네트워크 햅틱 협업을 위한 햅틱 데이터 전송

Haptic Data Transmission in Networked Haptic Collaboration

유용희, Yonghee You*, 성미영, Mee Young Sung*, 김남중, Nam-Joong Kim*, 강진석, Jin Suk Kang*, 전경구, Kyungkoo Jun**
*인천대학교 컴퓨터공학과, **인천대학교 멀티미디어시스템공학과

요약˘ 이 논문은 촉각 가상 환경(Haptic Virtual Environment)에서 촉각(haptic) 데이터를 촉각 데이터 특성에 맞추어 네트워크에 적응적이고 효율적으로 전송할 수 있는 전송 알고리즘을 제안한다. 촉감 상호작용 측정을 위해 네트워크 햅틱 협업 응용을 작성하였고 지연, 지터, 손실에 따른 변화를 분석하였다. 이를 바탕으로 네트워크 트래픽에 적응할 수 있는 알고리즘을 구성하였다. 손실되거나 지터의 영향을 받은 패킷에는 간단한 선형예측 방법을 사용하여 보상하여 손실과 지터로 인한 오차를 줄였다. 이는 심각한 손실이나 지터에 의해 떨림 현상이 나타나는 햅틱 장치의 문제점을 개선하게 되었다. 또한 네트워크 협업에서 지연이 발생할 때 나타나는 클라이언트들 사이의 비동시성을 해결하기 위하여 완충시간을 두었다. 지연이 큰 클라이언트는 버퍼를 사용하지 않고 실시간으로 처리하고, 지연이 적은 클라이언트는 버퍼를 사용하여 전송받은 좌표를 완충시킨 후에 처리하는 방법을 사용하여 클라이언트들 사이의 햅틱 렌더링을 동기화 하였다. 제안된 알고리즘은 다양한 네트워크 상황에서의 협업에서 개선된 결과를 보였다. 이를 바탕으로 향후 선형예측 방법을 다양하게 적용시키고 서버와 클라이언트 사이의 동기화를 구현하는 알고리즘을 작성할 것이다. 본 논문은 다양한 네트워크 상황 에서 햅틱 데이터를 전송하고 처리하는 연구의 기초자료가 될 수 있을 것이다.

핵심어: Haptic Interaction, Networked Collaboration, Haptic Virtual Environment

1. 서론

최근 가상 환경에서는 3D 그래픽스와 오디오 및 비디오 데이터 등의 시각과 청각 데이터 활용뿐 아니라, 본 논문의 관심인 촉각 데이터의 추가적인 활용을 통하여 사용자 몰입(immersion)의 정도가 점점 높아지고 있다. 이 가상환경에 촉각장치를 이용한 인터페이스를 추가하면 촉각 가상환경 Haptic Virtual Environment; HVE)이 되며 이는 교육 시뮬레이션 같은 촉각 장치와 VE가 한 시스템에 같이 존재하는 상태를 말한다[1]. 이때 촉각 장치를 가진 시스템이 원격으로 연결이 되어 공동작업 혹은 협업을 하는 것을 협업 가상 환경(Collaborative Vitual Environment; CVE)이라고 한다. 협업가상환경을 이용한 응용들은 원격 의료교육 시스템 등이 있다. 이때 불안정한 네트워크 상황은 부정확한 힘 피드백(force-feedback)을 야기할 수 있다[2].

협업가상환경에서 햅틱 데이터는 모든 클라이언트에 의해 공유되는 가상 객체를 의미하기도 하고, 햅틱장치에서 가리키는 3D 좌표일 수도 있다. 이런 좌표의 전송은 기존 네트워크 게임 혹은 비디오, 오디오의 전송방법과 유사하지만 그래픽 렌더링 주기에 맞춘 초당 대략 60회 업데이트 패킷을 전송하는 반면 햅틱 렌더링을 위해서는 햅틱 랜더링 주기인 1kHz에 맞추어 전송하여야 한다는 점이 다르다. 그러므로 햅틱 데이터 전송은 실제 네트워크 환경에 매우 민감하여 협업 환경에서 지연(delay), 손실(loss), 지터(jitter)에 기인하여 협업을 적절히 구현 할 수 없다[3]. 이런 햅틱 데이터의 전송을 위해 다양한 연구가 진행되고 있다. Yutaka Ishibashi 등의 학자들은 비디오와 오디오 전송에 사용된 Virtual-time rendering(VTR) 알고리즘을 개선하여 햅틱의 그룹동기화에 대한 연구를 진행하고 있다[4]. 또한 K. Hikichi 등은 Queue Monitoring(QM) [5]을 사용하는 등 다양한 방법을 사용하여 햅틱 협업의 효율성을 높이고 있다.

