

# Deep Learning Based Group Synchronization for Networked Immersive Interactions

Joong-Jae Lee<sup>†</sup>

## ABSTRACT

This paper presents a deep learning based group synchronization that supports networked immersive interactions between remote users. The goal of group synchronization is to enable all participants to synchronously interact with others for increasing user presence. Most previous methods focus on NTP-based clock synchronization to enhance time accuracy. Moving average filters are used to control media playout time on the synchronization server. As an example, the exponentially weighted moving average(EWMA) would be able to track and estimate accurate playout time if the changes in input data are not significant. However it needs more time to be stable for any given change over time due to codec and system loads or fluctuations in network status. To tackle this problem, this work proposes the Deep Group Synchronization(DeepGroupSync), a group synchronization based on deep learning that models important features from the data. This model consists of two Gated Recurrent Unit(GRU) layers and one fully-connected layer, which predicts an optimal playout time by utilizing the sequential playout delays. The experiments are conducted with an existing method that uses the EWMA and the proposed method that uses the DeepGroupSync. The results show that the proposed method are more robust against unpredictable or rapid network condition changes than the existing method.

Keywords : Group Synchronization, Deep Learning, Immersive Interaction

## 네트워크 환경에서의 몰입형 상호작용을 위한 딥러닝 기반 그룹 동기화 기법

이 중 재<sup>†</sup>

### 요 약

본 논문에서는 네트워크 환경에서 원격사용자들의 몰입형 상호작용을 위한 딥러닝 기반의 그룹 동기화 기법을 제안한다. 그룹 동기화의 목적은 사용자의 몰입감을 높이기 위해서 모든 참여자가 동시에 상호작용이 가능하게 하는 것이다. 기존 방법은 시간 정확도를 향상을 위해 대부분 NTP(Network Time Protocol) 기반의 시간 동기화 방식에 초점이 맞추어져 있다. 동기화 서버에서는 미디어 재생 시간을 제어하기 위해 이동 평균 필터를 사용한다. 그 한 예로서, 지수 가중평균 방법은 입력 데이터의 변화가 크지 않으면 정확하게 재생 시간을 추종하고 예측하나 네트워크, 코덱, 시스템 상태의 급격한 변화가 있을 때는 안정화를 위해 더 많이 시간이 필요하다. 이런 문제점을 개선하기 위해서 데이터의 특성을 반영할 수 있는 딥러닝 기반의 그룹 동기화 기법인 DeepGroupSync를 제안한다. 제안한 딥러닝 모델은 시계열의 재생 지연 시간을 이용하여 최적의 재생 시간을 예측하는 두 개의 GRU(gated recurrent unit) 계층과 하나의 완전 연결 계층으로 구성된다. 실험에서는 기존의 지수 가중평균 기반 방법과 제안한 DeepGroupSync 방법에 대한 성능을 평가한다. 실험 결과로부터 예상하지 못한 급격한 네트워크 조건 변화에 대해서 제안한 방법이 기존 방법보다 더 강건함을 볼 수 있다.

키워드 : 그룹 동기화, 딥러닝, 몰입형 상호작용

### 1. 서 론

최근 10여 년 동안 새로운 사용자 상호작용 장치의 출현과 네트워크 기술의 향상으로 구글 안경, 페이스북의 Meta, 마이크로소프트의 Mesh 그리고 SteamVR과 같은 다양한 가상현

실(VR, Virtual Reality), 증강현실(AR, Augmented Reality), 그리고 확장현실(XR, Extended Reality) 시스템들이 출현하고 있다[1-4].

코로나 팬데믹으로 비대면(Untact) 사회의 활성화와 함께 새로운 패러다임으로 등장한 메타버스(metaverse)는 초월을 의미하는 '메타(meta)'와 세계 또는 우주를 의미하는 '유니버스(universe)'를 합성한 신조어다[5]. 수많은 메타버스 플랫폼이 출시되고 있는 가운데 그 공통점을 살펴보면 현실과 같

<sup>†</sup> 정 회 원 : (재)실감교류인체감응솔루션연구단 책임연구원  
Manuscript Received: October 24, 2022  
Accepted: October 25, 2022  
\* Corresponding Author : Joong-Jae Lee(arbitlee@chic.re.kr)

은 가상세계에서 자유롭게 일하고, 배우고, 놀면서 협업도 할 수 있다는 점이다. 즉, 물리적으로 같은 공간에 있지 않아도 이러한 공유 가상공간에서 상호작용을 할 수 있다. 그래서 사람들은 마치 한 공간에서 같이 보고, 듣고, 만져보고 느낄 수 있는 활발한 상호작용에 대한 관심이 늘어나고 있다. 이러한 양방향 상호작용이 위화감 없이 자연스럽게 이루어지기 위해서는 상호작용에 필요한 미디어 동기화뿐만 아니라 사용자 간의 그룹 동기화 기술이 요구된다. 시간상으로 동기화되지 않으면 한 공간에 함께 있는 환경을 구축할 수 없기 때문이다. 본 논문에서는 기존의 알고리즘 기반 그룹 동기화의 약점을 극복하기 위해서 데이터 중심의 새로운 딥러닝 기반 그룹 동기화 방법을 제안한다. 이를 통해 예측 불가능한 갑작스러운 네트워크 변화 상황에도 대처할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 네트워킹 환경에서 몰입형 상호작용을 제공하는 데 필요한 미디어 동기화 방법과 클럭 스큐 보정(clock skew compensation) 방법, 그리고 그룹 동기화에 관련된 기존 연구내용을 소개하고 3장에서는 예측이 어려운 급격한 네트워크 환경의 변화로 발생하는 동기화 오차를 최소화하기 위한 딥러닝 기반의 그룹 동기화 기법에 관해서 설명한다. 4장에서는 실험을 통해 제안한 방법과 기존 방법의 성능을 분석하고 5장에서 결론을 맺는다.

