The Watching Window: A Reactive Real-time Model-Viewer

Alister West, Keekim Heng
Computer Graphics and Vision Research Group
Department of Computer Science
University of Otago

Abstract

We are presenting an unencumbered virtual reality system called the "Watching Window".

This is robust and easy for a subject to use and cost effective compared to other Virtual Reality model viewing systems. The interactivity of the system is demonstrated though the implementation of a virtual pottery simulation. The system involves a subject who can move about within a large restricted area and can interact with a virtual environment by the movements of their head or hand. The hand and head are found from captured video streams from two video cameras. The system is designed to work in real-time, so speed and reactivity are of great importance, as is the ability to remove as much noise (inaccuracies) from the system as possible.

1. Introduction

Since the beginning of computer systems we have moved towards a 'desktop' environment. This is now considered standard, but with the advances in technology, more sophisticated methods of display, providing more interactivity, are becoming cheaper and more available to the average end user. We have implemented a system which makes it easy to provide parallax while viewing 3D models on a flat (2D) screen. The result is the ability to interactively view 3D models created by a reversible deformation technique.

Virtual Reality (VR) systems attempt to create a virtual environment which looks as convincing as possible to the user. We have grown up in a 3D world where we rely on depth cues to help us place objects around us, so a 2D VR system needs to provide enough artificial depth cues for us to overcome our instinctive rejection of the scene as 2D.

Different VR systems provide different levels of immersion into this imaginary environment. Such systems can be classified according to their level of 3D immersion (Mazuryk & Gervautz 1996).

2. Existing Systems

Desktop VR is like a window into a 3D world. This is what most computer modeling environments (such as Maya) and most modern 3D first person games (such as Quake) are based on. To interact with the virtual world you would usually have a 'learned' interface (one that is specific to that computer environment). A mouse and keyboard combination are most common for this.

FishTank VR (FTVR) systems use head positional information to provide a level of response to the user's movement. These systems are becoming quite cheap and are easy and fun to implement. A broad range of research has emerged in this area, such as face tracking and recognition (Agre 2001), feature point matching (Bobick 1999), multiple screens in wedge formation (Treadgold 2001) and many more. However, one disadvantage is that these systems tend to require a lot of room for back projection screens as a person in front of the screen would interfere with the projected images. Although these types of systems tend to be a lot cheaper than other methods large scale FTVR systems like The Cave (Leigh 2000) require a lot of room and resources and can still be very expensive.

Immersive VR requires a computer generated visual environment which is often displayed in a Head Mounted Display (HMD) unit. HMD units track head position, rotation and any interaction to display a full immersive environment.

There are many other VR systems but they can usually be classified into one or more of the above categories. FTVR has a strong following of modern research, an example of which is Illusion Hole.

2.1. IllusionHole

IllusionHole (Kitamura, Konishi, Yamamoto & Kishino 2001) is a FTVR system that provides a way for multiple users to view a projected 3D object on a table top like display unit. Each user sees an image which changes depending on their position around the object. A hole-like mask stops the users from seeing the other users' images unless

the users get too close together.

The paper suggests that three people is the best ratio of people to display-space. The more people there are, the smaller the image becomes. This is so that the images do not overlap.

Each user is equipped with a head tracking device and a pair of shutter stereo glasses. These glasses flicker on and off in syncrony with the display unit to provide a different image for each eye. This creates a stereo image that adjusts according to the position of the user. The IllusionHole system could find application in the fields of medicine and industrial design, where 2D images can be unclear.

Although this sort of system is ideal for small objects and a few users, widespread application is unlikely because of the time required to set up the environment. Having specialised small equipment like shutter glasses and head-mounted tracking units can also become cumbersome for many users. The amount of space able to be used by the users is limited by both the space available around the table and the range of movement within the screen before image overlap occurs.

