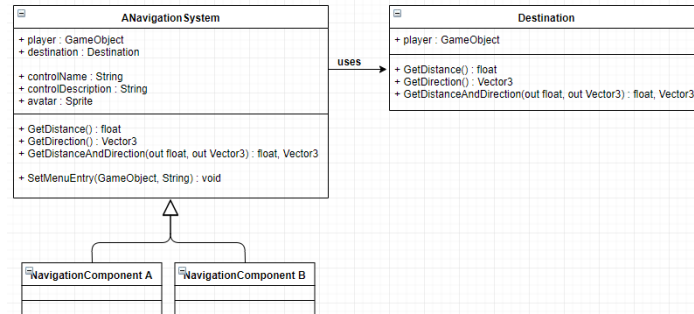


Fig. 1. Implementation of classes for a Navigation Prefab.

already available performance reasons might still make the use of placeholder environments desirable during the development of navigation and locomotion techniques. Without specific support placeholders are often chosen on convenience and availability, leading to either very simplistic environment or ones that do not represent the application domain well (e.g. using a game environment to test techniques for navigating industrial production sites). In our testbed we provide an easy-to-use tile system with different tile stiles reflecting typical, but generic, indoor and outdoor settings from which new environments can be quickly composed, or existing ones adapted quickly, e.g. to support a specific test task. The tiles are again provided as Unity prefabs, allowing to easily adapt and extend the tile collection with new tiles or themes and can therefore use the standard Unity interface to compose new environments.

To support the user tests (requirement (E)) the testbed also includes elements to present test users with descriptions of the test tasks that they should perform within the environment. The different prefabs are also instrumented to automatically provide a standardized protocol of all user interaction during a test and there is an additional mechanism that enables testers to easily add time-stamped event markers to the protocol during a user test (e.g. to mark specific interaction problems) by pressing previously defined buttons.

4 Implementation of the Testbed

The system was created in Unity 2018.3, using the VR API from Unity. In its current form the testbed is independent from device-related SDKs, thus making it possible to use the Unity project with any VR device. Cross-platform builds of the program can also be created with Unity. These two factors make the system flexible.

In the main scene the user can choose between different environments, navigation and control techniques. Every combination is possible, also to combine multiple controls and navigation systems. A laserpointer is used to select the menu entries. The user

can see hands of his avatar to get a virtual presence. Figure 2 shows the main menu with its UI.

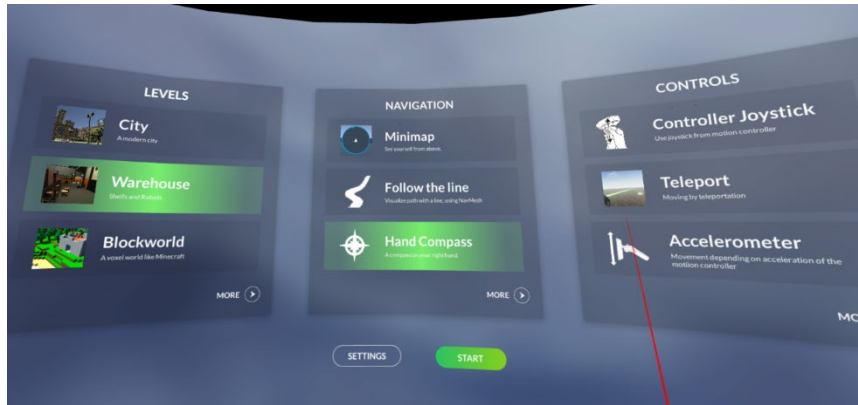


Fig. 2. The main scene of the system. Users can choose between different VEs, navigation and control techniques.

5 Example: Creating Navigation and Locomotion Techniques

A very simple navigation technique that is commonly used in virtual environments is a 3D virtual compass. The navigation prefab in our testbed already provides a function to calculate the direction and distance from the current position of the user to a specified destination. Such a navigation support technique can therefore be easily created by using this information and mapping it to some appropriate 3D graphics object that then points the user in the right direction. Figure 3 shows how a simple compass was created by simply combining the navigation prefab with some custom geometry for the compass.



Fig. 3. A model of a compass showing direction and distance to the destination. The compass is installed to the virtual wrist.

