A Conceptual Framework of Delay Perception in 3D Tele-Immersion[*]

Extended Abstract[†]

Leave Authors Anonymous
Institute
City, Country
example@email.com

Leave Authors Anonymous
Institute
City, Country
example@email.com

Leave Authors Anonymous
Institute
City, Country
example@email.com

ABSTRACT

3D Tele-Immersion (3DTI) allows distributed users to communicate in a same virtual space. It grows rapidly in recent years. However, few works were conducted to study user experience in 3DTI. Delay is an important factor that affects user experience. In this paper, we explore users' perception of network delay in 3DTI. We propose a conceptual framework that levels delay requirements of 3DTI tasks into three classes: *synchronous* tasks, *audiovisual* tasks and *visual Only* tasks. *synchronous* tasks involves interactions at the same time, e.g., instrument ensemble and the Rock-Paper-Scissors game. They require a low delay of 30 ∼ 100 ms. We suggest that 3DTI supports much more tasks of this delay-sensitive level compared to the 2D situation. The other two levels are common in 2D, e.g., teleconference and playing chess. We suggest that they require a delay of 200 ms or more in 3DTI. The requirement is lower because the users are less sensitive and more tolerable to delay in 3D. We describe an example study to illustrate our framework. For each level, the framework gives suggestions of network engineering to save network resources and improve the user experience.

CCS CONCEPTS

• **Human-centered computing** → *HCI theory, concepts and models*;

KEYWORDS

Delay perception; 3D tele-immersion

ACM Reference Format:
Leave Authors Anonymous, Leave Authors Anonymous, and Leave Authors Anonymous. 2019. A Conceptual Framework of Delay Perception in 3D Tele-Immersion: Extended Abstract. In *Proceedings of ACM SigCHI conference (CHI'19).* ACM, New York, NY, USA, 13 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

The past centuries have witnessed the growth of communication technology. The invention of the telephone has saved a great deal of time and money by displacing meeting. Recently, Distributed Interactive Multimedia Environments (DIMEs) is getting popular. It provides convenience for teleconference [46], tele-collaboration [2, 14], robotic telepresence [32, 47, 49], and so on. Development of communication technology is never separated from the studies of user experience. For example, delay of 150 ms provides a good user experience for most audio-mediated applications [14, 62]. It has become an industrial standard that contributes to telephone network engineering [28]. 2D DIME also benefits from the studies of user experience, e.g., in the regions of pointing [2, 25], distance perception [1, 6] and delay perception [19, 66, 70].

3DTI emerged in the last past decades [39, 43, 44, 59]. Microsoft Research's Holoportation [54] was impressive. They presented an end-to-end 3DTI pipeline with high-quality, real-time reconstructions of an entire space. Furthermore, parallel computing devices such as GPUs are getting more powerful. Immersive displays such as Head-Mounted Displays (HMDs) are becoming popular. In a word, both the improvements of algorithm and hardware make 3DTI hopeful to be practical in the near future.

The focus of previous 3DTI works has been mainly on technical implementations. However, few works were conducted to study user experience in 3DTI. In particular, no work has been done to study users' perception of network delay in an advanced 3DTI system.

Network delay is a crucial factor that affects user experience [7, 65, 67, 75]. The studies of delay perception is useful. Numerous works have been carried out to explore delay perception in telephone and 2D DIME. Given a specific task, noticeability and acceptance of delays are important factors to measure [19, 65, 66, 75]. On the one hand, a network service should achieve the acceptable delay as far as possible. On the other hand, there is no need to improve the network while the delay is already unnoticeable. Beyond that, many

[*] Produces the permission block, and copyright information
[†] The full version of the author's guide is available as `acmart.pdf` document

CHI'19, April 2019, ElPaso, Texas USA
2019. ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

other strategies were proposed to save network resources and improve the user experience.

Despite of the sufficient research in 2D, it is still necessary to rebuild the framework of delay perception in 3D. The reason is the large difference between 3DTI and 2D DIME: First, 3DTI can support more applications and improve some existing tasks in a more nature manner. We have to discuss them case by case; Second, 3DTI refers to a higher level of immersion, which offers more visual cues. Previous work [70] have pointed out that video increase users' tolerance to delay. This effect may be enhanced in 3D.

In this paper, we propose a conceptual framework of network delay perception in 3DTI. It levels delay requirements of 3DTI tasks into three classes: *synchronous* tasks, *audio-visual* tasks and *visual only* tasks, which require network delays of about 50 ms, 200 ms and 400 ms respectively. We designed the framework through a comprehensive review on delay perception and 3DTI systems. For each level in our framework, we summarized suggestions on network engineering. To validate the framework, we first followed the mainstream works to build up our 3DTI systems. Then, we conducted a controlled study to illustrate our framework. To our knowledge, we are the first to investigate users' perception of network delay in a full 3D tele-immersion system.

The contribution of our work is threefold: First, the framework infers a significant change of network delay perception in 3D. We recommend that the 3DTI developers should first assess his application through our framework; Second, we summarize suggestions on network engineering to cope with different level of tasks. These suggestions can help saving network resource and improving the user experience; Third, our project is open-source [?]. We give necessary explanation in the system overview to make sure that the readers can easily build up a similar system.

We construct the paper as follow: In section 2, we present our framework. In section 3, we give an overview of our experimental system. In section 4, we describe the controlled study to illustrate our framework. In section 5, we supplement related works on system implementation and existing works on 3DTI delay perception. In section 6, we discuss the limitation of our work. At the end we draw the conclusion.

2 A CONCEPTUAL FRAMEWORK OF DELAY PERCEPTION IN 3DTI

We present a conceptual framework of delay perception in 3DTI. The framework divides 3DTI tasks into three levels: tasks with *synchronous interaction*, *conversation* and *only visual feedback*. Correspondingly, the three levels of tasks call for high, middle and low requirement of network delay. For each level, the framework also gives suggestions to improve the perceived quality of network.

The framework is supported a comprehensive reviews of previous works. Delay perception is a well-understood research question in audiovisual DIME. **Our framework partly relies on previous theories and study results**（这句话不清楚）. However, there is a large difference of delay perception between 3DTI and audiovisual DIME. Thus, we take the features of 3DTI into account in order to form our framework.

Noticeability and tolerance of delay are two important factors that were measured by most studies [19, 65, 66, 75]. Some studies also focus on users' perception of overall network quality. These metrics are users' subjective rating in a specific task, and with a specific delay. In our study, we assess subjective feedback via questionnaires similar to [66]. The questionnaire is on a 5-point Likert scale (Table 5).

Label	Question	Scale
quality	How do you feel during the experiment?	Bad <−> Excellent
noticeability	Can you perceive the delay in the connection?	Very much <−> Not at all
tolerance	To what extent where you annoyed by the delay?	Severe annoyance <−> No annoyance

Table 1: Questions and scale.

