Proceedings of the 1993 JSME
International Conference on Advanced Mechatronics
Tokyo, JAPAN - Aug. 1993

Virtual Reality System with Coherent Kinesthetic and Visual Sensation of Presence

Yasuyuki Yanagida and Susumu Tachi

RCAST, The University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo 153 Japan
Phone: +81-3-3481-4468 Fax: +81-3-3481-4580
E-mail: yyanagi@tansei.cc.u-tokyo.ac.jp
tachi@tansei.cc.u-tokyo.ac.jp

ABSTRACT

To realize highly realistic sensation of presence in virtual environment, it is necessary to implement virtual reality system fitting human sensory mechanism, which integrates multiple kinds of sensory information. We aim at "natural" virtual reality system including multiple input/output channels for sensory information signals. As the first step, we developed a system using human motion measurement and visual information display, which provides coherent kinesthetic and visual sensation of presence. Using the class of C++ language, we developed a method for describing virtual environment, which is flexible, easy to construct the environment, and efficient for image generation.

INTRODUCTION

To construct natural virtual environment, it is important to satisfy the following conditions:(1)

1) the virtual environment should be displayed to the operator constructing natural three dimensional space,
2) the operator should be able to move around in the virtual environment and the interaction between the operator and the environment should be took place in real time,
3) the operator himself/herself should feel as if he/she *is* inside the virtual environment (self-projection).

Furthermore, it will be effective for higher reality to display multiple kinds of sensory information such as vision, auditory sensation, force/tactile sensation, and so on. When displaying these kinds of information, it is important that they should be displayed coherently so that they are recomposed together in operator's sense.

Coherent kinesthetic and visual sensation

We define the term *coherency* to be the state that the operator is self-projected in the virtual environment, such that kinesthetic sensation and displayed sensory information are integrated consistently in the human sensation. To realize this, we propose the concept of 'virtual human', who exists in the virtual environment and acts and feels as human in the real world would do.(2)

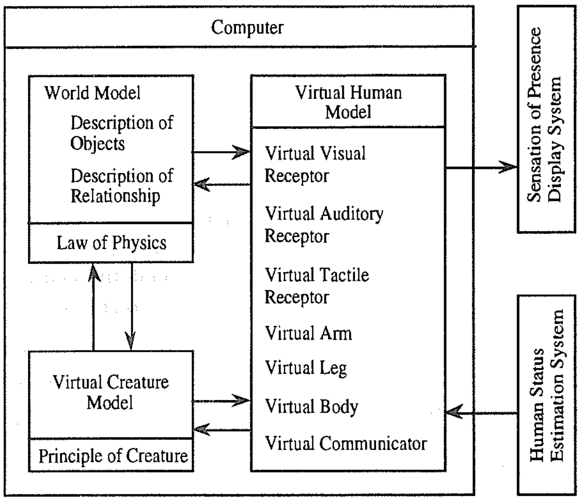


Fig. 1 Concept of the system

This coherency among sensory information is still important for 'basic' virtual reality system which measures the human motion and displays visual information. In this case, the coherency between kinesthetic and visual sensation has to be implemented. Many of the 'conventional' virtual reality systems make use of CAD data, which form "closed world" by itself. However, it seems to be difficult to extend them such that they include other kinds of sensory information to generate multiple sensory signals coherently.

In order to achieve the real 'self-projection' feature, the schemes found in 'conventional' systems, which locate only 'camera' and 'hand' in the virtual environment, does not seem to be sufficient. Rather, constructing 'virtual human' model in the virtual environment, which can be considered as a copy of the operator, and drawing the arms, the legs, the body, etc. as well as the hand of the virtual human, it is supposed that more natural sensation of presence is achieved. Moreover, it is considered to be natural if the image which the virtual human's visual receptor would receive is generated and displayed to the operator in the real environment, rather than the image by the 'virtual camera' is generated. It is supposed to be able to display multiple kinds of sensory information coherently by generating information for other kinds of sensation, such as auditory sensation and force/tactile sensation. This concept is illustrated in Fig. 1.

