

Growing IVY: Building the Immersive Visual environment at York

M. Robinson, J. Laurence, J. Zacher, A. Hogue,
R. Allison, L. R. Harris, M. Jenkin, W. Stuerzlinger

Centre for Vision Research, York University
4700 Keele St., Toronto, Ontario, Canada, M3J 1P3

*matt@cs.yorku.ca, laurence@hpl.crestech.ca, zacher@hpl.crestech.ca, hogue@cs.yorku.ca
allison@cs.yorku.ca, harris@yorku.ca, jenkin@cs.yorku.ca, wolfgang@cs.yorku.ca*

Abstract

When we move about within our environment, we are presented with a range of cues to our motion. Virtual reality systems attempt to simulate these various sensory cues. IVY – the Immersive Visual environment at York – is a virtual environment being constructed at York University to investigate aspects of human perception and to examine the relative importance of various visual and non-visual cues to the generation of an effective virtual environment. This paper describes the motivation behind the design of IVY, and describes the design of the essential hardware and software components.

Key words: Immersive displays

1. Introduction and Motivation

When we move within an environment, we are presented with a range of cues to our motion within it. Our visual system provides optic flow, tracking, perspective, occlusion and other cues. Our vestibular system provides cues to the actual motion of our head, and the direction of gravity. Knowledge of our intent to move, feedback from our muscles, and many other direct and indirect cues are also present. Presenting self motion cues in isolation or in unusual combinations have helped us assess their contribution to our sensation of self motion [2, 3]. For example, it can be useful to pair different visual and physical motions. One approach to generating such cue conflicts is to use virtual reality.

Using virtual reality to study human perception introduces its own set of problems. The virtual environment must generate cues that agree with cues normally generated if the person was actually sensing in the real world. If this is not the case, then the unintentional conflicts introduced by the virtual reality will confound the intended conflicts introduced to investigate perceptual mechanisms. Unintentional conflicting cues include accommodation and cues from head tracker cables.



Figure 1: The Virtual Reality Tricycle. The VR Trike permits different combinations of visual and non-visual cues to be presented to an operator, but uses a HMD to generate the visual cues.

Over the past few years, we have developed a Virtual Reality Tricycle (see Figure 1) which permits different combinations of visual and non-visual cues to be presented to the user. The VR Trike is a self-sufficient display device, requiring only a tether for power. Subjects ‘ride’ the Trike which can either be mounted on rollers so that it does not physically move and hence does not generate correct vestibular cues to the rider, or it can be ridden around an open environment and hence can be used to present the rider with the appropriate physical cues to their own motion. The Trike is instrumented so that wheel motion and steering angle are known. The onboard computer (an SGI O2) combines this information with relative head motion sensed by the 6DOF head tracker to present simulated visual displays to the user via a V8 binocular head mounted display.



Figure 2. The Tumbling Room. The Tumbling room is an 8' cube mounted on a horizontal axis. The room can be rotated about this axis under computer control. The internal visual structure of the room is highly polarized in terms of its visual cues. Subjects are placed within the room and can either be rotated with the room – so that the visual environment remains visually upright even though the room has been physically rotated, or they can be kept physically upright while the room is rotated about them.

Although the Virtual Reality Trike has proven to be extremely effective for the examination of the perceptual cues that contribute to our sensation of self-motion, the device is limited by (i) the rather limited computational power of the onboard SGI O2 processor, (ii) the limited visual display offered by the HMD – including both limited screen resolution (VGA) and accommodation issues, and (iii) limitations of the head tracker – including both cabling issues and issues related to the mass of the visual display and head tracker assembly.

These limits are of particular concern for a set of experiments that we plan to conduct over the next few years to better understand how various sensory cues contribute to our perception of the direction of gravity. It has been known for a number of years that our perception of the direction of gravity can be manipulated in a number of ways. At York University, Ian Howard has developed the 'Tumbling Room'[1] which has proven extremely effective at generating a perceived gravity direction which is not aligned with the true gravity vector (see Figure 2).

Various experiments conducted within the room have demonstrated that subjects' perception of the direction of gravity can be manipulated through the presentation of an appropriately polarized visual display. Although the tumbling room is an effective tool for generating a perceived direction of gravity that does not agree with the normal gravity direction, it has a number of limitations. As the Tumbling Room is a real room, it takes large and powerful motors to rotate it. It is also very time consuming to change the internal visual texture of the room.

We have begun the construction of IVY – Immersive Visual environment at York. IVY will be used as a 'virtual tumbling room' to examine fundamental issues related to human perception, and to issues related to perception in virtual environments. When fully completed, IVY will consist of an enclosed cube 8' on a side. Each of the sides, including the ceiling and floor, will be rear-project

surfaces capable of presenting stereo visual displays to the observer inside of the cube at 60hz.

