34

Collaborative Tele-Haptic Application and Its Experiments

Qonita M. Shahab, Maria N. Mayangsari and Yong-Moo Kwon
Korea Institute of Science & Technology, Korea

1. Introduction

Through haptic devices, users can feel partner's force each other in collaborative applications. The sharing of touch sensation makes network collaborations more efficiently achievable tasks compared to the applications in which only audiovisual information is used. In view of collaboration support, the haptic modality can provide very useful information to collaborators.

This chapter introduces collaborative manipulation of shared object through network. This system is designed for supporting collaborative interaction in virtual environment, so that people in different places can work on one object together concurrently through the network. Here, the haptic device is used for force-feedback to each user during the collaborative manipulation of shared object. Moreover, the object manipulation is occurred in physics-based virtual environment so the physics laws influence our collaborative manipulation algorithm. As a game-like application, users construct a virtual dollhouse together using virtual building blocks in virtual environment. While users move one shared-object (building block) to desired direction together, the haptic devices are used for applying each user's force and direction. The basic collaboration algorithm on shared object and its system implementation are described. The performance evaluations of the implemented system are provided under several conditions. The system performance comparison with the case of non-haptic device collaboration shows the effect of haptic device on collaborative object manipulation.

2. Collaborative manipulation of shared object

2.1 Overview

In recent years, there is an increasing use of Virtual Reality (VR) technology for the purpose of immersing human into Virtual Environment (VE). These are followed by the development of supporting hardware and software tools such as display and interaction hardware, physics-simulation library, for the sake of more realistic experience using more comfortable hardware.

Our focus of study is on real-time manipulating object by multiple users in Collaborative Virtual Environment (CVE). The object manipulation is occurred in physic-based virtual environment so the physic laws implemented in this environment influence our manipulation algorithm.

We build Virtual Dollhouse as our simulation application where user will construct a dollhouse together. In this dollhouse, collaborative users can also observe physics law while constructing a dollhouse together using existing building blocks, under gravity effects. While users collaborate to move one shared-object (block) to desired direction, the shared-object is manipulated, for example using velocity calculation. This calculation is used because current available physic-law library has not been provided for collaboration. The main problem that we address is how to manipulate **a** same object by two users and more, which means how we combine two or more attributes of each user to get one destination. We call this approach as shared-object manipulation approach.

This section presents the approach we use in study about the collaborative interaction in virtual environment so people in different places can work on one object together concurrently.

2.2 Related Work

In Collaborative Virtual Environment (CVE), multiple users can work together by interacting with the virtual objects in the VE. Several researches have been done on collaboration interaction techniques between users in CVE. (Margery, D., Arnaldi, B., Plouzeau, N. 1999) defined three levels of collaboration cases. Collaboration level 1 is where users can feel each other's presence in the VE, e.g. by representation of avatars such as performed by NICE Project (Johnson, A., Roussos, M., Leigh, J. 1998). Collaboration level 2 is where users can manipulate scene constraints individually. Collaboration level 3 is where users manipulate the same object together. Another classification of collaboration is by Wolff et al. (Wolff, R., Roberts, D.J., Otto, O. June 2004) where they divided collaboration on a same object into sequential and concurrent manipulations. The concurrent manipulation consists of manipulation of distinct and same object's attributes.

Collaboration on the same object has been focused by other research (Ruddle, R.A., Savage, J.C.D., Jones, D.M. Dec. 2002), where collaboration tasks are classified into symmetric and asymmetric manipulation of objects. Asymmetric manipulation is where users manipulate a virtual object by substantially different actions, while symmetric manipulation is where users should manipulate in exactly the same way for the object to react or move.

