HUMAN-CENTRIC CONTROL OF VIDEO FUNCTIONS AND UNDERLYING RESOURCES IN 3D TELE-IMMERSIVE SYSTEMS

BY

WANMIN WU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Professor Klara Nahrstedt, Chair and Director of Research
Associate Professor Indranil Gupta
Professor Thomas Huang
Professor Ramesh Jain, University of California, Irvine

Abstract

3D tele-immersion (3DTI) has the potential of enabling virtual-reality-like interaction among remote people with real-time 3D video. However, today's 3DTI systems still suffer from various performance issues, limiting their broader deployment, due to the enormous demand on temporal (computing) and spatial (networking) resources. Past research focused on system-centric approaches for technical optimization, without taking human users into the loop. We argue that human factors (including user preferences, semantics, limitations, etc.) are an important and integral part of the cyber-physical 3DTI systems, and should not be neglected.

This thesis proposes a novel, comprehensive, human-centric framework for improving the qualities of 3DTI throughout its video function pipeline. We make three major contributions at different phases of the pipeline. At the sending side, we develop an intra-stream data adaptation scheme that reduces level-of-details within each stream without users being aware of it. This human-centric approach exploits limitations of human vision, and excludes details that are imperceptible. It effectively alleviates the data load for computation-intensive operations, thus improves the temporal efficiency of the systems. Yet even with intra-stream data reduced, spatial efficiency is still a problem due to the multi-stream/multi-site nature of 3DTI collaboration. We thus develop an inter-stream data adaptation scheme at the networking phase to reduce the number of streams with minimal disruption to the visual quality. This human-centric approach prioritizes streams based on user views and excludes less important streams from transmission. It considerably reduces the data load for networking, and thus enhances the spatial resource efficiency. The above two approaches (level-of-details reduction within a video stream and view-based differentiation among streams) work seamlessly together to bring both temporal and spatial resource demands under control, and prove to improve various qualities of the systems. Finally, at the receiving side, we take a holistic approach to study the "quality" concept in 3DTI environments. Our human-centric quality framework focuses on the Quality-of-Experience (QoE) concept that models user's perceptions, emotions, performances, etc. It investigates how the traditional Quality-of-Service (QoS) impacts QoE, and reveals how QoS should be improved for the best user experience. This thesis essentially demonstrates the importance of bringing human-awareness into the design, execution, and evaluation of the complex resource-constrained 3DTI environments.

To Xiao & Sicheng

Acknowledgments

First and foremost, I would like to express my deep gratitude to my advisor, Professor Klara Nahrstedt, for initially giving me the opportunity to join this fascinating tele-immersion project, for pushing me forward while giving me the freedom to go for things that excited me, for all the valuable discussions and countless comments on my drafts, for the willing advice whenever needed, and for the continuing support, belief, and encouragement. I will be forever grateful for having Klara as my advisor.

I also sincerely thank my other committee members, Professor Thomas Huang, Professor Ramesh Jain, and Professor Indranil Gupta, for offering insightful feedback and constructive suggestions on my thesis. My special thanks go to Professor Gupta who has taught me a great deal during our collaboration on the inter-stream adaptation work.

Many thanks to my colleagues in the tele-immersion project, particularly Zhenyu Yang, Ahsan Arefin, Renata Sheppard, Zixia Huang, Pooja Agarwal, Raoul Rivas, and Shu Shi, for offering generous assistance in my experiments, and for proof-reading my papers. I would also like to specially thank our collaborators at University of California at Berkeley, Gregorij Kurillo, Ramanarayan Vasudevan, and Professor Ruzena Bajcsy. Thanks to their support and help, I was able to collect data-sets for the intra-stream adaptation work at their lab.

The completion of this thesis marks the end of my many years as a student. Among many outstanding teachers I would like to especially thank Ms. Zuohua Wang, Ms. Yaying Jiang, Ms. Yuan Liu, Mr. Jin Huang, Ms. Xi Hua, and Dr. Fei Wu. I am also deeply grateful to Professor Kien A. Hua at University of Central Florida for mentoring me, believing in me, and encouraging me to always aim high.

I am grateful for all the friends with whom I spent my time at UIUC, who made the life in the corn field fun and memorable. In particular, I would like to thank the present and past members of Multimedia Operating Systems and Networking (MONET) research group, including Bin Yu, Jin Liang, Wenbo He, Hoang Nguyen, Ying Huang, Long Vu, Muyuan Wang, Qiyan Wang, Thadpong Pongthawornkamol, Rahul Malik, Shameem Ahmed, Debessay Fesehaye, Ravishankar Sathyam, Roger Cheng, Jigar Doshi, Naveen Cherukuri, and Kurchi Subhra Hazra. Sincere appreciation is extended to Anda Ohlsson, Lynette Lubben, and Mary Beth Kelley who helped me in administrative matters.

I would like to express my earnest gratitude to my parents for their support, without which none of my achievements would have been possible. Being the only child of them, and the only child in our extended family who came abroad to study, I feel deeply indebted for their understanding, unconditional support, and countless sacrifices to give me the best possible education. I also thank my parents and parents-in-law for devotedly taking care of my son while I complete my thesis.

Last, but certainly not least, I dedicate this thesis to my dear husband Xiao and my beloved son Sicheng (Matthew). They have enlightened my life in so many ways. Words can hardly express how grateful I am for having them in my life.

This material is based in part upon work supported by the National Science Foundation under Grant Numbers SGER-0840323, CSR-0834480, CSR-0720702, SGER-0724464, SCI SGER-0549242, and NeTS-0520182.

Table of Contents

List of Tables

List of Figures

List of Symbols

$CZLoD$	Color-plus-depth level-of-details in a 3D video stream, quantified by the number of vertices after meshing
$JNDG$	Just Noticeable Degradation Ratio in CZLoD
$JUADG$	Just Unacceptable Degradation Ratio in CZLoD
p	Probability for determining psychophysical thresholds
f_i	2D reference frame with frame number i
F_i	Color-plus-depth frame reconstructed from f_i
$CZLoD(F_i)$	Number of foreground vertices in F_i
FR	Frame rate
W	Size of running window for computing frame rate
TH_{var}	Variance threshold (detailing parameter)
TH_{fr}^h	Upper limit of normal frame rate range
TH_{fr}^l	Lower limit of normal frame rate range
$DR^a(\mathsf{F}_i)$	Actual degradation ratio of CZLoD in F_i
$DR^t(\mathsf{F}_i)$	Target degradation ratio of CZLoD in F_i
$N_0(\mathsf{F}_i)$	CZLoD of F_i when $TH_{var} = 0$
$N_v(\mathsf{F}_i)$	CZLoD of F_i when $TH_{var} = v$
$\mathcal{F}$	Mapping function from DR^t to TH_{var}
$err(\mathsf{F}_i)$	$= \lvert DR^a(\mathsf{F}_i) - DR^t(\mathsf{F}_i)\rvert$, error of degradation ratio
TH_{err}	Threshold of err to trigger variance learning
Δ_u, Δ_d	Decrease and increase sizes for DR^t adjustment
G_i	Gateway node at site i, $1 \leq i \leq N$
I_i	Inbound bandwidth limit at G_i in terms of the number of streams it can receive
O_i	Outbound bandwidth limit at G_i in terms of the number of streams it can send
$d_{in}(\mathsf{G_i})$	Actual in-degree at G_i in terms of the number of streams it receives
$d_{out}(\mathsf{G_i})$	Actual out-degree at G_i in terms of the number of streams it sends

$cost(\mathsf{G}_i \Rightarrow \mathsf{G}_j)_{\mathbb{T}(\mathsf{s})}$	The overlay path (multi-hop) latency from G_i to G_j on $\mathbb{T}(\mathsf{s})$
$cost(\mathbb{P})$	The end-to-end latency of the overlay path $\mathbb{P}$
$\mathbb{P}(\mathsf{G}_i \rightarrow \mathsf{G}_j)_{\mathbb{T}(\mathsf{s})}$	The overlay path from node G_i to G_j in $\mathbb{T}(\mathsf{s})$
X	Rejection ratio of subscription requests in overlay construction
$u_{i \rightarrow j}$	Number of subscription requests made by G_i to G_j
$\hat{u}_{i \rightarrow j}$	Number of subscription requests rejected in $u_{i \rightarrow j}$
$\mathbb{L}$	Set of all overlay links between every pair of gateway nodes
$\mathbb{G}$	Total set of gateway nodes $\mathbb{G} = \{\mathsf{G}_i \| 1 \leq i \leq N\}$
R_i^j	jth renderer node at site i, $1 \leq i \leq N$
N	Number of participating sites
M	Number of active tele-immersive sites
$\vec{V}_i^j$	User view selected on R_i^j
$\mathbb{S}_i^{local}$	Local streams produced at site i
$\mathbb{S}_{subscribed}$	Set of all streams subscribed by at least one renderer
$\mathbb{S}$	Set of all streams in the system
s_i^p or s	Video stream produced at site i with index p, $\mathsf{s}_i^p \in \mathbb{S}_i^{local}, 1 \leq p \leq \|\mathbb{S}_i^{local}\|$
$\mathbb{G}(\mathsf{s})$	Multicast group of gateways subscribing to stream s
$\mathbb{G}_{multicast}$	Total set of multicast groups, $\mathbb{G}_{multicast} = \{\mathbb{G}(s) \mid s \in \mathbb{S}_{subscribed}\}$
G^{s}	Source gateway of s (where it is produced)
$\mathbb{T}(\mathsf{s})$	Overlay tree (of gateways) constructed to deliver stream s
rfc_i	Remaining forwarding capacity of G_i, computed as $O_i - d_{out}(\mathsf{G}_i) - \hat{m}_i$.
$\hat{m}_i$	Number of streams that (1) originate from G_i, (2) are subscribed by at least one other gateway, and (3) have not yet been disseminated to any other node in the existing forest
$\mathsf{req}(\mathsf{s}_j^q)$	Subscription request specifying that G_i requests to receive stream s_j^q
$\mathbb{G}'(\mathsf{s})$	Set of gateways that subscribes to s and is able to receive it (or in other words, contained in $\mathbb{T}(\mathsf{s})$, $\mathbb{G}'(\mathsf{s}) \subset \mathbb{G}(s)$
$\mathbb{S}_{\vec{V}_i^j}^{requested}$	Request stream set: the subset of streams G_i determines as the important streams to serve $\vec{V}_i^j$. $\|\mathbb{S}_{\vec{V}_i^j}^{requested}\| = N \times \kappa, \mathbb{S}_{\vec{V}_i^j}^{requested} \subset \mathbb{S}$
$Q_{i \rightarrow j}$	Criticality for a node G_i to lose any stream originating from G_j
g	Granularity in tree construction, essentially the number of trees the algorithm attempts to construct at once
$\vec{C}_{\mathsf{s}}$	Camera view of stream s

$CF_{\mathsf{s}}^{\vec{V}_i^j}$	$=\vec{V}_i^j \cdot \vec{C}_{\mathsf{s}}$: contributing factor of stream s with respect to view $\vec{V}_i^j$
$TH_{contribution}$	Threshold used for contribution factor during stream selection
K	Number of streams selected from each site to serve a view
p_k	Priority degree, $1 \leq k \leq K$
$\mathbf{P}$	$=< p_1, p_2, ..., p_K >$: priority scale, the ordered set of priority degrees
$\mathcal{P}(\vec{V}_i^j, \mathsf{s})$	: $f(\vec{V}_i, \mathsf{s}) \mapsto p_k$ the mapping function from contributing factor to priority degree
$\mathbb{C}(\vec{V}_i^j, \mathsf{s})$	The set of candidate nodes that has s to serve G_i for $\vec{V}_i^j$, and also satisfies the latency constraint
$\mathbb{C}_1(\vec{V}_i^j, \mathsf{s})$	The set of nodes that has available bandwidth to serve s to G_i, $\mathbb{C}_1(\vec{V}_i^j, \mathsf{s}) \subseteq \mathsf{C}(\vec{V}_i^j, \mathsf{s})$
$\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$	$= \mathbb{C}(\vec{V}_i^j, \mathsf{s}) - \mathbb{C}_1(\vec{V}_i^j, \mathsf{s})$
$I_j^{\mathsf{s}'}$	Preemption impact at a candidate node for stream s'

1 Introduction

1.1 3D Tele-Immersion

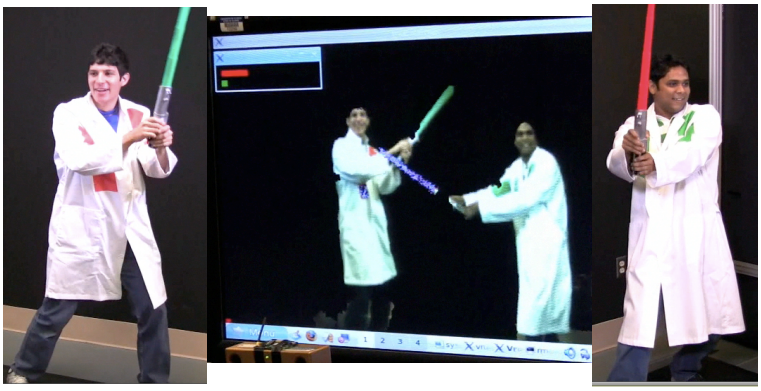
1.1.1 Background

When Princess Leia's three-dimensional (3D) hologram appeared before Obi-Wan in the 1977 Star Wars movie, it vividly depicted one of our wildest dreams - making remote communication just-like-being-there. Since the advent of Internet and particularly the Internet Protocol (IP) in the 1990s, there has been a surge of interest in video-conferencing technologies over the networks. Those technologies, although quite primitive with low-resolution display and low interaction speed, represent some of the earliest attempts to eliminate the geographical boundaries among people and enable remote interaction through video.

In recent years, conventional video-conferencing systems gradually evolve into the more advanced *telepresence systems*, making one step closer to our end goal of completely immersive video communication. Telepresence systems, featuring life-sized high-definition displays, have gained tremendous momentum in the industry, manifest in the success of Cisco TelePresence [1], Polycom Telepresence [82], Teliris Telepresence [4], and HP Halo [46], etc. However, the existing telepresence systems are still very limited in several ways: (1) their application is constrained to solely desktop video-conferencing scenarios, (2) they are often cost-prohibitive, especially with the need of high-capacity networks, limiting the broader deployment beyond enterprises, and (3) they only provide two-dimensional (2D) visual information, whereas in the physical world communication is designed to be naturally 3D.

As a significant step towards making video communication more immersive, 3D tele-immersion (3DTI) emerges as the next-generation telepresence technology. Unlike the existing video-conferencing or tele-presence systems, 3DTI uses arrays of stereo cameras to capture users from different perspectives, and visualizes all the 3D representations of remote users in a joint virtual-reality environment. With the cyber-space directly mapping a collaborative physical-space, the technology has the potential to considerably enhance the sense of immersion and telepresence of users. Impressive progress has been made in the past few years, and 3DTI prototypes have been demonstrated in a number of applications such as cyber-archeology, rehabilitation, collaborative dancing, and gaming [11][32][39][99][106] (Figure 1.1).

(a)



(b)

Figure 1.1: Tele-immersive applications: (a) collaborative dancing with two remote users, and (b) virtual lightsaber duet.

1.1.2 Software and Hardware Components

Behind the scene, 3DTI is a complicated, multi-phase, distributed pipeline of functions operating on video data. Figure 1.2 abstracts the 3DTI run-time software pipeline (from a sending site to a receiving site), and Figure 1.3 shows the hardware setup in an example tele-immersive site. As Figure 1.2 shows, the pipeline has three main phases: *capture/3D reconstruction*, *data dissemination/transmission*, and *3D rendering/visualization*. Distributed tele-immersive environments are connected via the data dissemination components, such that the video streams from all sites are exchanged, aggregated and presented to the users in a collaborative 3D world (refer to Figure 1.1).

- **Capture/3D Reconstruction Phase.** Tele-immersive video is often captured by an array of synchronized cameras surrounding the physical

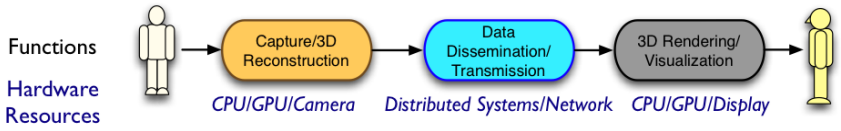


Figure 1.2: Main video functions and the underlying hardware resources in 3DTI pipeline.

environment, with each camera contributing a unique view of the scene (refer to Figure 1.3). Unlike conventional multi-view video conferencing/lecturing systems, each camera here is a stereo unit, typically equipped with binocular or trinocular lenses, and connected to a host computer via IEEE 1394 (FireWire) interface. Hardware trigger is used to synchronize all cameras to grab images at the same instants of time. This is done by periodically sending a hardware trigger signal from the parallel port on the server to a pair of general purpose I/O pins on each camera. At interactive rates, the host computer grabs image frames synchronously from all lenses and produces a *stream* of color-plus-depth frames.

- **Data Dissemination/Transmission Phase.** Figure 1.4 shows the data dissemination topology in multi-site tele-immersive system. We use a hierarchical topology for scalability. At the *local* level (i.e., within each site), the end hosts (i.e., camera host computers, display host computers) are managed by a *service gateway*, which is responsible for both outbound and inbound traffic. For outbound traffic, the gateway collects all local streams from the camera host machines, and disseminates them out to the other remote gateways. For inbound traffic, it receives the video streams from remote sites, and forwards both local and remote streams to the local displays. At the *global* level, all gateway nodes are managed by a *session controller* which handles the membership and overlay topology construction for all gateway nodes.

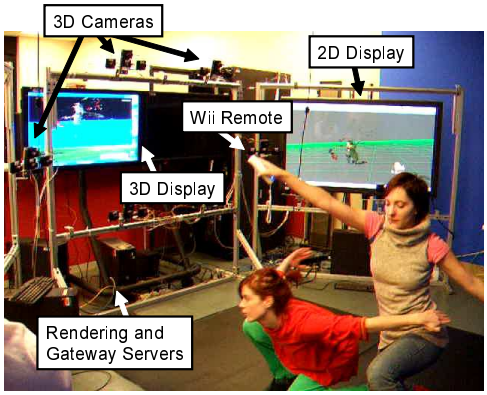


Figure 1.3: A sample tele-immersive site

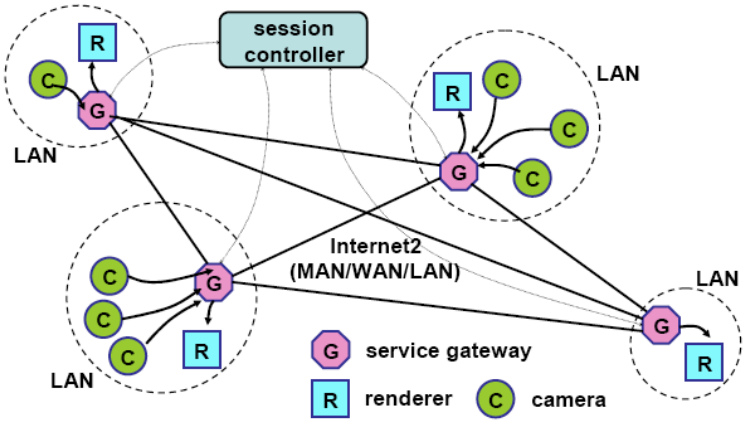


Figure 1.4: Data dissemination setup in multi-site tele-immersive systems (figure adapted from [113]).

- **3D Rendering/Visualization Phase.** At the receiving sites, all local and remote streams are rendered into a joint virtual world as depicted in Figure 1.1. It is worth noting that the video streams from different cameras are simply rendered together in an aggregated way. This is because creating a redundancy-free 3D model by cross-correlating the multi-view streams in real time is an unsolved challenge, particularly given the stringent timing requirement of interactive applications. Tele-immersive systems therefore resort to offline camera calibration in order to fuse multi-camera stream data online. With calibration parameters, multiple depth-mapped video streams can then be rendered directly into a shared world coordinate system. The complete depth information allows for the selection of arbitrary vantage points as in other free-viewpoint video [93]. Multiple displays are placed in different positions and angles to present the scene with different viewpoints, such that the participant can observe it even when she moves or turns (Figure 1.3). Each display is connected to a computer (i.e., *renderer*) which mainly handles real-time rendering of the 3D video streams. It also provides users an interface to change the rendering viewpoint in the 3D cyber-space (via mouse or wireless controllers like the Wii remote shown in Figure 1.3). This is important because the power of the 3D data representation lies in its capability of allowing users to see the cyber-space from arbitrary view angles.

1.2 Problems

This pipeline of 3DTI is severely resource demanding, as illustrated in Figure 1.5. Temporally, the one-way latency (or end-to-end delay from the capturing site to the rendering site) for interactive applications needs to be small (e.g., no more than 150 milliseconds), and the frame rate needs to be reasonable, preferably

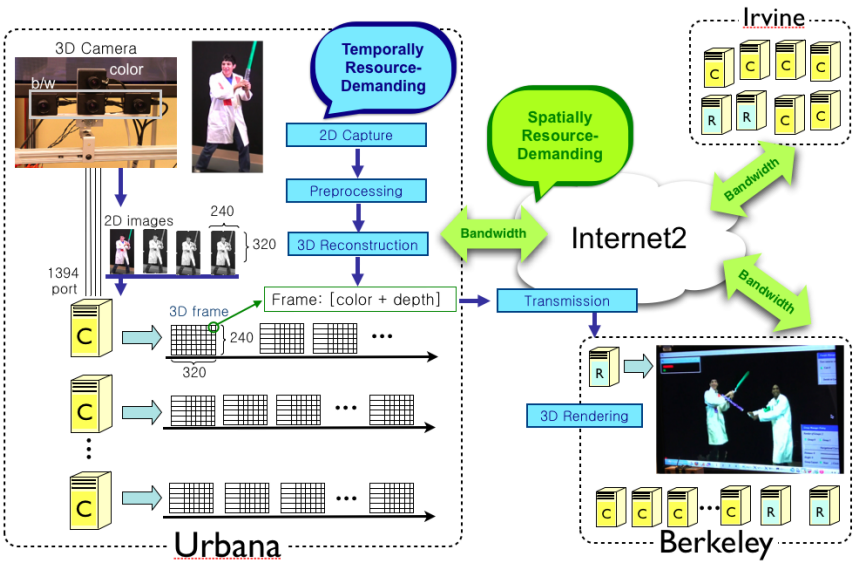


Figure 1.5: 3DTI is temporally and spatially resource-demanding. Three sites are shown: Urbana, Berkeley, and Irvine. The capture/3D reconstruction modules are detailed in Urbana site (the computers labeled with "C" are the camera host computers, and there are multiple for multi-view capture). The 3D rendering/visualization modules are detailed in Berkeley site (the computers labeled with "R" are the renderer host computers, and multiple are present for multi-view rendering).

over 10 frames per second (fps), meaning that each frame has to be processed within 100 milliseconds. This is non-trivial because the operations that need to be applied on every frame (such as 3D reconstruction, rendering as shown in Figure 1.5) are very expensive. We refer that as the **temporal challenge** in the environment. Spatially, the use of 3D representation and multi-view capturing leads to a high demand on network bandwidth, because within a stream not only color but also depth information is encoded and multiple streams need to be sent from each site for different views. With 10 cameras running at a pixel resolution of 320x240, a frame rate of 10 fps, 5 bytes for each pixel (3 bytes for RGB color and 2 bytes for depth), 10 cameras, for example, the outgoing traffic alone would require $10 \times 320 \times 240 \times 10 \times 5 \times 8 = 307$ Mbps of bandwidth. Obviously, the resource demand directly depends on the amount of video data to process and/or disseminate. We refer to that as the **spatial challenge** in this thesis.

Great research efforts have been devoted to making the systems more resource-efficient [13][39][59][79][89][99], but the focus has been primarily on *system-centric*, algorithmic optimizations of the video functions (Figure 1.2) or data reductions that leave end users out of the loop. Therein, resource allocation runs without considering the user inputs, preferences, or semantics. Data are not prioritized according to user needs but rather treated equally for resource competition. Tradeoffs among different Quality-of-Service (QoS) metrics are not

carefully examined in terms of their impact on the overall perceived quality or Quality-of-Experience (QoE) of users. Such approaches are inherently limited because the human factors, an important and integral part of the cyber-physical tele-immersive environment, are neglected.

Without more intelligent data adaptation and accompanying algorithms for controlling video functions, immersive tele-immersive communication would not be possible with today's hardware and communication infrastructure. The immense amount of data generated from multi-view cameras in each site causes long computing delay - hurdling the interactivity (e.g., frame rate) of the systems. It also causes network congestion that in turn causes packet loss (flickering or freezing effect on the display), prolonged network delay (inconsistency across sites), and decreased or unstable frame rate, all of which negatively impact user experience. In another word, to promote the overall system performance and eventually - the overall perceived quality by the user - we must develop intelligent and efficient adaptation schemes on tele-immersive video data.

However, this is a grand challenge with three main unanswered research questions. First, how do we reduce/adapt data to address the temporal and spatial challenges aforementioned in a way that does not hurt the perceived visual quality but actually improves it? Eliminating data is beneficial for lessening the resource load, but it intuitively comes at the cost of sacrificing the visual quality as e.g., details are removed, or resolutions are down-sampled. How do we intelligently reduce/adapt data to achieve better perceived quality for users? Second, how do we improve the performance of the systems given the stringent resource constraint? Efforts to improve qualities in 3DTI are largely complicated by the dire demand on temporal and spatial resources even with data reduced. For example, in Chapter 4 we will see how the resource constraints render the data dissemination problem to be NP-complete. Then how do we efficiently improve qualities subject to the constraints? Third, how do we quantify user experience, and how does it relate to system performance? What is user experience? Does it simply refer to user satisfaction? Or is it equivalent to the perceived quality of video? How do technical metrics (e.g., frame rate, delay) impact user experience? How do we measure their relationships? We refer to this as the **quality challenge** in the thesis.

1.3 Our Approaches

1.3.1 Human-centric-ness

In this thesis, we take the paradigm shift towards the Human-Centric Computing (HCC) model in addressing the above research challenges. HCC represents a set of principles and strategies that "bear the human focus from the beginning to the end" [53]. Since the ultimate goal of tele-immersion is to deliver compelling experience to end users, we believe that taking a more human-centric perspec-

tive is crucial. We argue that human factors (such as user inputs, preferences, semantics, and limitations etc.) are an essential part of the cyber-physical tele-immersive systems, and should be carefully taken into consideration throughout the design, development, and evaluation process of 3D tele-immersion.

The HCC model empowers us to center our thinking around users, and make according design choices throughout the development and implementation of the systems. The instillation of human-awareness allows us to develop new approaches for managing the daunting resource demand in the complex cyber-physical tele-immersive systems. In the thesis, we demonstrate that by taking human into our control loop, we can have better understanding in various aspects of the 3DTI systems, such as (1) which image details are actually imperceptible to human eyes, thus unnecessary (Chapter 3), (2) with the same resources, whether one should rather want lower frame rate with higher spatial resolution or higher frame rate with lower spatial resolution (Chapter 3), (3) which streams are semantically more important to users at a given point of time (Chapter 4), and (4) which is more important - increasing the number of views or reducing end-to-end latency - in terms of improving perceived usefulness of the systems (Chapter 5).

Our thesis statement thus follows.

> *Human-centric approaches are important and useful for yielding the best and stable overall quality of 3D tele-immersive systems under resource constraints.*

The key insight is that we can exploit the semantics and constraints of users, reduce/adapt data and control the video functions (shown in Figure 1.2) accordingly to achieve the most effective improvement on the overall quality.

1.3.2 Proposed Solutions

This thesis essentially proposes a *comprehensive*, *human-centric* framework for managing video data and functions across the tele-immersive pipeline. Our approach is comprehensive because it involves all components of the video function pipeline. It is also comprehensive in the sense that both temporal and spatial resource challenges are considered and tackled. Figure 1.6 gives an overview of the framework. Note that following the HCC model, the cyber-physical 3DTI system not only contains the technological environment, but also the human users in distributed sites. Our high-level methodology is to monitor, either online or offline, various machine and human states of the cyber-physical environment, and dynamically adapt data and control the video functions inside. Below we first present an overview of our solutions and then provide more details.

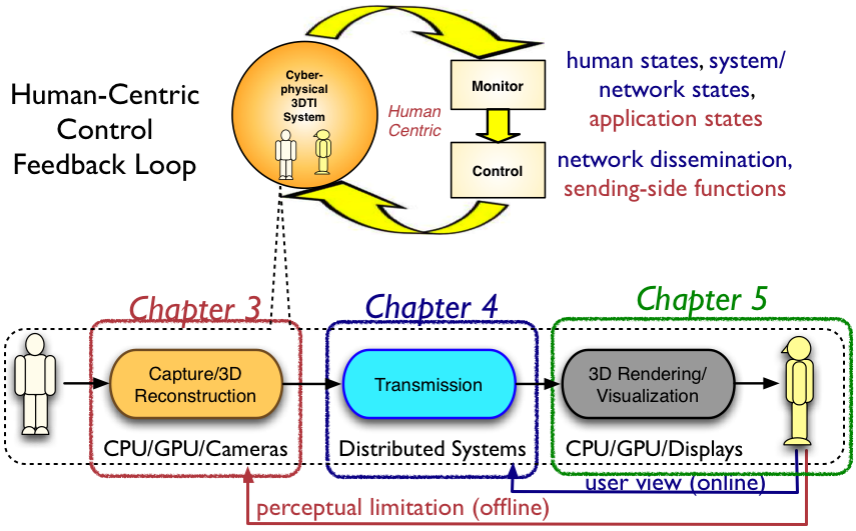


Figure 1.6: Overview of the thesis: we essentially propose a comprehensive, human-centric framework for managing video data and functions throughout the tele-immersive pipeline.

Overview

As an overview, in addressing the temporal challenge as mentioned in Section 1.2, we first reduce and adapt data on the frame level within each video stream in the first stage of the pipeline, i.e., capturing/3D reconstruction (Figure 1.6). This solution, called "intra-stream data adaptation", substantially reduces the processing time for each stream and therefore improves interactivity (with better frame rate and lower delays) and overall perceived quality of the systems. This part of work is described in Chapter 3.

