Previous Next Contents

A PSEUDO-IMMERSIVE VIRTUAL ENVIRONMENT - A SHELL FOR RESEARCH

Jing-Jing Fang{{bio}}
Douglas E.R. Clark{{bio}}
John E.L. Simmons {{bio}}

Heriot-Watt University
Edinburgh, Scotland

Editor: Richard Blade

ABSTRACT

The authors built a pseudo-immersive three-dimensional computer generated virtual environment (VE) as a shell for research in detailed collision detection [4] [5] [6] [7]. The purpose of the virtual world construction described in this paper is to handle engineering applications such as the mechanical assembly and disassembly of complex systems. A virtual space ball is attached to the VE in order to control the motion of virtual objects in that environment. This software interface simulates a real three-dimensional space ball device. The contribution of the pseudo-immersive system removes the absolute necessity for expensive Virtual Reality equipment and software for research work on virtual reality.

1. Introduction

The system in its entirety consists of two parts: the shell, comprising the pseudo-immersive virtual environ-ment, and the kernel, which allows collision detection study. In this paper only the shell part of the research is presented in detail since it is the construction of this environment, primarily as a test base for virtual reality (VR) research, which forms the main thrust of the final system. As a result, it is now possible for research in aspects of immersive virtual reality to be carried out using a low performance basic workstation.

To date little work appears to have been done to address the construction of virtual environments. Zyda et al. [12], [13] developed a NPSNET system by using both the hardware of Ethernet with a T1 link between workstations and a software interactive technique to simulate a multi-user virtual environment. The aim of their research was to build up a large-scale, networked three-dimensional virtual environment. In the University of Bath (UK), Bayliss et al. built a virtual manufacturing system [2], [10] based on their own SVLIS geometric modeller, using the Inventor 3D graphics package around a Silicon Graphics workstation.

These self-built virtual worlds were developed around expensive equipment and extensive software, whereas the work described in this paper proposes a relatively inexpensive pseudo-immersive virtual environment implemented on a workstation. After development, the resulting software can then be ported to a fully immersive VR system.

2. The Pseudo-Immersive Environment

The authors constructed a three-dimensional virtual environment using C++ and the openwin library around a basic Sun microSPARC machine (4.6 MFLOPS, 59.1 MIPS). No immersive equipment such as headgear or sensory feedback devices were used in this process, the interactive interface device being a simple two-dimensional mouse. This pseudo-immersive virtual environment was created primarily as a background for research in virtual object assembly [4], [5], [6]. The main purpose of the environment was to simulate realistic motion of virtual objects in a virtual world. It is important, in particular, to be able to simulate situations in which collision of moving objects might be expected to occur. Typical examples arise naturally in mechanical assembly operations of engineering objects, and secondly in the simulation of environments incorporating gravity [7]. The principle advantages of this kind of system are that it eliminates the unpleasant side of effects of fully immersive systems, yet provides a platform for fundamental research in the area without being committed to any of the currently available costly commercial systems.

Figure 1 shows the interface layout arrangement of the virtual environment screen. Basically, this virtual environment consists of four display windows and a control menu schema. The isometric window and the perspective window display the output from the other two control window - virtual space ball control and orthogonal movement control. The user controls the motion of the free object from these two control windows. Three-dimensional rotation and translation are carried out by a virtual space-ball in the virtual space-ball window. Movement occurs by dragging the free object on screen and is carried out in the orthogonal control window. All of these software interfaces will be described in the following sections.

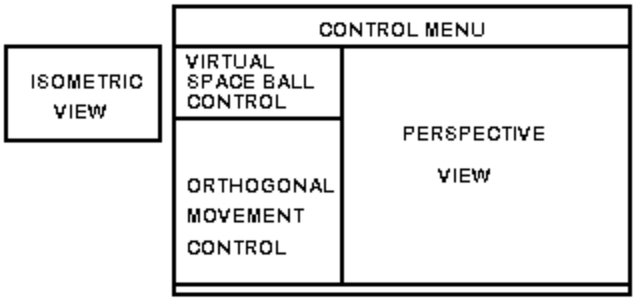


Figure 1: User interface layout of the virtual environment.