2. IVY Design

The design of any large-scale immersive visual display must deal with tasks related to video generation, and display, operator tracking, content generation, and input device management. Many of the solutions are application specific. For example, the solutions used by the Immersion Square at the Technical College Bonn-Rhein-Sieg and Fakespace's CAVE™ are very different but are directed at different applications. Given our desire to be able to generate a 'Virtual Tumbling Room', with a completely enclosed stereo visual world, many of these earlier approaches are not appropriate.

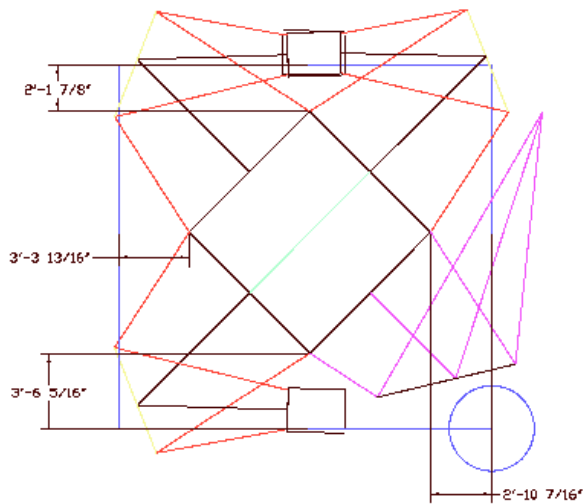
2.1 Video Generation

Video for IVY is generated by a 16 processor SGI Onyx2 "reality monster". Multiple video pipes are used to generate video for each of the 6 walls. Due to the load-bearing design of the floor, the floor is actually generated with two different projectors. The use of a single computer to generate the various video streams provides considerable simplification in terms of generating synchronized video to each of the walls of IVY.

Due to the high bandwidth of the video display (120hz video at 1280x1024) and the physical separation between the SGI Onyx2, it is necessary to convert the video signal to a digital signal that runs over on optical fibre cable and then reconstitute the video signal at the projection site. This physical separation has implications for input devices, and an input device server is being built to allow standard input devices to be used with IVY.

2.2 Video Projection and Display

Projecting video onto the walls of an immersive visual space is complicated by the relatively long throw distances required for a wall of size 8'x8'. For IVY we have chosen to use Featherlight mirror surfaces to reduce somewhat the



(a) Autocad drawing showing layout of the four vertical walls, the projectors (mounted in two stacks at the top and bottom of the image), and the four featherlight mirrors.



(b) View of the floor support taken from the 2nd story of the VGR lab. A catwalk around the space provides access to the projectors and mirrors. Note that IVY's floor is located 4' above the physical floor of the lab.

Figure 3. Physical layout of IVY.

physical footprint of the display. Figure 3(a) sketches the physical layout of the four vertical walls of IVY and the placement of the mirrors and projectors. The four vertical walls consist of aluminum frames covered with fabric projection surfaces. The corners of the display are trimmed with clear plexiglass in order to reduce the shadow lines associated with the seams between the walls. A similar approach is used on the floor and ceiling joints.

IVY is being constructed in the two-story Vision, Graphics and Robotics Laboratory at York University. IVY's floor is located four feet above the ground, which permits the visual display required for the floor surface to be projected, again via Featherlight mirrors.

Physically, the floor is a sandwich of a thick (4") glass panel which provides physical support so that users can walk within IVY, a fabric projection screen, and a thin glass layer on the top to protect the projection screen. The floor is held within a steel support cradle that also provides support for the walls. The ceiling is suspended above the rest of IVY from the physical ceiling of the lab. Again, in order to reduce the physical footprint, the video signal is bounced off of a Featherlight mirror.

The ceiling is slightly larger than the dimensions of the floor and is suspended slightly above the walls. This permits airflow within IVY, and it also provides a space through which various sensors – and especially video cameras – can view the user within IVY. One of the walls

is hinged so that it can be opened to permit a user to enter and exit the facility.

Once inside IVY, with the entry wall rotated back in place, the user is surrounded by displayable video surfaces.

2.3 Operator Tracking

In order to generate correct views and stereo displays, it is important to track accurately the true head position and orientation of the user. Given the desire to generate 6 complete video walls, most existing tracking technologies are inappropriate. For IVY we have chosen instead to rely on a hybrid tracking system comprised of an inertial tracker which is worn by the user (see Figure 4), combined with a slower video-based tracker mounted near the ceiling. The inertial tracker provides extremely fast updates, but drifts, requiring occasional correction by the video tracker system. Kalman filtering will be used to combine the two sets of tracker data.

The inertial tracker communicates with the external world via a wireless Ethernet connection. The user is not physically tethered to the outside world.