However, with multi-view cameras capturing a scene, multiple video streams (even internally containing less data) still pose a challenge for the bandwidth resource. In addressing the spatial challenge as mentioned in Section 1.2, we reduce and adapt data further on the stream level (or more specially, by dropping unimportant streams) in the dissemination process (Figure 1.6). This scheme, called "inter-stream data adaptation", significantly reduces the demand on network bandwidth and therefore improves spatial resource efficiency and overall perceived quality of the systems. This part is presented in Chapter 4.

The first two parts of our work as described above demonstrate that user experience can be improved by taking human-centric approaches in the design and implementation of the work. But how about evaluation? Up to now, we have only considered various performance metrics of the system (e.g., latency, frame rate) as well as different experience metrics of the users (e.g., perceived video quality), but have not yet systematically considered the term "quality" of the system. In addressing the quality challenge as mentioned in Section 1.2, we consider the last stage of the pipeline, i.e., visualization/3D rendering (Figure 1.6)

where users experience the whole collaboration, and systematically study the "quality" concept in 3DTI systems. This part is detailed in Chapter 5.

In the following, we present more details of the three major pieces of our thesis.

Details

1. **Chapter 3** - where we first tackle the **temporal challenge** (described in Section 1.2) with an **intra-stream data adaptation** scheme.

   - *Challenge considered* - The problem of low and flickering frame rate (e.g., only 3-5 fps) has been plaguing today's tele-immersive systems. Higher interactivity (e.g., at least 10 frames per second) is desired for the physical activities in tele-immersion that involve a large amount of body motion, such as dancing and playing basketball. As we know, data reduction is definitely an effective way of reducing the processing/transmission delays in the systems and hence improving interactivity, but improper dropping of pixels or frames may degrade perceived quality of the video. How do we smartly reduce/adapt data in addressing the temporal challenge, and where do we address it in the pipeline?
   - *Approaches and insights* - In Chapter 3, we develop an intra-stream data adaptation scheme that addresses the challenges by *transparently* reducing spatial details that are imperceptible to human eye. The scheme, designed in the first stage of pipeline - capture/3D reconstruction, executes on the camera host machines (Figure 1.4) to reduce processing/transmission delays throughout the system. Our key idea is to psychophysically study the detection thresholds for the spatial resolution in tele-immersive video, and utilize them for reducing data. The insight is that imperceptible details can be excluded from the frames without compromising the perceptual quality.
   - *Human-control loop* - In the context of the control loop (Figure 1.6), the states being monitored are the application states including reconstruction time, frame rate, per-frame spatial resolution, and so forth; the functions being controlled are the sending-side video functions in the capture/3D reconstruction phase. The resource constraints considered are mainly CPU processing cycles (time) in all stages of 3DTI, as well as human perceptual limitation which we actually exploit for the purpose of perception-based adaptation. The human-centric-ness here refers to our approach of exploiting human visual limitation and reducing imperceptible spatial resolution so as to reduce time overhead (for shortened end-to-end delay and improved frame rate) throughout the pipeline.

2. **Chapter 4** - where we tackle the **spatial challenge** (Section 1.2) with an **inter-stream data adaptation** scheme.

   - *Challenges considered* - Bandwidth demand is one of the key challenges in 3DTI. Even with the spatial details internally reduced within each stream using our intra-stream scheme described above, the multi-view capturing setup of 3DTI environments (Figure 1.3 and Figure 1.4) still requires a large amount of bandwidth for delivering the multi-view camera streams. As described in Section 1.2, the outgoing traffic from a tele-immersive site can be as high as 300 Mbps. Let's assume a reduction of 75% by intra-stream schemes (Chapter 3), the resultant demand on bandwidth can still be as daunting as 75 Mbps for each site. When the number of tele-immersive sites increases, the demand will be increasingly severe. So how do we achieve further data reduction/adaptation to avoid network congestion? Also, how do we disseminate the video data across multiple collaborating sites under the stringent resource constraints (on bandwidth, latency) as mentioned in Section 1.2?

   - *Approaches and insights* - In Chapter 4, we develop a complementary inter-stream data adaptation scheme to work with the intra-stream data adaptation scheme in Chapter 3. It addresses the spatial challenges by largely reducing the number of streams in network dissemination. This scheme is naturally located in the second stage of the pipeline - data dissemination/transmission, where bandwidth resources belong. The key insight is that we can prioritize video streams according to their contributions to the user views, i.e., if a user is looking at a peer from a front view, then those cameras capturing the peer's back are not important. Thus we can only disseminate those streams that are important to a user's viewpoint in the virtual world. Such a reduction of streams leads to a substantial reduction of network congestion, and in turn leads to an improvement on the overall interactivity (with less packet loss and/or retransmissions that would have been caused by congestion). However, even with important streams identified, the construction of a multicast topology subject to the resource constraints remains a major challenge. We explore several heuristic algorithms and compare their performances under various conditions.

   - *Human-control loop* - In the context of the control loop (Figure 1.6), the states being monitored are the human states (user views in all sites), and system/network states (bandwidth availability), and the functions being controlled are the network dissemination functions. The resource constraints considered include the network bandwidth bound (locally per site), and the end-to-end latency bound (glob-

ally across sites). The human-centric-ness comes into play where we prioritize stream set and distribution topology based on users' viewpoints.

3. **Chapter 5** - where we tackle the **quality challenge** (Section 1.2) with a comprehensive **quality framework**.

   - *Challenges considered* - the first two parts of our work focus on the design and implementation processes of a system. Recall that HCC refers to the paradigm where one bears the human awareness from the beginning to the end (Section 1.3.1). At the "end", we consider a simple yet important research question - what does "quality" mean in 3DTI? What are the qualities that matter to the users in 3DTI? Chapter 3 and 4 touch upon several important quality metrics including latency, frame rate, visual quality, etc., but we have not systematically study the quality concept in 3DTI systems, particularly at the rendering end. Essentially, what is the Quality-of-Service (QoS) from the system's perspective, and what is the Quality-of-Experience (QoE) from the user's perspective? What are the measurement methodologies of them? What are their relationships? What are the taxonomies of all the quality metrics?
   - *Approaches and insights* - In this chapter, we take a human-centric perspective at the concept of "quality" in 3DTI. We develop a comprehensive QoS-QoE framework to address the above challenges. Particularly, we classify the quality metrics for the user-level QoE construct to quantify user experience, as well as the application-level QoS construct to quantify system performance. We propose a quality framework to define them, measure them, and more importantly, correlate them.
   - *Human-control loop* - In the context of control loop, this quality framework monitors human states (QoE dimensions) as well as system states (QoS parameters) offline. The results provide practical implications on which qualities to optimize, how to evaluate their impact, and how to improve the control of video functions more effectively in all stages of the pipeline.

1.4 Contributions

The most important contribution of this thesis is a general, holistic, human-centric framework for data management and control in tele-immersive environments under stringent resource constraints. We bear the human-awareness from the beginning to the end in our design, development, and evaluation. We leverage various human factors (including semantics, preferences, control patterns,

and limitations) in the environments to achieve more intelligent and effective control on the data plane. As multimedia systems become increasingly interactive and complicated with all types of sensors, we foresee many opportunities to apply such framework in achieving the best resource usage in constrained environments. Although we focus on video data in this work, we believe our approach is generally applicable on other sensory data modalities such as audio and haptic. It is also clearly extendable to other applications beyond 3D tele-immersion, such as multi-camera surveillance systems and multi-sensory assistive living homes.

The other contributions of this thesis can be summarized as follows:

- To the best of our knowledge, we present the first human-centric design and evaluation framework for 3D tele-immersive systems. We demonstrate that human-centric approaches have unique benefits for resource adaptation in the challenging 3D tele-immersion domain.

- We are the first to conduct a psychophysical study to measure the perceptual thresholds of a critical factor in 3D tele-immersive video stream, called "color-plus-depth level-of-details" (Section 3.2 - Section 3.3). We show that a significant amount of degradation on this factor would not be noticeable to average users. This is (to our best knowledge) the first attempt to apply psychophysical principles and perception-based methodologies into 3D tele-immersive systems.

- Leveraging the results from the psychophysical study, we design and implement an intra-stream data adaptation scheme for 3D tele-immersive video (Section 3.4). Our approach is able to adapt the video quality in a way that reduces considerable amount of data while preserving the perceived visual quality within stream. Our experiments show that our scheme can actually significantly improve the overall perceived quality of 3D tele-immersive systems with the use of much less resources. Over 90% of the about eighty users we sampled reported that they found the adapted video better than the unimpaired video.

- We are among the first to study the characteristics of view control in 3D tele-immersive space (Section 4.2). The subjective evaluation results offer practical implications for our design of inter-stream data adaptation schemes.

- We propose human-centric inter-stream adaptation algorithms by exploiting the viewpoint of users (Section 4.5 - Section 4.6). We show that a considerable amount of network bandwidth can be saved by our approach, while the perceived visual quality of users is maintained.

- To the best of our knowledge, we are also the first to develop a human-centric QoS-QoE quality correlation framework for 3DTI (Chapter 5). The

framework represents a significant step towards characterizing and classifying all the quality metrics in 3D tele-immersion as a human-centric paradigm shift.

2 Literature Review

2.1 Data Reduction and Adaptation

The ultimate goal of 3D tele-immersion is to enable people to interact across distance just as if they were co-located physically. This is achieved by fusing the 3D representations of geographically distributed users into a virtual reality environment in real time. The history of 3D tele-immersion can be traced back to about a decade ago, when the researchers demonstrated the first networked tele-immersive application that could run at 2 to 3 frames per second (fps) [99]. Various efforts have been made to improve the efficiency of the systems in diverse components such as depth reconstruction [58][111], coordinated data transport protocol [79], rendering [100], as well as real-time 3D video compression [112]. Despite the notable improvement, tele-immersive systems are still far from becoming a commodity due to the high interactivity demand and heavy computational complexities. In Chapter 3, we tackle the challenge from a different perspective by examining data redundancy in terms of psychophysical principles. We believe our approaches are orthogonal to the system-centric algorithmic improvements, and thus can be combined to provide greater performance benefits.

In fact, psychophysics is not new to the multimedia community. The JPEG codecs [50], for example, compress images by eliminating high frequency details that are invisible to human eyes. Audio compression algorithms, such as MP3, exploit psychoacoustic principles to reduce information that is less audible to human ears [41]. Recently, psychophysics is also being applied to haptic feedback where the samples with imperceptible changes are removed from network transmission [95]. Perhaps the most relevant to our work is the recent psychophysical study conducted by De Silva *et al.* that considered the Just Noticeable Difference in Depth (JNDD) in 3D video [91]. However, the context is very different from this work in that the video content therein is for offline-generated 3D-TV. The real-time requirement of tele-immersive video leads to the emergence of a new definition of spatial resolution (or level-of-details) that is not applicable in 3D-TV video (this will become more apparent in Section 3.2). We also develop (to our knowledge) the first perception-based adaptation scheme for 3D tele-immersion.

2.2 Data Dissemination

The streaming of 3DTI systems often involves multiple sites, with each participating site being the source of multiple 3D data streams and also the receiver of many more streams from the other peers. Most existing peer-to-peer media streaming solutions focused on the topology construction for a single session/stream [17][42]. The coordination among co-existent, competing streaming sessions was not considered. Ott *et al.* [80] studied the coordination of multi-stream delivery for tele-immersive systems. However, the protocol was designed for two-site collaboration. The interconnection and topology construction among multiple sites was not addressed.

The streams produced from a tele-immersive site are semantically correlated. Hosseini *et al.* [44] studied the dynamic topology management in video-conferencing among peers, but they assumed the streams were independent and identical in terms of priority. We find that when the demand or stress for system resources is overly high, it is crucial to take a prioritized approach, where the limited resources are allocated first for the most important data. We exploit and take advantage of the semantics of the streams, and prioritize them based on the user view. Understanding the semantic correlation among streams is critical in the new generation of video-mediated systems like 3D tele-immersion. Only by differentiating the streams by their importance can we design mechanisms to utilize the resources most efficiently.

Most existing work in overlay multicast targeted at maximizing resource usage (e.g., aggregate bandwidth) only [42][44], whereas in peer-to-peer media streaming systems, there are works dealing with dynamics, but mainly for node joins/leaves (churn). In 3DTI systems, the peers availability is usually assumed to be stable, but the dynamics are caused by adaptation semantic changes. Moreover, the existing solutions to the dynamic problem (e.g., peer churn) are re-active, such as soft leave [44], buffering [23], and topology rearrangement [85]. We aim to minimize interference pro-actively. That is, we select a peer node that is less likely to lose the requested stream due to system dynamics. This approach is orthogonal to the existing re-active techniques, and can be combined with them to further reduce disruption.

2.3 Quality of Experience Measurement

While Quality of Service (QoS) is well defined, the meaning of Quality of Experience (QoE) is being argued. For instance, the standardization group ITU-T suggests that QoE should be represented by Mean Opinion Score (MOS), a Likert-scale rank for subjective testing of voice/video quality [5]. Beauregard *et al.* formulated QoE as "the degree to which a system meets the target user's tacit and explicit expectations for experience" [14]. Some other informal definitions are "subjective measure of a customer's experiences with a vendor",

"user perceived performance", and "the degree of satisfaction of users". In the various formal and informal definitions, QoE has been framed as a subjective single-value measure. We make the first attempt to define QoE as a multi-dimensional measure that has both subjective and objective dimensions.

The relationships between QoS and QoE have been blurry. Researchers used to think QoE as an extension or subset of QoS [49][56]. For example, perceived media quality has been a de facto standard for QoE measurement. However, this perspective is limited, as user experience is much broader than the perceived quality in a single media channel. Instead, we consider the perceived media quality as a subjectively measured QoS metric, with QoE representing the holistic experience of users under the influences of perhaps more than one media channels. Guided by theories from psychology, we propose to consider the two constructs as distinct components on a causal chain, where QoS metrics represent the environmental factors that influence QoE.

In Section 5.2.4, we also surveyed a number of existing papers in multimedia systems [10][15][19][24][34][35][47][60][66][74][87][113]. We find that our classification of QoS and QoE is a much broader framework compared to the state-of-the-art, making a superset of the qualities considered in the existing works.

3 Intra-stream Data Adaptation

3.1 Introduction

As mentioned in Section 1.2, today's 3DTI systems face significant challenges due to their huge demand on temporal resources. Many operations that need to be applied on every frame are computationally intensive, such as depth correlation and 3D rendering. On the other hand, the interactivity (or timing) requirement of 3DTI systems is very high, particularly for high-motion physical activities such as dancing and martial arts. Many of the existing 3DTI systems run at 3-5 frames per second which is barely acceptable to users. We observe that the time overhead needing to be reduced actually directly depends on the amount of data involved in the video streams. That is, if we somehow reduce video data, we can reduce the processing time, and hence improve the overall interactivity. It is challenging though because we have to make sure the elimination of data does not incur negative impact on users' perception on the video quality.

As described in Section 1.3.2, we tackle the problem with a human-centric, perception-based approach. It is known that Human Visual System (HVS) has perceptual limitations, so the research question is whether it is possible to exploit these limitations and reduce data load and/or rate without impairing much perceived quality. Actually, similar questions have been studied in traditional video-conferencing systems for factors such as jitter [18], audio-visual sync [96], latency [48], and frame rate [8][61]. However, 3DTI video possesses unique characteristics whose perceptual impact is little understood.

Perhaps the most important trait that distinguishes 3DTI from the traditional video-conferencing is its color-plus-depth video format as the visual representations to users. Therefore, the density and accuracy of the texture and depth maps is a new and critical factor in 3DTI video, which we combine and refer to as the *Color-Plus-Depth Level-of-Details (CZLoD)* factor. In this work, we make the first attempt to psychophysically study this factor in polygon-based 3DTI video and utilize its results in optimizing the resource usage of the 3D reconstruction video function (see Figure 1.6). We employ the method of limits from psychophysics [36] to examine two perceptual thresholds - Just Noticeable Degradation (JNDG) and Just Unacceptable Degradation (JUADG). We evaluate forty stimuli in four varied conditions with different contents and pixel resolution settings. The results indicate that the threshold levels are actually

fairly generous (i.e., a fair amount of degradation can be suffered) and are related to both activity type and pixel resolution. In general, fine motor activity exhibits lower threshold levels than gross motor activity, and lower resolution video exhibits lower threshold levels than higher resolution levels.

In light of the findings, we design and implement a perception-based real-time intra-stream adaptation scheme for CZLoD in 3DTI. Implemented as a closed feedback loop, the adaptor monitors various interdependent Quality-of-Service (QoS) parameters to determine the appropriate degradation ratio for CZLoD. The actual degradation, nevertheless, is achieved by controlling a *detailing parameter*, whose mapping to the degradation ratio is unpredictable as it varies with environments and activities. Thus a learning algorithm is used to learn the quantitative model of the relationship between the detailing parameter and the CZLoD degradation ratios. We evaluate the adaptation scheme in a real-world 3DTI testbed, and the experimental results demonstrate that the proposed scheme can achieve considerable improvement in frame rate without impairing perceived detailing quality. We also record the generated 3DTI video with and without adaptation respectively and conduct a crowdsourcing subjective study to compare their overall quality. The collected responses show that 96.2% of users think the video with adaptation is better than the unimpaired video.

Our main contributions can be summarized as follows: (a) we identify a new factor that characterizes the density/accuracy of depth and texture maps in the emerging real-time, color-plus-depth, polygon-based 3DTI video, (b) we describe a psychophysical study for assessing the perceptual thresholds on this factor in various conditions, (c) we apply the results to practice by developing an online adaptor and show that the perception-based adaptation scheme is beneficial in achieving visual quality enhancement as well as resource reduction.

In the following, we begin by discussing 3DTI video in greater depth (Section 3.2). A psychophysical study of perceptual thresholds is then described (Section 3.3). We present the construction of our perception-based quality adaptation scheme (Section 3.4). The chapter summarizes by discussing the limitations of the study and implications of our research (Section 3.5).

3.2 Background on 3D Reconstruction

To avoid any confusion, we have to first point out that 3DTI video is different from the commonly known stereoscopic video (as in 3D movies) which creates depth illusion with two offset imaging sequences for the two eyes of viewers respectively. Unlike such stereoscopic video, 3DTI video refers to the *color-plus-depth* video, created and visualized *in real time*.

In this section, we describe the generation of color-plus-depth frames in greater detail. As Figure 3.1 illustrates, after the raw frames are fetched from the stereo camera, they are first preprocessed (e.g., resizing, rectification). Then one

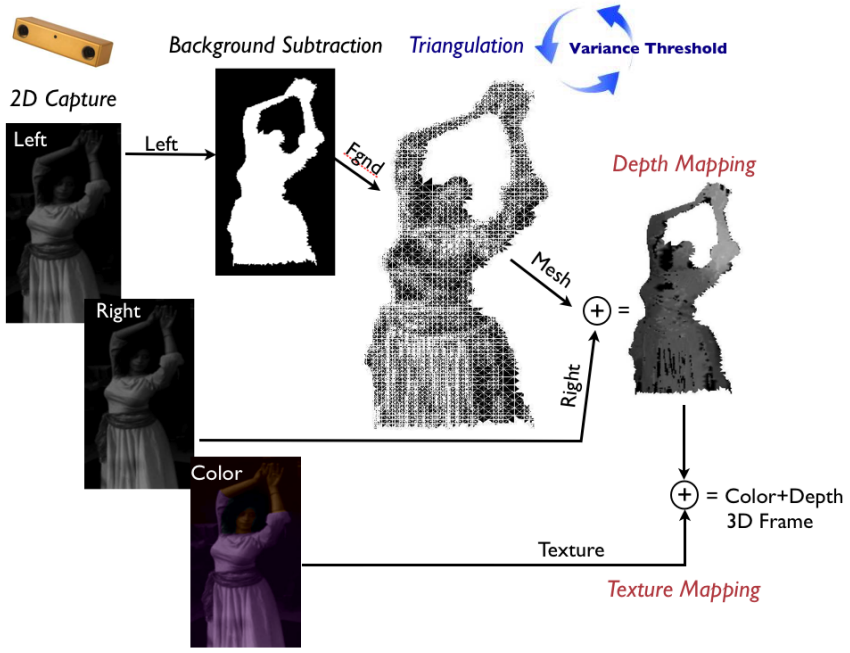


Figure 3.1: Hybrid depth mapping process: expensive depth cross-correlation is only performed on vertices after meshing, and depths for the other pixels are linearly interpolated.

of the images is used as the "reference frame" (e.g., the image from the left eye of the camera as shown in Figure 3.1), where background subtraction is performed. The next major step would be to reconstruct the 3D information of the frame for each foreground pixel. However, obtaining an accurate, dense (per pixel) depth map on commodity hardware turns out to be very time-consuming. In addition, transmitting the full-sized texture information would be quite costly in network bandwidth as well as in visualization latency, given the multi-site, multi-stream nature of tele-immersion. For these reasons, several 3DTI systems use the polygonal modeling approach [58][76][101], which is widely employed in real-time 3D computer graphics, to alleviate both temporal and spatial overheads.

Specifically, after background subtraction the reference frame is decomposed into polygons (e.g., triangles) based on texture homogeneity. This is done by recursively refining bisection until the intensity variance within every polygon is less than a threshold TH_{var}. Importantly, afterwards the expensive depth-correlation operation is only performed on mesh vertices. The depth calculation for the other pixels can thus be largely simplified by linear interpolation. Similar subdivision can be applied for textures, too. Since now only the coordinates and textures of vertices need to be transmitted (and those of the remaining pixels to be approximated at the receiving/rendering side), such region-based representations (and the accompanying hybrid depth and texture mapping algorithms) lead to a reduction of frame size as well as data manipulation time in all stages of the tele-immersion pipeline, making them favorable for the resource-intensive

tele-immersion applications.

We can observe that the number of foreground vertices after meshing regulates the color-plus-depth granularity of the mesh. It also determines the density/accuracy of 3DTI video due to the disparate treatment of vertices and non-vertices in 3D reconstruction and texture mapping. Hence, we refer to this metric as the *Color-plus-Depth Level-of-Details (CZLoD)* metric, which characterizes the spatial (including z-axial) and textural richness and accuracy of 3DTI video. Clearly, it is largely impacted by the setting of the *variance threshold* $TH_{var}(\in \mathbf{Z})$. The smaller the variance threshold is, the finer the meshing is, the more dense/accurate the depth and texture maps will be. Therefore, the variance threshold TH_{var} is a *detailing parameter* for the CZLoD of 3DTI video.

We are concerned about whether there are perceptual limits on the *degradation* of CZLoD for the purpose of data reduction. We thus mathematically formulate the metric of *degradation ratio* (DR). Suppose we denote the 2D reference frame as f_i (i is frame number), and the 3D frame generated from it as F_i. Assume $N_0(\mathsf{F}_i) \in \mathbf{N}^0$ is the number of foreground vertices computed on f_i if TH_{var} were set to 0, and $N_v(\mathsf{F}_i) \in \mathbf{N}^0$ is the number of foreground vertices computed on f_i if TH_{var} were set to v ($v >= 0$), the degradation ratio of CZLoD on the frame F_i can then be expressed as

$$DR(\mathsf{F}_i) = 1 - \frac{N_v(\mathsf{F}_i)}{N_0(\mathsf{F}_i)} \tag{3.1}$$

where $0 \leq DR(\mathsf{F}_i) < 1$.

Obviously, the lower the degradation ratio, the more vertices the frame F_i contains, the higher definition and accuracy it has in both texture and depth, and the more sharp and stereoscopic it may potentially be (depending on perceptual mechanisms).

3.3 Motivating Subjective Study

The purpose of the psychophysical experiment is to measure two perceptual thresholds of CZLoD degradation: (a) Just Noticeable Degradation (JNDG), and (b) Just Unacceptable Degradation (JUADG). Identification of these thresholds can guide us to develop perception-based CZLoD adaptation mechanism for resource saving without impairing the perceived visual quality. We employ the Ascending Method of Limits [36] as the experiment methodology. It is one of the oldest and most widely used approaches in psychophysics for determining thresholds of sensations. The methodology, originally designed to measure singular intensity such as light luminance and sound frequency, was slightly modify in order to measure degradation level by means of comparison. In our study, CZLoD conditions are presented in sequential pairs, one being an unimpaired reference, and one being the same video impaired. The magnitudes of

impairment are presented in an ascending order.

3.3.1 Stimuli Generation Engine

While 2D video quality studies can utilize a pool of 2D video sequences offered by the Video Quality Expert Group (VQEG) [2], there were no standard test data for 3DTI video. Since 3DTI is essentially a live pipeline from cameras to renderers, a naive way of obtaining test sequences would be to record different test sequences multiple times with different configuration of *treatment factors* (which refer to the sources of variation that are "of particular interest to the experimenter" [28]). However, this clearly not only requires large amount of experimenter efforts but also suffers from uncontrollable varying conditions such as captured content and illuminances. Therefore, we propose a stimuli generation engine suitable for general 3DTI video studies. To ensure the treatment factors be only varied within homogeneous blocks [73], we decouple the capturing part from the 3D reconstruction part so that different configurations could be applied during each phase but on the same exact image samples if desired.

Figure 3.2 depicts the three distinct stages of the engine. In Stage 1, a number of frames of the experiment activities are synchronously captured, preprocessed, and stored. A number of parameters can be configured at this stage, including the desired pixel resolution, whether to use rectification or background subtraction, the number of frames to take, etc. To generate lower pixel resolution images, the raw frames can be downscaled. In Stage 2, the engine retrieves the specified 2D frames and repeatedly performs 3D reconstruction with varying parameters such as the variance threshold TH_{var}, whether to use trinocular or binocular stereo matching, etc. The host computers then send their reconstructed frames to a renderer that aggregates the frames and writes them to disk storage. In the final stage, the 3D frames were replayed as stimuli with possibly varying parameter such as frame rate. In a word, the systematic decomposition allows automatic generation of stimuli with the flexibility of controlling desired treatment factors while keeping blocking and nuisance factors (e.g., content) fixated [28].

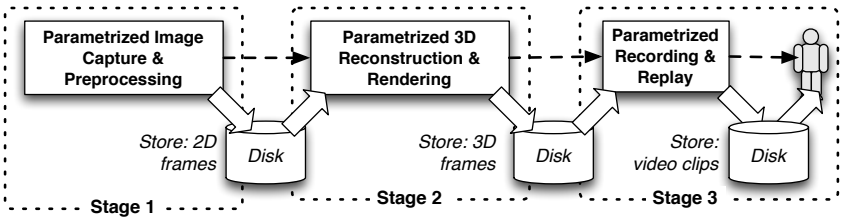


Figure 3.2: Stimuli generation engine: in Stage 1, 2D frames are taken, preprocessed, and stored; in Stage 2, 3D reconstruction are repeatedly applied with varied parameters; in Stage 3, the stored 3D frames are replayed with parameterization to generate the final stimuli.

Table 3.1: Codes for the four stimuli blocks with different contents (exercise and lego) and resolutions (low and high).

	Gross Motor Activity	Fine Motor Activity
Low Res. 320x240	Exercise-L	Lego-L
High Res. 640x480	Exercise-H	Lego-H

3.3.2 Stimuli

With the experimental methodology in mind, we generated forty stimuli, which we make available for the research community[1]. Below we discuss their conditions, properties, and resource characteristics.

There are two factors that may have impact on the perception of CZLoD impairment: sequence content and pixel resolution of raw frames [29][51][68][105] (called "blocking factors"). To explore their relationships with the treatment factor CZLoD, we created 2 (contents) $\times$ 2 (pixel resolutions) groups (called "blocks") of stimuli, each having a different configuration of the blocking factors. For the first factor - content, we categorized the most frequent 3DTI activities into two types and recorded a representative video for each type: (a) gross-motor activities such as Tai-Chi training [40], dancing [90], physical rehabilitation [65] that involve large body movement, and (b) fine motor activities such as car sale [39], telemedicine [25], cyberarcheology [32], object/tool instructions [89] that involve finer body movement (e.g., on hands) and manipulation of objects. For the former type, we recorded a person (performer) doing an elbow exercise (commonly used in telepresence physical therapies), while for the latter type, we recorded the performer showing a small Lego house where details are not only more demanding for the object but also for the finger movement of the performer. For the second blocking factor - pixel resolution, we chose two levels that had been mostly used in 3DTI systems [59][86][99]: (a) high - 640×480, and (b) low - 320×240. The four stimuli blocks were coded as shown in Table 3.1.

Then the stimuli generation engine (Figure 3.2) was employed to generate 10 levels of stimuli in each block. In the first stage, the Dragonfly2 stereo camera (Point Grey Inc.) was employed for recording the 2D images for each of the four block conditions. After 2D frames were acquired, 3D frames were generated for each block by repeatedly running binocular stereo matching on the same stored images with varied TH_{var} for obtaining different CZLoD degradation settings. Finally, the 3D color-plus-depth frames were rendered with a fixed frame rate 10 fps (chosen according to [61]) for the subjects. Following the ITU-R BT.500 standard [51], each test sequence was 10-second long, so about 100 frames were included.