## 2. 관련 연구

미디어 동기화 방법은 여러 방식으로 분류할 수 있다. 한 예로 합성 동기화(synthetic synchronization) 또는 라이브 동기화(live synchronization)로 구분할 수 있다[6]. 전자는 네트워크를 통해 전송된 미디어가 수신되는 즉시 재연(rendering)되어야 하고 저장된 미디어와 동기화하여 재연시키는 경우이다. 이에 반해서 라이브 동기화는 여러 종류의 미디어가 네트워크를 통해 전송되고 동기화되어 함께 재연되는 방식이다.

다양한 센서 데이터를 실시간으로 송수신하는 동기화 응용 프로그램은 클럭의 정확도가 매우 중요하다. 미디어 동기화의 성능은 시스템마다 클럭 차이를 보정하여 같은 시간에 미디어를 획득, 전송 및 재연할 수 있느냐에 좌우되기 때문이다. 대표적인 클럭 보정 방법으로는 NTP(Network Time Protocol) 또는 PTP(Precision Time Protocol)와 같은 네트워크 프로토콜이 있다[7, 8].

한편 Fig. 1은 NTP 또는 PTP를 이용하여 기준 클럭과의 오프셋(offset)을 맞추기 위해 수행하는 클럭 스큐 보정이 어떤 부작용을 가져올 수 있는지를 보여준다. Fig. 1(a)는 NTP 서버와 시간 오프셋이 없으므로 제시간에 재연된다. 그러나 Fig. 1(b)의 경우에는 클라이언트 B에 대해 양(positive)의 오프셋을 가지므로 클럭 스큐를 보정하면 첫 번째 프레임을 재생하기 전에 미디어 재생이 정지되거나 시간이 초과하게 된다. Fig. 1(c)는 오프셋이 음(negative)인 경우로 첫 번째

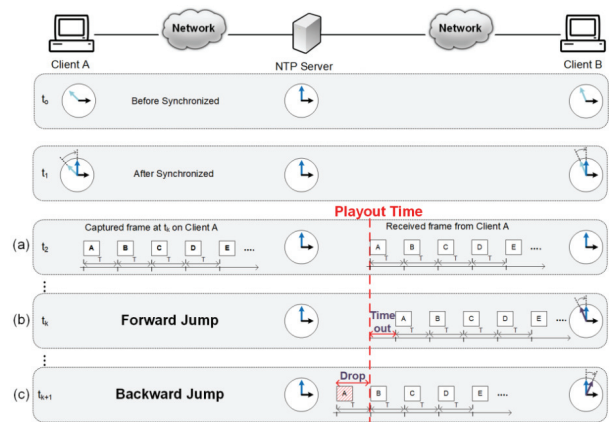


Fig. 1. Clock Skew Compensation for Client B's Clock using a NTP Server with (a) No Offset, (b) Positive Offset, and (c) Negative Offset Compared to the NTP Servers' Clock, Respectively

프레임을 재생할 때 건너뛰거나(jump) 재생 시간을 지나쳐서 프레임을 버려야 한다(drop). 이런 문제점을 개선하기 위한 연구로서 빈번한 클럭 스큐 보정으로 인한 미디어의 재연 품질을 떨어뜨리는 부작용을 최소화하기 위한 균형(Balanced) 클럭 스큐 보정 방법이 있다[9]. 이 방법은 상호작용 응용 프로그램에서 요구하는 동기화 수준에 따라서 클럭 스큐 보정 시점을 조절한다. 고정 주기에 클럭 스큐를 보정하지 않고 미디어 간 동기화를 인간이 인식할 수 있는 범위에 따라서 동적으로 보정을 수행하는 방식이다.

동기화 기법을 분류하는 다른 예로는 송신자(sender)기반 또는 수신자(receiver)기반 방법이 있다[10]. 송신자 측에서 먼저 미디어를 동기화한 후에 네트워크를 통해 전송하여 수신 직후에 재연하는 방식과 미디어가 전송된 후에 동기화하여 재연하는 방식이다. 여기서 송신자기반 방법은 네트워크의 신뢰성 보장되거나 전송 지연을 무시할 수 있는 경우에 사용된다. 그러나 대부분의 인터넷 환경은 이런 조건을 기대하기 어려우므로 수신자기반 방법을 통한 동기화 기법이 많이 사용되고 있다.

