Head-Tracked Stereo Viewing with Two-Handed 3D Interaction for Animated Character Construction

Russell Turner[1], Enrico Gobbetti[1,2], and Ian Soboroff[1]

1. UMBC, CSEE Department, Baltimore MD 21228-5398, USA
2. CESDIS, NASA Goddard Space Flight Center, Greenbelt, MD, USA

E-mail: {turner|gobbetti|ian}@cs.umbc.edu

TO APPEAR IN PROC. EUROGRAPHICS'96

Abstract

In this paper, we demonstrate how a new interactive 3D desktop metaphor based on two-handed 3D direct manipulation registered with head-tracked stereo viewing can be applied to the task of constructing animated characters. In our configuration, a six degree-of-freedom head-tracker and CrystalEyes shutter glasses are used to produce stereo images that dynamically follow the user head motion. 3D virtual objects can be made to appear at a fixed location in physical space which the user may view from different angles by moving his head. To construct 3D animated characters, the user interacts with the simulated environment using both hands simultaneously: the left hand, controlling a Spaceball, is used for 3D navigation and object movement, while the right hand, holding a 3D mouse, is used to manipulate through a virtual tool metaphor the objects appearing in front of the screen. In this way, both incremental and absolute interactive input techniques are provided by the system. Hand-eye coordination is made possible by registering virtual space exactly to physical space, allowing a variety of complex 3D tasks necessary for constructing 3D animated characters to be performed more easily and more rapidly than is possible using traditional interactive techniques. The system has been tested using both Polhemus Fastrak and Logitech ultrasonic input devices for tracking the head and 3D mouse.

Keywords: Interactive 3D Graphics, Stereoscopic Display, Head-Tracking, Two Hand Input, Virtual Tools, 3D Character Modeling, Animation Systems

1. Introduction

The creation of computer animated characters, from simple caricatures to realistic-looking humans, is fundamentally a creative process. It is therefore essential that animation software systems be accessible to non-technical, creative animation artists. Ideally, the computer should provide for the computer-based animator an electronic artistic medium as flexible as any traditional medium such as clay or paper and pencil.

The latest layered construction techniques, which model anatomical features, have shown promise in creating character models that deform automatically around an articulated skeleton. But purely geometric models, although they can be very expressive, usually require too much user intervention to achieve realistic-looking results. Physically-based elastic models provide more realistic behavior, but at the price of increased CPU requirements and difficulty of control. With proper use of constraints, however, deformable models can be controlled by kinematic and geometrical models. Recent improvements in processor speeds now make it possible to simulate certain kinds of moderately complex physical models in real-time. For these reasons, we believe that a hybrid approach in which layered models are constructed using a combination of geometric, kinematic and physically-based techniques, is the most promising one. Our LEMAN (Layered Elastic Model ANimation) system [Turner 93][Turner 95], allows three-dimensional animated characters to be built up from successive layers of skeleton, muscle, fat and skin. Using an artist's anatomical approach, the character is represented as a simulated elastically deformable skin surface which is wrapped around a kinematically modeled articulated figure. The character may then be animated by moving the underlying figure.

The ideal 3D character model should provide a good compromise between interactive speed and realism, and between animator control and physically realistic behavior. The exact details of such a model are no more important, however, than the types of interactive technique used to construct and animate it. High-performance 3D graphics workstations and a variety of multi-dimensional input devices have begun to make highly interactive, direct manipulation environments practical. Finding the right kinds of interaction metaphors by which these devices can control a 3D character model, however, requires experimentation with many of the various possibilities.

It is increasingly evident that the classical user-interfaces based on 2D input devices and 2D mindsets are inadequate for these kinds of applications, which require users to specify complex spatial information. In particular, the low-bandwidth communication between the 2D user-interface and the application, together with the inherent limitations of the 2D mouse for interactive 3D motion specification, make it extremely difficult for the users to perform their work with simple intuitive actions [Conner 92][Gobbetti 93][Gobbetti 95]. The feedback provided to the users is also a problem: the limited information about the structure of the three-dimensional world conveyed by a fixed visual image often forces the application to rely on multiple views to provide additional depth information. This requires users to combine the separate views to form a mental model of complex objects, adding further complexity to what is often a very difficult task [Herndon 92], and forcing users to concentrate on how to obtain what they want instead of the task itself.