More complex navigation techniques support the user by first calculating a route through the virtual environment and then generating appropriate cues to guide and direct the user. As an example, we show how a simple “follow-the-line” technique can be implemented. It is a 3D correspondence to the known 2D presentation in car navigation systems. As a first step the recommended route has to be calculated. In the final application this might be provided by the internal system or be prescribed by some other means. For the user tests we use the Unity Navigation System to generate routes to specified target locations. This system is normally used to move Non-Player-Characters (NPCs) independently. For it to work the developer must mark the walkable areas. Unity visualizes objects that are marked as “Navigation Static” as a blue mesh and saves them in a map data-structure to improve runtime performance (see Figure 4).

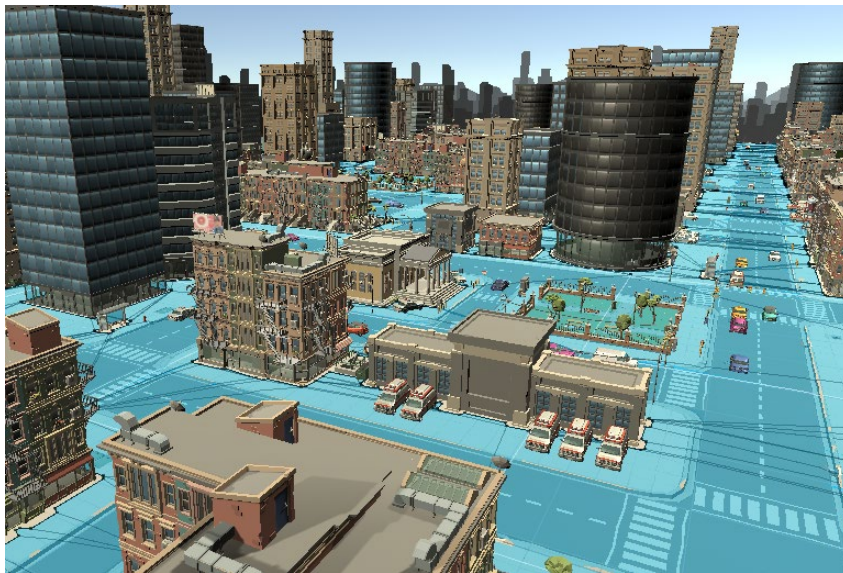


Fig. 4. City environment with blue marked NavMesh.

This enables Unity to rapidly calculate routes to different destinations, which makes it possible to rapidly create a variety of navigation tasks for test, simply by specifying a set of different destinations. In the “follow-the-line” the calculated route is then simply displayed as a shimmering line between the avatar’s location and the destination (Fig 5). Using the route calculated the returned array with corners is then simply connected to a Line Renderer component to make it visible for the user.

Once a developer has implemented the described navigation concepts they can be easily reused. The corresponding techniques are wrapped in an abstract class so that everybody can work with them without a deeper knowledge of the implementation underlying them.



Fig.5. The “follow-the-line” technique uses Unitys NavMesh and LineRenderer function.

6 Example: Creating Test Environments

Test environments are created by composing predefined tiles to form an environment that is suitably complex to test the navigation or locomotion technique of interest, without creating unnecessary detail. Depending on the eventual target environment either indoor or outdoor environments might be suitable. Outdoor environments are typically of larger scale and often contain large background objects like landmark buildings or landscape features. Indoor environments are often more limited in size but can be more challenging for navigation, especially if they feature repetitive architectural features. To create test environments that are suitable for serious use cases like industrial training it is essential, that the environment is reflective of the virtual environment in which it is applied later. In our testbed we provide different sets of tiles suitable both for outdoor environments (Fig 4) and indoor environments (Fig 6). Developers can easily extend the set with either individual tiles (e.g. to add a specific feature required for a user test) or complete tile sets (e.g. to add a new “theme” with completely different features from those initially provided (e.g. process industry vs. discrete manufacturing). The test environments can be easily combined and modified using standard Unity functionality.

As a last step test designers must adjust some variables like jumping and step height, dimensions of the avatars and navigable areas if automatic routing (see above) is required, define the test-tasks and set-up data collection.



Fig. 6. An indoor VE with training tasks.

7 Evaluation

To evaluate the testbed, we conducted two different tests to (A) validate the functionality and assess in how far the goals of simplification and time saving were achieved (B).