2.4 Software Environment

At York University we have developed VE, a Virtual Environment software library. VE supports the display of immersive environments for which more than one video display is required to present the environment to the user.



Figure 4. Components of the inertial tracker. The cross-shaped device in the upper left is the inertial tracker. The microprocessor (lower center) is worn by the user, and broadcasts tracking data offboard to the base station (lower left).

The VE library provides support and a framework for a number of key areas:

- screen geometry and positioning
- parallel rendering of screens on multiprocessor machines
- a general input device mechanism
- an event-based programming model
- a simple language for specifying the physical setup of the environment,
- including screens, input devices, and user-specific parameters

VE does not abstract the actual rendering of 3D geometry, nor is it tied to a specific implementation. The library is composed of three layers:

core - this layer provides basic support which is useful for most implementations and which does not constrain the implementation. The core provides the language parsing and geometry support.

core implementation - this layer is an interface which is common to every implementation. The core implementation includes general interfaces to the event model and input driver support.

implementation - this layer is specific to a particular platform and interface upon which VE is built. This is the layer that will be tied to a specific graphics language. For example, the "glxmt" implementation is a multithreaded implementation that uses the GLX interface (i.e. OpenGL on top of X). There is always a "null" implementation that does nothing except provide a placeholder for the minimal set of necessary core implementation calls. This implementation should be used when you only need the functionality of the core library and not of the implementation.

An important note is that the goal of VE is not to completely abstract the application from the underlying infrastructure. An application will need to be written in accordance with the VE library, a specific implementation of the VE library, and a particular rendering platform. The goal is to try to abstract out common components where the effort is worthwhile - providing an abstract 3D rendering library similar to OpenGL would not provide sufficient return on effort.

To use the VE library, a function must be defined that will generate the geometry for a scene. This function must be thread-safe in that multiple threads must be able to call the function simultaneously. This function should just generate the geometry, it should not load a view or projection matrix.

An almost complete example (minus the actual 3D drawing code) is shown below.

```
#include <ve.h>

void redisplay(VeWindow *w, long time,
              VeWallView *wv) {

    ...code for generating geometry goes here...
}

int main(int argc, char **argv) {
    veInit(&argc,argv); /* initializes library */

    /* define function to render geometry */
    veImplDisplayFunc(redisplay);

    /* set special options before windows are allocated */
    veImplSetOption("depth","1"); /* allocate z-buffer */

    /* create doublebuffered windows */
    veImplSetOption("doublebuffer","1");

    veInitEnv(); /* opens windows, starts renderer */

    veImplEventLoop(); /* handle events */
}
```

To walk through this example, the first step is to initialize the VE library with the following call:

```
veInit(&argc,argv);
```

This will parse any VE-specific command-line arguments and initialize the library. A number of initializations can be done between the `veInit()` and `veInitEnv()` calls, including:

- Setting program-wide implementation options.
- Explicitly loading input drivers.
- Defining rendering callback function

When `veInitEnv()` is called, all defined windows are opened and all event handlers are initialized, but they will not begin processing events and running handlers until `veImplEventLoop()` is called. The program will never return from `veImplEventLoop()` – it remains in this loop processing events until the program exits.

3. Current Status

IVY has just been moved to its new, permanent home in the Computer Science Building at York University. We expect to complete installation of IVY within this space in early 2002, and to be able to conduct experiments with IVY to investigate perceptual cues to the perceived direction of gravity by mid-2002.

4. Conclusions

IVY is an immersive visual environment currently under construction at York University, Canada. IVY is a 6-sided cube in which all of the walls of the cube are rear-projected video surfaces, including the ceiling and floor.

IVY is being constructed to be a virtual tumbling room. As such it will be used to enable researchers at York to investigate the factors that influence the perceived direction of gravity.

Completely surrounding the user with visual displays, requires a tracking technology which does not interfere with the user within the environment and which does not require a physical tether of any form between the user and the hardware. IVY utilizes a wireless inertial tracker coupled with a video-based tracker to accomplish this.

Finally, we have described the software environment that has been built to permit different virtual environments to be constructed. This environment permits the development of complex visual displays while not requiring the user to understand the details of generating the display or the various input and tracking devices that are available.

References

- [1] Allison, R. S., Howard, I. P., Zacher, J.E. (1999). Effect of field size, head motion and rotational velocity on roll vection and illusory self-tilt in a tumbling room. *Perception*, 28: 299-306.
- [2] Harris, L. R., Jenkin, M. and Zikovitz, D. C., Visual and non-visual cues in the perception of linear self motion, *Experimental Brain Research*, 135: 12-21.
- [3] Redlick, F. P., Jenkin, M., and Harris, L. R., Humans can use optic flow to estimate distance of travel, *Vis. Res.*, 41:213-219, 2001.