

Interaction Methods for Architecture in Virtual Environments

*Uwe Woessner, Joachim Kieferle, Johannes Drosdol
High Performance Computing Center Stuttgart; University of applied sciences,
Wiesbaden; DaimlerChrysler
www.hlrs.de; www.architektur.fh-wiesbaden.de; www.daimlerchrysler.com*

Virtual and Augmented Reality have become common tools in the architectural design process. This technology assists all participants in gaining planning certainty. To interact with the digital models, different input methods are used and adapted by the authors like keyboard, on screen menu, tangible interface, tablet PC and PDA. While every input method has certain advantages and disadvantages, currently tablet PC and PDA best cover the users' general needs. A usability test shows that reviews with tablet PC are faster compared to other interfaces.

Keywords: *Human-Computer Interaction, Interface, Virtual Reality, Augmented Reality*

Introduction

Over the past years, Virtual Reality and Augmented Reality (VR and AR) have become productive tools to communicate architecture between all participants in an architecture project. Especially in large projects, it has proven in research and commerce to improve the communication as well as the planning certainty. In order to explore the virtual representation of architecture, all participants have to be able to navigate through and interact with the virtual model as intuitive as possible, to use it as a tool to discuss and answer all questions. However, there is still a gap between this intention and reality. In the following, we will show and compare different input methods like keyboard, on screen menu, 3D-mouse, tangible interface, tablet PC and PDA to interact with the architectural model as well as simulation data in virtual environments like a CAVE

with the software COVISE / COVER.

How to interact in a virtual environment?

To judge the quality of input devices, one has to differentiate between

- the possibilities a virtual model supplies (interaction) and
- how to interact with the virtual environment (interface).

Both have to be well suited. The interaction is modelled into the digital model, e.g. which variations are built into the model, whereas the interface is defined by the hard- and software used.

There is a great variety of interesting interfaces developed and used by different VR groups. The

following paper only compares the interfaces for architectural interactions, which the authors use in their everyday work.

A rough overview characterizes the three devices:

- Keyboard using shortcuts: fast and easy access, limited overview.
- 3D-mouse and CAVE on screen menu: good for navigation within the virtual world, difficult to choose items from menu especially in high latency cases due to heavy models, limited overview.
- Tablet PC and PDA: fast and intuitive, good overview. Fusion of software interaction methods and model interaction methods.

Interactions

With the increasing number of project participants from different disciplines, the number of interactions necessary to answer all questions posed during a session increases as well.

To reflect the principle of general and specific interactions, the necessary interface information for general interactions are hard coded or implemented as native plug-ins in COVER and the model specific ones are defined in the VRML-code of the digital model. Some samples are shown below.

Samples of general interaction:

- Walk (3 degrees of freedom: move $[x,y]$ $z = \text{stay}$ on ground and heading)
- Drive (4 degrees of freedom: move $[x,y,z]$ and heading)
- Fly (6 degrees of freedom: move $[x,y,z]$, heading, pitch and roll)
- Scale model
- Apply clipping planes
- ... and others

Samples of model specific interactions:

- Different viewpoints
- Switching through variations
- Start animations

- Place new elements
- Modify elements (scale, move, rotate)
- ... and others

Implementing interactions in the software and digital models

Seamlessly integrating the interactions into a common interface, an interface reacting according to the model, is the main intention.

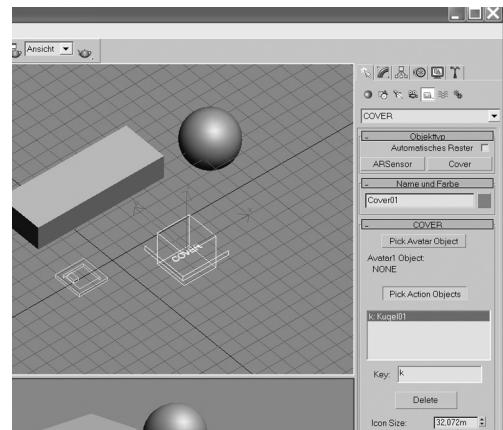
Most of the interactions are defined within the modelling tool, in this case 3D-Studio MAX. This leads to an automated workflow without the need of modifying files by hand. The standard VRML plug-in has been extended by state full sensors (toggle button), switch nodes, AR sensors and keyboard input to come closer to this automated workflow (see fig. 1). Some functionality like the ability to move objects just build on a naming convention und thus can be used even from within standard CAD systems."

Interfaces

CyberStick I

None of the commercially available input devices has been specifically designed for virtual environments. They have been derived from standard joysticks, desktop mice or game pads and are thus

Figure 1
Modified VRML Plug-in.



only suboptimal when used in an immersive virtual environment. Therefore, we decided to design our own 3D input device, the „CyberStick I“.