CHARACTERISTICS OF THE SYSTEM

Based upon the above consideration, we developed a virtual reality system by human motion measurement and visual information display, so that the coherency between kinesthetic and visual sensation is achieved. The system has the following characteristics:

- Natural three dimensional virtual space is constructed. To do this, we designed the geometry of generation and display of the image carefully.
- The system is 'fast' enough to realize real-time interaction between the operator and the virtual environment. To realize fast system, the measurement process should have as little time delay and little cycle time as possible, together with fast hardware and algorithm for image generation.
- The system has large degrees-of-freedom to realize natural motion of the operator in the virtual environment without limiting the operator's motion to less degrees-of-freedom. By use of redundant degrees-of-freedom for right arm (measured by exoskeletal manipulator) and head translation information (measured by parallel link mechanism) as well as head orientation information, natural kinesthesis of the operator in the virtual environment is implemented.
- The system can be expanded to multi-sensory system. Currently the system is constructed on kinesthesis and visual information, and it is necessary to be able to add other kinds of sensory information to the system easily. For this purpose, model construction method can easily be expanded to the one integrating multiple sensory information. This can be implemented by generating multiple sensory information displayed to the human operator (image, sound, force, etc.) from the integrated model describing virtual environment.

SYSTEM CONFIGURATION

The system consists of human motion sensing system (tele-existence master manipulator system), a computer for motion sensing (NEC PC-9801 BX, CPU i486SX 20 MHz + ODP), computers for virtual environment construction and image generation (Silicon Graphics IRIS Indigo Elan and XS24, CPU R3000/3010 33 MHz), and a head-mounted display (HMD). The system configuration is shown in Fig. 2, and the block diagram of the system is shown in Fig. 3.

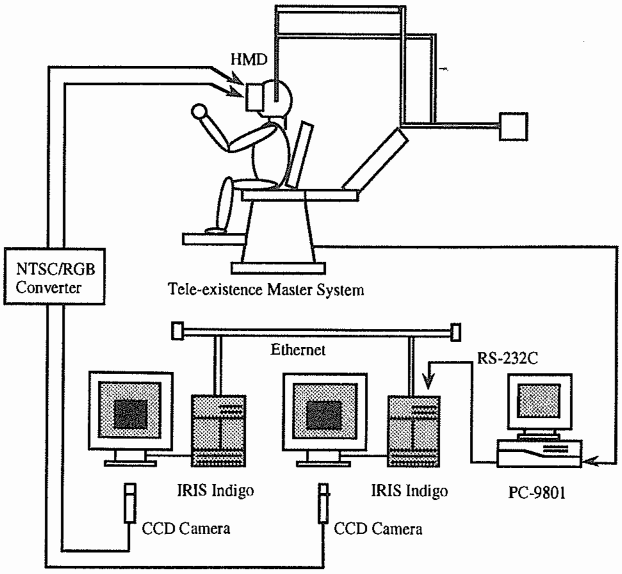


Fig. 2 System configuration

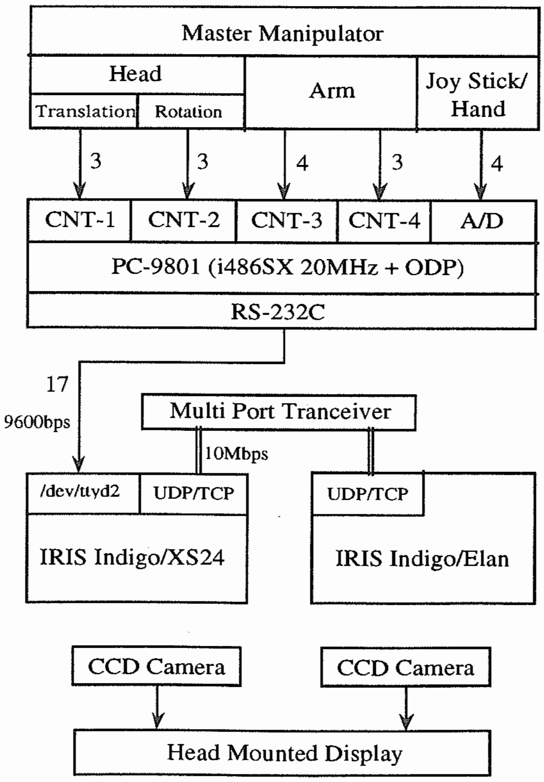


Fig. 3 Block diagram of the system

Human motion sensing subsystem

Our motion sensing system has 17 degrees of freedom (3 for head translation, 3 for head rotation, 7 for right arm joint motion, 1 for hand grip, and 3 for joy stick). The motion of the human operator is measured by rotary encoders and potentiometer, whose output signals are converted into numerical values by up-down counter boards and an A/D converter board installed on the PC. This sensing system can measure human operator motion far faster (more than 1kHz) and with much less time lag (less than the cycle time) than popular three dimensional position/orientation sensing devices used in ordinary VR systems, such as magnetic sensor.