Based on these limitations, it is evident that user interface metaphors which enable users to work directly in three dimensions must be developed. Virtual reality research, starting from the basic assumption that human beings are well equipped to interact with the world they live in, should strive to make users interact with virtual worlds in the same way they interact with the real world, thus making the interaction task much more natural and easier to learn. The difficulties associated with achieving the important goal of immersion have led researchers in virtual environments to concentrate more on the development of new input and display devices than on higher-level techniques for 3D interaction. It is only recently that techniques for interaction with synthetic worlds have tried to go beyond straightforward interpretation of physical device data [NSF 92][Conner 92]. By contrast, recent research in the 3D interaction field has focused on exploring responsive 3D interfaces with better affordances, functional fidelity and mental appeal [Conner 92][Robertson 93][Herndon 94][Gobbetti 95]. This research, while dramatically improving the expressive power of desktop computers to accomplish 3D tasks, has not taken advantage of the latest developments of virtual reality technology to increase the bandwidth and fidelity of man-machine communication. In most cases, the interaction tools reside in 3D space, but are operated with the 2D mouse and presented to users using a conventional perspective view on a workstation monitor.

We believe that it is important to bridge the gap between these two research directions by applying the results of immersive virtual reality hardware research to enhance, rather than completely replace, current desktop configurations, and to use the additional possibilities offered by such improved configurations to develop new interaction metaphors. In doing so, we believe that we can develop a restricted, but high-quality form of virtual reality, one that would give a compelling sense of immersion within the confines of the desktop world, but would at the same time be expressive and ergonomic enough for people to use for extended periods of time to do practical work. Recent research on "fishtank VR" [Ware 93] has started to focus on these problems.

In this paper, we demonstrate how a new interactive 3D desktop metaphor based on two-handed 3D direct manipulation registered with head-tracked stereo viewing can improve the animated character construction process. First, we will provide a short overview of the LEMAN animated character model. Next, we will describe our device configuration and interaction metaphor. Then, we will describe how some important character construction tasks are accomplished using our interface. The paper concludes with a discussion of related work, an evaluation of the results obtained and a view of future work.

2. The LEMAN Animated Character Model

As described in detail in [Turner 93] the development of the elastic surface layer model is an attempt to simulate the layered anatomical structure of an animated character so as to minimize computational effort by using for each layer the modeling techniques which are most appropriate. Since the skin is the outermost layer and the only one directly visible, we concentrate CPU effort on this by modeling it as a simulated deformable elastic surface [Terzopoulos 87][Lee 95]. The underlying layers are then modeled using geometric and kinematic techniques which act on the surface as force-based constraints. In particular, reaction constraints prevent the surface from penetrating the underlying muscle and fat layers, pushing the skin out, while point-to-point connective tissue spring constraints pull the surface in.

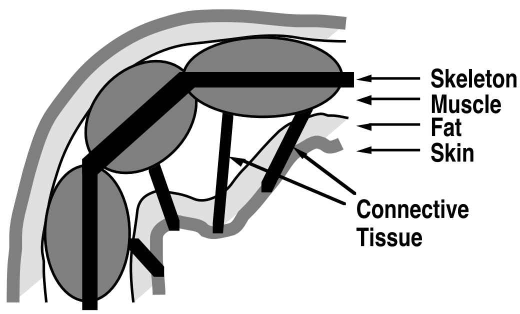

Figure 1. Components of the Elastic Surface Layer Model

The skin is implemented as a simulation of a continuous elastic surface discretized using a finite difference technique [Terzopoulos 87]. The surface is represented as a rectangular mesh of 3D mass points, together with their physical characteristics (e.g. mass, elasticity) and their current state information (e.g. position, velocity). When the numerical solver is turned on, the state is evolved over time at a fixed simulation time step. At periodic intervals, the surface is rendered on the screen, resulting in a real-time continuous simulation.

The surface is bounded at its poles and constrained by reaction constraints, point-to-point spring forces and other environmental forces such as gravity and air pressure. At each time step, the spring forces acting at each surface point are calculated, according to the Hooke's law spring constant, and added to the total applied force for that point. Other environmental forces are then calculated and added in to the total applied force. Then the point is checked to see if it is inside any of the muscle layer surfaces, in which case reaction constraints are applied. Rather than simply adding forces, reaction constraints remove undesirable forces and replace them with forces that move the elastic surface towards the constraint surface with critically damped motion [Platt 88]. Forces perpendicular to the constraint surface are not affected, so the elastic surface may slide along the constraint surface until it reaches a local energy minimum.

