Full-scale and Real-time Virtual Experiments in Dynamics by using an Immersive Projection Display and Hand Manipulation

Hiroshi Nakano
Center for Multimedia and Information Technologies
Kumamoto University
Kurokami 2-39-1, Kumamoto 865-8555
Japan
nakano@cc.kumamoto-u.ac.jp

Kenichi Tokunaga
Faculty of Management Information Science
Nagoya University of Commerce
Business Administration
Sagamine 4-4, Komenoki, Nisshin, Aichi 470-0193
Japan
tkenichi@nucba.ac.jp

Noritaka Osawa
National Institute of Multimedia Education
2-12 Wakaba, Mihama-ku
Chiba 261-0014
Japan
osawa@nime.ac.jp

Hidenori Akiyama
Graduate School of Science and Technology
Kumamoto University
Kurokami 2-39-1, Kumamoto 865-8555
Japan
akiyama@eecs.kumamoto-u.ac.jp

Abstract— **We improved our three-dimensional (3D) virtual experiment system on web browser for distance learning of physics and adapted for an immersive projection display and hand manipulation. The system realizes full-scale and real-time virtual experiment by using the virtual environment system; TEELeX (Tele-Existence Environment for Learning eXploration) at NIME (National Institute of Multimedia Education, Japan) and it3d (Interactive Toolkit library for 3D applications) library. Learners not only do 3D virtual experiments on web browser but also face an experience of full-scale and real-time experiments with user interfaces in the virtual environment.**

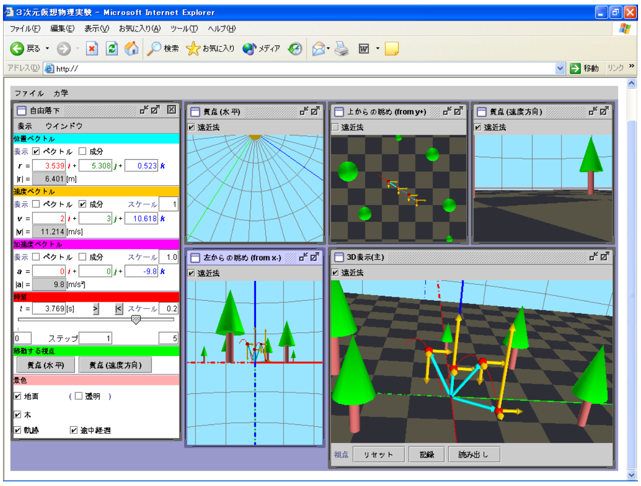
I. INTRODUCTION

Distance educations (DE) for natural sciences and technologies are studied by many teachers, schools, universities and so on through the spread of the World Wide Web (WWW). A lot of works of Java applets have been developed [1], [2] and these applets are effective and helpful in understanding scientific phenomena. Although very few applets support three-dimensional (3D) views.

In recent years, 3D graphics performance of personal computers (PCs) is being enhanced dramatically and the cost is also going down. The performance is almost better than graphic workstations shipped in few years ago. Java3D application programming interface (API) [3] appeared and it can provide 3D graphics on Java.

We have developed 3D virtual experiment system on web browser for distance learning for physics as shown in Fig. 1 [4]. All of the following specific features are realized on web browser; several views from different viewpoints with perspective or parallel projections, views from the moving material point, simultaneous update of all views, real-time motion, visualized vectors, precise scale of geometry and internationalization.

We started to improve and adapt the system for virtual reality. Learners can do 3D virtual experiments on web browser anytime and anywhere, furthermore, they can take another step in virtual reality installations to face an experience of full-scale and real-time experiments in virtual reality with the same system. This paper mainly presents extensions of the system for adapting the full-scale virtual environment with graphical user interfaces and hand manipulation.