A 3DTI developer may expect a certain recommended delay for his application. However, previous works [48] [?, ?, ?] pointed out that delay perception is largely dependent on user differences and context. Thus, we should investigate the noticeable and tolerable boundary of delay, which is statistically suitable for most users. We first refer to a psychology concept called Just Notice Difference (JND) [64, 76]:

- *JND*: With other variables fixed, the value for which 50% of the subjects perceive a difference in their quality.

Most related studies recommend using *noticeable delay* and *tolerable delay* as certain values in discussion. We define them as follows:

- *Noticeable Delay*: the threshold delay that most users can just perceive. In our experiment, we define it as the 50% JND of zero delays, i.e., more than 50% of participants score 1 point for noticeability.
- *Tolerable Delay*: the threshold delay that most users can just tolerant. In our experiment, we define it as the just intolerable delay minus its 50% JND, i.e., less than 50% of participants score 4 or 5 points for disruptiveness.

We illustrate the relationship between noticeable delay and tolerable delay in figure xxx.

The noticeable delay is very insightful for network engineering. On the one hand, developers should try to improve the network service, to reach the noticeable delay. On the other hand, when a service is already within noticeable delay, we can appropriately increase the delay to have more room for smoothing or recovering packet loss [76].

[NOTE] Tolerable delay indicates a boundary that is nearly intolerant. An application can not simply target at it, because it is already a bad service. Instead, the tolerable delay can be used to assess the Quality of Experience (QoE). [?] suggests a linear correlation between end-to-end delay and user experience. We can use noticeable delay and tolerable delay to determine the correlation.

[NOTE from zsyzgu] Here I change my mind about the tolerant delay. Delay can be perceived by cues for sure, however, the tolerance of delay in a specific task may depend on much more factors such as fairness and interactivity [27, 48]. So we are not going to model the tolerant delay anymore, but list factors to affect it according to previous work. We will also introduce a way to measure tolerance of a specific delay in a specific task, in our example experiment.

We next introduce the three synchronization levels. The basic idea is the observation that user perceive delay by cues. We determine the level of a task by judging if it contains cues of *synchronous interaction*, *conversation* and *visual feedback* 这个是按照什么来分的？看关键词感觉不到维度. As Table 2 shown, we recommend their noticeable delay and tolerable delay by summarizing previous audiovisual works and adapting to the 3D situation （从audiovisual adapt到3D的策略和原则是什么？）.

[NOTE from LZP] In [65], the trial only has three delays, 0ms, 500ms and 1000ms. But it has three kinds of delay, symmetric, moderator and random. Meanwhile, paper[64] is similar to this one. In [70], participants rate the conversation on five-point scale items, but the results are all between 3.5 and 4.5. So I pick the biggest delay as the MOS 3.5 delay. For the rest of examples in conversation section, I can't summarize them. And in [37], the MOS is in the range of 0 to 4, but I think it is the same as 0 to 5.

1. Synchronous Interaction

synchronous gesture. If two distributed users have to gesture exactly at the same time, they may be able to perceive delay like checking their own movement. As [52] explained, 100 ms is an upper boundary for users to fell that the system is running instantaneously. For a better performance, a delay of 30 to 50 ms is needed [11].

Imaging that a pair is playing rock-paper-scissors in a 3DTI system with a delay of 100 ms. They expect to perform the gesture exactly at the same time. However, at least one

Levels and *Examples*	Noticeable Delay	Tolerable Delay	3.5 MOS
Synchronous Interaction	**20 - 50 ms**	**50 - 100 ms**	?? ms
Rock-Paper-Scissors [24]	40 ms	?? ms	70 ms
Virtual car driving [55]	50 ms	200 ms	?? ms
example [?]	?? ms	?? ms	?? ms
Conversation	**100 - 150 ms**	**300 - 400 ms**	?? ms
3D Visual Communication [75]	120 ms	?? ms	?? ms
Video Group Discussion [65]	500 ms	1000 ms	500 ms
Audiovisual telecommunication [70]	?? ms	?? ms	500 ms
Take turns reading random numbers aloud as quickly as possible [37]	?? ms	?? ms	80 ms
Take turns verifying random numbers as quickly as possible [37]	?? ms	?? ms	120 ms
Take turn verifying city names as quickly as possible [37]	?? ms	?? ms	180 ms
Free conversation [37]	?? ms	?? ms	200 ms
example [?]	?? ms	?? ms	?? ms
Visual Feedback	**150 - 500 ms**	**500 - 1000 ms**	?? ms
example [?]	?? ms	?? ms	?? ms
example [?]	?? ms	?? ms	?? ms
example [?]	?? ms	?? ms	?? ms
example [?]	?? ms	?? ms	?? ms
example [?]	?? ms	?? ms	?? ms
example [?]	?? ms	?? ms	?? ms

Table 2: The three synchronization levels.

player will find that his partner gesture at least 100 ms slower, which may cause annoyance.

synchronous speaking. A pair of users would be sensitive to delay if they have to speak at the same time. For example, when counting down together in a 3DTI system with a delay of 100 ms, at least one user will hear repeated sounds more than 100 ms apart. As a reference, the human ear can distinguish an echo from the original direct sound if the delay is more than 100 ms [73]. Notice that synchronous speaking is different from a conversation (turn talking).

instrument ensemble. With the development of the high-quality network, it is clear that networked music performance has a future [8]. 3DTI makes these tasks more nature. For example, two distributed musicians can practice piano duet through 3DTI.

A realistic musical interaction assumes a one-way delay of less than 25 ms [9]. Beyond this threshold, the groove-building-process cannot be realized by musicians. [68] suggests a delay between 10 - 20 ms for providing a stabilizing effect on the tempo. For a relatively worse network, a coping strategy was discovered that allowed the performers to maintain a solid tempo up to 50 - 70 ms of delay.

2. Conversation

Conversation is an important cue for delay perception. In face-to-face situations, we have learned to unconsciously manage a conversation using the timing of the small pauses in speech [63].

[TODO] Theory: Turn Talking Model.

Chatting. [TODO]

Remote guidance. [TODO]

3. Visual Feedback

[TODO] A short introduction.

[TODO] Theory: Situation Awareness Theory.

[TODO] Theory: Grounding Theory.

Silent collaboration. [TODO] Example: Surgery Simulation.

Imitation. [TODO] Example: Building Block.

Turn-based game. [TODO] Example: Playing chess.

[NOTE] Most actual networks will not exceed such a large delay [14] [?, ?]

[NOTE] Suggestion for network design

(1) An application should reach the delay which leads to MOS of 3.5 points.
(2) If already within the noticeable delay, we can appropriately increase the delay to have more room for smoothing and recovering packet loss.
(3) Assistant synchronization can be integrated in an application. For example, we can use synchronized flickers in both side to help a Rock-Paper-Scissors game.

