Steerable Interactive Television: Virtual Reality Technology Changes User Interfaces of Viewers and of Program Producers

Ronald Pose
School of Computer Science & Software Engineering,
Monash University,
Clayton, Victoria 3168, Australia.
Ronald .Pose@infotech.monash.edu.au

Abstract

Television has traditionally been a passive medium from the viewer's perspective. The viewer sits in front of the television receiver and passively absorbs what is presented. On the other hand immersive virtual reality systems engage the user and bring the user into the virtual world, often as a participant rather than just as an observer. This paper looks at applying virtual reality display technology, the Address Recalculation Pipeline, to the familiar technology of television. In so doing it transforms the relationship between the viewers and the television program producers. It can be made compatible with conventional television for those without the means or inclination to buy the new technology, and provides a base for future developments in virtual reality to be brought to the mass market. Linking virtual reality technology to mass entertainment has the potential to stimulate consumer interest and hence research and development funding in this demanding area, as well as providing an exciting, interactive system in its own right. The user interfaces of both the television production team and of the viewer are changed dramatically with this technology. This paper outlines how the user interfaces differ from conventional ones. An overview of the virtual reality display technology is given to set the scene.

1. Introduction

Researchers and practitioners have long been grappling with the serious problem of latency in responding to movement of a user in a virtual environment. This problem is most difficult to handle in the visual domain since the sheer quantity of data and processing required to update the view delivered to the user is beyond the capabilities of even high-end computer graphics systems. The problem, while simply described as the lag between a user's head position or orientation changing and the updating of the displayed virtual view to reflect that change, has rather severe consequences, and had eluded solution by many practitioners. While it is well known how to do the computation required to generate computer graphics images appropriate to the orientation of the observer in the virtual world, the problem is computationally extremely expensive. This has meant that even the fastest of conventional graphics computers have not been able to achieve the desired performance of over 60 correct frames per second.

It turns out that by solving this problem in the domain of virtual reality systems one has also provided a tool with much more general application. The problem of latency or lag in virtual reality systems has an analogue in systems involving telerobotics. In such systems a remote robotic device is controlled by a human guided by video input from the robot. As with virtual reality systems, any lag in response to changes in viewing orientation can lead to disorientation and a consequent loss of ability to achieve fine control of the robotic device.

Another potential application domain for this new architecture is in a form of interactive television system. A conventional television displays the viewpoint and orientation determined by the director and cameraman. Watching such a system is a purely passive affair. This new architecture offers the possibility of viewers being able to decide and change their viewing orientation in real-time, independently of viewers using other receiving apparatus. An interactive television system is possible using this technology and it can even be implemented in a way that is compatible with current television systems.

In this paper I will first give an outline of the new way of thinking about the original problem that led to the revolutionary new architectural approach. The key to the success of this project has been to consider the system as a whole. The technology was originally developed for virtual reality systems using head-mounted displays [1]. This paper concentrates on the user interface issues whereas another paper [2] elaborates on the application of this virtual reality technology to *Steerable Interactive Television*, which provides a form of telepresence.

Having set the scene by changing the way one thinks about the virtual reality problem, I will outline briefly how this new technology can be applied *to Steerable Interactive Television.*

It should be mentioned at the outset that the technology described herein has not only been shown through simulation and analysis to be of great benefit, but has also been developed to the stage of a working prototype Address Recalculation Pipeline (ARP). Thus the ideas have been demonstrated to be practical for virtual reality. I argue that the extension of this to a form of interactive television is not difficult, and could provide the mass market that will make the technology more affordable. The user interfaces though, especially for the television production team, are quite unconventional, and will require new working methods to be successful.

2. Conventional Approaches to the Problems

One approach to building virtual reality systems has been to slow the update rate of the display so as to enable the computer system to cope with the processing load. This leads to a rather jerky displayed world with latency being at least the time between updates of the display. Experience with motion pictures and with television has shown that the higher the frame rate the better.

An alternative approach is to reduce the processing load by reducing the quality and resolution of the images displayed. While the display update rate is increased and latency somewhat reduced with this approach, the realism of the virtual world is sacrificed leaving one with cartoon-like imagery rather than anything approaching photo realism.