본 논문에서는 지연(delay), 손실(loss), 지터(jitter)등의 불규칙적인 네트워크 상황에서도 햅틱 협업이 효율적으로 성취될 수 있도록 햅틱 데이터를 전송하는 알고리즘을 제안하고자 한다. 이를 위하여 특정한 햅틱 협업을 수행하는 네트워크 응용 프로그램을 작성하였고, 다양한 네트워크 상황하에서 실행하여 본 후, 이에 따른 문제점을 개선하는 알고리즘을 제안하였다.

주저자: 유용희, 교신저자: 성미영, 본 연구는 교육인적자원부 두뇌한국21(BK21) 사업, 산업자원부 지방기술혁신사업(RTI05-03-01), 산업자원부 산업기술평가원 지정 인천대학교 멀티미디어연구센터의 지원으로 수행되었음.

본 논문의 2절에서는 관련연구에 대하여 설명하고, 3절에서는 네트워크 햅틱 협업 응용에 대하여 설명한다. 4절에서는 햅틱 데이터 전송 시 다양한 네트워크 상황에 따른 보상 방법에 대하여 논하며 실험에 대한 결과를 분석한다. 5절에서는 본 논문의 결론과 향후 연구계획을 서술한다.

2. 관련연구

관련연구에서는 촉각 렌더링에 다하여 간략하게 살펴보고, 실험 시스템 구현에 사용되는 OpenHaptics Toolkit, QUANTA (The Quality of Service Adaptive Networking Toolkit) [6] 네트워킹 기술에 대하여 알아본다.

2.1 촉각 렌더링(Haptic Rendering)

촉각 렌더링[7]은 높은 충실도와 실제감을 가진 시뮬레이션 시스템 또는 교육시스템 등에서 물리적 세계와 가상환경과의 실감적인 상호작용을 보여주는 기술이다[8]. 3D 가상환경의 객체와 햅틱 장치의 위치를 나타내는 포인터 사이의 거리를 측정한 후 Spring-Damper 알고리즘[9]을 사용하여 힘 피드백을 계산하고 이를 햅틱장치로 전달하여 사용자가 실제로 촉감을 느낄 수 있게 하는 역할을 한다.

2.2 OpenHaptics Toolkit

SensAble사의 OpenHaptics Toolkit은 3D 환경에서 손쉽게 Phantom 촉각 장치를 사용할 수 있도록 해주는 응용프로그램[4]으로 OpenGL API를 기반으로 한다. OpenHaptics Toolkit는 총 5개로 나누어져 있다. 그 구조는 그림 1과 같다.

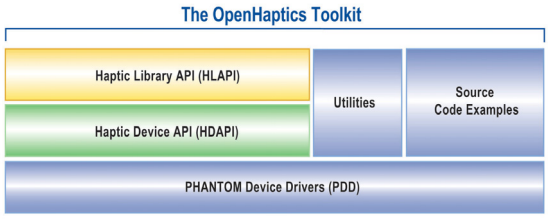

그림 1. OpenHaptics 구조도

Haptic Device API(HDAPI)는 촉각장치의 저수준(하드웨어계층)의 접근을 지원하며 프로그래머가 햅틱 장치에 직접 힘 피드백 혹은 재질 등의 값들을 적용시킬 수 있다. Haptic Labrary API(HLAPI)는 고수준(응용계층)의 촉각 렌더링을 지원하며 OpenGL의 피드백 버퍼를 사용하여 햅틱 렌더링을 하기 때문에 촉각 응용을 개발 시 기존 OpenGL 응용들의 재사용성이 높다. Utilities 는 vector/matrix 관련 수학 함수들과 촉각 장치를 사용하는데 필요한 다양한 기능들을 포함하고 있다.