2.3 Our Research Issues

In this research, we built an application called Virtual Dollhouse. In Virtual Dollhouse, collaboration cases are identified as two types: 1) combined inputs handling or same attribute manipulation, and 2) independent inputs handling or distinct attribute manipulation. For the first case, we use a symmetric manipulation model where the option is using common component of users' actions in order to produce the object's reactions or movements. According to Wolff et al. (Wolff, R., Roberts, D.J., Otto, O. June 2004) where events traffic during object manipulations is studied, the manipulation on the same object's attribute generated the most events. Thus, we can focus our study on manipulation on the same object's attribute or manipulation where object's reaction depends on combined inputs from the collaborating users.

We address two research issues while studying manipulation on the same object's attribute. Based on the research by Basdogan et al. (Basdogan, C., Ho, C., Srinivasan, M.A., Slater, M. Dec. 2000), we address the first issue in our research: the effects of using haptics on a collaborative interaction. Based on the research by Roberts et al. (Roberts, D., Wolff, R., Otto, O. 2005), we address the second issue in our research: the possibilities of collaboration between users from different environments.

To address the first issue, we tested the Virtual Dollhouse application of different versions: without haptics functionality and with haptics functionality, to be compared. As suggested by Kim et al. (Kim, J., Kim, H., Tay, B.K., Muniyandi, M., Srinivasan, M.A., Jordan, J., Mortensen, J., Oliveira, M., Slater, M. 2004), we also test this comparison over the Internet, not just over LAN. To address the second issue, we test the Virtual Dollhouse application between user of non-immersive display and immersive display environments. We analyze the usefulness of immersive display environment as suggested by Otto et al. (Otto, O., Roberts, D., Wolff, R. June 2006), as they said that it holds the key for effective remote collaboration.

2.4 Taxonomy of Collaboration

The taxonomy, as shown in Figure 1, starts with a category of objects: manipulation of distinct objects and a same object. In many CVE applications (Johnson, A., Roussos, M., Leigh, J. 1998), users collaborate by manipulating the distinct objects. For manipulating the same object, sequential manipulation also exists in many CVE applications. For example, in a CVE scene, each user moves one object, and then they take turn in moving the other objects.

Concurrent manipulation of objects has been demonstrated in related work (Wolff, R., Roberts, D.J., Otto, O. June 2004) by moving a heavy object together. In concurrent manipulation of objects, users can manipulate in category of attributes: same attribute or distinct attributes.

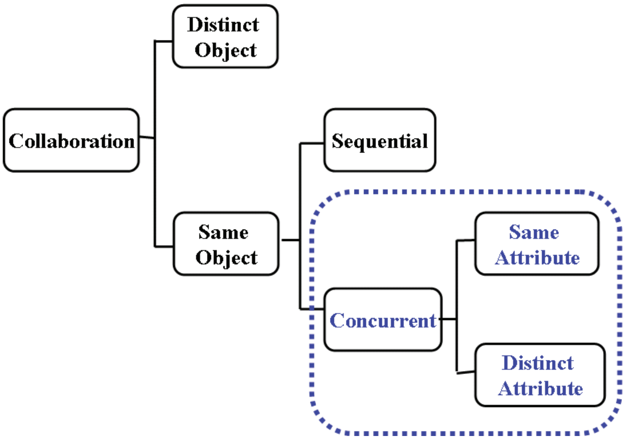


Fig. 1. Taxonomy of collaboration

2.5 Demo Scenario-Virtual Dollhouse

We construct Virtual Dollhouse application in order to demonstrate concurrent object manipulation. Concurrent manipulation is when more than one user wants to manipulate the object together, e.g. lifting a block together. The users are presented with several building blocks, a hammer, and several nails. In this application, two users have to work together to build a doll house.

The scenario for the first collaboration case is when two users want to move a building block together, so that both of them need to manipulate the "position" attribute of the block, as seen in Figure 2(a). We call this case as SOSA (Same Object Same Attribute). The scenario for the second collaboration case is when one user is holding a building block (keep the "position" attribute to be constant) and the other is fixing the block to another block (set the "set fixed" or "release from gravity" attribute to be true), as seen in Figure 2(b). We call this case as SODA (Same Object Different Attribute).

