Proceedings of International Symposium On the Future I&C for NPP, pp. 321-326, 2002

Development of A New Interface for Constructing Interactive Virtual Environments

Hirotake Ishii, Keisuke Endou, Hidekazu Yoshikawa
Graduate School of Energy Science, Kyoto University, Japan
Phone: +81-774-38-4403, Fax: +81-774-38-4406, E-mail: hirotake@uji.energy.kyoto-u.ac.jp

`

Abstract

The goal of this study is to develop new software for constructing interactive virtual environments of machine-maintenance training using virtual reality technology. The problem of existent virtual reality software is that special knowledge is required in order to construct interactive virtual environments where a user can assemble and disassemble virtual apparatus. Therefore this paper presents a new interface through which a user intuitively constructs interactive virtual environments by attaching ActCube, which is a rectangular parallelepiped with a peculiar function, to 3 dimensional models as like building blocks.

1. Introduction

Virtual reality technology can be applied to various kinds of industrial applications such as simulations, entertainment and machine-maintenance training[1]. To build a virtual reality system, both of the hardware and the software are necessary to be developed. Nowadays, the hardware, which can be used to build a virtual reality system, has been rapidly progressed and high performance hardware can be used at very low cost. On the other hand, many kinds of support software for constructing virtual environments have been developed in recent years[2][3]. These support software have been version-upped frequently and the number of the available functions has been increased. The result is that the Graphical User Interface of the support software becomes very complex and novice users cannot use them without the introduction by the experts in the computer field.

In this study, we propose a new interface for constructing virtual environments more intuitively, especially for constructing machine maintenance training environments. This interface allows users to construct interactive virtual environments by using a block-metaphor. The properties of virtual objects are represented as 3 dimensional blocks in the virtual environment and the virtual objects can be constructed by combining plural blocks. By using a block-metaphor, we expect that it becomes possible for novice users to construct complicated virtual environment easily.

In this paper, the methods for simulating virtual environments are introduced first, and the information required to construct virtual environments is explained. Then the configuration of ActCube, which is a rectangular parallelepiped with a peculiar function to restrict the movement of virtual objects, and the designed interface for constructing virtual environment by using ActCube are described.

2. Information required to simulate virtual environments

2.1. Methods for simulating virtual environments

Constructing an interactive virtual environment where human can perform just like in a real world corresponds to defining how to simulate the movement of virtual objects according to the user's gesture. There are two methods to simulate the movement of virtual objects. One is the physically based simulation[4][5] and the other is rule-based simulation[6]. The physically based simulation calculates the movement of virtual objects by solving equations of motion and calculating the force exchanged between virtual objects using the information about the 3-dimensional shape, the modulus of elasticity and so on. The physically based simulation can simulate realistic movements of virtual objects and the workload for constructing virtual environment is very small. But the calculation load to simulate physically based virtual environment is very large and even if high performance computers are available, it is difficult to simulate virtual environment in real time, especially in the case that the shape of the virtual objects is very complex[7]. On the other hand, the rule-based simulation calculates the movement of virtual objects according to the simple rules, which are defined in advance for each state of virtual objects. For example, in the case that a nut is in the state 'fixed by a volt', the nut can rotate only around the volt. Because the calculation process of the rule-based simulation is very simple, the calculation load is very small, but the movement of virtual objects is not so real.

In this way, in order to construct and simulate virtual environments effectively, it is desirable to use the physically based simulation and the rule-based simulation together. But to simulate virtual environments by the rule-based simulation, a lot of rules are necessary to be defined one by one. If the size of virtual environment becomes large, the workload to construct the virtual environment also becomes very large. The reason the interface of the support software for constructing virtual environments becomes very complex is that the interface to define the rules for the rule-based simulation becomes very complex.

2.2. Information required to simulate rule based virtual environments

In this study, the authors aimed at developing a new support system with which users can construct interactive virtual environment easily, especially for machine-maintenance training environment.

In the machine maintenance-training environment, a lot of virtual objects such as nuts, volts, spanners interact with each other. To simulate such a virtual environment, it is necessary to make it possible to simulate many kinds of interactions between virtual objects as shown in Fig. 1. As mentioned in section 2.1, there are two methods to simulate interactive virtual environments, physical-based and rule-based. It may seem that both of the simulation methods can be applied to simulate machine- maintenance training environment, but from the view point of calculation load, the physically based simulation method can not be applied to simulate all interactions between virtual objects because the real-time response is absolutely required to make it possible for the user to feel the immersed experience. So, it is a primary condition to adopt the rule-based simulation method to simulate machine maintenance-training environment.