Common interactions in virtual environments are navigation, selection by “ray picking”, selection from options in a menu, as well as grabbing and moving objects. To be able to fulfil all these tasks with the same input device, an analogue joystick, a mouse wheel as well as multiple buttons were added to the input device. Further requirements have been to allow for one-handed interaction, at least three buttons, a trigger and shape to perform a pointing gesture as well as a vibration engine to be able to add force feedback to menu items or other input elements, suitability for both left- and right-handed interaction. After several design studies, a symmetric shape in the size of a typical remote control was chosen. The casing hosts a sensor for the magnetic tracking system, the electronics, an engine with an eccentric weight, an analogue joystick, an optical

wheel, a trigger and two additional buttons. A wireless version of this input device for optical tracking systems is currently under development.

On screen menu

The on screen menu is displayed in the virtual world as a flat 3D board. Submenus appear as additional boards. Buttons are “pressed” by pointing at the menu with the 3D mouse and clicking a button on the input device.

As the distance between the users hand and the menu can be several meters and due to the latency of the graphics hardware when viewing large models, it is sometimes difficult to hit the right button. However, in development phases, it is a useful tool, which can be turned on and off with a keyboard shortcut.

Tangible interface (Augmented Reality)

Integrating traditional techniques like architecture

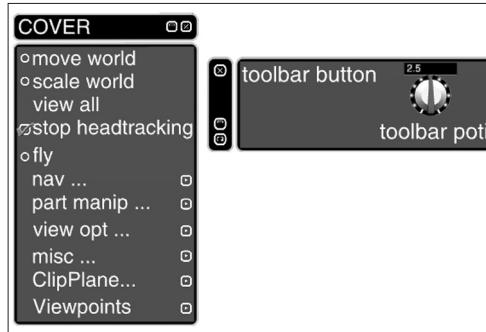


Figure 2
3D Mouse CyberStick I.

Figure 3
On screen menu and toolbar.

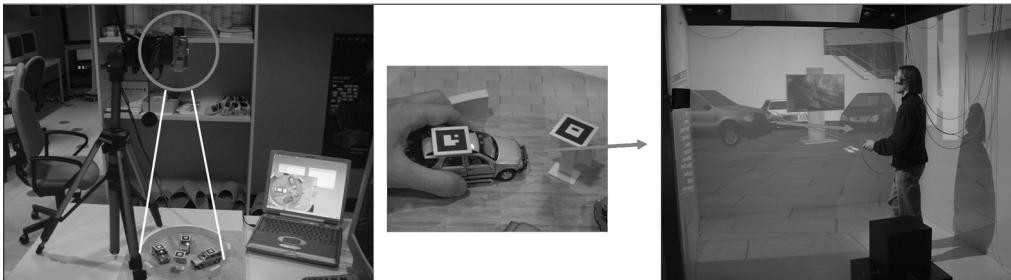


Figure 4
a) A camera tracks the architecture model. b) Moving the marked exhibition elements is c) transferred to the CAVE to move the virtual model.

Figure 5
A typical AR / VR session
setup.



models with VR-technology improves the design process and makes input more intuitive. A camera tracks markers attached to models, the movement of the models is translated into motion of virtual models in the CAVE. This e.g. allows composing exhibitions in model scale and reviewing them at scale 1:1. Removing elements from the real scene removes elements from the virtual scene. The handling of this tangible interface, compared to interaction with other input devices in VR, is more efficient and very easy to learn for novice users.

Keyboard shortcuts

Keyboard shortcuts (small and capital letters, func-

tion keys) are a very fast way to interact with models. However, one has to either work in pairs - asking the partner to press a key - or leave the CAVE environment to press the key oneself, which decreases the degree of immersion.

In the beginning keyboard shortcuts (like “F” to switch between facade variations, “v” to switch between video walls) seemed to be a good solution compared to the onscreen menu. With the growing size and complexity of the digital models, the number of shortcuts increased dramatically. For example, forty-four shortcuts in one model plus standard COVER shortcuts made it difficult to find the right key to press. One had to keep a printout of the shortcut list. Especially when working with different models it became confusing very fast.

Tablet PC

To overcome these keyboard input constraints, a tablet PC user interface was developed, to give the necessary overview. A WLAN connects the tablet PC with the virtual environment. The tablet PC user interface fuses the interactions provided by the software COVER (e.g. walk mode, fly mode, scale) as well as the ones provided by the model (e.g. switching through variations, scaling certain building elements, starting animations).