3. A User Interface for Character Construction

Character construction is an important creative aspect of physically-based layered character animation since so much of the character's animated behavior is determined by its structure. We therefore focus our efforts on improving the user-interface for this task which, given the structure of the layered model, is mainly concerned with providing effective ways to perform the following operations [Turner 93]:

- **Skeleton building:** as a first step, a hierarchical skeleton must be created. This requires specification of all the joints in the articulated figure, their positions and orientations with respect to each other, and their hierarchical relationships. In addition, the joints' range of motion must be specified, which often requires manipulating the figure into various postures to observe the effects of different values. The result is a "stick figure" skeleton of pure joints, and is analogous to an armature used in traditional sculpture or stop motion animation;

- **Adding muscles:** the next step is to "flesh out" the skeleton by adding solid geometrical shapes to the joints which move with the skeleton and form the basic shape of the articulated figure. These shapes, which correspond to body shape masses in traditional figure drawing, constitute the muscle layer of the layered character model and are based on deformed implicit surfaces such as spheres and superellipses. Their parameters and dimensions, as well as orientations with respect to their associated joints must be specified;
- **Connecting the skin:** the physically-based skin surface is attached to the figure at the skin boundaries and also through simulated connective tissue, which takes the form of "rubber-band" force constraints between points on the skin surface and points on the underlying muscle layer. This is a complex spatial task since it requires associating every point on a global rectangular surface mesh (the skin) with a local point on the surface of a hierarchical shape (the articulated figure). In addition, the stiffness of each rubber-band constraint may be specified globally or individually;
- **Sculpting the fat layer:** the fat layer is represented as a geometric distance separating the skin a muscle layers. It is specified as a simple thickness parameter at each point on the skin, and therefore corresponds to sub-cutaneous fat which moves with the skin. This operation therefore involves associating a single scalar parameter with each point on the skin surface.

All these operations require the specification of 3D data and the understanding of complex 3D information. The user interface should therefore be designed so as to facilitate these operations.

3.1 Device Configuration and Interaction Metaphor

In our configuration, a six degree-of-freedom head-tracker and CrystalEyes shutter glasses are used to produce stereo images that dynamically follow the user head motion. We have used both Polhemus Fastrak magnetic and Logitech ultrasonic sensors to track the head and 3D mouse motion. As in [Deering 1992], the left and right viewing frusta are recomputed at each frame from the physical position of the viewer's eyes and the physical position of the display surface. Each of the frusta is a truncated pyramid with apex at the eye position and edges passing through the corners of the rendering viewport. The position of the left and right eyes are computed from offsets from the tracked head-position, while the position of the display surface in tracker coordinates is determined by a calibration step that has to be re-executed only when the screen monitor or the tracker reference frame are moved. Our current calibration procedure simply consists of measuring the position of the four corners of the workstation monitor with the 3D mouse. Thanks to the registration between virtual and physical coordinates, 3D virtual objects can be made to appear at a fixed location in physical space which the user may view from different angles by moving his head. The combination of physically accurate perspective, stereo viewing, and motion parallax provide a compelling illusion of the existence of the simulated 3D objects.

To construct 3D animated characters, the user interacts with the simulated environment using both hands simultaneously: the left hand, controlling a Spaceball, is used for 3D navigation and object movement, while the right hand, holding a 3D mouse, is used to manipulate the objects appearing in front of the screen through a virtual tool metaphor. In this way, both incremental and absolute interactive input techniques are provided by the system. This combination of input techniques provides several benefits.

Thanks to head-tracking, camera motion can take advantage of simultaneous position and velocity control, and a single control mode has characteristics which are at the same time appropriate for close inspection, exploration, and navigation [Ware 90]. In our system, the Spaceball incrementally controls a virtual vehicle, and tracked head and right hand positions are interpreted local to that vehicle. Relying on an incremental device such as the Spaceball for vehicle control reduces the user's fatigue, as opposed to solutions based on absolute devices such as those presented in [Ware 90], since the left hand can rest on the Spaceball support and only very small finger movements are necessary for motion control.