그룹 동기화에 관한 응용사례를 살펴보면 원격 교육 시에 발생하는 그룹 동기화 오차에 대한 인간의 인지 결과 분석에 관한 연구가 있다[11]. 여기서 원격 교육은 단방향 통신으로 이루어지는데 이때 비디오와 오디오 간의 의존성을 콘텐츠의 단방향 통신방식으로 분석하였다. 실험 결과, 두 종류의 미디어 간의 느슨한 결합보다는 밀접한 결합이 요구되는 콘텐츠에서 동기화 오차를 더 쉽게 인지할 수 있음을 보였다. 그러나 이 연구는 원격사용자 간의 양방향 통신을 고려하지 않았다. 다른 예로서, 그룹 동기화를 인간과 로봇이 협업할 때 조화롭게 작업을 수행할 수 있도록 하는 문제에 적용한 사례이다. Givigi et al.는 네트워크 제어 방법을 통해서 조립 또는 물건 이송 작업에서 인간과 로봇 사이의 동작을 동기화하는 방법을 제안하였다[12]. 이 방법은 최적 제어에 필요한 매개변수를 조절할 때 딥러닝 기반 방법이 학습에 필요한 훈련 데

이터 획득에 쉽지 않고 학습 시간이 많이 든다는 점을 해결하기 위해서 고안되었다. 그러나 이 그룹 동기화 방법은 주로 동작 데이터에 국한되기 때문에 다양한 미디어에 대한 동기화 방법으로 확장하기는 적합하지 않다.

### 3. 딥러닝 기반의 원격사용자 그룹 동기화 기법

#### 3.1 원격사용자들의 상호작용을 위한 그룹 동기화

그룹 동기화란 원격사용자 간의 상호작용에 필요한 다양한 미디어에 대한 미디어 내(intra media) 동기화, 미디어 간(inter media) 동기화를 포함하여 그룹 내 모든 사용자에게 대한 가장 높은 수준의 동기화를 의미한다.

Table 1은 본 논문에서 사용하는 용어와 그에 대한 설명을 요약한 것이다.

Fig. 2는 본 논문에서 구현한 그룹 동기화의 전체 구조로서 2개 이상의 클라이언트, 클럭 동기화를 위한 NTP 서버, 미디어 데이터 전송을 위한 릴레이 서버(Relay server), 그룹 동기화에 필요한 playout delay를 조절하는 동기화 서버(SyncServer)로 구성된다. 각 클라이언트는 데이터 캡처링, 네트워킹, 코덱, 동기화, 렌더링 소프트웨어 모듈을 포함한다. 미디어 종류에는 비디오, 오디오, 햅틱 데이터가 있다.

Fig. 3은 앞서 언급한 동기화 서버의 역할을 개념적으로 보여준다. 동기화 서버는 모든 클라이언트에서 전송되는 미디어를 동시에 재연할 시간인 PT(playout time)를 결정한다. 이때 네트워크, 코덱, 시스템 상태에 따라서 발생하는 지

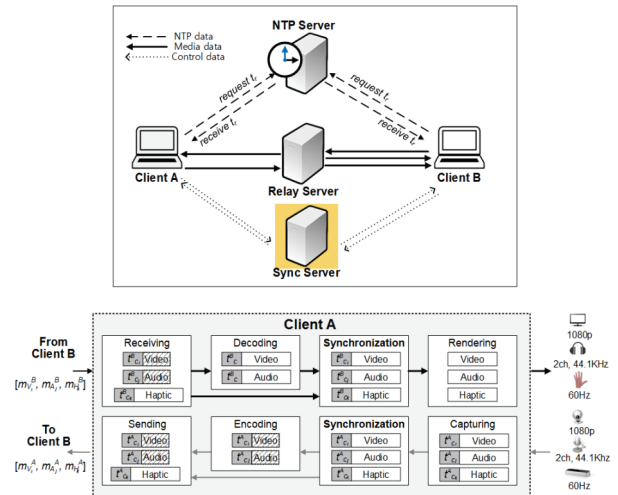


Fig. 2. Overall Architecture for Group Synchronization

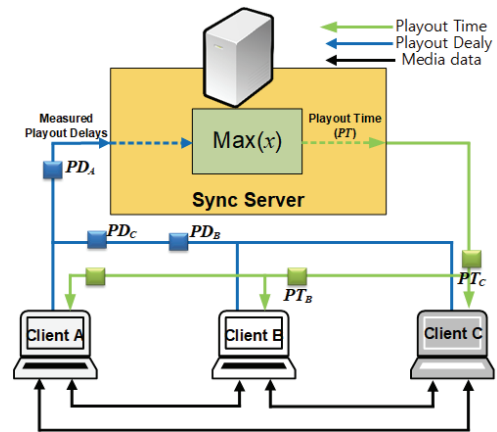


Fig. 3. Conceptual Diagram for Group Synchronization based on Sync Server

Terminology	Description
mvB	video data received from clientB (V: Video, A: Audio, H: Haptic)
PD	playout delay which means the time elapsed between a given media frame is captured at the sender side and right before it is ready to be rendered at the receiver side.
PT	the point in time where the media is scheduled to render.
Playout Scheduler	a module that imposes a fixed playout time for each media frame with respect to the predetermined playout delay of the transmission.
Playout Controller	a module that monitors the playout delay of each media stream and determine the playout time of each media.
Intra Playout Controller	a module that controls the playout delay between each media frame.
Inter Playout Controller	a module that controls the playout delay between two or more media frame.
Group Playout Controller	a module that controls the playout delay between two or more clients.