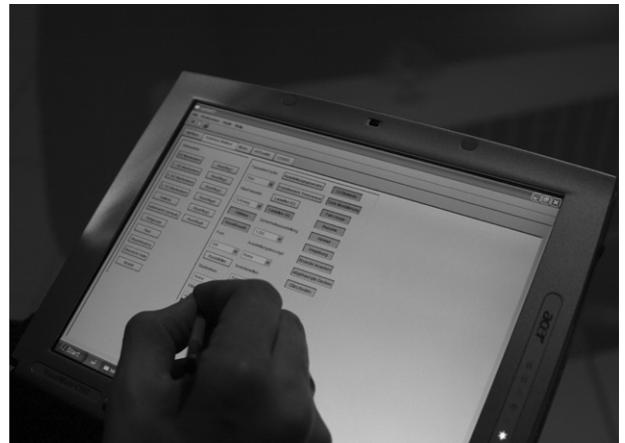


Figure 6
Tablet PC supported session.

Comparison keyboard and on screen menu versus tablet PC

A usability test with about forty novice users at a responsive workbench compares the performance and ease of use between the combination keyboard / on screen menu and a tablet PC. A typical architectural walkthrough with navigating through a building, rotating elements as well as applying two cutting surfaces on a scaled model shows that working with the tablet PC interface is about 36 % faster on average.

The better overview on the tablet PC assists the participants in finding the interaction functions faster and better (fig. 9-a). Its touch screen is precise and can be used very intuitively. Acceptance

of the tablet PC interface (quoting 1 or 2) is 83 %, whereas keyboard shortcuts and on screen menus only reach 62 %. Therefore, the general acceptance to use the interface over a longer time clearly shows the advantage of the tablet PC (fig. 9-b).

However, users still find the weight of the tablet PC too high; it should be reduced in the future to make carrying over a longer time more comfortable. Only 50% of the users participating in the about 20 minutes lasting usability test quote it as comfortable (fig. 10-a).

In the usability setup with a responsive workbench, the tablet PC tended to distract the users attention a little bit more from the scene, lowering the degree of immersion. This might come from having to switch

Figure 8
a) Finding functions is easy b) I can imagine working several hours with this interface; 1 = very true ... 5 = not at all.

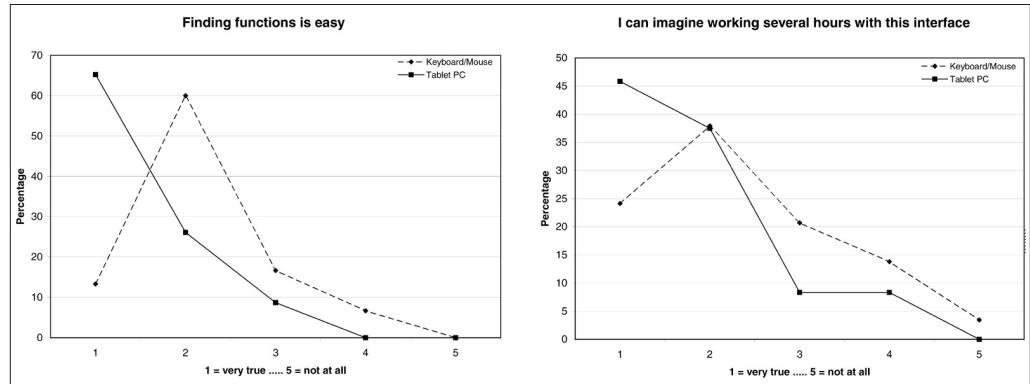
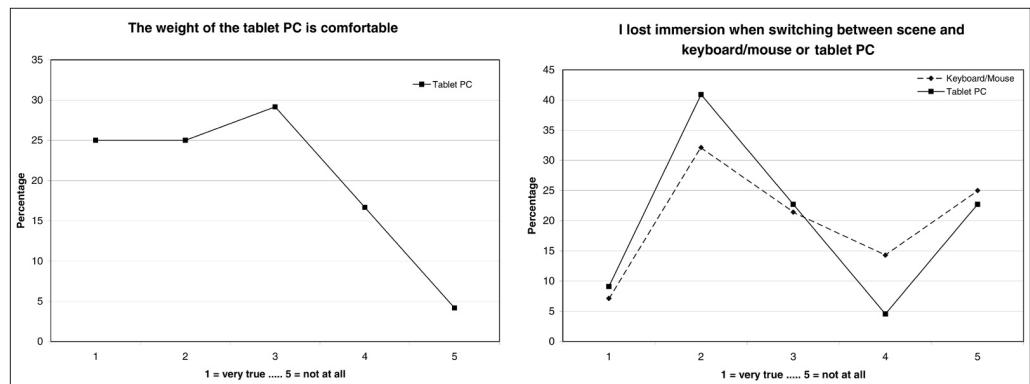


Figure 9
a) Weight of the tablet PC is comfortable b) I lost immersion when switching between scene and keyboard/mouse or Tablet PC; 1 = very true ... 5 = not at all.



between two screens (workbench and tablet PC) (fig. 10-b).

