Using "CaveUT" to Build Immersive Displays With the Unreal Tournament Engine and a PC Cluster

Jeffrey Jacobson
University of Pittsburgh
jeff@planetjeff.net

Abstract

CaveUT is a set of software modifications to the graphically powerful game Unreal Tournament (UT). This demonstration uses CaveUT to drive a portable PC-based two-screen monoscopic immersive display. (fig 1) Each screen can rest in any orientation to the viewer and will appear to be a window (or window facet) looking into a single virtual world. Each screen is driven by a standard PC running a UT client, with an extra PC acting as both server and operator console. This design demonstrates methods which can be used to build larger displays with up to 32 screens. Hardware consists of standard PCs, low-cost digital projectors and common hardware, much of which could even be borrowed for a temporary installation. Furthermore, UT is an effective software platform for developing VR applications. Advanced authoring tools (3DS Maxx and Maya) built-in authoring support and a great deal of freeware content are available for UT. Researchers have used UT and its competitor "Quake" for scientific research. (Lewis & Jacobson, 2002) CaveUT is open-source freeware available at http://planetjeff.net, while UT itself is inexpensive and partially open source. The purpose of CaveUT is to provide an affordable way for researchers to build immersive displays.

CR Categories: I.3.7 Virtual Reality, D.2.6 Interactive Environments, C.2.3 Distributed Applications

Keywords: VR, CAVE, Game Engine, CaveUT, Unreal Tournament, Immersive Display, Virtual Environment.

1 Description of Demonstration

This demonstration will show how a multiscreen immersive display can be configured using common office equipment. Required components for each screen are: one small digital projector (DLP or LCD), a projection surface, a standard PC with a good video card (e.g. geForce4), Windows or Linux, a copy of the computer game, Unreal Tournament (UT) and the CaveUT freeware. The display also needs a simple network hub and an extra PC to act as the server and operator console. Up to 32 screens can be arranged in any orientation to the viewer, such as an enclosure or a video wall. (See http://planetjeff.net) The software cost is $50 for UT and nothing for CaveUT. The hardware can be purchased for approximately $5000 per screen, or simply borrowed for a temporary installation.

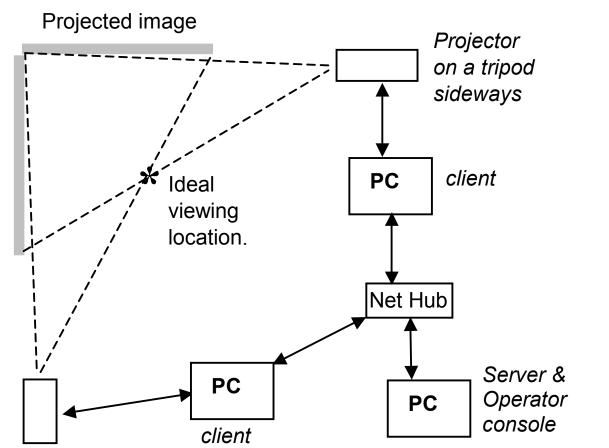

Figure One: Overhead Diagram of the two-screen UT-Cave.

This demonstration shows the simplest possible arrangement for an immersive multiscreen display. Two digital projectors are each mounted sideways on tripods and pointed into the 90-degree corner formed by the screens. (fig 1) Each projector displays on the screen opposite from it, so their beams cross, and a viewer is expected to stand in front of the display, looking directly into the seam between the two screens. On each screen, one edge of the image lies along the seam between the two screens.

CaveUT rotates and (perspective) corrects the images to the form a single contiguous view of the virtual world. (fig 2) The view in figure two looks best only for a user looking directly into the

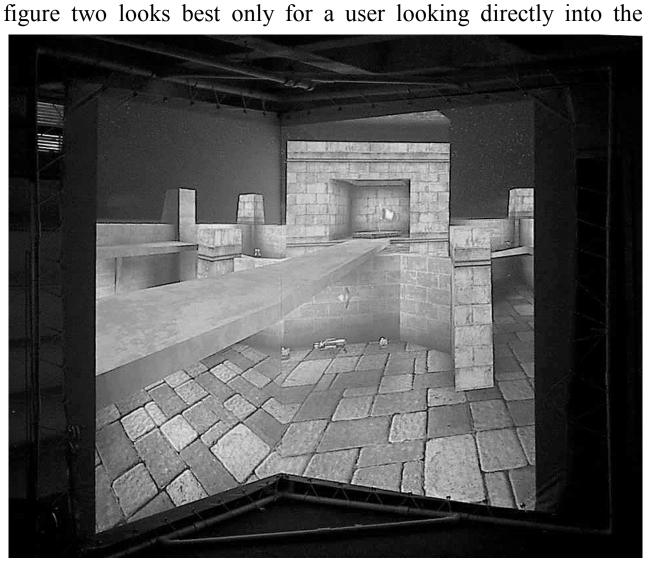

Figure Two: Overhead Diagram of the two-screen UT-Cave.

corner where the screens join, and standing as close as possible without getting in the way of the projection beams. The effect for him or her is a immersive display affording a roughly 110-degree field of view. Note that the lower part of the image in figure two looks bent, because the camera cannot be placed in the user's ideal viewing location and still photograph the whole display.