The 10 stimuli for each block were coded as $\mathsf{S}_0, \mathsf{S}_1, ...\mathsf{S}_9$ with increasing levels of degradation ratio in CZLoD (Figure 3.4), with an approximate step size

[1] http://cairo.cs.uiuc.edu/projects/tele-immersion/datasets/mm11-czlod

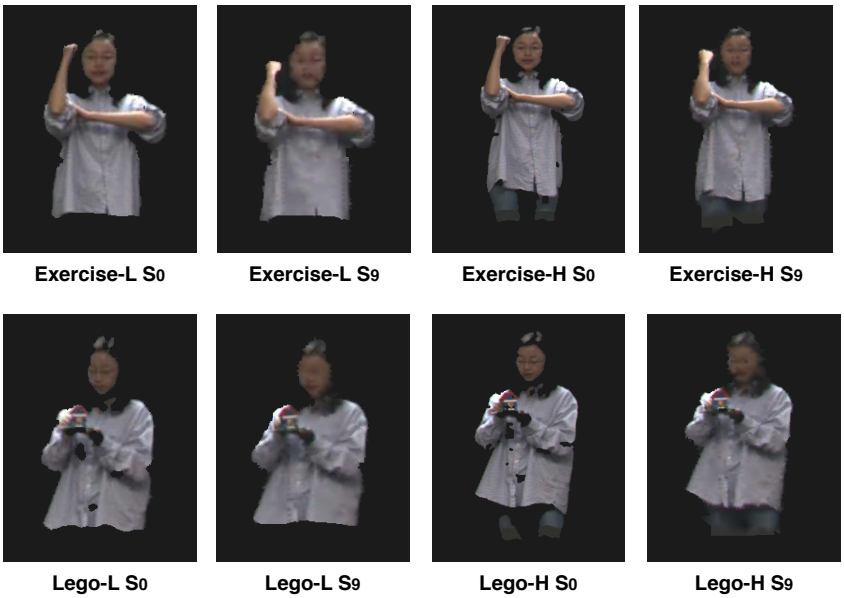


Figure 3.3: Stimuli snapshots: please refer to Figure 3.4 for the stimuli codes and their degradation ratios.

of 10% (degradation ratio). Figure 3.3 shows the snapshots of the lowest and highest stimuli conditions for each block. For each stimulus, the degradation ratio was calculated by averaging across all frames (relative standard deviation measured to be 2.08% - 2.83% for all stimuli). Therein, S_0 was the unimpaired reference stimulus ($TH_{var} = 0$). The TH_{var} values for other stimuli were manually chosen to approximately achieve the expected degradation ratio (it was impossible to be exact). Two sets of TH_{var} values are used, one for the lower pixel resolution blocks (Exercise-L/Lego-L), and the other for the higher resolution blocks (Exercise-H/Lego-H). Figure 3.5 presents the actual number of triangles and vertices after the meshing process.

To demonstrate that varying CZLoD level may indeed potentially save resources, we also measure the resource usage of the stimuli as presented in Figure 3.6. It is clear that the frame processing time (left axis) and frame size (right axis) decreases almost linearly as the degradation level grows.

3.3.3 Participants, Procedures, and Apparatus

- *Participants.* We followed the ITU standard in conducting the experiment [51]. Sixteen adult participants were recruited, primarily graduate students and staff in Department of Computer Science at University of Illinois at Urbana-Champaign[2]. All had normal or corrected vision. Four participants were Indian, three were American, two were Chinese, two

[2]The exact age distribution is unknown because some subjects expressed unwillingness to disclose age.

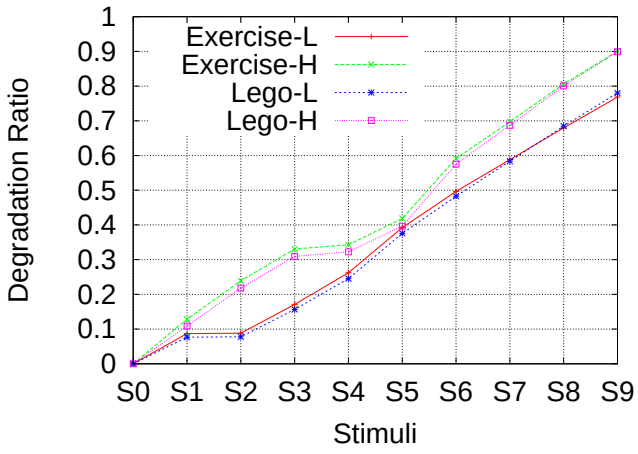


Figure 3.4: The degradation ratios gradually increase with the stimuli levels.

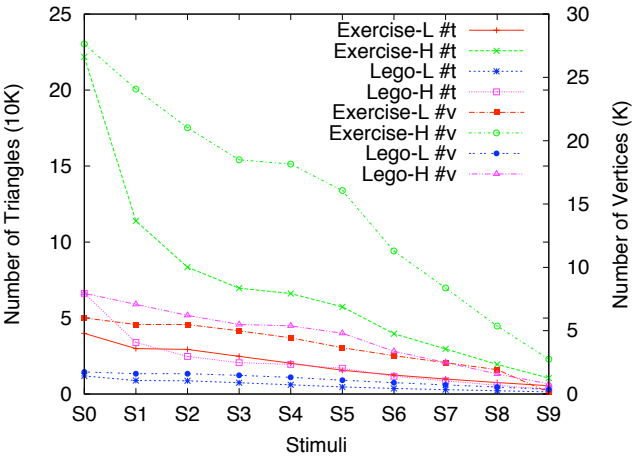


Figure 3.5: The actual number of triangles (#t) and numbers of vertices (#v) gradually decrease with the stimuli levels.

were German, three were Bangladeshi, one was Mexican, and one was South African. The sample consisted of 6 women (37.5%) and 10 men (62.5%). Regarding the level of experience with 3DTI video, the sample consisted of 5 experts (31.25%) and 11 novices (68.75%). A subject was labeled as experts if he/she had at least two years of research experience in tele-immersion, or as novices if he/she had seen 3DTI video at most a couple of times if not at all.

- *Procedures.* The sequence of blocks presented was: Exercise-L, Exercise-H, Lego-L, and Lego-H. Figure 3.7 shows the experimental process (adapted from [51]) within each block. Pairs of stimuli were presented automatically using a script with the ascending levels of degradation. For each pair, the first video was the unimpaired reference video, shown to mitigate memory effect [81], and the second video was the impaired one. In between the pair,

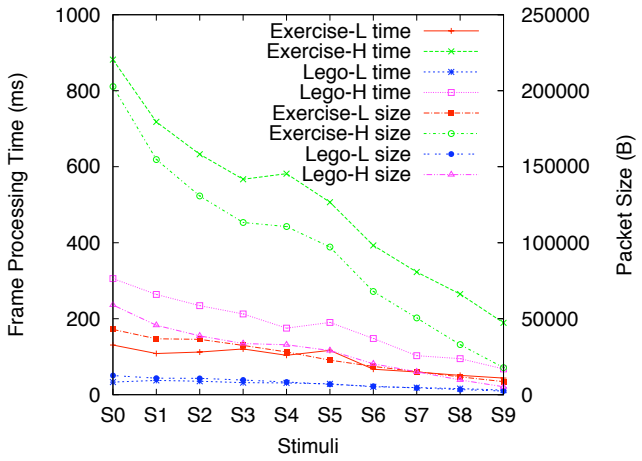


Figure 3.6: The frame processing time (left axis) and frame size (right axis) decrease almost linearly as the degradation level grows.

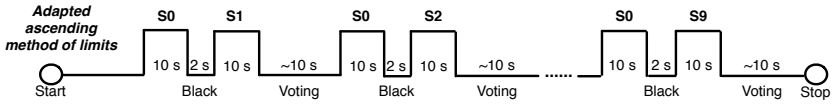


Figure 3.7: Experiment procedure: sequential {unimpaired, impaired} pairs of stimuli were shown, with ascending degradation ratios. Each stimulus was 10-sec long, the interval showing a black screen was 2-sec long within pair, and the voting period between pairs was about 10-sec long [51].

there was a 2-second interval with black screen [51]. The voting period after each pair was about 10 seconds long, when the observer was asked if he/she could tell any difference between the two clips, and whether he/she thought any video had unacceptable quality. The subject was told that they could take break any time during the experiment.

- *Apparatus.* The experiment was conducted in the MONET (Multimedia Operating and Networking) laboratory at the University of Illinois at Urbana-Champaign. Participants were asked to be seated in front of a LCD monitor during the experiment with a standard viewing distance [51]. The detailed specification of the monitor used is listed in Table 3.2. 3D displays were available but not used mainly for usability concerns. Despite their rapid growth, today's state-of-the-art 3D displays are not yet ready to be deployed for 3DTI activities. For example, typical stereoscopic displays require observers to wear goggles to perceive the depth effect, which is intrusive and thus unsuitable for physical activities often conducted in 3DTI environments. The new autostereoscopic displays eliminate the need of wearing glasses for viewers; however our previous experience with them indicates that the technology was far from mature as they caused considerable discomfort for viewers. Lambooij et al. gave a general review

of the visual discomfort caused by stereoscopic and autostereoscopic displays [67]. Therefore, in this experiment we still resorted to conventional displays for visualization. However, it is still worth noting that 3D displays are only to hypothetically *improve* depth perception [57], not to *enable* it. In fact, depth perception is achieved by a variety of visual cues (such as shading, texture gradient, linear perspective, motion parallax, occlusion, etc.) that are still relevant in 3DTI video regardless of the type of display used [45]. We chose to trade the possible increase of depth perception for the visual comfort of users which was believed to be more important.

Table 3.2: Detailed specification of the monitor used in the psychophysical experiment.

LCD Monitor Model	Acer X222W
Dimensions (WxDxH)	51.4 cm x 20.4 cm x 41.8 cm
Resolution	1680 x 1050 / 60 Hz
Dot Pitch	0.282 mm
Response Time	5 ms
Brightness	300 cd/m2

3.3.4 Human Study Results

In psychophysics, perceptual thresholds are defined to be the stimuli intensities (in our case, CZLoD degradation ratios) that can be detected/accepted some p portion of the time, and $p = 50\%$ is often used [36]. Figure 3.8(a) shows the measured JNDG and JUADG thresholds in four blocking conditions using probability 50% (equivalent to taking medians in the data). There are several observations.

- *Existence of perceptual thresholds:* There do exist perceptual thresholds on the magnitude of CZLoD degradation that the viewers can detect and accept. The average JNDG across all conditions is 61.5%, suggesting that degradation below 61.5% is not noticeable to average users (Equation 3.1). This implies that we can *transparently* reduce a considerable amount of resource usage by degrading CZLoD without actually impairing the perceived quality. The existence of JUADG (average 81.25%) indicates the degradation should be bounded by this upper limit otherwise it might make the overall quality unacceptable.

- *Impact of pixel resolution:* JNDGs in both content blocks (Exercise and Lego) are lower for the two 320x240 conditions (Exercise-L/Lego-L) than for the corresponding 640x480 conditions (Exercise-H/Lego-H), indicating that it might be easier for subjects to notice degradation with lower pixel resolution than with higher resolution. This is possibly because lower resolution already loses many details than the higher resolution (in our

case, four times); thus any further degradation would become more noticeable. Likewise, JUADGs of Exercise-L/Lego-L are lower than those of Exercise-H/Lego-H.

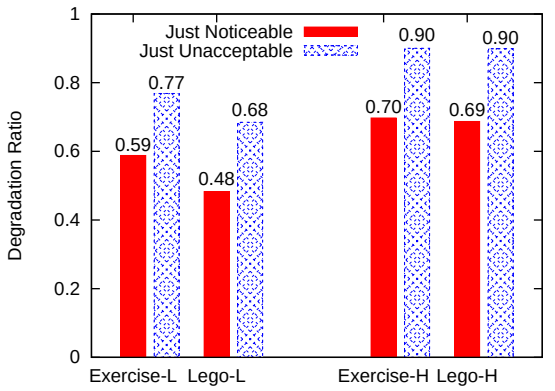
- *Impact of content:* Within the same pixel resolution condition, the thresholds for the 320x240 resolution (Exercise-L/Lego-L) vary with content, while those for the 640x480 resolution (Exercise-H/Lego-H) do not. Exercise-L has a higher JNDG than Lego-L, meaning that it is harder to notice the degradation in the exercise scene that contains only elbow movement than in the Lego video involving finer granularity object and finger movement. This is partly due to the fact that the arm in elbow exercises requires much less details than the Lego object. Since viewer attention tends to focus on the arms in motion, any changes in the other part of the video tends to get unnoticed, a phenomenon often referred to as "change blindness" in visual perception [92]. Similarly, the tolerance for degradation (JUADG) is higher in Exercise-L than in the Lego-L.

Figure 3.8(b) presents the cumulative probability distribution of the response data. We have the following observations.

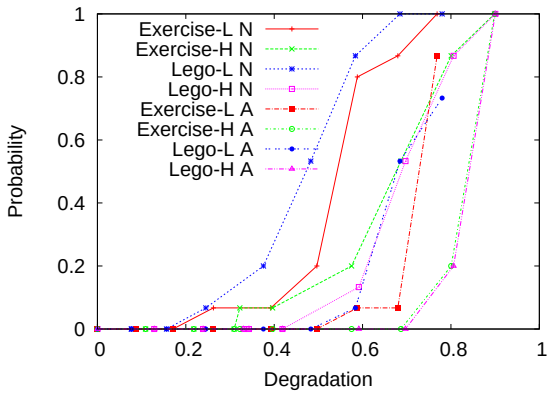
- *Relationship between noticeability and acceptability thresholds:* For every condition, the **A** curve is always on the right side of the **N** curve, which indicates that as degradation increased, at some point subjects would start to notice the distortion yet feel it was still acceptable, but after some further degradation, they would start to feel the video quality was becoming unacceptable.

- *Noticeability and acceptability offset:* The offsets between the **N** and **A** curves in each condition are similar - mostly close to 10%-20%. Considering the step size in our stimuli was about 10%, this means about 20%-30% more degradation than the noticeable region would make the quality be perceived as unacceptable. The reason why we consider step size is that, say for the first six stimuli levels presented (S_0 versus $\mathsf{S}_1, \mathsf{S}_2, ..., \mathsf{S}_6$, Figure 3.7), the subjects did not notice a difference between the unimpaired and impaired video, but at S_7 they suddenly started to notice the degradation (say, of ratio 70%, meaning JNDG = 70%), it is unclear any ratio between that of S_6 and S_7 is detectable or not due to the discrete stimuli levels we used. Hence, it is safer to take the step size into consideration here.

- *Impact of pixel resolution and content:* The observations we have from Figure 3.8(a) about the impact of the two blocking factors (when p is 50%) are also generally true for other p values (y-axis). For example, Lego-L has lower JNDG and JUADG than Exercise-L. Lego-H and Exercise-H mostly have the same responses. The lower resolution blocks (Exercise-L



(a)

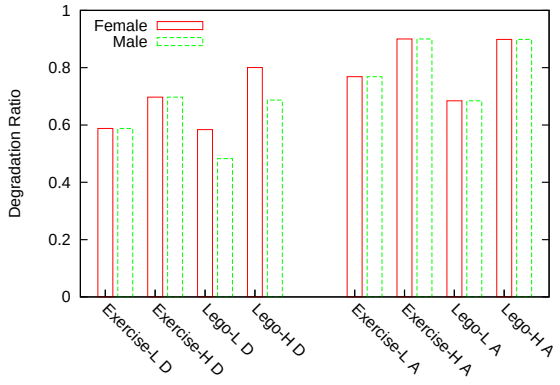


(b)

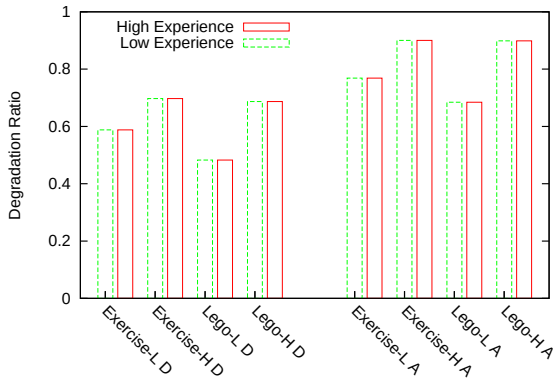
Figure 3.8: (a) perceptual thresholds on four conditions taken by medians, and (b) psychometric curves of JNDG and JUADG (in the legends, "N" stands for JNDG, and "A" stands for JUADG).

and Lego-L) generally have lower thresholds than the higher resolution blocks (Exercise-H and Lego-H).

We also compare the responses based on gender and level of experience, yet have not observed significant impact (Figure 4.11). Regarding gender, most results are the same for male and female subjects, only except that in Lego activities, female subjects detect the degradation one level earlier than their male counterparts. For all other conditions, the thresholds of female and male viewers are exactly the same. As for the level of experience, experts and novices show no difference in any JNDG or JUADG threshold. In a word, we have not noticed significant distinguishing effect of gender or level of experience on the perceptual thresholds measurement.



(a)



(b)

Figure 3.9: No significant impact of (a) gender or (b) experience was found.

3.4 CZLoD-based Intra-stream Adaptation Scheme

3.4.1 Overview

QoS parameters are characterized by spatial (intra-frame) and temporal (inter-frame) requirements. For 3DTI video, the spatial parameter refers to the spatial (including z-axial) resolution, and the temporal parameter corresponds to the frame rate. Naive 3DTI applications are often implemented without considering the temporal-spatial balance. The results from our psychophysical study suggest that CZLoD provides tele-immersion developers a powerful tool to control the detail complexity of the video, and in turn control the frame processing time or the frame rate. A major implication of our findings is that transparent degradation on spatial resolution (CZLoD) is possible to achieve "free" saving on resources without users being aware of it, i.e., by degrading CZLoD to a level where the distortion is just unnoticeable.

In addition, when frame rate drops to a level that it hurts overall usability,

the degradation on CZLoD can further increase (yet within acceptable ranges) to reduce data load and thereby elevate frame rate. Furthermore, past research has implied a curvilinear relationship between spatial quality and frame rate, e.g., improvements in frame rate become less noticeable above approximately 10 frames per second [61]. Therefore, when frame rates are found to be higher than necessary, the CZLoD degradation ratio can be lessened (if possible) to recover or promote the detailing quality thereby reducing frame rate. In a word, we can manipulate the CZLoD degradation ratio to achieve a balance between the temporal quality (frame rate) and the spatial quality (CZLoD).

The thresholds obtained in our psychophysical study (Section 3.3.4) are valuable in guiding the adaptation process. When frame rate is excessively high (indicating that we may upgrade CZLoD to include more details), CZLoD does not need to be promoted beyond what the users can perceive (1-JNDG). When frame rate is excessively low (meaning that we may downgrade CZLoD to exclude some details), CZLoD should not be reduced below what the users can accept (1-JUADG).

Based on these principles, we propose a novel, human-centric, real-time, intra-stream adaptation scheme (at the sender side) for 3DTI video. We design the adaptor as a closed feedback loop [9] for the control of detailing quality in 3DTI video. Figure 3.10 illustrates the framework of the adaptation. It has three major components: QoS Monitor, Decision Engine, and Variance Calculator.

QoS Monitor is responsible for collecting and analyzing time series of QoS parameters (e.g., frame processing time, frame size, reconstruction time), and extracting meaningful information online to notify Decision Engine for triggering adaptation. Decision Engine computes an appropriate target CZLoD degradation ratio for the 3D reconstruction process. Since degradation ratio is actually controlled by manipulating the variance threshold (Section 3.2), a Variance Calculator component is hence used to compute the correct variance threshold given a target degradation ratio from Decision Engine. Yet a challenge is that the mapping from a desired CZLoD degradation ratio to a variance threshold is unpredictable due to its dependency on scenes (e.g., clothing texture, skin colors, presence of objects, lighting illuminance). Therefore, Variance Calculator dynamically learns a quantitative model between the CZLoD degradation ratio and the appropriate variance threshold. Based on the model, it computes the proper variance threshold given a target degradation ratio, and feeds it into the 3D reconstruction pipeline for video quality adaptation.

3.4.2 Design and Implementation

QoS Monitor

Various CZLoD-related QoS parameters are inter-dependent in tele-immersion. Figure 3.11 depicts the most relevant parameters and their dependencies identified using the Granger-causality graphs [84] over profile data. QoS Monitor

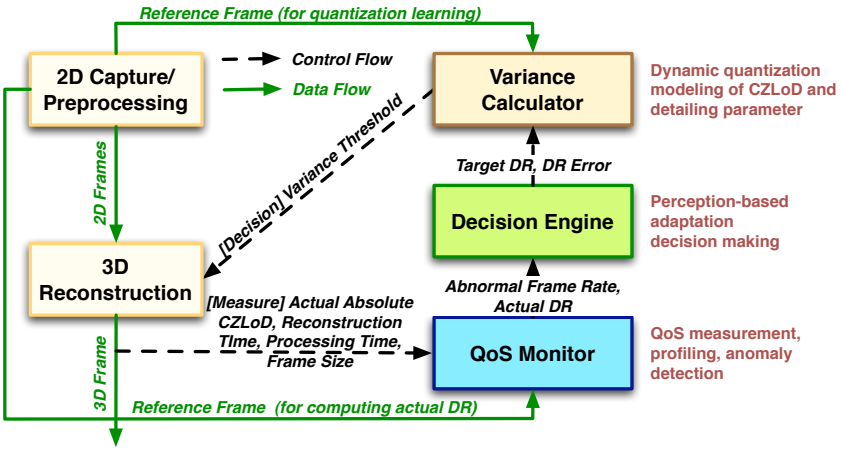


Figure 3.10: Adaptation scheme overview: refer to Section 3.4.1. DR stands for Degradation Ratio.

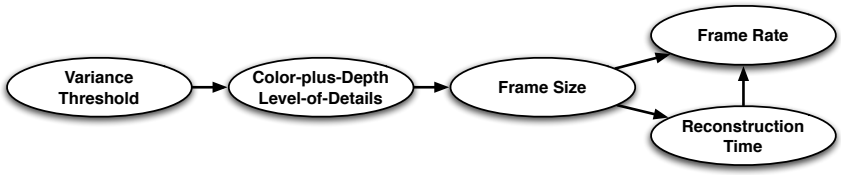


Figure 3.11: Dependency graph of QoS parameters monitored.

continuously collects time-series meta-data of these parameters for each frame and performs online analysis and profiling, and provides "feedback" to Decision Engine. The feedback includes two types: (a) *frame rate events* (excessively high or low) for triggering increase or decrease of degradation ratio, and (b) *actual degradation ratio* $DR^a(F_i)$ of every frame F_i.

Since providing real-time feedback in the control loop is a key, a simple yet efficient range checking approach is used for evaluating frame rate. Essentially, if the frame rate drops below a lower-limit threshold (TH^l_{fr}), Decision Engine is notified for increasing degradation ratio (for lower CZLoD quality); if the frame rate increases beyond an upper-limit threshold (TH^h_{fr}), Decision Engine is notified for decreasing degradation ratio (for higher CZLoD quality). The thresholds should be set according to the perceptual characteristics of frame rate [61]. Compared to the single threshold method where $TH^l_{fr} = TH^h_{fr}$, range thresholding is important for avoiding the flickering effect that can occur when a parameter constantly switches between low and high levels as it hovers near the threshold. For the same reason, the frame rate is not computed on per frame basis, but averaged over a running window of size W (frames).

Further, through our experiments we observe that transient CPU anomalies sometimes occur in systems that may cause abnormally low or high frame rate for a single frame or two. These outliers, if included in the frame rate calculation,

will damage its accuracy and in turn affect the adaptation decision. Therefore, we use a learning-based anomaly detection approach to detect and exclude these outliers when computing frame rate. The idea is that with collections of training data we first employ clustering analysis to acquire a profile of the normal range of frame rate with varying frame sizes. QoS Monitor then continuously monitors the frame size and reconstruction time of each frame F_i, and determines if it is an outlier by measuring deviation to the range means [54]. Figure 3.12 demonstrates the profile and outliers using real-world data we collected.

Apart from the frame rate reports, QoS Monitor also evaluates the actual degradation ratio of each frame F_i, $DR^a(F_i)$, and reports it to Decision Engine for taking corrective measure. $DR^a(F_i)$ needs to be measured because the sequence complexity and resource condition are constantly changing, meaning it is possible that a target degradation ratio would not be achieved exactly as desired. It is worth pointing out that the precise computation of $DR^a(F_i)$ requires the original 2D frame f_i be reconstructed by setting $TH_{var} = 0$ (refer to Equation 1). To facilitate the computation, the 2D capture/preprocessing component of the live 3DTI pipeline periodically sends a reference frame f_r to QoS Monitor, on which it applies 3D reconstruction with $TH_{var} = 0$ and computes the reference CZLoD expressed as $N_0(F_r)$ (Section 3.2). Since this is relatively an expensive operation, it is only periodically performed, which is believed to be reasonable considering that performer motion cannot change dramatically within a short period of time, i.e., $N_0(F_r)$ would be very close to $N_0(F_i)$ due to their temporal proximity. Using the latest available $N_0(F_r)$ to approximate $N_0(F_i)$, QoS Monitor can then compute $DR^a(F_i)$ (Equation 1) and report it to Decision Engine.

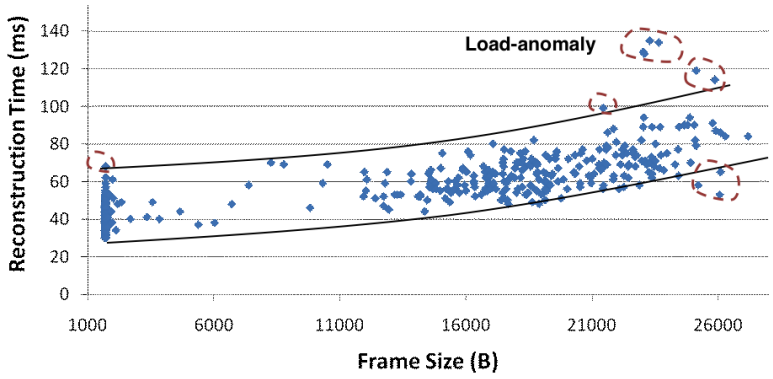


Figure 3.12: Clustering analysis of reconstruction time against frame size: for the identification of outliers to exclude in frame rate calculation. The outliers are those computed under CPU anomaly.

Decision Engine

The foundation of the adaptation logic in Decision Engine is based on the perceptual thresholds (JNDG and JUADG) on the color-plus-depth spatial resolu-

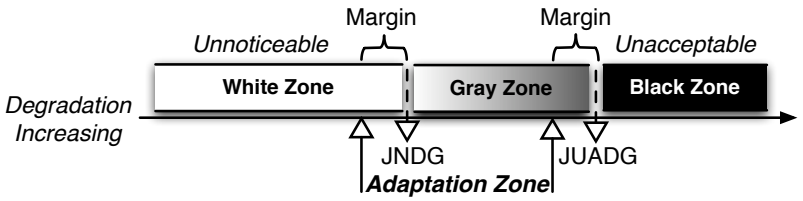


Figure 3.13: The Just Noticeable (JNDG) and Just Unacceptable (JUADG) thresholds decompose the range of CZLoD degradation to three zones.

tion of the video. The thresholds decompose the CZLoD quality of 3DTI video into three zones: *white zone* where distortion is minimally noticeable, *gray zone* where the distortion gradually becomes noticeable yet still acceptable, and *black zone* where the degradation is unacceptable. The basic idea of Decision Engine is to dynamically adjust the target degradation ratio primarily in the gray zone, except with some margins (Figure 3.13). The margins are introduced to account for the step size in our psychophysical experiment (as discussed in Section 3.3.4) as well as environmental and user dynamics. Hence, if we denote the margin size as B_n and B_a ($0 \leq B_n, B_a \leq 1$) for noticeability and acceptability thresholds respectively, the adaptation zone can be defined as $[JNDG - B_n, JUADG - B_a]$ in terms of degradation ratio.

As mentioned above, Decision Engine receives two types of information from QoS Monitor: (a) abnormal frame rate, and (b) DR^a of every frame. Upon receiving alarms of abnormal frame rate, Decision Engine computes an appropriate target degradation ratio. For this purpose, a linear control mechanism is used. Basically, an abnormally low frame rate ($FR_i < TH^l_{fr}$) means the need for lower CZLoD quality (or higher degradation ratio), thus the engine computes the target degradation ratio as $DR^t(F_i) = DR^a(F_{i-1}) + \Delta_d$ where $DR^t(F_i)$ denotes the target degradation ratio (greater DR means more degradation), $DR^a(F_{i-1})$ denotes the actual degradation ratio of the last frame F_{i-1} (reported by QoS Monitor), Δ_d denotes the adjustment size for increasing DR. This ratio is then used for all frames until the next time adaptation is triggered. Similarly, an unnecessarily high frame rate ($FR_i > TH^h_{fr}$) triggers the engine to produce a desired degradation ratio as $DR^t(F_i) = DR^a(F_{i-1}) - \Delta_u$ where Δ_u is the adjustment size for decreasing DR.

The settings of Δ_d and Δ_u can be based on various protocols, e.g., AIMD (Addictive Increase/Upgrade Multiplicative Decrease/Downgrade) or proportional to frame rate deviation from normal mean. Although such more aggressive changes may result in faster reaction time, they may also incur more abrupt changes in the detailing resolution of the video. We find that the simple constant small size is sufficiently effective in responding to frame rate anomalies while being able to maintain a gradual and graceful change that is less noticeable. However, we do need to ensure the target degradation ratio is bounded within

the adaptation zone (Figure 3.13). Simply, when increasing DR, if $DR^t(F_i)$ reaches the upper limit of the adaptation zone (close to the black zone in Figure 3.13), Δ_d is set to 0, i.e., no degradation is allowed any more otherwise the quality would become unacceptable. Likewise, for decreasing DR, if $DR^t(F_i)$ reaches the lower limit of the adaptation zone (close to white zone), Δ_u is set to 0, i.e., further improvement on the detailing quality would not be noticeable anyway thus unnecessary. Besides the calculation of target degradation ratio, Decision Engine also computes the adaptation error between the actual and target degradation which will be used for Variance Calculator (as explained below).

Variance Calculator

Given the target CZLoD degradation ratio DR^t, Variance Calculator is responsible for determining the proper value for the detailing parameter TH_{var} in the 3D reconstruction. However, the mapping $\mathcal{F}$ from DR^t to TH_{var} is nontrivial because it highly depends on external conditions such as scene complexities. Therefore, we dynamically learn a quantitative model in order to predict the correct TH_{var} value for a desired DR^t.

The learning process is performed only when Decision Engine finds that the adaptation error $err = |DR^a(F_i) - DR^t(F_i)|$ is larger than some threshold $err > TH_{err}$, meaning that significant changes in scenes might have happened that make the previous model less applicable. To learn a new model, Variance Calculator repeatedly applies 3D reconstruction on the frame f_i with exponentially increased TH_{var} values, and the resultant $DR^a(F_i)$ values are logged. This process runs in parallel with the actual 3DTI pipeline and is thus unobtrusive. Then the used TH_{var} values and their resultant $DR^a(F_i)$ values are fed into a least-square regression module to develop an exponential model as follows [16].

$$\mathcal{F} : TH_{var} = e^{a \cdot DR^t + b} \tag{3.2}$$

where e is the Euler Number, and a and b are constants. With this simple model we are able to achieve a high accuracy (median residual of 0.022%) with as few as 10 training points (please refer to Section 3.4.3). With the model available, Variance Calculator is then able to set a proper variance threshold after 2D preprocessing and before 3D reconstruction for a desired degradation ratio. Figure 3.14 demonstrates the goodness of fit (with a median residual of 1.55%).

