A Comparison of Output Quality among Haptic Media Synchronization Algorithms

Shuchun Sun, Takeshi Fujimoto, Yutaka Ishibashi, and Shinji Sugawara
Graduate School of Engineering,
Nagoya Institute of Technology
Nagoya 466-8555, Japan
{shuchun, fujimoto}@mcl.nitech.ac.jp, {ishibasi, shinji}@nitech.ac.jp

Abstract

In this paper, we make an output quality comparison of haptic media synchronization algorithms in work where a user lifts and moves a virtual object and in a remote haptic drawing system. As a haptic interface device, we use PHANToM DESKTOP or SPIDAR-G AHS in the former work and PHANToM Omni in the latter. We handle five haptic media synchronization algorithms: Virtual-Time Rendering (VTR), Buffering, Skipping, Queue Monitoring (QM), and the adaptive buffer control. By subjective assessment, we clarify the quantitative relationships among the algorithms. Assessment results show that VTR is the most effective.

1. Introduction

Because of rapid growth of haptic interface devices, a number of researchers pay attention to networked virtual environments using haptic media [1]. In the environments, there are many applications such as networked real-time games [2], collaborative work [3], and remote haptic drawing systems [4]. Since a user can touch an object in a 3-D virtual space based on computer graphics (CG) with a haptic interface device, we can largely improve the efficiency of collaborative work and immerse ourselves in playing networked real-time games. However, when we do these types of work through a network, network delay jitter disturbs the temporal relations of haptic media; thus, the output quality of haptic media may seriously be degraded. Therefore, when we deal with haptic media transmission, it is necessary to maintain the temporal relations of haptic media by carrying out media synchronization control [4]-[8].

In [4], the authors adopt the *Virtual-Time Rendering* (*VTR*) and *Skipping* for media synchronization control and investigate the influence of network delay jitter on a remote haptic drawing system in which an instructor teaches a learner how to draw a figure while the instructor and learner are feeling the sense of force interactively. Also, in [5], the authors deal with VTR, *Buffering*, and Skipping as haptic media synchronization algorithms and examine the influence of the network load on the output quality in work where a user lifts and moves a virtual object with a haptic interface device. In [6], a media synchronization algorithm called *Queue Monitoring* (*QM*) [9] is used in networked collaborative work where two users lifts and moves a virtual object cooperatively, and the influence of network delay jitter on the output quality of haptic media is investigated by subjective assessment. Furthermore, in [7] and [8], the *adaptive buffer control* (called *ABC* in this paper) algorithm for haptic media synchronization is proposed for remote surgery. However, the quantitative relationships among the above algorithms have not been clarified so far. To clarify ranges of network delay and delay jitter in which the algorithms work effectively, we need to compare the output quality of the algorithms.

In this paper, we handle VTR, Buffering, Skipping, QM, and ABC as haptic media synchronization algorithms. We deal with two types of work to investigate the influences of network delay and delay jitter. One is work in which a user lifts and moves a virtual object [5] (this work is employed in a networked real-time game [10]), and the other is a remote haptic drawing system [4]. Then, we compare the output quality of the five haptic media synchronization algorithms by subjective assessment.

The rest of this paper is organized as follows. Section 2 describes the two types of work adopted in the paper. Section 3 outlines experimental systems. Section 4 explains the five haptic media synchronization algorithms. Section 5 describes assessment methods, and assessment results are presented in Section 6. Section 7 concludes the paper.

ICAT 2008
Dec. 1-3, Yokohama, Japan
ISSN: 1345-127

2. Work Descriptions

In this paper, we adopt two types of work with largely-different characteristics. One is work in which a user lifts and moves a virtual object (called *work 1* in this paper), and the other is a remote haptic drawing system (called *work 2*). In work 1, a user does the work with PHANToM DESKTOP [11], [12] or SPIDAR-G AHS [13] based on a client-server model. Work 2, in which two users do the remote haptic drawing instruction work with PHANToM Omni [11], [14], is based on a Peer to Peer (P2P) model.

In the two types of work, each terminal transmits *media units* (*MUs*), each of which is the information unit for media synchronization control. Each MU includes position information of a haptic interface device. The transmission rate of MUs is 1 kHz in work 1 [12] and 30 Hz in work 2 [4]. In what follows, we explain the details of the two types of work.