3次元仮想物理実験 - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 お気に入り メディア
アドレス(D) http://
移動 リンク
ファイル 力学
自由落下
表示 ウインドウ
貫点 (水平)
上からの眺め (from y+)
貫点 (速度方向)
左からの眺め (from x-)
3D表示(主)
視点 リセット 記録 読み出し

位置ベクトル
表示 ☑ ベクトル □ 成分
r = 3.539 i + 5.308 j + 0.523 k
|r| = 6.401 [m]
速度ベクトル
表示 □ ベクトル ☑ 成分 スケール 1
v = 2 i + 3 j + 10.618 k
|v| = 11.214 [m/s]
加速度ベクトル
表示 □ ベクトル □ 成分 スケール 1.0
a = 0 i + 0 j + -9.8 k
|a| = 9.8 [m/s²]
時刻
t = 3.769 [s] > |< スケール 0.2
0 ステップ 1 5
移動する視点
貫点 (水平) 貫点 (速度方向)
景色
☑ 地面 (□ 透明)
☑ 木
☑ 軌跡 ☑ 途中経過
☑ 遠近法 (貫点 (水平))
□ 遠近法 (上からの眺め (from y+))
☑ 遠近法 (貫点 (速度方向))
☑ 遠近法 (左からの眺め (from x-))
☑ 遠近法 (3D表示(主))

Fig. 1. The original system provides 3D virtual experiments in dynamics on a web browser (InternetExplorer on Windows XP, locale: Japanese).

II. SPECIFIC FEATURES

A. Original Features

Our original 3D virtual experiment system for dynamics [4] has been provided as a Java applet on web browser as shown in Fig. 1, which supports Java2[5] and Java3D[3] API. It also runs as a Java application. Several 3D views are attached to the single 3D virtual world by using Swing API [6], and all the views are simultaneously updated. Perspective and parallel projections are selectable for each 3D views. The viewpoint can be attached on the moving material point and this impact can not be given in real experiments. The vectors of position, velocity and acceleration can be visualized as 3D arrows respectively. The axes for the rectangular coordinates, the celestial map, precisely scaled checker-patterned ground plane and some trees are provided for recognizing the scale of the virtual world. The real-time or correctly scheduled time scale are important for the motion in dynamics and are implemented.

We improved and adapted the system for virtual reality, keeping all of the original features.

B. Virtual Environment System

We adopted TEELeX[7] as a virtual environment system; TEELeX is an abbreviation for Tele-Existence Environment for Learning eXploration developed by National Institute of Multimedia Education (NIME) in Japan. TEELeX can give us immersive experiences of the 3D environment, and this is applicable to an open and flexible learning.

TEELeX uses immersive projection technology that a multi-screen display enables one or more persons to experience the sensation of being completely surrounded by high-resolution, 3D video and audio. The origin is the CAVE[8] developed by University of Illinois.

Various kinds of devices are prepared for users; position/orientation tracker and sensor gloves, 3D mouse, track ball, joystick, voice cognition, are so on. The position/orientation tracker and the sensor glove for the left hand are supported by our experiment virtual experiment system in order to realize user interface in TEELeX.

TEELeX has a cubic display composed of six screens for five rear-projections (front, right and left side, back, and ceiling) and one front-projection (floor). We use only a front screen for our virtual experiment system at present, however, it is still effective with one large screen (3m × 3m) and it seems easy to support plural displays because new views can be added to our system's 3D world quite easily.

C. GUI Library for Virtual Environment System

We adopted it3d[9] as a graphical user interface (GUI) library for virtual environment systems; it3d is an abbreviation for Interactive Toolkit library for developing 3D applications. It3d has a 3D widget library for multimodal interfacing, and an interaction-recognition library.

The input/output library for distributed devices has a uniform programming interface style for various types of devices, and it supports position/orientation trackers and the sensor gloves which are used by our experiment system. The library utilizes multicast peer-to-peer communication between devices and drawing clients and it reduces load of clients.

It3d is implemented by using the Java language and the Java 3D class library to enhance its portability.

D. Constructing Virtual World

We developed our system by using Java2 and Java3D API. Java3D API provides high-quality, scalable, platform-independent 3D graphics into Java applications and applets. All the 3D views in the system are drawn by Java3D API.