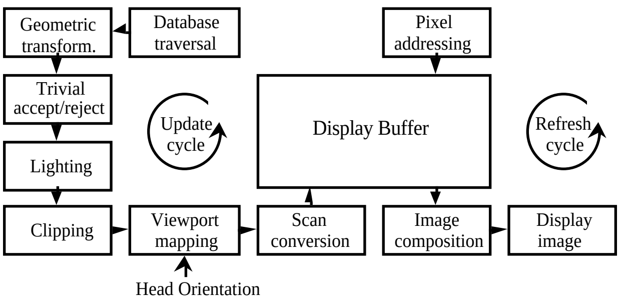

Fig. 1. Conventional Virtual Reality Display Architecture.

The state-of-the-art in conventional virtual reality systems relies on using the fastest available graphics processing and rendering equipment as depicted in Figure 1. This tends to be very expensive with the very best systems costing over a million dollars. In effect the problem is tackled indirectly through the use of massive amounts of processing power. Even so it can be observed that the trade-off between latency and image quality still has to be made, and that with reasonably realistic looking scenes the latency is still a significant problem. An extremely successful Virtual Reality computing system is the PixelPlanes system developed at the University of North Carolina [3]. This uses thousands of processors to give extremely impressive rendering performance and very good latency characteristics, however PixelPlanes is rather expensive technology that most developers could not afford.

3. Re-examining the Problem

In order to deal with this problem of latency it is helpful to forget for a moment the conventional ways of computer graphics and instead look at the overall environment in which we are working.

The problem as stated earlier is one of mismatch between what is displayed to the user and what should be displayed according to the virtual world model in which the user is situated. This mismatch usually is due to the latency involved in updating the display to reflect the virtual reality. The easy solution, involving reducing image quality so one can keep up with the virtual world, creates a mismatch of a different kind, but still spoils the illusion of virtual reality.

Let us examine what is happening in the virtual world itself. It could be that some part of the virtual environment is changing or moving in some way. While indeed this does occur, it is also true that in general only a small part of the virtual world is changing at any time, and typically the virtual world contains many static objects or scenes. Thus we can deal with such activity without recomputing everything, and since the objects involved are generally independent of the user, a slight delay in reflecting their changes does not cause the serious problems mentioned above.

The other event, which is much more significant, is the movement of the user within the virtual environment. Any slight movement of the user's head leads to a change in everything the user sees, hence it looks to the user as though the observed environment has moved completely. Thus one can see that it is the relative movement between the user and the objects in the virtual world which is observable to the user, and it is the user's own movement which has the most dramatic effect, since it affects the display of even static parts of the virtual world.

Now we should look more closely at the various types of user movement, to see what the consequences are. Consider the obvious movement, translation, for example walking forward. When translating, objects in the world appear to move relative to you. Close objects appear to move more than distant ones. Very distant objects may not appear to move much at all. Thus for translations we have to concentrate especially on close objects, and can largely ignore the distant background. In typical scenes the majority of the complexity of the scene is in the background, hence the problem appears tractable. We are also fortunate in that people do not tend to translate so very quickly so changes tend to happen at a manageable rate.

Another user movement is that of rotation of the head. Here we have a rather different effect. Even a small rotation immediately causes everything in the observed view to appear to move, even the background and static objects. There is also the added problem that rotations are much faster and more frequent than translations; one tends to look around and observe one's environment by rotating one's head or eyes. It looks as though everything has to be recomputed and re-rendered.

However, let us not forget the idea of stepping back and looking at the overall environment. It seems that the environment is staying relatively static but the user's view of the environment is changing quite rapidly and somewhat unpredictably, mainly due to rotation. The key point here is that the environment itself is not changing dramatically, so one needs to find a way in which the representation of the displayed environment also stays relatively static, and hence can be computed without the serious problems with which we are concerned.