연 시간의 변화를 반영한다. 구체적으로 그림에서 보듯이 각 클라이언트로부터 PD(playout delay)를 전송받아 가장 큰 값으로 PT를 결정하여 이 결과를 모든 클라이언트에 전송, 갱신하는 방식으로 그룹 동기화를 수행한다.

그룹 동기화를 수행하기 위해서는 각 클라이언트의 PD 값을 제어하지만 응용 서비스의 특성에 따라 목표로 하는 동기화 수준을 조절할 필요가 있다. 이런 경우에는 고정적인 PD 값을 계산하여 전달하는 일반적인 동기화 서버는 적합하지 않다. Fig. 4는 평균, 최대, 최소, 중간값 등의 다양한 연산을 통해 PT 값을 계산하는 방법을 보여준다. 본 논문에서는 PD 값을 제어하기 위해서 지수 가중평균(EWMA, exponentially weighted moving average) 기반 방법을 사용한다[13].

#### 3.2 DeepGroupSync: 딥러닝 기반 그룹 동기화 모델

기존의 EWMA 기반 방법은 알고리즘 주도적 방식인 데 반해서 본 논문에서는 데이터 주도적 방식인 딥러닝을 그룹 동기화에 적용한다. 급격한 네트워크 환경의 변화, 코덱 또는

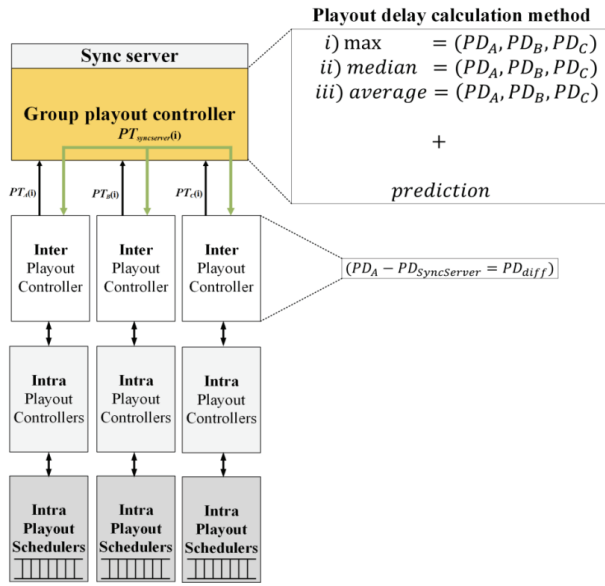


Fig. 4. Group Playout Delay Calculation in Sync. Server

시스템 부하로 인해 PD 값의 변동 폭이 큰 경우 EWMA 기반 방법은 데이터 변화 동향을 반영하는 방식이므로 이러한 상황에 빠르게 대처할 수 없다. 이런 점을 해결하기 위해 본 논문에서는 데이터의 특성을 잘 반영할 수 있는 딥러닝 기반 그룹 동기화 방법을 제안한다.

Fig. 5는 제안한 방법에 대한 개념도이다. 제안한 딥러닝 모델 DeepGroupSync의 입력값은 각 클라이언트로부터 t 시점에 계산된 각 미디어의 PD 값이며 출력값은 입력값으로부터 계산한 t+1 시점 PD에 대한 예측값이 된다. 즉, 딥러닝 모델을 통해 t+1 시점의 최적해(optimal value)를 도출하고 동기화 서버와 연결된 모든 클라이언트에 전송하여 재연시간을 조절한다. 각 클라이언트에서 수신하는 각 미디어의 PD 값은 다음 Equation (1)과 같다.

$$PD(t) = t - S(t) \tag{1}$$

여기서, PD(t), S(t)는 각각 t 시점에 수신한 미디어의 PD 값과 미디어 전송시간을 의미한다.

최적의 예측 PD 값은 모든 미디어 데이터를 재연하는데 필요한 최소 지연 시간이며 동시에 모든 미디어가 동기화되는 시간으로서 Equation (2)와 같이 정의할 수 있다.

$$PD^-(t+1) = \max \{ \forall PD(t) \} \tag{2}$$

여기서, PD-(t+1), PD(t) 는 각각 t+1 시점의 예측 playout delay 값, t 시점의 PD 값을 나타낸다.

