An Integrated Haptic Data Transmission in Haptic Collaborative Virtual Environments

YongheeYou [1],MeeYoungSung [1],KyungkooJun [2]
[1]*Department of Computer Science & Engineering, University of Incheon*
[2]*Department of Multimedia Systems Engineering, University of Incheon*
{yhinfuture,mysung,kjun}@incheon.ac.kr

Abstract

Haptic Collaboration Virtual Environment (HCVE) is an enhanced virtual reality space with haptic interface support. HCVE users are connected together over the network and are able to work together by using sense of touch, i.e. haptics as well as audio and visual interfaces. In HCVE, the communication of haptic data is challenging because of time-varying network conditions and extremely high data rate. To mitigate such difficulties, we propose a linear prediction algorithm and a buffering scheme which is an integrated scheme for haptic data transmission. The prediction algorithm is to minimize the negative effects from network delay, loss and jitter, while the buffering scheme is to easily synchronize haptic interaction. For the evaluation of our proposed schemes, we build an experimental test bed for HCVE. As the result, we observe that our schemes are effective in improving the quality of haptic experiences. The quantitative measurement results are presented

1. Introduction

Haptic Collaboration Virtual Environment (HCVE) is a virtual reality space with haptic interface support. Users of HCVE cooperate each other over the network by using the haptic interfaces as well as audio and visual interfaces. In this section, the main idea of haptic rendering, main challenges of HCVE and related research are presented.

The goal of haptic rendering is to enable a user to touch, feel, and manipulate virtual objects through haptic interfaces as realistically as possible [1], [2]. A force-feedback device can generate kinesthetic information and temporal tactile information. By using these perceptual cues such as shape, stiffness, texture and friction, haptic rendering can render various properties of a virtual object. Unlike visual rendering, a minimum update rate to achieve realistic haptic rendering depends on the properties of virtual objects and a force-feedback device. Recommended update rates are 1 KHz and 5 KHz-10 KHz for a rigid surface and a textured surface respectively. For a transformable object, it is advised to keep the rate as fast as you can.

With improved reality and immersion experience, HCVE is particularly proper for educational simulations [3], [4]. Time-varying network conditions pose challenges to successful communication of haptic data [5]. Adverse network links sometimes cause irregular force-feedback which deteriorates the haptic experiences [6]. The transmission of the haptic data, which mainly consists of the position information of haptic device pointers and that of objects manipulated by haptic device pointers, is basically similar to the multimedia streaming. However, it is much more demanding because the haptic rendering rate required for satisfactory haptic experience is quite higher than that of graphic rendering; 1 KHz for haptics, in contrast to 30 Hz for graphics.

To meet such challenges, there have been various research efforts. For group synchronization control, Y. Ishibashi et al propose virtual time rendering algorithms [7]. Hikichi et al employ a queue monitoring algorithm [8] designed for efficient haptic collaboration. However, these approaches have limitations that they are not able to cope with delay, loss and jitter at the same time like an integrated haptic data transmission system..

In this paper, we propose an efficient scheme for the haptic data transmission to mitigate negative effects from delay, loss and jitter. The core idea is based on prediction and buffering. The prediction is to cope with loss and excessive delay employing extrapolation, while the buffering is to deal with moderate delay and jitter. To evaluate the proposed schemes, we build an experimental HCVE test bed with network emulator, NIST Net that enables us to simulate real network conditions and perform tests under different delay, loss and jitter conditions. After experiments of the performance of our proposed schemes, we conclude our system is efficient.

In section 2, we describe our haptic collaboration application. The details of our proposed schemes are presented in section 3, the experiment results are discussed in section 4, and section 5 concludes our work and discusses the future work.

6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007)
0-7695-2841-4/07 $25.00 © 2007 IEEE
IEEE COMPUTER SOCIETY

3. Haptic Collaboration Application

We describe in this section the development of the haptic collaboration application that enables human-to-human haptic interaction over the Internet.