First, we briefly outline the on-line layout of the dynamic event handler of the pseudo-immersive virtual environment. The X-Window System (Openwin) in the Sun workstation offers a method for handling and specifying events on a canvas. The program waits for the following events - load, path control, quit, view control, motion control and virtual space ball control (see Figure 2). The first three events are located in the control menu, whereas the other three events are exhibited in the four windows as an input/output. Once the objects are loaded into this environment, the pre-processing procedure carries out a data base processor in order to manipulate the dynamic linked lists of the objects for later use in collision detection. The interface containing the six events is the main communication between the user and the system, whereas the sub-event containing free drawn, piecewise-linear, gravity, rotation and translation is under the event motion control. These five sub-events are described in the later sections, 3.1 and 3.2.

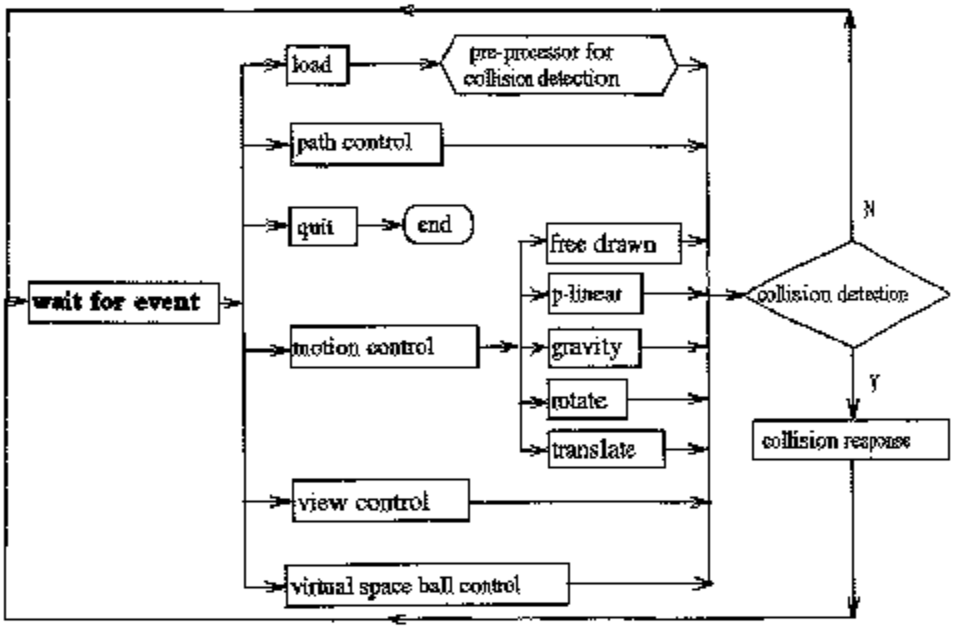


Figure 2: Dynamic Event Handler in Virtual Environment

The two objects in this environment are first created by the ACIS Test Harness [1]. They are loaded in by means of a load button provided in a loop. The fixed object is located either in the center or at the bottom of the orthogonal window, whereas the free object is located by means of a mouse by clicking in any position in the orthogonal window.

Once one of the collision detection methods is applied, the free object is then moved by one of the choices in the sub-loop. When collision occurs between the objects, collision response is signaled by a simulated sound of metal striking, and meantime, the color of the penetrating parts of the object pair is changed on the VDU.

3. Control Menu and Response

In this pseudo-immersive system there are three main functions - path control, space ball control, and view control (see Figure 3). They will be described here in the specific context of collision detection, but may clearly be used in a much wider range of VR ‘experiments'.

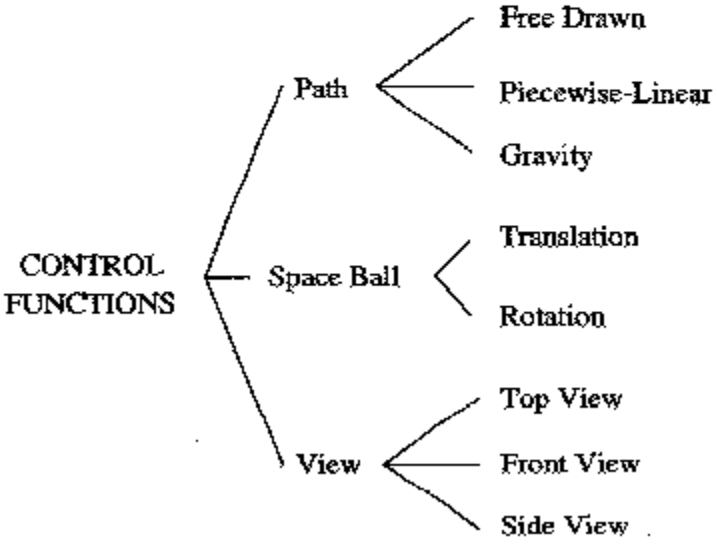


Figure 3: Control menu scheme.

