Chapter 1

Virtual Environments

In this chapter a detailed survey on the current technology in Virtual Environments (VE) that allows the highest degree of intuitive interaction is given. This includes various display systems and input devices as well as most important software toolkits.

1.1 Display Systems

According to [44] there exist three different classes of display systems. CRT-based displays such as Head-Mounted Displays, Virtual Model Displays such as Responsive Workbenches and ReachINs, and Spatially Immersive Displays (SID) such as CAVEs and DOMEs. It is unreasonable to assume that a single display device could support any or all VE tasks equally well. Instead it is obvious from the proliferation of display types that most display types are perfectly suited for some task, sufficient for some other tasks, and ill-suited, impossible, or intractable for others. Thus, to assess the effectiveness of a display device is to assess how well the device matches representative user tasks.

1.1.1 Head Mounted Displays (HMD)

In its simplest form, an HMD consists of an image source and collimating optics in a head mount [94]. The HMD can then become more elaborate in several ways. There are one or two display channels. These channels display graphics and symbology with or without video overlay. They can be viewed directly and occlude external vision for a fully immersive experience, or a semitransparent combiner is used with see-through capabilities to the outside world. In this augmented reality mode, the HMD is able to overlay symbology, or other information on the world view. The HMD image source can either be CRT

(Cathode Ray Tube (small TV-tube)), LCD (Liquid Crystal Display) or in the near future RSD (Retinal Scanning display) mounted on the head, or the image is brought up to the head through a fiber-optic bundle. An HMD uses a simple headband for mounting on the head, or the optics and the displays are integrated into an aviator's flight helmet. This latter device is a specialized case of the HMD, called the helmet mounted display. HMDs have often been used to display Virtual Environments in the past. Nowadays they are not used very often due to their uncomfortable usage, the level of contamination of the CRT close to the eyes and the low resolution. However, in Augmented Reality see-through HMDs are still used quite often.
Typical application areas come from the manufacturing industries where real objects are overlayed with virtual information providing, for example, assembling aid.

1.1.2 ReachIn Display

The ReachIn display is a unique Human-Computer Interface (see Figure 1.1). The system combines a stereo-visual display, a haptic device and a six degree of freedom positioner. The innovative use of a semi-transparent mirror creates an interface where graphics and haptics are co-located. The user interacts with the virtual world using one hand for navigation and control and the other hand to touch and feel the virtual objects. It is possible to see and feel the object in the same place.
The description of an excellent interface design for the ReachIn display system

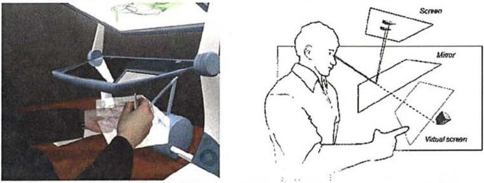

Figure 1.1: The ReachIn is a very intuitive display system for many kinds of applications. [ReachIn Inc.,2001]

can be found in [76]. ReachIn displays are typically used for medical training, product development and design engineering.

1.1.3 Responsive Workbench

The Responsive Workbench in Figure 1.2 is a high resolution tabletop display system which has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)* [63, 64]. The Responsive Workbench consists of a RGB projector

Figure 1.2: The Responsive Workbench provides a natural metaphor for visualizing and interacting with three-dimensional computer-generated scenes. On the right hand side two engineers are working collaboratively in front of the same display.

which projects stereo images over a mirror onto a display plane. This projection plane is sloped by 20°. A separate image is computed for each eye, and the computer quickly alternates the display of the two views. Users wear shutter glasses, which cover the left eye while the right eyes image is displayed, and vice versa, thus producing the stereoscopic effect. A Polhemus six degrees of freedom sensor is attached to the shutter glasses for head tracking. This allows the system to compute the correct perspective image for any user location. The users interact directly with three dimensional virtual objects within the viewing frustum using six degrees of freedom input devices.
Typical application areas of the Responsive Workbench are delivered by engineering and medical visualization.

