

# Sensing Surfaces - Bringing the Desktop into Virtual Environments

Matthias Bues, Roland Blach, Frank Haselberger<sup>1</sup>,

<sup>1</sup> Fraunhofer Institute for Industrial Engineering, Stuttgart, Germany

---

## Abstract

*There are many classes of information for which a suitable 3-D representation does not (yet) exist, and for which users are familiar with more or less efficient desktop interfaces. Therefore, a convenient way of accessing GUI-based systems from within a Virtual Environment can greatly enhance application possibilities and leverage integration with other IT facilities. We describe Sensing Surfaces, which follows the approach of mapping the contents of a GUI desktop to arbitrary textured geometry in the VE.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism : Virtual Reality

---

## 1. Introduction

As Virtual Environments continue to enter more areas of productive application, the need for a convenient access to non-visual information becomes increasingly important. Although the long-term objective of 3-D user interface research is to provide a 3-D representation for most categories of information, there remain many cases where the conventional 2-D desktop interfaces are quite suitable for the task at hand. As long as 3-D interfaces and representations for these do not really exist, importing 2-D interfaces into the VE has some considerable advantages:

- Transparent use of functionality of GUI-based applications
- Delegation of functional interface handling to a independent (remote) component
- Users can work with their familiar applications and user interfaces

From the alternative of taking along an additional display, e.g. a web pad or a PDA, into the VE, or using the immersive display of the VE itself, we pursued the second option, for the main reason of not wanting to encumber the user with additional gear. Concerning input devices, we investigated various methods of pointing and numerical/text input. An important objective was maximum flexibility concerning which GUI-based applications can be integrated in the VE. This suggests doing the distribution on a very low protocol

level, i.e. distributing the display by its framebuffer contents and sending pointing device and keyboard events. A widely used implementation of this distribution approach is Virtual Network Computing (VNC)<sup>2</sup> which we chose as the distribution method for our implementation.

## 2. Related Work

There have been several approaches of integrating 2D-desktop applications in virtual environments. Two different directions can be observed:

- Hardware oriented: Integration of handheld and sometimes see-through-displays
- Software oriented: Projecting image content onto planes in 3D (2D-primitives, texture mapping). This can be further distinguished by the system component performing the 2D rendering:
  - Integrating 2D-image generating functionality (type-setting, linedrawing, etc.) in the virtual environment operating software
  - Accessing 2D display content generated by other, probably remote, systems

The hardware oriented approach was addressed by several researchers, e.g. Watsen et. al<sup>4</sup>. A basic problem with the integration of additional display hardware in projection based immersive virtual environments is the different contrast,

brightness and distance of the two displays. A clear advantage of hardware based access is the haptic user interaction with the 2D system. On the software-based side, there were some examples of integrating GUI-based application functionality in the virtual environment operating software, one being VRMosaic which integrates a WWW browser into a Virtual Environment. <sup>1</sup>. However, this type of application-level integration limits its use to content and applications accessible through the HTTP protocol.

### 3. Implementation

The Sensing Surfaces concept is integrated in our virtual environment development system *Lightning L3*, the successor to *Lightning* <sup>3</sup>. The implementation is based on VNC (Virtual Network Computing) <sup>2</sup>. VNC provides a distribution mechanism for GUI-based environments on the lowest conceivable level by transmitting frame buffer contents to the remote client and receiving keyboard and pointing device events, inserting these into the server-side input queue. This approach allows it to support many platforms, including X and even Microsoft Windows. We did the integration by building a VNC client which writes its output to a texture which can then be mapped onto arbitrary geometric objects in the scene. By supporting the full RFB protocol, the Sensing Surface VNC client can be used with any VNC server.

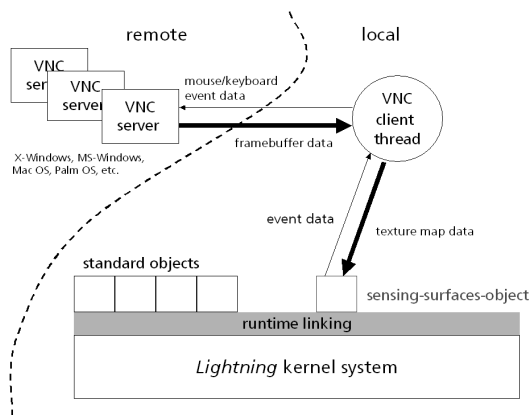


Figure 1: Integration of the Sensing Surfaces Component

## 4. Case Studies

### 4.1. Controlling the VE application

Many VE applications have their user interface split into two parts: Some of the functionality is accessible through a 3-D user interface, but the rest can only be controlled through a conventional GUI running on a console outside the VE. Using a Sensing Surface, this part can be taken inside the VE without almost no modification.

### 4.2. Enhancing VE-based CAD evaluation

In product engineering applications, the use of Sensing Surfaces enhances productivity by making the functionality of the CAx desktop accessible from within the Virtual Environment. In a VR-CAD-Evaluation session, the engineer has direct access to the original drawing of the visualized product by accessing his workstation which runs the CAD system through the Sensing Surface.

### 4.3. Information Visualization

Sensing surfaces was integrated in an information visualization application where a SAP R/3 system was connected to an immersive virtual environment. Users can navigate through this database structure, load 3D geometry, and attach annotation data. Sensing Surfaces provided an interface to the text-based 2D-GUI of the SAP/R3 system to retrieve and enter textual information.

## 5. Conclusion and Future Work

The results of our work done so far indicate that the seamless integration of conventional GUIs with VE can greatly enhance the possibilities of many VE applications. The transparent and loosely coupled cross-linking allows integration with many application types and systems with minimum implementation effort. Future directions of our work include investigation of other input devices like a Trackpoint as familiar from notebooks and one-handed keyboards like the Octima.

## References

1. I. G. Angus, H. A. Sowizral, VRMosaic: Web Access from within a Virtual Environment *IEEE Computer Graphics and Applications*, Vol. 16, No.3, pp. 6-10
2. T. Richardson, Q. Stafford-Fraser, K. R. Wood, A. Hopper, Virtual Network Computing. *IEEE Internet Computing*, Vol. 2, No. 1, Jan/Feb 1998, pp 33-38
3. R. Blach, J. Landauer, A. Rösch, A. Simon, A Flexible Prototyping Tool for 3D Real-Time User-Interaction. *Proceedings of the Eurographics Workshop on Virtual Environments 98*, 1998 pp. 195-203
4. K. Watsen, R. Darken, M. Capps, A Handheld Computer as an Interaction Device to a Virtual Environment. *Proceedings of the International Projection Technologies Workshop*, Stuttgart, Germany, May 10-11, 1999