3.1 PATH

The free object can move along any of the three coordinate-aligned planes by means of the function path-control. There are three sub-functions in path-control. First, the free drawn path facilitates snapping and dragging of the free object on screen in the orthogonal movement control window (see Figure 1). The second path-control method is drawn by piecewise-linear path and this path is determined by the user (see Figure 4). The third, gravity path-control, simulates a gravitational situation for an object falling along a vertical plane (see [7] for details).

3.2 CONTROLLER

The virtual space ball controller (see section 4) simulates the mechanics and action of a real three-dimensional space ball and controls a virtual object when it is freely moving in space. Rotation is carried out about an arbitrary axis passing through the center of the object. The motion of the free object in space is either translation or rotation and is controlled by the virtual space ball.

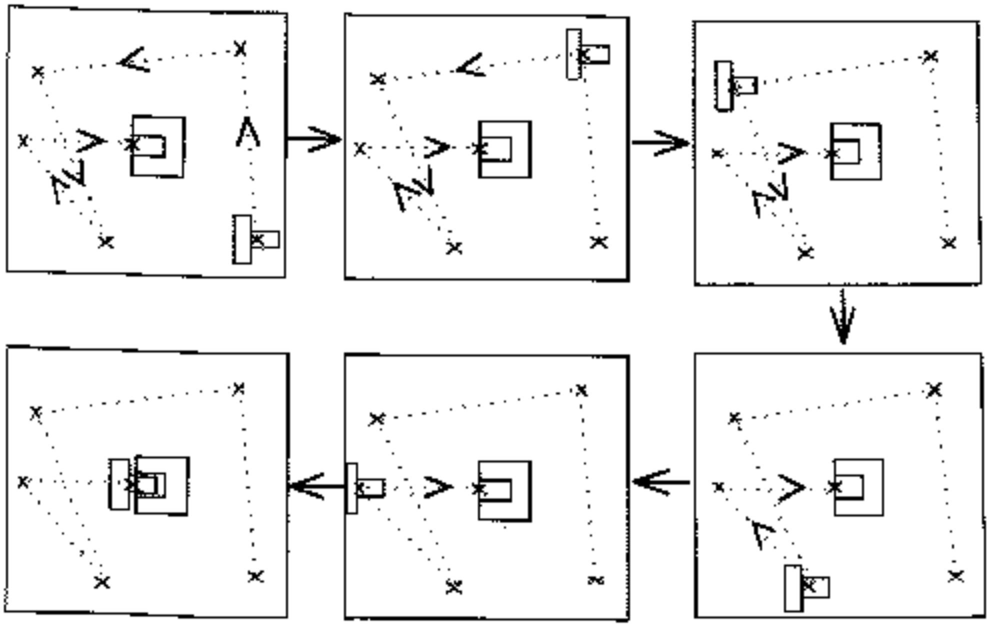


Figure 4: Piecewise-linear path.

3.3 VIEW

In the orthogonal view control window, one of the three possible orthogonal views - either front view, side view, or top view - is presented at any time. Meanwhile, when motion occurs in the free object, an isometric view and a perspective view window displays the relative position of the two objects simultaneously.

When interference occurs between the first two polygon planes in an assembly pair, there is a simulated sound of metal striking and the color of the first penetrating parts of the object pair is changed on the VDU. Depending on which of the various collision methodologies is currently in use, the penetrating parts will be drawn in one of five ways: the bounding box, the bounding sphere, the bounding box of the smaller box, the bounding sphere of the small box [5], or, the polygon pair [6].

4. Virtual Space Ball

The virtual environment makes heavy demands on the interactive graphics facilities, but enables the user to move virtual objects freely in the three-dimensional space of the environment using a commonly available two-dimensional mouse. The underlying idea of the virtual space ball originally comes from [3], [8], and [9], but has been simplified by the authors with the object of giving a better intuitive understanding, or "feel", for the environment.

A virtual space ball is attached to the virtual environment in order to control the motion of the virtual objects in space. The software interface that has been developed simulates a three-dimensional space ball device.

Rotation in space of the free object is achieved by rolling the surface of the virtual space ball (see Figure 5). There are two methodologies in the rotation algorithm to carry out arbitrary spatial rotations of a virtual object.