[NOTE] Examples of Applications

(1) *Rock-paper-scissors*:
(2) *Piano Duet*:
(3) *Chorus*:
(4) *Countdown Together*:
(5) *Chat*:
(6) *Tell-a-lie Game*:
(7) *Building Blocks*:
(8) *Interview*:
(9) *Playing Chess*:
(10) *Building Blocks without Chatting*:
(11) *Magic The Gathering*:
(12) *3D version of Hearthstone*:
(13) *Surgery Simulation*:
(14) *Playing Chess without Seeing Your Partner*:
(15) *Real-time teaching*
(16) *dancing*

We present an empirical framework of delay perception of 3D Tele Immersion. Tasks in 3DTI can be assigned to 3 different levels: tasks with enforced synchronous interaction, tasks with free conversation and tasks with visual-only communication. Tasks of different levels have different requirements of delay, as shown in figure 1(表格)

There are two indicators of delay measured by most researches: noticeability and disruptiveness of delay.

Noticeability Noticeability of delay means whether user can perceive the transmission delay in the system or not. In common case if they can communicate very fluently in the visual space as if they were communicating in the real world, then it means they can barely notice the delay.

Disruptiveness Disruptiveness means whether user can tolerate the delay in the system or not. If they believe the delay hinder them from interacting fluently or the delay disrupt the fairness of certain task or game, the disruptiveness will be very severe.

Countless work has been done on measuring these two factors in previous researches. However, all of them are conducted in audio-only experiments and 2D video experiments, as shown in figure 2(表格介绍以往工作). In audio-only telecommunication research, it is widely acknowledged that the noticeability and tolerance of delay are around 150ms and 400ms (引用). As for audiovisual experiments noticeability and tolerance of delay are around 200ms and 500ms. (need citation)

However, the framework of delay in 3DTI tasks is different from both of them. According to our experiment result, tasks can be divided into three levels. In each level, tasks have similar threshold of delay noticeability due to their common audio and visual factors, while threshold of delay tolerance usually vary in a range because of other factors.

Enforced synchronous interaction The common feature of these tasks is that they all require users to do something at the very same time. Users in this kind of tasks tend to notice delay easily through observing partner's behavior. Besides, high delay is usually intolerable because it will have obvious negative impact on QoE and fairness with game. Delay noticeability is around 50ms, which is much lower than the limit of 2D video interaction. Delay tolerance vary from 50ms to 150ms.

Some typical examples of enforced synchronous interaction are Rock-paper-scissors game, real-time musical collaboration and count-down game. Rock-paper-scissors require both sides of users to show their hand gestures simultaneously. If delay between two sides grows higher than the threshold of noticeability, user will perceive that his or her partner seems to show gestures later than expected. If it exceeds the tolerance delay, then both players will suspect that their partner shows gestures slowly intentionally and the fairness with the game is broken. More details of this game will be described in our experiments. Real-time musical collaboration requires musicians to work on the same pace. It has an even higher limitation on delay tolerance because if one of the musician lags behind or go ahead just a little fraction of a musical note (say a quaver), the general harmony of music will be destroyed completely. Thus, noticeability and tolerance are almost the same in this task. Count-down game demands that both players to count from 10 to 1 simultaneously. This is a rather low-requirement task of level 1 because players can tolerate delay up to 150ms. (evidence?)

Visual-only Interaction the common feature of tasks in this level is that audio communication is not available through the whole process. Users can only use body gestures to interact with each other. These tasks are not possible in former 2DTI systems because users cannot express their meaning fully and fluently through a 2D video screen. However, in 3D visual space, since users can see each other in a 3D shape, observe body gestures more easily and understand each other more quickly. Noticeability and tolerance of tasks in this level is much higher than the other 2(3?) levels because users usually work in a slow pace in these tasks and without audio signals they hardly perceive the system delay. Delay noticeability is around 1000ms and tolerance is 2000ms.

Some typical examples are chess game, Reversi game and ? Chess game involves two players who can play the game without talking to each other. At the beginning of chess games, players can perceive delay more easily because they move and change turns frequently. But as games go on, it takes more and more time for them to think. The slower the game pace is, the less indicators of delay they can perceive. Reversi game is similar to chess game but it requires more moves. When one side of the player moves, the other side will help remove discarded dots and put on new dots.

3 SYSTEM OVERVIEW

Our 3DTI system fuses two distributed scenes into a virtual space in full 3D. The end-to-end delay is 50 ms, i.e., the time interval between a user acts and his remote partner sees. The system consists of inexpensive commercial devices ($ 7000). Figure xxx illustrates the pipeline of our system. In particular, the system removes background and retained only individuals and similar objects in both scenes.

The project is open-source [?]. The motivation of this section is to provide necessary information for the readers to rebuild a similar system.

Hardware and Software Overview

Hardware. The system consists of two capture sites in the two distributed rooms. At each capture site, we had three depth cameras for capturing, a PC for computing and an HMD for rendering. Realsense D415 (depth cameras) were used to capture a volume of $2m \times 2m \times 2m$. The locating place of each camera and its contribution to 3D mesh are illustrated in Figure xx. Each PC had an Intel i7-7700k CPU and a GTX 1080Ti GPU. HTC Vive was used to present the fused reconstruction of both sides. Ten Gigabit network cards (Intel X520-SR2) were used to connect the two capture sites.

Software. OpenCV was used for camera calibration. CUDA was used for image processing and the kernel algorithm. Unity3D was used to implement the high-level application. It fetches live reconstruction from the kernel and renders it in HTC Vive. Python was used for audio transmission.

Calibration

Calibration between Cameras. The *camera calibration module* in OpenCV was used to calibrate the cameras. Each pair of cameras took ten snapshots (1080p color images) of a glass-made flat checkerboard. Then, OpenCV aligned their coordinates ($SD < 1pixel$).

Calibration between HMD and Cameras. The HTC Vive was calibrated by setting the original point in its software. We placed the original point of the cameras at the same position by using the checkerboard. Hence, we aligned the HTC Vive with the cameras. This calibration is not necessarily accurate because the users can hardly perceive the error [?]. This step also aligned the coordinates of the two capture sites.

Preprocessing

Depth Processing. The cameras acquired depth images of 640 × 480 pixels at 30 FPS. The Realsense D415 is based on binocular disparity. Thus, disparity values (instead of depth values) were used in the processing to improve accuracy. We applied median filtering, spatial filtering, hole filling and temporary filtering on the depth images.

Color Processing. The cameras acquired color images of 960 × 540 pixels at 30 FPS. The exposure settings were manually adjusted. We used one RGB camera as a reference and warped the other cameras to this reference by white balancing and linear mapping.

Background Removal. The system removed unnecessary background and retained only shared objects and individuals. In the calibration step, we recorded RGBD images as the background. At runtime, we removed pixels that are similar to the background based on thresholds.

3D Reconstruction

3D reconstruction is the kernel algorithm of a 3DTI system. We developed a real-time CUDA implementation of 3D reconstruction similar to KinectFusion [29]. First, the algorithm integrated depth images into a data structure named Truncated Signed Distance Function (TSDF) Volume [13]. Next, the 3D mesh was extracted from the TSDF Volume using Marching Cubes [42]. Then, the algorithm projected color images on the 3D mesh for colorization.