4. The New Model

The ancient Greeks found a similar problem in trying to describe the motions of the planets and stars. They had a concept of planets moving relative to stars and to the Earth, and stars moving relative to the Earth. A model involving *crystal spheres* was developed. In this model the heavenly bodies were *painted* onto various layers of crystal spheres which were centred on the Earth and could move relative to one another. Objects on close spheres move more compared with objects on distant spheres. Various complex models of the movements of heavenly bodies were formulated in terms of these spherical shells surrounding the Earth. We now know that much of the observed motion of the stars is due to the rotation of the Earth, but since the observed motion is relative, a successful model centred on the Earth is possible.

We can use a similar model for a virtual world. Concentric spheres centred on the user would have the objects of the virtual world painted on them. Close objects on inner spheres and distant objects on outer spheres. In essence all possible views from the user's position are already rendered onto the spheres, and it only remains to display appropriate views as seen by the user. A rotation merely involves displaying a different portion of the sphere. A translation will require some updating of the spheres, however outer spheres will change much less than inner spheres, and hence may require little or no updating. Of course changes within the virtual environment will have to be reflected in the spheres, but typically only a subset of the spheres will be involved.

5. Implementation of Viewport Independent Display Memory

We have devised a model in which we render the images of the virtual world onto a surface or surfaces surrounding the user. However it may appear that in so doing we have actually created a larger processing bottleneck, both in generating the images and rendering them, and in selecting the appropriate view to display. On first examination it may appear we have significantly greater rendering overheads such as scan conversion, clipping etc. than a conventional system, however this is rarely the case, and is only found if the scene has polygons evenly spread out in all three dimensional directions. With a viewport independent display memory one must scan convert all polygons received. This is the worst case scenario for a conventional system, but for guaranteed interactive response one must allow for the worst case. Many conventional rendering systems are designed to cope with situations approaching the worst case scenario [4]. The rendering overheads for a conventional system may be reduced if the user is not looking at a complex part of the scene, however as the system has no control over the user's choice of direction for viewing, it is fair to assume the user is looking at the most polygonally dense section of the world. The viewport mapping is indeed computationally expensive however this task has been offloaded into relatively simple and cheap dedicated hardware, the *Address Recalculation Pipeline* [1].

The latency is determined by how long it takes to select the required portion of the encapsulating surface and write the image to the display device. This is handled by the *Address Recalculation Pipeline* which runs at the speed of the display device and introduces negligible latency. The new improved approach is depicted in Figure 2 and a comparison of the latency components with those of conventional systems is shown in Figure 3.

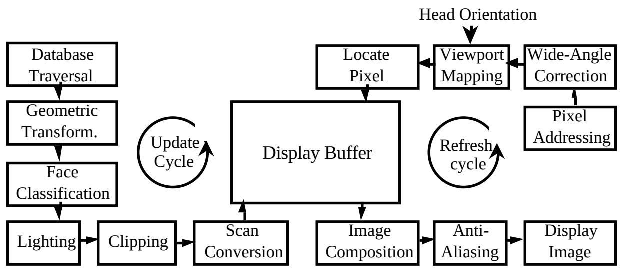

Fig. 2. Delayed Viewport Mapping.

The selection of the appropriate region of the sphere to display can be modelled as projecting rays from the eye onto the surface of the sphere, so defining the area to be seen. Alternatively one can view the process as one of changing coordinate systems from real-world to spherical world. This indeed requires a great deal of computation for each pixel displayed and would be infeasible in software. However the approach is ideal for a pipelined hardware implementation, which while using fairly fast hardware has a fairly simple structure and can be built economically [1].

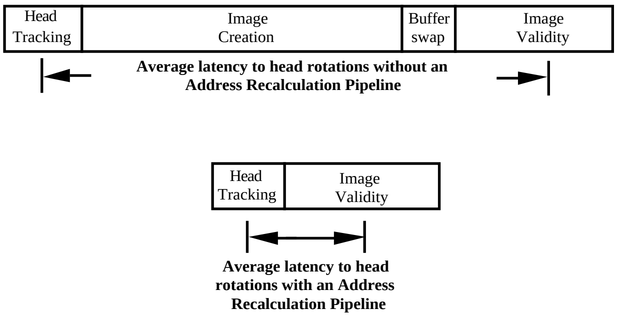

Fig. 3. Comparison of rotational latency components.

