In Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (IEEE-MFI '94), Las Vegas, Nevada, USA, pp. 455-462, October 1994.

Coherency of Kinesthetic and Visual Sensation in Virtual Reality System

Yasuyuki Yanagida and Susumu Tachi
Research Center for Advanced Science and Technology
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: yanagida@star.rcast.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, including kinesthetic and visual sensation. To provide coherency between kinesthetic and visual sensation of presence, we have proposed the concept of virtual human. However, this concept has not been realized mainly because of difficulty in description of virtual environment. We developed an object-oriented method for describing virtual environment, which is flexible, easy to construct the environment, and efficient for image generation, using the class of C++ language. We configured a system using human motion measurement and visual information display, and implemented the virtual environment description method on the system. Using this system, we made an experiment on the influence caused by the condition whether the coherency between kinesthetic and visual sensation of presence is satisfied. The result shows that there is significant difference between the case which the above condition is satisfied and that does not.

I. Introduction

To construct natural virtual environment, it is important to satisfy the following conditions [0]:

- the virtual environment should be displayed to the operator constructing natural three dimensional space,
- 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,
- the operator himself/herself should feel as if he/she is inside the virtual environment (self-projection).

The recent progress in computer technology enabled to display graphics images generated by computers in real time, and in accordance with technology of human motion measurement and three dimensional visualization technology, systems realizing the concept of virtual reality has come true. However, the 'conventional' virtual reality systems are not sufficient for further reality in the following sense:

1. Many systems locate only *cameras* and *hands* in the virtual environment, which results in poor self-projection sensation. The operator feels as if he/she is floating in the virtual space, with little feeling that there exists his/her own body.
2. The degrees of freedom of measured motion is not enough to enable operator's natural movement in the virtual environment. For example, many systems use only rotational information of the head motion and does not use translational information, so that the translational motion caused by the body motion wouldn't be reflected in the virtual space. This results in insufficient mapping from the real operator's motion to the motion in virtual environment, hence the sensory information generated based on the status of the virtual environment does not match with the real operator's kinesthetic sensation.

Here we define the term *coherency* to be the state that satisfies the following condition:

> *The operator is self-projected in the virtual environment, such that kinesthetic sensation and displayed sensory information are integrated consistently in the human sensation.*

This coherency among multiple kinds of sensory information is still important for 'basic' virtual reality system

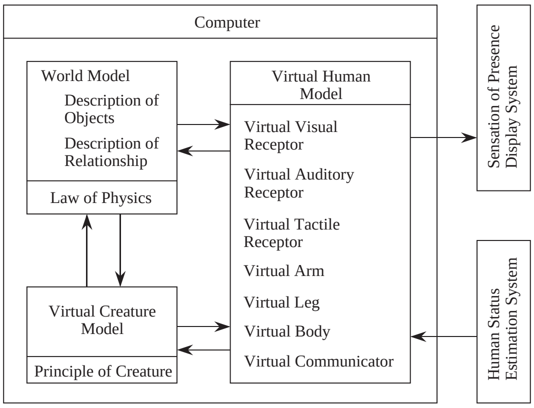


Fig. 1. Concept of the system.

which measures the human motion and displays visual information. In this case, the coherency between kinesthetic and visual sensation has to be implemented.

In order to achieve the self-projection feature, we have proposed the concept of *virtual human*, who exists in the virtual environment and acts and feels as human in the real world would do [0]. The virtual human can be considered as a copy of the operator. By 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 would be natural to consider the image displayed to the operator in the real environment is identical to what the virtual human's visual receptor would receive. It is considered that the coherency among multiple kinds of sensory information such as auditory sensation and force/tactile sensation is achieved by a single model of virtual human. This concept is a realization of the concept of *tele-existence* to the virtual environment, and is illustrated in Fig. 1.

This concept of virtual human, however, has not be implemented satisfactorily. The main reason for making it difficult to realize this concept is the lack of method for consistent description and manipulation of objects in virtual environment, including virtual human itself. Many virtual reality systems make use of CAD data, which form closed world by itself, and their method for describing virtual environment is provided in the form of function library. Such a method is useful for interactive design of constructing three dimensional shape. However, it seems to be difficult to describe consistent and real time manipulation of the objects, and to extend them such that they include multiple kinds of sensory information to generate multiple sensory signals coherently.