3.4.3 Performance Evaluation

We evaluated the adaptation scheme in a real-world 3DTI system. The Bumblebee2 stereo camera (Point Grey Inc.) was used. It was connected to a host computer (Intel Xeon quad-core CPU 2.8GHz and 2GB RAM) via an IEEE

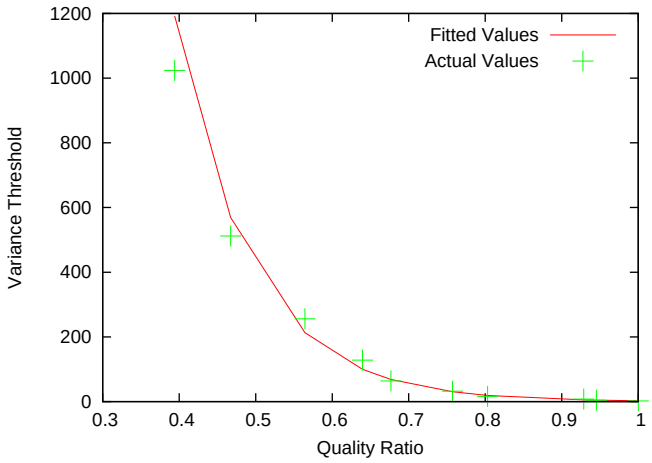


Figure 3.14: Variance learning accuracy is very high with as few as ten data points.

1394b card. The pixel resolution used was 320x240. Two professional lamps (Brightline SeriesONE) were employed to produce soft and diffused lighting condition. The renderer machine had an Intel Xeon CPU 2.3GHz, 2GB RAM, and an NVIDIA GeForce 9800 graphics card. The scene was an experimenter performing arm exercises in front of the camera.

We compared two conditions - with and without adaptation, using the same experimental setup. Both objective and subjective measurements were collected. The technical metrics such as frame rate, frame size, target and actual degradation ratios were logged, and the rendered video was recorded on the renderer for subjective evaluation. The algorithmic parameters settings were: $W = 5, TH^l_{fr} = 8, TH^h_{fr} = 12, B_n = B_a = 10\%, JNDG = 59\%, JUADG = 77\%, \Delta_u = \Delta_d = 5\%$.

Figure 3.15(a) shows the frame rate comparison. In this case, the adaptation scheme achieved an average of 27% improvement on frame rate from about 8 fps to 10 fps. According to [61], any improvement below a frame rate of 10 fps is considerably noticeable to users.

We also compared the frame rate with some CPU stress. For this, a process was run together with 3DTI that took at peak 16% of the CPU load. This could simulate the conditions where the CPU is less powerful or higher pixel resolution is configured. As Figure 3.15(b) shows, the frame rates without adaptation dropped to about 6-7 fps with several sudden dips to 3 fps. On the other hand, the frame rates achieved with adaptation (with the same conditions) remained relatively stable around 9 fps (with average improvement being 39.6%).

Figure 3.15(c) shows the actual degradation ratios used with the projected target ratios. The prediction accuracy was high, with a median residual of 0.022%. The average degraded ratio was 22.7%, with a standard deviation of 0.063%. Considering that the JNDG is around 60% (Figure 3.8), there was still

Table 3.3: Rating scale used to compare videos with and without adaptation.

-3	-2	-1	0	+1	+2	+3
Much Worse	Worse	Slightly Worse	The Same	Slightly Better	Better	Much Better

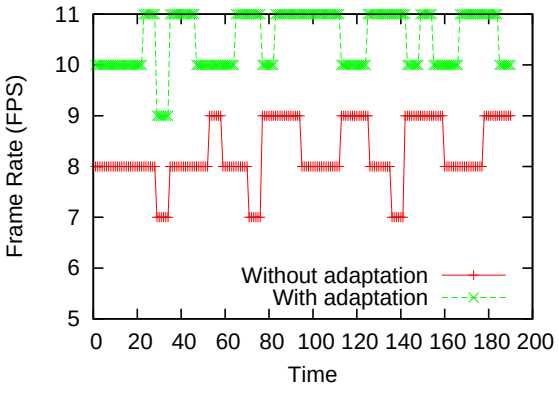
much room for more reduction if the frame rates were below desired thresholds.

We also conducted a user study to compare the visual quality of the recorded video. The crowdsourcing methodology was used due to the simplicity of the experiment. Following the ITU standard for double stimuli video comparison study [51], we made a short video with the following structure (in sequential order): (a) five seconds of text illustrating the purpose of the study, (b) two seconds of text indicating "Video 1" to be shown, (c) ten seconds of Video 1, (d) two seconds of text indicating "Video 2" to be shown, and (e) ten seconds of Video 2, and (f) ten seconds of text asking the rating question: "Compared to Video 1's quality, Video 2's quality is: [the scale shown in Table 3.3]?" [51]. The video was uploaded to Youtube and was advertised to a mailing list (containing graduate students, staff, and professors in Department of Computer Science). The ranking data were collected anonymously through an online Google Doc Form. A total of 81 responses were collected. Three of them were discarded because respondents notified the experimenter that they submitted by mistake. Figure 3.16 shows the collected ratings.

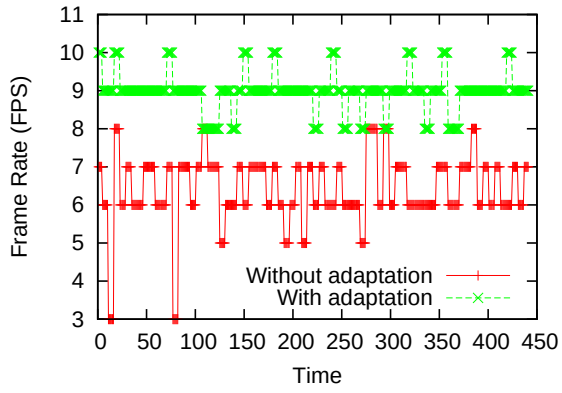
Among the 78 responses, 96.2% of the users thought the video with adaptation turned on had a better quality than the video with adaptation turned off, and 3.8% thought they were the same. 12.8% (of total) gave a "(+1) Slightly Better" ranking, 51.3% gave a "(+2) Better" ranking, and 32.1% gave a "(+3) Much Better" ranking. Clearly, our adaptation scheme not only saves system resource (i.e., CPU load), but also improves subjective video quality.

3.5 Conclusion

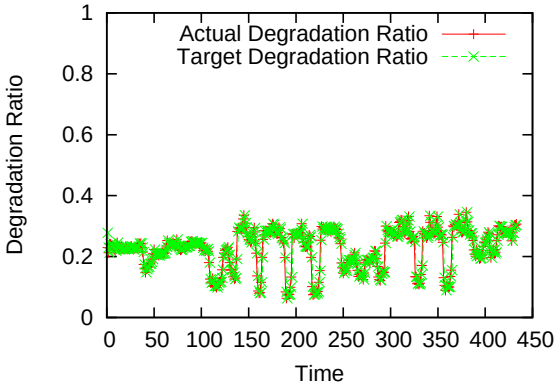
This chapter identifies a new critical quality factor, called Color-plus-Depth Level-of-Details (CZLoD). CZLoD characterizes the density and accuracy of depth in real-time color-plus-depth based 3DTI video. A psychophysical study of the perceptual thresholds of CZLoD is performed and presence of two perceptual thresholds - Just Noticeable Degradation (JNDG) and Just Unacceptable Degradation (JUADG) is demonstrated. Taking CZLoD as a guiding parameter, we design an online human-centric QoS adaptation scheme to dynamically adapt the video quality. Our experiments show that the adaptation scheme considerably reduces the temporal resource demands while enhancing the perceived visual quality (refer to the **temporal challenge** in Section 1.2). The user study shows that 96.2% of the users voted for the higher quality video



(a)



(b)



(c)

Figure 3.15: Adaptation performance: (a) the frame rates with and without adaptation where the CPU has no additional load other than tele-immersion, (b) the same frame rate comparison with a 16% CPU stress generated for both conditions, and (c) the degradation ratios are well below the just noticeable threshold, with high prediction rate regarding variance threshold.

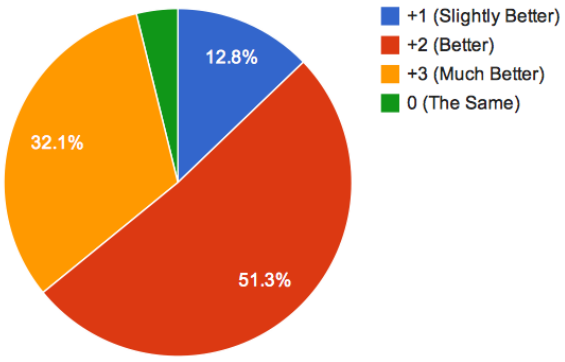


Figure 3.16: Subjective evaluation results comparing the quality of the adapted video against the unimpaired video. Sampled from 78 responses in a crowd-sourcing study.

produced through the real-time adaptation scheme over the unimpaired video.

Discussion. Subjective video quality research suffers from the limitation that the results might vary with the sequence content, and this study is no exception. While we attempt to be representative in choosing the 3DTI activities for the psychophysical study, we do not intend to draw any general conclusion about the specific values of JNDG and JUADG in all 3DTI applications. We also acknowledge that the acceptance thresholds very much depend on activities and users. Rather, the main contribution of our study is the identification of the existence of perceptual thresholds on a unique factor that has (to our best knowledge) never been explored in real-time color-plus-depth video. The measurements of the thresholds provide practical guidelines on their estimation in the field. We also demonstrate that by applying these thresholds to practice, we can adapt the detailing quality and achieve considerable resource saving as well as enhancement on the perceived video quality.

4 Inter-stream Data Adaptation

4.1 Introduction

In the previous chapter, we study intra-stream data adaptation, where imperceptible spatial details are excluded in each frame. There, the interactivity of the systems is greatly enhanced, because less time is needed to process the video data across the system pipeline (Figure 1.2). However, today's 3DTI systems still face significant challenges due to their high demand on spatial resources, as we have mentioned in Section 1.2. It is worth noting that the intra-stream data adaptation approaches described in the previous chapter partly target at improving frame rate, and thus the resultant bandwidth demand of each stream is not alleviated even with reduced frame size, as bandwidth/throughput is a product of frame rate and frame size.

In sought of further data reduction, we focus on the spatial challenge in multi-stream/multi-site 3DTI environments in this chapter. While the previous chapter considers the necessity and significance of data on the frame level (what level of spatial details is sufficient perceptually), this chapter considers the necessity and importance of data on the stream level (i.e., whether a stream deserves sending or not). In a word, this inter-stream adaptation scheme drops unimportant streams altogether for the sake of saving bandwidth, while the intra-stream adaptation scheme in the previous chapter drops unimportant pixels or details within the stream for the sake of improving interactivity. Since this inter-stream data adaptation scheme focuses on addressing the bandwidth challenge, it naturally fits into the data dissemination/transmission phase in the 3DTI pipeline (Figure 1.6).

More specifically, the huge demand of networking resources has restricted current 3DTI systems to work with only two sites. In fact, each 3D video stream can consume a large amount of bandwidth (as mentioned in Section 1.2), making even two sites of collaboration challenging enough. The problem is exacerbated if multiple sites are connected together, with each site producing tens of such large streams, which can easily exceed the bandwidth limit. Clearly, the "all-to-all" data distribution scheme, adopted by existing 2D video-conferencing and 3DTI systems ([12][30]), has to be abandoned as the scale of the 3DTI system grows. As a concrete example, the 3DTI system described by Yang *et al.* [116] involved two sites thousands of miles apart, each sending about ten streams to the other. With the measured resource limits, even three-site collaboration was

not possible if all streams from each site were sent to all other sites.

Some previous work has considered data reduction in a system-centric manner: these include background subtraction, resolution reduction, real-time 3D compression [64][70][112], and multi-stream adaptation [80][115]. However, as discussed in Section 1.2, those are primarily system-centric approaches that ignore human factors. In this chapter, we explore a human-centric methodology for tackling the spatial challenge. The key insight is that we can leverage the user *view* in the 3D cyber-space and only transmit those streams that contribute significantly to the view, so that the resources can be efficiently utilized while the visual quality is not noticeably affected.

But since we base our data selection and dissemination on user views, a major challenge is that we may have to constantly change the selected streams or dissemination topology whenever users change views. Before we describe any data protocols and algorithms, it is thus important to better understand what is *view* in 3DTI systems and how users actually control views. Interesting research questions include: (1) do users actually have any view preference or fixed viewpoint would suffice for them? (2) with 3D free views, do users change view at all? (2) do the view changes (if any) show unpredictable patterns or highly predictable patterns? Section 4.2 presents such an explorative study on view control in a 3DTI basetball-playing application. The key findings include: (1) users considerably prefer free view over fixed view, (2) users do change views in 3D free view mode, and the manipulation is mostly arbitrary and unpredictable across our users, meaning that view change is truly a challenge, (3) users tend to change views in a small-scaled, progressive manner, a phenomenon we refer to as *view locality*.

In light of these findings [110], we consider the problem of view-based inter-stream data adaptation. The basic idea is that for each user, a customizing subset of data streams that contributes most significantly to her view is selected and transferred. Such human-centric adaptation achieves resource saving without sacrificing the perceived visual quality. Specifically, the viewpoint of each user is automatically detected on the renderers, and the most contributing streams are determined based on a scalar product formula. Resource saving is thus achieved because the selected streams only constitute a small subset of streams.

While such view-based data selection implies considerable bandwidth reduction, the overlay dissemination topology among all gateways (Figure 1.4) for transmitting selected streams still remains a key challenge for multi-stream/multi-site 3DTI sessions. In Section 4.3, we first describe the system/network models and assumptions for this chapter. Then we detail the protocols for view-based inter-stream adaptation in Section 4.4. In Section 4.5, we formulate the stream dissemination problem in multi-stream multi-site 3DTI systems. We then focus on constructing an overlay topology with bounded end-to-end delay for the selected streams, and maximized utilization of available bandwidth at all gateway.

We evaluate a spectrum of heuristic algorithms for such overlay topology formation in the initialization phase of the systems (*static topology management*). In Section 4.6, we consider the practical challenges incurred by dynamic view changes possibly made by users in the run-time phase of the systems. We propose, compare, and evaluate three algorithms to handle the view dynamics in topology maintenance (*dynamic topology management*). With extensive experiments, we demonstrate that an algorithm that exploits view locality can achieve efficient bandwidth utilization, high topology stability, and great scalability.

4.2 Motivating Subjective Study

4.2.1 Overview

Tele-immersive communication is made possible by advanced video mediation that provides a joint virtual-reality-like environment for users to interact (Figure 1.1). Prior empirical literature has emphasized the strong value of such "shared visual context" in a range of remote collaborative tasks (e.g., [33][63]). However, it remains unclear how to present the visual context to best facilitate collaboration (e.g., in what views). As a result, the sense of presence is missing [69].

We are interested in exploring the *view* concept in 3DTI systems and the control preferences/patterns of users, because the understanding of these issues will offer valuable guidance on our design of view-based stream adaptation schemes (Section 4.5 and particularly 4.6). First and foremost, what is view? Actually, changing the view or perspective in a 3D space is equivalent to moving a "virtual camera" in the space. Figure 4.1 shows the model of the virtual camera to represent the view. It can be modeled by three vectors: *pos* is the position vector, *dir* is the capturing (or viewing) direction vector, and *up* is the upward direction vector.

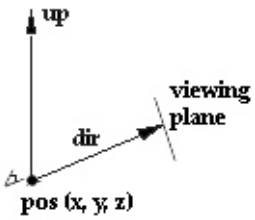


Figure 4.1: 3D (rendering) view modeled with a virtual camera in the cyberspace: *pos* is the position vector, *dir* is the capturing (or viewing) direction vector, and *up* is the upward direction vector.

As an overview of this section, we describe a study of three view conditions (single fixed view, multiple pre-defined views, and 3D free view). In the study, we invited users to participate in a remote basketball learning activity, and recorded/measured how users actually used the different view conditions and how they changed a free 3D view if it was available.

In this study, we aimed to understand human preferences on view in collaborative tasks, e.g., whether users preferred 3D free view or pre-defined views, and how users actually changed view. The research questions related to our inter-stream adaptation schemes include - (1) do we need to consider view change dynamics in our schemes? (2) If users change views, are there any predictable patterns or not? E.g., if view changes are mostly predictable, pre-fetching techniques can be applied to retrieve the streams associated with the new view. (3) Do users prefer free views or static views? If the latter, it would significantly simplify our stream selection because the possible views are known beforehand. (4) If users do change views, how do they change? Are the changes dramatic or progressive? Understanding of these questions will help us make more educated decisions in our algorithmic design.

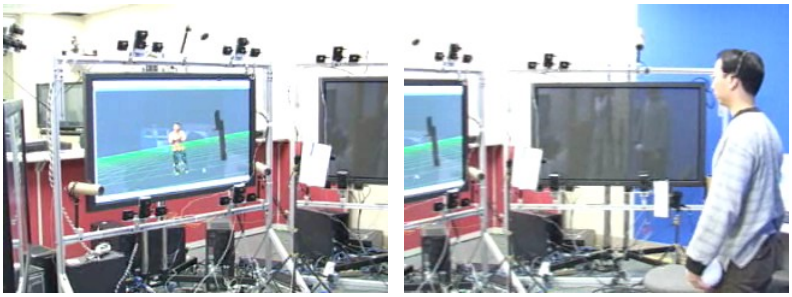
4.2.2 Participants, Procedures, and Apparatus

We used an existing 3DTI system to carry out the experiments. The system was set up in two remote laboratories in the Department of Computer Science at University of Illinois at Urbana-Champaign.

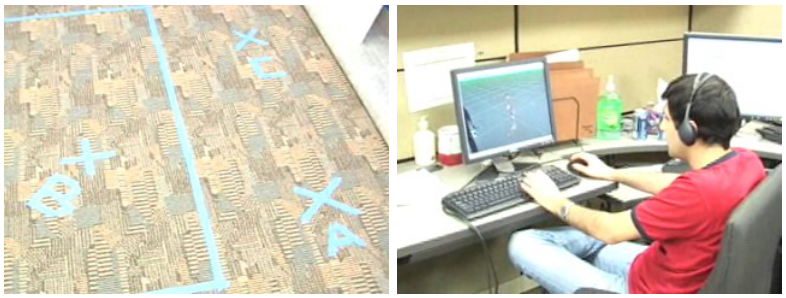
- *Participants:* Twelve participants were recruited from Department of Computer Science at University of Illinois at Urbana-Champaign in the United States to take part in the study, with an average age of 26. The participants were divided into six pairs. In each pair, the person with little experience of basketball ($\leq$ 1 year) was assigned to be the student, and the one with more experience ($\geq$ 3 years) became the coach. Both were given an opportunity to get comfortable with the environment/interface and talk with their partner in another room via a VoIP channel.

- *Student's Workspace:* Figure 4.2(a)-4.2(c) shows the student's work space which consisted of an array of cameras and a 61-inch NEC plasma display. Based on different camera conditions (which will be explained in more details in the next section), different cameras were used to capture the student in the scene (Figure 4.2(b)). The video streams were then aggregated by the local gateway and transmitted to the coach's site, and were also rendered locally on the large plasma display to show a mirrored view for the student, thereby providing the shared visual context.

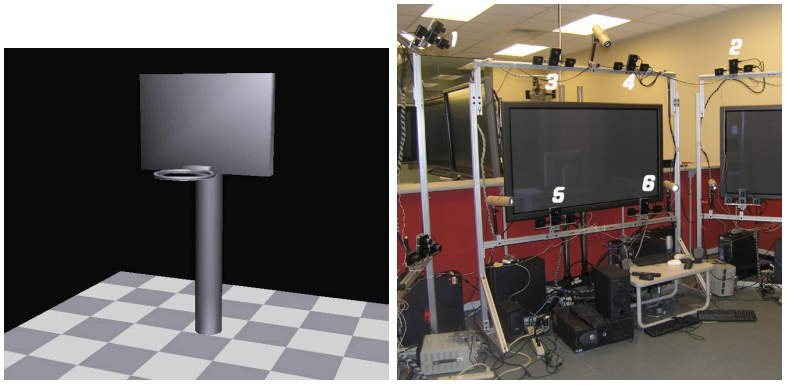
- *Coach's Workspace:* Figure 4.2(d) shows the coach's space with a renderer computer which received the video data from the student's site and rendered them on a 17" Dell desktop LCD display. The coach could see



(a) Student's space (b) Student



(c) Three shooting spots (d) Coach's space



(e) Virtual hoop (f) Camera setup in student's space

Figure 4.2: The student's space and the coach's space

the student's live video on the screen, and also change his/her viewing perspective of the cyber-space with a mouse and a keyboard.

- *Materials:* A virtual basketball hoop (Figure 4.2(e)) was rendered in the virtual world (which could be seen by both the coach and the student). Two toy plastic balls with a diameter of 6 inches were used by the student[1]. Both the coach and the student were wearing a wired headphone and microphone, and could talk via a VoIP connection.

- *Tasks.* The coach instructed the student (by audio) to learn basic basketball skills in two steps: (1) move the ball on top of the virtual basket (Figure 4.2(e)), and then drop it into the basket; (2) stand at three positions (i.e., A, B, and C, respectively, as marked on the laboratory floor, Figure 4.2(c)), and attempt to shoot the ball into the virtual basket. When shooting the ball, the coach corrected the student's pose by shaping his/her arms, hands, shoulders, knees, etc.

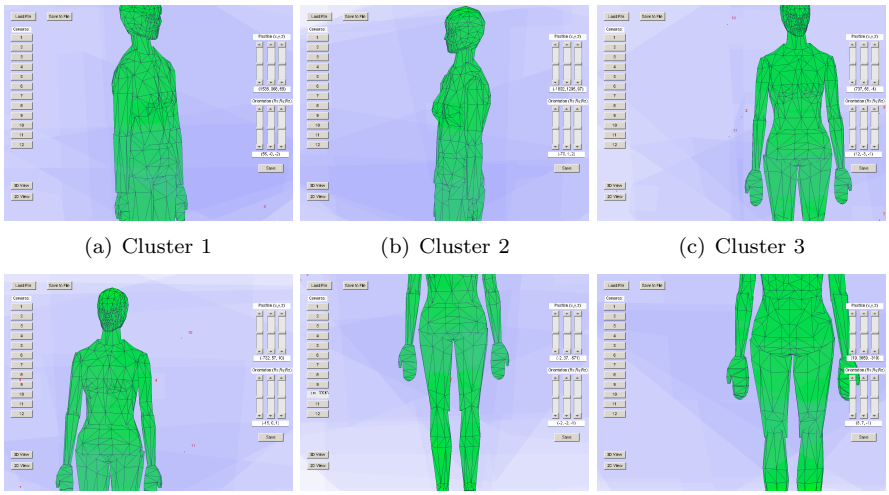
4.2.3 Camera Conditions

There are three general camera conditions in 3DTI systems: *single fixed view*, *multiple pre-defined view*, *3D free view*, respectively. The most simple view condition is *single fixed view*, where only a single camera is needed, thereby generating a fixed view for the remote users. Fixed view in general stimulates situation awareness by giving high level overview of the scene, but is constrained by the position and orientation of the camera. To address the problem of single fixed view, one can also use an interface of multiple pre-defined views that may increase situation awareness and reduce occlusion of objects with more perspectives. This is called *multiple pre-defined view* condition, but it is apparently still very limited. To overcome the constraints of fixed view and multiple pre-defined views, 3DTI systems can also offer a *3D free view* (Figure 1.3), where the users are allowed to manipulate the view of the cyber-space arbitrarily and hence observe the scene from almost anywhere.

Figure 4.2(f) provides a detailed view of the camera setup used in our experiments. Six camera clusters were involved in our study, Cluster 1-6 (marked in Figure 4.2(f)). Each cluster consisted of three black-and-white cameras (bottom) and one color camera (top) - refer to Figure 1.5. Figure 4.3 shows the simulated view of each camera cluster. Each camera condition was evaluated on each user. The order of the three sets was randomized to minimize the noise from learning effect.

- *Single fixed view:* The coach could see the scene from a fixed perspective because only one camera (Cluster 3) was used.

[1]The plastic balls were used to protect the equipment from being broken.



(a) Cluster 1 (b) Cluster 2 (c) Cluster 3

(d) Cluster 4 (e) Cluster 5 (f) Cluster 6

Figure 4.3: Simulated view of each camera cluster

- *Multiple pre-defined view:* Three cameras - one on the left of the student (Cluster 1), one on the center (Cluster 3), and one on the right (Cluster 2), but all in front of him/her- were used in this setting. The coach was thus able to switch among three corresponding viewpoints: `left`, `center`, and `right`, by clicking three option buttons on the renderer interface, respectively. The viewpoints in this scenario were physical for they were generated naturally from the cameras.

- *3D free view:* Six 3D camera clusters were used to reconstruct the 3D model of the scene. The coach could then drag the mouse freely in the renderer window to observe the scene from any viewpoint. The viewpoints in this scenario were *virtual* because they were simulated by reconstructing a 3D model from multiple 2D video streams.

Since the student users could not conveniently change view while performing activities, their view was unchanged across different configurations, which came from Cluster 3, as a mirrored view.

4.2.4 Human Study Results

We present the experimental results in this section along with implications for our work in Section 4.5 and Section 4.6[2]. At the outset of the study, we were interested to see how the coaching users controlled the view in different configurations and whether there were certain patterns when they switched the view. Results of this section came from numerical analysis of logged data for both multiple pre-defined views and 3D free view configurations.

[2]The first pair of participants were not able to complete the task, and thus were withdrawn from the data set.

Comparison

For single fixed view, all six coaches thought having the view fixed hindered the collaboration with the students with a mean score of 4.6 out of 5 (very much agree). Comparing the multiple pre-defined views with the 3D free view, 100% of the coaches thought being able to change the view freely helped the collaboration with a mean score of 4.8 out of 5.

The results indicate that the availability of 3D free view was clearly favored over the more constrained view conditions. View changes were needed, and almost inevitable. Clearly, we should carefully take them into consideration when designing our inter-stream adaptation schemes based on user views.

3D View Control Pattern

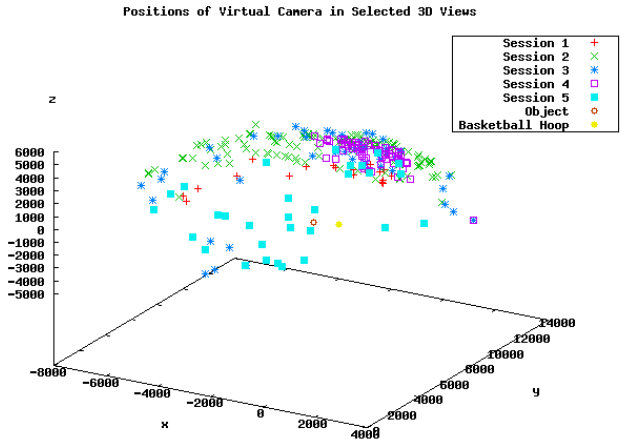


Figure 4.4: Positions of virtual camera in 3D view

Figure 4.4 shows the view (*pos* in Figure 4.1) selected by different coaches in the 3D free view mode. The *up* vectors of all virtual cameras were assumed to be vertically upward. The *dir* vectors were not drawn because the readers can imagine they were pointing from the positions of the virtual cameras to the student in the scene. Notice that the circle labeled "object" in Figure 4.4 indicates the position of the student in the 3D space.

First, one can clearly see that all users changed views a lot, meaning that the dynamics of changing view is a practical challenge we should consider in view-based stream adaptation. In Figure 4.4, there was a "clustered" area of selected view positions on the upper right part of the 3D space. Those positions were roughly 45-degree above the observed objects (i.e., the student and the basketball hoop). This was partly due to the nature of the tasks in our study, because that angle was ideal for the coach to check if the ball went into the basket.

Second, we observe that different users (coaches) showed very different preferences on view control and there did not seem to have a predictable pattern. For example, the coach in Session 5 was the only one who spent the majority of the time observing from the bottom of the horizon (with $z \leq 0$) in the 3D scene, and the coach in Session 2 appeared to favor the left half of the upper hemisphere (with $x \leq 0$) while others did not. This indicates that we should assume unpredictability of view change and handle the high level of dynamics.

Third, we also note that a large portion of view movement occurred within small distances. That is, view changes were made in a progressive manner. We refer to this phenomenon as *view locality*. We will explain in Section 4.6 how this can be leveraged for our algorithm design.

4.3 Models and Assumptions for View-based Adaptation

In the next two sections, we describe mechanisms for reducing and adapting streams based on user views. Before we jump into the algorithmic details, we first present the system/network models and assumptions as the foundation for our discussion later.

- ***Application Model.*** We consider the typical multi-stream/multi-site applications in a 3DTI system, where a group of remote users collaborate in a 3D virtual space in real time. The collaboration requires everybody to see everybody else in the virtual world. We define the *cyberspace* as the 3D virtual space that contains all the active remote users in tele-immersive sites in the 3DTI system. There might be also viewing sites, where passive remote users only join to watch the collaboration.

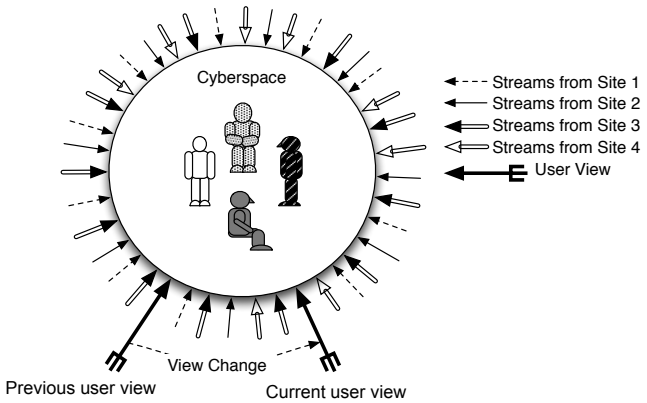


Figure 4.5: User view, view change, and camera views.