2.3 QUANTA 네트워크 라이브러리

QUANTA는 크로스 플랫폼의 고속데이터 전송을 목표로 하는 네트워크 툴이다[6], TCP, UDP를 사용하여 Reflector TCP/UDP, Parallel TCP, Reliable Blast UDP 같은 다양한 기능을 지원하며, IPv4, IPv6, thread, mutex 같은 기능들도 지원한다. 또한 CAVE(CAVE Automatic Virtual Environment) 시스템의 CAVERN(CAVE Automatic Virtual Environment Research Network) 라이브러리[10]가 그 전신으로 분산가상환경 (Distributed Virtual Environment; DVE) 시스템에 알맞은 구조로 되어 있다. 본 연구에서는 이런 장점을 지닌 QUANTA 라이브러리를 사용하여 네트워킹 모듈을 구현하였다.

3. 네트워크 햅틱 협업 응용

이 장에서는 네트워크 햅틱 협업 응용, 소프트웨어 구성, 네트워크 구성 그리고 햅틱 데이터 전송 시스템 알고리즘에 대하여 설명한다.

3.1 협업 응용 구성

하나의 공간 안에 1개의 큐브와 1개의 큰 구 그리고 2개의 작은 구가 있다. 공간은 모든 객체의 행동반경을 의미하며 구, 큐브의 행동을 제약한다. 2개의 작은 구는 클라이언트 햅틱 포인터를 나타내며 네트워크로 연결되어 있다. 2개 혹은 그 이상일 수 있다. 클라이언트들은 햅틱 포인터를 움직여 큐브를 움직일 수 있으며 큐브를 협업을 통하여 구가 있는 곳까지 옮겨야 한다. 구는 햅틱 렌더링이 되지 않으며 단지 목표지점을 의미하고 큐브가 구에 도착하면 구는 새로운 위치로 이동하게 된다. 응용의 사진은 그림 2에서 볼 수 있다.

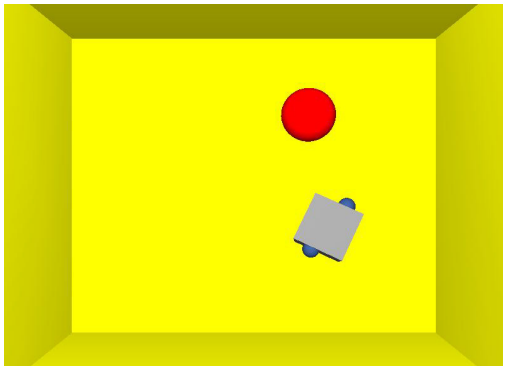

그림 2. 네트워크 햅틱 협업 응용

3.2 소프트웨어 구성

본 협업 시스템의 네트워킹 소프트웨어는 QUANTA 라이브러리를 사용하여 구현하였으며 촉각 데이터 전송을 위하여 실시간성이 뛰어난 UDP를 선택하였다. 햅틱 렌더링을 위해서는 SensAble사의 OpenHaptics Toolkit[4]를 사용하였다. OpenGL 피드백 버퍼를 직접적으로 사용하여 정적 객체인 벽면의 햅틱 렌더링을 위하여 HL라이브러리를 사용하였으며, 동적인 객체인 정사각형의 햅틱렌더링을 위하여 HD 라이브러리를 혼합하여 사용하였다. 큐브의 실재감 있는 움직임을 위하여 Open Dynamics Engine[11]를 사용하였다. 그래픽 렌더링을 위해서는 OpenGL을 사용하였다. 본 시스템의 소프트웨어 구성은 그림 3에서 볼 수 있다.