In this paper, a flexible method for virtual environment description using C++ class library is introduced, aiming at realize tele-existence to the virtual human and considering real time interaction between the operator in real world and virtual environment. Next, this method is implemented on the system which consists of human motion mesurement and visual display. Using this system, an evaluation experiment is made to show that that the construction and appropriate design of virtual human is important for manipulation in virtual environment.

II. Requirement for The System with Virtual Human

The virtual reality system which realize virtual human should satisfy the requirements listed below.

- Natural three dimensional virtual space should be composed. It is important that the size and location of the objects is described in real scale using physical units.

- The virtual human in the virtual environment should be able to move freely and naturally, that is, the system should have large degrees-of-freedom to realize natural motion of the virtual human. For example, by measuring head translation as well as head rotation, kinesthetic sensation of the operator's body can be implemented to the virtual environment.

- The system should be fast enough to realize real-time interaction between the operator and the virtual environment. To realize fast system, the motion measurement system should have as little time delay and small cycle time as possible, together with fast hardware and algorithm for image generation.

- The self projection scheme should be implemented. By displaying the operator's body, arm, etc. at appropriate position in accordance with his/her motion, he/she can feel as if he/she exists in the virtual environment.

- The size of the virtual human should exactly reflect that of the real operator. When the tele-existed world has the scale identical to that of the world which the real human lives, the size of the effector to the world (hand), the arm length which decides the location of the hand, and the location of the visual receptor (eyes) should be set identical to the real operator. If this condition is corrupted, for example, if the distance between two eyes differs from the actual, the convergence becomes different and perception of distance to the object would be skewed.

- The sensory information obtained by the receptor of virtual human should be displayed to the human operator correctly. Speaking of visual information, the displayed image should be generated considering the structure and parameters of visual display device. If the motion of the virtual human exactly reflects the motion of the operator and this condition is held, the coherency between kinesthetic sensation and displayed visual information is accomplished.

- There are various objects in 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 taken place coherently. To implement this requirement, 'object oriented' technology would be suitable.

III. Virtual Environment Description Method

Considering the requirement discussed in chapter II., the virtual environment description method was designed. The method was implemented using class library of C++ language, so that object oriented scheme is introduced.

In designing the method, the following items are taken into account.

- The world can be described easily and flexibly.
- Hierarchical description of the world is supported to handle multiple objects as a group.
- The attributes of each object should be specified using physical unit. The position and/or orientation of the object can be specified or modified in global and in local.
- The description model can be expanded to multisensory system, that is, the model can be contain attributes other than those used in image generation.

A. *Coordinate System*

1) Description of frame and point: The specification of the position and orientation of each object is described using the frame attached to that object. This frame is described using 4×4 homogeneous transform matrix. The shape of an object is described by the set of points on the object described by the coordinate system (frame) attached to that object.

Let $\{O\}$ be the frame of the world origin, and $\{A\}$ be the frame attached to the object A. The point P_A on the object A described locally with respect to $\{A\}$ is

$$^{A}P_A = [x, y, z, 1]^T, \tag{1}$$

where ^{T} denotes transpose.

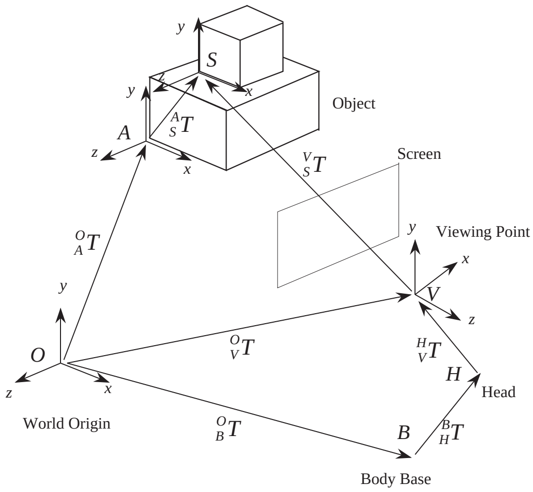


Fig. 2. Coordinate system used for image generation.