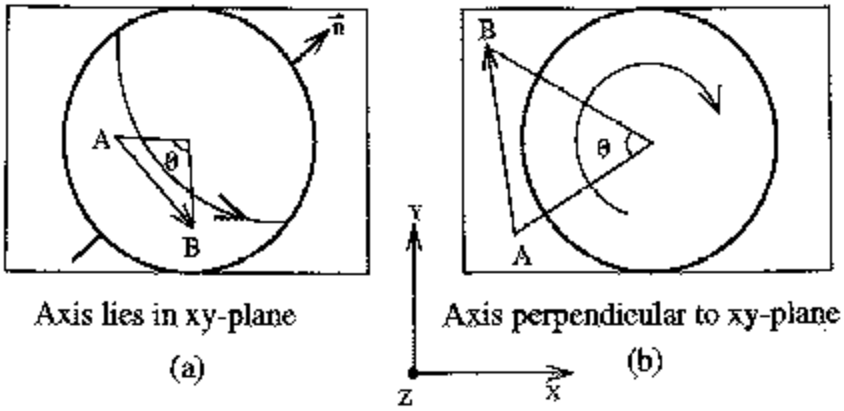


Figure 5: Rotation in space: virtual space ball instruction.

In order to rotate the free object through an angle *theta* about the axis specified by vector **n** in the plane of the screen, (Figure 5(a)), by using the two-dimensional mouse, the user first specifies, or locates, point A inside the circle, and then locates point B anywhere else in the window. If, on the other hand, point A is located outside the circle (Figure 5(b)), then the object rotates through an angle *theta* about a z-aligned axis passing through the object center. By way of illustration, Figure 6 shows a series of typical situations.

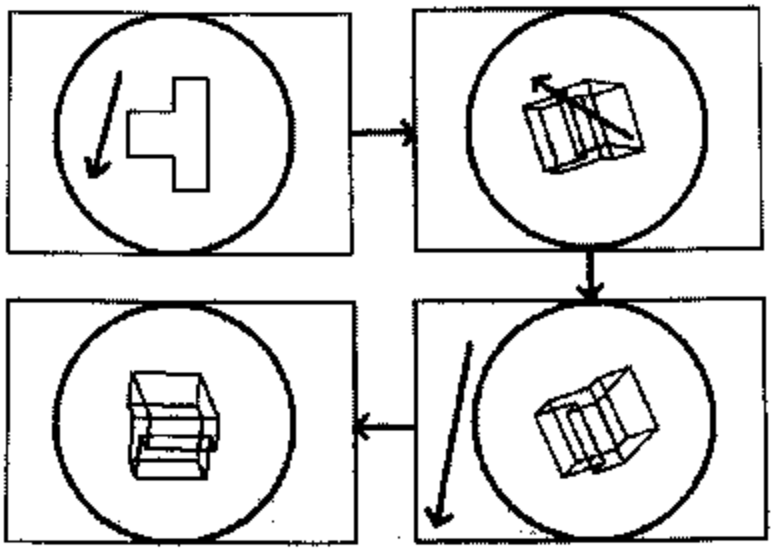


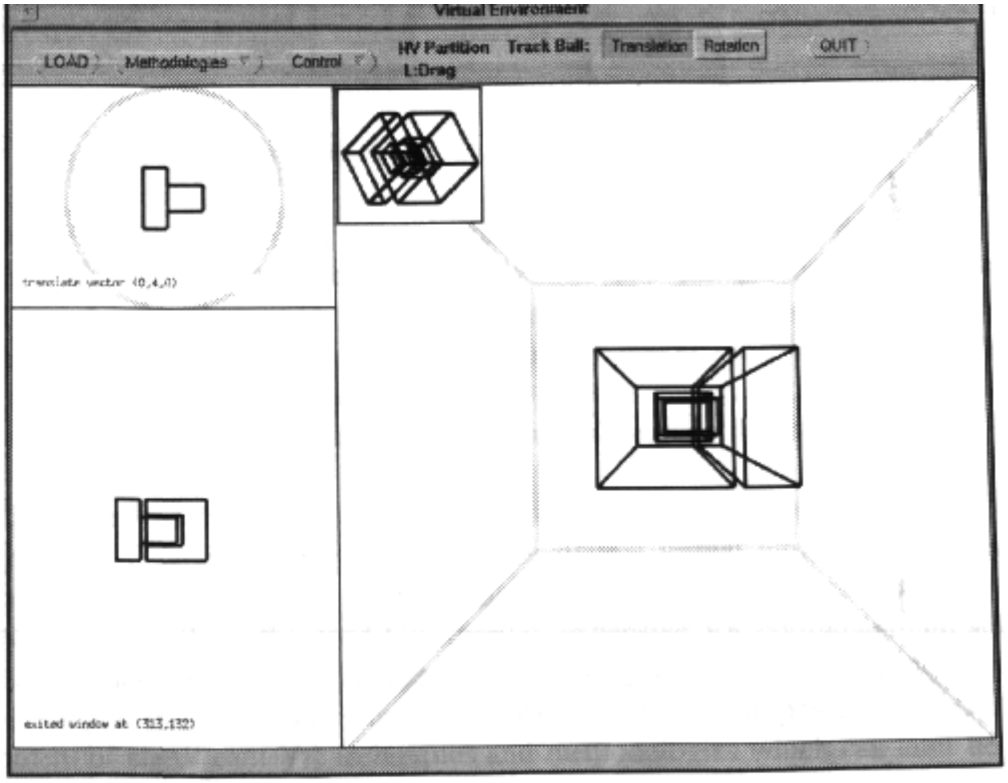
Figure 6: Virtual space ball in rotation status.