To construct a drawer in a virtual environment, it is necessary to simulate the restriction of the drawer's movement to parallel translation. And to construct a volt and a nut, it is necessary to simulate the restriction of the nut's movement to rotational movement around the volt.

Moreover, to simulate the machine maintenance-training environment, the information how range virtual objects can move is also necessary to be defined. For example, to simulate the movement of a door in a virtual environment, the information how range the door can be opened is necessary to avoid the invasion of the door into the wall.

In this way, to construct the virtual environment to be simulated with the rule-based simulation method, both of the information about what kind of interaction should be simulated and how range the virtual object moves are necessary to be defined.

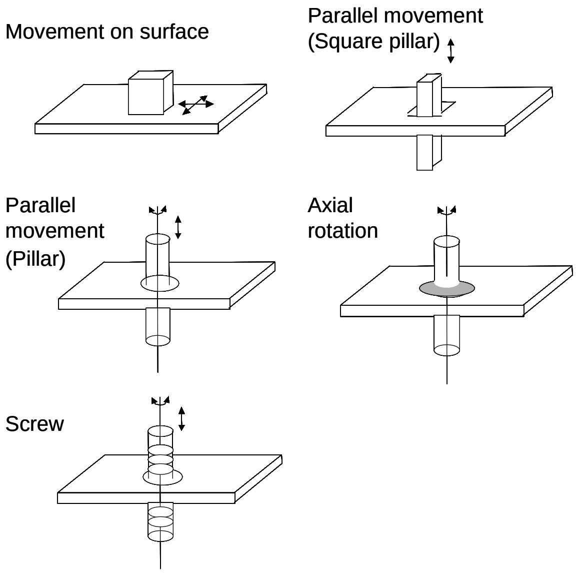

Figure 1: Interaction examples between virtual objects.

2.3. Design principle of support system

As mentioned in section 2.2, two kinds of information are necessary to construct interactive virtual environment. In this study, to make it possible to define these information intuitively, a new interface using a block metaphor is introduced.

The user of the new support software is supposed to be immersed into virtual environment when constructing interactive virtual environment. The user manipulates block-shaped virtual objects named ActCube which represents the properties of virtual objects. The user can construct virtual objects by attaching plural ActCubes to 3 dimensional models as like building blocks. The kind of the ActCube corresponds to the kind of interaction between virtual objects and the size of the ActCube corresponds to how range the virtual objects can move. The user can change the properties of virtual objects by changing the kind and the size of ActCubes.

In this way, by introducing the block-metaphor interface it is expected that the user can construct interactive virtual environments intuitively even if the user has no special knowledge such as computer science and computer programming.

In designing the support software with block-metaphor interface, some conditions are supposed as follows,

1. The virtual objects located in virtual environment are limited to rigid bodies which shape isn't transformed throughout the simulation.
2. 3 dimensional models of virtual objects are prepared in advance by using commercial applications such as 3D CAD.

3. The user of the support software uses their both hands to construct virtual environment and experience the constructed virtual environment.
4. The hardware interface of the support software measures the location and the orientation of the user's both hands and whether the hands are open or closed.

3. System Design

Figure 2 shows the interface design of the support software for constructing virtual environment. The interface consists of one workspace where virtual objects are constructed, 3 dimensional models of virtual objects, ActCubes which represents the properties of virtual objects and two hand-cursors (right hand and left hand) which manipulate 3 dimensional models and ActCubes.

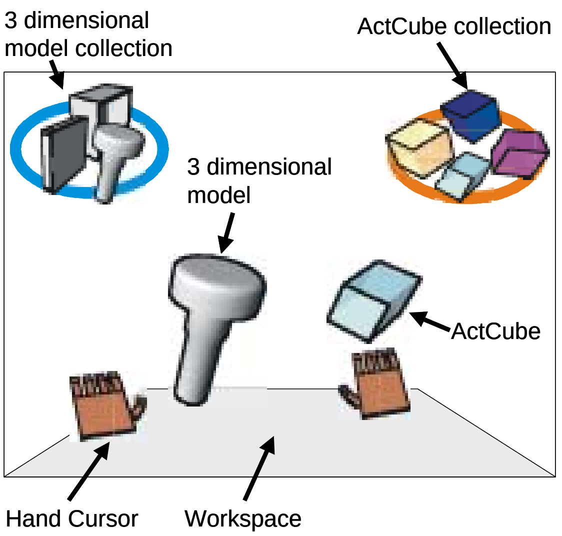

Figure 2: Interface design for constructing virtual environment.