Fig. 6은 제안한 DeepGroupSync 모델의 자세한 내부 구조를 보여준다. 다양한 딥러닝 모델 중 시계열 데이터를

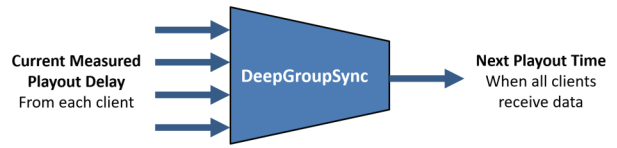


Fig. 5. Conceptual Diagram for Deep Learning based Group Synchronization

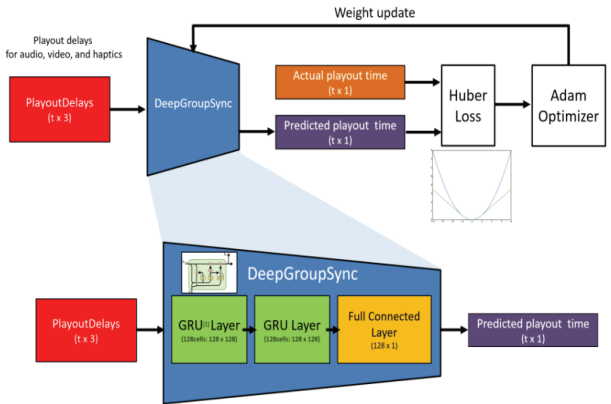


Fig. 6. Overall Architecture for Proposed Model for Group Synchronization, DeepGroupSync

학습할 수 있는 RNN(recurrent neural network)을 사용하는데 구체적으로 GRU(gated recurrent unit)를 사용한다. RNN은 주로 텍스트와 같이 위치 정보에 의존적인 데이터 또는 비디오와 음성같이 시간 의존적인 데이터 학습에 사용한다. DeepGroupSync는 시간에 따른 PD 값을 예측하므로 이 중에서 시간 의존적 데이터를 학습하는 RNN을 사용할 수 있다. 일반적으로 딥러닝 분야에서 RNN은 초기에 LSTM(long short-term memory) 모델을 많이 사용하였으나 DeepGroupSync 모델에서는 LSTM의 간소화된 버전인 GRU를 사용한다[14]. GRU는 LSTM과 비슷한 성능을 내면서 학습 시에 더 빠르게 수렴한다는 장점이 있다. 제안한 모델에서는 GRU를 2계층(layer)을 사용하기 때문에 단일 계층을 사용하는 것보다 데이터의 더 고차원적인 특성을 학습할 수 있다. 또한, 3계층 이상을 사용하면 일어날 수 있는 컴퓨터 시스템 자원 부족 현상을 피할 수 있다.

Fig. 6에서 볼 수 있듯이 제안한 모델을 학습하기 위해서 본 논문에서는 다양한 손실함수(loss function) 중에서 Equation (3)과 같은 Huber 손실함수를 사용한다[15]. 이것은 Huber 손실함수가 L1 norm이 이상치(outlier)를 효과적으로 배제할 수 있고 L2 norm은 정상치(inlier)를 최적화할 수 있는 특성을 반영할 수 있는 점을 고려한 것이다.

$$Huber(x) = \begin{cases} \frac{1}{2}x^2 & , |x| \leq 1 \\ |x| - \frac{1}{2} & , otherwise \end{cases} \tag{3}$$

#### 4. 실험 및 성능평가

본 장에서는 실제 네트워크 환경에서 원격사용자 간의 몰입형 상호작용 실험에서 획득한 데이터를 통해서 제안한 딥러닝 모델을 학습한 후에 그룹 동기화 방법의 성능을 평가한 내용을 설명한다.

##### 4.1 실험 환경 및 학습데이터 획득

###### 1) 실험 환경

Fig. 7과 같이 가상의 공유공간에서 원격사용자들이 상호작용하는 환경이다. 각 클라이언트는 1920×1080 해상도로 55인치 3D TV에 가상공간을 디스플레이 한다. 실험에 사용한 PC의 주요 사양은 Intel Core i7-6850K CPU, 32GB 메모리, NVIDIA GTX-1080이고 운영체제로는 Windows 10을 사용하였다. 릴레이 서버와 동기화 서버는 Intel E5-1067 CPU, 32GB 메모리, NVIDIA 300 시리즈 그래픽 카드를, 그리고 운영체제는 Windows Server 2008 R2를 사용하였다. 또한 NTP 서버는 Meinberg社의 LANTIME M1000으로 클럭 동기화에 필요한 기준 시간을 제공한다. 실험에서 몰입형 상호작용에 사용하는 비디오(30fps), 오디오(43fps), 햅틱 데이터(60fps)를 획득하기 위해 각각 full-HD 카메라, 마이크/헤드폰 콤보 장치, 그리고 HTC Vive 컨트롤러와 같은 입출력 장치를 사용하였다. 각 클라이언트에서 획득된 미디어 데이터는 릴레이 서버를 거쳐 다른 모든 클라이언트에 전송된다.

Fig. 8은 “다 함께 손”이라는 실험 시나리오의 한 장면으로 해당 애플리케이션은 Visual C++ 2013으로 구현하였다. 각 클라이언트의 3D TV에서 가상 손 모델과 빨간색 가상상자로 구성된 가상환경을 볼 수 있다. 이때 가상 손 모델은 사용자의 Vive controller 동작과 연동되어 이동한다. 두 사용자의 손이 함께 빨간색 상자 안에 있으면 오렌지색으로 상자 색상이 변경되고 Vive controller에 진동이 발생한다. 동시에 실시간으로 스트리밍되는 비디오 영상도 가상환경 옆에 디스플레이 된다. 이런 방식으로 비디오 영상에 보이는 사용자의 실제 물리적인 손동작과 가상공간에 재연되는 Vive controller에 의해 추적된 가상 손동작의 시간 차이를 통해 동기화 정확도에 대한 성능평가를 수행하였다.