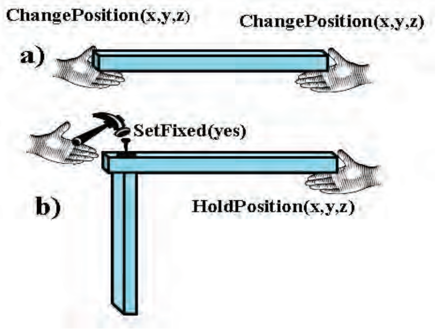
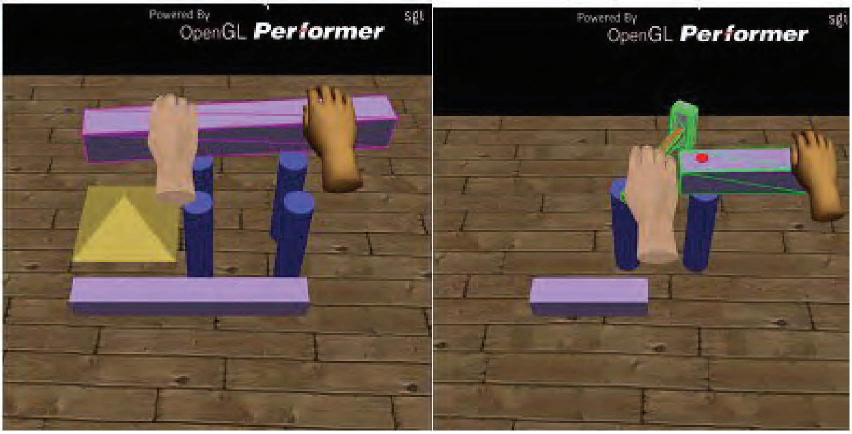


Fig. 2. (a) Same attribute, (b) Distinct attributes in Same Object manipulation

Figure 3 shows the demo content implementation of SOSA and SODA with blocks, hands, nail and hammer models.



(a) SOSA (b) SODA

Fig. 3. Demo content implementation

2.6 Problem and Solution

Even though physics-simulation library has been provided, there is no library that can handle physical collaboration. For example, we need to calculate the force of object that pushed by two hands.

In our Virtual Dollhouse, user will try to lift the block and another user will also try to lift the same block and move it together to destination.

After the object reaches shared-selected or "shared-grabbed" status, the input values from two hands should be managed for the purpose of object manipulation. We created a vHand variable as a value of fixed distance between the grabbing hand and the object itself. This is useful for moving the object by following the hand's movement.

We encountered a problem of two hands that may have the same power from each of its user. For example, a user wants to move to the left, and the other wants to move to the right. Without specific management, the object manipulation may not be successful. Therefore, we decided that users can make an agreement prior to the collaboration, in order to configure (in XML), which user has the stronger hand (handPow) than the other. Therefore, the arbitration of two input values is as following (for x-coordinate movement case):

```
Diff = (handPos1-vHand1) - (handPos2-vHand2)
If abs(handPow2) > abs(handPow1)
Hand1.setPos(hand1.x-diff,hand1.y,hand1.z)
Else if abs(handPow1) > abs(handPow2)
Hand1.setPos(hand2.x+diff,hand2.y,hand2.z)
```

After managing the two hand inputs, the result of the input processing is released as the manipulation result.