Motion information is transferred from the PC to the IRIS workstation via serial communication channel (RS-232C, 9600 bps). To transfer all of 17 degrees-of-freedom information, data updating rate is about 11 times per second. However, this part can be the bottleneck of the system and it can be improved using faster communication channel (e.g. ethernet).

Virtual environment construction/image generation subsystem

Two IRIS workstations are used to hold the virtual world data, to update the virtual environment according to the data from the master manipulator, and to generate images of the virtual environment for left and right eyes. The IRIS workstations have fast graphics hardware (100k polygons/second on Elan and 25k polygons/second on XS24), which is able to generate images of the virtual environment in real time. These machines also have audio subsystem including digital signal processor (DSP), providing advantage for handling visual and auditory information coherently.

These workstations are connected through ethernet cable and communicate to each other. The human motion data from the PC is received by a server process on the workstation and sent to client processes such as drawing processes using inter-process communication capability (socket interface) of the UNIX operating system. This is done so that the data is available equally by the client processes on all machines on the local area network, including the machine which receives data from the PC. The configuration of these processes is shown in Fig. 4.

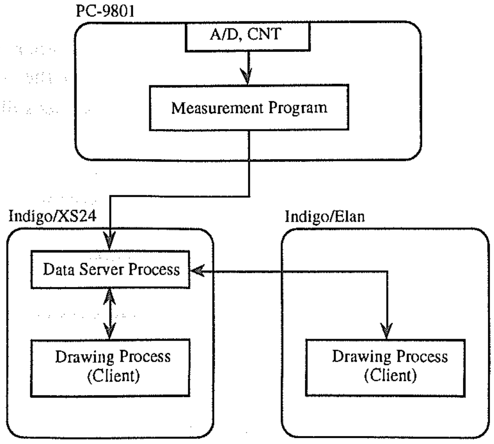


Fig. 4 Process configuration

Image generation scheme

The image of virtual environment is generated using coordinate system illustrated in Fig. 5. Here we use the following notation:

frame attached to the world origin: $\{O\}$
frame attached to the object A: $\{A\}$
frame attached to the object C: $\{C\}$
frame attached to the operator's body: $\{B\}$
frame attached to the operator's head: $\{H\}$
frame attached to the operator's eye: $\{V\}$

transform matrix from frame P to frame Q: ${}^{P}_{Q}T$

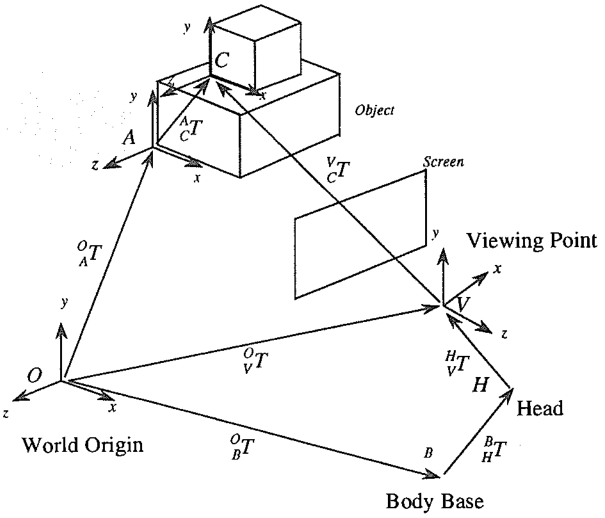


Fig. 5 Coordinate system used for image generation

The position and the orientation of the object with respect to the "world origin" frame, called modeling transformation, is calculated by multiplying transform matrices of objects composing the hierarchical object tree in order. For example, the transform matrix from the frame attached to the world origin to the one attached to the object C, whose location is assumed to be defined relative to the object A, is calculated as follows:

$$ {}^{O}_{C}T = {}^{O}_{A}T\,{}^{A}_{C}T \,. \tag{1} $$

Similarly, "viewing point" frame of the operator in the virtual environment is calculated for left/right eye, respectively (world origin → virtual operator's body → head center → eye):

$$ {}^{O}_{V}T = {}^{O}_{B}T\,{}^{B}_{H}T\,{}^{H}_{V}T \,. \tag{2} $$

Head movement (position/orientation) will be reflected in ${}^{B}_{H}T$. Describing ${}^{O}_{C}T$ via the viewing point, we get

$$ {}^{O}_{C}T = {}^{O}_{V}T\,{}^{V}_{C}T \,. \tag{3} $$

Multiplying the inverse of transform matrix describing viewing point frame, viewing transformation is executed:

$$ {}^{V}_{C}T = \left({}^{O}_{V}T\right)^{-1}\,{}^{O}_{C}T \,. \tag{4} $$

This matrix is multiplied to the vector describing the location of each point on the object (locally defined with respect to the frame $\{C\}$), and finally perspective transformation is executed to generate the image of the object on the projection screen(3).