Summarizing, rotation about either an arbitrary axis in the xy-plane or about a z-aligned axis is determined by the location of the point A. Rotating *theta* degrees about the normal to the screen plane is equivalent to rolling the object along z-aligned axis. Therefore, in order to achieve rotation about an arbitrary axis in the xy-plane, one should locate point A inside the circle, while for rotation about an axis parallel to the z-axis one should locate point A outside the circle (the location of point B is irrelevant in this latter case).

Similarly, when the setting is in translation status, the vector **AB** determines the translation transformation in the xy-plane if and only if A is located inside the space ball circle, and point B may be located anywhere in the window. A similar movement along an axis parallel to z-axis is determined by the location of A outside the circle. In this case, the movement of the object (in either direction) along the z-aligned axis is determined by the difference between the real y-coordinates of A and B on the display screen.

5. Auxiliary Views

In order to exhibit a realistic three-dimensional image on a two-dimensional display screen, and to give the user a better intuitive feel for the situation, two auxiliary views - isometric view and perspective view - are added into the virtual environment. The view controller is set to one of the three orthogonal views, top view, front view or side view. This provides the user with the ability to move the free object accurately along any of the three coordinate-aligned planes.



translate vector (0,4,0)

exited window at (313,132)

Figure 7: A protruding component partly inserted into a component with hole.

6. Contributions

Figures 7, 8, 9, and 10 demonstrate typical situations in the virtual environment. The kernel work on collision detection methodologies is implemented on the environment shell. Figure 7 shows the situation of an object with a protrusion (such as a T-Block) partly inserted into an object containing a depression or a hole (such as a keyhole). A similar insertion situation, this time for a thin shaft moving inside a relatively spacious hollow object, is shown in Figure 8. This situation simulates a stirring spoon inside a cup or, more generally, a probe situation. In Figure 9 we show another typical collision detection situation, but this time involving objects with curved faces. Finally, in a virtual gravitational environment, a falling object first rolls then falls onto a plane surface (see Figure 10).

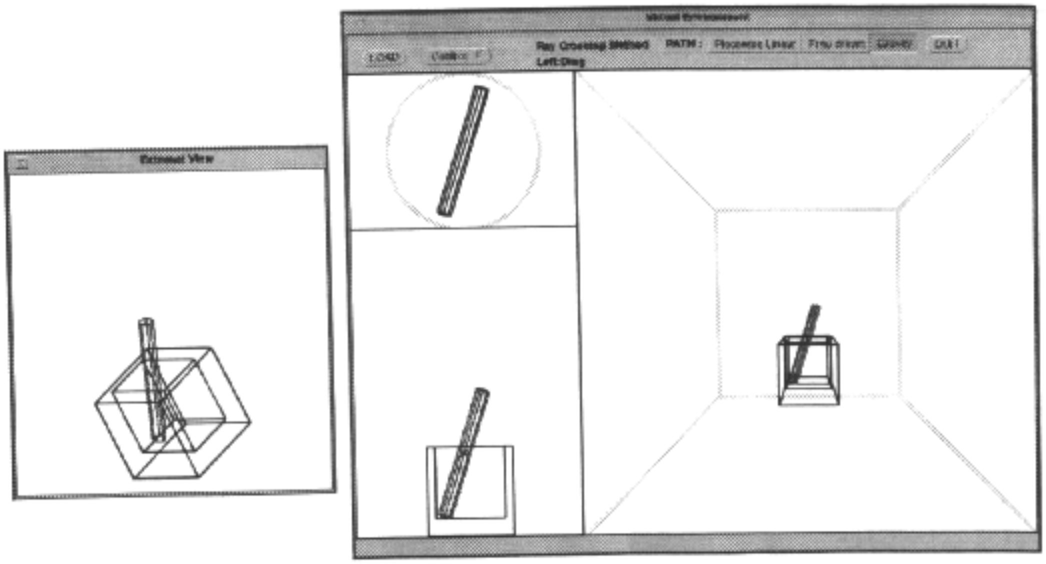


Figure 8: A thin shaft inserted into a hollow object.