Our application supports 6DOF (Degree Of Freedom) movement: X-Y-Z and Heading-Pitch-Roll, but due to capability of our input device, we did not consider Pitch and Roll as necessary to be implemented graphically.

```
X-Y-Z = (handPos1-vHand1 + handPos2-vHand2)/2
```

In Figure 4, the angle is the heading rotation (between X and Y coordinates). The tangent is calculated so that the angle in degree can be found.

```
tanA = (hand0.y-hand1.y)/(hand0.x-hand1.x)
heading = atan(tanA)*180/PI
```

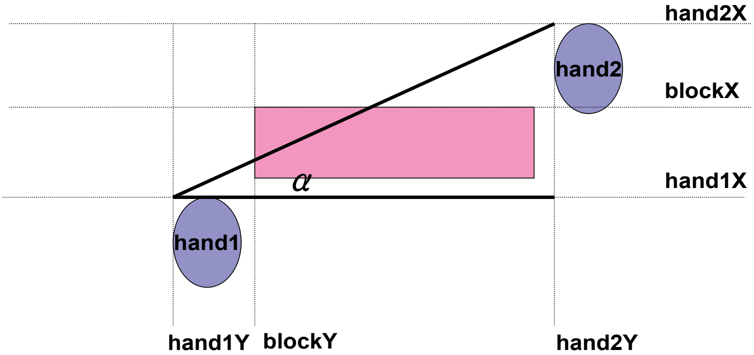


Fig. 4. Orientation of object based on hands positions

The final result of manipulation by two hands can be summarized by the new position and rotation as follows:

Object.setPos(X-Y-Z)

Object.setRot(initOri.x+heading, initOri.y, initOri.z)

Based on two user manipulation, three users manipulation can be calculated easily following the same algorithm. We have to choose which two hands against the other one hand (see Figure 5) based on hand velocity checking.

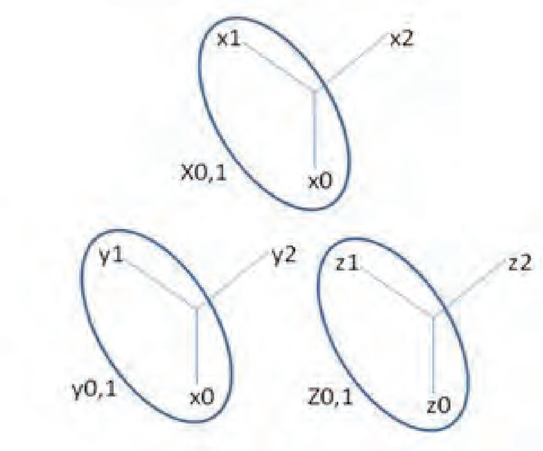


Fig. 5. Example of three users manipulation, Hand 0 and Hand 1 against Hand 2

After calculation, manipulation that made when three hands want to move an object together can be found below.

```
For each x, y, and z direction, check:
If abs(vel_hand0) >= abs(vel_hand1 + vel_hand 2)
Hand1 and hand2 follow hand0
Else if abs(vel_hand1) >= abs(vel_hand0 + vel_hand 2)
Hand0 and hand2 follow hand1
Else if abs(vel_hand2) >= abs(vel_hand0 + vel_hand 1)
Hand0 and hand1 follow hand2
```

2.7 Design of Implementation

(1) Virtual Dollhouse

We have built Virtual Dollhouse as our CVE. Our Virtual Dollhouse application is made based on OpenGL Performer (Silicon Graphics Inc. 2005) and programmed in C/C++ language in Microsoft Windows environment. VRPN server (Taylor, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., Helser, A.T. 2001) is used to provide management of networked joysticks to work with the VR application. We use NAVER Library (Park, C., Ko, H.D., Kim, T. 2003), a middleware used for managing several VR tasks such as device and network connections, events management, specific modeling, shared state management, etc.

The physics engine in our implementation is an adaptation of AGEIA PhysX SDK (AGEIA: AGEIA PhysX SDK) to work with SGI OpenGL Performer's space and coordinate systems. This physics engine has a shared-state management so that two or more collaborating computers can have identical physics simulation states. Using this physics engine, object's velocity during interaction can be captured to be sent as force-feedbacks to the hands that are grabbing the objects.

The architecture of our implementation can be seen in Figure 7.