1.1.4 Two-sided Responsive Workbench

The new two-sided Responsive Workbench has a horizontal and a vertical display, which are smoothly adjacent (Figure 1.3). By extending the normal

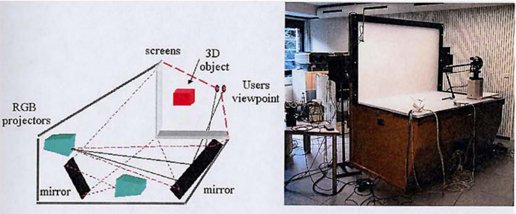

Figure 1.3: The two-sided Responsive Workbench is extended by an additional vertical screen. Thus the viewing frustum is substantially increased and virtual objects can be observed at the user's eye level. On the right hand side a Phantom Force Feedback device is integrated into the RWB's interaction space.

Responsive Workbench with an additional vertical screen, the viewing frustum is substantially increased and virtual objects can be observed at the user's eye level which was not possible before. Using current PC and workstation technology the two-sided Responsive Workbench is operated with a resolution of $1280 * 1024 * 96Hz$ stereo. Because of the extended viewing frustum the two-sided Responsive Workbench can be used in more application areas than the one-sided Responsive Workbench.
The two-sided Responsive Workbench can be used for engineering and medical visualization like the one-sided Responsive Workbench. Product design as well as various training applications are also applicable on the two-sided RWB.
It is important to mention that throughout this thesis the display system term RWB is refereing to the two-sided Responsive Workbench instead of the one-sided older version.

1.1.5 CAVE - CyberStage

The CyberStage, an improved version of the CAVE [27] has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)*. The immersive features of the

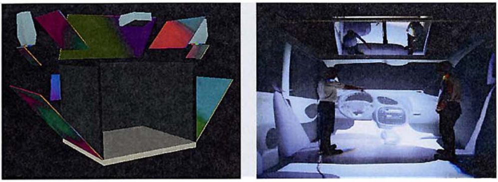

Figure 1.4: The CyberStage uses three mirrors for the side projections and either one mirror for the floor and front projection. Four sub-woofers are built into the floor allowing for rendering low frequency signals perceivable through feet and legs. On the right hand side two user's are working on the Volkswagen Sharan interior design.

CyberStage are based on four-side stereo image projection and eight-channel spatial sound projection both controlled by the position of the user's head followed by a tracking system. The sound projection is complemented by vibration emitters built into the floor of the system and allow for rendering low frequency signals perceivable through feet and legs. Therefore the CyberStage can create the illusion of presence in virtual spaces using various interfaces and interaction metaphors which visually and acoustically respond to the users' actions. These interfaces allow for navigation in virtual spaces and manipulation of virtual objects.
The design of the CyberStage was aiming at providing a high degree of immersion and presence for a wide spectrum of Virtual Environments and it has already been used in the following application areas:

- scientific visualization/sonification
- industrial data exploration
- product presentation and marketing
- product design and evaluation
- architectural planning and design
- exhibition design and evaluation
- media art installations and promotional infotainment

1.1.6 Powerwall

The Powerwall is a huge multi channel rear projection screen. It can have dimensions up to 8m wide and 3m high. The Powerwall consists of mostly two or three segments, where each segment is projected by one or two graphic channels, which in turn are powered by one or two projectors. Therefore in addition to three channel mono displays and six channel stereo displays it is also possible to have a six channel mono display. By using double projection for each segment, strong light-intensity and high brilliance is achieved. Light reflections and surface dispersions can be analyzed as good as with real finished models.
Powerwalls are typically used in automotive industry. Especially product designers have a suitable capability to visualize a complete car in a 1:1 scale.

1.1.7 Cylindrical Displays

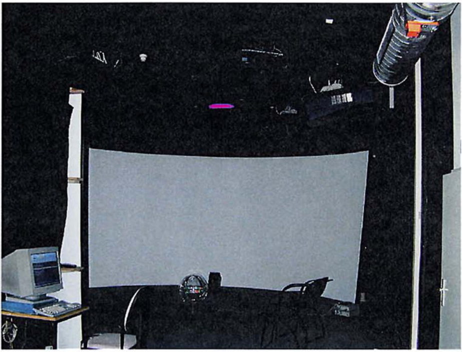

Figure 1.5: GMD's cylindrical display system is a 230° projection with 7m diameter.

Cylindrical displays are multi channel panoramic projections on cylindrical or spherical curved surfaces with a field of view up to 360°. Similar to the Powerwall, the Cylinder consists of multiple (mostly 3 for 180°) segments,

where each segment is projected by one channel powered by one projector. *Soft-edge-blending* creates a seamless and invisible integration of the separate images to form a unique full screen image of a very high quality. Cylindrical displays can be driven with analogue CRT projectors or with digital DLP projectors.
Cylindrical projection displays are ideally suited for a variety of applications in the presentation and simulation area due to their capabilities to host groups of up to 10 users.