2 Motivation

This demonstration shows how CaveUT can provide an immersive display technology for researchers students and practicioners who could not otherwise afford it. The low cost could be especially important in academic departments such as psychology, film, education, theater, archaeology, and architecture. Their contributions are likely to be indispensable to our understanding of how VR can best be used.

At $50 per license, Unreal Tournament (and therefore CaveUT) supports rapid content creation, ready-made networking, and a lot of available freeware content. Furthermore, the physical design of the two-screen display is a good starter kit at the cost of $10,000 for components and approximately one person-month to assemble. Later, the researcher can add more screens, magnetic trackers, stereographic projectors, or other advanced equipment.

3 Game Engines vs. Traditional Approaches

For both technical and economic reasons, the interactive graphics performance of modern video games exceeds anything produced by traditional methods. [Lewis and Jacobson 2002] In a game engine, like the one in UT, objects are stored in a database where they have been reduced to a form which can be rendered very quickly. These objects can be shown at rest or in motion, and they can change shape within limits. This approach is much faster than constructing virtual objects on the fly from first principles. Game engine technology is highly developed and will advance more, because it is funded by a multibillion dollar game industry, now larger than the movie industry.

Most applications for immersive VR are programmed in general purpose languages like C/C++ and built upon a graphics library, like OpenGL or DirectX. Additional libraries (VR Juggler, World Toolkit, SVE Toolkit, OneSaf, Dive, Massive, and others.) provide VR-specific capabilities. While slow in both performance and development time, this approach has the advantage when the developer needs a great deal of control. For example, to visualize a cloud front from a stream of radar data the cloud-front "object" must be generated in real time, frame by frame, from a very large number of data points. The game engine architecture is not suited for this.

For VR applications in general, objects and animations can be generated with advanced authoring tools like 3d Studio Maxx or Maya and/or with authoring support built into the VR application itself. While game engines have no natural advantage in this area, fan-made-content-driven games such as UT have excellent built-in authoring support. Traditional VR software packages usually import static models, but rarely do much more.

The danger of too much authoring support is that it can restrict what the author wants to build. For example, UT has a lot of human animation software built into its engine supporting such things as inverse kinematics. This allows the virtual environment developer to create excellent human figures with a deformable mesh body, full texturing, and a ready-made library of figures, movements and even responses to external events. The downside designer's workload increases, the more the intended content varies from, or even contradicts, the intended theme of the game. For a general purpose engine, such as UT or Quake, there are still wide areas of application they are suited for, and the author can always use different game engines for different purposes.

UT has a robust networking protocol, which allows players connected via the internet to interact in a shared virtual world. CaveUT exploits this capability to drive a multiscreen display, while each client provides a different view of the virtual environment. Each client maintains a complete copy of the virtual world, so only update information needs to be. In a multiscreen display, the client views are synchronized to form a contiguous view. Some traditional VR packages (i.e. OneSAF) have a similar internet distribution model. Some also have a separate facility for VR applications such as multiscreen displays. While CaveUT currently has fewer capabilities in this area (e.g. stereo, head-tracking) it has no natural disadvantage and will improve.

4 Extensions

Another freeware modification to UT for research purposes is **GameBots** [Adobbati et. al., 2001]. See http://www-2.cs.cmu.edu/~galk/gamebots. GameBots allows a third party software to control agents over a TCP connection using an extensible API and receive sensory information from the agent along the same TCP link. The purpose and current use of the tool is to allow researchers to drive agent behavior using an arbitrarily complex outboard AI engine. Furthermore, CaveUT can be made to display a view from an agent controlled from GameBots, rather than a standard player. This also allows the researcher to use GameBots to control the CaveUT user's apparent location and movement in the virtual environment, which is useful for many kinds of experiments.

CaveUT uses highly localized code which exploits standard features in the game and the OpenGL libraries. It is compatible with most other modifications to UT or UT-based applications.

5 Acknowledgements

CaveUT was developed at the University of Pittsburgh's Medical Virtual Reality Center, and was jointly sponsored by the NIH P30 Grant DC05205 through the Department of Otolaryngology, and the AFOSR contract F49640-01-1-0542 through the Department of Information Science. Much thanks to Dr. Michael C. Lewis, (Information Science) Dr. Mark S. Redfern, and Dr. Joseph M. Furman (Otolaryngology) for their personal support for this effort.

6 References

Adobbati, R., Marshall, A.N., Scholer, A., Tejada, S., Kaminka, G.A., Schaffer, S., Sollitto, C. 2001. GameBots: A 3D virtual world test bed for multiagent research. In *Proceedings for the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS 2001*, (Montreal, Canada).

Lewis, C., Jacobson, J. 2002. Game Engines for Scientific Research, *Communications of the ACM*, 45, 1, 27-48

http://planetjeff.net

http://www-2.cs.cmu.edu/~galk/gamebots