The different components of an animated character are created, connected and manipulated using virtual tools which are encapsulations of a physical appearance and a behavior [Conner 92][Gobbetti 93]. Since tools are manipulated with the right hand using absolute input, the user can have the visual impression of touching the virtual objects that are close to him. These virtual tools are in some ways 3D analogs of the types of interactive tools found in typical 2D drawing programs (e.g. select, resize, create-circle, spray-paint). However, since they are inherently three-dimensional, they are capable of performing more sophisticated 3D spatial tasks, and are often more intuitive for 3D operations than their two-dimensional counterparts since they correspond more closely to a real-world tool. Like a 2D drawing program, the various tools are arranged in a toolbar from which the current tool may be selected using the 3D mouse. Once selected, a copy of the tool is displayed at the current position of the 3D mouse, representing a 3D cursor. A visible "wand" extends a few centimeters out from the cursor and is used for picking objects and specifying positions in space, and a button on the 3D mouse allows picking and dragging operations. The large number of degrees of freedom and direct-manipulation capabilities of these virtual tools allow complex interactive operations, which might otherwise require several 2D widgets and 2D mouse, to be performed naturally with a single virtual tool.

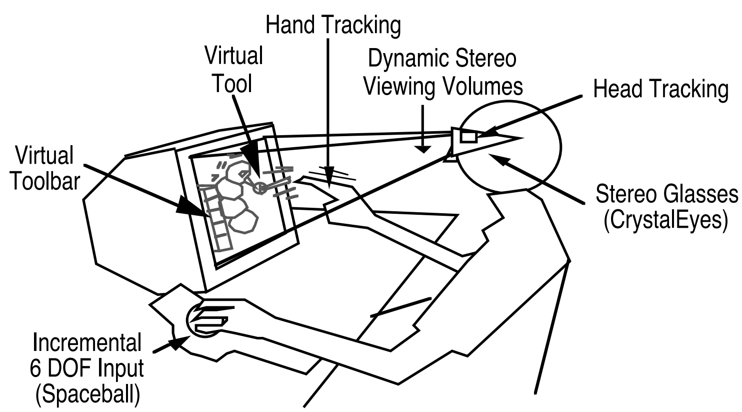

Figure 2. Device configuration and interaction metaphor

3.2 Spatio-temporal Realism

To give users the impression of manipulating real objects, it is important that the lag between their movements and the effects in the synthetic world be as small as possible. This is obtained in LEMAN by decoupling the simulation and tracking activities. At each frame, the following actions are taken:

- the latest values of the 3D mouse and head tracker sensors are read, and virtual camera position and virtual tool position are updated;
- events from the various devices are handled, and the behavior of the active tool is executed;
- if the skin is deforming, simulation steps are executed until the time that remains in current frame is less or equal to the time spent rendering the previous frame;
- the latest values of the motion tracker sensors are read again, and a new frame is rendered with the latest positions of the virtual camera and tool.

Since the motion tracker is sampled at a much higher rate than the frame rate of the application (60 Hz vs. 10 Hz), and since on our machine (an SGI Onyx RE2) computing the simulation is more expensive in time than rendering the character, reading tracker values twice per frame (once before computing the application behavior and once just before rendering the frame) allows the reduction of the most important lags, i.e. those of the objects directly associated with the physical position of the user. The perceived hand and head motion lag is determined by the rendering time and does not depend on the simulation time. This lag reduction technique is similar to just-in-time-synchronization [Wloka 95] with a priority given to the user's position as in DIVER [Pausch 93].

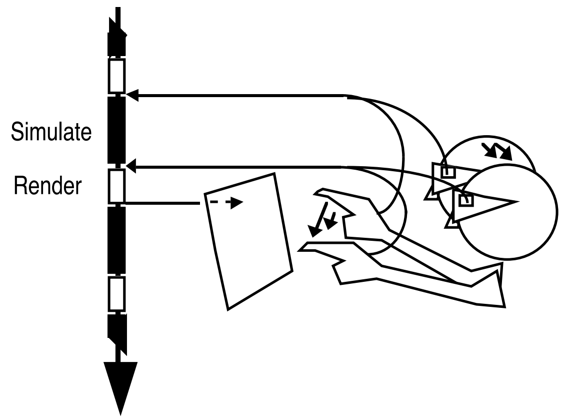

Figure 3. LEMAN Time-line

4. Constructing a Character

With LEMAN, the animator can build up a three-dimensional character model from scratch, adjusting parameters and moving it interactively throughout the process to test its appearance and behavior. Once the character is constructed, it can be saved to a file for future use. The main operations involved in character creation have been implemented so as to take advantage of head-tracked stereo-viewing and a two-handed direct-manipulation interface. The hand-eye coordination made possible by the registration between virtual and physical space which allows a variety of complex 3D tasks necessary for constructing 3D animated characters to be performed more easily and more rapidly than is possible using traditional interactive techniques. The following sections describe the main steps involved in the creation of a 3D characters, contrasting our user-interface with traditional desktop approaches such as those used in our previous work [Turner 95]. The pictures presented in the following sections were taken by tracking the position of the camera using the Logitech ultrasonic head tracker. Figure 4 (color plates) shows a user in the process of constructing a character.