2) Coordinate description with respect to world origin: Now we use the notation $^{X}_{Y}T$ to describe transform from frame $\{X\}$ to $\{Y\}$. Then the description of P_A with respect to the world origin frame $\{O\}$ is

$$^{O}P_A = {}^{O}_{A}T \; {}^{A}P_A. \tag{2}$$

Next, consider the case that the object A has a subobject S. Let $\{S\}$ be the frame attached to S and $^{S}P_S$ be the description of the point P_S on the object S, then the description of P_S with respect to $\{A\}$ will be

$$^{A}P_S = {}^{A}_{S}T \; {}^{S}P_S. \tag{3}$$

The description of P_S with respect to the world origin frame is calculated by multiplying transform matrices of objects composing the hierarchical object tree in order:

$$^{O}P_S = {}^{O}_{A}T \; {}^{A}P_S = {}^{O}_{A}T \; {}^{A}_{S}T \; {}^{S}P_S. \tag{4}$$

Thus the location of objects with respect to the world origin coordinate system can be calculated correctly even if the location of 'parent' object (${}^{O}_{A}T$) changes, as long as holding the local transform ${}^{A}_{S}T$.

3) Image generation: Now let $\{B\}$ be the frame attached to the operator's body, $\{H\}$ be the operator's head, and $\{V\}$ be the operator's eye. The image of virtual environment is generated using coordinate system illustrated in Fig. 2.

The position and the orientation of the object S with respect to the 'world origin' frame, called modeling transform, is calculated as described.

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 *to* eye):

$${}^{O}_{V}T = {}^{O}_{B}T \; {}^{B}_{H}T \; {}^{H}_{V}T. \tag{5}$$

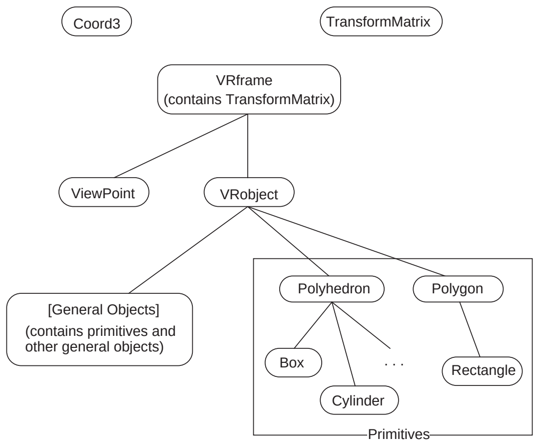


Fig. 3. Implemented classes.

Head movement (position/orientation) will be reflected in ${}^{B}_{H}T$. The viewing transform can be described as $({}^{O}_{V}T)^{-1}$. Describing the location of the object S via the viewing point, we get

$$ {}^{O}_{S}T = {}^{O}_{V}T \, {}^{V}_{S}T. \tag{6} $$

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

$$ {}^{V}_{S}T = \left({}^{O}_{V}T\right)^{-1} \; {}^{O}_{S}T. \tag{7} $$

This matrix is multiplied to the vector describing the location of each point on the object (locally defined with respect to the frame $\{S\}$), and finally perspective transform is executed to generate the image of the object on the projection screen [0].

B. *The Class Library*

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 without any special description language, together with flexibility to expansion for multi-sensory system.

Implemented classes are shown in Fig. 3. The class `Coord3` is used for position in three dimensional space or size specification. Operator such as summation, subtraction and inner product for this class are defined for easy handling, using overload capability for operators 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.

The `VRframe` class is provided to describe frames attached to each object. Each instance for this class contains one `TransformMatrix` object. Operations on this transform matrix are available. The `VRobject` class is derived from the `VRframe` class, and is implemented to provide framework for general virtual objects. An object of this class contains information of the node which constructs the world tree, i.e., the pointers to the parent and children objects. The transform matrix contained in this class object holds the local coordinate transformation relative to the frame of 'parent' object. The `ViewPoint` class is also derived from the VRframe class. This is used to specify location of virtual human's eye for image generation.

Derived from `VRobject` class, `Polyhedron` and `Polygon` classes are defined. `Polygon` is used to define a single polygon, and `Polyhedron` is used to define a general polyhedron in the virtual space. *Primitives* are defined by deriving from `Polyhedron` and `Polygon`. The currently provided primitives include `Box`, `Cylinder`, and `Rectangle`. Combining these simple primitives or defining other 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
    Myobject2 sub_object;   // another object
                             // (already defined)
  public:
    Myobject1();             // constructor
};