1.2 Input Devices

There exist many input devices as many graphics applications require the input of three dimensional co-ordinates in order to position objects in a virtual space. These devices can be classified into devices for the hands and for the feet. Devices which are used by the hands are: keyboard, mouse, joystick, trackball, stylus, touch screen, digitizer (graphics tablet), data and pinch glove, space mouse, cubic mouse, multiple button tools etc.
Input devices used with the feet are: Treadmill (for navigation simulation in Virtual Environments), driving wheels including pedals (for flight and racing games or simulators).
Not belonging to one of these two groups microphones and video cameras are also considered to be input devices.
Desktop applications typically use a mouse, trackball or joysticks for such input. Applications developed for Head-mounted displays mostly use gloves such as pinch gloves or data gloves. A variety of input devices are used or developed especially for projection based display systems. In this section a survey of the most important input devices is provided including multi-sensory feedback.

1.2.1 Pen-like Input Devices

In order to interact within a Virtual Environment, devices are needed to track the location of real objects. As this world is three dimensional, three measures should be used (X,Y and Z). The mouse and the joystick are examples of such tracked control devices. However, in Virtual Environments it is not sufficient to know the position of the input device only, the orientation is needed too. Sensors which measure position and orientation are usually electromagnetical or optical devices. But also ultra-sound based systems and inertial trackers are available.

Figure 1.6: The left image shows a pen-like stylus input device from Polhemus. On the right hand side pinch gloves are shown.

1.2.2 Pinch Gloves

Another common control device is the pinch glove (see Figure 1.6). This pair of stretch-fabric gloves contains sensors in each fingertip which detect contact between the digits of the user's hand. These gestures can be used for a wide range of control and interactive functions. Any combination of single or multiple contacts between two or more digits can be programmed to have specific meanings, ranging from simple on/off to multi-part, multi-action commands. Thus the pinch glove system provides a reliable and low-cost method of recognizing natural gestures. Recognizable gestures have natural meaning to the user: a pinching gesture can be used to grab a virtual object, and a finger snap between the middle finger and thumb can be used to initiate an action.
As it is very easy to integrate, the pinch glove system can be integrated into driving and flight simulators, interactive 3D video games or any application that requires a wide range of tactile gestures.

1.2.3 Cubic Mouse

The Cubic Mouse is a novel and intuitive input device which has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)* [41, 42]. This device consists of a cube-shaped case, three rods, and controls buttons (see Figure 1.7). Each rod passes through the center of two parallel faces of the case. The rods are perpendicular to each other and movable. They represent the X, Y, and Z

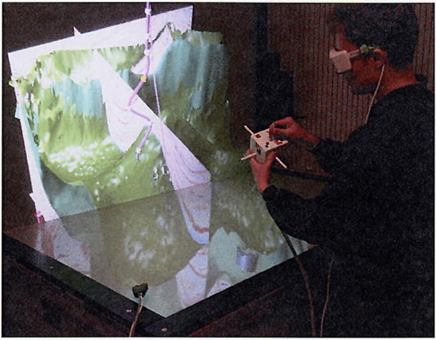

Figure 1.7: The user controls a geo-scientific scenario at the Responsive Workbench using the Cubic Mouse device.

axes of a co-ordinate system. In addition to the rods a six degree of freedom tracker sensor is embedded in the cube-shaped case. This tracker sensor is used to orient and position the virtual world in three dimensional space relative to the observer. In this way the rods stay aligned with the co-ordinate system axes. By pushing and pulling the rods motions of virtual objects are specified constrained along the X, Y, and Z axes. Typically the users hold the device in their non-dominant hand to position and orient the world, while the dominant hand operates the rods and the control buttons.

The Cubic Mouse is typically used for controlling volumetric data sets in geo-science and automotive applications. Selection tasks are rather complicated to perform.