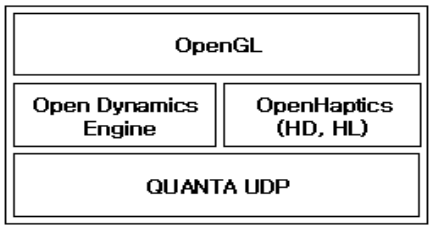

그림 3. 소프트웨어 구성

3.3 네트워크 구성

햅틱 협업 응용의 네트워크 구조는 서버-클라이언트 구조이다. 각각의 클라이언트는 햅틱 장치로부터 얻어진 햅틱 포인터의 좌표를 서버로 보내고, 서버에서는 모든 클라이언트로부터 얻어진 좌표와 큐브의 충돌을 감지하고 Spring-Damper 모델을 적용하여 큐브의 위치와 회전을 계산한 후 모든 클라이언트에게 전송하여 준다. 방금 설명한 네트워크 전송구조는 그림 4에서, 패킷의 구성은 그림 5에서 볼 수 있다.

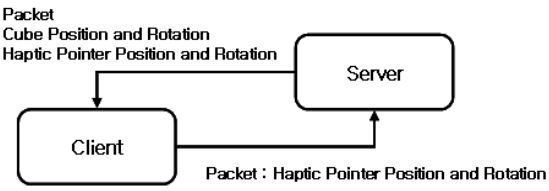

그림 4. 네트워크 전송 구조

Packet Type	Timestamp	Packet Sequence	Position	Rotation

그림 5. 햅틱 데이터 패킷

4. 햅틱 데이터 전송 알고리즘

협업 환경에서의 햅틱데이터 전송은 실제 네트워크 환경에 매우 민감하게 반응한다[12]. 이 장에서는 이런 상황에 대응하기 위하여 데드 레커닝(dead reckoning)[13], 버퍼링(buffering)과 동기화(synchronization) 기법[5]을 사용한 햅틱 데이터 전송 알고리즘을 소개한다. 시스템 구성도는 그림 6에 나와 있다.

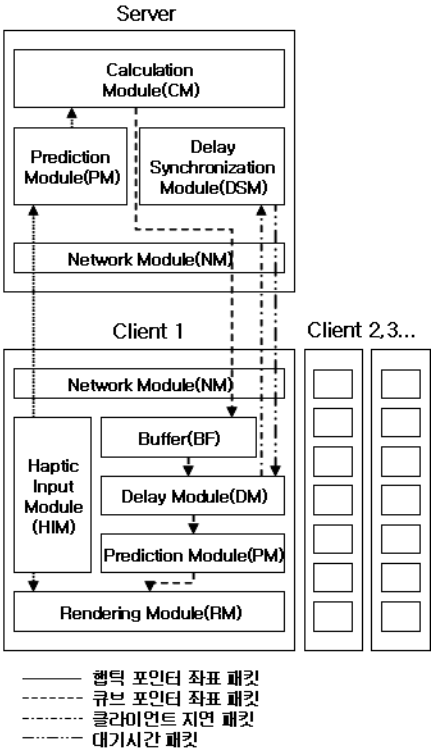

그림 6. 시스템 구성도

CM(Caluculation Module)은 서버에서 주기적으로 일어나는 충돌감지(collision-detection)와 물리(physics) 적용 등의 주기적인 계산이 일어나는 모듈을 총칭한다. NM(Metwork Module)은 네트워크의 전송이 일어나는 소켓을 말하며 RM(Rendering Module)은 주어진 좌표를 가지고 화면에 렌더링하는 모듈을 지칭한다. HIM(Haptic Input Module)은 햅틱 장비로부터 위치데이터를 받아오는 역할을 한다. 이번 장에서는 좌표데이터에 대해 보상하거나 또는 각 클라이언트간의 동기화를 맞추기 위해 사용되는 PM(Predection Module), DSM(display Synchronization Module), BF(Buffer), DM(Delay Module)에 대하여 자세히 설명할 것이다.