- ***View Model.*** We depict the view model in terms of a virtual camera in Section 4.2 (Figure 4.1). To simplify our discussion, we denote $\vec{V_i^j}$

(equivalent to the *dir* vector in Figure 4.1) as a rendering viewpoint in the cyberspace selected by the users at site i on the jth renderer, R_i^j, which is mathematically a vector in the 3D coordinate world and can change dynamically. We assume the users at different sites can change views simultaneously. Further, we assume a view change made by the users can be continuous or discrete, but must be *finite*, that is, it must have a starting point and an ending point in a finite amount of time (e.g., Figure 4.5).

- ***Overlay Network Model.*** $\mathbb{G} = \{\mathsf{G}_1, \mathsf{G}_2, ..., \mathsf{G}_N\}$ are the gateway nodes in all sites, where node G_i is in a site i (refer to Figure 1.4). The gateways in tele-immersive sites are $\{\mathsf{G}_1, \mathsf{G}_2, ..., \mathsf{G}_M\}$ $(M \leq N)$, and the gateways in viewing sites are $\{\mathsf{G}_{M+1}, ..., \mathsf{G}_N\}$ (if $M \neq N$). G_i $(1 \leq i \leq M)$ has a set of *local streams* produced by its local 3D cameras, denoted by $\mathbb{S}_i^{local}$. Managed by a session controller for membership and topology information, the gateway nodes form an application-layer overlay for data dissemination. We envision a middleware functionality in the session controller (Figure 1.4) to maintain an efficient overlay structure using one of the schemes described in Section 4.5.2 and 4.6.2. We assume the participating nodes are stable and trustworthy.

- ***Stream Model.*** Stream s_i^p is produced in a tele-immersive site i $(1 \leq i \leq M)$, with p being the camera index within the site $(1 \leq p \leq |\mathbb{S}_i^{local}|)$. A stream in our context is a 3D video stream with each frame containing both color and depth information. With the camera calibration parameters, we can map the capturing directions of the physical cameras from all sites into the 3D virtual space. Each stream thus has a three-dimensional capturing vector, i.e., *camera view*, within this global space (Figure 4.5). We define the *view* of a stream, $\vec{C}_\mathsf{s}$, to be the normal vector of the imaging plane of the camera that produces the stream s, which can be obtained by the calibration parameters acquired via the initialization phase. The camera views are determined by the physical placement of the 3D cameras, and thus are static. The streams from one site are semantically correlated, because the cameras are shooting the same scene, only from different angles. Streams are differentiated by their camera views, $\vec{C}_\mathsf{s}$ (Figure 4.5). We assume the streams are coded and transmitted independently using TCP as the transport layer protocol.

- ***Stream-View Contribution.*** It is clear that for a given rendering view or user view, camera streams are differentiated by their *semantic importance* to the view. For example, for a user view showing the front of a person, camera streams that capture his/her back are not important at all. Such semantic importance can be quantified by the *contribution factor* [113], $CF_\mathsf{s}^{\vec{V}_i^j}$, which is the scalar product of the two vectors:

$CF_{\mathbf{s}}^{\vec{V}_i^j} = \vec{V}_i^j \cdot \vec{C}_{\mathbf{s}}$.

- ***Publish-Subscribe Model.*** We use the general publish-subscribe model as a simple yet powerful distributed paradigm to represent view-based stream selection. The general publish-subscribe communication paradigm consists of three components: *publishers*, *subscribers*, and a *mediating infrastructure, i.e., rendezvous points*. The subscribers express interest in the data advertised by the publishers to the gateways. The publishers, unaware of who subscribe to what, simply deliver the data to the gateways. The rendezvous points then match the subscribers' interests with the data produced by publishers, and deliver the matching content to the subscribers. In our 3DTI systems (Figure 1.4), cameras become the publishers, the displays/renderers become the subscribers, and gateways and session controller become the rendezvous points. Basically, displays "subscribe" to a select set of camera streams that are important to the user views rendered. Such subscription information is collected first by the local gateway and then session controller for "matching". Matching refers to the process of identifying which particular gateway node is the best candidate to deliver the stream to the requesting gateway and display.

4.4 View-based Inter-stream Adaptation Protocols

As stressed in Section 1.2 and 4.1, the major goal of inter-stream adaptation is to reduce/adapt video streams based on their semantic importance in terms of contribution to user views. The basic idea is that by utilizing limited bandwidth resources to serve the most important streams, we can reduce considerable data in dissemination in order to address the spatial challenge (Section 1.2). More specifically, we leverage the user view in the cyber-space such that only a subset of streams that are contributing to the view are transmitted across the Internet. The major benefit is that we can reduce the amount of required bandwidth without sacrificing much visual quality for the user. This is because the data that are not subscribed/delivered do not contribute to the user's view, thereby not noticeably affecting the visual quality.

Conceptually, there are two main steps in such adaptation schemes for each user-selected view - *(1) stream selection*, and then *(2) parent selection* for each stream. Stream selection refers to the process of the local gateway (Figure 1.4) selecting the subset of important streams for the user view, and parent selection refers to the process of the session controller (Figure 1.4) selecting, for each important stream, a gateway node as the parent to deliver the stream to the requesting gateway. There are two major phases when this two-step process occurs: session initialization, and session run-time. In the session initialization

phase, stream selection first occurs in *every* joining gateway, and all the subscription information (i.e., which streams are important to which gateways) is collected by the session controller, and a globally optimal topology is then determined during the parent selection step (this part is described in Section 4.5). After that, the system enters the session run-time phase, where any new view change made by any user triggers the two-step process dynamically (this part is described in Section 4.6).

Schematically, when a new view $\vec{V}_i^j$ is selected on a renderer R_i^j at site i, R_i^j notifies the local gateway G_i, which then selects a subset of streams (that originate from other sites), denoted as $\mathbb{SS}_{\vec{V}_i^j} \subset \mathbb{S}$ that are important to serve the new view. This is done by computing the contribution factor $CF_{\mathsf{s}}^{\vec{V}_i^j}, \forall \mathsf{s} \in \mathbb{S}$, and selecting *from each tele-immersive site* the top K streams with the highest contributing factors. Large K is good for achieving higher visual quality, but may increase the rejection ratio due to the limited bandwidth in the system. Smaller K reduces rejection ratio, but may affect visual quality for the user. We find that setting K to $|\mathbb{S}_j^{local}|/2$ (where $|\mathbb{S}_j^{local}|$ is the total number of streams produced from G_j, $1 \leq j \leq M$) achieves fair balance between the two factors. Alternatively, thresholding can also be used, that is, select streams with $CF(\vec{V}_i^j, \mathsf{s}) \geq TH_{contribution}$. The above procedure is referred to as **stream selection** phase. G_i then sends *subscription requests* to the session controller, specifying the streams it request to receive. For each selected stream $\mathsf{s} \in \mathbb{SS}_{\vec{V}_i^j}$, session controller searches for a "good" parent (gateway) node G_p in the system from which G_i should receive the stream from. This process is referred to as the **parent selection** phase. Session controller then sends replies back to G_i specifying which G_p is for each requested stream (if available). Finally, G_i sends a request to G_p, and makes connection to receive the stream from it. We present more details of the protocol below.

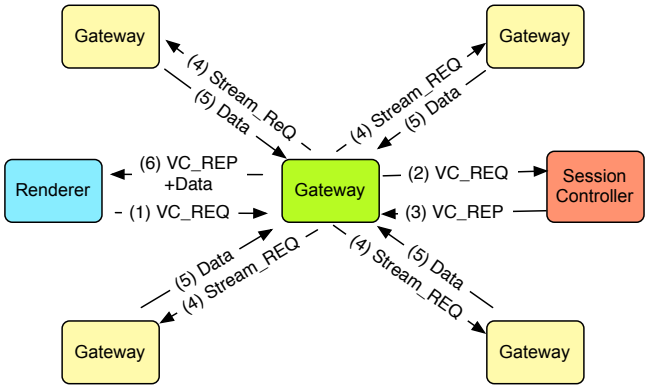


Figure 4.6: Publish-subscribe protocol in data dissemination.

Figure 4.6 illustrates the protocols from a display/renderer requesting a view to it retrieving the contributing streams. Every view change made by the user

triggers a view request. It is assumed that before a session starts, all gateways register with the session controller (Figure 1.4), and report the information (e.g., calibration parameters) of their local streams (if any). The session controller collects the stream meta-data, assigns the global stream identifiers ($streamID$), and then broadcasts them back to all participating gateways. Below we describe the basic steps in the protocol:

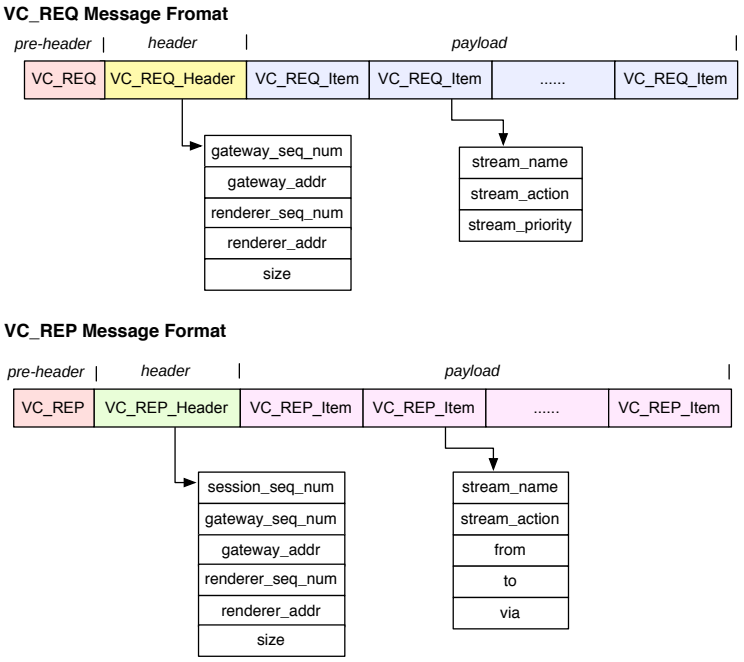


Figure 4.7: View change request/reply formats

1. **Requesting Renderer (R_i) $\xrightarrow{VC\_REQ}$ Local Gateway (G_i).** On detecting a view change made by the user, the renderer sends a VC_REQ (view change request) message to the local gateway. Figure 4.7 shows the format of this request message. It has a small pre-header specifying the message type (i.e., VC_REQ), followed by a header (i.e., VC_REQ_Header), and a payload (a list of view change request items, VC_REQ_Item's, each for a stream from another site). The message header, VC_REQ_Header, has five fields as shown in Figure 4.7. When sending the view change request message, the renderer fills in the latter three: (c) $renderer\_seq\_num$ - a monotonically increasing sequence number the renderer uses to keep track of its requests, (d) $renderer\_addr$ - its own IP address, and (e) $size$ - the size of the message payload being sent. The message is filled up by the renderer, and sent to the local gateway.

2. **Local Gateway (G_i) $\xrightarrow{VC\_REQ}$ Session Controller.** On receiving the VC_REQ message from a requesting renderer (R_i), the local gateway fills in the first two entries in the message header: (a) $gateway\_seq\_num$ - a

monotonically increasing sequence number the gateway uses to keep track of its requests, and (b) *gateway_addr* - the IP address of itself. The message payload consists of a list of view change request items (VC_REQ_Item), with each item representing a request for one 3D video stream. The gateway performs stream selection as described above, fills up the *VC_REQ_Items* fields, and then forwards this request message to the session controller.

3. **Session Controller $\xrightarrow{VC\_REP}$ Local Gateway (G_i).** On receiving the VC_REQ message from the requesting gateway G_i, the session controller performs parent selection, and sends a VC_REP (view change reply) message back. Figure 4.7 shows its format, which is similar to VC_REQ, with a small pre-header specifying the message type (in this case, VC_REP), followed by a header (i.e., VC_REP_Header), and finally a payload (a list of view change reply items, VC_REP_Items). The message header, VC_REP_Header, has six fields, with five same ones copied from the received VC_REQ_Header, and an additional field, *session_seq_num*, which is a monotonically increasing sequence number the session controller uses to keep track of its requests. The message payload consists of a number of view change reply items (VC_REP_Item), with each item representing a reply for one requested stream. The session controller generates these reply items by examining the current topology and network/system dynamics, and selecting a parent node, if possible, to serve the requesting gateway G_i with each requested stream in VC_REQ_Item. The details of the algorithm (e.g., how the parent node is selected) is deferred to the next sub-section "Overlay Construction". Each reply item has six fields as shown in Figure 4.7. The *stream_name* is copied from the *name* field in the corresponding view change request item, specifying the stream name. The *stream_action* field is marked (1) either *Add_Stream* if the requested stream originates from the local site (i.e., the local gateway can serve the renderer), or (2) *Relay_Stream* if the requested stream originates from a remote site, or (3) *Drop_Stream* in case of preemption (Section 4.6.2). The $from$ field specifies the IP address of the source gateway of the stream (i.e., the local gateway where the stream originates). The *to* field specifies the IP address of the requesting renderer (R_i). The *via* address specifies the IP address of the "parent" gateway node the session controller finds to serve the stream.The message is filled up by the session controller, and sent to the requesting gateway (G_i). Note that the parent selection step is done immediately at run-time, but in session initialization it is held off until all view requests have been received.

4. **Local Gateway (G_i) $\xrightarrow{Stream\_REQ}$ Remote Gateways ($\mathsf{G}_{p1}, \mathsf{G}_{p2}, ..., \mathsf{G}_{pm}$).** The requesting gateway examines the VC_REP message, and then sends a $Stream\_REQ$ message to the kth "parent" gateway (G_{pk}) specified in each $VC\_REP\_Item_k$ (refer to Figure 4.6). The $Stream\_REQ$ message

contains the stream name and the IP address of its requesting gateway.

5. **Remote Gateway (G_p) $\xrightarrow{Data}$ Local Gateway (G_i).** The remote gateway (G_p) then resolves *Stream_REQ* by sending the requested stream to G_i.

6. **Local Gateway (G_i) $\xrightarrow{Data}$ Requesting Renderer (R_i).** Receiving the stream from the relay gateway G_p, the local gateway distributes this stream to the requesting renderer R_i, together with a view change reply message (*VC_REP*). To this point, the view change request is resolved.

As described in Step 3 above, session controller is responsible for identifying a good parent node for each stream request. The session controller maintains three types of information: (a) membership, (b) network and system dynamics, and (c) topology. For membership, it acquires a list of gateway nodes, and a list of participating camera streams via the session initialization protocol. It also keeps track of the network and system dynamics such as point-to-point latency and end-to-end available bandwidth between pairs of gateway nodes. Such dynamic information is needed for the overlay construction. Most importantly, the session controller maintains the overlay topology, i.e., how the gateway are connected on the application-level overlay for stream dissemination [108][113]. The next two sections present the details of algorithms that are executed on the session controller for such topology management in the static (session initialization) and dynamic (session run-time) phases, respectively.

4.5 Static Topology Management

Even with the view-based publish-subscribe data selection mechanism, we still find the overlay construction (particularly parent selection) in the initialization phase of a collaborative session among all gateways to be a key challenge (NP-complete). We refer to this as the static topology management problem. In this section, we mathematically formulate this problem and tackle it with several multicast tree-based heuristic algorithms and a randomized algorithm.

4.5.1 Problem

It is assumed in the session initialization, the subscription (view change) requests are globally collected at the session controller before the overlay topology is computed. That is, Step 3 described in Section 4.4 is deferred until all requests are received. Given all the subscription requests, the main goal of session controller is to organize an overlay structure to disseminate the streams among all gateways as requested. In the multi-stream/multi-site 3DTI environments, the overlay graph we are to construct is essentially a *forest* of multiple trees, with each tree designated to disseminate a stream among the set of requesting gateways. We define the *multicast group*, $\mathbb{G}(\mathsf{s}) \subseteq \mathbb{G}$, as the set of gateway nodes

that have requested (i.e., one of its displays subscribed to) the stream s. We exclude the edge hosts (i.e., the cameras, the displays) from the overlay structure for the sake of simplicity. We use the terms nodes and gateways interchangeably.

For each multicast group $\mathbb{G}(\mathsf{s})$, a multicast tree $\mathbb{T}_\mathsf{s}$ needs to be constructed to disseminate the stream s from the source to all other nodes. Note that each tree $\mathbb{T}_\mathsf{s}$ only includes the gateways in $\mathbb{G}(\mathsf{s})$. The gateways that are not requesting the stream s are not used in $\mathbb{T}_s$ because they are already highly loaded with their streams. In reality, each site often has tens of streams to disseminate, and hence it acts as a source for that many trees. The construction of such a forest is complicated by several characteristics of multi-site 3DTI environments: (1) multiple system constraints: each site has inbound and outbound bandwidth limits, and the end-to-end delay between any pair of nodes has to be small in order to guarantee interactivity; (2) a dense graph: since the participant typically wants to see the other participants from a wide field of view, the overlay graph consisting of all gateways often has very high density (i.e., the average in/out-degrees of all nodes are large); hence, the construction of the forest needs to be carefully coordinated because the bandwidth resources are shared among all trees.

Due to the huge demands of computing and networking resources in multi-site 3DTI collaboration, we have two constraints and one optimization goal to satisfy in the overlay construction problem.

- *Constraint I (bandwidth)*: Each node has inbound I_i and outbound O_i bandwidth limits in the unit of number of streams (i.e., $I_i, O_i \in \mathbf{N}$), which can be dynamically measured by existing probing tools like Pathload [55]. A node should never receive data more than its inbound bandwidth limit (i.e., $d_{in}(\mathsf{G}_i) \leq I_i$, where $d_{in}(\mathsf{G}_i)$ is the actual in-degree of node G_i in the overlay), nor be delegated to send data more than its outbound bandwidth constraint (i.e., $d_{out}(\mathsf{G}_i) \leq O_i$, where $d_{out}(\mathsf{G}_i)$ is the actual out-degree of G_i).

- *Costraint II (latency)*: In 3DTI, remote participants are rendered into the cyber-space in real time for interactive collaboration. Therefore, the expected end-to-end latency or cost between from any source to destination node on any overlay path, $cost(\mathsf{G}_i \Rightarrow \mathsf{G}_j)_\mathsf{s}$ (for $1 \leq i, j \leq N$ and $i \neq j$, for any s that is subscribed), should not exceed a bound[3], B_{cost}, in order to guarantee interactivity.

- *Optimization Goal (request rejection ratio)*: Due to the two stringent constraints listed above, we cannot guarantee that all subscription requests are satisfied. The metric we wish to minimize is the total rejection ratio of all requests in the system, denoted by X. Suppose the number of subscription requests made by node G_i to G_j is $u_{i\rightarrow j}$ (i.e., $u_{i\rightarrow j}$ number of

[3]As it is impossible to guarantee hard real-time bound in asynchronous network, we only attempt to satisfy an upper bound on expected latency from the source to the destinations.

streams originating from site j are subscribed by at least one display at site i), among which $\hat{u}_{i\to j}$ are rejected, we thus have

$$X = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \frac{\hat{u}_{i\to j}}{u_{i\to j}} \tag{4.1}$$

More specifically, the *forest construction problem* can be formulated as follows.

Forest Construction Problem. Given (1) a completely connected graph $(\mathbb{G}, \mathbb{L})$ consisting of all gateway nodes $\mathbb{G}$ and the network links among them $\mathbb{L}$, (2) an in-degree bound $I_i \in \mathbf{N}$, and an out-degree bound $O_i \in \mathbf{N}$, $\forall \mathsf{G}_i \in \mathbb{G}$, (3) a cost denoted as $cost(\mathsf{G}_i \Rightarrow \mathsf{G}_j) \in \mathbf{Z}^+$ for each edge in $\mathbb{L}$, which denotes the latency between pair of gateway nodes, and (4) a set of multicast groups $\mathbb{G}_{multicast} = \{\mathbb{G}(\mathsf{s}) \mid \mathbb{G}(\mathsf{s}) \in \mathbb{G}, \forall \mathsf{s} \in \mathbb{S}_{subscribed}\}$, where $\mathbb{S}_{subscribed}$ is the set of streams that are subscribed by at least one renderer, the goal (of session controller) is thus to find a *spanning forest*, $\mathbb{F} = \{\mathbb{T}(\mathsf{s}) \mid \forall \mathsf{s} \in \mathbb{S}_{subscribed}\}$, with each tree $\mathbb{T}(\mathsf{s})$ being a spanning tree that connects the source node of s, G^{s}, to a subset of the other nodes $\mathbb{G}(\mathsf{s})' \subseteq \mathbb{G}(\mathsf{s}) - \mathsf{G}^{\mathsf{s}}$ in order to deliver stream s, such that the total fraction of excluded nodes, $\sum(|\mathbb{G}(\mathsf{s}) - \mathbb{G}'(\mathsf{s})|)/|\mathbb{G}(\mathsf{s})|$ $\forall \mathsf{s} \in \mathbb{S}_{subscribed}$, is minimized, subject to the constraint that $\forall \mathsf{G} \in \mathbb{T}(\mathsf{s})$, $d_{in}(\mathsf{G}) < I_i$ and $d_{out}(\mathsf{G}) < O_i$, and the total cost from the source node to each destination node for any stream is bounded, i.e., $cost(\mathbb{P}(\mathsf{G}^{\mathsf{s}} \to \mathsf{G})_{\mathbb{T}(\mathsf{s})}) < B_{cost}$, where $\mathbb{P}(\mathsf{G}^{\mathsf{s}} \to \mathsf{G})_{\mathbb{T}(\mathsf{s})}$ represents the overlay path from G^{s} to G in tree $\mathbb{T}(\mathsf{s})$.

Wang *et al.* [104] proved that the problem of finding a solution subject to two or more constraints in any combination in the multicast routing problem is NP-complete. We study several heuristic algorithms to address the problem.

4.5.2 Heuristic Algorithms

We will discuss three tree-based algorithms and a randomized algorithm that are executed in the session controller. In all cases, the trees in the multicast forest are constructed incrementally, that is, within a multicast group $\mathsf{G}(\mathsf{s})$, all requests for the stream s are processed sequentially in a randomized order (basic node join algorithm). The order in which trees are constructed affects the overall optimization goal, due to the inter-dependencies among the trees. The inter-dependencies are caused by the shared limited resources of the nodes that are present in multiple trees. We describe three tree-based algorithms - LTF, STF, and MCTF, and a simple randomized algorithm.

Basic Node Join Algorithm

We formally define the *subscription request* as $\mathsf{req}_i(\mathsf{s}_j^q)$ (refer to *VC_REQ* in Figure 4.6), specifying that G_i requests to receive stream s_j^q which originates from site j with index q. The basic node join algorithm, running at session

controller, is used to process a request $\mathsf{req}_i(\mathsf{s}_j^q)$, i.e., joining the node G_i into the existing tree $\mathbb{T}(\mathsf{s}_j^q)$. Since the 3DTI session involves a dense graph, we desire *load balancing* among all nodes such that no one would be particularly overloaded. The basic idea is thus to find a "close-by" node (for the concern of latency) with the maximum available bandwidth left in the existing tree, to serve as the parent to the requesting node. Several metrics can be used to find the "close-by" node, such as network coordinates, round-trip time measurement, and geographical distances, and we choose to use the last one in our experiments.

Before attempting to join the node G_i into the tree $\mathbb{T}(\mathsf{s}_j^q)$, the algorithm first checks the in-degree of G_i. If $d_{in}(\mathsf{G}_i) < I_i$, it proceeds to the next step. Otherwise, it rejects the request because the inbound bandwidth limit is saturated. After passing the inbound check, the algorithm looks for a parent node G_p in the existing tree $\mathbb{T}(\mathsf{s}_j^q)$ with available out-degree and the maximum *remaining forwarding capacity* (*rfc*) among all nodes in $\mathbb{T}(\mathsf{s}_j^q)$, subject to the latency constraint that the cost from G_i to the source of $\mathbb{T}(\mathsf{s}_j^q)$ (i.e., G_j) would be smaller than a real-time bound, if G_i were connected to G_p.

The rfc_i of node G_i denotes the available portion of out-degree that can be used for forwarding streams. It is computed as $rfc_i = O_i - d_{out}(\mathsf{G}_i) - \hat{m}_i$, where $\hat{m}_i$ denotes the number of streams that (1) originate from node G_i, (2) are subscribed by at least one other gateways, but (3) have not yet been disseminated out to any other node in the existing forest. This reservation mechanism ensures that we minimize the probability that a whole tree cannot be constructed because the source node is saturated. If no such eligible G_p can be found in $\mathbb{T}(\mathsf{s}_j^q)$, the request $\mathsf{req}_i(\mathsf{s}_j^q)$ is rejected. In this case, the tree is said to be *saturated*. The pseudo code of the algorithm can be found in the authors' technical report [109].

Figure 4.8(a) is an example where only one tree is shown for simplicity. F is the new node to join the existing tree of six nodes, $\{A, B, C, D, E, S\}$, where S is the root. Among the nodes, E has no out-degree left to serve F (i.e., $rfc = 0$), in which 4 is reserved for its out-streams ($\hat{m}_i$) and 4 is already taken in other trees ($d_{in}(\mathsf{G}_i)$). D has the largest rfc (22-8-0=14), but has a cost $(8 + 3 + 3 = 14)$ exceeding the upper cost bound 10. A has the second largest rfc (15-5-3=7), and has a cost (4+5=9) smaller than the bound. Therefore, A becomes the parent to serve F. Again, this basic node join algorithm seeks to achieve load balancing, which is essential in such a dense graph as a multi-site 3DTI session.

Tree-based Algorithms

We now describe (and compare) three tree-based algorithms which differ in the order of processing subscription requests and hence tree construction for each stream. For each algorithm, the basic node join algorithm is used to process a single request.

- *Largest Tree First (LTF) Algorithm.* The intuition is to construct the largest tree (largest in terms of number of nodes subscribed to a particular stream, i.e., $|\mathbb{G}(\mathsf{s})|$) first so that even if the last few trees cannot be constructed due to saturation, the rejection ratio should be small because we are left with the smallest trees. Specifically, we first sort all multicast groups based on the size, and then construct the spanning trees one by one from the largest multicast group to the smallest one.

- *Smallest Tree First (STF) Algorithm.* As a comparison to LTF, we also study the reversed algorithm which starts from the smallest multicast group, and ends with the largest one. Our hypothesis is that the rejection ratio of LTF should be smaller than that of STF.

- *Minimum Capacity Tree First (MCTF) Algorithm.* This algorithm considers the difficulty of tree construction in terms of the *forwarding capacity* of a tree. The intuition is that the larger this value is, the easier it is to construct the tree. That is because new requests are easier to accommodate with a tree containing large aggregate forwarding capacity. A node G_i's forwarding capacity is $O_i - m_i$, where m_i is the number of streams G_i has to send out (i.e., the number of streams that originate from G_i and are subscribed by at least one other gateway). The forwarding capacity of a tree $\mathbb{T}(\mathsf{s})$ is the sum of the forwarding capacity of all nodes in the multicast group $\mathbb{G}(\mathsf{s})$. This algorithm sorts all multicast groups in the ascending order based on the aggregate forwarding capacity, and starts from the multicast group with the least aggregate forwarding capacity, to the one with the largest.

- *Randomized Algorithm (RJ).* LTF, STF, and MCTF all seek to build the trees one by one, that is, only when it finishes processing *all* subscription requests in one tree will it move on to construct the next one. In contrast, we propose a randomized algorithm, called "Random Join" (RJ), which randomizes all subscription requests for the whole forest, with no prioritization on any tree. Again, the basic node join algorithm is used to process each request.

Somewhat surprisingly, our simulation in Section 5 finds that RJ generally outperforms the other tree-based algorithms. One of the reasons that the randomized algorithm works better is that every node in multi-site 3DTI collaboration is likely to be overloaded with subscription requests, because a participant typically wants to see a large portion of other participants from a wide field of view. In tree-based algorithms, a node is much more likely to be congested in the first few constructed trees if it is the source, or a node near the source. This increases the probability of rejection in the construction of the latter trees because the node's total bandwidth is shared among different trees. In contrast,

the randomized algorithm achieves good load balancing because it distributes the tasks of request processing among different trees randomly.

In light of these results, we next propose further optimization to the basic RJ algorithm by exploiting the semantic correlation among streams.

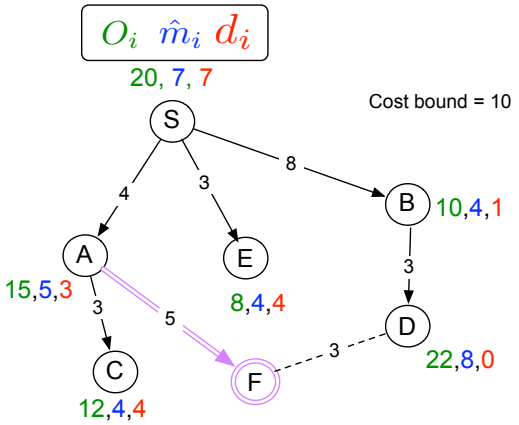
Exploiting Correlation

In 3DTI environments, the streams generated from one site have high semantic correlation among each other, because the cameras are often capturing the same scene, only from different angles. We exploit the inter-stream correlation to minimize the level of loss in times of saturation. As a motivating example, suppose a site A subscribes to four streams from site B (s_b^1, s_b^2, s_b^3, s_b^4) and one stream from site C (s_c^7). Then losing one stream from B is less critical than losing the single stream from C, since the former reduces the visual quality of a scene, while the latter loses a scene. Therefore, to minimize the level of loss for each participant, we *selectively* drop streams (i.e., reject requests) when the tree to join is saturated.