2.1. Overview

Figure 1 shows a virtual 3D room for the haptic collaboration. The objective is that users work together to move the cube to where the sphere is by 'haptically' touching the cube with their probes. The probes, one probe for one user, are drawn as small balls in the figure and represent the haptic interface pointers. Users are able to lift, push, or rotate the cube if they manipulate the probes cooperatively: for example, to lift up the cube, at least two users should position their probes underneath the cube and move them upward without losing balance. Since all the users collaborate over the network, the network conditions can affect the quality of the collaboration.

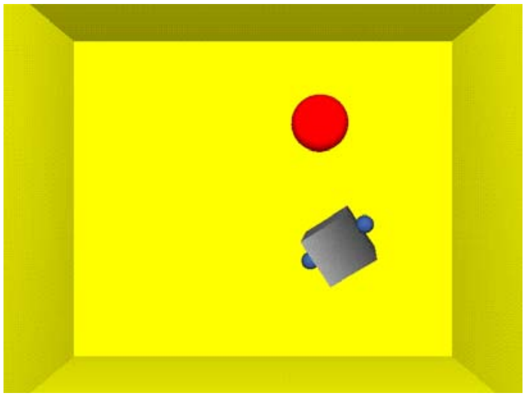

Figure 1. Snapshot of the haptic Collaboration application

2.2. Software Setup

The aforementioned haptic application depends on several software libraries as shown in Figure 2. QUANTA networking library is used to implement the haptic data transmission over UDP. For haptic rendering, we use OpenHaptics toolkit [9] which provides both Haptic Library API (HLAPI) and Haptic Device API (HDAPI). HLAPI is for rendering haptically static objects such as the room in Figure 1 and able to generate high quality force-feedback based on OpenGL frame-buffer. HDAPI is used for rendering dynamic objects such as the cube. Open Dynamics Engine (ODE) is also used to enhance the movement animation of the cube. The detailed information of Openhaptics and QUANTA Networking Toolkit is stated below.

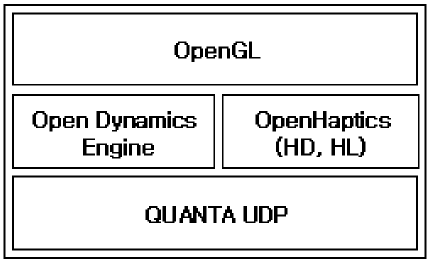

Figure 2. Software Libraries for haptic collaboration application implementation

2.2.1 OpenHaptics

Figure 3 illustrates the structure of the OpenHaptics Toolkit from SensAble [9] which is an application that enables software developers to add haptics and true 3D navigation to a broad range of applications, including 3D design and modeling. OpenHaptics is patterned after OpenGL® API, making it familiar to graphics programmers and facilitating integration with new or existing OpenGL. This toolkit handles complex calculations and provides low-level device control for advanced developers. The architecture of OpenHaptics Toolkit is shown below.

HDAPI (Haptic Device API) is a low-level foundational layer for haptics. It is best suited for developers who are familiar with haptic paradigms and sending forces directly. This includes those interested in haptics research, telepresence, and remote manipulations. HLAPI (Haptic Library API) is designed for high-level haptics scene rendering. It is targeted at developers who are less familiar with haptics programming, but desire to quickly and easily add haptics to graphics applications. Utilities include mathematical and necessary functions such as vector and matrix calculations that are used for haptic devices.