We did investigate a hardware design based on a spherical system but found some annoying characteristics. First, in mapping a sphere onto a memory array one tends to waste a lot of memory, or else have a very complex mapping scheme. Second, the apparent sizes of pixels in a sphere vary greatly as one moves from equator to the poles. Third, the coordinate transformations involve trigonometric functions which are moderately expensive to implement. By far the dominant issue is the first, in that we are using much more display memory than a conventional graphics system, and since we are not accessing it in a nice regular pattern, we are forced to use quite fast memory, hence we do not want to waste any of it.

Essentially the basic model will work for any surface surrounding the user. A sphere is intuitively obvious in that everything is equidistant, however one can go to the other extreme and look at a tetrahedron. What we eventually chose was a cube. It has very nice properties in that no memory is wasted since its surfaces are the same shape as memory chips. The coordinate transformations are also much simpler in that they are linear and can be implemented essentially with a matrix multiplication. Pixel sizes do not vary as much as for a sphere. While there may be some concern about strange effects occurring in corners and edges these have been shown not to be significant.

A prototype display memory system based on the cube has been implemented and functions well. We call this method using viewport independent rendering, *Delayed Viewport Mapping*, and the hardware realization, an *Address Recalculation Pipeline* [1]. A technique called *Priority Rendering* has been developed to handle display updates efficiently [5].

6. Priority Rendering

Objects in the virtual world will appear to change at different rates. The apparent movement of the object due to a rotation of the user is handled automatically by the *Address Recalculation Pipeline* which implements *Delayed Viewport Mapping*. Other changes in the way the object is perceived are due to the object's own animation and due to translational movement of the user in the virtual world. It is possible to classify further the kinds of changes in the appearance of the object into those involving its displayed size and those involving its displayed position. Objects will appear larger when one is closer to them and smaller when one is farther away. Relative side-to-side movement will cause an apparent change in the position in which the object is displayed. Of course any animation of the object itself can lead to both these effects. The effects of user translation can be quantified using simple geometric analysis [5]. Using this approach one can calculate the amount of time for which the current displayed image of the object will remain valid, and hence determine the required rate of update of the object's displayed image.

Priority Rendering takes advantage of the fact that not all objects need to be updated at the same rate. By sorting the objects into a priority order based on their required update rates, one can render the most important changes in object appearance first, thus guaranteeing that if perchance there is insufficient rendering capacity, one has achieved the best possible result. Even in such an unfortunate case the latency of response to head orientation changes is not affected, since that is handled by the *Address Recalculation Pipeline*.

This however is not the main advantage of *Priority Rendering*. Its most dramatic property is that the overall rendering required has been reduced significantly. We have shown that an order of magnitude reduction in required rendering is not unreasonable due to this approach [5]. This occurs because the average required update rate for all objects in the virtual world is much less than the display update rate. By balancing the rendering load among multiple renderers updating multiple display memories at various update rates and composing the final image for display, the combination of *Priority Rendering*, image composition, and the *Delayed Viewport Mapping* provided by the *Address Recalculation Pipeline* can outperform conventional Virtual Reality implementations.

7. Applying the *Address Recalculation Pipeline* to *Steerable Interactive Television*

We have seen briefly how one can build technology to support immersive virtual reality. We are also familiar with conventional television and film production and viewing techniques. Now we will investigate how to meld these technologies to provide a much more interactive and exciting experience for the viewer, while avoiding the expense and difficulties associated with full-blown virtual reality systems.

In virtual reality systems one displays views of a virtual world to the user. Instead of having computer generated images one could use a video source such as a digital television camera. One could also have a hybrid, augmented reality approach.

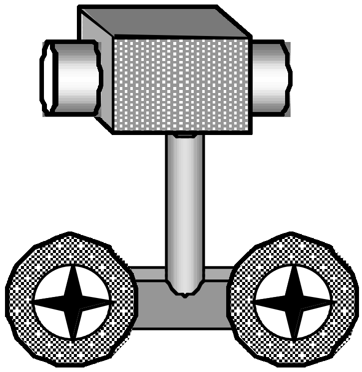

Fig. 4. Double-ended fish-eye lens camera on dolly.