The resolution of TSDF volume was 256 × 256 × 256 voxels. In the TSDF processing, we used a weighted average where $W = \frac{1}{Dist}$ on different cameras to minimize the error. In the colorization, we upsampled each triangle to four quartered parts to sample more colors (Figure 1). Because the users are more sensitive to the texture but not the shape [?].

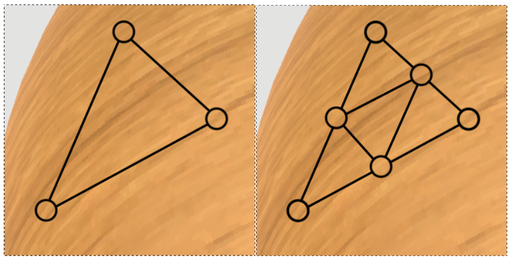


Figure 1: Left: Mesh without Supersampling; Right: Mesh witht Supersampling.

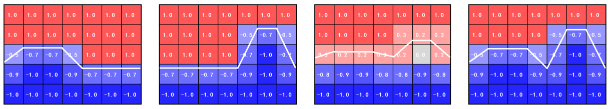
我们修改了TSDF的方法，使得它能够融合两端的场景。1、我觉得这段需要引入一些公式。2、图做半版的就行了，还没到需要用长图的程度。结合Figure 2来说。

[from lwq] We modify TSDF to adapt our demand of merging point clouds from two clients. Figure 3.4 is the weighted combination of the two profiles from different clients. The combination rules are:

$$V_z = \min\{\frac{\sum_{local} W_{i,z} S_{i,z}}{\sum_{local} W_{i,z}}, \frac{\sum_{remote} W_{j,z} S_{j,z}}{\sum_{remote} W_{j,z}}\} \tag{1}$$

$$W_{i,z} = \frac{1}{dist(i,z)} \tag{2}$$

where $S_{i,z}$ are SDF value of zth volume get by ith camera and $dist(i,z)$ is the distence from ith camera to zth volume. V_z is the final SDF value of zth volume after combination.



1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0
-0.5	-0.7	-0.7	-0.5	1.0	1.0	1.0
-0.9	-1.0	-1.0	-0.9	-0.7	-0.7	-0.7
-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0

1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	-0.5	-0.7	-0.5
1.0	1.0	1.0	1.0	-0.7	-1.0	-0.7
-0.7	-0.7	-0.7	-0.7	-0.9	-1.0	-0.9
-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0

1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	0.3	0.2	0.3
0.3	0.2	0.2	0.3	0.2	0.0	0.2
-0.8	-0.9	-0.9	-0.8	-0.8	-0.9	-0.8
-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0

1.0	1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	-0.5	-0.7	-0.5
-0.5	-0.7	-0.7	-0.5	-0.7	-1.0	-0.7
-0.9	-1.0	-1.0	-0.9	-0.9	-1.0	-0.9
-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0

Figure 2: Left to right: 1)SDF value from camera 1; 2)Merged SDF value from camera 2; 3)SDF value by simple average; 4)Merged SDF value from our project.

End-to-end Delay

The lowest end-to-end delay of our system was 52 (± xx) ms. The frame rate of 3D reconstruction was 30 FPS. It was controlled by the frequency of the synchronized depth cameras. In average, a frame (33 ms) consisted of 19 ms processing and 14 ms idling. The remote images had one frame of latency. So the end-to-end delay was about 33 + 19 = 52 ms. For artificial delay, we buffered remote data for frames. Figure 3 shows the pipeline in details. Notice that the frame rate of rendering was 90 FPS so that the users did not feel dizzy.

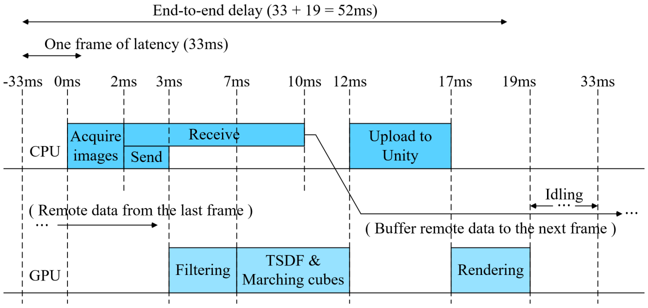


Figure 3

4 USER EXPERIMENT

[Yu] The goal of the experiment is ...

The experiment has two parts. Part A is a chess game with different conditions of cues. Results support the idea of classification in our framework. Part B is the Rock-Paper-Scissors game. It requires a low delay. We present a simple assistant design to help synchronization.

The experiment is also an example. It illustrates how to measure the noticeable delay and the acceptable delay for a specific application.

Part A: Playing Chess

Part A was to study the variety of delay perception in different synchronization level. The task was a chess game between pairs of participants in two rooms, with two conditions of cues: audiovisual mode and visual only mode.

Experimental design. We used a within-subjects experimental design. Each pair of participants played chess in two sessions of Communication Channel (CC): *audiovisual CC* and *visual only CC*, which correspond to the 2nd and 3rd synchronization level. The CC conditions were assigned to participant pairs in a Latin square design. Delay was another within-subjects factor with five conditions (50, 150, 250, 450, 750 ms in *audiovisual CC*; 150, 450, 1050, 1550, 2050 ms in *visual only CC*). In each session, we tested the five delay conditions in five trials. The delay conditions were assigned in a random order. In particular, we encouraged participants to chat in the *audiovisual CC.*

Task. In each trial, two distributed participants played chess "face-to-face" for three minutes. We adopted *Reversi* as the task. *Reversi* is simple enough that the participants can learn it in a short time. The game involves frequent interaction: when capturing, a participant should ask his partner to remove the captured chess. The participants have enough chances to perceive a noticeable delay.

In the physical world, each player interacted with a chessboard and chess pieces on his own side. The 3DTI system fused the two physical scenes into a virtual space. In the virtual space, each player could see not only his chess pieces but also his partner' s chess pieces.

Part B: Rock-Paper-Scissors

Part B was to evaluate the impact of delay in a high delay requirement situation. We used the Rock-Paper-Scissors game. We designed a synchronized audio cue to help users to synchronize with each other. The study assesses its effect on user experience.

Experimental design. This part was also a within-subjects design. Each pair played the Rock-Paper-Scissors game in two sessions: with and without the synchronized audio cue. We applied Latin square to the two sessions. In each session, we tested the delay of 50, 83, 117 and 150ms in four trials. The order of delay conditions was random. We also adopted Latin square on part A and part B.

Task. In each trial, a pair continuously drew the Rock-Paper-Scissors gestures until one of them won for ten times. There were two conditions to test: with and without the synchronized visual cue. In an actual network, it is possible to synchronize the time of two systems with almost zero milliseconds apart (the NTP protocol [?]).