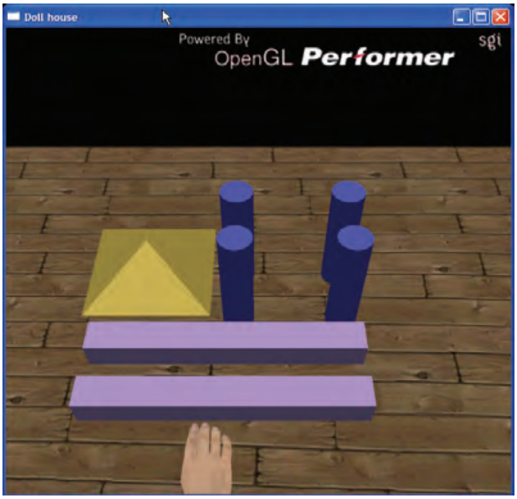


Fig. 6. Virtual Dollhouse as VCE

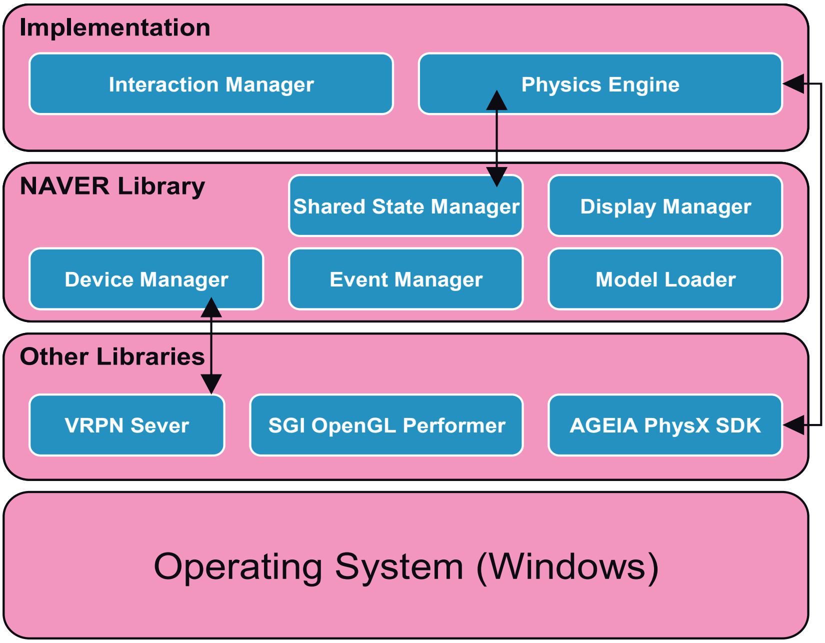
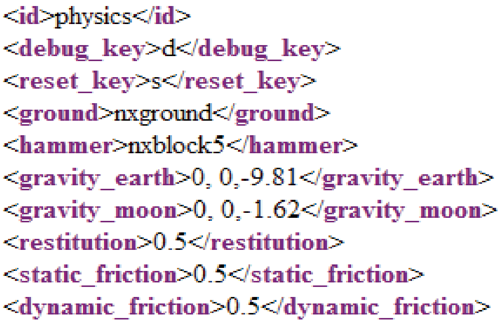


Fig. 7. System architecture of the implementation

To enable easy XML configuration, the application is implemented in a modular way into separate DLL (Windows' dynamic library) files. Using pfvViewer, a module loader from SGI OpenGL Performer, the dynamic libraries are executed to work together into one single VR application. All configurations of the modules are written in an XML file (with .pfv extension). The modules can accept parameters from what are written in the XML file, such as described in this figure below.