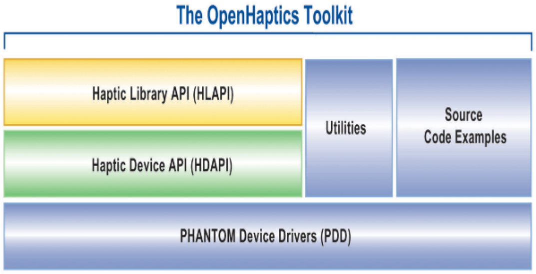

Figure 3. The Openhaptics Toolkit

2.2.2 QUANTA Networking Toolkit

QUANTA (The Quality of Service Adaptive Networking Toolkit) is a cross-platform adaptive networking toolkit for supporting the diverse networking requirements of latency-sensitive and bandwidth-intensive applications [10]. It provides Reflector

TCP/UDP, Parallel TCP and Reliable Blast UDP by using TCP and UDP. In addition, it supports the features such as IPv4, IPv6, thread and mutex. Since QUANTA inherits CAVERN (CAVE Automatic Virtual Environment Research Network) from CAVE (CAVE Automatic Virtual Environment) Systems, its structure is suitable for DVE (Distributed Virtual Environment) Systems. In this study, we developed a network module with the QUANTA Library.

2.3. Client and Server Architecture

The haptic collaboration is based on the client and server architecture as shown in Figure 4: clients send their own haptic data to a server, which in turn performs calculation necessary for the haptic rendering. The haptic data of clients, mainly haptic pointer positions, are obtained from haptic interfaces attached to client machines. With the received haptic position data, the server detects possible contacts between the client haptic pointer and the cube, and applies the spring-damper model to obtain positions and rotation angles of the cube. Then the server sends the information of the cube and the received haptic pointer positions to clients and clients calculate force-feedback values and display the scene.

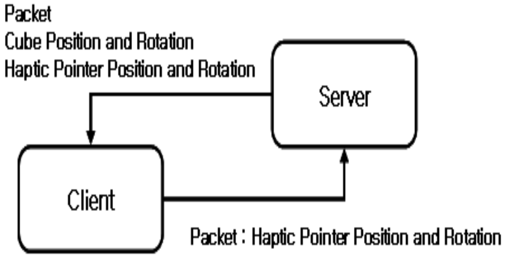

Figure 4. Client/Server Architecture

3. Haptic Data Transmission Scheme

In this section, we propose a scheme for the haptic data transmission. Considering that the haptic data is susceptible to the network conditions [1], the basic idea of our scheme is the well-balanced combination of extrapolation, buffering and synchronization as shown in Figure 5.

Before discussing our scheme in detail, we first introduce some of supporting modules to compose our scheme. Calculation Module (CM) of the server is responsible for the collision-detection and physics calculation, Network Modules (NMs) of the client and the server are for communicating the haptic data in UDP, Rendering Module (RM) of the client is for graphic rendering, and Haptic Input Module (HIM) of the client is for collecting data from the haptic interfaces.

Key modules to implement our proposed scheme are Prediction Module, Delay Synchronization Module, Buffer, and Delay Module. We discuss each of the modules in detail in the following subsections.

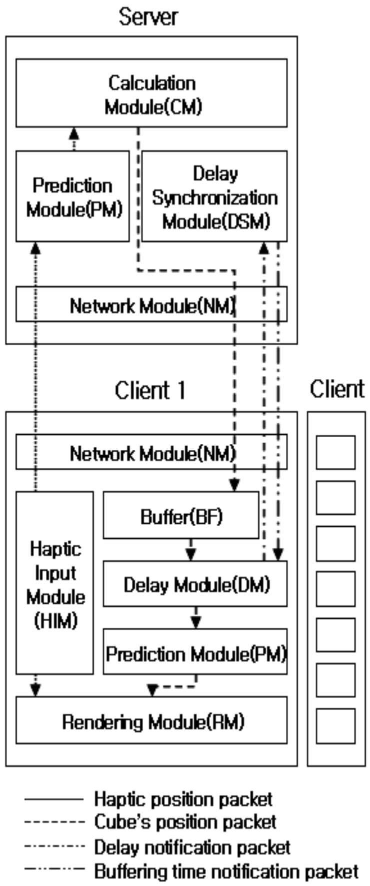

Figure 5. Scheme for Haptic Data Transmission

3.1. Prediction Module

Prediction Module (PM) is to compensate for packet loss and jitter. Since clients and a server transmit packets every 1 ms, PMs on both the clients and the server check whether packets arrive every 1 ms. If packet loss is detected, PM applies a linear algorithm to predict next positions of the haptic pointers and the cube. In the case of jitter, out-of-order packets are detected by packet sequence numbers and discarded. PM compensates for the discarded packets in the similar way to the lost packet compensation.