4.1 Skeleton Building and Muscle Creation

In a direct manipulation user environment, it is usually easier to build both muscle and skeleton layers simultaneously, since the geometric link shapes provide something tangible to manipulate. Articulated figures are constructed by selecting one of the shape creation tools (e.g. sphere, superellipse) and selecting a point in space causing a new joint be created at that location. Dragging the tool, by holding the 3D mouse button down while changing its position and orientation, changes the size and shape of the muscle associated with the joint. For example, drag-translating the tip of the sphere creation tool wand will vary the radius of the sphere, while drag-rotating the superellipse creation tool will vary its curvature parameters. The direct correspondence between the physical location of the hand tracker and the effect on the manipulated objects gives invaluable help when creating the character, since the size of the character's muscles and the position of the joints are controlled directly in the physical space (see Figure 5, color plates). Traditional user-interfaces such as the one used in [Turner 95] offer only indirect control over these parameters.

The system maintains a current joint, which is displayed in a highlighted form and can be specified using the selection tool. Newly created joints are made children of the current joint, allowing the skeleton hierarchy to be easily specified. Selecting and dragging an existing joint with a shape creation tool will modify the parameters of the existing joint rather than creating a new one. Model editing and model creation can thus be done with a single tool. By combining head-tracking, vehicle movements controlled by the Spaceball, and model creation using 3D tools using a 3D mouse, an entire skeleton can be created and viewed from all angles in a few minutes without any interaction mode changes in the application.

In order to check that the articulated figure has been properly constructed, the user must be able to move the skeleton into various postures easily. This is done in the LEMAN system using the inverse kinematics tool, which implements a standard technique used in computer-based character animation that allows the posture of a kinematic chain (e.g. arm, leg, spine) to be specified by moving the end joint of the chain, known as the end-effector. To change the posture of the character's skeleton, the user simply selects the end-effector joint with the inverse kinematics tool and drags it to the desired new position. The entire kinematic chain up to the nearest branching point in the skeleton hierarchy will then follow the motion of the end-effector. The direct correspondence between physical and virtual locations allows for a direct control in space of the end-effector movement, while the control offered by a non head-tracked application can only be approximate. The behavior of the character can thus be tested more effectively. Figure 6 (color plates) shows the inverse kinematics positioning of a character's body.

4.2 Attaching the Skin

The skin is created and attached to the articulated figure using the skin attachment tool. The skin surface is created and its top and bottom boundaries are fixed to their respective joints by selecting the joints with the tool in sequence from bottom to top. Once both boundaries are specified, the differential equation solver starts up and the skin, initially in a spherical shape, shrinks down around the muscle layer of the character (see Figure 7, color plates). The connective tissue attachments between the skin and the muscle layers may be created by issuing a global "attach" command which causes all points on the skin surface to be attached to the nearest point on the muscle perpendicular to the skin surface. These attach points may then be adjusted with the skin attachment tool by selecting individual points on the skin surface, dragging the wand to the desired attach point on the muscle surface, and releasing. Selection of regions of the skin surface can be done naturally by touching the virtual skin with the tool. The user can view the effects of the changes from different angles while using the skin attachment tool, simply by moving his head or, if larger motions are needed, by using the Spaceball. Errors caused by editing operations using a single perspective view are thus reduced, since motion parallax effectively coveys the needed 3D information.

4.4 Sculpting the Fat Layer

The final character shape may be sculpted to a fair degree by locally adjusting the thickness of the fat layer. Since the fat thickness is a single parameter associated with each point on the skin surface, it can be controlled very naturally using the "liposuction" tool, which increases or decreases the fat thickness of the skin in a Gaussian distribution around the selected point on the skin surface (see Figure 8, color plates). The implementation of fat thickness is described in detail in [Turner 93]. This tool is analogous in some ways to a spray-can tool in a 2D paint program in that the longer the user holds down the mouse button, the more fat is injected into the skin. The orientation of the tool along its main axis differentiates between injection or removal of fat: a clockwise rotation with respect to the initial orientation injects fat, while a counter-clockwise rotation removes it. Figure 9 (color plates) shows the final character, which was created in only a few minutes. Texture-maps have been added using a conventional user-interface.