4.1 PM(Prediction Module)

PM은 햅틱 포인터의 좌표 혹은 햅틱 객체의 좌표(큐브)가 전송되는 동안 패킷이 손실되었거나 지터에 영향을 받았을 경우 손실된 패킷을 예측하는 모듈이다. 패킷이 손실되었을 경우를 파악하는 방법은 햅틱 렌더링이 1초에 1,000번씩 고정적으로 실행됨을 이용한다. 이에 따라 서버의 CM도 1초에 1,000번씩 실행되고 매 실행 시마다 패킷의 도착을 확

인한다. 만약 새로운 패킷이 도착하지 않았다면 이는 손실로 간주한다. 지터의 경우 마지막 도착한 패킷의 시퀀스 번호가 최근 들어온 시퀀스 번호보다 클 경우 이를 지터로 간주한다. 지터로 인하여 지연된 패킷은 사용하지 않고 폐기한다.

손실된 패킷을 예측하는 것은 수식 (1)과 (2)를 따른다.

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

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

X_n은 예측된 좌표를 나타내고 X_{n-1}은 마지막 도착한 좌표를 나타낸다. V_{n-1}은 마지막 도착한 좌표와 그 전에 도착한 좌표의 차로 구해진다.

4.2 DSM(Delay Synchronization Module)과 DM(Delay Module)

네트워크 햅틱 협업을 실시할 때 지연은 협업에 큰 영향을 미칠 수 있다. 클라이언트에서 햅틱 렌더링 되는 객체(큐브)의 좌표와 다른 클라이언트의 햅틱 렌더링 되는 객체의 좌표가 다르다면 서로 다른 곳에서 협업을 실시하여 사실상 협업의 의미를 감소시키기 때문이다. 이런 각각의 클라이언트에서 일어날 수 있는 지연에 의한 비동시성을 해결하기 위해 서버에 DSM을 클라이언트에 DM을 둔다.

클라이언트에서 DM은 매 5초마다 서버로부터 클라이언트로 전송되는 패킷의 지연시간을 구하여 이를 DSM 서버로 전송하게 된다. 전송된 지연시간은 NM을 거쳐 서버의 DSM으로 전송된다. 모든 클라이언트의 지연시간을 수집한 DSM은 각각의 클라이언트를 위한 완충 시간을 생성해 클라이언트에게 전송해준다. 완충시간을 구하는 수식은 수식 (3)과 (4)를 따른다.

$$M = MAX(D_1, D_2, D_3, \ldots\ldots, D_n) \quad (n = \text{클라이언트 수}) \quad (3)$$

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

D_n은 각 클라이언트에서 보내온 지연시간을 의미하고 M은 지연시간 중 가장 큰 값을 가진다. M에서 각 클라이언트의 지연시간을 뺀 완충시간인 R_n을 구한다. 이렇게 구해진 완충시간은 DM으로 보내지게 된다. DM은 이 완충 시간을 이용하여 비동시성을 해결하게 된다.

완충시간이 0인 클라이언트는 현재 지연이 가장 큰 경우이다. 완충시간이 가장 큰 경우에는 전송받은 좌표를 PM을 거쳐 바로 RM으로 보내어 햅틱 렌더링을 실시한다. 하지만 완충시간이 0보다 클 때에는 전송받는 모든 좌표들은 BF(Buffer)에 모이게 된다. 이렇게 모인 좌표들은 햅틱 렌더링 시에 다음과 같이 수식 (5)와 (6)에 따라 사용된다.

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

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

C는 현재 시간(ms)을 R_n은 완충시간을 나타내고, P_n은 현재시간과 완충시간과의 차이 값이며, $I(P_n)$은 Pn의 크기에 따라 정렬된 인덱스 값을 반환한다. CP_n는 RM에서 사용될 좌표를 나타낸다. 수식 5에 따라 R_n ms 이전에 전송받은 좌표를 CP에 입력하여 햅틱렌더링에 사용하게 된다. BF에 완충시간 R_n ms 동안 입력된 좌표들은 패킷 시퀀스 번호에 의해 정렬된다. 버퍼링의 효과로 완충시간 R_n 이하인 지터의 영향도 사라지게 된다.