The position prediction is executed as follows:

$$X_n = X_{n-1} + V_{n-1} \quad (1)$$

$$V_{n-1} = X_{n-1} - X_{n-2} \quad (2)$$

where Xn is the predicted position while Xn-1 corresponds to the previous position, Vn-1 is the velocity of the cube or haptic pointer's position.

3.2. Delay Synchronization Module, Buffer and Delay Module

Delay harms the haptic collaboration. One example of the negative effects of delay is that, in the case of the application of Section 2, all clients see the same cube but all in different positions.

In order to cope with this spatial desynchronization problem, we propose Delay Synchronization Module (DSM) at the server side and Delay Module (DM) at the client side. The whole idea of DSM and DM for synchronization is that clients should put off their haptic rendering until all of them become ready to start the rendering simultaneously. For this, DSM decides buffering times for each client, i.e. how much time each connected client should wait before its rendering. Clients are required to report their measured delay to the server every 5 seconds and DSM calculates the buffering times for each client as follows.

$$M = MAX(D_1,D_2,D_3,\ldots,D_n) \quad (3)$$

$$R_n = M - D_n \quad (4)$$

where Di is the delay reported from client i, M is the maximum of Di, Ri is the buffering time for client i. For example, if M is 100 ms and D1 is 10 ms, the buffering time for client 1 becomes 90ms.

During the buffering time, clients store the received haptic packet data into their Buffer (BF). Since every packet has its sequence number, the packets in BF can re-ordered in sequence, thus eliminating jitter effect. When starting the haptic rendering with the data in BF, DM uses the following equations to calculate the haptic positions.

$$P_n = C - R_n \quad (R > 0) \quad (5)$$

$$CP_n = BF[I(P_n)] \quad (6)$$

where C denotes the current time in ms, R_n denotes the buffering time, P_n is the difference between the current time and the buffering time, $I(P_n)$ is the index of the buffer, and CP_n is the position data that will be rendered.

4. Experiments

We perform experiments to evaluate the efficiency of our proposed scheme. In this section, we describe a test bed, assessment methods and discuss results.

4.1. Test-Bed

The test bed is composed of three parts; a server, clients, and a network emulator, NISTNet, which is able to emulate a wide variety of network conditions [2]. The haptic device that we use for experiments is PHANToM Omni [9]. We set its rendering rate to 1 KHz. Other details are shown in Table 1.

Table 1. Test-bed Configuration

Server Computer	Client Computer
AMD Athlon TM	Sensable PHANToM Haptic device
64 Processor 3500+	Dell Precision PWS380 Intel®
2.21GHz 1.00GB	Pentium 4 CPU 3.20GHz, 1.00GB
RAM	RAM
OS: Microsoft	NVIDIA Quadro FX1400 Graphic
Windows XP	Card
Professional	OS: Microsoft Windows XP
Version 2002	Professional Version 2002 Service
Service Pack 2	Pack 2

4.2. Assessment Method

To evaluate the performance of PM, we compare the cube positions obtained from the clients with the predicted positions at the server under various network conditions of delay, loss, and jitter.

For DSM and DM, we record the cube positions at each client and compare them each other to see the differences. The expression used for evaluation is as follows:

$$e_i = \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2 + (z_i - z_i')^2} \quad (7)$$

where xi, yi, zi are the current position of the cube at client i , while xi`, yi`, zi` are the latest cube position calculated at the server. The smaller ei is, the more accurate our scheme is.