Myobject1::Myobject1
{
    // set local position of child objects
    box1.setposition(-1.0, 0.0, 0.0);
    box2.setposition( 1.0, 0.0, 0.0);
    sub_object.setposition(-2.0, 0.0, -3.0);
    // register these objects as children of
    // this object
    addmember(&box1);
    addmember(&box2);
    addmember(&sub_object);
}
```

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, as long as the location of each sub-object does not change relative to its parent object.

C. *Technique for efficient image generation*

To realize real time interaction between operator and the virtual environment, fast image generation is necessary. A few techniques are adopted for efficient image generation.

1) Geometry compilation: As discussed in section III.A., location of objects in the world (global matrix) can be calculated by multiplying transform matrices describing local transform from the root of the world tree to the object in order. However, if the object *never* moves, calculation of such matrices whenever a new image is generated is a waste of CPU resource. The library provides the functionality to calculate transform matrices and vertex coordinates of such objects only once at startup. Here we call this calculation *geometry compilation*. Using geometry compilation, a great deal of calculation can be omitted and hence higher image generation rate is obtained.

2) Block clipping: 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. 4.

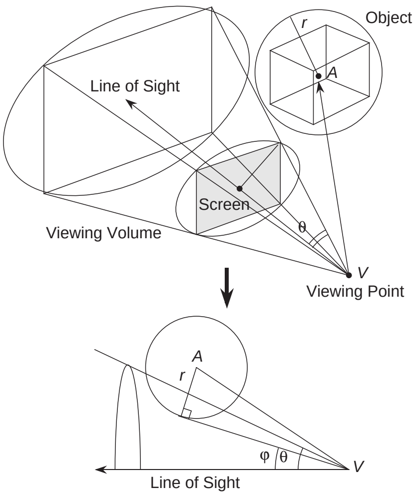


Fig. 4. Block clipping scheme.

IV. Experimental System

To show the feasibility of the virtual reality system including virtual human, we configured an experimental system[0] by human motion measurement and visual information display, so that the coherency between kinesthetic and visual sensation is achieved.

A. *Hardware Configuration of the System*

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. 5, and the block diagram of the system is shown in Fig. 6.

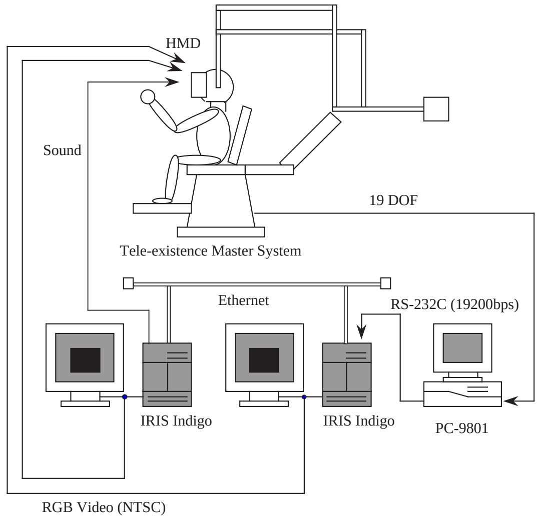


Fig. 5. Configuration of the experimental system.

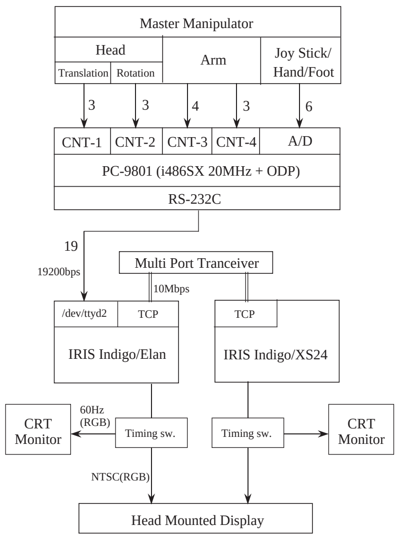


Fig. 6. Block diagram of the system.

1) Human motion measurement subsystem: The motion measurement system has 19 degrees of freedom (3 for head translation, 3 for head rotation, 7 for right arm joint motion, 1 for hand grip, 3 for joy stick, and 2 for foot pedals). 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 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 sensors.

Motion information is transferred from the PC to the IRIS workstation via serial communication channel (RS-232C, 19200 bps). To transfer all of 19 degrees-of-freedom information, data updating rate is about 25 times per second.

2) 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 system, 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. 7.

The field of view of the projection screen is set identical to that of the HMD, so that the operator perceives virtual objects of correct size and at correct distance.

B. *Example Application*

Using the system described above, we made an example 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 system and slave robot system [0] are implemented. The arm, hand, legs, and body of the virtual operator are rendered, so that the 'virtual human' is realized. This program has two modes to realize tele-existence to the virtual environment; Virtual operator mode and Virtual slave robot mode.

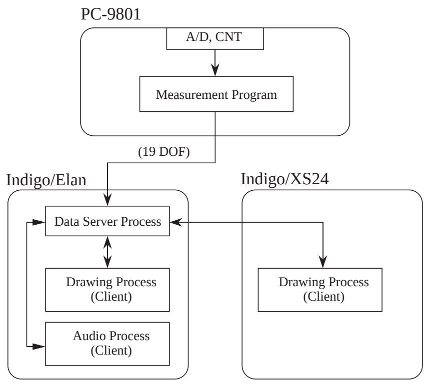


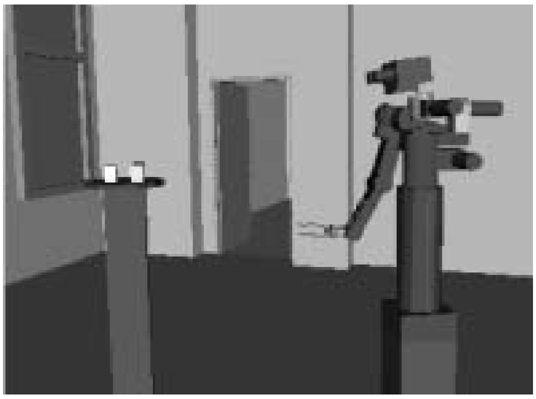
Fig. 7. Process configuration.

When in 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. Head translation and rotation are both implemented, 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 system.

