

Configuring Multiscreen Displays With Existing Computer Equipment

Jeffrey Jacobson
www.planetjeff.net
Department of Information Sciences,
University of Pittsburgh

An immersive multiscreen display (a UT-Cave) may be assembled from common home/office equipment which can be borrowed in most research settings. The simplest design requires two LCD projectors, three personal computers, the corner of a room, a network hub and cables. The required software is an inexpensive but graphically powerful computer game, Unreal Tournament (UT), and a freeware patch called "CaveUT." Unreal Tournament is partially open source and supports rapid authoring of visually rich virtual worlds, complex animations, and software modifications such as alternative physics or artificial intelligence. PC-based game hardware and game engines such as the one for UT deliver superior real-time graphics at a tiny fraction of the cost for traditional immersive multiscreen displays, such as the CAVE™. While currently having fewer features, the UT-Cave, like the traditional CAVE-like displays, is useful for research in vehicle simulation, human balance, architectural simulation, novel human-computer interfaces and much more. Except for UT's game engine, which is very inexpensive, everything about the UT-Cave is free and open-source at "www.planetjeff.net/ut". UT-Caves with more screens have been assembled using the same basic design principles illustrated here. A sample experiment is presented which uses a four-screen UT-Cave controlled from the control program, "LabView."

INTRODUCTION

The "UT-Cave", described here is the simplest example of how a researcher may configure an immersive panoramic display from equipment s/he is likely to already have available. From this example, the reader can generalize to larger UT-Cave's with more screens.

The two screen design requires two standard LCD projectors, a PC for each projector and an extra for the operator's console, the game Unreal Tournament (UT) for each PC, a simple network hub, some hardware and the free software "CaveUT". (A "UT-Cave" is any PC-based immersive panoramic display which uses the CaveUT software.) The projection surface can be the clean white corner of a room, or some construction with adjoining screens.

CaveUT is a set of software modifications which allows UT to support a multi-view display with off-axis perspective correction, making an integrated multi-screen display possible. Unreal Tournament is partially open-source and programmable. It supports the authoring of complex virtual worlds, fundamental changes in game behavior (i.e. different physics), artificial intelligence and control of virtual actors from an external program.



Figure One: The display of a very simple V-Cave. The man in the foreground is in the ideal viewing location. The virtual environment he sees (VAE 2002) was authored by several researchers as an educational tool.

Immersive Panoramic displays have been around for some time, starting with the original CAVE™ (Cruz-Neira, et al. 1993) and now including many PC-based displays, such as the UT-Cave. Immersive panoramic displays are useful for (a) research in human balance effects resulting from visual input (b) cockpit and vehicle simulation for training and design evaluation (c) visualizing existing or proposed architectural spaces

for functional evaluation (d) development of novel human-computer interfaces (e) and other research less related to human factors.

Most existing installations are best for scientific visualization, and all have features, like stereographic imaging and head-tracking, not yet added to CaveUT. However, many researchers cannot afford the high installation cost and long application development times required for traditional displays. A UT Cave can deliver superior graphics capability at a tiny fraction of the cost, because it uses both the hardware and the software developed by the game industry. The CaveUT software and the sample UT-Cave design are open-source and free to non-commercial purposes at www.planetjeff.net/ut

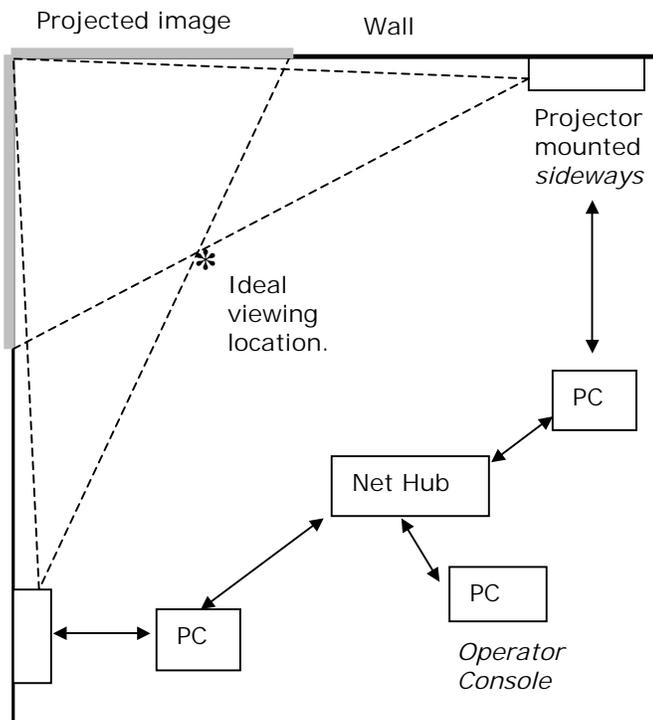


Figure Two: Diagram of a two-walled UT-Cave.