4.3. Experiments

In the first experiment, we evaluate PM under 15% packet loss condition. As shown in Figure 6 where the x axis indicates the elapsed time in millisecond and the y axis represents errors, our proposed scheme is more accurate than the case without PM. The average error of our scheme is 0.012 while 0.09 without PM.

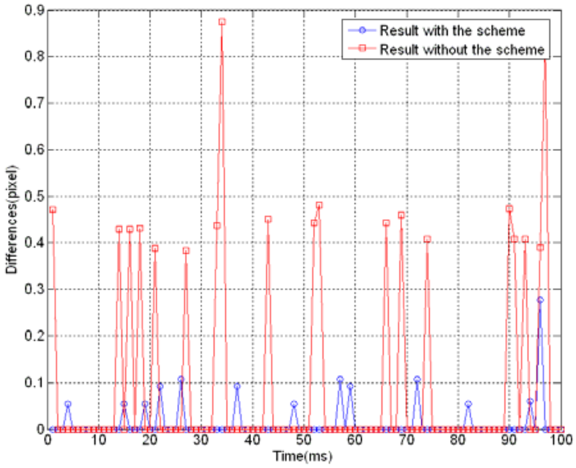

Figure 6. Haptic data transmission under 15% losses

In the second experiment, we evaluate PM under 50 ms jitter. As shown in Figure 7, our proposed scheme is still more accurate than the case without PM. The average error of our scheme is 0.102 while 0.863 without PM. One thing to note is that the average error in the case of loss is much smaller than in the case of jitter, thus it implies that PM works better for the packet loss.

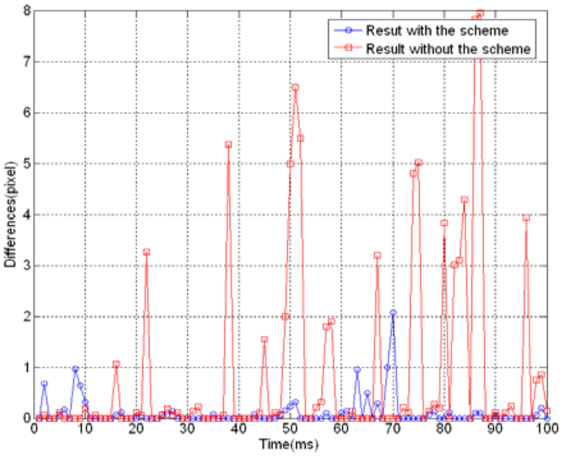

Figure 7. Haptic data transmission under 50ms delay

In the third experiment, we evaluate DSM and DM. We install one server and two clients, and apply 100 ms delay to the link between the server and one of the clients so that two clients are in asynchronous state. In Figure 8 and 9, the circle plotted lines are the results at the client's side under 100 ms delay and the square plotted lines represent the results under no delay. Also, Figure 8 and 9 are for without the scheme and with the scheme respectively, and the y axis represents the differences between the cube's positions at the server and at the client.

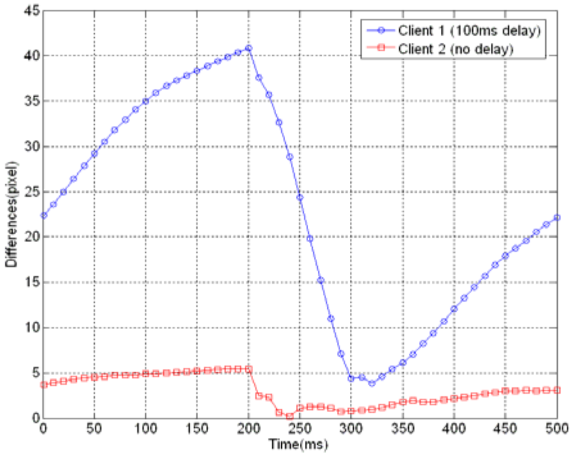

Figure 8. Haptic data transmission under 100ms delay without the scheme

As the results, we observe that our scheme is still able to achieve better performance than the case without the scheme. In Figure 8, the differences between the lines are quite large, thus the haptic collaboration is hardly achievable. However, in Figure 9 in which our scheme is activated, the differences are negligible.