3.1. Workspace

The virtual environment for constructing virtual objects is called "Workspace". The user of the support software is immersed into the Workspace and all the operation to construct virtual environments can be conducted there and it is unnecessary for the user to go and back from 2 dimensional Graphical User Interface to the Workspace. It is very important because if the users need to change their interface it will prevent their creation activity. The size of the Workspace can be changed according to the size of the virtual environment. The collection of 3 dimensional shapes and ActCubes are also located in the Workspace.

3.2. Hand cursor

Hand cursor is a virtual object which shape is like a human hand and linked to the user's real hands. There are two Hand cursors corresponding to right hand and left hand of the user. The Hand cursor moves in the Workspace freely according to the movement of the user's real hand. The user can manipulates 3 dimensional models and ActCubes through these Hand cursors.

3.3. Three-dimensional model

3 dimensional model is one of the skeletons which is the basis of virtual objects. At the beginning, 3 dimensional model is only a surface of a virtual object and has no ability to interact with the other 3 dimensional model. But when the ActCube is attached, the 3 dimensional model becomes an interactive virtual object and can perform like a real object.

3.4. ActCube

ActCube is a virtual block which has the information about the kind of the interaction between virtual objects and the range of the virtual object's movement used for simulating virtual environment based on Rule-based simulation. The ActCube is also the boundary box used to detect the collision between virtual objects.

In a Workspace, ActCube is represented as a rectangular parallelepiped as shown in Figure 3. ActCube can be attached to 3 dimensional models and defines the kind of the interaction and the range of the movement of the attached virtual object.

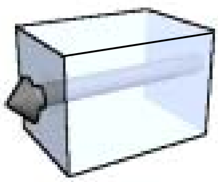

Figure 3: Appearance of ActCube.
(Arrow shows the main coordinate of ActCube)

There are several kinds of ActCubes according to the kind of the interaction and for each kind of ActCube there are two types of ActCube, parent and child. The interaction between virtual objects are simulated only when the ActCubes attached to the virtual objects are the parent ActCube and the child ActCube of the same kind and they are combined in the virtual environment. That is, the ActCube plays a kind of a joint which connects 2 virtual objects like a hinge and restricts the movement of attached virtual objects.

At the present stage, 5 kinds of ActCubes are designed as follows,

1. **Grasp ActCube**: Grasp ActCube is used for constructing a virtual object which can grasp other virtual objects. (Example: Wrench which can rotate some kinds of nuts)
2. **Parallel Movement ActCube**: Parallel Movement ActCube is used for constructing a virtual object which movement is restricted to the parallel movement towards the other virtual object. (Example: Drawer which can be drawn by a hand)
3. **Axial Rotation ActCube**: Axial Rotation ActCube is used for constructing a virtual object which movement is restricted to the rotational movement around the other virtual object. (Example: Nut fit in a volt)
4. **Hinge ActCube**: The movement of the attached virtual object is the same as the case of Axial Rotation ActCube except the rotation angle is limited within 360 degrees. (Example: Door)
5. **Screw ActCube**: Screw ActCube restricts the movement of a virtual object to the spiral around the other virtual object. (Example: Screw)

The number of the ActCube will be increased according to the progress of the system development.

4. Simulation Procedure of Virtual Environment

The virtual environment constructed by using the support software is simulated in the following steps,

Step 1. The user tries to move one of virtual objects located in the virtual environment. Go to Step2.

Step 2. Investigate whether an ActCube combined with another ActCube exists among the ActCubes attached to the moved virtual object. If any ActCube combined with another ActCube exists, the information about the location and the orientation of the user's hands is sent to the ActCubes and go to Step3. Else go to Step4.

Step 3. New location and orientation of the moved virtual object is calculated in consideration of the restriction by the ActCubes and the results are reflected to virtual environment. Go to Step5.

Step 4. Calculate new location and orientation of the moved virtual object by using the information about the location and the orientation of the user's hand. Go to Step5.

Step 5. Calculate new location and orientation of all virtual objects located in virtual environment based on Physically based simulation method and Rule-based simulation method. Go to Step6.

Step 6. If any collision is occurred between the ActCubes as the result of Step4 and Step5, go to Step7. Else return to Step1.

Step 7. Investigate 2 conditions such as (1) whether the kind of the collided ActCubes is the same, (2) whether the types of the collided ActCubes are parent and child. If all the conditions are met, 2 ActCubes are combined and return to Step1. Else return to Step1.