###### 2) 학습데이터 획득

딥러닝 모델을 학습하기 위해서는 데이터 획득이 매우 중요하다. 영상, 음성, 텍스트 등은 다양한 오픈 데이터가 존재하지만, 본 논문의 실험에서 필요로 하는 동기화 데이터는 부족한 상황이다. 따라서 본 논문에서는 Fig. 9와 같이 딥러닝 모델을 학습하기 위해서 동기화 모듈의 세부 모듈인 Playout Scheduler에서 미디어 데이터와 타임스탬프(timestamp)를 저장하고 또 다른 세부 모듈인 Playout Controller에서 EWMA 계산 모듈 전후에서 각각 playout delay 값을 저장한다. Fig. 10은 실제로 획득한 학습데이터의 한 예를 보여준다.



Fig. 7. Experimental Environment. Each Client Captures and Sends Video, Audio, and Haptics Data to the Other Client

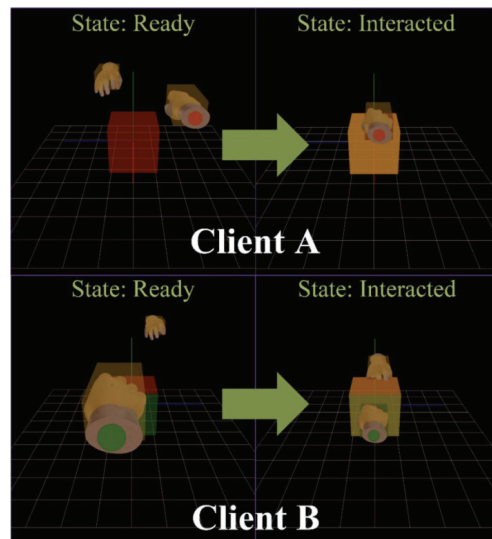


Fig. 8. Virtual Space (“Hands Together”) Displayed in Each Site. The Top and Bottom Halves Show the View from Client A and Client B, Respectively

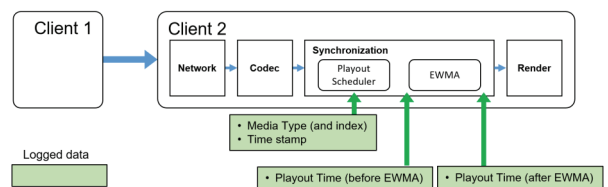


Fig. 9. Data Logging Points and Data Types in a Client

Fig. 11은 2시간 동안의 각 미디어 데이터에 대한 PD 값의 변화를 보여준다. 여기서 EWMA 기반 방법은 이동 평균을 이용하는 방식으로 PD 값의 변화에 대한 예측값의 정확도 측면에서 볼 때 비교적 큰 잡음이 없을 때는 실제 값에 근접하게 추정하였음을 볼 수 있다.

그러나 Fig. 12에 원으로 표시한 구간처럼 입력 데이터에 급격한 잡음이 발생하면 빠르게 대응할 수 없다. 그 이유는 EWMA 알고리즘의 특성상 다시 안정상태로 수렴하는 데까지 시간이 걸리기 때문이다.