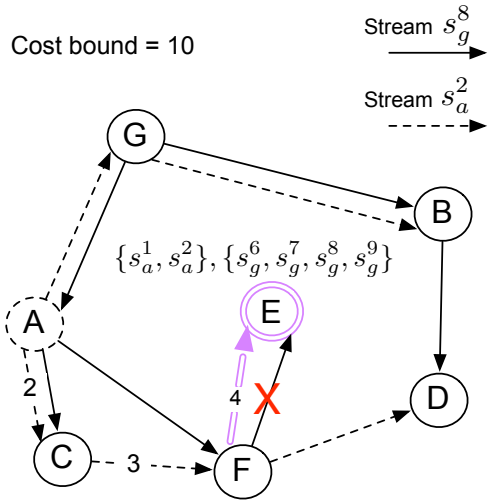
We describe a modified RJ algorithm, called CO-RJ, which exploits stream correlation. First, we introduce the concept of *criticality* for a node to lose a stream. Recall that $u_{i\to j}$ is the number of streams that node G_i subscribes from node G_j. The *criticality* for node G_i to lose a stream s_j originating from G_j is $Q_{i\to j} = \frac{1}{u_{i\to j}}$ for $1 \le i, j \le N$ and $i \ne j$. In the previous example, the criticality for node A to lose any stream from node B is thus $\frac{1}{4}$, and that to lose s_c^7 is 1.

In CO-RJ, whenever a request is rejected due to tree saturation, the algorithm looks for a victim request with a smaller criticality value than the current request. If such a victim can be found, CO-RJ rejects the victim request, and satisfies the current request. More specifically, when a request $\mathsf{req}_i(\mathsf{s}_j^p)$ (node G_i requesting stream s_j^p) is rejected due to tree saturation, the algorithm checks the trees that have been constructed on the following four conditions: (1) if there is a stream s_k^q ($k \ne j$) with $Q_{i\to k} < Q_{i\to j}$, and (2) G_i is a leaf node in tree $\mathbb{T}(\mathsf{s}_k^q)$ (or more simply $\mathbb{T}_k$), and (3) the parent of G_i in $\mathbb{T}_k$, node G_h, has already joined the tree $\mathbb{T}(\mathsf{s}_j^p)$ (or more simply $\mathbb{T}_j$), and (4) the cost between G_i and the source for stream s_j^p (i.e., G_j), if connecting G_i to G_h, is less than the real time bound (i.e., $cost(\mathsf{G}_j \Rightarrow \mathsf{G}_i)_{\mathbb{T}_j} < B_{cost}$). If the four conditions are all satisfied, CO-RJ removes the edge $\mathsf{G}_h \to \mathsf{G}_i$ in $\mathbb{T}_k$ and add a new edge $\mathsf{G}_h \to \mathsf{G}_i$ in tree $\mathbb{T}_j$. In other words, G_i loses s_k^q instead of s_j^p. This operation is done with minimal cost, as G_i was a leaf node in tree $\mathbb{T}_k$, hence removing the old link would not cause relocation of any other nodes in $\mathbb{T}_k$.

Figure 4.8(b) is an example showing two trees rooted at node A (for stream s_a^2) and G (for stream s_g^8), respectively. The label on the edge denotes the latency between the two nodes. E has joined the tree for stream s_g^8 but wishes to receive stream s_a^2 too. E's subscription contains two streams from site A



(a) Basic node join algorithm



(b) CO-RJ algorithm

Figure 4.8: Examples of algorithms

(s_a^1, s_a^2), and four streams from site G ($\mathsf{s}_g^6, s_g^7, s_g^8, s_g^9$). Therefore, the criticality for E to lose a stream from A is $\frac{1}{2}$, and that from G is $\frac{1}{4}$, i.e., $Q_{E\to G} < Q_{E\to A}$. Assume the tree of s_a^2 is saturated, i.e., no eligible node can be found to serve E based on the bandwidth and delay constraints (Section 4.5.1). We have (1) $Q_{E\to G} < Q_{E\to A}$, (2) E is a leaf node in the original tree of s_g^8, (3) node F, which is the parent of E in tree s_g^8, actually has the stream s_a^2, and (4) if connecting E to F in the tree of s_a^2, the cost (2+3+4=9) would be smaller than the bound. Since all four conditions are satisfied, CO-RJ will remove the link $F \to E$ in the tree of s_g^8, and add the link $F \to E$ in the tree of s_a^2 as shown in Figure 4.8(b). In other words, F serves E with the new stream s_a^2 instead of s_g^8 although F itself is saturated.

4.5.3 Performance Evaluation

Simulation Setup

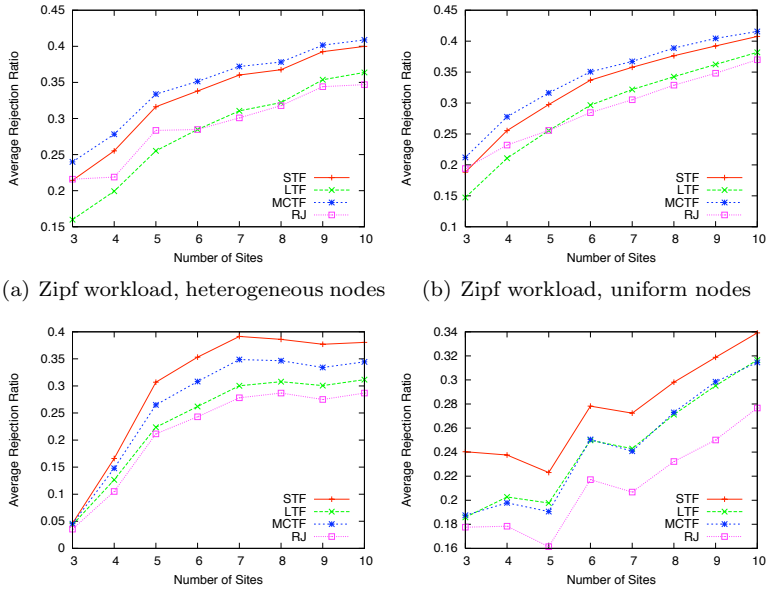
- *Topology*. We use the real Internet topology (i.e., Mapnet [3]) to evaluate the algorithms. We randomly select 3-10 nodes in the experiments. The costs of edges are computed based on the geographical distances between the nodes.

- *Node Resource Distribution*. We configure the experiment parameters close to real-life settings. According to the measurement by our implemented 3DTI system [116], the available bandwidth of tele-immersive sites on Internet2 could vary between 40 Mbps and 150 Mbps, and a 3D video stream after using a series of reduction techniques (e.g., background subtraction, resolution reduction, real-time 3D compression [70][112]) is approximately 5-10 Mbps. We evaluate two types of node capacity distribution: (1) *uniform*: a capacity of $O_i = I_i = 20 \pm \epsilon$, where $1 \le i \le N$ and ϵ is uniformly distributed between 0 and 5. The number of streams each site has to send is 20. (2) *heterogeneous*: fifty percent of the nodes have large capacity (30), twenty-five percent have medium capacity (20) and the other twenty-five percent have small capacity (10). The number of streams each site has to send is chosen uniformly between 10 and 30.

- *Subscription Workloads*. We mainly evaluate two types of subscription workloads: (1) *Zipf-distributed*: it has been shown that the stream popularity in multimedia applications follows a Zipf-like distribution [20]. We find this to be intuitively true in 3DTI environments, as the front cameras that capture people's faces are likely to be subscribed by most sites. (2) *random*: the randomized workload is to account for the possibility that the streams have more or less similar popularity in some 3DTI applications, such as surveillance and group collaboration. For both Zipf-distributed and random workloads, two hundred samples are generated to enumerate

the possible subscriptions (i.e., which streams are subscribed by which sites).

Rejection Ratio



(a) Zipf workload, heterogeneous nodes (b) Zipf workload, uniform nodes

(c) Random workload, heterogeneous nodes (d) Random workload, uniform nodes

Figure 4.9: Rejection ratios

Figure 4.9 shows the average rejection ratios (defined in Section 4.5.1) achieved by the tree-based algorithms and the basic randomized algorithm, under different node resource distribution and subscription workloads.

First, we notice the general trend is that the rejection ratio is increasing with the number of sites. This is because the total subscription workload grows much faster than the total available resources to serve the subscription requests. The resource per node is almost constant, whereas the subscription load grows with the total number of available streams.

Second, the data support our hypothesis that the LTF algorithm should perform better than STF. For example, with heterogeneous nodes under random workload (Figure 4.9(c)), LTF is about 25% better than STF. The rationale is that even if the last few trees cannot be constructed because of saturation, the number of rejected requests should be small because we are left with the smallest trees.

Third, as mentioned before, somewhat surprisingly RJ generally achieves the lowest rejection ratio in different experimental settings. For example, with uniform nodes under random workload (Figure 4.9(d)), RJ is about 26.7% better than STF, while 16.7% better than LTF and MCTF. Although LTF sometimes

obtains close performance to RJ (Figure 4.9(a) and 4.9(b)), it is computationally more expensive, because tree-based algorithms require sorting of all multicast groups, while RJ just randomly picks requests to serve. Therefore, RJ turns out to be the simplest but the most favorable solution in the unique problem context.

Granularity Analysis

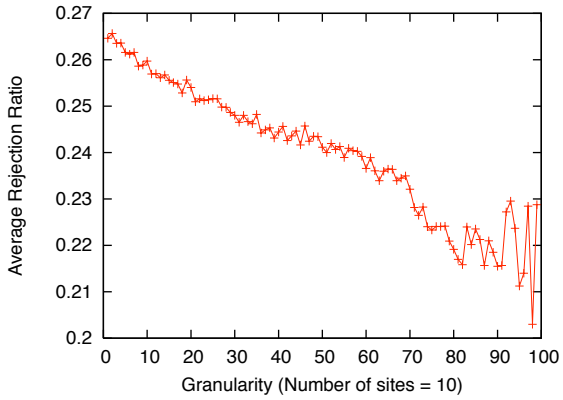
We observe that the RJ algorithm and the tree-based algorithms (LTF, STF, MCTF) are actually at two extreme ends of a more general spectrum of algorithms. We define the number of trees an algorithm attempts to construct at once as the *granularity*, g ($1 \leq g \leq |\mathbb{G}_{multicast}|$ and $g \in \mathbf{N}$, where $|\mathbb{G}_{multicast}|$ is the total number of multicast groups, or trees to construct). As two extreme cases, the granularity of all aforementioned tree-based algorithms is 1, while that of the randomized algorithm is $|\mathbb{G}_{multicast}|$. We perform experiments by incrementing the granularity value.

A modified LTF algorithm, called Gran-LTF, is used in this experiment as it is the best-performing tree-based algorithm among the three tree-based algorithms. Instead of constructing the trees one by one as in the original LTF algorithm, Gran-LTF first sorts all multicast groups in a descending order based on the size of the groups. It then picks the first g (number of) multicast groups for spanning tree construction. Within the set of g multicast groups (thus g trees), the requests are processed randomly using the basic node join algorithm (Section 4.5.1). Only after finishing processing all requests in the g multicast groups, the algorithm proceeds to the pick the next g trees to construct, and so forth.

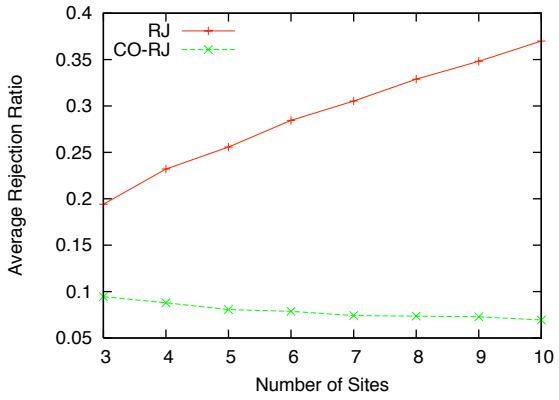
Figure 4.10(a) shows the result with ten uniform nodes under random workload. Note that when $g = |\mathbb{G}_{multicast}|$, Gran-LTF becomes RJ. We observe that generally the larger the granularity, the lower the rejection ratio. Although there is a small fluctuation region in the end (where granularity is large), the basic RJ algorithm is computationally simpler than others. The graphs for other experimental settings look similar to Figure 4.10(a) when N grows from 3 to 10.

Correlation

Finally, we compare the CO-RJ algorithm (Section 4.5.2) with the original RJ algorithm. Figure 4.10(b) shows the result with heterogeneous nodes under Zipf-distributed workload. In order to account for stream correlation, the definition of rejection ratio is modified as: $X' = \sum_{i=1}^{N}(\sum_{j=1}^{N} \frac{\hat{u}_{i\rightarrow j}}{u^2_{i\rightarrow j}}) \cdot u_{i\rightarrow x}$, where $u_{i\rightarrow x} = min(u_{i\rightarrow j})$ for $1 \leq j \leq N$. Figure 4.10(b) shows that CO-RJ's rejection ratio decreases as the number of sites grows, while RJ performs worse. When $N = 10$, CO-RJ is a factor of 5 better than RJ, which demonstrates the strength of the optimization based on stream correlation.



(a) Impact of granularity



(b) Impact of correlation

Figure 4.10: Granularity and correlation

In this section, we introduced the idea of selectively transmitting streams based on their contribution to user view. We explored a spectrum of heuristic algorithms to address the challenge of overlay construction with the selected streams. We found that a simple randomized algorithm worked well in this problem context. We hence proposed further optimization to the basic randomized algorithm by exploiting stream correlation. The experimental results demonstrated that the optimization mechanism achieved significant improvement over the basic algorithm.

4.6 Dynamic Topology Management

4.6.1 Problem

In the previous section, we explore static topology construction with the assumption that session controller receives all subscription requests in the session initialization phase before computing an overlay graph topology. However, after session starts to run, a key challenge is that dynamic view changes may cause frequent updates on the topologies. More specifically, a view change in one site may incur change in its inbound stream set, and thus affect the downstream sites that are receiving streams from it. Assume the user makes a view change at site A in Figure 4.8(b), such that stream s_g^8 is not needed any more. As a result, the offspring node C that used to receive the stream from A will be disrupted, having to relocate to a new parent.

Our results from Section 4.2 confirms that view change is likely to be frequent, unpredictable, and in small scale. In this section, we consider the practical challenge of dynamic changing views, and manage the topology at run-time. We explore new approaches to avoid stream disruption proactively, i.e., a peer that is less likely to lose the requested stream is considered better candidate as overlay neighbors during the run-time maintenance of the overlay topology. We propose, compare, and evaluate three algorithms for the stream selection and parent selection process as mentioned in Section 4.4. Similar to the algorithms in Section 4.5.2, these algorithms mainly run at the session controller for topology management. Our experiments show that an algorithm called Priority First that exploits view locality (Section 4.2) performs better than the other two algorithms in terms of resource utilization.

4.6.2 Heuristic Algorithms

Stream Selection

The stream selection algorithm takes place at each local site, particularly on the gateways G. Given a new view $\vec{V}_i^j$ selected by the user on renderer R_i^j at site i, the gateway G_i examines all the streams in the system, $\mathbb{S}$, with respect to this view. Then K streams with the highest contribution factors are selected. The

streams selected from each site are then sorted in the descending order of the contributing factors with respect to $\vec{V}_i^j$, and the stream ranked k ($1 \leq k \leq K$) is assigned the *priority degree* p_k of a priority scale. A priority scale $\mathbf{P} =< p_1, p_2, ..., p_K >$ is an ordered set of p_i, where $p_k > p_{k+1}$ holds. We denote the priority mapping function as $\mathcal{P}_k(\vec{V}_i^j, s) : CF(\vec{V}_i^j, \mathsf{s}) \mapsto p_k$ where the contributing factor $CF(\vec{V}_i^j, \mathsf{s})$ is mapped to a priority degree. We denote the ordered set of streams as

$$\mathbb{S}_{\vec{V}_i^j}^{requested} = \{\{\mathsf{s}_1^1, \mathsf{s}_1^2, ..., \mathsf{s}_1^K\}, \{\mathsf{s}_2^1, \mathsf{s}_2^2, ..., \mathsf{s}_2^K\}, ..., \{\mathsf{s}_M^1, \mathsf{s}_M^2, ..., \mathsf{s}_M^K\}\} \tag{4.2}$$

where streams $\mathsf{s}_j^{\{1...K\}}$ are those selected from site j ($1 \leq j \leq M$), as ordered in the descending order of priority. Thus, stream s_j^k has the priority degree p_k, i.e., $\mathcal{P}(\vec{V}_i^j, s_j^k) > \mathcal{P}(\vec{V}_i^j, s_j^{k+1})$ and $\mathcal{P}(\vec{V}_i^j, \mathsf{s}_j^k) = p_k$, where $1 \leq j \leq M$ and $1 \leq k < K$. The insight of the stream selection process is that by differentiating the streams according to their semantic importance with respect to a view, we can efficiently utilize the limited resources by only delivering those semantically important streams.

Parent Selection Algorithms

The parent selection algorithms take place in the session controller (Figure 1.4). For each selected stream, the gateway G_i checks whether it has the stream already. If so (e.g., those streams produced locally), the gateway delivers it to the renderer. Otherwise, a *subscription request* is generated in the form of $[view, streamID, priority]$ or $[\vec{V}_i^j, \mathsf{s}_j^k, p_k]$. A *priority queue* Q_k ($1 \leq k \leq K$) is then constructed on the session controller for each priority degree, and the set of requests are fed into the queue that has their priority degree in a random fashion (for fairness). For example, stream s_3^1 (stream # 1 from site 3) will be fed into queue Q_1, s_2^3 will be fed into Q_3, and so forth. Then the session controller finds the non-empty queue of the highest priority, removes a request $[\vec{V}_i^j, \mathsf{s}_j^k, p_k]$ from that queue, and looks for a "good" parent node to serve s_j^k using one of the approaches presented in Section 4.6.2. Note that we are using application-level overlay for the data dissemination, so any node (passive or active) that has (i.e., either produces or receives) s_j^k is a possible *candidate parent*. If such parent can be found, a reply is sent back to G_i, so that G_i can start connection with the parent node to receive s_j^k. Otherwise, rejection is reported.

We consider the following criteria when comparing the candidate nodes: (a) *bandwidth capacity*: the overlay link from the candidate node to the requesting node G_i should have sufficient bandwidth capacity to serve the stream; (b) *latency constraint*: if G_i were connected to the candidate node for receiving the stream, the end-to-end latency from the source of the stream to G_i should be smaller than a soft real-time bound; and (c) *chance of losing the stream*: we also consider the chance for the candidate node to lose the stream due to its

own view change.

Next we discuss the details of several algorithms. We first introduce the following notations. (a) *Candidate Set*: $\mathbb{C}(\vec{V}_i^j, \mathsf{s})$, the set of nodes that has stream s to serve view $\vec{V}_i^j$ satisfying the latency constraint. (b) *Candidate Subset 1*: $\mathbb{C}_1(\vec{V}_i^j, \mathsf{s})$, the subset of $\mathbb{C}(\vec{V}_i^j, \mathsf{s})$ which has available bandwidth. $\mathbb{C}_1(\vec{V}_i^j, \mathsf{s}) \subseteq C(\vec{V}_i^j, \mathsf{s})$. (c) *Candidate Subset 2*: $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$, the subset of $\mathbb{C}(\vec{V}_i^j, \mathsf{s})$ which does not have available bandwidth. $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s}) = \mathbb{C}(\vec{V}_i^j, s) - \mathbb{C}_1(\vec{V}_i^j, \mathsf{s})$. $\mathbb{C}(\vec{V}_i^j, \mathsf{s})$ addressed the latency constraint, and its subset $\mathbb{C}_1(\vec{V}_i^j, \mathsf{s})$ further resolves the bandwidth concern. We then propose three fundamental approaches for parent selection (Section 4.4), with further consideration of the reliability of a candidate node for serving the requested stream.

- **Random (Rand)**: Assuming no knowledge about the future and unpredictable user view change patterns, a natural and simple approach is to use a randomized algorithm. If there are candidates with available bandwidth (i.e., $\mathbb{C}_1(\vec{V}_i^j, \mathsf{s}) \neq \emptyset$), randomly select one as the parent for G_i. If there is no such candidate but $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$ is not empty, we select a candidate node G_p from $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$ that has a lower-priority stream s' to preempt, that is, $\mathcal{P}(\vec{V}_i^j, \mathsf{s}') < \mathcal{P}(\vec{V}_i^j, \mathsf{s})$. Further, among all such candidate nodes, we select one that has the minimum *preemption impact slot* (we defer the more detailed description of the preemption mechanism to Section 4.6.2). If no candidate is found, report rejection. It is worth pointing out that this algorithm is randomized in the sense that it randomly picks a parent node from the candidate set. In Section 4.5, the randomness of the RJ algorithm refers the randomized order of processing subscription requests.

- **Proximity First (Pxf)**: This algorithm is based on the observation in our previous human experiments [110] that when users change view, they would usually change in small scale and thus stay in proximity relatively (Section 4.2). We refer to this phenomena as *view locality*. Therefore, a candidate node G_p with a view $\vec{V}_p$ closest to $\vec{V}_i^j$ in the 3D space is likely to stay in the proximity (and thus have s) even when the user at site j changes her view. If $\mathbb{C}_1(\vec{V}_i^j, \mathsf{s})$ is not empty, for each node G_g in the set, compute its *view proximity* to $\vec{V}_i^j$, as $\vec{V}_g \cdot \vec{V}_i^j$ for each renderer in site g. Select the node with the highest view proximity to serve s. If there is no candidate with available bandwidth and $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$ is not empty, use the same preemption mechanism as in Rand to preempt (Section 4.6.2). If both candidate sets are empty, report rejection.

- **Priority First (Prf)**: We further observe that a more reliable measure that captures the reliability of a candidate node for serving s is the priority degree of the stream to the node. That is, it is least likely for a node to lose a stream that is of the highest priority to it. If there are nodes with available bandwidth, sort the nodes of $\mathbb{C}_1(\vec{V}_i^j, \mathsf{s})$ in descending order of

the priority of s with respect to its own view $\vec{V}_g$. Select the node with the highest value of $\mathcal{P}(\vec{V}_g, \mathsf{s})$ to serve G_i with s. If there is no candidate with available bandwidth but $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$ is not empty, again use the preemption mechanism. If no candidate can be found, report rejection. Note that this algorithm is also based on view locality.

Figure 4.11 illustrates the three different algorithms with an example. Assume s is the stream requested by G_6, and $K = 4$. As shown in the figure, $\mathsf{G}_1, ..., \mathsf{G}_4$ are the nodes that have s. Suppose the latency requirement eliminates G_2 as a candidate. G_1, G_3, and G_4 do have enough spare bandwidth left to serve s, so they fall into $\mathbb{C}_1(\vec{V}_6, \mathsf{s})$. Assume $\mathcal{P}(\vec{V}_6, \mathsf{s}) = p_1$, meaning s is among the most important streams to G_6. Among the candidate nodes, Rand will pick any random node, say G_3 to serve G_6. Pxf would pick a node with the closest view to G_6, say G_1. And Prf would check the priority of s to the candidate nodes. Hence the node G_1, to whom s has the highest priority p_1, becomes the parent node to G_6.

Preemption

As aforementioned, distributed 3DTI system has an overly high demand/stress for networking resources given the huge amount of data to deliver over COTS components. We believe it is important to take a prioritized approach in such context so that the limited resources are utilized efficiently. When no candidate has the spare bandwidth to serve s for G_i and s has a relatively high priority, we use the preemption mechanism to find a candidate node in $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$ and preempt one of its lower-priority streams so that s can be delivered to G_i.

Unlike 2D video streams, 3D video streams are presented in an aggregated fashion on the renderers. Therefore, preempting some unimportant streams is much less noticeable to the users than in a 2D video-mediated system. For example, if the user is looking at a person's front view, losing some side stream is not as visually disrupting. The key is to identify the stream that is least important.

At preemption, we aim to minimize the impact to the downstream nodes in the overlay when performing preemption. So for each candidate node G_g in $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$, we check if there is any stream s' the node is sending/forwarding that has a lower priority than the requested stream s, that is, $\mathcal{P}(\vec{V}_g, \mathsf{s}') < \mathcal{P}(\vec{V}_i^j, \mathsf{s})$. If so, we compute the preemption impact for the slot/stream s', denoted by $I_g^{\mathsf{s}'}$, as follows. Initially, $I_g^{\mathsf{s}'}$ is set to 0. In the overlay tree of s', traverse the sub-tree rooted at G_g (excluding G_g). If the priority degree of s' to a downstream node G_h ($\mathsf{G}_h \neq \mathsf{G}_g$) is p_i, add $K - i$ to $I_g^{\mathsf{s}'}$. The final sum is the preemption impact for G_g on stream s'. Among the candidate nodes in $\mathbb{C}_2(\vec{V}_i^j, \mathsf{s})$ that has a lower-priority slot, select the node G_g with the minimum preemption impact out-slot, and use it to serve s to G_i instead.

If G_g is not a leaf node on the overlay tree for the victim stream s', we

traverse the subtree of G_g to repair the disrupted nodes. Suppose s' has priority p_m to a downstream node G_d in the subtree rooted at G_g, we check whether $m \leq K/2$, that is, if it is one of the more important streams. If so, a new request $[\vec{V}_d, \mathsf{s}', \mathcal{P}(\vec{V}_d, \mathsf{s}')]$ is generated and placed into the corresponding priority queue Q so the lost stream can be retrieved.

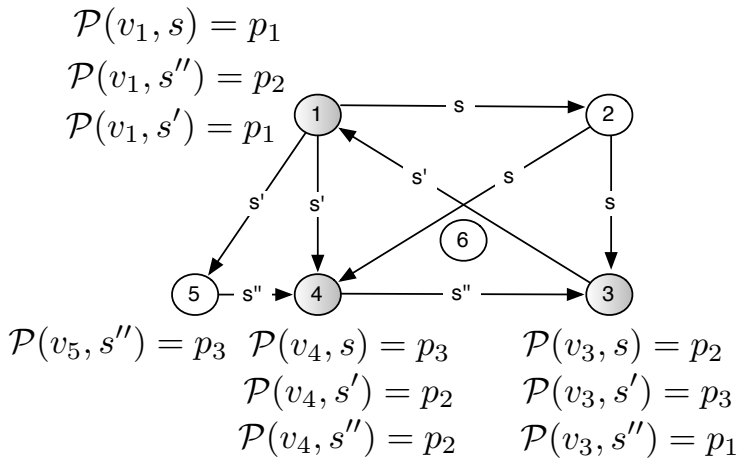


Figure 4.11: Example of preemption impact.

We reuse Figure 4.11 to illustrate the preemption algorithm. Assume G_1, G_3, and G_4 in this case do not have any spare bandwidth left to serve s, so they fall into $\mathbb{C}_2(\vec{V}_6, \mathsf{s})$. Among the candidate nodes, G_3 has a lower-priority outgoing stream s', and G_4 has lower-priority outgoing stream s''. We can then compute the preemption impact as $I_3^{\mathsf{s}'} = 3(\mathsf{G}_1) + 2(\mathsf{G}_4) + 1(\mathsf{G}_5) = 6$ and $I_4^{\mathsf{s}''} = 3(\mathsf{G}_3) = 3$. Hence, G_4 has the minimum preemption impact slot of s''. We thus preempt s'' and use this out-slot to serve s to G_6.

4.6.3 Performance Evaluation

Simulation Setup

We study the algorithms with extensive simulation. Below we describe the topology, node resources, and dynamic workload for the simulation setup. The simulator is written in C/C++.

- **Topology**. We vary the total number of sites (or nodes) from 22 to 46, and differentiate three types of nodes in the topology: (1) *tele-immersive nodes*: which have original streams to send out, representing the gateways in 3D tele-immersion sites that have cameras installed, (2) *super viewing nodes*: which have direct connection (on the overlay level) to the tele-immersive nodes and available bandwidth to serve M views without any loss (i.e., $K \times M$ streams), and (3) *normal viewing nodes*: which only have direct connection with the super viewing nodes. We evaluate five

Table 4.1: Node distribution for dynamic inter-stream adaptation experiments.

Total	**Tele-immersive**	**Super Viewing**	**Normal Viewing**
22	5	7	10
28	5	8	15
34	5	9	20
40	5	10	25
46	5	11	30

configurations (in terms of number of nodes) as shown in Table 4.6.3. The delay along each link is normal distribution of $N(50, 10)$, (i.e., average 50 ms). The maximum tolerable delay is 200 ms from the source to the destination.

- **Node Resources**. Each tele-immersive node produces 8 streams, among which at most 4 streams should be selected to serve a view, i.e., $K = 4$. Each stream has the data rate of 1 (unit). We evaluate two types of bandwidth distribution: (a) *uniform*: each link (among super and normal vertices) has the same bandwidth bound from 2 to 5 where the number means how many streams can be transmitted, and (b) heterogeneous: around 70% of the links have 3 or 4, and the rest have 2 or 5.

- **Dynamic Workload**. We evaluate two types of view change workload: (a) *uniform* - which simulate the user data we observe [110]: the view change interval is a normal distribution of $N(60, 10)$, (i.e., average 60 seconds). The view change pattern is 95% of the time, a random walk with $N(35°, 5°)$, and 5% of the time, a more dramatic view change of 90° is applied. (b) *Zipf* - which is common pattern in data selection [20]: specifically ten view directions are pre-defined to uniformly divide the cyberspace with an alpha being 1.0, and the view change follows the distribution Zipf(10, 1.0) (i.e., the exponent is 1.0). Each simulation runs for 100 virtual minutes. Each run is executed 6 times to compute the confidence interval.

Rejection Ratio

Figure 4.12(a)(d)(g) show the overall rejection ratios for different algorithms when the bandwidth bound (BW) is 2, 5, and heterogeneous (denoted by HT), and when the workload follows normal distribution. The values are averaged across different runs. We do not show graphs for BW= 3, 4, because they actually look quite similar to that of BW=5.

First, we observe that the locality-based approaches generally perform better than the randomized one. For example, when the total number of nodes is 46 and BW is 2, Prf and Pxf are roughly 33.3% better than Rand. With the same view change workload, utilizing view locality does significantly reduce future

rejections. Second, the rejection ratio of Prf is 7%-10%, which is reasonable (confirming the choice of K).

Since different streams have different semantic importance, we also evaluate the breakdown of the rejected requests. Figure 4.12(b)(e)(h) show the results. On average, the rejection ratio of the streams with p_1 is around 1%. Considering the overall rejection ratio is 8∼9% and there are four priority degrees in total, the margin of improvement is about 50%.