7.1 Comparative User Tests

To validate the functionality, that is to ensure that the testbed is actually suitable to conduct user tests we conducted a test with 13 test users as subjects, using a variety of the navigation techniques in different test environments. The group of 13 test subjects consisted of employees and students of the university. They were between 20 and 40 years old. All participants had previous knowledge in the field of computer science. Only one person had no experience with VR applications.

The test users completed 26 test runs using the testbed system to compare different navigation techniques. In each run the subjects were given the task of reaching a destination in different environments with different control and navigation techniques. Fixed scenarios were used during the tests. All test subjects conduct the tests under the same conditions using the same PC, HMD and controller set and having the same range of motion available (in terms of real space). Data was collected using the automatic logging feature of the testbed and an additional questionnaire on user experience delivered in Google Forms to determine whether the subjects regarded the combination of navigation techniques and environment as suitable. The test scenarios were chosen so that the components clearly differ from one another and the test subjects could clearly perceive this difference. Each subject was given a navigation task. The first task involved comparing a MiniMap in two different environments. The navigation component was first tested in an outdoor environment (a city) and then in an indoor environment

(warehouse). These two environments are very different. In the outdoor environment, free movement was possible with the user only being limited by obstacles such as buildings or cars. In this way, an almost direct route to the goal could be taken. In the indoor environment, the user is in a kind of labyrinth. The directions on the MiniMap are not as helpful in an indoor environment in this setting, since this navigation aid only points in the direction of the destination. Due to the labyrinthine structure, it is not possible to find a route according to the cardinal direction. Teleportation was as the locomotion technique in both scenes. This control can be used intuitively by inexperienced subjects and enables rapid progress. Subjects who were not familiar with these forms of navigation and control were instructed and trained in the functions.

The results of the questionnaire confirm the assumption that the MiniMap navigation component is less suitable within the warehouse environment than in the city environment. The next question deals with the deviations made during navigation. The deviations from the specified route are significantly higher than in the city environment. The subjects also rated the MiniMap as more suitable in the city environment than in the warehouse environment, which is shown by the answers to the fourth question. At the end of each test run, the subjects were asked if they considered the software suitable for comparing navigation and locomotion components in VR environments. This was confirmed without exception.

7.2 Expert Review by experienced VE developer (B)

To assess in how far the testbed achieves the goal of simplification and time saving (B) we also conducted an expert review with an experienced VE developer. For this purpose, a developer experienced in the development of VR applications was instructed in the system and asked to create her own components. The entire project was made available to the developer, who received personal instruction in the system and documentation of the software.

The software was then used to create a new combination of navigation techniques and a corresponding test environment. In the new system, the user is shown a glowing 3D model of a monster, which must be followed to reach the goal.

The developer stated that the development environment for the navigation components was very clear. The training period in the modular system was about two hours. The development time, including testing of the new component, was approximately one hour. Another hour was spent making optical improvements. The existing navigation components were used as an example in the development of the new system and provided a quick overview of the functions of navigation components. This accelerated the development of the new component. The developer indicated that a quick overview of the modular elements would be helpful to find your way around more quickly. Since a few days passed between the introduction to the system and the actual development, details on how it works can be forgotten. A quick overview would summarize the important key data and clearly illustrate the functionality of the modular system. This overview will be available to future developers. It was emphasized that the development environment is particularly suitable for inexperienced VR developers. The environment provides a quickly understandable

overview of the connection between prefabs, scripts and 3D models. The simplicity of the system motivated the developer to deal with the system beyond the task at hand.

8 Conclusion and Outlook

As part of this work, a testbed environment was created in which it is possible to develop, compare and test navigation and locomotion components.

A review of VR research helped to understand the problems in navigating virtual environments, especially in serious applications. Navigation and locomotion techniques for industrial use cases must find a suitable middle ground between immersive experience and comfortable control that differs significantly from both typical entertainment or training use cases. It is therefore often necessary to adapt techniques and to validate it in an appropriate setting.

The visual structure and complexity of virtual environments has a significant impact on the ability of users to orientate themselves in VR. To be useful test environments should mimic the final application environment both in structure and visual detail. Therefore, an approach for the quick composition of test environments was developed and appropriate assets were developed to create attractive virtual test environments for industrial use cases. The architecture provides for easy extension to additional areas, e.g. additional elements to depict typical environments from process industry.