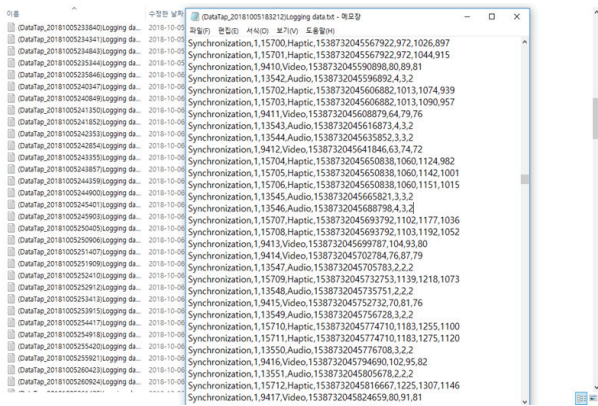


Fig. 10. An Example of Data Logging

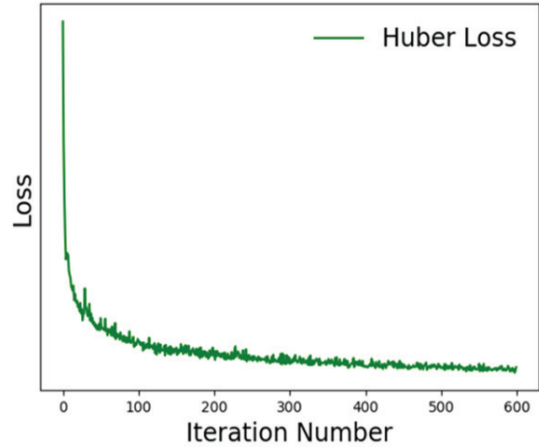


Fig. 13. The Plot of Training Loss: The Training Loss Goes Down Over Time

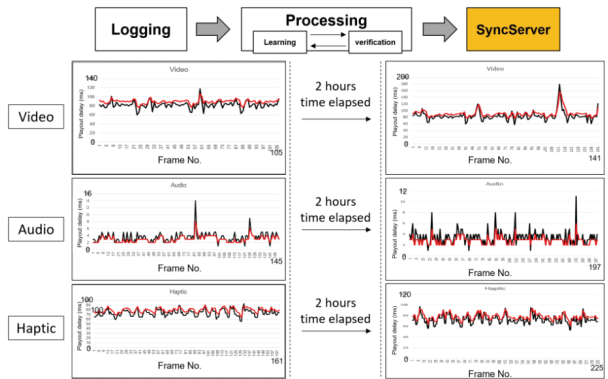


Fig. 11. The Plot Shows the Variation of Playout Delays for Each Media

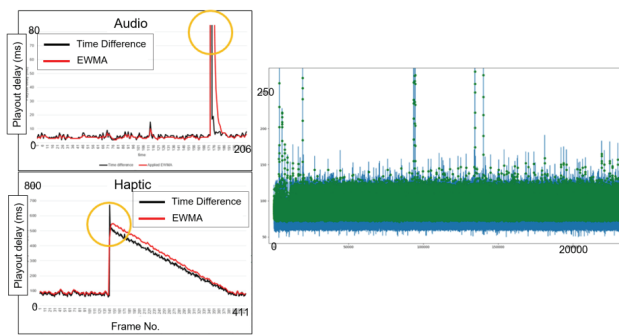


Fig. 12. Predicted Playout Delays based EWMA Compared to Actual Playout Delays

4.2 실험 결과

제안한 방법의 성능을 평가하기 위해 앞서 소개한 실험 시나리오에 대해 5시간 동안 총 1,787,816개의 PD 값을 저장하여 전체 데이터의 80%(1,430,252개)를 딥러닝 모델 학습에 사용하였고 나머지 20%(357,564개)로 성능을 검증하였다. Fig. 13에서는 600회에 걸쳐 딥러닝 모델을 학습시켰을 때의 손실(Huber loss)함수 값이 학습 횟수가 증가할수록 감소함을 볼 수 있다.

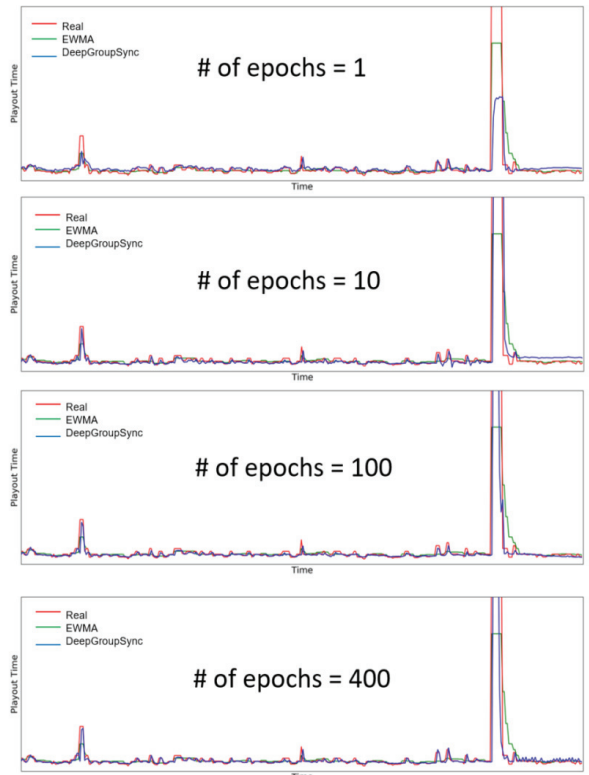


Fig. 14 The Experimental Results According to the Number of Iterations

Fig. 14는 학습 횟수에 따라서 실제 참값(빨간색), EWMA 기반 방법(녹색), 제안한 방법(파란색)의 playout delay 값을 비교한 결과이다. 전반적으로 살펴보았을 때 EWMA 기반 방법보다 제안한 방법이 실제 값을 더 정확하게 예측함을 알 수 있다.

특히, Fig. 15에서 확대된 (a)-(c) 구간처럼 갑작스러운 PD 값의 큰 변화가 일어났을 때 기존의 EWMA 기반 방법은 실제 값에 추종하여 안정화되기까지 시간이 오래 걸렸지만

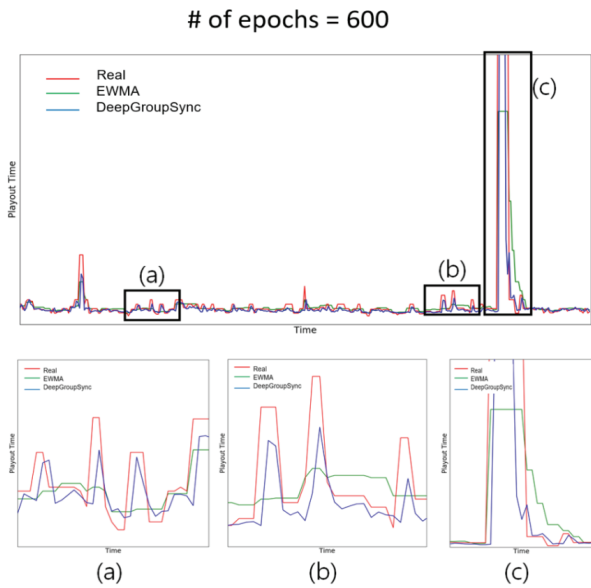


Fig. 15. Comparison Result of Two Methods: EWMA-based Method vs. Proposed Method

제안한 방법은 네트워크의 변화 특성을 학습하였으므로 더 빠르게 실제 값을 추종할 수 있음을 볼 수 있다.