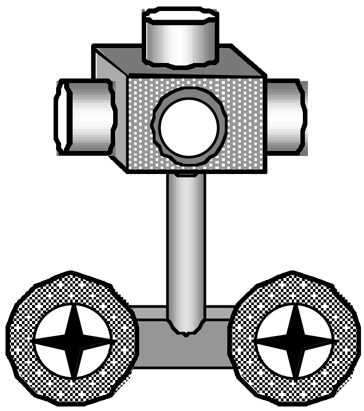

Fig. 5. Five-sided 90˚ lens camera on dolly.

Let us consider how one could replace the computer graphics renderers in the system described above, with video input. We still want to use a viewport independent display memory embodied in an *Address Recalculation Pipeline* system. Thus we want to generate images on the encapsulating display memory. A conventional television camera has a limited field of view. Equipped with a fish-eye lens this can approach 180˚. So it seems feasible to use two cameras so equipped to cover the encapsulating surface. Luckily the *Address Recalculation Pipeline* can handle the distortions of such lenses without penalty. Figure 4 depicts such an arrangement. Note that with an effective 360˚ view it becomes difficult to mount the camera without obstructing the view. In fact one has to sacrifice some of the view to allow the camera to be supported, but the traditional human operator standing behind the camera is impossible. Instead one would use a slim post mounted on a remotely-controlled wheeled platform, and try not to look directly down.

There is a similar problem in getting the video signal from the camera. Any cables would potentially be visible, so a wireless transmission system must be employed.

There is however another disadvantage to such an arrangement. Even high resolution video cameras produce rather pixelated images when you try to get all the detail of a 180˚ view field into their limited number of pixels.

A better arrangement from that point of view is to use an array of six cameras, each with a 90˚ field of view. In effect using a camera for each of the faces of the cubic display memory. Because of the need to mount the camera, one would probably omit the downward facing camera and use that face for mounting the whole assembly. Such an arrangement is depicted in figure 5.

In all such arrangements one cannot escape losing part of your viewing field since it is difficult to have the camera assembly suspended in mid air. In practice this will not be a problem since only a small area around one's feet will be blacked out.

While the viewers will not be upset about the slight loss of viewing freedom, the director and cameraman have lost their traditional places behind the camera and must treat the production process as a telerobotic operation [6].

8. Technical Difficulties

The mechanical difficulties in mounting the cameras have been mentioned above. It is quite feasible to use a remotely-controlled motorized platform to move the camera assembly around. The camera operator would use a joystick or other device to navigate around the set. This is an approach analogous to telerobotics [6].

Another aspect that affects this is that the traditional zoom lens is not really suitable for this application. Consider that the effect of zooming in is in some ways like moving towards the subject, but with a 360˚ view you would want corresponding zooming out in the opposite direction and appropriate effects in the side views and top view. This makes zooming impractical. Instead one must physically move the camera. Thus the motorized wheeled mount must be capable of very smooth movement.

A form of viewer initiated zooming is possible using the intrinsic viewing lens compensation mechanism of the *Address Recalculation Pipeline*. It is conceivable that such a mechanism could be invoked automatically by the programme producer if the viewer chose to allow it.

The other logistical difficulty is that of getting the video signal from the camera assembly. Trailing cables are not practical since they would be visible to the viewer. Similarly power cables would intrude on the scene. Battery operation can solve the power problem and wireless transmission of the video signal deals with the other problem. It should be noted however that there is quite a deal of information to be transmitted from the camera assembly. This is equivalent to five normal video channels. There must also be provision for control signals to be transmitted from the camera operator or director and received by the motorized camera assembly.

As well as transmitting the video information, it is also desirable to transmit and monitor the state of the camera assembly. Information about the current location and orientation of the cameras is of use later on, especially to ensure that a transmission compatible with conventional television receivers is possible.