2.1. Work 1

In work 1, we use a client-server model that consists of a client and a server (see Fig. 1). A user of the client moves a rigid cube (the length of each side is 1/4 of the height of a virtual space surrounded by walls, a floor, and a ceiling as shown in Fig. 2) as an object in a virtual space by using a cursor (i.e., a position which the user tries to touch or are touching with a haptic interface device) of PHANToM DESKTOP or SPIDAR-G AHS (denoted by a hemisphere beneath the cube in Fig. 2). In the case of PHANToM DESKTOP, the height of the virtual space is 89.7 mm, the width is 129.7 mm, and the depth is 89.7 mm; in the case of the SPIDAR-G AHS, the height is 200 mm, the width is 120 mm, and the depth is 200 mm. The cursor moves in the space when the user manipulates the stylus of PHANToM DESKTOP or the grip of SPIDAR-G AHS with his/her hand. In our experimental system, the user lifts and moves the cube so that the cube contains a target (denoted by a sphere in Fig. 2; its diameter is equal to the length of each side of the object) which revolves along a circular orbit at a constant velocity. We do not carry out collision detection among the target, the orbit, and the object or cursor.

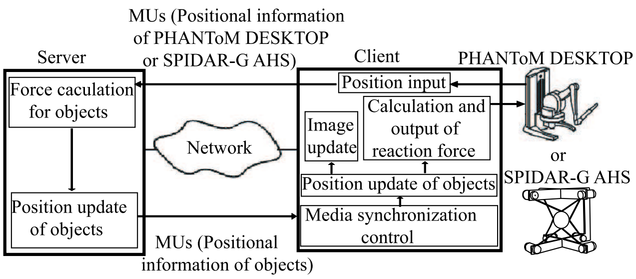

Figure 1. System model of work 1.

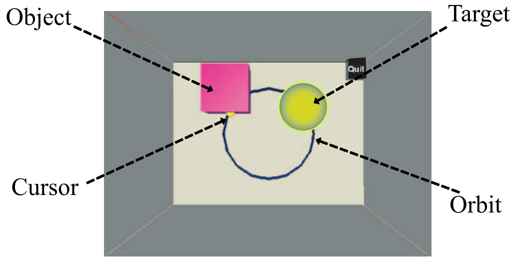

Figure 2. Displayed image of virtual space in work 1.

The server manages the information of the virtual space and updates the space [5]. The client inputs the position of the cursor, carries out the media synchronization control over MUs which are received from the server, and outputs the reaction force to the user through PAHNToM DESKTOP or SPIDAR-G AHS. The reaction force is calculated based on the Spring-Damper model [12] so that the force is proportional to the penetration depth of the cursor into the object and the velocity of the cursor.

2.2. Work 2

In work 2, we use a P2P model which consists of two terminals (see Fig. 3). While an instructor and a learner are feeling the sense of force interactively through PHANToM Omni connected to each terminal, the learner draws a figure with a single stroke of brush as shown in Fig. 4 (the instructor and learner draw a cartoon character in Fig. 4, but they drew a spiral in our experiment for simplicity) on a virtual canvas (height: 152 mm, width: 214 mm) by following the trace of the instructor. The instructor and learner terminals carry out media synchronization control after receiving the position information (i.e., MUs) from each other. At each terminal, the reaction force which is output to PHANToM Omni is calculated based on the position comparison between the PHANToM Omni cursor (i.e., the tip of the paintbrush) of the instructor terminal and that of the learner terminal. Then, the reaction force is calculated so that the force is proportional to the distance between the instructor's cursor and the learner's cursor [4]. In addition, updating figure on the virtual canvas is also carried out [4], [15].

3. Experimental Systems

Here we explain the experimental systems of work 1 and work 2. In this paper, we deal with two types of networks. One uses a network emulator (NIST Net [16]), and the other employs routers.

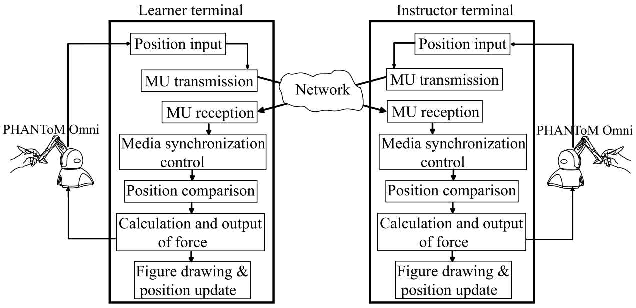

Figure 3. System model of work 2.