We do not show the graphs for the Zipf-distributed workload, which look quite similar to Figure 4.12(b)(e)(h). For example, when the total number of nodes is 46 and BW is 5, Prf and Pxf are roughly 35.3% better than Rand for Zipf-based workload. Prf and Pxf also perform similarly in terms of rejection ratio, but the overall rejection ratio of Prf is 5%-9%, which is a little lower than in the uniformly distributed data.

View Change Tolerance

We evaluate the system interference in terms of the number of victims incurred in each algorithm. The victim is defined as the victim stream that gets preempted for a high-priority stream. Figure 4.12(c)(f)(i) present the results.

First, although in terms of rejection ratio Pxf performs better than Rand, we see that Rand has lower number of victims than Pxf. For example, when the total number of nodes is 46 and BW is heterogeneous, Rand has about 16.7% less interference than Pxf.

Second, we observe that Prf performs much better than Rand and Pxf generally. When the size of the system is 46 and BW is 2, for example, Prf is about 46.7% better than Rand and 50% better than Pxf. And the trend is that as the number of vertices increases, the difference between the performances of Prf and Pxf or Rand increases.

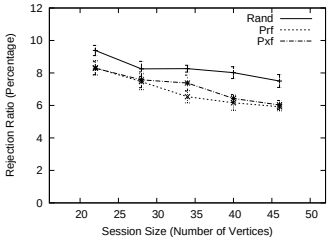
For Zipf-based data, Prf and Pxf generally achieve lower numbers of victims than Rand. For example, when the total number of nodes is 46 and BW is 5, Rand has about 64% more victims than the two other algorithms. Prf is still generally better than Pxf, but the discrepancy (1%-10%) is smaller than that in the uniformly distributed workload.

In summary, our contribution is twofold: (1) we identify the challenges of dynamic topology maintenance in distributed 3DTI systems; (2) we compare three algorithms and demonstrate that Priority First achieves efficient resource utilization and high tolerance under different dynamics patterns by exploiting view locality in a fine granularity.

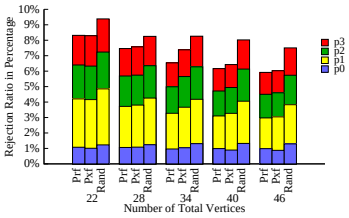
4.7 Conclusion

In this chapter, we considered the practical challenges in relieving the bandwidth demand for multi-site 3DTI collaboration. Basically we leveraged user view to

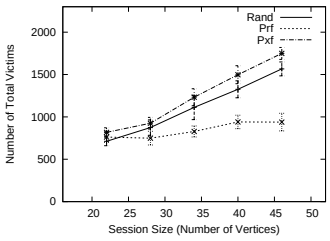
prioritize video streams for transmission. We first conducted a subjective study to understand what view means and how users actually change views. In light of the results, we studied a suite of algorithms for view-based stream adaptation. We tackled the key challenge of static overlay construction by a spectrum of heuristic algorithms during the session initialization phase. We also addressed the challenges introduced by dynamic user view changes present in the session running phase.



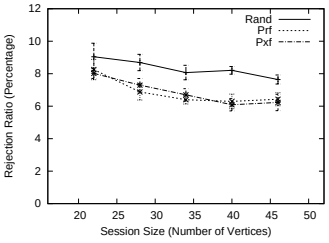
(a) Rejection Ratio (BW=2)



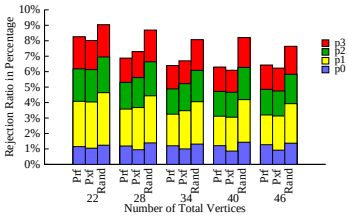
(b) Rejections Breakdown (BW=2)



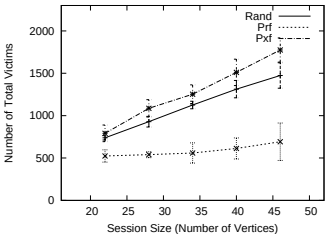
(c) Number of Victims (BW=2)



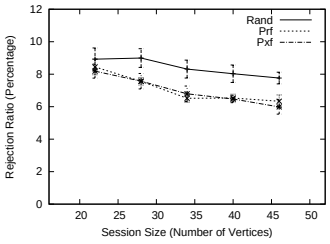
(d) Rejection Ratio (BW=5)



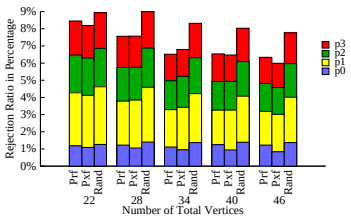
(e) Rejections Breakdown (BW=5)



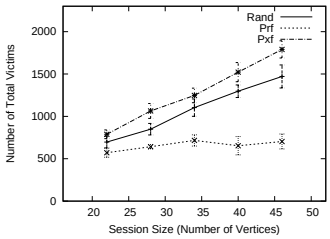
(f) Number of Victims (BW=5)



(g) Rejection Ratio (BW=HT)



(h) Rejections Breakdown (BW=HT)



(i) Number of Victims (BW=HT)

Figure 4.12: Experimental results comparing the performances of Priority First, Proximity First, and Random algorithms for dynamic inter-stream adaptation.

5 Comprehensive Quality Framework

5.1 Introduction

Recall that in Section 1.3.1 we mention that HCC represents a set of principles and strategies that bear "the human focus from the beginning to the end" [53]. Chapter 3 and 4 apply HCC into the design, development and execution of 3DTI systems. As our design and implementation paradigms shift to the human-centered domain, the evaluation of the system is the only puzzle piece left. In this chapter, we apply HCC principles in the last phase of 3DTI pipeline (i.e., rendering/visualization as shown in Figure 1.2), where the user experience can be eventually evaluated.

We have improved various quality metrics in the previous two chapters. To name a few, end-to-end latency, frame rate, perceived visual quality, and bandwidth utilization. However, we have not holistically or systematically considered the "quality" concept in the 3DTI context. What is exactly "quality"? Can we derive a taxonomy of quality metrics that can be measured in the ending phase of the system? From the human-centered perspective, what is user experience, or Quality-of-Experience (QoE)? How does it compare to the traditional Quality-of-Service (QoS)? What are their relationships?

Empirical findings have shown us that systems excelling in QoS can completely fail for user adoption due to the gap between system- and human-centric evaluations [27]. However, there has been little understanding about the user-centered measure - Quality of Experience (QoE) - in the multimedia communities [6][49][56][72][98]. Researchers have made attempts to add subjective questions in the performance assessments in multimedia systems [10][24]. However, the existing ad hoc methodologies only leave us with a bewildering welter of "quality" metrics that are application-specific and not practically generalizable.

So what is QoE? How can we model it? What are the relationships between QoS and QoE? How do we measure their qualities and relationships? These unanswered research questions became the motivation for our work. Guided by the theories in psychology, cognitive science, sociology, and information technology, we model QoE as a *multi-dimensional construct of user perceptions and behaviors*. As shown in Figure 5.1, the relationship between QoS and QoE is formed as a causal chain of "environmental influences → cognitive perceptions → behavioral consequences [71]", where QoS metrics represent the environ-

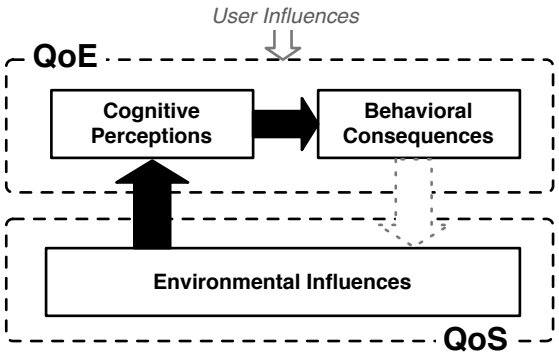


Figure 5.1: The relationship between QoS and QoE is formed as a causal chain of "environmental influences → cognitive perceptions → behavioral consequences.

mental factors that influence QoE[1]. We describe a methodology for correlation mapping between the constructs in the quality framework.

In the context of our control loop (Figure 1.6), the QoS metrics we evaluate are mainly the objective, application-level QoS perceivable by users at the receiving side, and the QoE metrics we consider are the experience metrics users actually perceive. For example, frame rate is the QoS users can perceive, and the sense of telepresence is what the users actually experience. Based on the characteristics of 3DTI systems, we build up a taxonomy of dimensions for both QoS and QoE in the framework, along with a classification of the metrics that are commonly used in practice and our previous work such as end-to-end delay, visual quality, and frame rate (Chapter 3, 4). Finally, we explore the methodology of QoS-QoE mapping (correlation) by applying the framework to our empirical studies of a 3DTI system. The results from a two-week controlled study and a one-year uncontrolled field study are presented.

In summary, our contributions in this chapter include the following. We first provide a clear definition of QoE and its conceptual model in 3DTI systems. Instead of thinking QoE as an extension or subset of QoS [49][56], we propose to consider the two constructs as distinct components on a causal chain. Last but not least, we present a methodology to compute the mappings from QoS to QoE, which can offer useful insights for 3DTI designers and practitioners. The results present the first deep study to model the multi-facet QoE construct, map the QoS-QoE relationship, and capture the human-centric quality modalities in the context of 3DTI systems.

[1]There is also a feedback loop from QoE to QoS (as shown by the dashed arrow in Figure 5.1), where the requirements and responses of users may drive the configuration of desired QoS. However, in this work we mainly focus on the QoS → QoE mapping, as the understanding of this relationship may significantly advance the field.

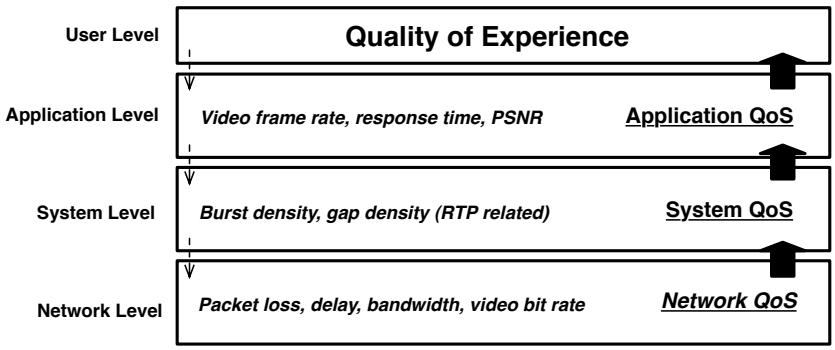


Figure 5.2: QoS refers to a set of measures for tuning or quantifying the performance of applications, systems, and networks. In particular, the application QoS metrics, strongly influenced by the underlying system and network QoS, are those possibly perceptible by users, thereby directly correlated with QoE.

5.2 Our Approaches

5.2.1 Overview

In this section, we provide the definitions of QoS and QoE, describe their conceptual relationships, and give an overview of the conceptual framework being built.

QoS refers to a set of measures for tuning or quantifying the performance of applications, systems, and networks. Figure 5.2 illustrates the QoS factors in the protocol stack and their conceptual relations with QoE. For example, we have considered frame rate, response time (end-to-end delay), bandwidth usage, visual quality, etc. in Chapter 3 and Chapter 4. In particular, the application QoS metrics, strongly influenced by the underlying system and network QoS, are those possibly perceptible by users, thereby directly correlated with QoE.

While QoS is well defined, the meaning of QoE is being argued. For instance, the standardization group ITU-T suggests that QoE should be represented by Mean Opinion Score (MOS), a Likert-scale rank for subjective testing of voice/video quality [5]. Beauregard *et al.* formulated QoE as "the degree to which a system meets the target user's tacit and explicit expectations for experience" [14]. Some other informal definitions are "subjective measure of a customer's experiences with a vendor", "user perceived performance", and "the degree of satisfaction of users".

In the various formal and informal definitions, QoE has been framed as a subjective measure. According to psychology theories [71], environmental stimuli greatly influence one's cognitive perceptions, and in turn shape behavioral intentions and outcomes. If we treat technological systems as the "environments", their influences, quantified by QoS metrics, may lead to subjective and objective responses of users, both of which we consider part of the "user experience". Our definition for QoE thus follows.

Definition: *QoE is a multi-dimensional construct of perceptions and behaviors of a user, which represents his/her emotional, cognitive, and behavioral responses, both subjective and objective, while using a system.*

Figure 5.1 illustrates the general relationship between QoS and QoE. Notice that QoE is not only influenced by the technological environment, but also by the human factors that strongly embed user's experiences and cultural backgrounds[2].

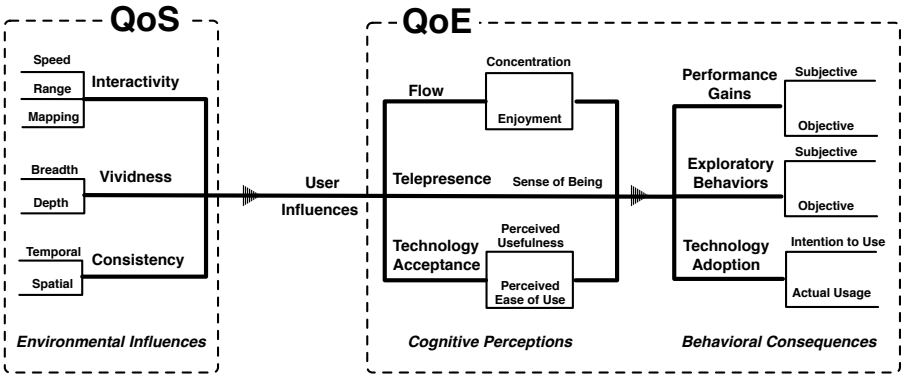


Figure 5.3: Dimensions of the Quality-of-Experience (QoE) and Quality-of-Service (QoS) in 3DTI systems and their relationships.

Next, we present the experiential quality framework in 3DTI systems along with its theoretical foundations. The research methodology is to consider user's multiple roles for the modeling of QoE dimensions (Section 5.2.2), and heuristically find application-level QoS metrics that can have significant impact on user-level QoE (Section 5.2.3). Accompanying the modeling of QoS is a classification of the common metrics being used. Finally, we survey the existing 3DTI systems and compare them within our quality framework (Section 5.2.4). Figure 5.3 presents our integrated quality framework.

5.2.2 Quality of Experience Construct

As Figure 5.1 shows, the QoE model and its representative dimensions include both the cognitive perceptions and behavioral consequences of users.

Cognitive Perceptions

We consider three main dimensions of cognitive perceptions: *psychological flow*, *perceived technology acceptance*, and *telepresence*. These dimensions characterize user's three roles in 3DTI systems : executant of tasks, user of technology,

[2]In this work we mainly focus on the technological influences, and leave the investigation of the cultural and experiential influences to future work.

and participant in group telecommunication. Likert scale has been most commonly used to evaluate the cognitive perceptions of users.

- *Flow*. For the role of task executant, flow can measure "the holistic sensation that people feel when they act with total involvement", which is the main intrinsic motivation for people to perform activities that provide no discernible extrinsic rewards [21]. When in the flow state, people focus their full attention on the task at hand; they perceive a sense of control and great enjoyment. The intense experiential involvement is a natural moment-to-moment flow of mind, and is found universal in human activities such as reading, chess playing, and rock climbing [22]. Flow was originally characterized via eight components, including clear goals, feedback, challenge/skill balance, concentration, sense of control, loss of self consciousness, distorted sense of time, and intrinsic enjoyment. Although these are valuable components, the flow concept was too broadly defined, failing to capture some specific characteristics of the technological environments. Subsequent research on computer-mediated interaction has adapted its list of metrics [37][43][62]. Based on our empirical findings in previous research [90][114], we identify three metrics that are significantly relevant to 3DTI systems : *concentration*, *intrinsic enjoyment*, and *sense of control*.

- *Perceived Technology Acceptance*. The flow metrics convey the psychological experience of users without considering the technological environments. We use the Technology Acceptance Model (TAM) [26] to further account for user's perceptions/attitudes toward the technology in the role of a technology user. The *perceived usefulness* and *perceived ease of use* are the two belief variables of TAM. The former represents "the degree to which a person believes that using a particular system would enhance his/her performance", whereas the latter defines "the degree to which a person believes that using a particular system would be free of effort". The flow metric "sense of control" and the TAM's belief variable "perceived ease of use" are strongly related, and are thus combined into one. According to the theory of reasoned action [31], beliefs about the consequences of performing the behavior largely shape one's behavioral intentions and consequences. By treating 3DTI systems as IT systems, we can apply TAM and examine how the two belief metrics predict user adoption of technology.

- *Telepresence*. Users in 3DTI systems are also participants in remote telecommunication. Telepresence characterizes user's perceptual "sense of being" or "sense of presence" in the holistic communication environment rather than in the real world. Users have reported their telepresence experience in various ways, e.g., "I'm noticing a different awareness, somewhat like an out of body experience", "I feel like our body exists in the

3D virtual environment, rather than the real world" [90], "My immediate surroundings became less important and/or noticeable - as if I almost forgot them", "I felt like I came back to the 'real world' after a journey." In fact, the difference between virtual reality and other media was defined as a difference in the level of presence [97]. So the metric is a significant indicator of user experience in 3DTI systems .

Behavioral Consequences

Behavioral consequences are the results of cognitive perceptions (Figure 5.1). They can be *subjective* or *objective*. Subjective consequences refer to one's perspectives and desires, which are only available in the subject's consciousness. Objective consequences, in contrast, refer to one's actual conducts, actions, and performances, which can generally be observed and quantified. We analyze three dimensions of behavioral consequences: *performance gains*, *technology adoption*, and *exploratory behaviors*.

- *Performance Gains*. Performance gains represent the amount of increase in user's performance on certain tasks, which can be measured subjectively and objectively. The metrics of this dimension depend on the actual application environments and task requirements. Researchers usually design controlled studies to quantify performance gains in well-specified tasks, where the widely used metrics are the ratio of successful attempts and completion time [87]. It is hypothesized that cognitive experience is positively correlated with performance gains.

- *Technology Adoption*. *Intention to use* (subjective) and *actual usage* (objective) are the two variables for technology adoption. They are directly related with user's perceptual 'technology acceptance' (Section 5.2.2). For technological systems, *intention to use* is regarded as the major subjective metric in user experience evaluation [47][52][62][78]. An advantage of this metric is its relative ease of assessment. Its objective counterpart - actual system usage - is an important indicator for the extent of technology adoption. Nevertheless, researchers need to observe users over time to quantify this metric (e.g., six months of field study [102]), which can be challenging in controlled studies. According to the theory of planned behavior [7], behavioral intention is a strong predictor of actual behaviors. Thus, "intention to use" often becomes the substitute in actual evaluations [52].

- *Exploratory Behaviors*. Exploratory behaviors represent user's spontaneous exploration of the technology with no particular preset plans or goals. It has been shown that cognitive perceptions are positively correlated with the yield of exploratory behaviors [37]. Exploratory behaviors can be measured subjectively and objectively. The metrics here are application-specific as those for performance gains. As a simple example,

in evaluating web-based services, researchers would ask users to rate for statements like "I often click on a link just out of curiosity" and "Surfing the web to see what's new is a waste of time" (reverse-scaled) [77]. The actual amount of exploratory behaviors can be measured objectively by observing users in uncontrolled studies.

Figure 5.3 outlines the QoE construct, its dimensions and metrics that we have identified and modeled in terms of their inter-relationships.

5.2.3 Quality of Service Construct

In environmental psychology, the term "environment" is broadly defined to include natural, built, cultural, social, and informational settings [38]. In 3DTI systems , the technological environment serves as the primary context, bringing the most direct and significant impact on QoE. A variety of QoS metrics have been used to quantify the performance of systems. However, QoS lacks a conceptual framework of classification. We study the application-level QoS metrics that can directly influence QoE (Figure 5.2), and provide a taxonomy of such metrics in 3DTI systems .

Jonathan Steuer [97] proposed two dimensions of telepresence: *vividness* and *interactivity*, both of which we find essential for creating compelling user experience in 3DTI systems . In the quality framework, we label "telepresence" as the cognitive perception, and its two dimensions as the environmental influences (refer to Figure 5.1).

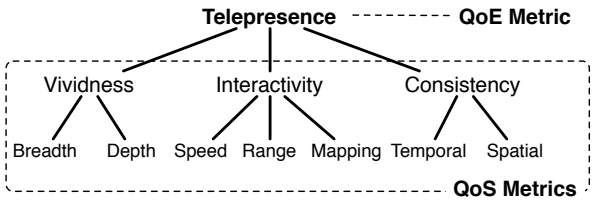


Figure 5.4: Dimensions and classification of QoS in 3DTI environments (adapted from [97]).

- *Vividness.* Vividness means "representational richness of a mediated environment" [97] which is modeled by the amount of sensory information simultaneously presented to the users. It has two dimensions: *breadth* and *depth*, where breadth refers to the number of sensory channels, while depth refers to the resolution in each of these perceptual channels. Vividness breadth can translate to a number of metrics in 3DTI systems , including the presence of media channel (e.g., visual, auditory, haptic, textual, graphical), end device sensing range (e.g., camera, microphone), and the number of views served (as we have studied in Chapter 4). Vividness depth corresponds to metrics such as CZLoD (Chapter 3), peak signal

to noise ratio (PSNR), pixel resolution (such as 640x480 and 320x240 we have considered in our CZLoD experiments), haptic feedback accuracy, visual tracking precision, video frame loss, and audio amplification factor.

- *Interactivity.* Interactivity represents "the extent to which users can participate in modifying the form and content of a mediated environment in real time" [97]. There are three factors that express interactivity: *speed*, *range*, and *mapping*. Speed refers to the rate at which user input can be assimilated to the environment. This metric is directly related to end-to-end delay, one of the most critical QoS metrics in 3DTI systems [103], one that we have considered in Chapter 4. Other metrics include reaction/response time, image freeze time, jitter, video frame rate (as we have considered in Chapter 3), audio nominal rate, and graphics update rate. Interactivity range represents the scale of control options for users to change the mediated environment. A typical example in 3DTI systems is the ability to change viewpoint in a holistic 3D environment as we have explored in Section 4.2. Other commonly used metrics are interface flexibility, customization degree, number of control options, number of accessible parameters. Finally, interactivity mapping measures the capability of the 3DTI systems to map user control to actual changes in the mediated environment, i.e., how natural and intuitive the user interface is, which is generally applicable to all human-computer interactions.

- *Consistency*. An essential concern for 3DTI systems is not addressed in Steuer's telepresence model: *consistency*. The consistency requirement has been formally modeled in the human communication theory [94], where the term is coined as *mutual manifestness*. Therein, the communicative principle states that facts in the communication environment should be mutually conveyed to the participating agents; otherwise, the difference of perceived contexts will lead to misunderstanding and confusion. In the traditional face-to-face settings, the actual environment is naturally consistent to everyone physically present. When it comes to virtual reality, however, consistency has to be explicitly achieved by proper design and implementation of the mediation systems. There are two dimensions of consistency in 3DTI systems : *spatial* and *temporal*.

  - *Spatial consistency* refers to the topological scale of state synchronization, i.e., a site may know a subset (*partial consistency*) or total set (*global consistency*) of states in the system (as illustrated in Figure 5.5). In large-scale systems, it is often not practical or necessary to achieve global spatial consistency. The commonly used metrics for spatial consistency include coverage, completeness, and consensus.
  - *Temporal consistency* refers to the degree of time synchronization of all states in the 3DTI systems, which is hypothesized to impose a

more significant impact on user QoE than its spatial counterpart. In 3DTI systems , the local states are exchanged over networks to create the shared communicative context for everyone, which inevitably incurs inconsistencies due to the existence of propagation delays, lossy links, etc. Researchers have proposed conceptual models to characterize temporal consistency in distributed environments. Figure 5.5 illustrates the *absolute consistency* and *delayed consistency* models for temporal consistency in 3DTI systems [83], where the former ensures that all operations execute at the same time across the system and the latter trades the degree of consistency for response time by allowing local operations to instantaneously take effect. The corresponding QoS metrics for temporal consistency include phase difference, dropping ratio (due to synchronization), uniformity of flow, drift distance, and continuity.

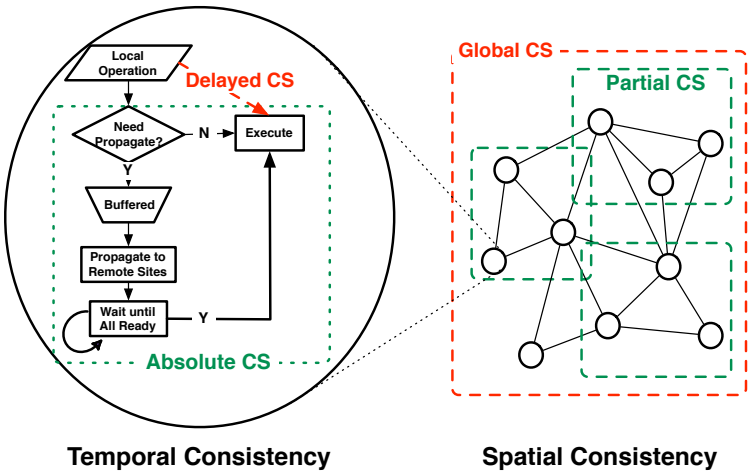


Figure 5.5: Temporal and spatial consistency (CS) models in 3DTI environments. Temporal consistency can further be characterized by absolute or delayed consistency; whereas spatial consistency can be characterized by global or local consistency.

In summary, we identify three important dimensions of QoS that are tightly connected to 3DTI systems : *context vividness*, *interactivity*, and *consistency*, as shown in Figure 5.4. How the 3DTI system is designed on these QoS dimensions directly shapes the user experience.

5.2.4 Comparison

In this section, we fit the existing work on 3DTI evaluation into the quality framework, and examine how the constructs cover the cases in reality. In total, there are three components in the causal chain framework: (1) environmental influences, which include variables: Interactivity Speed (IS), Interactivity Range

(IR), Interactivity Mapping (IM), Vividness Breadth (VB), Vividness Depth (VD), Temporal Consistency (TC), and Spatial Consistency (SC), (2) cognitive perceptions, which include variables: Flow Concentration (FC), Flow Enjoyment (FE), Telepresence (TP), Perceived Usefulness (PU), and Perceived Ease of Use (PEU), and (3) behavioral consequences, which include variables: Performance Gains (PG), Exploratory Behaviors (EB), and Technology Adoption (TA).

Table 5.1 summarizes the fitting results. We observe that most of the existing studies have considered a small subset of these measures (max ratio = 6/16, min = 1/16). IR and TA are among the least used ones, due to their difficulty of evaluation and quantification in reality. SC is also rarely evaluated as it is less discernible by users compared to TC.

Table 5.1: Existing 3DTI systems and the factors considered in their evaluations.

	Environmental Influences							*Cognitive. Perceptions*					*Behaviors*		
	IS	*IR*	*IM*	*VB*	*VD*	*TC*	*SC*	*FC*	*FE*	*TP*	*PU*	*PEU*	*PG*	*EB*	*TA*
[10]	√								√		√		√	√	
[15]	√	√				√	√								
[19]			√		√								√		
[24]	√			√	√			√	√					√	
[34]				√	√						√	√	√		
[35]						√									
[47]								√	√		√	√	√		√
[60]	√														
[74]				√					√		√	√			
[87]			√	√									√	√	
[113]	√					√									

5.2.5 Empirical Mapping between QoE and QoS

In this section, we present an empirical mapping methodology to correlate QoS metrics with QoE. We also describe two empirical studies conducted in a 3DTI system as simple, practical examples.

Mapping Methodology

We understand the conceptual causal relationship from QoS and QoE, but how are the individual metrics related? It is important to assess the mapping relations in a finer granularity to provide more useful design implications. In the QoS research, analytical frameworks have been developed to mathematically compute the correlations between QoS metrics. For example, Nahrstedt *et al.* [75] presented a QoS broker model, in which equations were developed to translate application QoS requirements to network QoS requirements, e.g., from sample loss rate to packet loss rate and from sample rate to traffic interarrival time. When we study the QoS-QoE mapping, however, such analytical methodology can hardly apply because of the gap between the subjective QoE metrics

and objective QoS metrics. As a result, methodology of empirical studies is developed. We describe three steps of the empirical methodology as follows.

1. Specify the metrics in each dimension of the QoS and QoE model. The selection of QoS metrics is application-specific, thus it is important for 3DTI designers to characterize and examine the application environment. The QoE metrics are much more general, and those captured in the framework (Figure 5.3) can be directly adopted.

2. Collect measurements of these metrics by conducting empirical experiments at the receiving side (Figure 1.2). For QoS metrics, a quantitative evaluation of the application performance is needed. For QoE metrics, the subjective experiential metrics such as concentration and enjoyment can be measured by collecting questionnaire responses using Likert scale, whereas the objective metrics such as the actual system usage and performance gains can be quantified by logging and calculating experimental outcomes. The QoS metrics can be tuned in the system to acquire different user responses under different circumstances. Furthermore, experiential experiments may be needed to observe exploratory behaviors of users. This may require the researchers to conduct field studies to record user behaviors while they use the applications.

3. Compute the correlations between measured pairs of QoS and QoE metrics, analyze measurements, and bind the resultant correlation values with their statistical significance. This step helps us understand which factors in QoS contribute to which factors in QoE and how large the individual contribution becomes. The QoS/QoE correlations and contribution factors can provide significant implications for application designers and guide them to make more educated decisions in the parameter tradeoffs according to QoE requirements of the users.

Subjectivity v.s. Objectivity. Cognitive scientists distinguish the actual physical environments from the cognitive environment [94], because an individual's total cognition is a function of her physical environment and her cognitive abilities. This indicates that subjective and objective measurements on the same metrics can lead to different results. Let us consider the subjective and objective metrics for interactivity. Subjective results can be obtained by having the users rank the *noticeability* or *disruptiveness* of delay as they perceive on a Likert scale. Objective measurements can be performed on the system to find out the actual latency in the unit of time. These results can differ greatly due to the user's visual perception ability. In the context of 3DTI systems , the cognitive abilities of users generally translate to the perceptive thresholds on different dimensions of sensory information (e.g., auditory, visual, haptic). The gap between one's cognitive environment and the physical environment should be taken into account in the evaluation of 3DTI systems. Next we present our

empirical studies to illustrate the application of the methodology in practice.