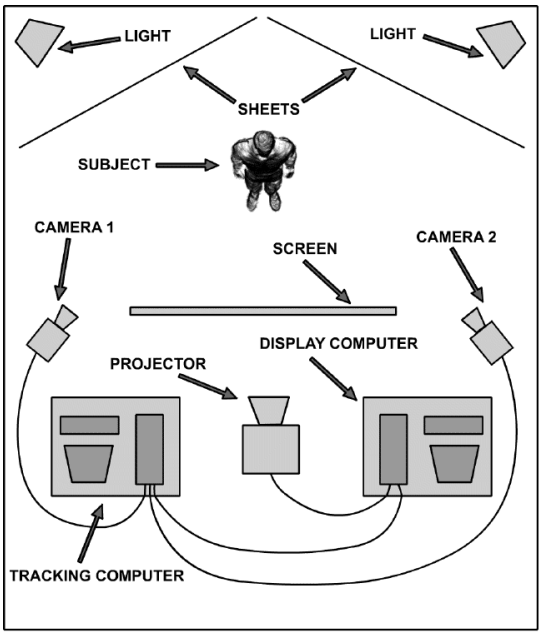

Figure 1. Virtual modeling system interface

3. The Watching Window

The Watching Window holds many advantages over previous methods. It is cheap to build, has a lot of usable space in front of the screen, it is easy and intuitive to use and does not require the user to change or wear any special equipment.

We present the idea of an unencumbered FTVR system where the head and hand positions are calculated through vision tracking techniques. We call this system the Watching Window as it recreates a 'window' onto virtual scenes.

It is primarily used for viewing and altering free-form deformation models created interactively using a blendeforming system. We also present a comparison between images displayed for 3D goggles (red/blue) and without. The system uses two computers and two video cameras as shown in Figure 1.

One computer is the tracking computer which receives all picture data from the cameras and calculates significant point positions in space. The other computer uses the positional information to create a virtual environment that reacts to the user's body movements.

3.1. Setting up the Watching Window

As a practical vision application several steps had to be taken to obtain accurate information from the camera to the computer as there is an inherent amount of noise when dealing with real direct input.

Background subtraction was initially unreliable because of the wide range of clothing colours people would wear on any given day. A white sheet was chosen as a backdrop which was back-lit to overcome the shadows from the surrounding lights and reflections from the sun.

Real-world coordinates were achieved by calibrating the cameras to account for lens distortion. By using Tsai's point analysis technique (Tsai 1986) we could find out where the cameras were in space and what their orientation was in comparison to the origin (predefined as beneath the center of the screen). The real-world coordinates were then calculated from the two projected views from the video cameras.

As noise was an inherent part of the system in all stages, Linear Kalman Filtering was used to filter out noise. The 2D head and hand positions in each image and the triagulated 3D world-position points of the head and hand were all filtered to some degree. The Kalman filtering smoothed out values so that when the feature points got to the display computer they did not jump around so much as the jumping around produced flickering on the display screen.

The physical space required for this system was quite large and more space would be preferred. As our system was confined in a small room (about 3m wide by 5m long by 2.4m high) the tracking system was adjusted to take this into account.

3.2. Feature Extraction

We implemented the Watching Window so that it would be simple to use and would accommodate a large range of people. Our aim was for people to be able to just walk in and use the system so we decided on an unencumbered vision tracking system. Head tracking was done in the following