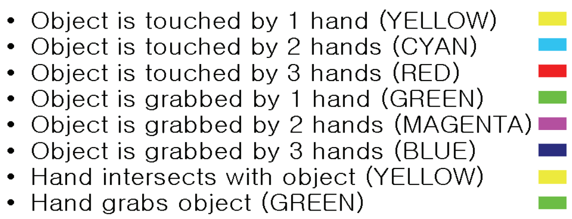


```
<id>physics</id>
<debug_key>d</debug_key>
<reset_key>s</reset_key>
<ground>nxground</ground>
<hammer>nxblock5</hammer>
<gravity_earth>0, 0,-9.81</gravity_earth>
<gravity_moon>0, 0,-1.62</gravity_moon>
<restitution>0.5</restitution>
<static_friction>0.5</static_friction>
<dynamic_friction>0.5</dynamic_friction>
```

Fig. 8. Configuration of physics simulation in XML file

(2) Three-Users Design and Implementation

Interaction status on the same object by three users is shared by showing several different states. These states are touched and selected by one, two, or three users. For user’s graphical feedback purpose, these states are described by color yellow, cyan, green, magenta, red, and blue respectively (Figure 9).

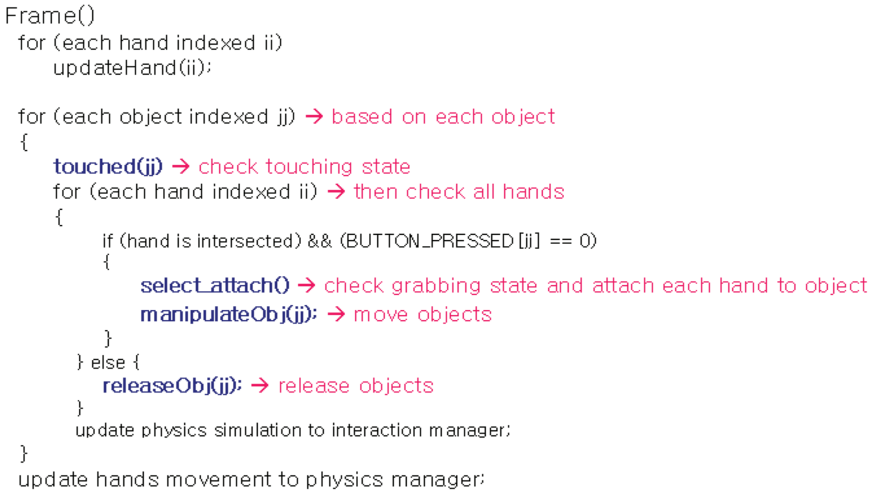


- Object is touched by 1 hand (YELLOW) 🟨
- Object is touched by 2 hands (CYAN) 🟦
- Object is touched by 3 hands (RED) 🟥
- Object is grabbed by 1 hand (GREEN) 🟩
- Object is grabbed by 2 hands (MAGENTA) 🟪
- Object is grabbed by 3 hands (BLUE) 🟦
- Hand intersects with object (YELLOW) 🟨
- Hand grabs object (GREEN) 🟩

Fig. 9. Graphical feedback for three-users

Each user is represented by one hand avatar. We modify our previous algorithm in order to check all these "touch" and "select" status easier. We check object status instead of hand status that we used in our previous algorithm. "Select" status only can happen after "touch" status. In a frame, we will check the touching status for each object and define how many hand and which hand that touches the object. Still in the same looping of each object, we check the selecting status of that object and doing manipulation for that object based on how many hand selects that object.

We made our application fit with Joystick and SPIDAR (Sato, M. 2002) - WAND input. These devices will be used in our testing to give input to our simulation. BUTTON_PRESSED in the figure below represents the "selecting or grabbing" button from Joystick or WAND button.



```
Frame()
 for (each hand indexed ii)
    updateHand(ii);

 for (each object indexed jj) → based on each object
 {
    touched(jj) → check touching state
    for (each hand indexed ii) → then check all hands
    {
        if (hand is intersected) && (BUTTON_PRESSED [ii] == 0)
        {
           select_attach() → check grabbing state and attach each hand to object
           manipulateObj(jj); → move objects
        }
      } else {
        releaseObj(jj); → release objects
      }
      update physics simulation to interaction manager;
 }
 update hands movement to physics manager;
```

Fig. 10. Algorithm of the object and hand status

The algorithm for shared-object manipulation is extended from two user manipulation into three user manipulation. The calculation of movement for three users is made based on two users's manipulation. The different is we have to choose which two hands against the other one hand based on hand velocity checking.

