Survey of Virtual Environment Technologies and Techniques

Steve Bryson
Computer Sciences Corporation
Applied Research Branch
Numerical Aerodynamic Simulation Systems Division
MS T045-1 NASA Ames Research Center
Moffett Field, Ca. 94035-1000
bryson@nas.nasa.gov

1 Introduction

1.1 What is an Immersive Virtual Environment, and How do I Get One?

Immersive virtual environments is the integration of computer graphics and various input and display technologies to create the *illusion of immersion* in a computer generated reality. The user interacts with a world containing seemingly real, three-dimensional objects in three-dimensional space that respond interactively to each other and the user. The methods, techniques and technologies involved in attaining this illusion are the subject of this course. The techniques discussed are based on currently available commercial technologies, from the perspective of immersive systems that have actually been developed.

The attainment of the illusion of immersion place severe demands on all elements of a computer system. The display must enhance the illusion, the computer system must sense the user as unobtrusively and naturally as possible, and the entire illusion must 'hang together'. It turns out to be quite difficult to create systems that have the required performance to sustain the illusion of immersion. The usual methods of designing integrated interactive systems are, while important and relevant, not sufficient to create working immersive virtual environments. Notable success has been attained in 'hard-wired' applications, and quite a bit of work has been done in the construction of 'virtual reality design systems'. One lesson that stands out is that at the current state of technology the successful creation of the illusion of immersion is a delicate art, in which many parameters of a system have to be 'just right'. In this course the designers of both applications and general systems will discuss the various design philosophies and approaches that have proven successful. There is a great deal of research that must be done to make a real science out of immersive virtual environment design. Until that research is carried out, it is the experience of those who have grappled with the illusion of immersion that can provide the insights needed for more and more successful systems to be developed. Thus the presentations in this course will be based on personal experience, and so will be filled with largely anecdotal accounts of what works and what does not.

A note on nomenclature: I often use the phrases “Immersive virtual environments”, “virtual environments”, “virtual reality”, and “VR” interchangeably. When I use any of these phrases, I always mean “immersive virtual environment” as described in the first paragraph. This is somewhat different from what has become the common usage of the phrase “virtual reality”, which has been used to describe just about anything in computer graphics, and sometimes the use of computer graphics to mimic the real world. As Ted Nelson pointed out during a SIGGRAPH’89 panel, all of computer interaction involves virtual reality. While this is true if the words “virtual reality” are taken literally, this is not what was meant by those who started the virtual reality field. Thus, for example, in my usage of the phrase, photo-realistic rendering, holograms, and anything requiring off-line rendering of frames do not count.

What is special about immersive virtual environments is the paradigm they introduce into computer-human interfaces: to interact with a computer-generated thing, as opposed to a computer-generated picture of a thing. This paradigm goes almost as far back as three-dimensional computer graphics, but it is only in the last few years that the paradigm has become feasible as an actual interface. Another perspective on this new paradigm is that we are controlling objects in the computer environment directly, not through buttons, sliders, keyboards, or mice. We are reaching out and grabbing, pointing, pushing, and talking to things just as we do in the real world. This paradigm integrates interactive devices, computation, and display in a way previously found only in certain classes of applications.

Immersive virtual environments involve a whole new set of constraints. The primary constraint is one of speed. All computation and rendering involved in virtual environments must take place at high speeds (at least ten frames/second) to support the illusion of immersion. The overhead involved in high-level programming constructs often introduce unacceptably slow performance. Another constraint is the desire for natural interaction. These kind of constraints have been addressed in fields such as flight simulation and action game design. The methods of meeting these constraints in these cases are, however, often very expensive, always special case, and usually highly constraining on the kinds of things the programs could do. Immersive virtual environment designers can certainly learn from these other fields. The task of the virtual environment field is to learn how to satisfy these and other constraints in ways that can be applied to any application which would find an immersive virtual environment interface useful. It is almost certainly too soon to expect a single method or design philosophy to be developed for all cases, but hopefully we will soon see a general set of guidelines which show how to develop a working virtual environment system.

Speaking personally, I feel rather strongly that the development of virtual environment systems is a very different and new branch of computer science. It has yet to develop an integrative theory of design. In my experience I have many times thought that I understood how to create a virtual environment only to discover that there is some crucial aspect that I had not accounted for. Things like this are only discovered during the development phase and often force significant design modifications. Every time I think I know a paradigm for virtual environment development, the very next application I try to build does not fit that paradigm. I have seen many of my colleagues have the same experience. In a sense virtual environments is a field with very little data. There are very few working applications, and fewer controlled experiments. We do, however, have a sizable body of isolated facts and principles of what works and what does not. It is my hope that this course contributes to the widespread design of interesting, working immersive virtual environment applications.