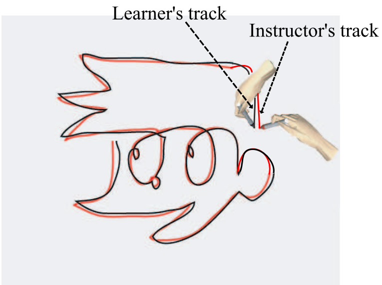

Figure 4. Displayed image of virtual space in work 2.

3.1. Work 1

(a) Case with NIST Net

In this experiment, a server (CPU: Xeon 3.06 GHz, OS: Windows 2000) and a client (CPU: Pentium4 1.5 GHz, OS: Windows 2000) are connected to each other via NIST Net (see Fig. 5). By using NIST Net, we generate an additional delay for each MU transmitted from the server to the client according to the Pareto normal distribution [16]; for simplicity, we generate no additional delay for each MU transmitted in the reverse direction.

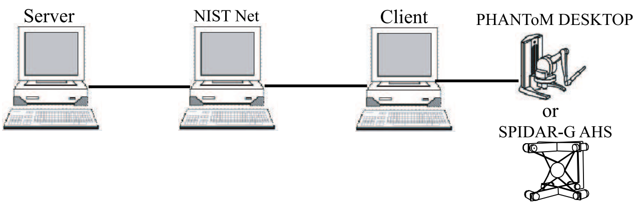

Figure 5. Experimental system of work 1 (a)

(b) Case with routers

As shown in Fig. 6, the experimental system consists of a server (CPU: Xeon 3.06 GHz, OS: Windows 2000), a client (CPU: Pentium4 1.5 GHz, OS: Windows 2000), two data terminals (i.e., a data sending terminal and a data receiving terminal), two 100BASE-T Ethernet switching hubs, and two routers (Cisco 2611). The two routers are connected to each other by a V.35 serial cable (full duplex transmission of 2 Mbps). The data receiving terminal and the client are connected to the routers via one of the switching hubs, and the data sending terminal and the server are connected to the routers via the other switching hub.

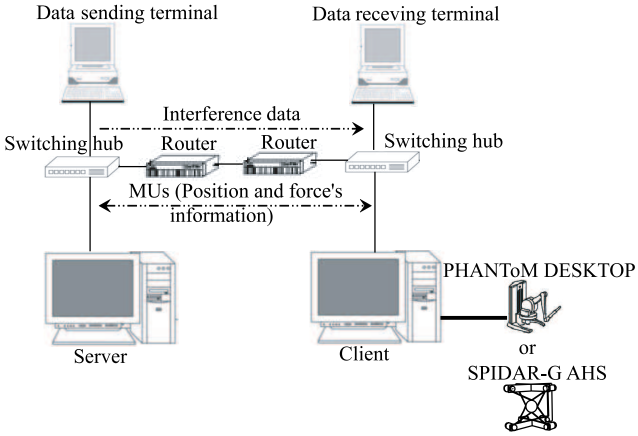

Figure 6. Experimental system of work 1 (b).

In order to generate a traffic flow of interference, the data sending terminal sends fixed-size data messages of 1472 bytes each to the data receiving terminal at exponentially distributed intervals. For transmission of MUs and interference data messages, we use the UDP protocol in this paper.

3.2. Work 2

(a) Case with NIST Net

In the remote haptic drawing system, an instructor terminal (CPU: Pentium4 2.8 GHz, OS: WindowsXP) and a learner terminal (CPU: Pentium4 2.8 GHz, OS: WindowsXP) are connected to each other via NIST Net (see Fig. 7). NIST Net is used to generate an additional delay for each MU transmitted between the server and the client according to the Pareto normal distribution.

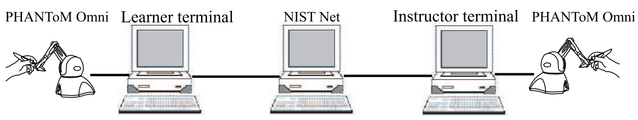

Figure 7. Experimental system of work 2 (a).

(b) Case with routers

As shown in Fig. 8, the experimental system consists of an instructor terminal (CPU: Pentium4 2.8 GHz, OS: WindowsXP), a learner terminal (CPU: Pentium4 2.8 GHz, OS:

WindowsXP), two data terminals, two 100BASE-T Ethernet switching hubs, and two routers (Cisco 2611). The network configuration of the experimental system and the generation and transmission methods of interference data messages are the same as those in work 1 (b).