It is worth pointing out that the decision to incorporate a virtual space ball into the shell was motivated by both cost considerations and the prospect of large scale applications. As virtual reality becomes more established in mass education, this form of input control, as opposed to a costly real 6D mouse, makes sense on economic grounds [11]. Furthermore, as with the pseudo-immersive environment itself, this once again frees the system from any particular commercially available input equipment.

In this paper we have described the construction and implementation of a pseudo-immersive virtual environment. The main aim of this virtual world construction is to handle engineering applications such as the mechanical assembly and disassembly of complex systems without the need for complex expensive VR equipment and software. The user interface for this environment is a virtual space ball. An important feature of this environment is that it does not involve expensive VR equipment, but allows the development of significant VR techniques and methodologies which can later be ported to a fully immersive VR environment.

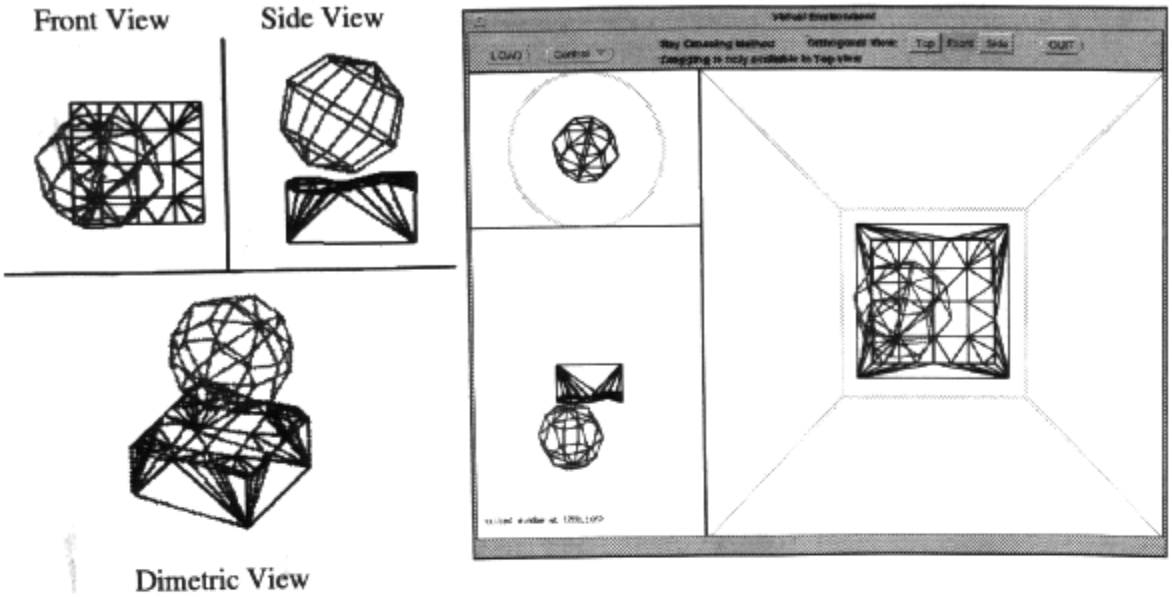


Figure 9: A ball on a double curved surface.