The field of view of the projection screen is identical to that of the HMD, so that the operator recognizes virtual objects of correct size and at correct distance. To ensure this relationship, the size and location of the graphics window displayed on the CRT is adjusted such that each of four corners of the graphics window is just located on the corner of the image taken by CCD camera (Fig. 6). This calibration is done using joy stick of the master system only once when the camera is initially set.

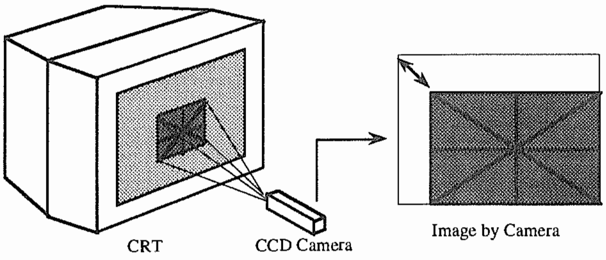


Fig. 6 Window calibration

WORLD DESCRIPTION METHOD

Requirement for the method

We designed the method for describing virtual environment such that the conditions listed below should be satisfied.

- The world should be described easily and flexibly.
- There are various objects is the virtual environment, each of which has attributes about location, color, shape, and so on. It is necessary that the data are structured for each object and that interface for the operations on objects are defined in order that definition and/or modification of the attributes is took place coherently. To implement this requirement, 'object oriented' technology would be suitable.
- To handle multiple objects as a group, hierarchical description should be available.
- The attributes of each object should be specified using physical unit—for example, meter for size and position, degree for orientation. It is preferable if position and/or orientation of the object can be specified or modified in global and in local.

Implementation

The virtual environment is described using the 'class' of C++ language. Implementing in the form of class library, it is easy to describe the virtual environment by this method without any special description language, together with flexibility to expansion for multi-sensory system.

Implemented classes are shown in Fig. 7. The class `Coord3` is used for position in three dimensional coordinate system or size specification. Operator such as summation, subtraction and inner product for this class are defined for easy handling, using overload capability of C++ language. The class `TransformMatrix` denotes 4 × 4 homogeneous transform matrix used for translation, rotation, scaling, and other geometric operations. Operators such as summation, subtraction, multiplication, scalar multiplication/division, and inversion are also defined here.

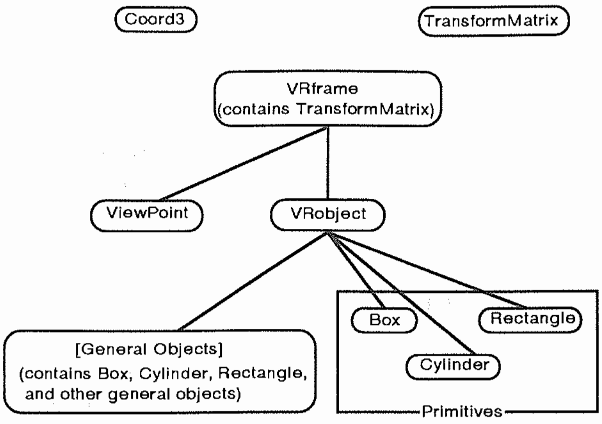


Fig. 7 Implemented classes

The `VRframe` class is provided to describe frames attached to each object. Each instance for this class contains one `TransformMatrix`. Operations on the transform matrix are available. The `VRobject` class is derived from the `VRframe` class, and is implemented to provide framework for general virtual objects. The `ViewPoint` class is also derived from the `VRframe` class. This is used to specify location of virtual human's eye when the image is generated.

Derived from `VRobject` class, a few classes for simple objects called 'primitive' are defined. The currently provided primitives include `Box`, `Cylinder`, and `Rectangle`. Strictly speaking, rectangle is not the real object that has non-zero volume, but it is provided for fast rendering. Combining these simple primitives, it is able to describe objects with rather complex shape.