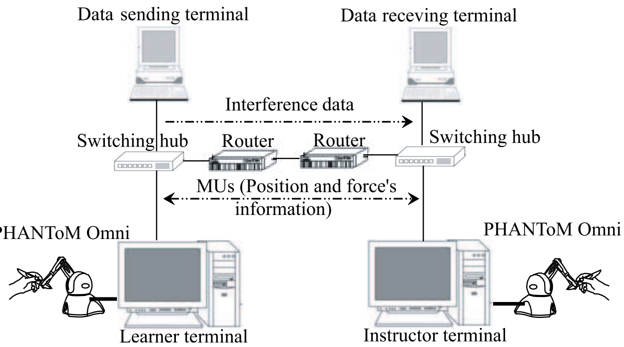

Figure 8. Experimental system of work 2 (b).

4. Synchronization Algorithms

In this paper, we make a comparison of output quality among the following five haptic media synchronization algorithms: VTR, Buffering, Skipping, QM, and ABC. The five algorithms carry out order control. The control discards an MU without outputting the MU if its sequence number is smaller than those of MUs which have been output.

VTR has a virtual-time axis which can be contracted or expanded dynamically according to the network delay jitter. Media synchronization is maintained by outputting MUs along the virtual-time axis [5]. Buffering carries out only initial buffering control and pausing control [5]. The initial buffering control exerts a constant-time buffering of the first MU. Skipping outputs MUs on receiving the MUs, and it outputs only the latest MU which is stored in the receiving buffer and discards the other MUs left in the buffer [5]. QM deletes the oldest MU in the buffer if the value of counter (a counter is set to each MU in the buffer, and the counter is incremented by one whenever an MU is output) exceeds a threshold value T_h. At the same time, all the counters' values are reset to zero [9]. ABC dynamically extends the buffering time of MUs according to the network delay under the adaptive buffer approach, which determines the buffering time by observing the network delay, and the time-adjustment mechanism, which determines the output time of each MU by adding the buffering time to the generation time [7]

In the case of PHANToM DESKTOP of work 1 (a), we adopt parameter values of each haptic media synchronization algorithm which are shown in Table 1. In a preliminary experiment, the operation of PHANToM DESKTOP became difficult in VTR when the maximum allowable delay $\Delta_{\rm al}$ [5] was larger than or equal to 65 ms; thus, the maximum value of $\Delta_{\rm al}$ was set to 65 ms. Also, since we had no large difference in experimental results when the value of $\Delta_{\rm al}$ was less than or equal to 35 ms, the minimum value of $\Delta_{\rm al}$ was set to 35 ms. Furthermore, we set an estimated value of the maximum network delay jitter (i.e., the initial buffering time) $J_{\max}$ [5] to 10 ms because there was almost no change in experimental results when $J_{\max} \leq \Delta_{\rm al}$ [5].

In Buffering, when the initial buffering time X was smaller than or equal to 10 ms, the experimental results were almost the same as those in the case of X = 10 ms. In addition, the experimental results were hardly improved when X was larger than or equal to 20 ms. Therefore, X was set to 10 ms and 20 ms.

Skipping does not have any parameter.

In QM, since the efficiency of work was not improved when the threshold value T_h was larger than or equal to 15, the maximum value of T_h was set to 15. Note that the minimum value of T_h is 1.

In ABC, since there was no large change in the efficiency of work when the number of observed MUs N was larger than or equal to 200, we set the maximum value of N to 200. When the value of N was smaller than 7, the efficiency of work became worse than that in the case of N = 7; thus, the minimum value of N was set to 7.

Parameter values of each algorithm for PHANToM DESKTOP of work 1 (b), SPIDAR-G AHS of work 1 (a) and (b), and PHANToM Omni of work 2 (a) and (b) are determined in the same way as that in the case of PHANToM DESKTOP of work 1 (a). The parameter values are shown in Tables 1 and 2.

5. Assessment Methods

We have carried out subjective assessment to compare the output quality among the five haptic media synchronization algorithms. The subjective assessment of work 1 (a) which uses PHANToM DESKTOP, that of work 1 (a) which employs SPIDAR-G AHS, and that of work 2 (a) were carried out separately for each subject. Also, the subjective assessment of work 1 (b) which uses PHANToM DESKTOP, that of work 1 (b) which employs SPIDAR-G AHS, and that of work 2 (b) were performed separately for each subject. In the subjective assessment, we had fifteen subjects whose ages were between 20 and 24. We enhanced the single-stimulus method of ITU-R BT. 500-11 [17], which is a recommendation for subjective assessment of television pictures. This is because there is no standard for subjective assessment of haptic media.

After practicing several times under the condition that there was no additional delay or no data load, each subject gave a score based on the scores listed in Table 3 according to the degree of deterioration in haptic and visual feelings on the condition that there exist additional delays or data

Table 1. Parameters of each algorithm in work 1.