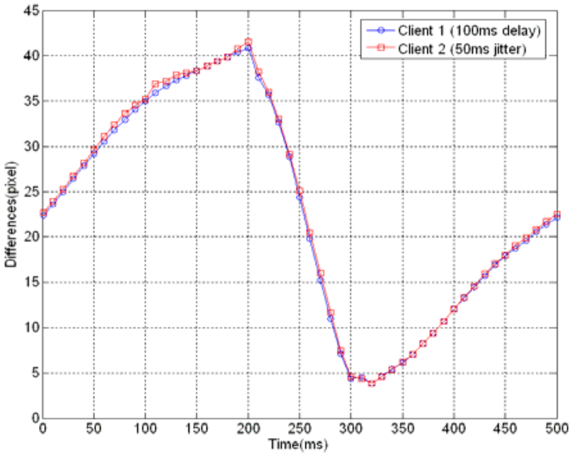

Figure 9. Haptic data transmission under 100ms delay with the scheme

5. Conclusion and Future Work

The objective of our study is to develop a haptic data communication scheme that contributes to the successful implementation of HCVE. For this, we propose the linear prediction and buffering schemes which are effective to mitigate the negative effects of delay, loss, and jitter conditions. In the experiments, we evaluate the performance of our proposed schemes and find that the results are promising.

For future works, we plan to adopt more elaborative interpolation methods for PM and search for other synchronization schemes. In addition, we need to develop quantitative evaluation methods for haptic collaboration which has more qualitative and subjective characteristics.

As another future research topic, we will work on the QoS algorithms which cope with significant delay, jitter and loss in haptic data transmission [11], [12], [13], [14], [15], [16], [17], [18], [19]. We will also investigate some of the adaptive transmission protocols for haptic interactions. In addition, we need to assess the subjective quality of the haptic interactions in an objective way. Later, we need to study the inter-client synchronization problem in haptic-based CVEs [20], [21], [22], [23], [24], in order to allow consistent collaboration among many participants.

6. Acknowledgement

This work was supported by the Brain Korea 21 Project in 2007, by grant No. RTI05-03-01 from the RegionalTechnologyInnovationProgramoftheMinistry ofcommerce,IndustryandEnergy(MOCIE).

7. References

[1] A. Boukerche, S. Shirmohammadi and A. Hossain "A Prediction Algorithm for Haptic Collaboration", *Proceedings of HAVE 2005*,IEEE,2005,pp.154-158

[2]NISTNet:http://wwwx.antd.nist.gov/nistnet/

[3] M. A. Srinivasan, and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges", *Computers and Graphics*,Elsevier,1997,pp.393-404.

[4]Hosseini,Mojtabeetal.,"AHapticVirtualEnvironmentfor Industrial Training" *In Proceedings of HAVE 2002 IEEE International Workshop on Haptic Audio Visual Environments and their Applications*,IEEE,Ontario Canada 2002,pp.25-30.

[6]. M. O. Alhalabi, S. Horiguchi, and S. Kunifuji, "An experimental study on the effects of network delay in cooperative shared haptic virtual environment", *Computer and Graphics*,Elsevier,2003,vol.27,pp.205-213,

[5]. K. Hikichi et al.,"Architecture of Haptics Communication System for Adaption to Network Environments", *IEEE International Conference on Multimedia and Expo Proceedings*,IEEE,2001,pp.744-747

[7]. Y. Ishibashi, T. Hasegawa, and S. Tasaka, "Group Synchronization Control for Haptic Media in Networked Virtual Environments", *Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE, 2004, pp. 106-113