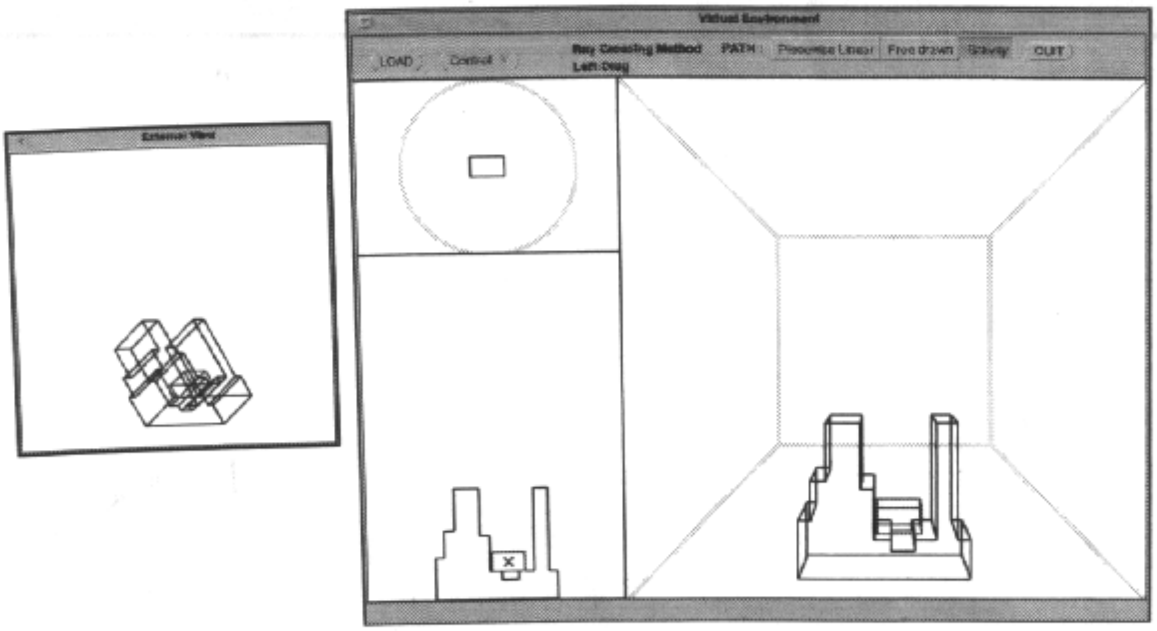


Figure 10: A falling object in a virtual gravitational environment.

ACKNOWLEDGMENTS

Jing-Jing Fang gratefully acknowledges the support of a scholarship provided by the National Science Council of Taiwan, R.O.C.

Beginning Previous Next Contents

REFERENCES

[1] Spatial Technology Inc. *ACIS Geometric Modeler API Reference.* 2425 55th Street, Building A, Boulder, Colorado 80301. {{return}}

[2] Bayliss, G.M., Bowyer, A., Taylor, R.I., Willis, P.J. *Virtual Manufacturing.* CSG 94 - Set-theoretic Solid Modelling Techniques and Applications, Winchester, UK, April 13-15, 1994. {{return}}

[3] Chen, M., Mountford, S.J., Sellen, A. *A Study in Interactive 3-D Rotation Using 2-D Control Devices.* ACM Computer Graphics, Vol. 22, No. 4, August 1988. {{return}}

[4] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *Collision Detection Methodologies for Rigid Body Assembly in a Virtual Environment.* Journal of Virtual Reality Society, Vol. 1, No. 1, pp 41-48, June, 1995. {{return}}

[4] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *Collision Detection Methodologies for Rigid Body Assembly in a Virtual Environment.* Journal of Virtual Reality Society, Vol. 1, No. 1, pp 41-48, June, 1995. {{return}}

[5] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *Dynamic Collision Detection in Virtual Worlds Using HV Partition.* Proceedings of ASME Computers in Engineering Conference and the Engineering Database Symposium, pp. 905-910, Boston, September, 1995. {{return}}

[5] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *Dynamic Collision Detection in Virtual Worlds Using HV Partition.* Proceedings of ASME Computers in Engineering Conference and the Engineering Database Symposium, pp. 905-910, Boston, September, 1995. {{return}}

[5] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *Dynamic Collision Detection in Virtual Worlds Using HV Partition.* Proceedings of ASME Computers in Engineering Conference and the Engineering Database Symposium, pp. 905-910, Boston, September, 1995. {{return}}

[6] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *An Accurate Collision Detection Method for Virtual Assembly and Locking.* Proceedings of Framework for Immersive Virtual Environments - FIVE Conference, London, U.K., pp. 151-162, 18-19 December 1995. {{return}}

[6] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *An Accurate Collision Detection Method for Virtual Assembly and Locking.* Proceedings of Framework for Immersive Virtual Environments - FIVE Conference, London, U.K., pp. 151-162, 18-19 December 1995. {{return}}

[6] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *An Accurate Collision Detection Method for Virtual Assembly and Locking.* Proceedings of Framework for Immersive Virtual Environments - FIVE Conference, London, U.K., pp. 151-162, 18-19 December 1995. {{return}}

[7] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *Collision Detection in a Virtual Gravitational Environment.* Proceedings of 1996 SCS Simulation MultiConference In Synthetic Environments'96, Vol. 28, No. 4, pp. 5-10, Louisiana, April 8-11, 1996. {{return}}