Algorithm	Parameter	Work 1 (a) PHANToM DESKTOP	Work 1 (a) SPIDAR-G AHS	Work 1 (b) PHANToM DESKTOP	Work 1 (b) SPIDAR-G AHS
VTR	Maximum allowable delay $\Delta_{\rm al}$ [ms]	35, 45, 55, 65	30, 45, 60, 75	50, 55, 60, 65	
Buffering	Initial buffering time X [ms]	10, 20			
Skipping	-	-			
QM	Threshold T_h	1, 5, 10, 15			
ABC	Number of observed MUs N	7, 100, 150, 200			

Table 2. Parameters of each algorithm in work 2.

Algorithm	Parameter	Work 2 (a)	Work 2 (b)
VTR	Maximum allowable delay $\Delta_{\rm al}$ [ms]	100, 120, 140, 160, 180, 200	20, 40, 60, 80, 100, 120
Buffering	Initial buffering time X [ms]	20, 40, 60, 80, 100	
Skipping	-	-	
QM	Threshold T_h	1, 5, 10, 15	
ABC	Number of observed MUs N	7, 30, 60, 90	

loads. In the assessment, we selected the five haptic media synchronization algorithms randomly and set the average additional delay or the data load randomly. Thus, we obtained the *mean opinion score* (*MOS*) [17].

Table 3. Five-grade impairment scale.

Score	Description
5	Imperceptible
4	Perceptible, but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

In work 1, each subject carried out the subjective assessment at the client. In work 2, each subject made the subjective assessment as an instructor[1], and one of the authors always manipulated PHANToM Omni as a learner. From a small set of preliminary experiment, we deduced that a duration of 25 seconds of a test sample is sufficiently long for getting the opinions of subjects in both work 1 and work 2. Thus, the measurement of the quality was carried out for 25 seconds from 5 seconds[2] after the beginning of each experiment run. As for the total assessment time per subject, work 1 (a) and work 1 (b) which use PHANToM DESKTOP and work 1 (a) and work 1 (b) which employ SPIDAR-G AHS took about 40 minutes each. Also, work 2 (a) took around 40 minutes, and work 2 (b) took approximately 50 minutes.

[1] As a result of a preliminary experiment, we could see no large difference in the output quality among the five haptic media synchronization algorithms at the learner terminal. This is because the learner did not need stronger force than the instructor to operate the PHANToM Omni, and he/she may regard the degradation caused by the additional delay or data load as the force exerted by the instructor [4].

[2] We lifted and moved the object from the floor to the target within the 5 seconds in work 1. In work 2, we started the remote haptic drawing instruction work within the 5 seconds.

6. Assessment Results

In this section, we present assessment results of work 1, and then we present those of work 2.

6.1. Work 1

6. 1. 1 Case of PHANToM DESKTOP

We show the assessment results in the cases with NIST Net and with routers.

(a) Case with NIST Net

We show the MOS values of the five algorithms as a function of the average additional delay in Fig. 9, where we also plot the 95% confidence intervals. The MOS values of each algorithm in Fig. 9 are obtained at parameter values which have the highest MOS value for each average additional delay. We also set the standard deviation of the additional delay from the server to the client to 10 ms.

From Fig. 9, we can see that the MOS values of VTR, Skipping, and QM are higher than those of Buffering and ABC. Since the buffering time is long in Buffering and ABC, the *average MU delays* of Buffering and ABC are larger than those of VTR, Skipping, and QM. The average MU delay is defined as the average time from the moment an MU is generated until the instant the MU is output. It should be noted that as the average MU delay increases, the work becomes harder [5].

We also made the experiment with the standard deviations of 5 ms and 15 ms. As a result, we obtained almost the same results as those when the standard deviation is 10 ms.

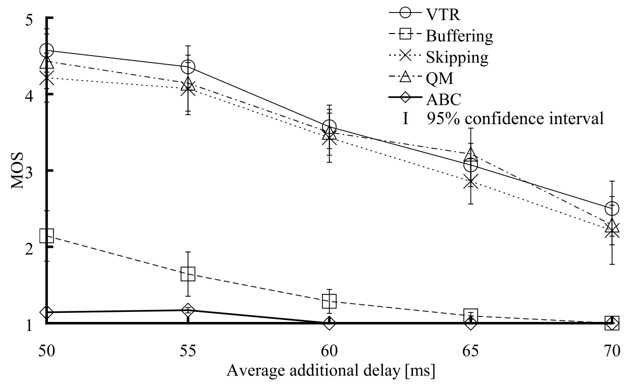

Figure 9. MOS of work 1 (a) (PHANToM DESKTOP).