SimpleUniverse class in Java3D API is ready to be used easily, however, it is not so convenient for multi-view systems. We therefore developed two important classes in order to realize virtual world with enhanced portability as shown in Fig. 2.

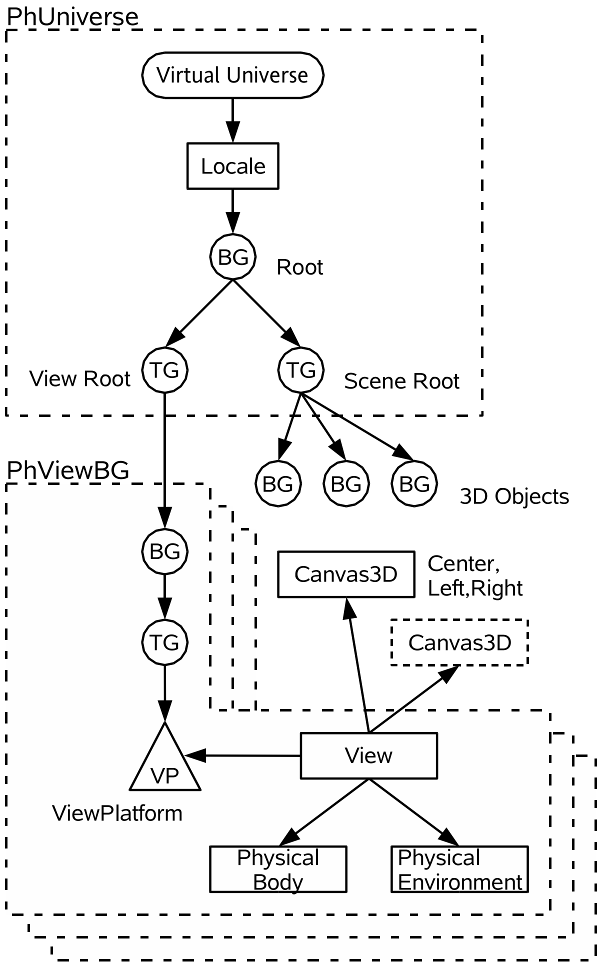


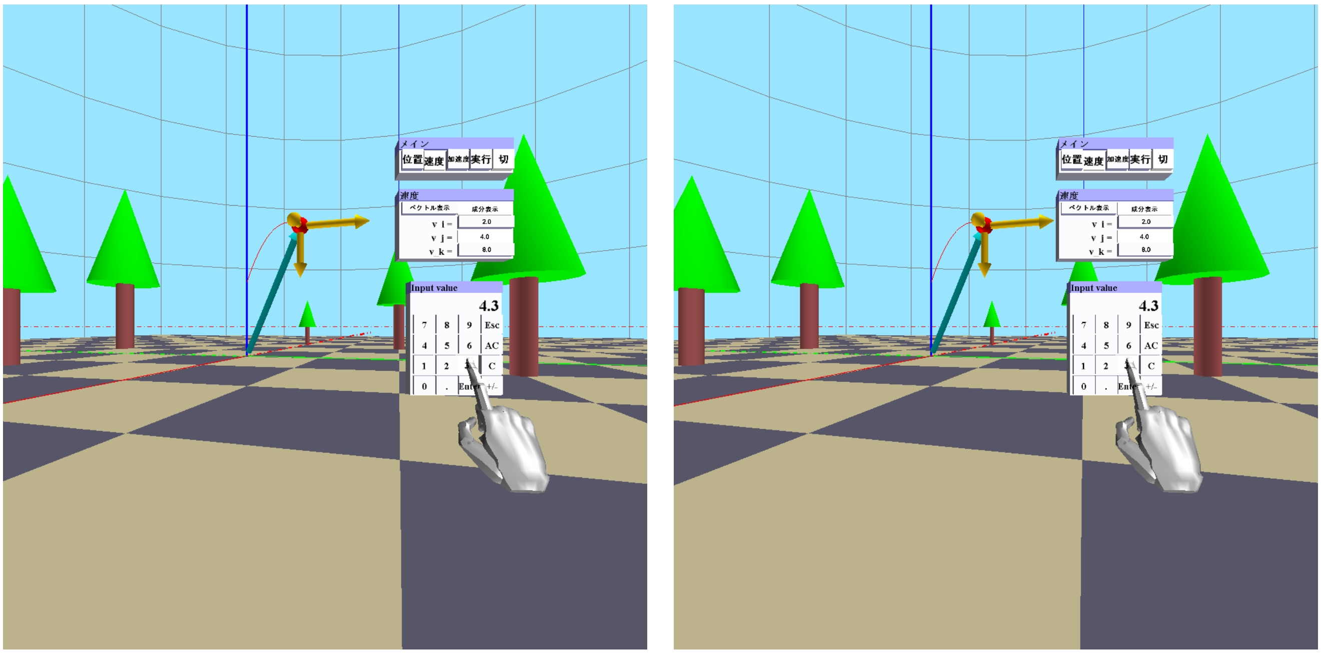
Fig. 2. Program structure for virtual world with enhanced portability. BG, TG, circle and triangle are corresponding to BranchGroup, TransformGroup, Group and Leaf nodes, respectively.