5. Related Work

5.1 Interactive 3D Modeling

It is only recently that some 3D interactive desktop modeling systems have started to make good use of 3D space for interaction purposes. AT&T's embryonic CAD modeler [Weiner 89] was one of the first applications showing the importance of multi-modal input for shape design. However, its user interface made little use of 3D space, mostly using three-dimensional menus and limiting direct manipulation to point dragging and orientation specification. 3-Draw [Sachs 90] and the polygonal modeler described in [Shaw 94] extend these techniques to two-handed input, an idea first proposed for 2D user-interfaces by Buxton and Myers [Buxton 86]. Both systems implement 3D two-handed input by using two Polhemus trackers to let users manipulate the modeled surface's orientation and a 3D cursor concurrently. Since these systems do not provide accurate head-tracked viewing, it is difficult to correlate the physical motions with their effect in the synthetic world. The lack of incremental input capabilities is likely to increase the user's fatigue. The object-oriented graphical toolkits *UGA* [Zeleznik 91], from Brown University, *Inventor* [Strauss 92], from Silicon Graphics, and *VB2* [Gobbetti 93], developed by one of the authors at the Swiss Federal Institute of Technology in Lausanne, demonstrate how the increase in correlation between manipulation and effect on controlled objects makes three-dimensional widgets more powerful and simpler to understand than their two-dimensional counterparts. Modeling applications have been developed using each of these toolkits [Gobbetti 95][Conner 92], but none of them have so far taken advantage of registration between virtual and physical world to provide direct interaction with synthetic objects. Michael Deering's HoloSketch [Deering 95] offers both head-tracking and hand-tracking to improve the usability of a shape modeling application, but no incremental devices are used to aid object inspection and scene navigation.

5.2 Head-tracked Stereographic Visual Displays

Stereo graphics has long been used to enhance the user's perception of depth in comparison to standard graphics displays. However, most of the work in stereo-rendering for computer graphics deals with the generation of stereo pairs in situations where the position of the viewer is not well controlled (as in movie theaters or when generating images on a desktop workstation without using head tracking). The objective of this kind of research is to obtain easy-to-control (and often exaggerated) "stereo-effect". Examples are discussed in [Roese 79], [Schmandt 83], [Hibbard 91], [Hodges 91], [Hodges 92], and an excellent overview of these techniques is presented in [McKenna 92]. These models often give the user direct control of the stereo projection (by adjusting intuitive parameters such as "image separation"), and cannot be used when exact registration between virtual and physical locations is needed. With these systems it is impossible to obtain the hand-eye coordination necessary for direct interactive manipulation. Recently, some authors have concentrated on the exact mapping between virtual and physical locations using head-tracked stereo-rendering on workstation monitors [Deering 92][Deering 95] or projection screens [Cruz-Neira 93][Krueger 94]. This technique is particularly important for implementing augmented reality systems. In our system, we adapted the approach described in [Deering 92] for correcting distortions caused by the curvature of screens of CRTs whose screen is a spherical section, to our display configuration, based on a Sony CRT with a cylindrical section-shaped screen.

6. Conclusions and Future Work

In this paper, we have presented how a new interactive 3D desktop metaphor based on two-handed 3D direct manipulation registered with head-tracked stereo viewing can be applied to the task of constructing animated characters. In our configuration, a six degree-of-freedom head-tracker and CrystalEyes shutter glasses are used to produce stereo images that dynamically follow the user's head motion. 3D virtual objects can be made to appear at a fixed location in physical space which the user may view from different angles by moving his head. To construct 3D animated characters, the user interacts with the simulated environment using both hands simultaneously: the left hand, controlling a Spaceball, is used for 3D navigation and object movement, while the right hand, holding a 3D mouse, is used to manipulate the objects appearing in front of the screen through a virtual tool metaphor. In this way, both incremental and absolute interactive input techniques are provided by the system. Hand-eye coordination is made possible by registering virtual and physical space, allowing a variety of complex 3D tasks necessary for constructing 3D animated characters to be performed more easily and more rapidly than is possible using traditional interactive techniques.

We have experimented with both Polhemus Fastrak magnetic trackers and Logitech ultrasonic trackers to input the head and 3D mouse motion. While the Polhemus is less obtrusive and does not suffer from occlusion problems, its performance degrades considerably when operated close to the workstation monitor due to emitted magnetic fields, which cause jitter, and the presence of metal in the environment, which distorts position values and lowers correspondence between the physical and virtual coordinate systems. The Logitech trackers do not suffer these kinds of problems and work quite well close to the screen, as long as care is taken not to occlude the sensor microphones.