Our 3DTI system provided zero-delay audio cues for both users to help them gesture exactly at the same time. The cue was an audio source of four seconds, with "tick" sounds at the 2nd, 3rd second and a "tack" sound at the 4th second. We told participants to gesture when their hear the "tack" sound.

Participants

We advertised our experiment on social media. Sixteen pairs of participants took part in our experiment (32 in total, xx females). They all came from the campus, aged from xx to xx. Participants were paid 150 yuan for the 90 minutes long study. The ten participants with the most conversation turn received extra 50 yuan.

Previous works have pointed out that the individual user differences affect study results of delay perception [?]. In our experiment, we control the source of participants carefully:

- *Relationship*: Each pair of participants are familiar with each other (friends, classmates or partners). This setting is to improve the conversation quality.
- *First language*: All the participants are native Chinese speakers. Chinese conversations are a little bit harder to predict compared to English conversations [?], which may lead to a larger noticeable delay (about xx ms).
- *Experience in DIME*: Our participants have relatively high education levels. According to the self-report questionnaire, they are quite familiar with audiovisual multimedia (xx points in average, 5 for experts) and AR/VR (xx points in average).

Thus, our study results are rigorous but relatively low in the external validity. We recommend a larger amount of participants if the readers need a more general result.

Procedure

Before the experiment, we invited the participant pair to a room and explained our study. We explained the rule of *Reversi* and *Rock-Paper-Scissors*. Next, the pair had ten minutes to experience the physical interaction of these two games. We asked the participants to remember the feeling of physical interaction and regard it as a zero-delay experience. Then, we introduced our experimental procedure to the participants.

Part A had $2sessions \times 5trials = 10trials$, which lasted for 40 minutes. Part B had $2sessions \times 4trials = 8trials$ (20 minutes). In each trial, the participants experience the remote VR game. After each trial, participants filled in a short survey and rested for one minute. The questions in the survey are shown in Table 3.

Label	Question	Scale
quality	How do you feel during the experiment?	Bad <–> Excellent
noticeability	Can you perceive the delay in the connection?	Very much <–> Not at all
tolerance	To what extent where you annoyed by the delay?	Severe annoyance <–> No annoyance

Table 3: Questions and scale.

After each session, participants had a five minutes break. We conducted brief interviews with some subjective questions as followed:

- How do you notice the delay? What are the cues?
- What makes you annoyed in the task?
- Any other comments?

In particular, we explained to participants about the concept of network delay. In our system, there was a local end-to-end delay of about 20 ms, which is slightly noticeable for the users. We told the participants that we do not consider the local delay in the questionnaire. Instead, we evaluate the network delay, which is the time interval between your partner acts and you see. A participant can perceive the network delay by judging if his partner is slow in speaking and gesturing.

Results

[NOTE from zsyzgu] I have several ideas about the result:

- In 3DTI, the users are much more tolerable than in audiovisual DIME. The immersive environment makes users focus on the interaction itself.
- A simple synchronized audio cue can help synchronization in tasks with high network requirement.
- Conversation is a stronger cue compared with visual feedback.
- Delay perception of 50 ms and 150 ms in audiovisual model have no significant difference.
- The individual user differences are larger in 3DTI.

Delay Effects. [TODO]
Attention: Analysis from only 6 groups of data.

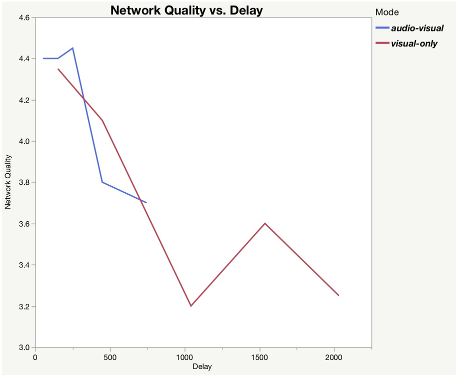


Figure 4: Effects of delay on quality score in part A(Playing Chess).

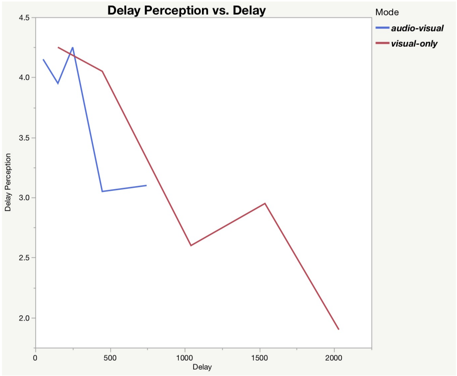


Figure 5: Effects of delay on noticeability score in part A(Playing Chess).

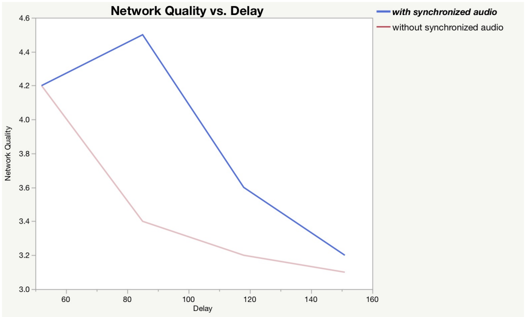


Figure 6: Effects of delay on quality score in part B(Rock-Paper-Scissors).

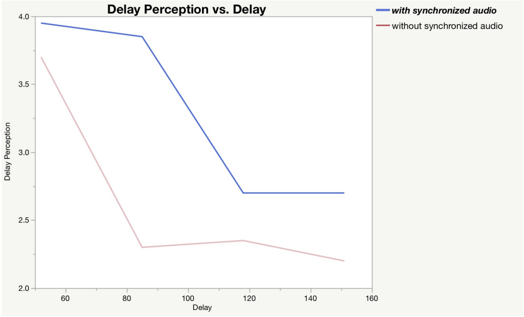


Figure 7: Effects of delay on noticeability score in part B(Rock-Paper-Scissors).

Data Analysis. [TODO]

Difference	0.30000	t Radio	0.582772
Std Err Dif	0.5148	DF	15.92888
Upper CL Dif	1.3917	Prob > |t|	0.5682
Lower CL Dif	-0.7917	Prob > t	0.2841
Confidence	0.95	Prob < t	0.7159

Table 4: Oneway analysis of noticeability score by mode(Delay = 150ms)

Difference	1.0000	t Radio	1.678363
Std Err Dif	0.5958	DF	14.9258
Upper CL Dif	2.2705	Prob > |t|	0.1141
Lower CL Dif	-0.2705	Prob > t	0.0570
Confidence	0.95	Prob < t	0.9430

Table 5: Oneway analysis of noticeability score by mode(Delay = 450ms)

5 RELATED WORK

In this section, we first review 3D tele-immersion techniques. We summarize the necessary components of current 3DTI systems to guide our implementation. Then, we have a look about existing studies on delay in 3DTI. Much more related researches were conducted in audiovisual Distributed Interactive Multimedia Environments (DIME). We discuss them later to help forming our framework.