### 5. 결론 및 향후 연구

본 논문에서는 네트워크, 코덱, 시스템 부하로 인한 급격한 잡음이 발생할 때 원격사용자들의 동기화된 몰입형 상호작용을 지원하는 딥러닝 기반의 그룹 동기화 기법을 제안하였다. 기존 방법은 시계열 입력 데이터가 점진적으로 변경될 때 좋은 예측 결과를 보이나 상당히 큰 잡음이 포함된 입력 데이터에 대해서는 대응력이 현저하게 떨어진다. 제안한 방법은 이런 문제점을 해결하기 위해서 입력 데이터 속성을 학습하여 반영할 수 있는 딥러닝 기반 모델인 DeepGroupSync를 제안하였고 실험을 통해서 기존 방법보다 네트워크의 변화 특성을 더 잘 반영함을 확인하였다. 그러나 제안한 방법은 학습에 필요한 실시간 미디어 정보를 모든 클라이언트와 동기화 서버로부터 획득해야 하므로 학습데이터 준비과정이 복잡한 애로사항이 있다. 향후 연구에서는 첫째, 학습데이터 획득 과정을 자동화하는 연구, 둘째 서로 다른 동기화 정확도가 있어야 하는 다양한 상호작용 콘텐츠에 대한 추가 실험을 통해서 제안하는 방법의 확장성과 그에 따른 성능향상에 관한 연구를 진행하고자 한다.

### References

[1] Google [Internet], <https://developers.google.com/glass/>, visited in October 2022.  
 [2] Meta [Internet], <https://www.meta.com/en/>, visited in October 2022.

[3] Microsoft [Internet], <https://www.microsoft.com/en-us/mesh> visited in October 2002.  
 [4] SteamVR [Internet], <https://steamcommunity.com/steamvr>, visited in October 2022.  
 [5] J. D. N. Dionisio, W. G. Burns III, and R. Gilbert, "3D virtual worlds and the metaverse: Current status and future possibilities," *ACM Computing Surveys (CSUR)*, Vol.45, No.3, 2013, pp.1-38.  
 [6] R. Steinmetz and M. Thomas, "Multimedia synchronization techniques: Experiences based on different system structures," *ACM SIGCOMM Computer Communication Review*, Vol.22, No.3, pp.90-91, 1992.  
 [7] D. L. Mills, "Network time protocol version 4 reference and implementation guide," *Electrical and Computer Engineering Technical Report 06-06-1*, 2006.  
 [8] K. B. Lee and J. Eldson, "Standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Standard 1588*, 2004.  
 [9] T. Y. Lee, E. Lee, J. Surh, J. J. Lee, and B. J. You, "Balanced clock skew compensation for immersive networked interactions based on inter media synchronization levels," In *2018 10th Computer Science and Electronic Engineering (CEECE)*, pp.119-124, 2018.  
 [10] Z. Huang, K. Nahrstedt, and R. Steinmetz, "Evolution of temporal multimedia synchronization principles: A historical viewpoint," *ACM Transactions on Multimedia Computing Communications and Applications*, Vol.9, No.1s, pp.1-23, 2013.  
 [11] H. M. M. M. Lwin, Y. Ishibashi, and K.T. Mya, "Human perception of group synchronization error in remote learning dependencies of voice and video contents in one-way communication," *International Journal of Communications, Network and System Sciences*, Vol.15, No.3, pp.31-42, 2022.  
 [12] S. N. Givigi, K. M. Cabral, and P. T. Jardine, "Group synchronization in coordination tasks via network control methods," *IFAC-PapersOnLine*, Vol.53, No.2, pp. 10182-10187, 2020.  
 [13] P. Cisar, S. Bošnjak, and S. M. Cisar, "EWMA algorithm in network practice," *International Journal of Computers Communications & Control*, Vol.5, No.2, pp.160-170, 2010.  
 [14] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.  
 [15] P. J. Huber, "Robust estimation of a location parameter," *Breakthroughs in statistics* (pp.492-518), Springer, New York, NY, 1992.



### 이 중 재

<https://orcid.org/0000-0001-7732-6162>

e-mail : arbitlee@chic.re.kr

1999년 숭실대학교 컴퓨터학부(학사)

2001년 숭실대학교 컴퓨터학과(석사)

2005년 숭실대학교 컴퓨터학과(박사)

2005년 ~ 2006년 숭실대학교 교육대학원  
연구교수

2006년 ~ 2010년 한국과학기술연구원 인지로봇연구단 박사후연구원

2010년 ~ 2013년 LG전자기술원 미래IT융합연구소 책임연구원

2013년 ~ 2021년 (재)실감교류인체감응솔루션연구단 책임연구원

관심분야: Networked Virtual/Augmented Reality,

Computer/Robot Vision, ADAS