Our future work will concentrate on developing more tools for character construction and character animation, the goal being the creation of a system where all interaction is done in three dimensions. We would also like to improve registration between virtual and physical space by developing visual calibration procedures. Such a system, we believe, would provide a prototype user-interface metaphor useful in a variety of highly-interactive desktop VR applications in areas such as surgical simulation, surface modeling and scientific visualization as well as animation.

Acknowledgments

This work was supported by National Science Foundation Interactive Systems grant number IRI-9503093. Development of the original LEMAN system was supported by a grant from the Swiss National Foundation. We would also like to thank the reviewers for their helpful comments and suggestions.

Bibliography

Butterworth J., Davidson A., Hench S., Olano TM (1992) 3DM: A Three Dimensional Modeler Using a Head-Mounted Display. *Proceedings SIGGRAPH Symposium on Interactive 3D Graphics*: 135-138.

Buxton W, Myers B (1986) A Study in Two-Handed Input. *Proceedings of the ACM SIGCHI*: 321-326

Cruz-Neira C, Sandin DJ, DeFanti TA (1993) Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. *Proceedings ACM SIGGRAPH*

Conner DB, Snibbe SS, Herndon KP, Robbins DC, Zeleznik RC, Van Dam A (1992) Three-Dimensional Widgets. *ACM SIGGRAPH Symposium on Interactive 3D Graphics*: 183-188.

Deering M (1992) High Resolution Virtual Reality. *Proceedings ACM SIGGRAPH* : 195-202.

Deering M (1995) HoloSketch: A Virtual Reality Sketching/Animation Tool. *ACM Transactions on Computer-Human Interaction*. 2(3): 220-238.

Gobbetti E, Balaguer JF (1993) VB2: A Framework for Interaction in Synthetic Worlds. *Proceedings ACM UIST*.

Gobbetti E, Balaguer JF (1995) An Integrated Environment to Visually Construct 3D Animation. *Proceedings ACM SIGGRAPH*.

Herndon KP, Zeleznik RC, Robbins DC, Conner DB, Snibbe SS, Van Dam A (1992) Interactive Shadows. *Proceedings UIST*: 1-6.

Herndon KP, van Dam A, Gleicher M (1994) Workshop Report: The Challenges of 3D Interaction. *SIGCHI Bulletin*, October.

Hibbard E (1991) On the Theory and Application of Stereographics in Scientific Visualization. *EUROGRAPHICS State of the Art Reports*: 1-21.

Hodges L (1991) Basic Principles of Stereographics Software Development. *Proceedings SPIE 1457*: 9-17.

Hodges L (1992) Tutorial: Time-Multiplexed Stereoscopic Computer Graphics. *IEEE Computer Graphics and Applications*: 20-30.

Krueger W, Froehlich B (1994) The Responsive Workbench. *IEEE Computer Graphics and Applications* 14(3):12-15

Lee Y, Terzopoulos D, Waters K (1995) Realistic Modeling for Facial Animation. *Proceedings ACM SIGGRAPH*: 55-62.

McKenna M, Zeltzer D (1992) Three Dimensional Visual Display Systems for Virtual Environments. Presence 1(4): 421-458.

NSF (1992) Research Directions in Virtual Environments, *NSF Invitational Workshop,* University of North Carolina at Chapel Hill, March 23-24: 154-177.

Pausch R, Conway M, DeLine R, Gossweiler R, Miale S (1993) Alice and DIVER: A Software Architecture for Building Virtual Environments. *Proceedings InterCHI.*

Platt JC, Barr AH (1988) Constraint Method for Flexible Models, *Proceedings ACM SIGGRAPH* : 279-288

Robertson GG, Card SK, Mackinlay JD (1993) Information Visualization Using 3D Interactive Animation. *Communications of the ACM* 36(8):57-71.

Roese J, McCleary L (1979) Stereoscopic Computer Graphics for Simulation and Modeling. *Proceedings ACM SIGGRAPH*: 41-47.

Sachs E, Roberts A, Stoops D (1990) 3-Draw: A Tool for the Designing of 3D Shapes. *IEEE Computer Graphics and Applications* 11(6): 18-26.

Schmandt C (1983) Spatial Input/Display Correspondance in a Stereoscopic Computer Graphics Workstation. *Proceedings ACM SIGGRAPH*: 253-261.