(b) Case with routers

We show the MOS values versus the data load in Fig. 10, where we also plot the 95% confidence intervals. The data load is defined as the average number of interference data bits transmitted in a second at the data sending terminal. The MOS values of each algorithm in Fig. 10 are obtained at parameter values which have the highest MOS values for each data load.

From Fig. 10, we find that the MOS value of Buffering is almost the same as those of VTR, Skipping, and QM, which are higher than that of ABC when the data load is heavier than about 1.25 Mbps and lighter than around 2.00 Mbps. This is because there was no large difference in the average MU delay among VTR, Buffering, Skipping, and QM; the average MU delay of ABC was longer than those of the other four algorithms.

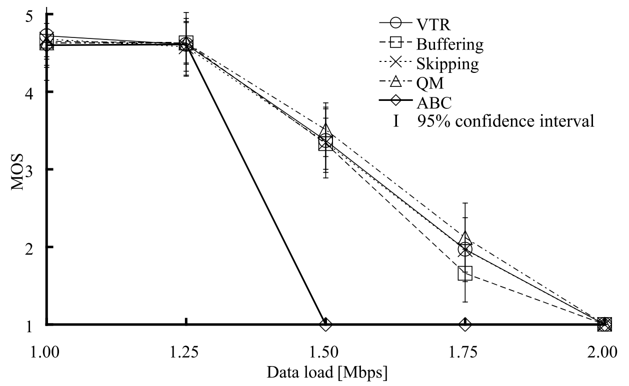

Figure 10. MOS of work 1 (b) (PHANToM DESKTOP).

6. 1. 2 Case of SPIDAR-G AHS

(a) Case with NIST Net

We show the MOS values as a function of the average additional delay in Fig. 11. The parameter values of each algorithm are chosen in the same way as that in 6.1.1 (a).

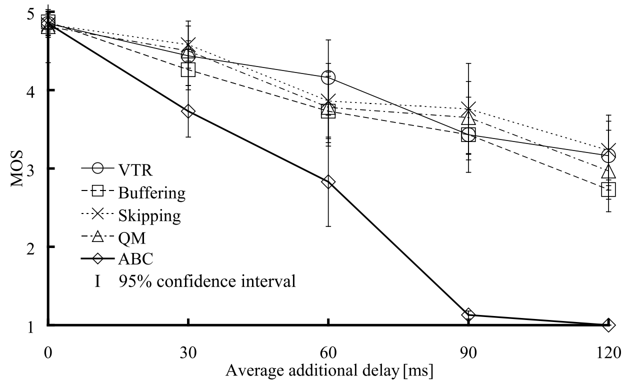

Figure 11. MOS of work 1 (a) (SPIDAR-G AHS).

From Fig. 11, we can see that the MOS values of VTR, Buffering, Skipping, and QM are higher than that of ABC when the average additional delay is larger than about 0 ms. However, the MOS value of Buffering tends to be slightly smaller than those of VTR, Skipping, and QM.

Furthermore, we made the experiment with the standard deviations of 5 ms and 15 ms as in 6.1.1 (a). As a result, we obtained almost the same results as those in the case where the standard deviation is 10 ms.

(b) Case with routers

The MOS values versus the data load are shown in Fig. 12. The parameter values of each algorithm are chosen in the same way as that in 6.1.1 (b).

In Fig. 12, we can see that the MOS values of the five algorithms in the case of SPIDAR-G AHS have a similar tendency to those in the case of PHANToM DESKTOP (i.e., in Fig. 10).

6.2. work 2

(a) Case with NIST Net

We show the MOS values versus the standard deviation of the additional delay in Fig. 13. In the figure, the average additional delay between the instructor terminal and the learner terminal is set to 100 ms.

From Fig. 13, we note that the MOS values of VTR and Buffering are the highest, and those of Skipping and QM are the second highest, and that of ABC is the lowest. This is because VTR and Buffering absorb network delay jitter

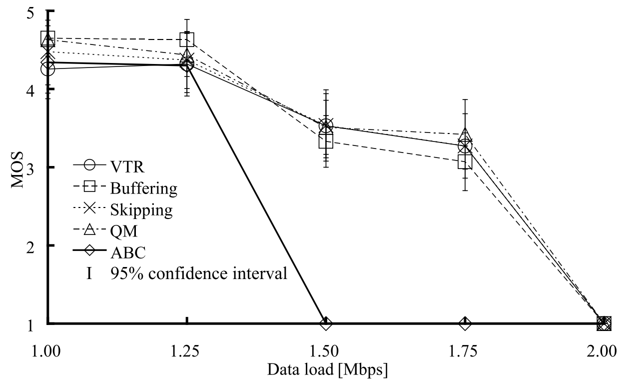

Figure 12. MOS of work 1 (b) (SPIDAR-G AHS).