PhUniverse class in Fig. 2 is a root node for a virtual world and has two TransformGroups for view and scene branches. PhViewBG class provide a view construction and it can be added to any TransformGroups, such as the TransformGroup for view branch in PhUniverse class. We can attach not only a number of 3D objects but also a number of views to our virtual world. It means Several 3D views can be attached to the single 3D virtual world, and all the views are simultaneously updated. A number of Canvas3D classes can be attached to a View object in the PhViewBG class and stereo properties can be added by setMonoscopicViewPolicy for each Canvas3D class. Finally stereo screens for several views are realized with these classes.

E. Full-Scale Views in Virtual Environment System

View parameters should be set carefully to obtain full-scale view in virtual environment systems.

The projected area of 1000 × 1000 dots is assigned for a screen (3m × 3m) of TEELeX. Two 3D views for left and right eyes are created by two Canvas3D objects on Swing's JFrames components without frames by using of setUndecorated() method, which is supported from



v_i = 2.0
v_j = 4.0
v_k = 8.0
4.3

Fig. 3. Stereo views for left- and right-eyes in the full-scale experiment (locale: Japanese).

Java version 1.4. Each 3D view has an area of 1000 × 1000 dots on different display. Many graphics board for general PCs recently have two outputs and therefore such a dual outputs system can be constructed easily. Two outputs are sent to different projectors and are projected to the same screen through differently circular polarized glass.

The view model's back clip policy should be set to PHYSICAL_SCREEN, which specify that the associated distance is from the screen and in units of physical distance, by setBackClipPolicy() method of the View class. Both the screen's physical widths and heights should be set to 3m by setPhysicalScreenWidth() and setPhysicalScreenHeight() methods of Screen3D component for each Canvas3D. The field of view should be adjusted to 90 degrees by the method of setFieldOfView() in View class, because TEELeX has a cubic display and learners are assumed to be standing at the center of the cube.

Screen copies of two outputs are shown in Fig. 3. It is not easy to feel depth within these figures. However, width and depth are sufficiently realized and hand object is shown as almost the same size as a learner's hand in TEELeX. Figure 4 indicates the size comparison of several objects by the use of another view featured in the original experiment system[4].

F. Connecting Virtual World to it3d

Our experiment system seems to be connected to it3d easily, because both it3d and our system are implemented by using the Java language and the Java 3D class library to enhance its portability, however, there is a problem for the connection.

It3d library has a own tree-structured virtual world and it is very convenient to build new applications along its guideline by reference to API documents and sample programs. We, however, wanted to adapt our system for virtual reality, keeping all of the original features. We therefore must find out how to connect our system's virtual wold with it3d's one. We introduce the procedure which can be applied to connecting general Java3D's virtual worlds with it3d's one.