DESCRIPTION AND DESIGN

The two-screen design shown in figure two is the simplest complete example of a "UT-Cave", but it illustrates all of the important design issues and functional abilities. In this demonstration, two tripod-mounted digital projector project into the corner of a room, as shown in figure one. Each projector displays on the wall perpendicular to its own, and one edge of each projection lies along the corner itself, so the two images abut. Each projector is driven by a PC, running a copy

of Unreal Tournament (UT) with the CaveUT modifications. Both projector PCs and a third one, the operator's PC, are connected through an unmanaged network hub.

To start the UT-Cave, the operator begins a networked multiplayer UT game on the console computer, and enters the game as the only player. This begins a server process on the console machine, while the projector PCs each run UT as a client. The clients connect to the server using the game's built-in networking protocol and join the game. Each client enters the game in "spectator" mode, making it a bodiless camera in the virtual world. The spectator is set to be co-incident with the location of the operator's player, which ordinarily results in a duplicate of the display on the operator's console. However, each client's CaveUT modifications rotates the spectator's view, which causes the client to produce its designated view of the virtual world. CaveUT also modifies the perspective correction for each screen to complete the effect of a single contiguous display across all screens.

For example, the UT-Cave in figure one shows two client displays, one looking (in the virtual world) 45-degrees to the left, while the other looks 45-degrees to the right. The vanishing point in each display is moved so that the composite scene has correct perspective for the man standing in the foreground. This setup is different from the one in figure two in that the projectors were horizontal, not sideways. One projector rested upright on a bookshelf and the other hung upside-down from the ceiling.

UT-Caves with more screens work in much the same way, with view rotations and perspective corrections specified for each screen. The screens may be in any orientation with respect to the viewer and do not even have to be connected. Currently, the operator has to specify the optimal viewing location for the UT-Cave during installation and configure each projector machine's UT client accordingly. Real-time tracking has not yet been added.) Examples of other UT-Caves include the Earth Theater at the Carnegie Museum of Pittsburgh with five front-projected screens (Lewis and Jacobson, 2002) and the BNAVE at the University of Pittsburgh with three rear-projected screens and one front-projected floor screen. (Jacobson, et al. 2001)

USES IN HUMAN FACTORS RESEARCH

As with any immersive projective display, the researcher can use the UT-Cave to present the viewer a wide range

of interactive imagery to instrument many types of experiments. Generally, the UT in the UT-Cave can present virtual environments filled with static objects, avatars representing other "players" (usually a human figure) and artificially intelligent agents representing fictional people or other actors.

Architectural Visualization

Speculative and existing spaces could be displayed in a UT-Cave for functional evaluation. An office building and its occupants could be simulated to help researchers evaluate the visibility and accessibility of the emergency exits. In an emergency evacuation, traffic bottlenecks may appear in unexpected places, because the traffic patterns will be different from normal.

Alternative Human-Computer Interfaces

As with any immersive and/or 3D computer interface, the UT-Cave is a useful tool for innovation in controls, interaction design and visualization. An interesting subset of this is mixing physical props with projected imagery. For example, a firefighter could point a real fire hose at a virtual fire which would "shoot" virtual water into the virtual world.

Balance Research and Rehabilitation.

A central factor in human balance is the correspondence between visual input, vestibular cues, and Somatosensation (the position of body segments combined with sense of touch). A UT-Cave can be used to generate the visual scenes needed for experiments in this area. There is no particular advantage with a game engine like UT for visually simple scenes. However, UT should work best for complex scenes like a shopping mall, especially with crowds of people, situations difficult for people with certain balance disorders.

Cockpit and vehicle Simulation

A UT-Cave could be configured with physical props to simulate a vehicle cockpit. A driving simulator of this type could easily generate views for all of the windows as well as the rear and side-view mirrors. A UT-Cave could also be used to present a synchronized moving scene for someone on a stationary bicycle. This could be used to train bicycle riders to deal with potentially dangerous traffic situations.

Research in these areas have been going on for some time, but especially since the first computer-based

immersive panoramic display, the CAVE™ (Cruz-Neira, 1991) A lot of important research has been done, but the high cost of the CAVE and similar devices has excluded many researchers who cannot afford the hardware or the programming staff to write the VR applications.

ADVANTAGES OF THE UT-CAVE

Low Cost

The low cost of a UT-Cave allows investigators with a lot of skill, but little budget, to perform significant experiments in the uses of immersive interfaces and virtual reality. Good research could come from students, new faculty, and researchers in typically under-funded departments like psychology and film. By contrast, the typical cost of a CAVE™ or a similar device is in the hundreds of thousands of dollars.