To complement the video information it is desirable also to transmit stereo or surround sound to give the correct audio perspective aligned to the viewing orientation. Thus, as the camera assembly is rotated, so the soundstage should also be rotated to match it. Moreover, since we do not know the viewing orientation of the user at this stage, we have to transmit audio from an array of microphones, in much the same way as we transmit the video from an array of cameras. The reconstruction of an appropriate stereo or surround sound image thus occurs at the consumer's receiver, rather than at the production end of the system. Of course, a predefined 'front' view can be transmitted to allow for those with conventional television receivers to get a fairly conventional viewing and audio experience.

Solving these problems involves careful design, but fairly straightforward engineering practices. No technological breakthroughs are necessary. It is thus an eminently practical project to construct such a device.

9. The Viewer's Perspective

At the consumer end of the system, ideally one has a television set equipped with an *Address Recalculation Pipeline*. The signal is received from the broadcast medium, either cable or wireless, and is fed directly into the display memory, replacing the graphics rendering engines which are normally used for virtual reality.

What the viewer then has available is an almost complete surround image available, around which he can navigate using the *Address Recalculation Pipeline* equipped with a suitable control device such as a games joystick.

In order to get the spatial nature of the sound correct, the receiver has to be equipped with some audio processing hardware or software which takes in the orientation information of the camera array and the current viewing orientation and, in effect selects the appropriate microphone sound signals as the 'front' left and right speaker inputs. More sophisticated surround sound processing and even Convolvatron style 3D audio processing is possible. It is necessary to do some processing of the audio signal to maintain a good degree of realism and immersion in the programme, but a simple rotation of the sound stage is probably adequate for an economical yet effective system.

Of course there also has to be the means to receive the transmitted audio video and control information. This could, in its simplest form, just take the form of a number of conventional television tuners, each say receiving the signal for a cube face. More efficient use of broadcast spectrum is of course possible with some form of encoding of the signal. The exact form of the broadcast transmission is an engineering and spectrum management issue, but has no impact on the technical feasibility of the system.

A typical consumer *Steerable Interactive Television* receiver along the above lines is depicted in figure 6.

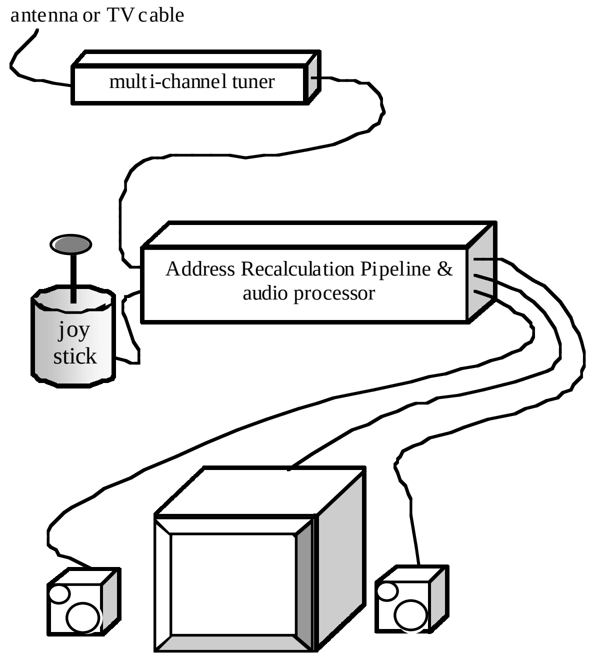

Fig. 6. Typical Steerable Interactive Television Set.

10. The Conventional Television Viewer

The system as described above is not hugely expensive. There is no computer graphics required. The *Address Recalculation Pipeline* is easily mass produced as a single chip doing the rotation and distortion correcting calculations, coupled with an appropriate amount of memory for the viewport independent display. However there will always be those who only have a conventional television receiver. For them a conventional transmission corresponding to the 'front' view and sound from the camera assembly is provided. This might even take the form of a face of the 'cube' which is set up to be the preferred or director's viewing suggestion.

The approach of having a separate transmission for conventional television viewers is the simplest system. For cable TV broadcast it is probably appropriate since bandwidth availability is not a problem. However for wireless transmission, available broadcast spectrum has to be used economically. In such a case it would be possible to use the conventional television view as one of the cube face images, thus eliminating some of the redundancy in transmission.