1.2.4 Phantom Force Feedback Device

Force feedback provides direct perception of three-dimensional objects and directly couples input and output between the computer and user. It acts as a powerful addition to graphics display for problems that involve understanding of 3D structure, shape perception, and force fields [22]. There exist many force and tactile feedback devices. A well known one is the Phantom Force Feedback device from Sensable Inc. (see Figure 1.3 and Figure 1.8). The Phantom

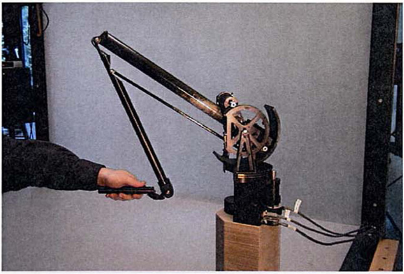

Figure 1.8: The Phantom Force Feedback Device integrated into the Responsive Workbench's interaction space. [Sensable Inc.,2001]

six degree of freedom prototype is a desk mounted force feedback system that provides six degree of freedom force and torque feedback. The system consists of the device itself, accompanying power electronics, a remote/safety switch, and a PCI interface card. The device is leveraged from haptic technology developed and implemented in standard Phantom three degree of freedom force feedback devices.
The addition of force display to computer graphic systems has produced a factor of two improvement in rigid-body motions during simulated drug docking, enabled new types of manipulation, and provided users with a stronger sense of understanding [82]. For surgery training applications tactile and force feedback is of essential importance.

1.3 Virtual Environment Software Toolkits

Many toolkits for the development of stand-alone VE applications exist today. They provide the programmer with a high-level interface to represent complex geometry in a scene graph and to render that scene graph. The programmer is shielded from the details of dealing with low-level graphics and system APIs, and can concentrate on the development of the application itself. Several attempts to offer similar toolkits for distributed VE application development have been made in recent years (eg. DIVE [24], VR Juggler [5], WTK [84, 85]). This section provides a survey on some important VE software toolkits. The particular reason for choosing these software toolkits is their uniqueness with

respect to their special capabilities for implementing Virtual Environments.
VR Juggler and Avango are unique in terms of their real-time rendering capabilities. Also the easy configuration for different display systems and input devices make them being different compared to other software toolkits such as e.g. Massive etc.
Also the scripting interface that allows for run-time modifications and easy prototyping is a outstanding feature.
DIVE on the other hand is a VR software toolkit that enables users to share and manipulate data very effectively and further allows different types of communication.
The following sections are presenting the mentioned toolkits in more detail.

1.3.1 DIVE

The Distributed Interactive Virtual Environment (DIVE) is an internet-based multi-user VR system where participants navigate in three dimensional space and see, meet and interact with other users and applications [24]. The key requirements for a Virtual Environment development system that the DIVE's design addresses are: performance, extensibility, and portability.
The first DIVE version appeared in 1991. The DIVE software is a research prototype covered by licenses. Binaries for non-commercial use, however, are freely available for a number of platforms. The software supports the development of Virtual Environments, user interfaces and shared synthetic applications. DIVE is especially tuned to multi-user applications, where several networked participants interact over a network. Dynamic behaviour of objects are described by interpretative Tcl scripts evaluated on any node where the object is replicated. Scripts are triggered by events in the system, such as user interaction signals, timers, collisions, etc. DIVE imports and exports VRML and several other formats. It is integrated with the World-Wide-Web and is HTTP/FTP/HTML/MIME compliant.
DIVE applications and activities include virtual battlefields, spatial models of interaction, virtual agents, real-world robot control and multi-modal interaction [40]. Through its various versions, DIVE is available on the almost all platforms.

1.3.2 VR Juggler

VR Juggler is an open source virtual platform for virtual reality application development created at the Virtual Reality Applications Center at Iowa State University [5] and is the follow-up of the CaveLib project, although the new architecture is very different.

Applications developed in VR Juggler can transparently move between a wide range of VR systems, operating systems, and graphics APIs. VR juggler's design addresses several key requirements for a Virtual Environment development system namely, performance, flexibility, rapid application development, extensibility, and portability.
Flexibility is achieved by placing common abstractions over I/O devices. New devices can be added easily, and existing devices can be reconfigured or replaced, even while an application is running.
To allow optimal performance, applications are given direct access to graphics APIs, currently including OpenGL and Iris Performer [88, 89]. VR Juggler includes built-in support for performance monitoring of applications and graphics subsystems. It supports multiple-processor machines and will support distributing applications across multiple machines. Small base classes provide a skeleton for application development, while the abstractions of I/O devices simplify programming. During run-time, any VR Juggler application can be controlled or reconfigured by a Java-based graphical interface. VR Juggler is designed to support a wide array of VE hardware on a variety of architectures. Several tracking systems, gloves, and input devices are already supported. VR Juggler supports projection-based displays, and includes support for head-mounted devices.