The important point of the above simulation method is that the translation of the information about virtual objects is limited to the way through the ActCubes. This means that there is a kind of information pipes between virtual objects. Therefore it is unnecessary for the virtual objects to access the other virtual objects directly to get the information about the location and the orientation of the other virtual objects. This advantage makes the construction workload very small, because the user need not to consider about all virtual objects constructed in advance and only need to consider about the existent ActCubes.

5. Construction of Virtual Environment

In order to construct virtual environments by using the support system, the parts of virtual objects are needed to be constructed first. And then virtual objects can be constructed by assembling the constructed parts. Figure 4 and Figure 5 show the construction process of virtual object's part and how to attach ActCube to 3 dimensional models respectively. The details of the virtual environment construction are as follows,

(1) Read one 3 dimensional model into the Workspace.
(2) Read one ActCube into the Workspace.
(3) Attach the ActCube to the 3 dimensional model to define the kind of the interaction.
(4) Resize the ActCube to define the range virtual object can move.
(5) Repeat (2) – (4) until all ActCubes are attached to the 3 dimensional model.
(6) Save the 3 dimensional model with ActCubes as a virtual object's part.
(7) Repeat (1) – (6) until all parts are constructed necessary for constructing a virtual object.
(8) Read all parts saved at (7) into the Workspace.
(9) Construct virtual objects by assembling virtual object's parts.
(10) Save the constructed virtual object.
(11) Repeat (1)-(10) until all virtual objects are constructed necessary for constructing virtual environment.

(12) Locate the virtual objects into virtual environment to define the initial location of virtual objects.

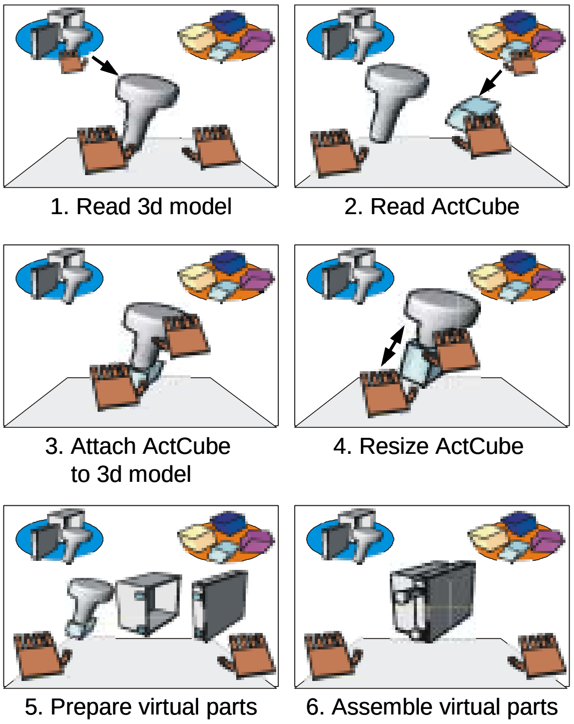

Figure 4: Construction procedure of virtual objects.

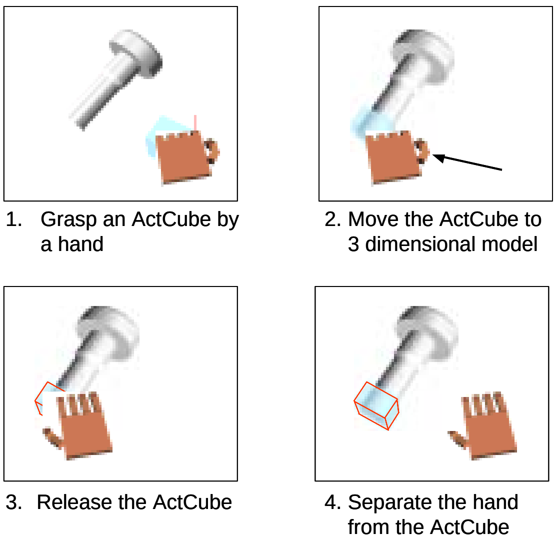

Figure 5: Work of attaching an ActCube to 3 dimensional model.

assembling virtual apparatus, virtual object's parts are located in the virtual environment instead of locating virtual objects.