Actual objects are defined deriving from `VRobject` class. These classes can contain primitives and/or other object classes, which enables hierarchical description of the virtual environment. An example of the object definition is as follows:

```
class Myobject1 : public VRobject {
    Box box1, box2;          // primitive
    Cylinder cylinder1;      // primitive
    Myobject2 sub_object;    // another object
  public:
    Myobject1();             // constructor
    void draw();
};
```

Each object and primitive within this class ('child' object) contains one transform matrix, which holds relative location with respect to this 'parent' object. When moving an object in the virtual environment, we have only to modify the transform matrix of the parent object and need not modify one of the child object, if the location of each sub-object does not change relative to its parent object.

To keep high image generation rate, objects that are obviously out of sight are omitted in rendering. Consider a sphere around the origin of the frame attached to the object, involving all vertices which belong to the object. If any part of the sphere comes into he field of view, rendering for that object is executed, otherwise, omitted. This clipping scheme is shown in Fig. 8.

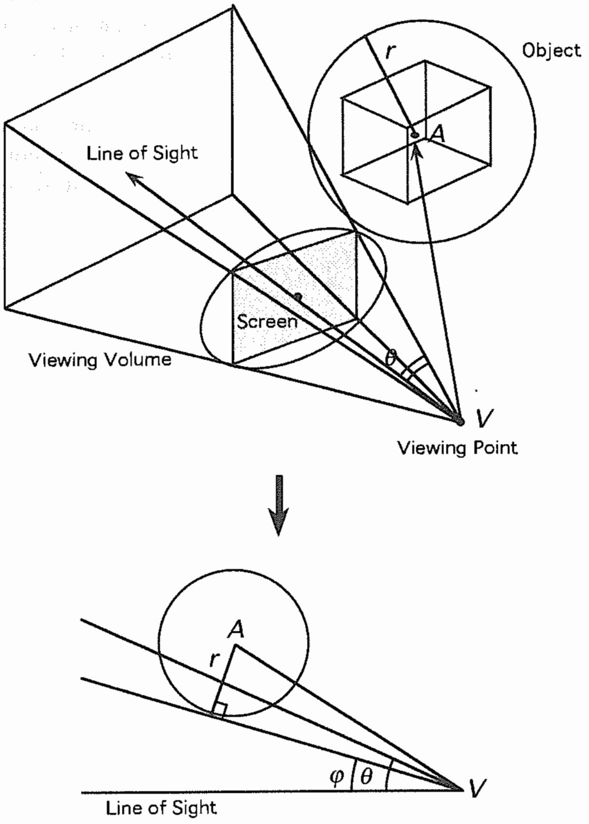
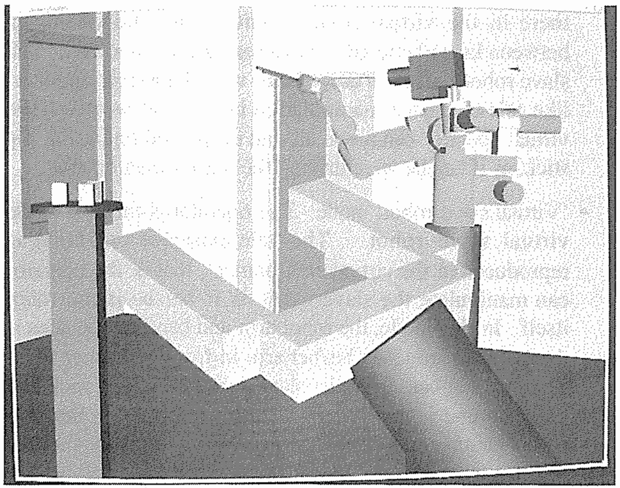


Fig. 8 Clipping scheme

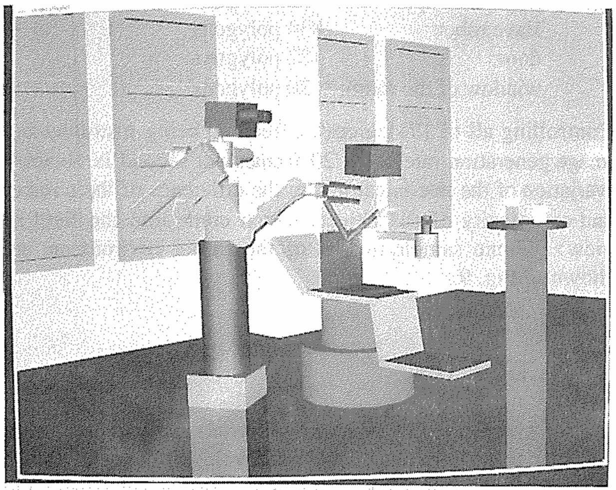
SAMPLE APPLICATION

Using the system described above, we made a sample application program. In this program, a virtual laboratory building is constructed based on the geometry of the real building in which our laboratory is located, and tele-existence master manipulator system and slave robot system$^{(4)}$ are implemented. The arm, hand, legs, and body of the virtual operator are rendered, so that the 'virtual human' is realized. This program has the following 'modes' to realize tele-existence to the virtual operator and to the virtual robot:

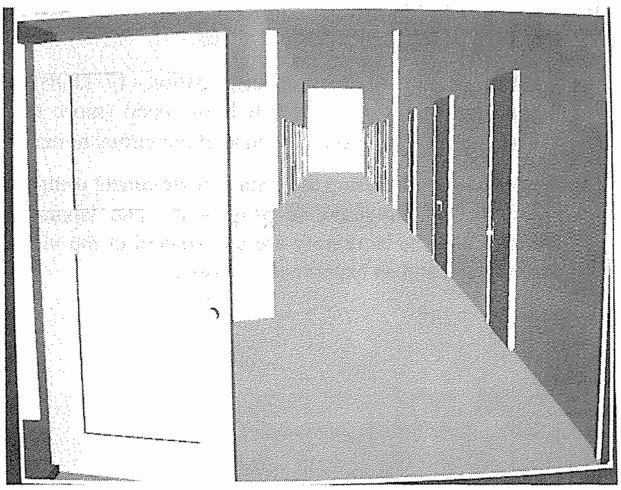
- 'Virtual operator' mode: The operator is mapped to the virtual operator in the virtual environment. The operator sits on the master system, and if he/she put his/her hand in front of the face, he/she can see his/her hand and arm. If he/she looks down, he/she can see his/her legs and body.



(a) Virtual human looking at his/her arm



(b) Slave robot



(c) Corridor of the building

Fig. 9 Images of the virtual environment

Head translation and rotation are both implemented in this mode, so that the operator feels as if he/she were sitting there in the virtual environment. Thus the coherency between kinesthetic and visual sensation is realized. The slave robot moves in accordance with the virtual operator, like tele-existence manipulation in the real world. The virtual operator can move around in the building using joy stick, as if he/she were driving the master manipulator.

- 'Virtual slave robot' mode: The operator is mapped to the virtual slave robot. The tele-existence operation is reproduced in the virtual environment, that is, the operator can manipulate the slave robot as if he/she is the robot itself. In this mode, the robot arm and hand can be seen if the operator stretches his/her arm in front of his/her eyes. The operator can drive the slave robot using joy stick and move around in the building. A simple operation building bricks is implemented in the virtual environment.

Objects in the virtual environment are constructed as follows:

master manipulator:	336 polygons
slave robot:	834 polygons
door:	122 polygons
window of the room:	30 polygons

Controlling all of 17 degrees-of-freedom being instrumented, image generation rate of 10-20 frames per second is obtained (variation of the rate is caused by the difference of the number and complexity of the objects which come into the field of view). Some sample images of the virtual environment are shown in Fig. 9.

CONCLUSION

A virtual reality system by human motion measurement and visual information display is developed considering the condition which provides coherency between kinesthetic and visual sensation of presence.

- The concept of the virtual human is introduced so that the sensation of self-projection is realized.
- A large degrees-of-freedom information (17 DOF) of human motion is measured at high speed (more than 1kHz), realizing high speed control of the virtual human.
- A method for describing the virtual environment using the class of C++ language is proposed. The laboratory building and the laboratory are constructed in the virtual environment and an experiment is made.

REFERENCES

(1) Tachi, S., "Artificial Reality and Tele-existence —Present Status and Future Prospect—", The Second International Conference on Artificial Reality and Tele-existence, pp. 7–12, Tokyo, Japan, 1992.

(2) Tachi, S., "Tele-Existence", Journal of Robotics and Mechatronics, Vol. 4, No. 1, pp. 7–12, 1992.

(3) Tachi, S., Arai, H., Maeda, T., "Tele-existence Simulator with Artificial Reality", Proc. of IEEE International Work Shop on Intelligent Robots and Systems, pp. 719–724, Tokyo, Japan, 1988.

(4) Tachi, S., Arai, H., Maeda, T., "Development of Anthropomorphic Tele-existence Slave Robot", Proc. of International Conference on Advanced Mechatronics, pp. 385–390, Tokyo, Japan, 1989.