There is a moral here: Every new immersive virtual environment application project is embarking on unknown and poorly understood ground, and is more a research project than an engineering project. Keep that in mind when promising schedules to people.

OK, so I've made it sound really hard. Sometimes it is. Yet the payoff in terms of new ways of using computers to do real work is truly fantastic. It is also one of the most fun fields in computers today.

1.2 A Brief Outline of What It Takes To Feel Immersed

To feel immersed, you must feel that there are things all around you. You must take these things literally as objects with position and properties. One good test of the illusion of immersion is the following: if you turn away from something in an environment, that thing should feel like it is still there, just out of your field of vision. In this case, the object has a strong sense of presence. If the things in the environment have behavior and react to your motions, you will take those objects more seriously as actual things and the illusion of immersion will be enhanced. The task of virtual environment systems is the creation of the illusion of immersion using various control and display devices hooked up to graphics computer.

In the classic (more than three years old) realization, virtual environments use head tracked, wide field stereoscopic displays that are coupled to the user's head. This allows the computer graphics system to correctly render the computer generated scene from the user's point of view. The display should completely fill the field of view, blocking out real-world dissonant images. Anecdotal evidence indicates that when a scene fills up more than 60 degrees of your view you become more involved in the scene, so wide field of view is important. When the you turn your head in such a system, the scene is drawn from the correct point of view. Objects seem to stand still while you turn your head, exactly as in the real world. In these systems, the user typically controls the environment through 'touching' the virtual objects via some direct interface, mimicking the real world interaction of picking up real objects. This is usually performed via an instrumented glove which tells the computer the position of the user's hand and the gesture the user may be making.

In order to create the illusion of immersion in a convincing world, it is not necessary to draw graphics that 'looks real'. Photo-realistic rendering is typically not necessary to create the illusion of reality. The key idea is that the virtual environment system is not trying to mimic the real world. It is trying to create a different world that one interacts with as if it were real. This requires the virtual world to have the behavior of a real world, and as one walks around one should see this virtual world from your moving point of view. This puts extremely demanding constraints on the virtual environment system: The computer must run at high speeds, performing any drawing and computation necessary in, at most, a tenth of a second. With slower performance the images look like successive pictures rather than a view into an environment. The hardware that tracks the position of the user must be simultaneously fast, accurate, and unencumbering. The display system must fill the user's field of view no matter what direction the user is looking, while presenting a high resolution display.

The demand of running at ten frames/second drastically limits the number of objects that may be rendered and the detail with which they may be drawn. The fastest computer graphics systems available today advertise drawing speeds of a million polygons/second. This means, for a system that runs at ten frames/second, that the scene can only contain at most 100,000 polygons. Actual performance is considerably slower than this. You will not be able to put very much detail in a scene which fills your field of view with less than 100,000 polygons.

Today systems exist that perform these tasks fairly well, though by no means as well as one would like. They are rather expensive and are still very much in the research phase. Subsets of virtual environment technology such as stereo workstation displays, hand tracking with gloves as three-dimensional mice, or head tracking to 'look around' an object can also be used in otherwise conventional workstation environments.

1.3 Why should you want a virtual environment?

Because they're cool, right? They are, but unless you actually enjoy doing nothing but giving LOTS of demos, you should have a good reason for building them. Virtual environments are a tool, and like all tools there are uses that make sense and uses that do not. Implementing a virtual environment is a lot of work, so before you start an implementation you should think about whether or not it makes sense.

Virtual reality has been the subject of a great deal of hype lately. Many claims have been made about how VR can be used to solve all sorts of problems. While many of these promises are quite realistic in the long run, current technology is very limited, and this limits the kinds of problems that VR can solve. Many obvious applications of virtual environment technology are beyond the reach of current technology. There are, however, many other applications that can be implemented in a virtual environment context now. Identifying these applications is surprisingly tricky. An application that almost works but contains some disabling bottleneck that cannot be avoided will not work well in a virtual environment. Anticipating these bottlenecks can be difficult. Which types of applications benefit from virtual environments is an open question. Much research, in the form of trying different applications, remains to be done.