In a CAVE environment, where one would have to go out and into the CAVE to press the keyboard buttons or work in teams of two (one in the CAVE, the other at the keyboard), one can expect completely different results. This assumption has to be confirmed in future tests.

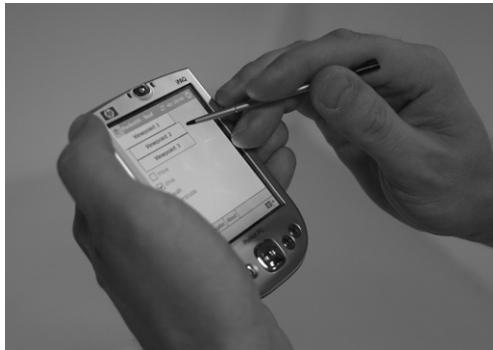
PDA - pocket PC

Similar to the tablet PC, the pocket PC user interface allows navigating through and interacting with the virtual worlds. A WLAN links the PDA with the CAVE. The limited PDA display resolution of 320 x 240 pixels and its display size currently makes the tablet PC the better choice. Nevertheless, further efforts will be made to make the PDA a common interface.

Conclusion and Outlook

Since interaction methods for architecture in VR mainly depend on the questions that one wants to answer with the virtual model, there is not "the" interface. Interaction with an onscreen menu in the virtual world itself and keyboard shortcuts have proven to be very effective for developers. However, for group discussion purposes it sometimes distracts the participants' attention.

Currently a wireless tablet PC due to its screen



size, variability and portability seems to be the most promising solution. Its weight makes it tiring to carry for a longer time. As soon as wireless PDAs will have a higher resolution and slightly bigger screen size, they might become the better choice.

Even though many interfaces and interactions with architecture models were tested, it is still a wide field for development. The users still should have a combination of flexible interfaces to choose the best ones for their specific needs.

References

- <http://www-i8.informatik.rwth-aachen.de/tdi03/talks/schirski.pdf>
- Bowman, D.A. (1998), Interaction Techniques for Immersive Virtual Environments: Design, Evaluation and Application, Human-Computer Interaction Consortium Conference '98 (HCIC).
- Donath, D. and Regenbrecht, H. (1999) Der Bleistift im 21. Jahrhundert. Das architektonische Entwerfen in interaktiven VR Umgebungen. (The pencil in the 21th century. The architectural design process in Virtual Environments), IAO Forum Architektur im Informationszeitalter, FhG Stuttgart, 21.-22.4.1999, proceedings, chapter 7, 10 p.
- Drosdol, J., Kieferle, J. and Wössner, U. (2003) The Integration of Virtual Reality (VR) into the Architectural Workflow - Experiences with an interprofessional project at DaimlerChrysler, Digital Design [21th eCAADe Conference Proceedings / ISBN 0-9541183-1-6] Graz (Austria) 17-20 September 2003, pp. 25-28.
- Kieferle, Joachim and Wössner, Uwe (2003) Combining Realities - Designing with Augmented and Virtual Reality, Digital Design [21th eCAADe Conference Proceedings / ISBN 0-9541183-1-6] Graz (Austria) 17-20 September 2003, pp. 29-32.
- Negroponte, Nicholas (1995) Total digital: die {Welt zwischen 0 und 1 oder Die Zukunft der Kommunikation}, München: Bertelsmann.
- Ogi, Tetsuro; Yamada, Toshio; Yamamoto, Koji, Hirose, Michitaka (2001) Invisible Interfaces for

*Figure 10
Wireless PDA with VR menu.*

- the Immersive Virtual World, Immersive Projection Technology and Virtual Environments 2001 [Proceedings of the Eurographics Workshop] Stuttgart (Germany) 16-18 May 2001, pp. 237-246.
- Preim, Bernhard (1999) Entwicklung Interaktiver Systeme, Springer: Berlin Heidelberg.
- Stoev, Stanislav; Schmalstieg, Dieter; Strasser, Wolfgang (2001) Two-Handed Through-the-Lens Techniques for Navigation in Virtual Environments, Immersive Projection Technology and Virtual Environments 2001 [Proceedings of the Eurographics Workshop] Stuttgart (Germany) 16-18 May 2001, pp. 51-60
- Volbracht, Sabine; Domik, Gitta (2000) Developing Effective Navigation Techniques in Virtual 3D Environments, Virtual Environments 2000 [Proceedings of the Eurographics Workshop] Amsterdam (Netherlands) 1-2 June 2000, pp. 55-64.