5. 햅틱 데이터 전송 실험

논문에서 제안한 알고리즘을 사용하여 햅틱데이터 전송 실험을 실시하였다. 실험에 사용된 테스트베드와 결과의 평가 방법 그리고 실험결과에 대하여 설명한다.

5.1 테스트베드

햅틱 렌더링의 불규칙성을 측정하기 위한 테스트 베드는 서버 컴퓨터, 클라이언트 컴퓨터, 그리고 서버와 클라이언트 컴퓨터를 연결하는 Linux 기반의 네트워크 에뮬레이터 Nistnet로 구성된다. 각 클라이언트 컴퓨터는 햅틱 장치를 포함하고 있으며, 네트워크 에뮬레이터 Nistnet는 지연(delay), 지터(jitter), 손실(loss) 등 다양한 네트워크 시나리오를 지원한다.

햅틱 장치는 SensAble사의 PHANToM Omni[4]이며 햅틱 렌더링 주기는 500~1,000Hz이다. 안정된 햅틱 렌더링을 위해서 1,000Hz의 렌더링 주기를 채택하였다. 그 외 하드웨어 구성은 표 1에서 볼 수 있다.

표 1. 하드웨어 구성

서버 컴퓨터	클라이언트 컴퓨터
– AMD AthlonTM 64 Processor 3500+ 2.21 GHz 1.00 GB RAM – OS : Microsoft Windows XP Professional Version 2002 Service Pack2	– Sensable PHANToM Omni 햅틱 장비 – Dell Precision PWS380 Intel(R) Pentium 4 CPU 3.20GHz, 1.00GB RAM – NVIDIA Quadro FX 1400 그래픽 카드 – OS : Microsoft Windows XP Professional Version 2002 Service Pack2

5.2 평가 방법

평가를 위한 실험의 시나리오는 네트워크 햅틱 협업 응용을 실행하면서 네트워크에 지연, 지터, 손실 등 다양한 네트워크 상황을 적용시키는 것이다. 성능 향상에 대한 평가에 있어서 PM의 경우, 네트워크 트래픽에 의해 손실된 좌표에 대한 예측의 정확성을 측정하기 위하여, 전송된 실제 좌표와 예측된 좌표 그리고 예측 되지 않았을 경우의 좌표를 비교한다. DSM과 DM의 성능을 파악하기 위해서는 지연 상황 발생 시에 양 클라이언트의 햅틱 객체(큐브)좌표를 비교하여 각 좌표의 차이를 확인한다.

5.3 실험 결과

네트워크 햅틱 협업 응용에 제안하는 햅틱 데이터 전송 알고리즘을 적용한 모델로 실제 네트워크 상황에서의 햅틱 데이터 전송 실험을 하였다. 또한 성능 비교를 위해 같은 응용에 제안하는 햅틱 데이터 전송 알고리즘을 적용하지 않은 모델로 동일한 실험을 실시하였다. 정량적 비교를 위하여 오차는 수식 (7)에 따라 현재 렌더링 되는 좌표와 서버에서 마지막으로 생성된 좌표와의 차이로 나타내었다.

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

x_i, y_i, z_i는 현재 렌더링 되는 좌표를 나타내고 x_i', y_i', z_i'는 서버에서 마지막으로 생성된 좌표를 나타낸다.

첫 번째 실험에서는 15%의 손실이 발생하였을 때 햅틱 데이터 전송을 실시하여 손실에 대한 PM의 성능을 측정하였다. 그림 7에서 x축은 시간(ms)를 y축은 오차범위를 나타낸다. 그림 7에서 볼 수 있듯이 우리의 햅틱 데이터 전송 알고리즘을 사용하였을 경우의 성능이 그렇지 않을 경우에 비해 우수하였다. 두 경우 모두의 평균 오차를 구하면 햅틱 데이터 전송 알고리즘을 사용하였을 경우가 0.012, 사용하지 않았을 경우가 0.09로 뛰어난 성능을 보였다.