3D Tele-immersion

Optimal techniques toward 3DTI became clear in the last decade. Basically, a 3DTI system requires three processes: reconstruction, transmission and rendering [18]. For 3D reconstruction, the volumetric algorithm has become mainstream. We applied TSDF Volume [13] and Marching Cubes [42] to fuse depth images into a polygonal mesh. We do not focus on transmission as [3, 56] did, but use 10 Gigabit optical fiber between computers instead. For rendering, we applied the head-mounted display (HTC Vive) because of its on-going growth. In this subsection, we review previous works of 3D reconstruction and rendering in details.

3D Reconstrucion. In early works, researchers used an array of cameras to capture dynamic scenes [17, 34]. For a given camera view, these systems create a polygonal model that will look correct. They do not actually construct a 3D model.

TELEPORT [20] composites video-textured surfaces within 3D geometric models. It uses only one camera. In 2002, researchers started to design virtual 3D environment with multiple cameras [22, 71]. However, their 3D reconstruction result was only point cloud. In 2008, Kurillo et al. presented a framework for remote collaboration and training of physical activities [39]. This work tried a reconstruction method with triangulation, but only reached the frame rate of about 5-7 FPS. [41] and [59] for the first time presented compelling real-time reconstruction techniques with multiple cameras. However, the lack of depth dimension indicated their modeling with only silhouette boundaries.

Researchers achieved the real-time performance of high-quality reconstruction in the last decade. In October 2011, Maimone et al. presented a 3DTI system with Kinects [43]. They developed a pixel-based mesh generation algorithm and reached a frame rate of 30 FPS. This work was followed by Beck et al.’s group-to-group telepresence system [3]. In the same month, however, Microsoft introduced KinectFusion [29] based on volumetric method. They described a novel GPU-based pipeline and achieved a better reconstruction quality. In the next year (2012), Maimone et al. also turned to volumetric methods [44] to improve the quality. A huge amount of works improved 3D reconstruction within the same framework as KinectFusion, in the region of scale [10, 53], noise reduction [35, 50, 51] and so on.

In 2016, Microsoft proposed a new pipeline named Fusion4D [15], which is highly robust to occlusions, large frame-to-frame motions, and topology changes. "The fourth dimension" is the time dimension, indicating that it leverages temporally coherence of physical scenes. In the same year, Microsoft integrated fusion4D into their 3DTI system Holoportation [54]. However, Fusion4D is extremely complex and not open-source. Even with costly devices, Holoportation has an end-to-end latency of 80ms, which can not be ignored in our study. In this paper, we apply a 3D reconstruction method similar to [44] (2012) for responsiveness.

3D Rendering. Rendering techniques in 3DTI systems can be mainly divided into three categories: light field displays, Spatially Immersive Displays (SIDs) and HMDs. The light field displays [21, 30, 33, 36] suffers from low resolution because neither computing nor rendering devices can support high-quality 4d light fields. SIDs were earlier, while HMDs are becoming popular nowadays.

Around year 2000, SIDs had become increasing significant [22]. CAVE [12] is a typical SIDs system, which consists of surround-screen projection. Users wear 3D glasses in a CAVE. Most 3DTI systems at that time applied rendering techniques similar to CAVE [4, 20, 22, 39, 71]. CAVE was design to support the one-to-many presentation. Latter researchers improved it for multi-user by using polarization or time-sharing [16, 23, 38]. Multi-user SID was used by an immersive group-to-group telepresence [3]. There is also a simplified technique called head-tracked auto-stereo display [5, 31], which allows 3D view without glasses. Some 3DTI system [43, 44, 57] used it for rendering. However, these glasses-free systems have to abandon the benefit of stereoscopy.

Recently, HMDs are becoming popular. More 3DTI systems tend to apply HMDs for 3D rendering [40, 45, 54, 69]. HMDs are basically cheaper and easier to deploy compared to SIDs. Another superiority of HMDs is their ability to support co-located collaboration [45, 54], i.e., users feel like exactly in the same place. For comparison, SIDs do not support rendering in full 3D, with which a 'window' separate users into two virtual spaces. In 2018, Microsoft proposed Remixed Reality [40]. This approach combines the benefits of augmented reality and virtual reality using 3D reconstruction and VR HMD. Users can not only see their environment but can also apply changes to it. Finally, we applied the head-mounted VR (HTC Vive) for rendering.

Delay Perception in 3DTI

User experience often relates to Quality of Service (QoS) including delay, bandwidth, jitter and packet loss [14]. Previous works have found that delay is one of the most crucial factors determining user experience in telepresence [7, 65, 67, 72]. For telephone, 150ms has been established as an industry standard for an acceptable delay [58, 61]. Also, a huge amount of related works have been conducted in audiovisual DIME.

3DTI is quite different from audiovisual DIME: first, high level of immersion offers more cues, i.e., users may be more sensitive to delay in 3DTI; second, with abundant sensory stimuli, users are more tolerant to delay [70]; third, 3DTI can support much more possible applications that we have to discuss them case by case. Thus, we have to rebuild the theoretical framework of delay perception in 3DTI.

In the area of 3DTI, most works focus on algorithm and pipeline. Negative impacts of large delay are widely reported [3, 20, 39, 43, 60]. However, only a few works were conducted to study delay perception in 3DTI [26, 74, 75]. These works do not exactly focus on delay perception. Moreover, they are limited by single scenario and immature techniques, e.g., the 2D screen was used to display 3D scenes. In this paper, we explore delay perception in a full 3D tele-immersion. We consider various scenarios and finally form a framework to understand this problem.

6 LIMITATION

1. The rendering quality of the system is not state-of-the-art.
2. The lack of eye contact.
3. The low external validity of the experiment.

7 CONCLUSION

This paragraph is for the conclusion.

ACKNOWLEDGMENTS

We thank all the volunteers, and all publications support and staff, who wrote and provided helpful comments on previous versions of this document. Authors 1, 2, and 3 gratefully acknowledge the grant from NSF (#1234–2012–ABC). *This whole paragraph is just an example.*

REFERENCES

[1] Ivelina V Alexandrova, Paolina T Teneva, Stephan De La Rosa, Uwe Kloos, Heinrich H Bülthoff, and Betty J Mohler. 2010. Egocentric distance judgments in a large screen display immersive virtual environment. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*. ACM, 57–60.

[2] Ignacio Avellino, Cédric Fleury, and Michel Beaudouin-Lafon. 2015. Accuracy of deictic gestures to support telepresence on wall-sized displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2393–2396.

[3] Stephan Beck, Andre Kunert, Alexander Kulik, and Bernd Froehlich. 2013. Immersive group-to-group telepresence. *IEEE Transactions on Visualization and Computer Graphics* 19, 4 (2013), 616–625.

[4] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. 2012. MirageTable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 199–208.

[5] Hrvoje Benko, Andrew D Wilson, and Federico Zannier. 2014. Dyadic projected spatial augmented reality. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 645–655.