2.8 Results

As result of our approach, we present the comparative study of two users and the simulation result of three users.

We have done comparative study for two users. Two users manipulate the same object together concurrently in: 1) PC and PC environment through LAN inside KIST and the Internet between KIST,Korea and Oita University, Japan through APII-Hyunhae-Genkai Network, 2) CAVE (Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R.V., Hart, J.C. 1992) and PC environment through LAN. The test also includes the comparative study between haptic (with force feedback) and non-haptic (no force feedback) device. We will use joystick as input device for PC environment. In the CAVE system, the input devices that used are SPIDAR for movement and WAND for object selecting/grabbing button.

Table 1 shows our experiment result. We test five times and calculate average time for completing the collaborative interaction.

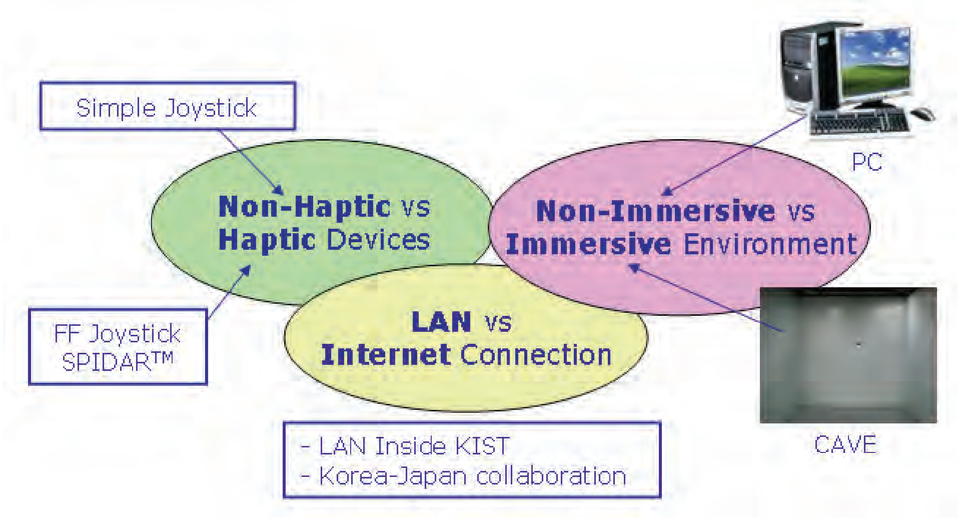

Simple Joystick
Non-Haptic vs Haptic Devices
FF Joystick SPIDAR™
PC
Non-Immersive vs Immersive Environment
LAN vs Internet Connection
CAVE
- LAN Inside KIST
- Korea-Japan collaboration

Fig. 11. Network Collaborative Interaction (NCI) Comparative Study

	PC-PC	PC-PC	CAVE-PC
	Non Force-Feedback	Force-Feedback	Force-Feedback
LAN Inside KIST	29.096s	21.344s	16.676 s
Internet (bw Korea and Japan)	43.55s	36.92s	-

Table 1. Comparison of network collaborative interaction in different immersion and different network environment

3. Summary

We have implemented an application for CVE based on VR systems and simulation of physics law. The system allows reconfiguration of the simulation elements so that users can see the effects of the different configurations. The network support enables users from different places to work together when interacting with the simulation, and see each other's simulation results.