두 번째 실험에서는 50ms의 지터가 있는 네트워크 상황을 설정하고 햅틱 데이터 전송을 실시하여 PM의 성능을 측정하였다. 지터 상황에서 제안하는 알고리즘을 사용한 경우의 평균오차는 0.102이고, 사용하지 않은 경우는 0.863이었다.

또한 위의 두 실험을 통하여 알고리즘을 사용한 경우에도 손실이 발생하였을 경우의 오차 범위가 지터가 발생했을 때의 오차범위에 비해 현저히 작음을 확인할 수 있었다. 이는 PM의 선형예측 알고리즘은 지터보다는 손실이 발생하였을 때 더욱더 효율적으로 작동함을 의미한다.

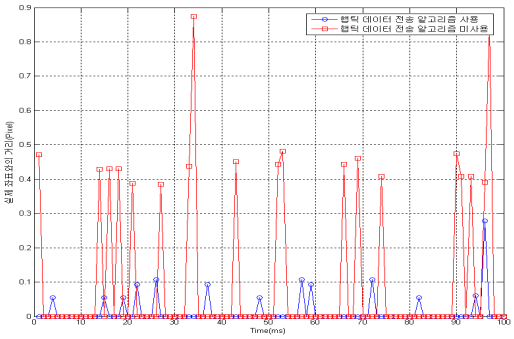

그림 7. 15% 패킷손실률에서의 햅틱 데이터 전송

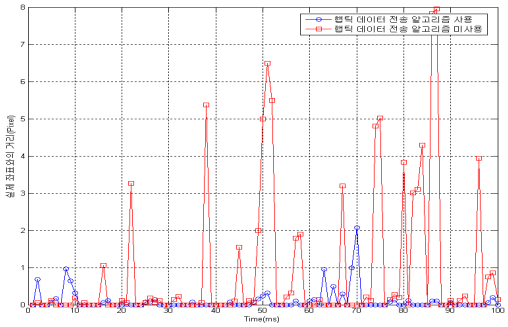

그림 8. 50ms 지터에 따른 햅틱 데이터 전송

세 번째 실험에서는 서버에 두 명의 클라이언트가 접속하였을 때 한쪽 네트워크에만 100ms의 지연을 주어 양 클라이언트 사이의 햅틱 객체의 이동이 비동기적으로 일어나도록 하였다. 이를 통해 DSM과 DM의 동작과 효율을 분석하였다. 그림 9와 10에서 좌표는 각각의 클라이언트로 전송된 햅틱 객체(큐브)의 좌표를 나타내고 기준이 되는 좌표는 가장 최근에 서버에서 생성된 좌표를 말한다. 그림 9에서 알 수 있듯이, 클라이언트 2는 비록 오차율도 적고 클라이언트 1에 비해 실시간으로 햅틱 객체의 좌표를 얻지만 각각의 클라이언트가 전송받는 좌표의 오차가 너무 커 협업이 불가능한 상황을 형성한다.

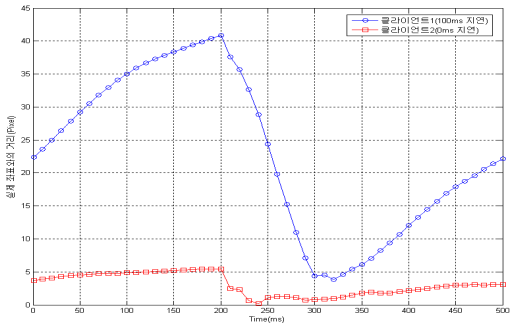

그림 9. 비동시성(100ms 지연)

그러나 그림 10에서는 보듯이 완충시간을 적용함으로써 협업을 가능하게 하였다. 또한 위에 설명한 것과 같이 클라이언트 1에 100ms 지연 실험을 수행하는 동안 클라이언트 2의 네트워크에 10~50ms의 지터를 적용하여 보았다. 실험결과 100ms의 완충 시간과 패킷 버퍼링을 통해 완충시간 이내의 지터에 따른 패킷 오류 등의 문제점들을 해결함을 확인할 수 있었다.