[6] Sabah Boustila, Antonio Capobianco, and Dominique Bechmann. 2015. Evaluation of factors affecting distance perception in architectural project review in immersive virtual environments. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*. ACM, 207–216.

[7] Kjell Brunnström, Sergio Ariel Beker, Katrien De Moor, Ann Dooms, Sebastian Egger, Marie-Neige Garcia, Tobias Hossfeld, Satu Jumisko-Pyykkö, Christian Keimel, Mohamed-Chaker Larabi, et al. 2013. Qualinet white paper on definitions of quality of experience. (2013).

[8] Alexander Carôt, Pedro Rebelo, and Alain Renaud. 2007. Networked music performance: State of the art. In *Audio engineering society conference: 30th international conference: intelligent audio environments*. Audio Engineering Society.

[9] Alexander Carôt and Christian Werner. 2007. Network music performance-problems, approaches and perspectives. In *Proceedings*

of the "Music in the Global Village"-Conference, Budapest, Hungary, Vol. 162. 23–10.

[10] Jiawen Chen, Dennis Bautembach, and Shahram Izadi. 2013. Scalable real-time volumetric surface reconstruction. *ACM Transactions on Graphics (ToG)* 32, 4 (2013), 113.

[11] Jessie YC Chen and Jennifer E Thropp. 2007. Review of low frame rate effects on human performance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 37, 6 (2007), 1063–1076.

[12] Carolina Cruz-Neira, Daniel J Sandin, and Thomas A DeFanti. 1993. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, 135–142.

[13] Brian Curless and Marc Levoy. 1996. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 303–312.

[14] Angus Donovan, Leila Alem, Weidong Huang, Ren Liu, and Mark Hedley. 2014. Understanding How Network Performance Affects User Experience of Remote Guidance. In *CYTED-RITOS International Workshop on Groupware*. Springer, 1–12.

[15] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. 2016. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 114.

[16] Bernd Fröhlich, Jan Hochstrate, Jörg Hoffmann, Karsten Klüger, Roland Blach, Matthias Bues, and Oliver Stefani. 2005. Implementing multi-viewer stereo displays. (2005).

[17] Henry Fuchs, Gary Bishop, Kevin Arthur, Leonard McMillan, Ruzena Bajcsy, Sang Lee, Hany Farid, and Takeo Kanade. 1994. Virtual space teleconferencing using a sea of cameras. In *Proc. First International Conference on Medical Robotics and Computer Assisted Surgery*, Vol. 26.

[18] Henry Fuchs, Andrei State, and Jean-Charles Bazin. 2014. Immersive 3d telepresence. *Computer* 47, 7 (2014), 46–52.

[19] David Geerts, Ishan Vaishnavi, Rufael Mekuria, Oskar Van Deventer, and Pablo Cesar. 2011. Are we in sync?: synchronization requirements for watching online video together.. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 311–314.

[20] Simon J Gibbs, Constantin Arapis, and Christian J Breiteneder. 1999. TELEPORT–Towards immersive copresence. *Multimedia Systems* 7, 3 (1999), 214–221.

[21] Daniel Gotsch, Xujing Zhang, Timothy Merritt, and Roel Vertegaal. 2018. TeleHuman2: A Cylindrical Light Field Teleconferencing System for Life-size 3D Human Telepresence. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 522.

[22] Markus Gross, Stephan Würmlin, Martin Naef, Edouard Lamboray, Christian Spagno, Andreas Kunz, Esther Koller-Meier, Tomas Svoboda, Luc Van Gool, Silke Lang, et al. 2003. blue-c: a spatially immersive display and 3D video portal for telepresence. In *ACM Transactions on Graphics (TOG)*, Vol. 22. ACM, 819–827.

[23] Dongdong Guan, Chenglei Yang, Weisi Sun, Yuan Wei, Wei Gai, Yulong Bian, Juan Liu, Qianhui Sun, Siwei Zhao, and Xiangxu Meng. 2018. Two Kinds of Novel Multi-user Immersive Display Systems. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 599.

[24] Yousuke Hashimoto and Yutaka Ishibashi. 2006. Influences of network latency on interactivity in networked rock-paper-scissors. In *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*. ACM, 23.

[25] Keita Higuchi, Yinpeng Chen, Philip A Chou, Zhengyou Zhang, and Zicheng Liu. 2015. ImmerseBoard: Immersive telepresence experience using a digital whiteboard. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2383–2392.

[26] Zixia Huang, Ahsan Arefin, Pooja Agarwal, Klara Nahrstedt, and Wanmin Wu. 2012. Towards the understanding of human perceptual quality in tele-immersive shared activity. In *Proceedings of the 3rd Multimedia Systems Conference*. ACM, 29–34.

[27] Yutaka Ishibashi, Manabu Nagasaka, and Noriyuki Fujiyoshi. 2006. Subjective assessment of fairness among users in multipoint communications. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*. ACM, 69.

[28] T ITU. 2003. Recommendation G. 107 The E-model, a computational model for use in transmission planning. (2003).

[29] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 559–568.

[30] Andrew Jones, Ian McDowall, Hideshi Yamada, Mark Bolas, and Paul Debevec. 2007. Rendering for an interactive 360 light field display. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 40.

[31] Brett Jones, Rajinder Sodhi, Michael Murdock, Ravish Mehra, Hrvoje Benko, Andrew Wilson, Eyal Ofek, Blair MacIntyre, Nikunj Raghuvanshi, and Lior Shapira. 2014. RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 637–644.

[32] Norman P Jouppi, Daniel J Scales, and Wayne Roy Mack. 2001. Robotic telepresence system. US Patent 6,292,713.

[33] Joel Jurik, Andrew Jones, Mark Bolas, and Paul Debevec. 2011. Prototyping a light field display involving direct observation of a video projector array. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*. IEEE, 15–20.

[34] Takeo Kanade, Peter Rander, and PJ Narayanan. 1997. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia* 4, 1 (1997), 34–47.

[35] Kourosh Khoshelham and Sander Oude Elberink. 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors* 12, 2 (2012), 1437–1454.

[36] Kibum Kim, John Bolton, Audrey Girouard, Jeremy Cooperstock, and Roel Vertegaal. 2012. TeleHuman: effects of 3d perspective on gaze and pose estimation with a life-size cylindrical telepresence pod. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2531–2540.

[37] Itoh K Kitawaki N. 1991. Pure Delay Effect on Speech Quality in Telecommunications. *IEEE J. Sel. Areas Comm*, 586–593.

[38] Alexander Kulik, André Kunert, Stephan Beck, Roman Reichel, Roland Blach, Armin Zink, and Bernd Froehlich. 2011. C1x6: a stereoscopic six-user display for co-located collaboration in shared virtual environments. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 188.

[39] Gregorij Kurillo, Ruzena Bajcsy, Klara Nahrsted, and Oliver Kreylos. 2008. Immersive 3d environment for remote collaboration and training of physical activities. In *Virtual Reality Conference, 2008. VR'08. IEEE*. IEEE, 269–270.