When in Virtual slave robot mode, the operator is mapped to the virtual slave robot. The tele-existence operation is implemented 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 of building blocks is implemented in the virtual environment.

Objects in the virtual environment are constructed as follows: master system (336 polygons), slave robot (834 polygons), door (122 polygons), window of the room (30 polygons), and so on.

Controlling all of 19 degrees-of-freedom being measured, image generation rate of 10–20 frames per second



(a) Slave robot

(b) Corridor or the building

Fig. 8. Images of the virtual environment.

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. 8.

V. Experiment

In order to evaluate the effect of coherency between kinesthetic and visual sensation, an experiment on operation of building blocks is made. This task is the one used to evaluate the typical characteristics of the tele-existence manipulation system [0].

An operator was asked to move from the initial place to the place near to the table by using joy stick. He was also asked to pick up one of the three blocks placed randomly on the table and put it on another block, and pick up again the rest of the block and place on the top of the two blocks by using his virtual hand. The time elapsed from the start of the locomotion till the end of the manipulation of building three blocks was measured.

The following nine conditions on virtual human and image generation were compared:

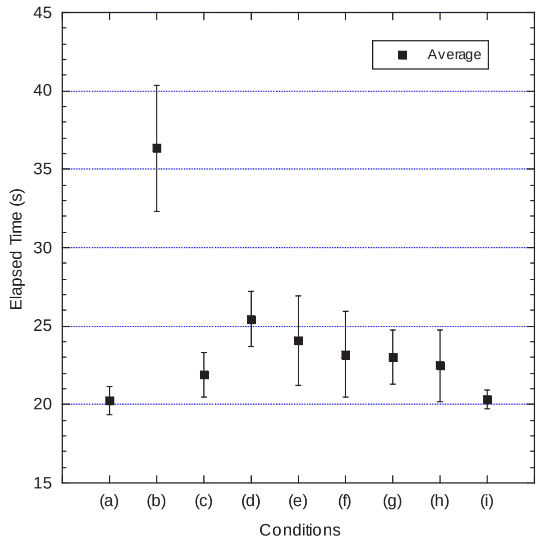


Fig. 9. Elapsed time for the completion of the task.

(a) Normal condition. All of the kinematic parameters and visual display conditions are set identical to the real operator.

(b) The head translation (caused by body movement) of the virtual operator was set four times as large as the real operator.

(c) The head translation of the virtual operator was switched off—there was no translation of the virtual operator's head regardless of whether the operator in the real environment moves his/her head or not.

(d) The head rotation of the virtual operator was set twice as large as the real operator.

(e) The virtual operator's eye distance is set twice as long as the real operator.

(f) The angle of the field of view for generated image is set to one and a half as large as the field of view of the HMD.

(g) Both the angle described in (f) and the head rotation described in (d) are set one and a half as large as actual.

(h) The virtual human's arm is set one and a half as long as the real operator.

(i) All of head translation (b), eye distance (e), and arm length (h) are set one and a half as large as actual.

Each run of the experiment consists of 5 trials and the averages of the data were compared. Fig. 9 shows the

TABLE I
RESULT OF THE T-TEST.

Condition	(a)	(b)	(c)	(d)
(b)	9.7649			
(c)	3.4428	9.1126		
(d)	5.3049	4.4637	3.6467	
(e)	3.2950	8.0768	3.0832	0.9724
(f)	1.8712	5.6234	0.9155	1.2529
(g)	2.9187	5.4443	1.2402	3.5212
(h)	2.0579	6.3175	0.7912	3.1612
(i)	0.1804	7.5183	2.2967	6.7583

Condition	(e)	(f)	(g)	(h)
(f)	0.5284			
(g)	0.7738	0.1000		
(h)	1.9025	0.4295	0.8297	
(i)	2.7927	1.9482	4.8402	2.3325

result of the average time elapsed for the completion of the task with a standard deviation. Comparing average elapsed time for normal condition (a) with other conditions, every condition except (i) seems to require longer time for completion of the task.

In order to determine the statistical significance of the apparent differences of resultant averages for the different display conditions, the data was analyzed using t-test with a risk level of 5 percent. Table I shows the resultant t value of the t-test.