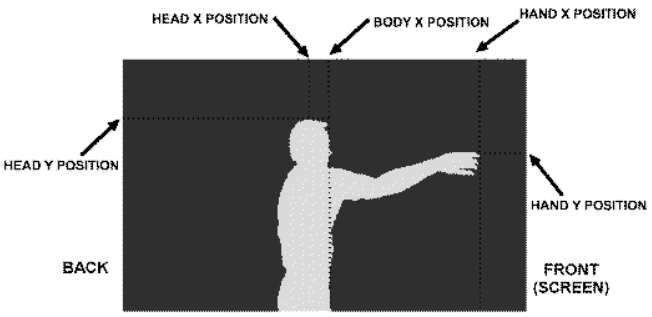

Figure 2. A Camera Frame demonstrating basic head and hand finding

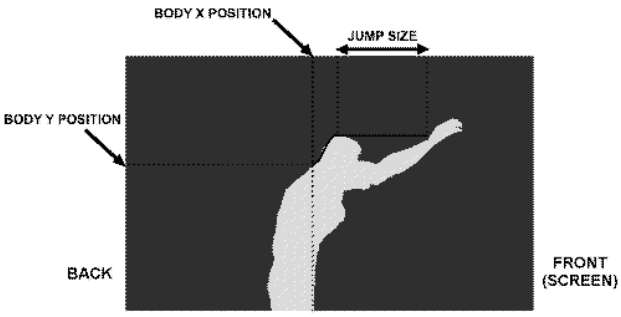

Figure 4. Distinguishing between hand and head

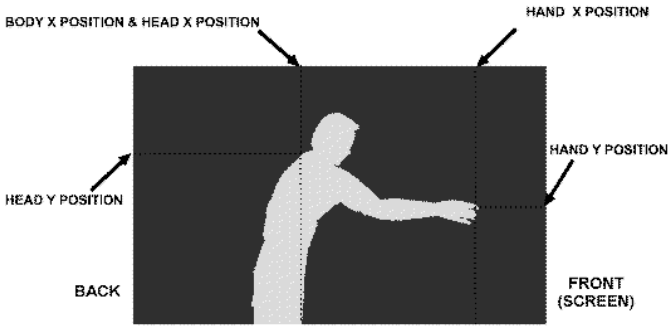

Figure 3. Mistaking the back for the head

way for each frame:

1. Assume that the head is lower than the top of the image and that the body or legs intersect the bottom of the frame. This was about 98% of the time with our cameras and setup. Take the most recent frame and subtract the initial scene (without a user in it) from it.

2. Search for the edge of the body (the silhouette extracted from the background subtraction) by scanning along the pixels at the bottom of the image from the side facing towards the screen.

3. Once the body is found search backwards from this point for the highest point along the outline. This is shown in Figure 2 Come down slightly and that is the center of the head. The amount moved down depends on the total height of silhouette and gives a good center head position for body positions near and far from the camera. The amount of up and down movement of the average is only about 50cms so having the amount dependent upon height is quite robust in normal circumstances.

4. Search in front of the body for the forward most point. If this position is past a threshold value then the arm is considered to be out and pointing. Near the end-point of the outstretched blob is taken as the center of the hand. As this was found to be inaccurate and jumping around between the pixels an average of the 50 forward most pixels was used as the hand position. Some trials were done seeing if we could point with a finger or not depending on the amount of distribution of the pixels but this was found to be too inaccurate.

5. Triangulate the head and hand positions from both cameras to calculate their positions in world-space. This information is then sent to the display computer where a ray is cast into the scene starting at the head position and going through the hand position to see what they are pointing at.

6. The display computer uses the positional and pointing information from the tracking computer to react to the users movements. By constantly changing the viewpoint of the OpenGL context to match the head position the correct view of the image is always displayed on the screen. Due to the noise and differences between users a point is drawn on the screen where the head-hand ray intersects it. This becomes occluded by objects in front of the screen to provide both help for the user and as another visual cue for the 3D scene.

7. If the body is not found then the tracking computer cannot calculate the head and hand positions and does not send any packets of information to the display computer. The viewpoint of the display computer is only updated with incoming head positions so if no head position information is read it remembers the last point.

Some problems with this approach are shown in Figure 3 where the back is incorrectly recognised as the head. Prob-

lems also arise when parts of the silhouette rise up behind the head like collars and backpacks. By searching for a local maxima we correctly find the head but introduce the error of not knowing where the head ends and the arm begins. This is shown in Figure 4. By limiting the amount of distance away from the body line the head could be it correctly finds the head in most silhouette images.