Shaw C, Green M (1994) Two-Handed Polygonal Surface Design. *Proceedings ACM UIST*: 212-215.

Strauss PS, Carey R (1992), An Object-Oriented 3D Graphics Toolkit. *Proceedings SIGGRAPH*: 341-347.

Terzopoulos D, Platt JC, Barr AH, Fleischer K (1987) Elastically Deformable Models, *Proceedings ACM SIGGRAPH*: 205-214.

Turner R, Thalmann D (1993) The Elastic Surface Layer Model for Animated Character Construction. *Proceedings Computer Graphics International* .

Turner R (1995) Léman: A System for Consructing and Animating Layered Elastic Characters. Proceedings Computer Graphics International.

Ware C, Arthur K, Booth KS (1993) Fishtank Virtual Reality. *Proceedings INTERCHI*: 37-42.

Ware C, Osborne S (1990) Exploration and Virtual Camera Control in Virtual Three Dimensional Environments *Proceedings ACM Workshop on Interactive 3D Graphics*: 175-183.

Wloka M (1995) Lag in Multiprocessor VR. Presence: 4(1): 50-63.

Weiner D, Ganapathy SK (1989) A Synthetic Visual Environment with Hand Gesturing and Voice Input. *Proceedings ACM SIGCHI*: 235-240.

Zeleznik RC, Conner DB, Wlocka MM, Aliaga DG, Wang NT, Hubbard PM, Knepp B, Kaufman H, Hughes JF, van Dam A (1991), An Object-Oriented Framework for the Integration of Interactive Animation Techniques. *Proceedings ACM SIGGRAPH*: 105-112.

COLOR PLATE - THESE IMAGES WERE TAKEN USING THE POLHEMUS VERSION, WHILE THE PUBLISHED ONES WERE TAKEN USING THE LOGITECH VERSION OF THE SOFTWARE

Figure 4. Constructing a character

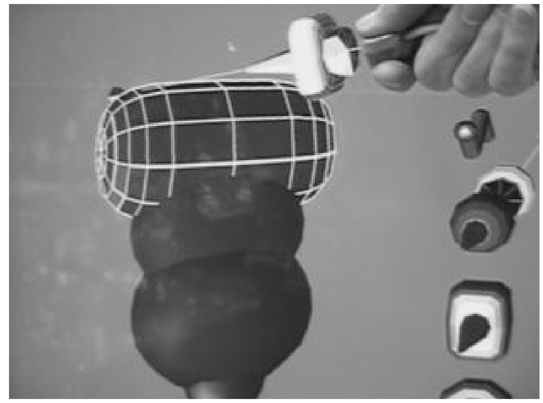

Figure 5. Skeleton creation

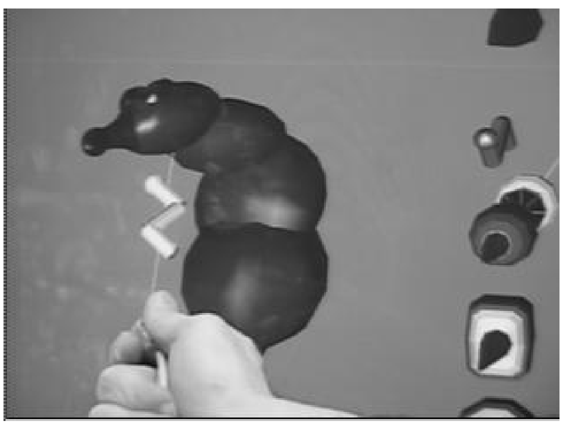

Figure 6. Inverse kinematics manipulation

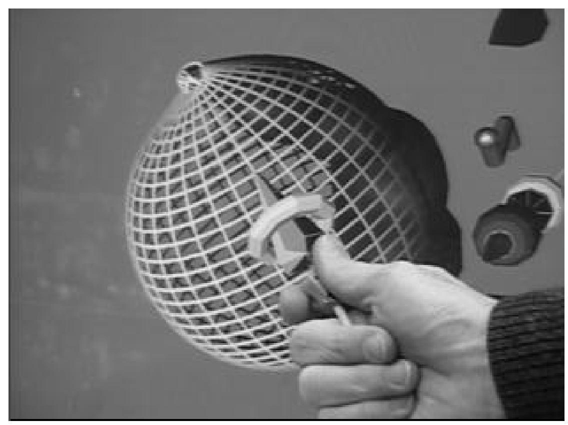

Figure 7. Skin creation

Figure 8. Sculpting the fat layer

Figure 9. The final character