Building the software architecture and the components based on the concepts of the Unity Editor helped to create a modular system that requires little training, allows developers to get started quickly and often makes the results directly applicable in the target application if this is also created in Unity. Thanks to the modular design of the test environment, it can be expanded quickly and easily. The project already includes three completed VEs, which can be quickly expanded or redesigned using a tile system. The system also offers basic functions for targeting and moving the avatar.

The user tests confirmed that the system can be used for its intended purpose. All subjects confirmed that the application is suitable for comparing locomotion and navigation components. Currently the program is to be made available to future students of the human-computer interaction course at our university in order to train them in the procedures for conducting and evaluating user tests. The automatic report generation offers further evaluation possibilities for tested components. For students who want to implement their own ideas, there is the option to get access to the project files and the documentation. By adapting the Unity interface, future developers will have a clearly designed interface that can be used effectively even with little knowledge of the Unity environment.

The expert review with an experienced developer confirmed that the tested accelerates the creation of components and test environments noticeably and makes it less complicated by providing a lot of common functionality. For the future work we plan to expand both the set of navigation and locomotion techniques provided and the set of themes and tiles for the creation of test environments. We are currently also gathering large scale feedback by using the testbed in student projects. We also aim to apply the testbed in a comparison with the final implementation in an exemplary application to

check if the results match and expand the data gathering and analytics functionality to simply the analysis of test results.

References

1. Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev (2004) 3D User Interfaces: Theory and Practice. In: Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.
2. Doug A. Bowman, David Koller, and Larry F. Hodges (1997) Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. In: Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97) (VRAIS '97). IEEE Computer Society, Washington, DC, USA, 45-.
3. Susanne Jul and George W. Furnas (1997) Navigation in electronic worlds: a CHI 97 workshop. In: SIGCHI Bull. 29, 4 (October 1997), 44-49. DOI: <https://doi.org/10.1145/270950.270979>.
4. M.Nabiyouni, A.Saktheeswaran, D.A.Bowman, and A.Karanth (2015) Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality. In: 2015 IEEE Virtual Reality (VR), pp. 243–244. IEEE.
5. Boletsis, Costas (2017) The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology. In: Multimodal Technologies and Interaction. 1. 24. 10.3390/mti1040024.
6. Richard Paris, Joshua Klag, Priya Rajan, Lauren Buck, Timothy P. McNamara, and Bobby Bodenheimer (2019) How Video Game Locomotion Methods Affect Navigation in Virtual Environments. In: ACM Symposium on Applied Perception 2019 (SAP '19), Solène Neyret, Elena Kokkinara, Mar Gonzalez Franco, Ludovic Hoyet, Douglas W. Cunningham, and Justyna Świdrak (Eds.). ACM, New York, NY, USA, Article 12, 7 pages. DOI: <https://doi.org/10.1145/3343036.3343131>.
7. Niels Christian Nilsson, Stefania Serafin, Frank Steinicke, and Rolf Nordahl (2018) Natural Walking in Virtual Reality: A Review. In: Comput. Entertain.16, 2, Article 8 (April 2018), 22 pages. DOI: <https://doi.org/10.1145/3180658>.
8. Dodiya, J., & Alexandrov, V. N. (2008). Navigation Assistance for Wayfinding in the Virtual Environments: Taxonomy. In: 18th International Conference on Artificial Reality and Telexistence 2008 (pp. 339-342). Yokohama, Japan: ICAT 2008.
9. Herndon, K. P., van Dam, A., & Gleicher, M. (1994) The challenges of 3D interaction: a CHI '94 Workshop. In: ACM SIGCHI Bulletin (pp. 36 - 43). New York: ACM.
10. Sebok, A., Nystad, E., & Helgar, S. (2004) Navigation in desktop virtual environments: an evaluation. In: Virtual Reality (2004) 8 (pp. 26 - 40). London: Springer-Verlag.
11. Santos, B. S., Dias, P., Pimentel, A., & Silva, S. (2008) Head-mounted display versus desktop for 3D navigation in virtual reality: a user study. In: Multimed Tools Appl (2009) 41 (pp. 161 - 181). online: Springer Science + Business Media, LLC.