An alternative is to have a fairly simple set-top box which feeds a conventional television receiver, but doesn't provide all the facilities of an *Address Recalculation Pipeline*.

11. More Sophisticated Viewing Systems

We saw earlier that the *Address Recalculation Pipeline* was developed with virtual reality systems in mind, in particular those with head-mounted display systems. For virtual reality systems we also devised a *Priority Rendering* scheme which also takes advantage of image composition to reduce significantly the computer graphics processing required for virtual reality.

While these are not required *for Steerable Interactive Television*, it is possible to have a hybrid, augmented reality style system. In such a system, the live video input is combined, using image compositional techniques, with a virtual reality system.

This could take two forms. One possibility is simply to merge a virtual reality system with the *Steerable Interactive Television* system, thus allowing the virtual world and characters to interact with the broadcast interactive television world. This of course is beyond the control or knowledge of the television producer, so any integration concerns are the problem of the viewer.

Another approach would be to broadcast details of a virtual world along with the live video and audio information. Thus the virtual world is also the creation in part of the television producer, and it should be easier to integrate it cleanly. Of course the user is free to modify the virtual world as he interacts with his copy of it.

The integration of real actors and computer generated characters has been tried in cinema productions. What I am proposing is the next step in which the user can personally interact with the virtual world, or even become part of it.

From the hardware viewpoint this kind of system is straight forward and really just involves the coupling of an *Address Recalculation Pipeline* based virtual reality system with the *Steerable Interactive Television* system. In effect the *Steerable InteractiveTtelevision* system appears as a stage in the pipeline. A possible arrangement is depicted in figure 7.

As well as letting the viewer change his viewing orientation, in effect steering the camera assembly in a virtual way, it is also possible to allow the user to zoom in or out. As mentioned earlier, it is impractical to use actual zoom lenses on the cameras, but it is possible to do the kind of electronic zooming done in many consumer video camcorders. An economical way to achieve this effect is to make use of the viewing lens distortion correction built into the *Address Recalculation Pipeline*. Instead of a fixed distortion correction corresponding to the head-mounted display or the camera array, it is possible to change the effective field of view dynamically.

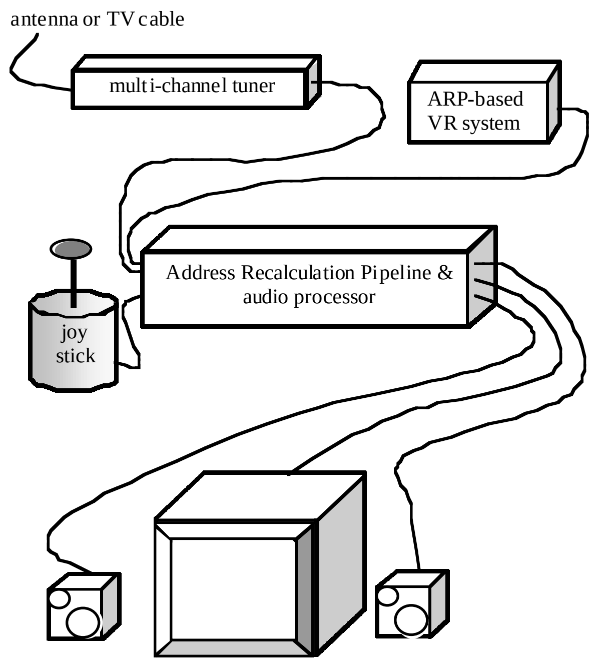

Fig. 7. Steerable Interactive Television augmented with VR.

Another way to use the system is to use a head-mounted display and tracking system. That is to treat the system much more like a conventional virtual reality system. The display independence of the *Address Recalculation Pipeline* enables a variety of display technologies to be employed at the consumer end of the system.

It is not however a simple merger of a *Priority Rendering, Address Recalculation Pipeline* system with that of *Steerable Interactive Television*. The video input is essentially flat and would have to be assigned a Z value and the virtual reality components would need to be positioned carefully in front or behind the video. This would be an interesting research topic in its own right.