[7] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *Collision Detection in a Virtual Gravitational Environment.* Proceedings of 1996 SCS Simulation MultiConference In Synthetic Environments'96, Vol. 28, No. 4, pp. 5-10, Louisiana, April 8-11, 1996. {{return}}

[7] Fang, J.J., Clark, D.E.R., Simmons, J.E.L. *Collision Detection in a Virtual Gravitational Environment.* Proceedings of 1996 SCS Simulation MultiConference In Synthetic Environments'96, Vol. 28, No. 4, pp. 5-10, Louisiana, April 8-11, 1996. {{return}}

[8] Hanson, A.J. *The Rolling Ball.* Graphics Gems III, Academic Press Limited, 1992. {{return}}

[9] Shoemake, K. *Arcball Rotation Control.* Graphics Gems IV, Academic Press Limited, 1994. {{return}}

[10] Willis, P., Bowyer, A., Taylor, R., Bayliss, G. *Virtual Manufacturing.* International Workshop on Graphics and Robotics, Germany, April 19-22, 1993. {{return}}

[11] Vince, J. *Virtual Reality Systems.* Addison Wesley, 1995. {{return}}

[12] Zyda, M.J., Lombardo, C. and Pratt, D.R. *Hypermedia and Networking in the Development of Large-Scale Virtual Environments.* Proceedings of the International Conference on Artificial Reality and Tele-existence, Tokyo, Japan, 6-8 July 1993. {{return}}

[13] Zyda, M.J., Pratt, D.R., Falby, J.S., Lombardo, C. and Killeher, K.M. *The Software Required for the Computer Generation of Virtual Environments.* Presence, Vol. 2, No. 2, pp. 130-140, 1993. {{return}}

BIOGRAPHIES

Dr. Jing-Jing Fang was born in Ja-Yi, Taiwan. In 1981, she received B.Sc. degrees in both the Departments of Applied Mathematic and in the department of Computer Science in Feng Chai University, Taiwan. In 1983 she received her M.Sc. degree in National Cheng Kung University, Taiwan. She was appointed as a lecturer in Accounting and Statistics at Tainan Economic College in 1983 and in Mechanical Engineering at National Cheng Kung University in 1987. From 1993 to 1996, she was sponsored by National Science Council in Taiwan to get her PhD degree at Heriot-Watt University, Edinburgh, UK. Now, she is an associate professor in the Department of Mechanical Engineering in National Cheng Kung University, Taiwan.

Miss Fang is a member of the Virtual Reality Society, the Society for Computer Simulation, and the Society for Chinese Engineers. Her research interests include computer graphics, geometry/solid modelling, virtual reality, and numerical analysis.

Contact information:
Dr. Jing-Jing Fang
Department of Mechanical Engineering
National Cheng Kung University
Tainan, TAIWAN
Tel: 886-6-2757575 ext 62126 (O)
Fax: 886-6-2352973
Email: fangjj@mail.ncku.edu.tw {{Return}}

Douglas E.R. Clark received his B.Sc. in Mathematics from London University in 1967 and his PhD from the University of Durham in 1970. In 1970 he was awarded a Royal Society European Fellowship to continue his research in the Geometry of Manifolds in the Mathematics Institute in the University of Amsterdam, Netherlands. He was appointed as a lecturer in Pure Mathematics at Heriot-Watt University, in 1971. Doctor Clark's research interests are in Differential Geometry, Low Dimensional Topology, and Computational Geometry.

Contact information:
Dr. D.E.R. Clark
Dept. of Mathematics
Heriot-Watt University
Riccarton, Edinburgh EH14 4AS, UK
Phone: 44-131-5111 Ext 3234
Email: derc@ma.hw.ac.uk {{Return}}

John E. L. Simmons studied engineering at Birmingham and Cambridge

Universities and worked for two international manufacturing companies before joining Durham University in 1984. In January 1991 he moved to Heriot-Watt University, where he is now Head of the Department of Mechanical and Chemical Engineering. Professor Simmons' research interests are primarily in the fields of engineering design and manufacture.

Contact information:
Professor J.E.L. Simmons, Head of Department
Dept. of Mechanical & Chemical Engineering
Heriot-Watt University
Riccarton, Edinburgh EH14 4AS, UK
Phone: 44-131-5111 Ext 3132
Email: mecjs1@pp.hw.ac.uk {{Return}}

Beginning Previous Next Contents