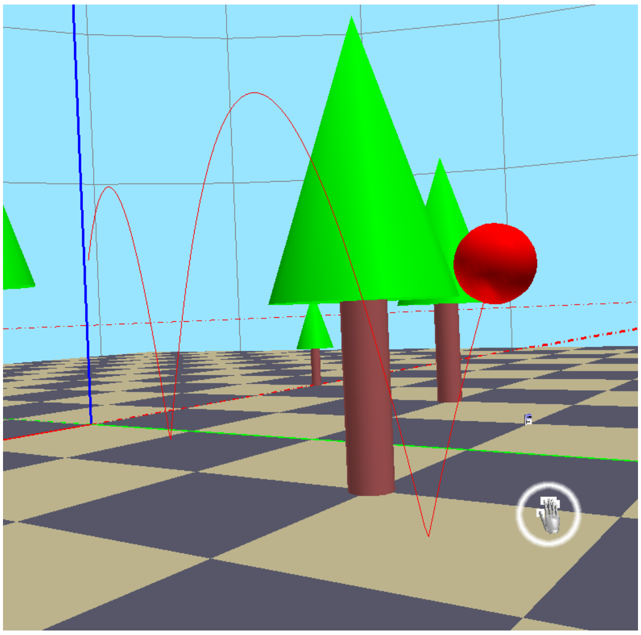


Fig. 4. View from a similar distance to the front-most tree, the ball and the glove, where the glove object of the leaner is white circled. (part of screen)

Figure 5 shows a diagram of connecting these different virtual world. New class MyContainer extending ContainerImpl3D in it3d was written as following.

```
import org.it3d.widget.*;
import javax.media.j3d.*;
public class MyContainer3D
        extends ContainerImpl3D {
```

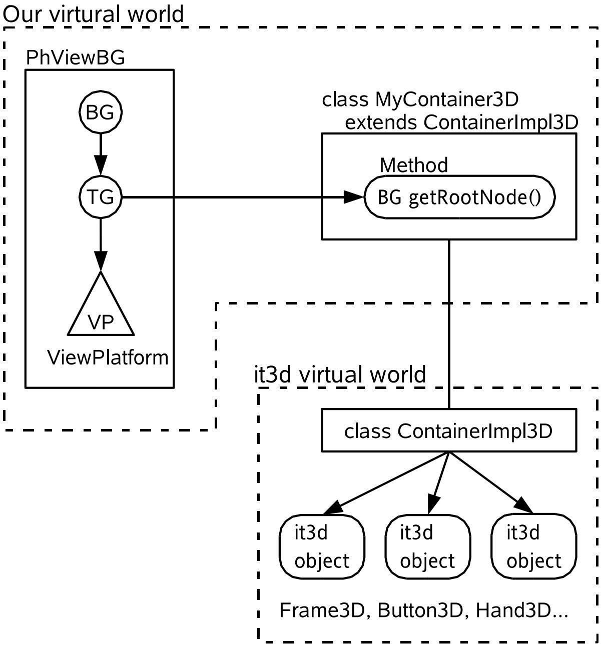


Fig. 5. Diagram of connecting virtual worlds between our system and it3d.

```
    public MyContainer3D() {super();}
    public BranchGroup getRootNode() {
        return rootNode;
    }
}
```

Almost all of the it3d's component can be added to this node because it extends it3d's class. It can pick up and return the root node of it3d's virtual world. The root node of it3d can be added to any TransformGroup, such as the view branch's one of our virtual world in Fig. 2. We therefore can use it3d's components through it as shown in Fig. 5. It3d's components also can be added to the scene branch of our system's virtual world, however, view branch is more convenient that GUI components follow view point and always stay in front of the learner as a cockpit.