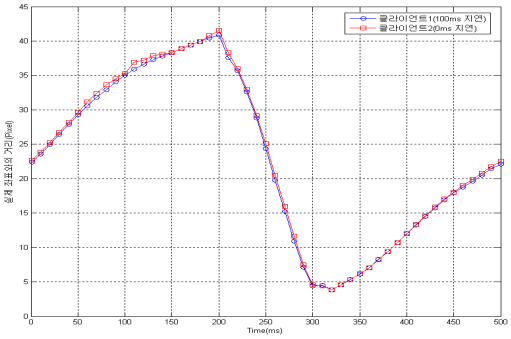

그림 10. 동기화(100ms 지연)

6. 결론 및 향후 연구 방향

본 연구를 통해 우리는 촉각 기반 가상환경 상에서 주요 논쟁점인 실시간성, 안정성 등이 네트워크 상황에 따라 어떻게 변화하는지 알 수 있었다. 또한 촉각 기반 네트워크 가상환경이 효율적이고 안정적으로 구동될 수 있는 전송 알고리즘을 제안하여 보았다. 협업 가상 환경에서 햅틱 협업은 네트워크 상황에 매우 민감하게 반응한다. 이는 햅틱 장치의 떨림 현상 혹은 클라이언트 간 비동시성으로 나타난다. 이를 해결하기 위해 손실이나 지터에 의해 실시간으로 확인이 불가능한 패킷의 경우에는 간단한 선형예측 방법을 사용하여 정확도를 높이고, 지연으로 인한 클라이언트 간 비동시성을 해결하기 위하여 완충시간을 주어 동기화를 시켰다. 본 논문에서 제안하는 알고리즘을 사용함으로써 햅틱 협업에 있어서 손실이나 지터에 의해 일어나는 햅틱 장치에서의 끊김 현상이 줄어드는 것을 확인 할 수 있었고 동기화의 성능이 향상되었음을 확인할 수 있었다.

향후에는 선형예측 알고리즘을 보완하여 정교한 예측 알고리즘을 적용하고, 서버-클라이언트간의 동기화 기법에 중점을 두어 연구를 진행할 것이다.

참고문헌

[1] Hosseini, Mojtabe et al. "A Haptic Virtual Environment for Industrial Training". In Proceedings of HAVE' 2002 IEEE Internatinal Workshop on Haptic Audia Visual Environments and their Applications; Ontario, Canada, November 2002

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

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

[4] Yutaka Ishibashi, Takahiko Hasegawa, and Shuji 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, 2004

[5] D. L. Stone and K. Jeffay, "An empirical study of delay jitter management policies," ACM Multimedia Systems, Vol. 2, No. 6, pp.267-279, Jan. 1995

[6] He, E., Alimohideen, J., Eliason, J., Krishnaprasad, N. K., Leigh, J., Yu, O., DeFanti, T. A. "Quanta a toolkit for hight performance data delivery over photonic networks". Future Generation Computer Systems, Special Issue:IGRID 2002, Vol.19(2003), Number 6, August 2003

[7] Mark, W., Randolph, S., Finch, M., Verth, J. V., and Taylor R. M. "Adding force feedback to graphics systems: Issues and solutions". SIGGRAPH 96 Conference Proceedings, pp.447-452, August. 1996

[8] Emanuele Ruffaldi, Dan Morris, Timothy Edmunds, Federico Barbagli and Dinesh K. Pai "Standardized Evaluation of Haptic Rendering Systems" Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems vol.0, 2006

[9] PHANTOM-Omni http://www.sensable.com/

[10] C. Cruz-Neira, D. Sandin, T. A. DeFanti. "Virtual Reality : The Design and Implementation of the CAVE". 93 Computer Graphics Conference, ACM SIGGRAPH, pp.135-142, August 1993

[11] ODE http://www.ode.org

[12] A. Boukerche, S. Shirmohamma야 and A. Hossain "A Prediction Algorithm for Haptic Collaboration" Proceedings of HAVE 2005. pp.154-158. October 2005

[13] "IEEE Standard for Information Technology-Protocols for Distributed Interactive Simulation Applications", IEEE 1278-1993, IEEE Computer Society, 1993.