3.3. Modeling

The display system was set up to demonstrate models floating approximately half a meter in front of the screen. This was seen as the effective viewing distance as it gave a good amount of parallax while keeping the model within the viewing area of the screen. If the model were to go outside the viewing area it would appear to be cut in half and would destroy the 3D illusion. To enhance the 3D effect we experimented with using simple red-blue glasses. This provided us with a cheap means of creating a superior 3D looking object. The rendered photo-realism of the images was sacrificed for wire-frame models as the wire-frames look better with the 3D glasses on.

We also implemented a very basic repulsion interactive system, controlled with the hand-blob from the tracking system. The hand-blob repels control points around the outside of the virtual object. Using the blendeforming technique (Mason & Wyvill 2001) for deforming objects based on local control points we implemented a virtual potters wheel. This is a cylindrical deformation applied over a rotating cylinder. The outer boundary of the control points originate on the surface of the object when there is no deformation, so when you alter the control points the objects surface goes through all the control points. Figure 5 shows the bounding cylindrical mesh and the control points for a basic example.

There were 12 control points down one side of the deformation volume in this example which altered the contained vertices into a nice cardinal spline through the control points (Figure 6).

The deformation technique works by defining a deformation volume and squashing and stretching the space inside according to a user defined function. It compresses where the volume will be shifting into and expands the volume behind the deformation to keep it continuous.

When the hand is near a control point the control point moves away from the hand position in a predefined direction. A model (such as a cylinder) is initially approximated by a list of vertices. These vertices are then projected forwards through the deformation space. This is calculated every frame and for models with less than about 1000 vertices it can do this in real-time (70 frames per second) on a GeForce2.

Some other simple applications were implemented to test

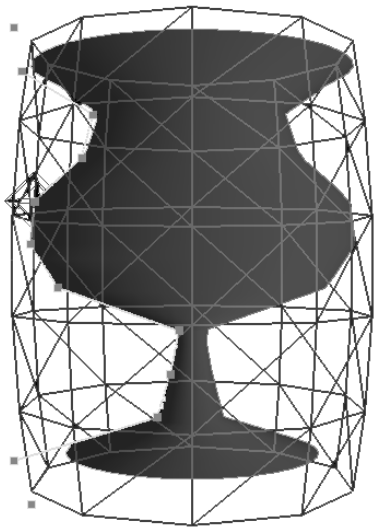

Figure 5. A bounding deformation mesh with control points deforming a cylinder

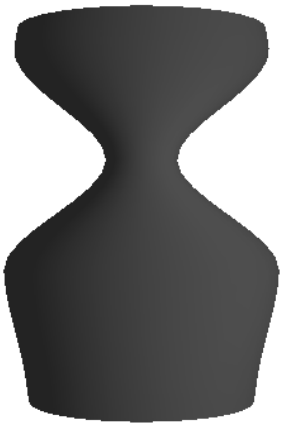

Figure 6. A deformed model using a cardinal spline to interpolate through the control points

the effectiveness of the FTVR system with both the rendered images and the red-blue stereo image. Scenes included floating objects, terrain fly-throughs and a starfield simulation.

4. Results and Discussion

The responsiveness of the system is important to enhance the perception of reality. The background subtraction was very noisy, although with a clean background we could reduce the amount of filtering on the images in 2D and 3D space. Since the background was less noisy less Kalman filtering was needed. As Kalman filtering flattens out spikes in the positional data, a small delay may be noticed if you move rapidly. This is because it may mistake the beginning of the sudden movement as a spike. This is unlikely to happen with the head-data as the head position movements are relatively stable so hardly any Kalman filtering was needed. This made the system much more responsive. The hand-data needed much more filtering but the effects were not noticeable.

If the filtering is increased you quickly get used to the lag and it becomes unnoticeable as you start adjusting and start moving in slower motion.