G. *Experiment in Virtual Environment*

A small button and a right hand appear at starting the experiment in virtual environment TEELeX as shown in Fig. 6. The hand is following and emulating the learner's right hand by the signal from the sensor glove; the position, orientation and the angles of all joints are represented.

When the learner push the small button by a finger of right hand in virtual world, a main menu will appear. The main menu is shown as a control panel which is constructed with five buttons on a panel, as shown as the top panel in Fig. 7. Such panels can be reset to the initial small button any time in order to avoid interfering the view by using the rightmost button on the main menu. Other four buttons have sub menus for setting parameters of position, velocity, acceleration, and running a simulation, respectively.

The initial value of three physical quantities; position, velocity and acceleration can be set as x-, y- and z-components of rectangular coordinates as shown in Fig. 7. When the learner push one of the component's button (the lowest button of the middle panel in Fig. 7), a panel like a calculator in order to input a value. After setting a value and pushing the enter button, the panel will disappear and the value will be entered.

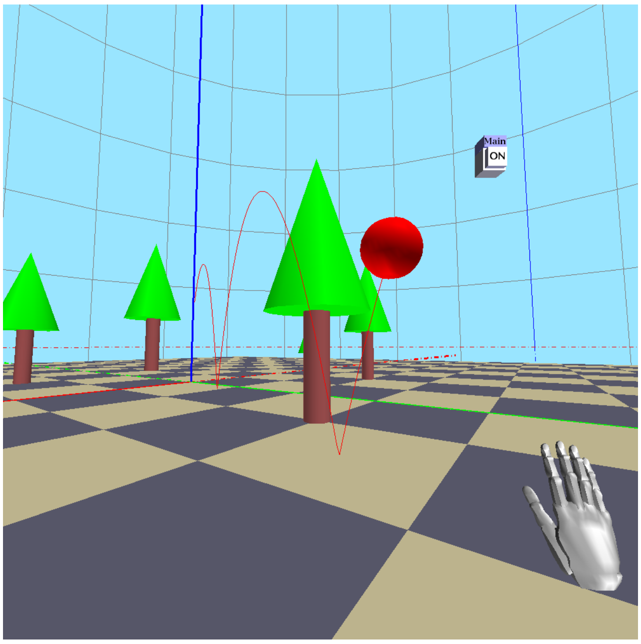
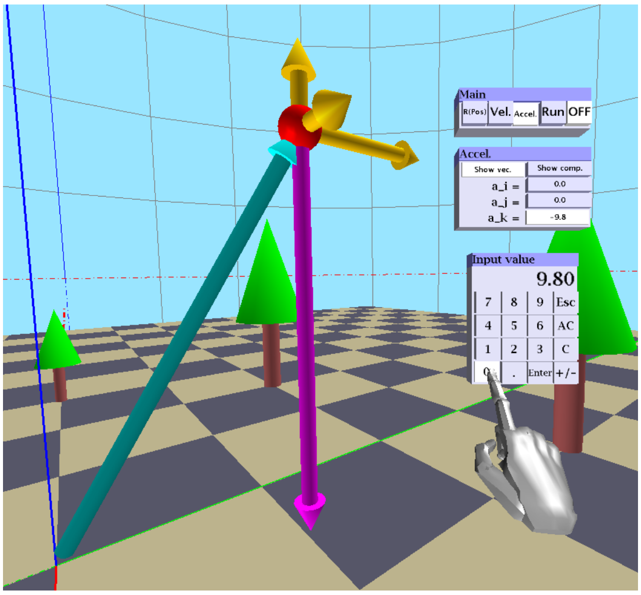


Fig. 6. Initial window of the virtual experiment (full screen, locale: English).



a_i = 0.0
a_j = 0.0
a_k = -9.8

9.80

Fig. 7. Setting parameters for the acceleration (part of screen, locale: English).

Two visible/invisible toggle buttons are prepared for each physical quantity. One is prepared to drawing the vector as a 3D arrow, and the other one for drawing three vector components. For example, the vectors are drawn for position and acceleration, and vector components represent velocity in the case of Fig. 7.