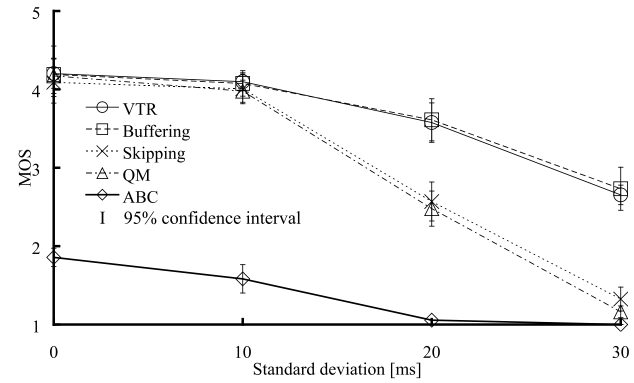

Figure 13. MOS of work 2 (a).

more effectively than Skipping and QM[3], which absorb network delay jitter more effectively than ABC.

(b) Case with routers

We show the MOS values versus the data load in Fig. 14. Figure 14 reveals that the MOS values of the five algorithms are almost the same. The reason is that the network delay jitter was tiny in this case.

From Figs. 10, 12 and 14, we can see that in work 1 (b), the MOS value of ABC is smaller than those of the other four algorithms when the data load is heavier than about 1.25 Mbps and lighter than around 2.00 Mbps; on the other hand, in work 2 (b), there is no large difference among the five algorithms, and the MOS value of ABC is higher than that of work 1 (b) when the data load is heavier than about 1.25 Mbps and lighter than around 2.00 Mbps. This is because the average network delay in work 2 (b) was smaller than that in work 1 (b).

From the above observations, we can say that when we use PHANToM DESKTOP, the output quality of VTR, Skipping, and QM is the highest in work 1 (a); the output quality of VTR, Buffering, Skipping, and QM is the highest in work 1 (b). When we use SPIDAR-G AHS, the output quality of VTR, Buffering, Skipping, and QM is the highest in both work 1 (a) and work 1 (b). Furthermore, in the case of work 2 (a), the output quality of VTR and Buffering is the highest, but there is no large difference in the output quality among the five algorithms in work 2 (b). Therefore, VTR is the most effective in all the types of work.

In this study, we also carried out objective assessment at the same time as the subjective assessment. As an objective assessment measure, we adopted the *average distance between cube and target* [5] in work 1 and the *following rate* [15] in work 2. The average distance between cube and target denotes the efficiency of work. The following rate means how much accurately the learner's brush stroke can catch up with the instructor's one. Since the objective assessment results had a similar tendency to the subjective assessment results, we carried out multiple regression analysis to investigate the relations between the subjective and objective assessment results. By the analysis, we found that agreement between the experimental values and estimated values of MOS is good. Therefore, we can estimate the MOS values with high accuracy from the average distance between cube and target or the following rate.

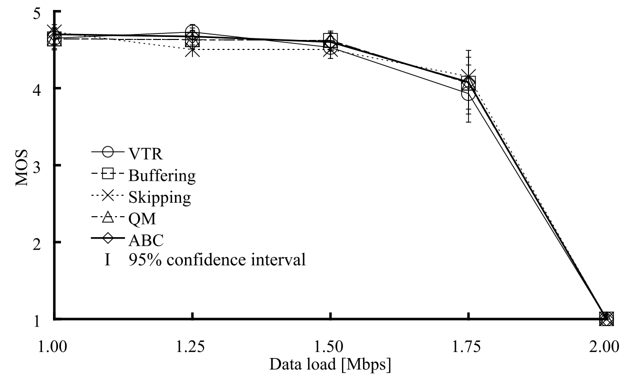

Figure 14. MOS of work 2 (b).