[8]. D. L. Stone and K. Jeffay, "An empirical study of delay jitter management policies", *ACM Multimedia Systems*, ACM,1995,Vol.2,pp.267-279

[9]PHANTOMOmnihttp://www.sensable.com/

[10]QUANTA,http://www.evl.uic.edu/cavern/quanta/

[11] M. Fujimoto, Y. Ishibashi, "Packetization Interval of Haptic Media in Networked Virtual Environments", *ACM NetGames05*,ACM,2005

[12] K. Hikichi, H. Morino, Y. Yasuda, I. Arimoto, and K. Sezaki, "The evaluation of adaptive control for haptics collaborationovertheinternet", *Proc. CQR (Communication Quality & Reliability) International Workshop*, May 2002, 218-222.

[13] K. Hikichi, H. Morino, I. Arimoto, I. Fukuda, S. Matsumoto, M. Iijima, K. Sezaki, and Y. Yasuda, "Architecture of haptics communication system for adaptation to network environments", *Proc. IEEE ICME*, IEEE,Aug2001,pp.563-566

[14]S.Matsumoto,I.Fukuda,H. Morino,K.Hikichi,K.Sezaki, and Y. Yasuda "The influences of network issues on haptic collaboration in shared virtual environments", *Proc. 5th PHANToM Users Group*,2000

[15]J.Marsh,M.Glencross,S.Pettifer,R.Hubbold,J.Cook,S. Daubrebet,"Minimisinglatencyandmaintainingconsistency in distributed virtual prototyping", *ACM SIGGRAPH Conference on the Virtual Reality Continuum and its Applications in Industry (VRCAI)*, ACM, Singapore, 2004, pp.386–389

[16] J. Marsh, M. Glencross, S. Pettifer, R. Hubbold "A robust network architecture supporting rich behaviour in collaborativeinteractiveapplications", *IEEE Transactions on Visualisation and Computer Graphics TVCG 12*, IEEE, March2006,pp.405–416.

[17] M. O. Alhalabi, S. Horiguchi, "Network latency issue in cooperative shared haptic virtual environment", *SPIE Third International Conference on Virtual Reality and Its Application in Industry*,SPIE,2003,vol.4756pp.199–205.

[18]D.Wang,K.Tuer,M.Rossi, L.NI,J.Shu,"Theeffectof time delays on tele-haptics", *Second IEEE Internatioal Workshop on Haptic, Audio and Visual Environments and Their Applications HAVE*,IEEE,2003,pp.7–12.

[19] R. S. Allison, J. E. Zacher, D. WANG, J. Shu "Effects of network delay on a collaborative motor task with telehaptic and televisual feedback", *ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry*, ACM Press, Singapore, 2004, pp. 375–381.

[20] B. Hannaford, J.-H. Ryu, Y. S. Kim "Stable control of haptics. In Touch in Virtual Environments",*McLaughlin M. L., Hespanha J. P., Sukhatme G. S., (Eds.).* Prentice Hall PTR,UpperSaddleRiver,NJ,2002,pp.47–70.

[21] C. Gunn, M. Hutchins, D. Stevenson, M. Adcock "Using collaborative haptics in remote surgical training" *WorldHAPTICS First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE Computer Society, Pisa, Italy,March2005,pp.481–482.

[22] J. P Hespanha, M. Mclaughin, G. S. Sukhatme, M. Akbarian, R. Garg, W. Zhu, "Haptic collaboration over the internet", *The Fifth PHANTOM Users Group Workshop*, 2000

[23]M.Mclaughlin,G.Sukhatime,W.Peng,W.Zhu,J.Parks, "Performance and co-presence in heterogeneous haptic collaboration", *IEEE Eleventh Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS)*,IEEE,2003

[24] X. Shen, F. Bogsanyi, L. NI, N. D. Georganas "A heterogeneous scalable architecture for collaborative haptics environments", *IEEE International Workshop on Haptic Audio and Visual Environments (HAVE)*, IEEE, 2003, pp. 113–118