There are some things that can be said at this point. Virtual environments almost certainly help if your application involves three-dimensions in some way. Virtual environment displays give a very good sense of the three-dimensional structure of objects. If your application requires three-dimensional control, then you will almost certainly benefit from the input aspects of virtual environments. Many applications which involve three or more abstract dimensions can be mapped to three spatial dimensions and then implemented in a virtual environment context.

The above criteria tell only if your application MIGHT benefit from virtual environments. They do not tell if the application would actually work as a virtual environment. Several kinds of things can make an implementation fail. Can the rendering task be done in less than a tenth of a second? Does the control need to be very precise? This is not possible with currently available trackers, so i.e. drawing programs would not work very well. How will the user control the environment? Manual tasks are fine with trackers and gloves, but how will you input text? You can't see the keyboard while in the environment. Does your application require very high quality rendering at high resolution? Current stereoscopic head-tracked displays are notorious for poor image quality. Is there some bottleneck such as large data access that can prevent your application from running at the speed required?

These problems are primarily technological in nature and will eventually go away as the technology improves. This will not be soon, however. For the foreseeable future (~ten years) virtual environments will be continually pushing at the boundaries of what the technology allows.

The kind of application that we know works well involves three-dimensional structures and graphics that do not have too much detail. Architectural walkthroughs and simple 'cartoon' environments are examples. Manual tasks such as the placing or orienting of objects also work well so long as the placement does not have to be too precise. Molecular modelling, 3D games, and simple sculpting work as interactive applications.

Very rich data applications benefit from virtual environments because the combination of three-dimensional display and control allow the user to intuitively explore the environment. An interesting class of applications which is currently finding some success is scientific visualization. These applications often involve the display of complex high dimensional data. Virtual environments create the capability of seeing the three dimensional structure of various graphical representations of data. Additionally, the ability to directly interact with data via, for example, hand tracking allows the user to cause data to be displayed selectively, thus allowing the exploration of complex data sets. This capability means that the user does not have to display all potentially interesting information at once, thus avoiding complex and difficult to understand displays. In addition to the obvious advantages of truly three dimensional display and interaction, the type of graphics that one draws in scientific visualization is well suited to virtual environments. Displays in scientific visualization are typically highly abstract and represent highly idealized representations of data. The graphic objects in a scientific visualization display are often fairly simple by design, and so can be drawn with the speed necessary for virtual environments. Simply put, most data involve abstract concepts whose graphic representation is entirely created by the user. This representation should contain only enough detail to present the desired data.

1.4 What you will get out of this course

In this course, you will be introduced to the basic hardware and software concepts behind virtual environment systems. The integration of these technologies into a working, high performance system will be discussed. Various interface ideas will be presented, along with several examples of scientific visualization applications.

The content of this course is existing technology and existing applications. It is primarily based on the experience of the instructors and people they know (see acknowledgments). There will be almost no discussion of research ideas or technologies under development.

This introductory survey will describe the basic technologies behind virtual environments. It will be followed by several applications described by those who designed them. It is hoped that you can learn from the successes and mistakes of others. Then you can go and have interestingly different successes and mistakes which will teach all of us new and interesting things about how to build virtual environments.

What will not be given is a formula for creating useful applications for given application problems. Virtual environments is still an infant field, and fundamental questions about the best way to build systems, to interface with the environment, and to design graphics for specific problems remain unanswered. Applying virtual environments in any field, not just scientific visualization, is a research effort. What you will have after this course is the background necessary to create your own solutions for designing your own approach in the application of virtual environments.

1.5 Acknowledgments

My knowledge of the construction of virtual environment systems is due to a career with many smart, distinguished researchers in the field. They span my years at VPL Research, the NASA Ames VIEW laboratory, and my current position at the Applied Research Branch of the Numerical Aerodynamic Simulation Systems Division at NASA Ames Research Center. These people are: at VPL, Jaron Lanier, Tom Zimmerman, Chuck Blanchard and Jean-Jaques Grimaud. At the NASA Ames VIEW Lab (now part of the Advanced Displays and Spatial Perception Lab), Scott Fisher, Warren Robinett, Doug Kerr, Rick Jacoby, Phil Stone, Mark Bolas, Ian MacDowall, Steve Ellis, and Beth Wentzel. At the Applied Research Branch, my current colleague and often coauthor Creon Levit. These people share credit for any correct insight presented in this survey. Any bad idea, of course, is entirely due to the author.