[40] David Lindlbauer and Andy D Wilson. 2018. Remixed Reality: Manipulating Space and Time in Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 129.

[41] Charles Loop, Cha Zhang, and Zhengyou Zhang. 2013. Real-time high-resolution sparse voxelization with application to image-based modeling. In *Proceedings of the 5th High-Performance Graphics Conference*. ACM, 73–79.

[42] William E Lorensen and Harvey E Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM siggraph computer graphics*, Vol. 21. ACM, 163–169.

[43] Andrew Maimone and Henry Fuchs. 2011. Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 137–146.

[44] Andrew Maimone and Henry Fuchs. 2012. Real-time volumetric 3D capture of room-sized scenes for telepresence. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*. IEEE, 1–4.

[45] Andrew Maimone, Xubo Yang, Nate Dierk, Andrei State, Mingsong Dou, and Henry Fuchs. 2013. General-purpose telepresence with head-worn optical see-through displays and projector-based lighting. In *Virtual Reality (VR), 2013 IEEE*. IEEE, 23–26.

[46] Jennifer Marlow, Scott Carter, Nathaniel Good, and Jung-Wei Chen. 2016. Beyond talking heads: multimedia artifact creation, use, and sharing in distributed meetings. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 1703–1715.

[47] Kana Misawa and Jun Rekimoto. 2015. ChameleonMask: Embodied physical and social telepresence using human surrogates. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 401–411.

[48] Mario Montagud, Fernando Boronat, Hans Stokking, and Ray van Brandenburg. 2012. Inter-destination multimedia synchronization: schemes, use cases and standardization. *Multimedia systems* 18, 6 (2012), 459–482.

[49] Carman Neustaedter, Gina Venolia, Jason Procyk, and Daniel Hawkins. 2016. To Beam or not to Beam: A study of remote telepresence attendance at an academic conference. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 418–431.

[50] Richard A Newcombe, Dieter Fox, and Steven M Seitz. 2015. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 343–352.

[51] Chuong V Nguyen, Shahram Izadi, and David Lovell. 2012. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*. IEEE, 524–530.

[52] Jakob Nielsen. 1993. Response times: the three important limits. *Usability Engineering* (1993).

[53] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)* 32, 6 (2013), 169.

[54] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. 2016. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 741–754.

[55] Lothar Pantel and Lars C Wolf. 2002. On the impact of delay on real-time multiplayer games. In *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*. ACM, 23–29.

[56] Fabrizio Pece, Jan Kautz, and Tim Weyrich. 2011. Adapting standard video codecs for depth streaming. In *Proceedings of the 17th Eurographics conference on Virtual Environments & Third Joint Virtual Reality*. Eurographics Association, 59–66.

[57] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. 2016. Room2room: Enabling life-size telepresence in a projected augmented reality environment. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*. ACM, 1716–1725.

[58] Alan Percy. 1999. Understanding latency in IP telephony. *Brooktrout Technology, Needham, MA* (1999).

[59] Benjamin Petit, Jean-Denis Lesage, Clément Menier, Jérémie Allard, Jean-Sébastien Franco, Bruno Raffin, Edmond Boyer, and François Faure. 2010. Multicamera real-time 3d modeling for telepresence and remote collaboration. *International journal of digital multimedia broadcasting* 2010 (2010).

[60] Suraj Raghuraman and Balakrishnan Prabhakaran. 2015. Distortion score based pose selection for 3D tele-immersion. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology*. ACM, 227–236.

[61] ITUT Rec. 2003. G. 114,". *One-way transmission time* (2003).

[62] G Recommendation. 2003. 114-One-way transmission time ITU.

[63] Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1978. A simplest systematics for the organization of turn taking for conversation. In *Studies in the organization of conversational interaction*. Elsevier, 7–55.

[64] Batu Sat and Benjamin W Wah. 2009. Statistical scheduling of offline comparative subjective evaluations for real-time multimedia. *IEEE Transactions on Multimedia* 11, 6 (2009), 1114–1130.

[65] Marwin Schmitt, Simon Gunkel, Pablo Cesar, and Dick Bulterman. 2014. Asymmetric delay in video-mediated group discussions. In *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*. IEEE, 19–24.

[66] Marwin Schmitt, Simon Gunkel, Pablo Cesar, and Dick Bulterman. 2014. The influence of interactivity patterns on the Quality of Experience in multi-party video-mediated conversations under symmetric delay conditions. In *Proceedings of the 3rd International Workshop on Socially-aware Multimedia*. ACM, 13–16.

[67] Marwin Schmitt, Simon Gunkel, Pablo Cesar, and Peter Hughes. 2013. A QoE testbed for socially-aware video-mediated group communication. In *Proceedings of the 2nd international workshop on Socially-aware multimedia*. ACM, 37–42.

[68] Nathan Schuett. 2002. The effects of latency on ensemble performance. *Bachelor Thesis, CCRMA Department of Music, Stanford University* (2002).

[69] Harrison Jesse Smith and Michael Neff. 2018. Communication Behavior in Embodied Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 289.

[70] Jennifer Tam, Elizbeth Carter, Sara Kiesler, and Jessica Hodgins. 2012. Video increases the perception of naturalness during remote interactions with latency. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2045–2050.

[71] Herman Towles, Wei-Chao Chen, Ruigang Yang, Sang-Uok Kum, Henry Fuchs Nikhil Kelshikar, Jane Mulligan, Kostas Daniilidis, Henry Fuchs, Carolina Chapel Hill, Nikhil Kelshikar Jane Mulligan, et al. 2002. 3d tele-collaboration over internet2. In *In: International Workshop on Immersive Telepresence, Juan Les Pins*. Citeseer.

[72] Andreas Vogel, Brigitte Kerherve, Gregor von Bochmann, and Jan Gecsei. 1995. Distributed multimedia and QoS: A survey. *IEEE multimedia* 2, 2 (1995), 10–19.

[73] Matthias Wölfel and John McDonough. 2009. *Distant speech recognition*. John Wiley & Sons.

[74] Wanmin Wu, Ahsan Arefin, Zixia Huang, Pooja Agarwal, Shu Shi, Raoul Rivas, and Klara Nahrstedt. 2010. " I'm the Jedi!"-A Case Study of User Experience in 3D Tele-immersive Gaming. In *Multimedia (ISM), 2010 IEEE International Symposium on*. IEEE, 220–227.

[75] Wanmin Wu, Ahsan Arefin, Raoul Rivas, Klara Nahrstedt, Renata Sheppard, and Zhenyu Yang. 2009. Quality of experience in distributed interactive multimedia environments: toward a theoretical framework.

In *Proceedings of the 17th ACM international conference on Multimedia.* ACM, 481–490.

[76] Jingxi Xu and Benjamin W Wah. 2013. Exploiting just-noticeable difference of delays for improving quality of experience in video conferencing. In *Proceedings of the 4th ACM Multimedia Systems Conference.* ACM, 238–248.