High Capability

Because of recent advances in game-based technology, a PC with the latest game card delivers a great deal of rendering power for a very low price. (Lewis and Jacobson, 2002) As of June 2002, NVIDIA'S new Quadro4 900 XGL video card costs only a few hundred dollars and renders up to 60 million triangles per second. (NVIDIA, 2002) With a separate PC driving each display, the total rendering power of a multi-screen display can rival the largest computers specialized for computer graphics. (Humpherys, et al., 2001)

Because of this, many PC-based immersive panoramic displays are being built. However, application development for most such displays remains expensive and time-consuming, because most practitioners use custom software, usually something they write themselves. For example, in OpenGL the programmer specifies every polygon and every texture in the scene. This has a clear advantage over a game engine for producing scenes where a great deal of control is needed, such as with scientific visualization. But to produce complex virtual worlds, with human animations, detailed artwork, artificial physics, the direct programming approach requires huge effort. By contrast, content can be produced for game engines using advanced mass-produced authoring tools, like 3D Studio Maxx and Maya. Neither can any graphics-engine development effort by researchers hope to match the money and expertise that the mass-market game industry can bring to bear. (Lewis and Jacobson 2002)

Open Source

While the rendering engine for UT is proprietary, everything else in the game's software is open source and available for non-commercial purposes. This allows a large community of fans, and some scientists (Lewis and Jacobson, 2002) to make changes and distribute them freely. This provides researchers with an expanding base of software add-ons, like CaveUT, and a very large selection of ready-made virtual environments, human figures and objects. The physical design of the two-screen UT-Cave is also freely available, and described in great detail at www.planetjef.net/ut.

Speed of Authoring

With a UT-Cave, the researcher can quickly develop visually-rich complex interactive virtual environments (VEs). Any VE built for UT will always display in any UT-Cave, with NO modification. Advanced CAD and animation tools such as Maya and 3D Studio Maxx are best for producing advanced content. Simpler models and animations should be made with UnrealEd, a free CAD and animation tool which comes free with the game. The rendering engine for UT is proprietary, but it supports a Java-like scripting language which runs on a virtual machine which is in turn part of UT.

Unreal Script and the object oriented structure of UT specifically support large and small modifications to the interaction properties of the game--alternative physics, custom network, simple AI, and so on. Very large processing tasks such as advanced artificial intelligence, can be programmed into some external software, which communicates with UT's engine. (Adobbati, 2001) (GameBots, 2002) (Laird, 2002) While making changes to UT is usually easier than with older VR software packages, it still requires a high degree of skill. Similarly, anyone can make simple models for the VEs in UT, artistic skill and familiarity with the tools are needed to make something that looks really good.

Networking

UT supports multiplayer games over the internet. So a player in France could "join" the multiplayer game running on a UT-Cave in the United States. S/he would ordinarily appear to the American in the UT-Cave as human figure in the virtual environment. The American would appear on the French person's view of the same virtual environment as a human figure, also. Obviously, network delays could become a problem depending on the distances and networking equipment used.

The networking code in UT treats the UT-Cave's spectator-players no differently from regular players. So, it could support a 10-player game, where five of the "players" are used for a four-walled UT-Cave and other five are regular desktops. The player on the UT-Cave's operator console would appear to the humans using the five desktop displays as nothing more than a sixth player. Said another way, in a multiplayer game of UT, spectator-players can be associated (as a group) to any of the players in the game. This allows the UT-Cave to display a panoramic view from the viewpoint of any regular player in the game.

A SAMPLE EXPERIMENT

The following example shows how a UT-Cave was used to implement an experiment on human postural reactions to sudden visual changes in the viewer's virtual environment. The experiment used the BNAVE, (Jacobson, et al., 2001) a PC-based immersive panoramic display at the Medical Virtual Reality Center at the University of Pittsburgh. By adding GameBots (Adobbati, 2001) (GameBots, 2002), another freeware modification for UT, to the UT installation on the operator's console, the tester was able to control the BNAVE display using "Lab View".

Lab View is a software package used by many experimental scientists to perform data acquisition and instrument control for lab experiments. It can be used to control actuators, read sensors, and execute programs written in it's built-in scripting language. At the appropriate times during the experiment, the experimenter would run a Lab View script which would send commands to the operator PC's UT installation, which was modified by "GameBots."