7. Conclusions

In this paper, we made a comparison of output quality among five haptic media synchronization algorithms: VTR, Buffering, Skipping, QM and ABC. By subjective assessment, we found that when we use PHANToM DESKTOP, the output quality of VTR, Skipping, and QM is the highest

[3] In QM of work 2, the MOS value was the highest when the threshold value T_h was one. In this case, QM outputs each MU on receiving the MU. Since the MU transmission rate of work 2 is low (i.e., 30 Hz), multiple MUs are hard to be saved at the same time in the buffer, and the number of MUs in the buffer is at most one. Therefore, QM behaves like Skipping, and large values of T_h have no effect.

in work 1 (a) (where a user lifts and moves a virtual object in the case with NIST Net); the output quality of VTR, Buffering, Skipping, and QM is the highest in work 1 (b) (where a user lifts and moves a virtual object in the case with routers). When we use SPIDAR-G AHS, the output quality of VTR, Buffering, Skipping, and QM is the highest in both work 1 (a) and work 1 (b). Furthermore, in work 2 (a) (which is a remote haptic drawing system in the case with NIST Net), the output quality of VTR and Buffering is the highest, but there is no large difference in the output quality among the five algorithms in work 2 (b) (which is the remote haptic drawing system in the case with routers). Therefore, VTR is the most effective in all the types of work.

As the next step of our research, we need to make a comparison of output quality for other types of work. We also plan to use other kinds of haptic interface devices.

References

[1] M. A. Srinivasan and C. Basdogan. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computers and Graphics*, 21(4):393–404, April 1997. 1

[2] Y. Ishibashi and H. Kaneoka. Fairness among game players in networked haptic environments: Influence of network latency. *Proc. IEEE ICME'05*, July 2005. 1

[3] H. Kaneoka and Y. Ishibashi. Effects of group synchronization control over haptic media in collaborative work. *Proc. the 14th International Conference on Artificial Reality and Telexistence (ICAT'04)*, pages 138–145, November/December 2004. 1

[4] T. Asano and Y. Ishibashi. A remote haptic painting lesson system. *Proc. SPIE Optics East, Multimedia Systems and Applications X*, 6777(28), September 2007. 1, 2, 5

[5] Y. Ishibashi, S. Tasaka, and T. Hasegawa. The virtual-time rendering algorithm for haptic media synchronization in networked virtual environments. *Proc. the 16th International Workshop on Communications Quality and Reliability (CQR'02)*, pages 213–217, May 2002. 1, 2, 4, 5, 7

[6] K. Hikichi, H. Morino, Y. Yasuda, I. Arimoto, and K. Sezaki. The evaluation of adaptive control for haptics collaboration over the Internet. *Proc. the 16th International Workshop on Communications Quality and Reliability (CQR'02)*, pages 218–222, May 2002. 1

[7] O. Wongwirat and S. Ohara. Haptic media synchronization for remote surgery through simulation. *IEEE Multimedia*, 13(3):62–69, July-September 2006. 1, 4

[8] O. Wongwirat and S. Ohara. Moving average based adaptive buffer for haptic media synchronization in telehaptics. *Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1305–1311, July 2005. 1

[9] D. L. Stone and K. Jeffay. An empirical study of delay jitter management policies. *ACM Multimedia systems*, 2(6):267–279, January 1995. 1, 4

[10] Y. Ishibashi and H. Kaneoka. Group synchronization for haptic media in a networked real-time game. *IEICE Trans. Commun., Special Section on Multimedia QoS Evaluation and Management Technologies*, E89-B(2):313–319, Feburary 2006. 1

[11] J. K. Salisbury and M. A. Srinivasan. Phantom-based haptic interaction with virtual objects. *IEEE Computer Graphics and Applications*, 17(5):6–10, September/October 1997. 2

[12] SensAble Technologies, Inc. Ghost SDK programmer's guide Version 3.0, 2004 2

[13] S. Kim, J. J. Berkley, and M. Sato. A novel seven degree of freedom haptic device for engineering design. *Virtual Reality*, 6(4):217–228, August 2003. 2

[14] SensAble Technologies, Inc. 3D Touch SDK OpenHaptics Toolkit programmer's guide Version1.0, 2004 2

[15] Y. Kurokawa and Y. Ishibashi. A group synchronization scheme with prediction in a remote haptic drawing system. *Proc. SPIE Optics East, Multimedia Systems and Applications X*, 6777(24), September 2007. 2, 7

[16] M. Carson and D. Santay. NIST Net - A Linux-based network emulation tool. *ACM SIGCOMM Computer Communications Review*, 33(3):111–126, July 2003. 2, 3

[17] ITU-R. BT. 500-11. Methodology for the subjective assessment of the quality of television pictures. *International Telecommunication Union*, 2002. 4, 5