12. User Interface for the Production Team

We have already seen that the traditional scene at a television or movie studio where the director, camera operator and others stand behind the camera and supervise what goes on, will not work using this technology. Essentially there is no concept of 'behind' the camera. We are dealing with a 360° view, and ideally with a corresponding surround sound system. Thus the production team has to be out of eye and earshot.

This is a completely new and radical idea for a film director and the rest of the team. Everything has to be done and controlled remotely and discreetly. It will no doubt be upsetting for some traditional film makers.

The conceptual change is much more serious than the physical production setting changes. Traditionally the film director decides what the audience will see and hear, from which camera angle etc. He has these tools available to try to conjure up his desired impressions in the viewers. With this technology much of that power is taken out of his hands. Certainly he presents a preferred view, but the users are free to look around elsewhere at what interests them. This makes life much more difficult for the director since nothing can be hidden off-camera. The means of getting across the desired story is also quite foreign. One has to encourage and entice the viewer into observing what is important rather than being explicit in showing him.

It remains to be seen whether current film and television producers can adapt, or whether it will take a new generation to explore and master this technology.

13. The Viewer's User Interface

The Steerable Interactive Television viewer does not have to face the radical changes that the production team faces. he can, if he desires, take a traditional passive television style approach. However the exciting part is that he now has the power to explore the television or film environment for himself, independent of other viewers on other receivers, and independent of the television producer.

It remains to be seen whether this technology is attractive for users, and how long it takes for them to become comfortable with it. I suspect that the consumer end of the system will present few difficulties compared with the production end.

14. Conclusion

We have seen how technology originally developed for virtual reality systems can be applied to a form of interactive television. In this way we can introduce virtual reality style immersive systems to the mass consumer market, hopefully getting the benefits and economies of mass production. We also saw how we can integrate our *Steerable Interactive Television* with virtual reality, taking advantage of our using the hardware already designed for virtual reality.

Not only have we described an exciting system which has the potential to transform our traditionally passive view of television into something much more engaging, but we have also opened up a whole new area of hybrid systems whereby users could interact virtually with the characters being broadcast, in a similar way to the ways demonstrated in some recent cinema releases where real actors interact with computer generated characters.

The system has not yet been built, although there are no technical developments necessary before such a system could be assembled. What will be necessary is lots of experience in how the various components should be integrated to give a cost-effective system with desired performance characteristics.

Much experience needs to be gained in using such a system and in designing suitable user interfaces for both the television production team and for the viewers at the consumer end.

We have great potential here to integrate television and computer technology in new ways, much more exciting and creative than traditional computer games. The user interfaces present a new challenge, particularly for the production team.

Acknowledgements

The prototype *Address Recalculation Pipeline* hardware was funded by an Australian Research Council small research grant. Patenting and commercialization of the *Priority Rendering* technology was supported by a Monash Research Fund grant. Matthew Regan was responsible for much of the original virtual reality based research.

References

[1] Regan, Matthew and Pose, Ronald (1993) An Interactive Graphics Display Architecture. Proceedings of IEEE Virtual Reality Annual International Symposium. (18-22 September 1993, Seattle USA), pp. 293-299.

[2] Pose, Ronald (2001) Virtual Reality Technology for Mass Entertainment: Steerable Interactive Television submitted to IEEE Virtual Reality 2001 (VR-2001).

[3] Fuchs, Henry et al. (1989) Pixel-Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor Enhanced Memories. Proceedings of SIGGRAPH 89. In *Computer Graphics*, Annual Conference Series 1989. pp. 79-88.

[4] Molnar, Steven and Fuchs, Henry. (1990) Advanced Raster Graphics Architectures. Chapter 18, Computer Graphics, Foley and van Dam, pp. 872-873.

[5] Regan, Matthew and Pose, Ronald (1994) Priority Rendering with a Virtual Reality Address Recalculation Pipeline. Proceedings of SIGGRAPH 94. In Computer Graphics, Annual Conference Series 1994. pp. 155-162.

[6] Regan, Matthew and Pose, Ronald (1994a) Virtual Reality and Telerobotics Applications of an Address Recalculation Pipeline. ISMCR'94: Topical Workshop on Virtual Reality, Proceedings of the Fourth International Symposium on Measurement and Control in Robotics, Houston.