GameBots (Adobbati, 2001) (GameBots, 2002) is also a freeware UT modification intended for scientific research. Once installed, it allows the researcher to open a standard TCP connection to UT server which causes a human figure, a "bot", to appear in the virtual world. This bot can be controlled through commands sent via a TCP connection according to a standardized API supported by the GameBots package. In this case, LabView opened the TCP connection and send movement commands to the bot.

Ordinarily, a client installation of UT is able to "spectate" a bot, just as it does a regular player, to display the view from the bot's location in the virtual world. In this experiment, LabView was able to create

and control a bot on the operator machine's UT server, which had both GameBots and CaveUT modifications active. The experimenter wrote programs in LabView's scripting language which effectively controlled the BNAVE's display, while the BNAVE was functioning as a UT-Cave. See figure three.

To instrument a repeatable experiment, the UT-Cave display must be controlled with software. The advantage of using LabView is that it can also be used to control actuators, read sensors and log data, all under program control. This allows fine control of the experimental conditions and data gathering with far less effort than manual protocols or writing custom software.

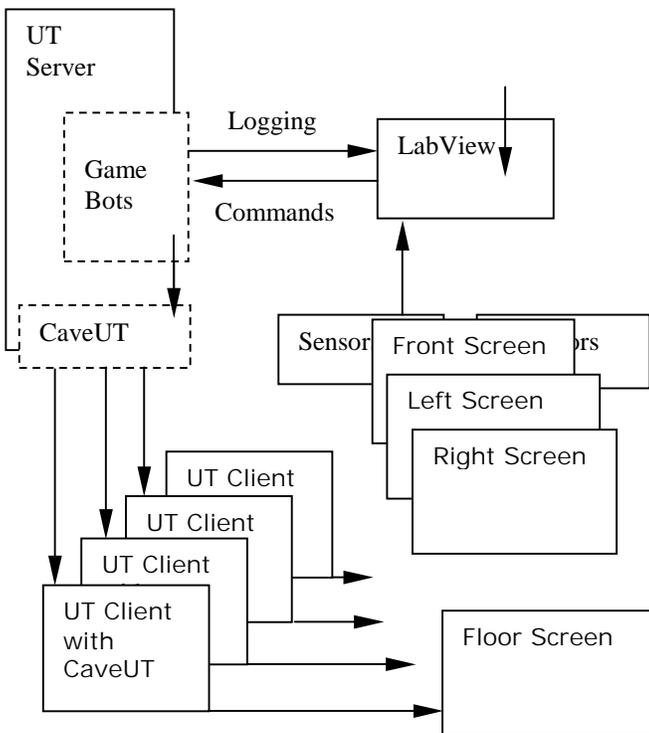


Figure Three: Software architecture of a UT-Cave display under LabView control.

GameBots also sends a great deal of logging and "sensory" information back to the external software on the other end of the TCP connection. In this experiment, the bot's moment-by-moment location within the virtual environment was the most interesting, because it reveals what the display was showing at each interval. LabView captured this logging data and wrote it to a file for later analysis along with data from the BNAVE sensors.

The data from this experiment are currently under analysis, and will be reported elsewhere in the literature.

ACKNOWLEDGEMENTS

CaveUT (CaveUT, 2002) 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. Thanks also to Dr. Patrick J. Sparto and Leigh Mahoney for their assistance with data acquisition and general assistance.

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, (Montreal, Canada, 2001).

Cruz-Neira, C., Sandlin, D.J., and DeFanti, T.A., (1993) Virtual reality: The design and implementation of the CAVE. In Proceedings of SIGGRAPH 1993 Computer Graphics Conference (Aug. 1993), 135-142.

GameBots (2002). www-2.cs.cmu.edu/~galk/gamebots

Humpherys, G., Edlridge, M., Buck, I., Stoll, G., Everett, M., and Hanrahan, P. (2001). WireGL: A scalable graphics system for clusters. In Proceedings of the 2001 Conference on Computer Graphics (Los Angeles, CA, 2001), 129-140

Jacobson, J. (2002). CaveUT 1.2, www.planetjeff.net/ut

Jacobson, J., Redfern, M. S., Furman, J. M., Whiney, L. W., Sparto, P. J., Wilson, J. B., Hodges, L. F. (2001). Balance NAVE: A Virtual Reality Facility for Research and Rehabilitation of Balance Disorders, Proceedings of Virtual Reality Software Technology, November 15-17, 2001

Lewis, C., Jacobson, J. (2002) Game Engines for Scientific Research, Communications of the ACM, January 2002

Laird, J.E. (2002). Using a computer game to develop advanced AI. Computer (July 2001), 70-75.

NVIDIA Inc. (2002) <http://www.nvidia.com>

Virtual Ancient Egypt (2002) <http://hemef.cfa.cmu.edu>