Figure 8 shows a menu for running a experiment. There are four buttons; two buttons have functions of

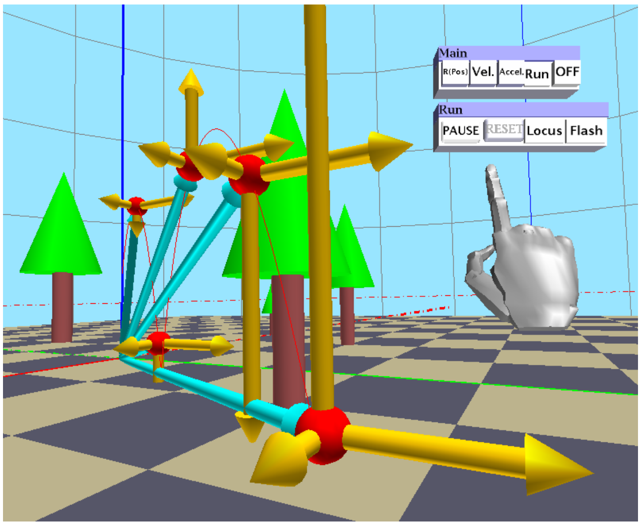


Fig. 8. Running virtual experiments in real-time (part of screen, locale: English).

start, pause, continue and reset and other two buttons are prepared for visible/invisible toggle switches named locus and flash. The locus of motion can be visualized as a tracing curve (locus button). The material point with vectors can be remains at intervals of the fixed time, like multi-flash photograph (flash button). For example the both toggle buttons are turned on in the case of Fig. 8.

Time scale is very important for many experiments and the real-time (time scale = 1) motion is effective for a free fall motion if possible. Such time scheduled motions can be implemented by whether thread in Java or Behavior class in Java3D API. We adopted the Behavior class because of its rich functionality. Not only the position of the material point but also vectors can move in real-time or correctly scheduled time scale.

H. Internationalization

All the strings on GUI components (3D buttons and labels etc.) in virtual environment are internationalized by the locale-sensitive objects and the Unicode system[10]. It means that every message is automatically shown in user's own language on 3D GUI component. Figure 3 shows the example of Japanese locale, and Figures 6, 7 and 8 show the example of English locale.

The Internationalization is implemented using the ResourceBundle class of Java2[5] with some files where the list of Unicode messages is written for each locale file[4], [11]. Integrated development environments are very useful to edit such files all together. Adding new locales and modifying internationalized strings can be easily done without recompiling the sources.

The property files are also useful to store default parameters; initial position, velocity, acceleration, flags and so on.

III. DISCUSSIONS

Fundamental controls for the experiment are realized in virtual environment by using the glove device and GUI components constructed by it3d. It is very convenient and important to continue experiments without going out to the real world in order to change parameters. Nevertheless, some user interfaces, for example the procedure to entering value, are complicated comparing to the general 2D GUI. The user interfaces should be improved to more suitable ones for virtual reality.

The rendering speed is sufficient with general 3D accelerated graphic boards for PCs and the real-time motion can be realized practically. It, however, causes another problem. When parameters are adjusted appropriately and the ball (the material point) are coming toward to the learners, they feel rather dangerous. The view point can be changed by the 3D mouse and it has a similar problem when move the view point quickly. The feeling depends on learners and, anyhow, we should be careful in this point to improve the system. At the beginning, we are very interested in attaching the viewpoint on the moving material because the impact can not be given in real experiments. It is not supported at present because of the same problem. The psychological effects are very important to be studied in parallel with such works.

TEELeX has a cubic display composed of six screens, nevertheless we use only a front screen for our system at present. It is still effective with one large screen and it seems easy to support plural displays, because new views can be added to our system's 3D world quite easily. Two or three displays may be supported by putting additional graphic boards to the Peripheral Component Interconnection (PCI) bus, however, supporting six displays may not be easy with one PC. Distributed system should be suitable and are in developing for TEELeX.