It is not hard to break the algorithms for feature tracking but this nearly never happens unless the user is trying to break them (and then it happens very easily). As the head is usually quite stable and balanced it is quite easy to track without much filtering. The hand blob is not so stable and the area of the hand on the image is only a few pixels it can be quite jerky when pointing, even if the hand is stationary. More filtering is used for the hand but an automatic procedure for determining the right amount of filtering would be ideal.

The red-blue glasses greatly increase the illusion of a floating 3D object despite the lack of rendered detail. One way of overcoming this loss of detail would be to use polarised glasses with two different polarised projectors. Combined with the pottery illusion responding to your 'touch' a rendered 3D image would result in a very powerful illusion. This is the reason why many VR systems use flickering glasses and displays as they allow the users to view full colour 3D models.

While interactive displays are fun people also enjoyed experimenting with the static scenes. Users wanted to see how well they could move around and look at the virtual models. While this was most effective with the red-blue glasses on, without them there was still a strong illusion if one eye was closed. The information received by two unencumbered eyes tells the user that the screen is flat and destroys the effectiveness of the 3D illusion. With only one eye open this visual cue is not recognised and the scene appears to that one eye like a perfectly normal 3D scene. Ignoring the 2D cues from both eyes could possibly be a learned skill and that with practice the user might be able to ignore the real-world cues (but not the VR cues) and see the screen effectively in 3D without goggles. More research should be done to see if this is actually the case.

Despite the simple algorithms and basic approach to the vision tracking system, it is still quite robust and accurate enough for what it was intended - an unencumbered 3D VR model viewer.

5. Conclusion

Overall, we have produced a cheap practical implementation of a real-time, unencumbered vision tracking system. By being easy to implement it offers an excellent platform for creating immersive FTVR applications, such as our blendeformer modeller and viewer, that are simple and intuitive to use. Interactive situations are also quite good as movement seems to distract from the illusion of it only being 2D. With 3D goggles the scene looks (graphically) worse because of the difficulty in providing accurate rendering of objects but produces a better illusion of 3D. Creating more realistic 3D images is important and will be a part of ongoing research with the Watching Window.

Expanding the system to include more cameras will help in the tracking accuracy. Increasing the number of screens will facilitate the immersive experience. Unfortunately expanding the system will also mean increasing the amount of space required. The system as it is is currently set up allows only 1 person at a time to use it.

Active research for the Watching Window will focus on improving the head and hand tracking, implementing a constraint model to test if certain model positions are possible or not and adjusting the tracking positions accordingly. Dealing with background noise so that the large amount of room the backdrop takes up will become available for other purposes is quite important for small labs. Although our back-lit sheets were adequate it would be better to develop a way of dealing with normal or even noisy backgrounds.

Dealing with background noise effectively could open up any area as a control space and the Watching Window could become a go anywhere - be any place, universal display and interface system.

References

Agre, P. E. (2001). Your face is not a bar code, Online. http://dlis.gseis.uda.edu/people/pagre/bar-code.html.

Bobick, A. (1999). Large occulsion stereo, *International Journal of Computer Vision* .

Kitamura, Y., Konishi, T., Yamamoto, S. & Kishino, F. (2001). Interactive stereoscopic display for three or

more users, *Computer Graphics Proceedings, Siggraph 2001*, pp. 231–240.

Leigh, J. (2000). Electronic visualization laboratory, Online. http://www.evl.uic.edu/spiff/maxine.

Mason, D. & Wyvill, G. (2001). Blendeforming: Ray traceable localized foldover-free space deformation, *Proceedings of Computer Graphics International 2001*, pp. 183–190.

Mazuryk, T. & Gervautz, M. (1996). Virtual reality. history, applications, technology and future, Research Paper, Institute of Computer Graphics, Vienna University of Technology. Online. http://visinfo.zib.de/EVlib/Show?EVL-1996-16.

Treadgold, M. S. (2001). *Out of the fish tank*, Master's thesis, University of Otago.

Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3d machine vision, *Technical report*, IBM, T. J. Watson Research Center.