6. System Integration

Figure 6 shows the hardware configuration of the support system. Because the support system is now under construction, a part of the Fig. 6 has not been developed yet. The support system consists of Simulation Subsystem, Measurement Subsystem and Display Subsystem. The Simulation Subsystem consists of one Main Computer (Operation System: Red Had Linux 7.2, Machine: Custom Made, Pentium III 1.26GHz Dual) and the Display Subsystem consists of one Graphics Workstation (Operation System: IRIX 6.2, Machine: SGI OCTANE 250MHz). The software to simulate the virtual environment and the construction workspace is executed on the Main Computer, and the software for rendering the virtual environment is executed on the Graphics Workstation.

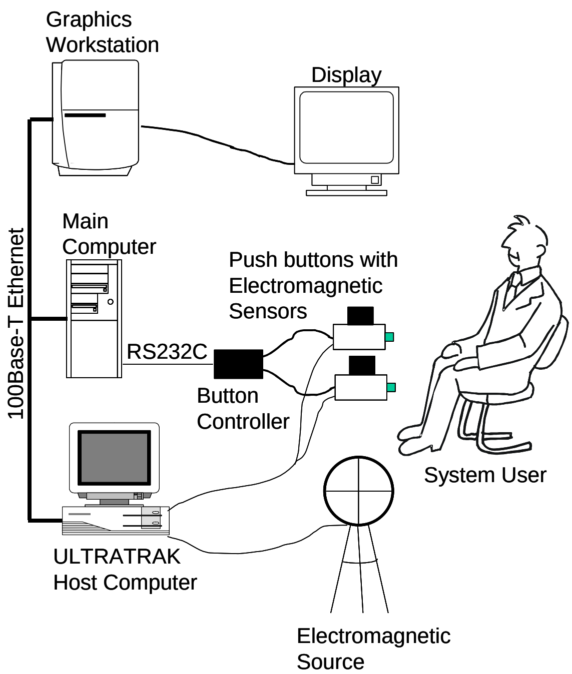

Figure 6: System Hardware Configuration.

The Measurement Subsystem consists of Ultratrak system and two hand-made push buttons. By using the Ultratrak system the movement of the user's both hands is measured and whether a hand is open or close is measured by the push buttons. From the Ultratrak system to the Main Computer, the information about the 3 dimensional location and the orientation of the user's both hands is transferred via TCP/IP socket communication. The push buttons are connected to the Main Computer via serial communication.

The software to simulate virtual environments is now under construction by using C++ language and the software to render the virtual environment is developed by using OpenGL Performer library.

7. Conclusion

In this study, a new interface for constructing interactive virtual environments is proposed. The proposed interface is based on the idea that in order to make it possible for the user to construct virtual environments intuitively, a block metaphor is most adequate and all the operations to construct virtual environments should be conducted not through conventional Graphical User Interface but in the virtual environment.

At the present stage, the system design and the hardware system development have been completed, and the system software is now under development. After the software development, some kinds of interactive virtual environments will be developed by using the developed support software. And some verification experiments will be conducted to compare the workload to construct virtual environments with the other construction support software.

8. References

[1] T. Tezuka, M. Yamamoto, H. Ishii and H. Yoshikawa, “A Study on VR-based Plant-Maintenance Training System”, *Journal of Tsinghua University (Science and Technology)*, Vol.37, No.S3, pp.53-64, 1997.

[2] H. Shimoda, H. Ishii, M. Yamamoto and H. Yoshikawa, “Application Studies of Virtual Reality to Training for Plant Operation and Maintenance”, *The Transactions of Human Interface Society*, Vol. 1, No. 4, pp. 35-42, 1999. (in Japanese)

[3] H. Ishii, T. Tezuka and H. Yoshikawa, “A Study on Design Support System for Constructing Machine-Maintenance Training Environment Based on Virtual Reality Technology”, *Proceedings of 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2635-2640, 1998.

[4] M. Harada, A. Witkin and D. Baraff, “Interactive physically-based manipulation of discrete/continuous models”, *Computer Graphics Proceedings, Annual Conference Series*, pp. 199-208, 1995.

[5] D. Baraff, “Fast contact force computation for non-penetrating rigid bodies”, *Computer Graphics Proceedings, Annual Conference Series*, pp. 23-34, 1994.

[6] H. Ishii, K. Kashiwa, T. Tezuka and H. Yoshikawa, “Development of Machine-Maintenance Training System using Petri Net and Virtual Environment”, *Proceedings of Cognitive Systems Engineering in Process Control*, pp. 138-143, 1996.

[7] B. Mirtich, “Timewarp rigid body simulation”, *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 193-200, 2000.