1.3.3 AVANGO

Avango is a software framework which has been developed at the Department of Virtual Environments of the *German National Research Center for Information Technology (GMD)* [96]. This section is introducing the main concepts of Avango. A more specific description of the Avango C++ and scripting API is provided in chapter 5. Additionally the way distribution is carried out is discussed there.
Avango was designed for building interactive Virtual Environment applications for all kinds of display systems. Avango's design addresses the following requirements for a VE development system: performance, flexibility, rapid application development, extensibility, and portability. Applications in Avango are typically a collection of scheme scripts, which create and manipulate Avango objects and define relationships between them (Figure 1.9). Objects in Avango are field containers, representing object state information as a collection of fields, similar to Inventor [102]. These objects support a generic streaming interface for storing and retrieving an object and its state to and from a stream. This is the basic building block for object distribution. While objects in Avango are implemented using the C++ programming language, Avango also features a scripting environment used for customizing the

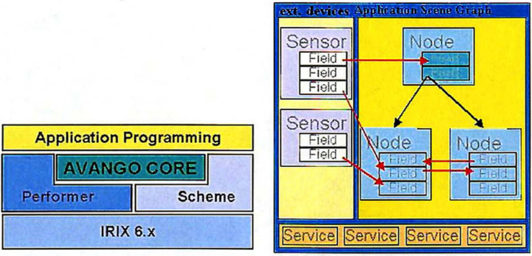

Figure 1.9: The left image shows the different modules on which the Avango core is positioned. The right images shows the Avango scene graph which consists of services, sensors and nodes.

runtime environment (e.g. setup of input and output devices) and developing applications. Avango includes the following concepts:

Viewer

All kinds of configurations of input and output devices can be assembled to viewers. The viewer is the interface between the user and the virtual world. Typical elements of a viewer are the visual, auditory and tactile displays as output devices and spatial trackers, audio and video sources as input devices.

Scripting

In addition to the C++ Application Programming Interface (API), Avango features a complete language binding to the interpreted language scheme. Scheme is an interpreted programming language descended from Algol and Lisp. It is a high-level language, supporting operations on structured data such as strings, lists, and vectors. This makes it possible to specify and change scene content, viewer features and object functionality and behaviour at runtime because all high-level Avango objects can be created and manipulated from Scheme. Script examples provided in chapter 5 given an idea of how applications are implemented with Avango.

Streaming

All objects are able to read and write their state from and to a stream. This is the basic facility needed to implement object persistence and network distribution. Persistence together with streaming support for objects make it possible to write the complete state of the system to a disk file at any time.

Distribution

All Avango objects are distributable, and their state can be shared by any number of participating viewers. Object creation, deletion and all changes at one site are immediately distributed to others. Details of distributed Avango are discussed together with the scripting interface in chapter 5.

Extensions

The system is extendible by subclassing of existing C++ system classes. This concerns objects as well as classes which encapsulate viewer features. Compiled extensions can be loaded into the system at runtime via Dynamically Shared Objects (DSO).

Interaction

Viewers provide input/output services which can be mapped to objects in the scene. Objects respond to events generated from input devices or other objects and deliver events to output devices.

Visual Rendering

Different displays have their appropriate rendering mechanism applied to the modelling hierarchy. Only the visual rendering has a direct access through the SGI Performer pipeline. The auditory and tactile rendering can be calculated on another computer connected to the master by a suitable network. The visual data processing is organized in a pipeline and computed in parallel by Performer. This rendering pipeline consists of a set of units, for:

- a database connection
- a user application
- the visual culling of the scene
- the intersection of objects
- the drawing of the scene

After the modelling hierarchy is updated to its actual state in the application process it is passed on to the culling process which strips all invisible objects. It is important to support this technique by dividing large geometry into smaller, cullable parts. The remaining part of the scene is passed on to the drawing process where it gets rendered on the screen using OpenGL. For configurations with more than one screen, the appropriate number of pipelines is used.

This chapter presented the basic technological components of Virtual Environments. The next chapter is focussing on interaction techniques and metaphors in VEs. It presents in detail related work to this thesis.