We adapted our system for virtual reality, keeping all of the original features. It therefore supports both web browsers and virtual environment systems as TEELeX. Learners can do the same experiment in both environments. When it runs as an applet on a web browser, several 3D views are attached to the single 3D virtual world, however, it does not support virtual reality. Large-scale virtual environment systems as TEELeX are not easy to access usually. There are several simple virtual environment systems; some kinds of stereo shutter glasses, stereo LCD displays, head mount displays and so on. We want to support them, nevertheless, there is a problem of input device in virtual environments. Glove devices are rather expensive and we therefore started to write a device emulator for the glove devices in order to support virtual environment systems without glove devices.

IV. CONCLUSIONS

We improved our 3D virtual experiment system for dynamics and adapted it for the immersive projection display and hand manipulation, keeping all of the original features. The system realizes virtual experiment in the virtual environment system by using TEELeX and it3d library. Learners can face an experience of full-scale and real-time experiment with user interfaces in the virtual environment. To realize the system, we did several technical improvements; developping the classes for virtual world with enhanced portability, connecting general Java3D's virtual world to it3d's one, and constructing internationalized control panels for experiments within the virtual world. The rendering speed is practically sufficient for real-time experiments with general 3D accelerated graphic boards for PCs, however, we should take account of psychological effects for learners in some cases, for example, ball are coming toward to them.

The programs and their source codes are opened at the web page[12].

Acknowledgments

This study was supported by a joint research programs at National Institute of Multimedia Education in Japan. The authors thank Mr. Masayuki Morita for his superb assistance.

References

[1] F-K. Hwang, Virtual Physics Laboratory: `http://www.phy.ntnu.edu.tw/java/`

[2] C. R. Wie, *Educational Java applets in solid state materials*, IEEE Trans. Educ., Vol.41, No.4, 1998, pp.354.

[3] H. Sowizral, K. Rushforth and M. Deering, *"The Java^{TM} API Specification"*, Addison-Wesley, 1998. `http://java.sun.com/products/java-media/3D/index.html`

[4] H. Nakano, K. Tokunaga and Y. Nakamura, *3D Visualization Effect on Virtual Experiments in Dynamics*, Proceedings Information Technology Based Higher Education and Training (ITEHT2002), Budapest, Hungary, 2002.

[5] M. Campione, K. Walrath, A. Huml and Tutuorial Team, *The Java Tutorial Continued: The Rest of the JDK*, Addison-Wesley, 1998.

[6] K. Walrath and M. Campione, *The JFC Swing Tutorial: A Guide to Constructing GUIs*, Addison-Wesley, 1999. `http://java.sun.com/products/jfc/index.html`

[7] K. Asai, N. Osawa, and Y. Y. Sugimoto, *Virtual Environment System on Distance Education*, Proceedings EUROMEDIA'99, pp.242-246 (1999). `http://www.nime.ac.jp/teelex-w/index-e.html`

[8] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, *Surround-screen projection-based virtual reality: The design and implementation of the CAVE*, Proceedings ACM SIGGRAPH '93, pp.135-142 (1993). `http://www.evl.uic.edu/pape/CAVE/`

[9] N. Osawa, K. Asai and F. Saito, *An Interactive Toolkit Library for 3D Applications: it3d*, Proceedings Eighth Eurographics Workshop on Virtual Environments, (EGVE2002), pp.149-157, (2002). `http://www.it3d.org/`

[10] J. Aliprand et al., *The Unicode Standard, Version 3.0*, The Unicode Consortium (Editor), Addison-Wesley, 2000. `http://www.unicode.org/`

[11] H.Nakano, T.Mizutani, Y.Nakamura and S.Matsuo, *Interactive Distance Education System for Real Experiments over the Web*, Proceedings International Conference on Intelligent Multimedia and Distance Education (ICIMADE2001), Fargo, ND, USA, 2001, pp.50-56.

[12] `http://cae.cc.kumamoto-u.ac.jp/products/Ph2/`