From our series of testing of the application over different networks and environments, we can conclude that the use of haptics functionality (force-feedback device) is useful for users to feel each other's presence. It also helps collaboration to be performed more effectively (no time wasted). However, network delays caused problems on the haptics smoothness. In the future, we will update our algorithm by studying the possible solutions like indicated by Glencross et al. (Glencross, M., Jay, C., Feasel, J., Kohli, L., Whitton, M. 2007).

We also conclude that the use of tracker-type input device like SPIDAR is more intuitive for a task where users are faced with a set of objects to select and manipulate. From the display view of point, immersive display environment is more suitable for simulation of dealing with object manipulation that requires force and weight feeling, compared to non-immersive display environment such as PC.

Acknowledgment

This work was supported in part by KIST (Korea Institute of Science & Technology) through Development of Tangible Web Technology Project.

4. References

AGEIA: AGEIA PhysX SDK, http://www.ageia.com

Basdogan, C., Ho, C., Srinivasan, M.A., Slater, M. (Dec. 2000) An Experimental Study on the Role of Touch in Shared Virtual Environments. In: ACM Transactions on Computer Human Interaction, vol. 7, no. 4, pp. 443-460

Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R.V., Hart, J.C. (1992) The CAVE: audio visual experience automatic virtual environment. In: Communications of the ACM, vol. 35, issue 6, pp. 64-72

Glencross, M., Jay, C., Feasel, J., Kohli, L., Whitton, M. (2007) Effective Cooperative Haptic Interaction over the Internet. In: Proceedings of IEEE Virtual Reality Conference 2007. Charlotte

Johnson, A., Roussos, M., Leigh, J. (1998) The NICE Project: Learning Together in a Virtual World. In: IEEE Virtual Reality Annual International Symposium (VRAIS 98). Atlanta

Kim, J., Kim, H., Tay, B.K., Muniyandi, M., Srinivasan, M.A., Jordan, J., Mortensen, J., Oliveira, M., Slater, M. (2004) Transatlantic touch: A study of haptic collaboration over long distance. In: Presence: Teleoperator and Virtual Environments, vol. 13, no. 3, pp. 328-337

Margery, D., Arnaldi, B., Plouzeau, N. (1999) A General Framework for Cooperative Manipulation in Virtual Environments. In: 5th Eurographics Workshop on Virtual Environments. Vienna

Otto, O., Roberts, D., Wolff, R. (June 2006) A Review on Effective Closely-Coupled Collaboration using Immersive CVE's. In: Proceedings of ACM VRCIA. Hong Kong Roberts, D., Wolff, R., Otto, O. (2005) Supporting a Closely Coupled Task between a Distributed Team: Using Immersive Virtual Reality Technology. In: Computing and Informatics, vol. 24, no. 1

Park, C., Ko, H.D., Kim, T. (2003) NAVER: Networked and Augmented Virtual Environment aRchitecture; design and implementation of VR framework for Gyeongju VR Theater. In: Computers & Graphics, vol. 27, pp. 223-230

Ruddle, R.A., Savage, J.C.D., Jones, D.M. (Dec. 2002) Symmetric and Asymmetric Action Integration During Cooperative Object Manipulation in Virtual Environments. In: ACM Transactions on Computer-Human Interaction, vol. 9, no. 4

Sato, M. (2002) Development of string-based force display. In: Proceedings of the Eighth International Conference on Virtual Reality and Multimedia, Workshop 2. Gyeongju

Silicon Graphics Inc. (2005) , "OpenGL Performer," http://www.sgi.com/products/software/performer/

Taylor, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., Helser, A.T. (2001) VRPN: A device-independent, network-transparent VR peripheral system. In: ACM International Symposium on Virtual Reality Software and Technology (VRST 2001). Berkeley

Wolff, R., Roberts, D.J., Otto, O. (June 2004) A Study of Event Traffic during the Shared Manipulation of Objects within a Collaborative Virtual Environment. In: Presence, vol. 13, no. 3, pp. 251-262