Each combinations of the conditions is statistically significant if $t > 2.776$. Comparing normal condition (a) with other conditions, (b), (c), (d), (e), and (g) was decided to be statistically significant, while (f), (h), and (i) was not.

The condition (b) emerged significant difference, and the subject reported that with this condition the task was hard to complete. This result shows that the sensation of presence is damaged if the head translation caused by the body motion does not reflect the actual motion. The result of (d) shows that the simple amplification of the head rotational motion regardless of the parameter of visual display causes degradation of reality. The condition (e) is the case which the virtual operator's 'face' becomes bigger, which shows that the simple change of the receptor results in less reality when tele-exist to an object whose size is different from the real human. The result of (f) shows the necessity of the consistency between the field of view of the visual display device and the generated images. The subject reports that the condition (g) provided natural feeling compared with (f), though the result of (g) was not as good as (a). The result of (h) shows necessity of careful scaling when tele-existing to objects such as robots with manipulators of different size. The result of (i) shows little difference from (a). Comparing (i) with (b), (e), and (h), every comparison showed statistically significant difference. This result shows that it is important to execute scaling considering all parameters when tele-existing to the object with different size.

VI. CONCLUSION

A virtual reality system by human motion measurement and visual information display which implements the concept of virtual human is developed considering the condition, providing coherency between kinesthetic and visual sensation of presence. A large degrees-of-freedom information (19 DOF) of human motion was measured at high speed (more than 1kHz), realizing real-time control of the virtual human. A method for describing the virtual environment using the class of C++ language is proposed, and a virtual environment including the laboratory building and the laboratory were constructed.

Using the system, an experiment was made to show the effectiveness when the condition requirements for the coherency between kinesthetic and visual sensation is set appropriately. The result shows that there are statistic significance between the case which the condition is set identical to the real operator and those of which the condition is violated.

REFERENCES

[1] S. Tachi: "Artificial Reality and Tele-existence—Present Status and Future Prospect," Proc. of the Second International Conference on Artificial Reality and Tele-existence, pp. 7–12, Tokyo, Japan, 1992.

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

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

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

[5] S. Tachi and K. Yasuda, "Evaluation Experiments of Tele-Existence Manipulation System," Proc. of the Third International Conference on Artificial Reality and Tele-Existence (ICAT '93), pp. 17–26, 1993.

[6] Y. Yanagida, S. Tachi, "Virtual Reality System with Coherent Kinesthetic and Visual Sensation of Presence," Proc. of the 1993 ISME International Conference on Advanced Mechatronics, pp. 98–103, Tokyo, Japan, 1993.

[7] S. Tachi and K. Yasuda, "Experimental Evaluation of Tele-Existence Manipulation System," Proc. of the 1993 ISME International Conference on Advanced Mechatronics, pp. 110–115, Tokyo, Japan, 1993.