Task-Specific Experiments

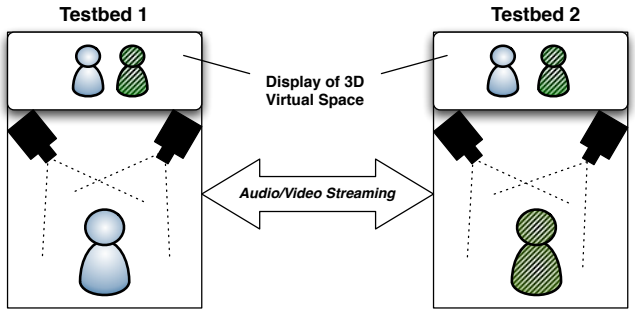


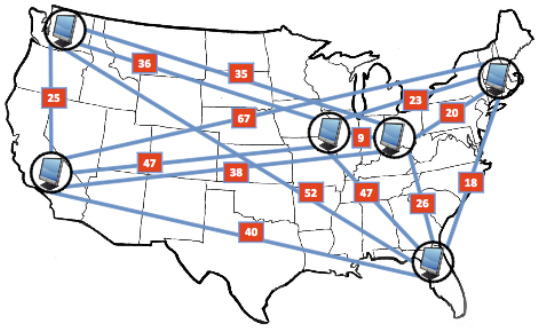
Figure 5.6: For the QoS-QoE experiment, we set up two separated 3DTI testbeds in the lab to simulate distributed environments. Each testbed contained a plasma display and two 3D camera clusters that were placed in a vertical axis to capture full human body. The 3D representations of users from two testbeds were merged into a joint virtual environment in real time for interaction.

Experimental Setup

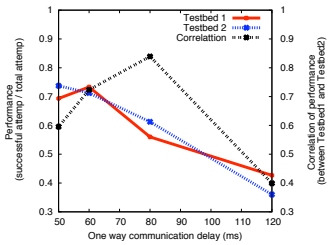
We recruited sixteen users to participate in four sets of task-specific experiments. Since the activities involved intense physical movement (e.g., rope jumping), college students (both undergraduate and graduate) were recruited, with nine female and seven male. We set up two separated 3DTI testbeds in the lab to simulate distributed environments. Each testbed contained a plasma display and two 3D camera clusters that were placed in a vertical axis to capture full human body. The 3D representations of users from two testbeds were merged into a joint virtual environment in real time for interaction. Figure 5.6 illustrates the setup.

As a simple example of the empirical methodology (Section 5.2.5), we identified metrics, collected data, and computed the correlations. The metrics listed in Figure 5.3 were used except 'exploratory behaviors' and 'technology adoption' which are hardly observed in the controlled studies. As a demonstrating example, we translated 'interactivity' to the QoS metric end-to-end delay, and 'vividness breadth' to the richness of communication channels (audio, video). The other used metrics are self-exploratory.

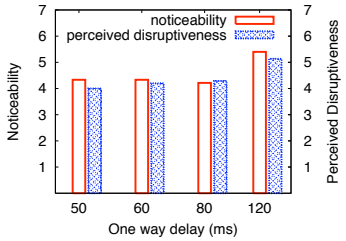
Both objective and subjective data were collected. For subjective measurements, a post-test questionnaire was filled up by each participant to answer descriptive questions on each metric. A Likert scale of seven points (1: strongly disagree, 7: strongly agree) was used for all questions. For objective measurements, we recorded the performance of users, where the ratio of successful attempts and completion time were mainly used as the metrics for 'performance gains'. With the collected data, we performed correlation tests between pairs of the QoS and QoE metrics, along with a statistical assessment of significance.



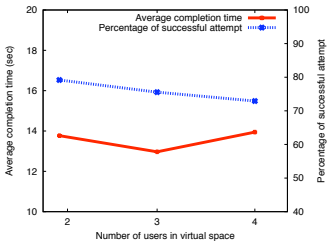
(a)



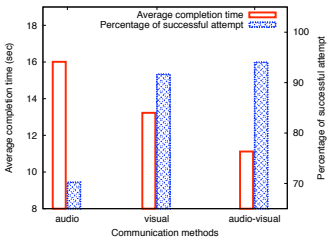
(b)



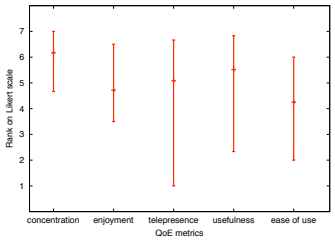
(c)



(d)



(e)



(f)

Figure 5.7: QoS-QoE experimental results - (a) Delays were measured on the Internet to determine the proper artificial delays introduced between the two testbeds, (b) interactivity (one-way delay) v.s. performance gains (successful attempts), (c) objective (one-way delay) v.s. subjective (noticeability/disruptiveness) interactivity, (d) vividness depth (crowdness in virtual space) v.s. performance gains (average completion time), (e) vividness breadth (presence of media channels) v.s. performance gains (average completion time), and (f) QoE rankings (min-avg-max).

Experiment 1

This experiment was to study the impact of QoS metrics 'interactivity' and 'consistency' on QoE metric 'performance gains'. We simulated the geographically distributed setup by artificially introducing one-way communication delays from Testbed2 to Testbed1 (remote delay) as well as inside the Testbed1 (local delay). Delays were set according to the real-world values we measured among the Internet2 nodes across the country. Figure 5.7(a) shows a small sample of measured delays (in ms) with respect to the geographical distances. We found that a majority of nodes had around 10ms, 30ms, and 70ms of one-way propagation delay between them, which became the values we used over the existent end-to-end delay (from capturing of a frame to rendering of it) of about 50ms in the system. To stress the system on interactivity and consistency, we chose the task of **rope jumping**. Specifically, one participant rotated a real rope in Testbed2, the other in Testbed1 tried to jump over it virtually by watching the display. With an increase of delay, displays might become inconsistent for the participants, with Testbed1 having the most delayed views. It was also expected that this would cause the participants to miss the jump over the rope as seen in the display of Testbed2. Nonetheless, to minimize the learning effect of users we randomized the injected delays for all experiments. Also, the users were not told the actual delay values until they finished the experiments and the questionnaires.

We gathered the number of successful jumps observed by the users in each testbed with varying artificial delay. The results are shown in Figure 5.7(b) and they represent the empirical mapping from the QoS metrics 'interactivity (speed)' and 'consistency' to the QoE metric 'performance gains'. As expected, the user performance, measured as the ratio of successful jumps, generally degraded with the increase of delays, as observed in both testbeds. The drop was about 50% from the delay of 80ms to 120ms. The correlation of the gathered data (refer to the right y axis) indicates the subjective consistency between the two testbeds. As shown in Figure 5.7(b), a high level of consistency existed up to 80ms delay (i.e., 30ms artificial) and reduced sharply when the delay reached 120ms (i.e., 70ms artificial delay which was measured between West Coast and East Coast in the U.S.).

Figure 5.7(c) shows the correlation between the subjective responses of users (on the notability and perceived disruptiveness of the delay) and the objective delay metric. When the end-to-end delay was below 80ms, it was hardly noticeable by the users. However, when increased to 120ms, it not only became perceptible, but also disruptive or distracting.

Experiment 2

The second experiment was to evaluate the impact of visual context quality on user experience. The parameter being varied was the spatial congestion of the virtual space, which mapped to the number of people in the virtual space. The task was a simple **charades game** between pairs of participants in two

testbeds. The participant in Testbed2 tried to guess the actions performed by the participant in Testbed1 without any conversations between them. The only communication channel was the joint visual virtual space. We limited the time of each guess to 30 seconds and recorded the timing of each successful guess. The experiment was done with 2, 3, and 4 users in the virtual space, which emulated the scenarios of distributed collaboration of 2, 3, 4 sites.

As shown in Figure 5.7(d), the user performance became little affected (percentage of successful attempts) by the crowd in the virtual user space. The percentage of successful attempts decreased less than 10% when we increased the number of users in the virtual space from 2 to 4. With the reasonable congestion in the virtual space (as the scale of collaboration grows), users could still perform fairly well without being too distracted.

Experiment 3

We designed the third experiment mainly to understand the effect of communication channel richness as an example of vividness breadth. The task was a **drawing game**. The participant in Testbed1 was given certain drawings of simple and complex shapes and signs. The participant in Testbed2 had to draw them correctly within 30 seconds each. There were three ways of communication tested: audio only, visual only, and audio-visual. The hypothesis was that increased context richness should lead to increased QoE.

We recorded the performance of participants, with the result shown in Figure 5.7(e). The average completion time to draw decreased significantly in the audio-visual setup. Also the number of successful attempts increased sharply in this case compared to the use of audio only. We also asked for user's rank to the question "It is important for me to have both audio and visual cues when interacting with a remote partner", and the average rank was 5.125 (positive) with a standard deviation of 1.78.

Other than the above experiments, participants were asked to perform different 3D interactions such as cyber-handshake, cyber-hug, and cyber-fight just to experience the system more. The other important QoE metric measurements are plotted in Figure 5.7(f), with the minimum, average, and maximum values shown. We observe that individual perceptions on the same environment were very different (e.g., the maximal difference is 6 on a 7-point scale), which indicates the importance of user customization in 3DTI systems .

Correlation Findings

We computed correlations between the measured QoS and subjective QoE metrics using the consolidated data of the three experiments, and performed a two-tail t-test on the correlation findings. The correlation graph is presented in Figure 5.8 showing only the links with strong significance ($p < 0.005$). The correlation value is labeled on each link.

There are several interesting observations. First, the measured correlations between interactivity and the presented metrics appear not strong. This is mainly due to the fact that we averaged the subjective responses for interac-

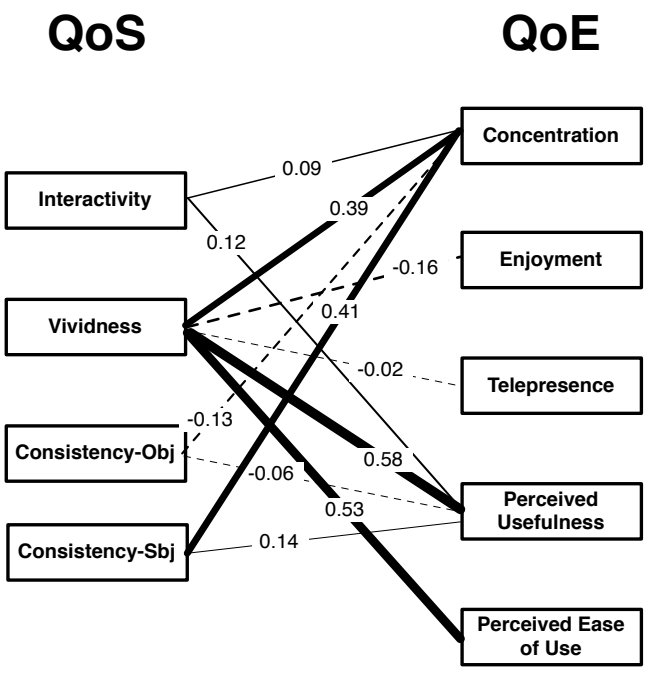


Figure 5.8: Correlations between QoS and QoE constructs - vividness (visual quality) has the highest correlation with three of QoE dimensions including concentration, perceived usefulness, and perceived ease of use.

tivity where the delay was not much noticeable in three of four cases. When the delay exceeds the perceptive threshold, we expect that users would lose sense of control (which corresponds to perceived ease of use), become distracted (less concentration), and feel lower degree of telepresence. Further quantitative studies on large interactivity delays need to be performed to confirm the hypotheses.

Second, the connection between vividness and several QoE metrics (concentration, perceived usefulness, and perceived ease of use) are among the strongest. The main reason is that the real-time 3D reconstruction algorithms in 3DTI systems are still challenging, so imperfections of images were present, including holes, flickering image, and spikes. This turned out to be the factors that affect users's QoE most in the system.

Third, we compare the results for objectively measured consistency (labeled 'Consistency-Obj') and subjectively rated consistency (labeled 'Consistency-Sbj'). The correlations results are very different, where only Consistency-Sbj has a strong correlation with 'Concentration'. Relating to the results shown in Figure 5.5, we find this connection very reasonable because perceived inconsistency led to focus distraction. The disagreement between the subjective and objective results is reminiscent of the theory that there is a gap between the actual environment and the cognitive environment (Section 5.2.3).

Experiential Experiments

Exploratory behaviors can hardly be measured in controlled studies with well-specified tasks. We present some of the results obtained in a field study with about twenty professional artists for over a year. There were no defined tasks, nor did we measure perceptions or performances. The main goal was to observe and record their exploration within the system.

The creativity of the artists led to a lot of interesting behaviors that surprised us. While the engineers were eager to improve on various QoS metrics (e.g., image quality, time synchronization, end-to-end delay), the artists desired retaining the imperfections of the system, with which they could make innovative improvisations (so called "glitch art"). As an example, due to a parameter configuration error the images of persons once became extremely 'spiky' with a lot of long triangular facets, but the artists requested the engineers to hold debugging, and went into the scenes, acting as if they had superpowers (e.g., with the effects of stretching arms) as seen in animation films. There was another time when the images of the upper and lower cameras became out of sync due to a software bug. While the engineers were trying to figure out the problem, one of the artists made an improvisation piece with her legs always moving seconds after the torso.

We cannot enumerate all exploratory behaviors that were observed, but they all raised interesting questions, enhancing our understanding about the measurements of user experience and its relationship with system performances.

5.2.6 Case Study of Non-Technical Factors

Experiment

The studies described in the previous section were mainly designed to examine the impact of technical factors on QoE. To best understand the impacts of non-technical factors on user experience, we conducted a large-scale case study, which took place during a public event organized in a university in the United States. The event last for two days, and was open and free to the public.

- *Setup and Equipments.* We set up two 3DTI sites in two separate places in a departmental building. Each was used by one gamer at a time. Gamers could not see or hear each other physically, and had to rely on telecommunication. Each TI site consisted of a 3D camera, a display, a black curtain and several host PCs. A Point Grey Bumblebee 3D camera was set up on a tripod with adjustable height. A black curtain was put up against the camera and behind the participant to facilitate background subtraction. A Philips 42" 3D-WOW-Display was used in each testbed to present the virtual world to the users. Wireless bluetooth headsets were provided for VoIP communication.

- *Game Design.* A lightsaber duel game was designed to take full advantage of remote interaction. Figure 1.1(b) illustrates how the game is played. Participant 1 in Site-1 puts on a lab coat with red patches and takes a green lightsaber, while Participant 2 in Site-2 puts on a coat with green patches and holds a red lightsaber. Each participant then tries to use the lightsaber to hit the opponent's color patches in the virtual space as much and as fast as possible in order to gain points. A collision detection module is developed in the renderer to automatically detect the hitting. Once a successful hit occurs, the sword and patches turn blueish for an electrifying effect; meanwhile, the game points, represented on a bar of the lightsaber's color, get updated. The lightsaber game is symmetric, meaning that the two players have identical goals and roles in the game.

- *Participants.* More than a hundred individuals with no prior experience with 3DTI systems have participated in the study. Fifty participants were able to complete the questionnaires. Among them, ten were female and forty were male. Six were below age 10, thirty-six were between 11 and 20, seven between 21 and 30, and one between 41 and 50. Over 80% of all participants were elementary and middle school children, indicating a possible target population for future 3DTI games. The participants were completely voluntary and not compensated for participation. item *Methods.* We adopt four different subjective measurement approaches to measure the QoS and QoE metrics: questionnaire, interviews, field notes and video taping. Questionnaires were distributed after the games. The questionnaire was designed to quantify the user experience using the QoS-QoE model, and the items therein were developed from the literature (Appendix). Two metrics, performance gains and technology adoption, were not used due to inapplicability. Each item was measured on a seven-point Likert scale, ranging from 1 (strongly disagree) to 7 (strongly agree). In addition to questionnaires, interviews helped to explore open observations from the participants and the viewers. Field notes helped us to find out observations from the researchers' points of view. Also, during public experiments, it is very common to miss users' valuable comments and facial expressions. Video taping (with permission) was thus used to gather such data for off-line analysis. Additionally, we developed a monitoring platform to continuously measure and record the real-world QoS metadata of the 3DTI system in uncontrolled environments. The QoS measurements were taken with time-stamps so that the QoS-QoE correlation could be studied.

Findings and Analyses

- *Age Influence.* We are interested in knowing how age impacts user experience. Figure 5.9 presents a comparison of QoS-QoE rankings by adults

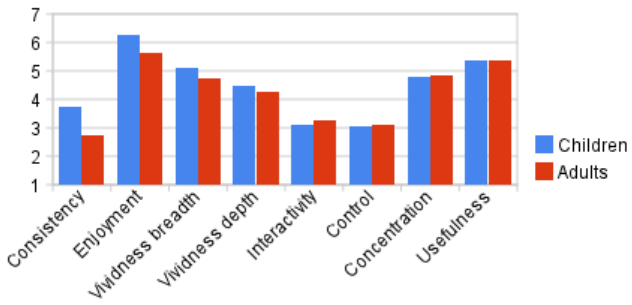


Figure 5.9: Subjective QoS-QoE comparison between adults and children (from left to right in descending order of difference)

(age 21 or above) and children (mostly below 15). The ranks were normalized on the scale of 1 (low/negative) to 7 (high/positive). From the results, we first note that the children are more tolerant of the inconsistency caused by network congestions with a rank 35.9% higher than adults. This is partly because that inconsistency became less noticeable as they moved their bodies much more vigorously than adults and got fully immersed into the playing. Second, we observe that the children found the interaction experience much more interesting than the adults with the "Enjoyment" rank 30.4% higher. The children lined up waiting for their turns, enthusiastically cheering for the players. During the game playing, they moved vigorously with excitement. Finally, adults were more concerned about technology adoption (a behavioral consequence in the QoE construct [107]). An adult participant commented "*This could be a great system for device manufacturing and trouble shooting*". On the other hand, the children were much more concerned about the telepresence and enjoyment. Some of their most frequent comments were *"I am the Jedi!"*, *"Does he get hurt?"*.

- *Social Aspect Influence.* The presence of friends or families significantly promotes engagement of children (both players and observers). Schoolchildren would excitedly acclaim their classmates for defeating the instructors. The players also expressed a much higher level of excitement when there were peers watching. Children enjoyed playing (competing) much more with their siblings, parents and close friends. *"Let's do this"*, *"Yeah! I'm stabbing her"*, *"Ha, I won again!"*, were some spontaneous comments that reflected the heightened engagement while playing with known peers. Most families and friends requested to play the games multiple times, which almost never occurred with players who were unknown to each other.

- *Physical Setup Influence.* Unlike traditional gaming systems, 3DTI systems require a more complicated physical setup (refer to Figure 1.3). We

find that the placement of different components significantly affects user experience.

- First, the relative positions and directions of cameras and displays in a TI environment impacts children's sense of orientation in the games. In 3DTI gaming, gamers have to use their full body to control the virtual photorealistic characters. This involves mapping from the physical 3D coordinate world to a virtual 3D coordinate world through the cameras' fields of view. Moreover, they need to orient themselves by locating or imaging where the remote peer would be in the physical surroundings. In our experiments, the camera and renderer were not aligned on the same plane so that the camera would not obstruct the users to see the screen. With such setup, we find that children had more difficulty orienting themselves than adults. Without being given any instruction, the adult users quickly figured out where to hit in the physical world in order to gain game points. Nevertheless, children tended to follow their experience with 2D game consoles and always acted toward the screens. Therefore, a more intuitive setup for children would be to have a zero angle between the front camera capturing direction and the screen displaying direction. The larger the angle, the more difficult it may be for children to find the correct orientation (refer to Figure 1.3).
- Second, the "sweet area" or activity space where the users can be visible to the cameras should be flexibly configured to accommodate different heights of people (e.g., in families) and different activity patterns. In the lightsaber game, for example, children needed much lower cameras to cover their whole body, and also a larger sweet area to enjoy the game, because they moved much more vigorously than adults and were more immersed into the game.

Although our QoS-QoE model provides us theoretical guidance on which QoS metrics may impact QoE, it focuses only on the technical environmental influences (e.g., response time, visual quality), and fails to account for the important application influences (e.g., game design), user influences (e.g., age, social aspects), and physical environment influences (e.g., device placement).

The 3D photo-realism and interactivity with remote peers with physical exertions are the most critical factors that stimulate people's interest in the new technology. In particular, children expressed tremendous excitement with the new gaming experience. Context vividness is less of a concern for them; rather, the 3D photorealistic telepresence in a virtual world with real opponents is the key factor.

In our experiments, we allowed people to communicate verbally over a VoIP module. Additionally, we developed a mobile interface for people to watch

the lightsaber duel and freely control their viewpoint from an iPhone. However, the audio and mobile components did not show strong contributions to improving user experience. This reflects the fact that QoS performances are not always as critical as engineers might expect for gaming experience; often times the non-technical aspects become decisive. Voice communication is useful for remote communication; however, it is recommended not to instrument the gamers even with wireless headphones; microphone arrays is a preferred option. Mobile interfaces provide flexible control options to manipulate the 3DTI cyber-environment. Nevertheless, the users experience greater enjoyment in the games if they can control the cyber-environment (i.e., their 3D photorealistic characters) with full body motion. The use of mobile devices should not interfere or obstruct the natural body movement of gaming.

Moreover, social aspects play an important role in the gaming experience. Due to psychological matureness, adults are not as interested in gaming as children, rather they are more concerned about the perceived usefulness of the technology. Children enjoyed the games much more when playing with their parents, siblings, classmates, or instructors. As noted by Schiesel [88], "Paradoxically, at a moment when technology allows designers to create ever more complex and realistic single-player fantasies, the growth in the now $18 billion gaming market is in simple, user-friendly experiences that families and friends can enjoy together." In 3DTI gaming, a great deal of social interaction is interleaved between the physical world and the virtual world. Our study shows that promoting such social interaction with games of moderate difficulty levels can significantly enhance the gaming experience.

Last but not least, unlike the existing gaming consoles, the 3DTI cyber-physical environment also poses new challenges in the physical setup of different components. The capturing and rendering devices should be carefully aligned to facilitate quick orientation of children in the cyber-physical space. Improper arrangement of the physical space can result in confusion, frustration, and loss of interest, hence demotes the gaming experience.

5.3 Conclusion

This chapter describes a significant step toward a general conceptual framework of QoE for multimedia applications. We construct a quality framework in the context of 3DTI systems with conceptual models of quality metrics for QoS and QoE. A methodology is presented to identify mappings between the two constructs, accompanied with empirical study examples. We also extend the framework by including influential non-technical factors as manifested in a case study.

6 Conclusion

6.1 Thesis Achievements

The past decade has witnessed the rapid growth of video-based telepresence environments. 3D tele-immersion, with its full-body, multi-angle 3D representations of users, emerge as one of the most promising telepresence technologies today, particularly for the support of physical activities such as sport training, dancing, and rehabilitation. However, the existing tele-immersive environments are crippled due to a huge demand for computing and networking resources that are needed to maintain the high interactivity (e.g., in end-to-end delay, video frame rate) and rich vividness (e.g., in video spatial resolution, depth accuracy) of the collaboration.

The main contribution of this thesis is to *improve qualities of 3D tele-immersive environments under stringent resource constraints*. To achieve this goal, we follow the human-centric principle and focus on those perceptually important qualities, i.e., interactivity and vividness, for the users. Our methodology is also human-centric in the sense that we leverage the semantics and constraints at the user level for the purpose of quality improvement. More specifically, we have developed solutions to address some of the most important problems in today's tele-immersive environments:

- **Intra-stream adaptation** (Chapter 3) - Temporal resource is known to be demanding for 3D tele-immersion, particularly in the end-to-end delay and refresh rate or video frame rate in interactive physical activities. As existing systems suffer from poor temporal performances, we improve them with a human-centric intra-stream approach. Specifically, we identify a critical factor that characterizes the spatial (including z-axial) resolution of tele-immersive video, and demonstrate that there are perceptual sensitives on this factor, i.e., a fairly generous degradation would go unnoticed due to the inherent limitations of the human visual system. Therefore, we leverage such limitations and develop a run-time adaptation mechanism to degrade spatial resolution and achieve interactivity improvement on various metrics such as frame rate and end-to-end delay. We demonstrate that such human-centric approach is particularly effective because it (for the first time) considers the limitations of human perception on 3D tele-immersive video and is able to reduce resource usage while improving the overall experiential quality of users.

- **Inter-stream adaptation** (Chapter 4) - In the previous chapter, we focus on the intra-stream (frame) level where unnecessary spatial details are excluded for interactivity improvement. Yet even with such reduction, today's Internet cannot support the high bandwidth demand for multi-source multi-stream tele-immersive environments. Congestion, as a result, causes packet loss, retransmission delay, and in turn incurs increased end-to-end delay, low and/or unstable frame rate, and flickering effect. We take a human-centric perspective in addressing these challenges. In this chapter, working seamlessly with the intra-stream approach in the last chapter, we evaluate an inter-stream methodology for data adaptation, where unimportant/unnecessary streams are excluded from the network dissemination. Our approach here is to exploit the user viewpoints in the collaboration environment, and prioritize video streams according to the contributions to the selected views. Accompanying the stream selection method is the topology construction algorithm that also takes into account the view interest (and the associated stream set) of different users as well as the network conditions on the Internet. Our experimental results show that we can largely reduce the bandwidth demand while maintaining important visual information. Indirectly, the interactivity of the environments is thus improved because of less congestion-related artifacts on the network.

- **Quality-of-Experience and Quality-of-Service** (Chapter 5) - In the first parts of the thesis, we have looked at various quality metrics, but an essential question remains - what does quality essentially in tele-immersion? Or more accurately, what qualities matter in tele-immersion? This research question motivates us to develop a classification of quality metrics that has direct impact on users in the interactive, video-based, tele-immersive environments. Importantly, these quality metrics are distinctly divided into Quality-of-Experience which represents user-level qualities and Quality-of-Service which captures the application-level qualities. We describe measurement of these quality metrics and also correlation methodology to identify and quantify the impact of Quality-of-Service on Quality-of-Experience. We believe this work offers a suite of metrics and methodologies to evaluate the qualities of tele-immersive environments and their users as well as pinpoint the specific qualities that need the most future improvement.

In this thesis, we have presented the first human-centric solutions that can provide better qualities for tele-immersive systems. We have implemented three of the most important approaches to improve different dimensions of qualities, and have demonstrated their effectiveness and benefits. We believe it becomes more and more clear that the human-centric computing model is important and useful for yielding the best and stable overall quality of 3D tele-immersive

systems under resource constraints.

6.2 Future Work

Our work focuses on video stream dissemination. As tele-immersive environments become more and more multi-sensory, including other sensory streams would be very interesting direction for future research. For example, how to explore the user interest in audio, haptic information and utilize them for more efficient data dissemination is challenging but well worth investigating.

We consider the aggregated impact of color and depth level-of-details. It would be interesting to separately investigate the perceptual effect of the two dimensions, and improve the adaptation algorithms to take into account the potentially different perception mechanisms on color and depth. Further, in this work we assume the variance threshold-based triangulation approach in the design and implementation of our adaptation schemes. Such adaptation for a different triangulation algorithm (e.g., not only based on intensity variance but also on depth) is definitely interesting direction for future research. Finally, we consider the case of two distributed sites in the collaboration. We improve the interactivity and vividness [97] of the 3DTI environment, but when multiple sites are present, the temporal consistency among all sites is a big challenge. For example, if a sending site has a frame rate of 10 fps, while another sending site has a frame rate of 5 fps, their data rendered at the same receiver site may look awkward and inconsistent. Extending our work to support such consistency is also a very interesting direction for future research.

We foresee many opportunities to apply the quality framework and mapping methodologies in the design and evaluation of 3DTI systems . Application practitioners can systematically find out the weighted contributions between quality metrics, thereby gaining a better understanding about the design choices on different QoS parameters. The presented framework also provides a conceptual basis for QoE specification where users can convey and evaluate their requirements.

References

[1] Cisco TelePresence, http://www.cisco.com/telepresence.

[2] http://www.its.bldrdoc.gov/vqeg/.

[3] Mapnet. *http://www.caida.org/tools/visualization/mapnet/Data/*.

[4] Teliris InterACT, http://www.teliris.com/telepresence-collaboration-tools.html.

[5] Subjective audiovisual quality assessment methods for multimedia applications, 1998. ITU-T Rec. P.911.

[6] Quality of experience: A strategic competitive advantage of microsoft unified communications. In *Whitepapers, Microsoft Inc.*, 2007.

[7] I. Ajzen. The theory of planned behavior. *Organizational behavior and human decision process*, 50(2):179–211, 1991.

[8] R. T. Apteker, J. A. Fisher, V. S. Kisimov, and H. Neishlos. Video acceptability and frame rate. *IEEE Multimedia*, 2(3):32–40, July 1995.

[9] K. J. Åström and R. M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[10] J. Bailenson, K. Patel, A. Nielsen, R. Bajscy, S.-H. Jung, and G. Kurillo. The effect of interactivity on learning physical actions in virtual reality. *Media Psychology*, 11(3):354–376, 2008.

[11] P. Bajcsy, K. McHenry, H.-J. Na, R. Malik, A. Spencer, S.-K. Lee, R. Kooper, and M. Frogley. Immersive environments for rehabilitation activities. In *ACM Multimedia*, pages 829–832, Beijing, China, 2009. ACM.

[12] H. Baker, N. Bhatti, D. Tanguay, I. Sobel, D. Gelb, M. Goss, W. Culbertson, and T. Malzbender. Understanding performance in coliseum, an immersive videoconferencing system. *ACM Transaction on Multimedia Computing, Communications, and Applications*, 1(2):190–210, 2005.

[13] H. H. Baker, N. Bhatti, D. Tanguay, I. Sobel, D. Gelb, M. E. Goss, W. B. Culbertson, and T. Malzbender. Understanding performance in Coliseum, an immersive videoconferencing system. *ACM Transaction on Multimedia Computing, Communications, and Applications*, 1(2):190–210, May 2005.

[14] R. Beauregard and P. Corriveau. User experience quality: A conceptual framework. In *Proceedings of the 1st international conference on Digital human modeling*, number 8 in ICDHM'07, pages 325–332, Berlin, Heidelberg, 2007. Springer-Verlag.

[15] A. Bharambe, J. Pang, and S. Seshan. Colyseus: a distributed architecture for online multiplayer games. In *Proceedings of the 3rd conference on Networked Systems Design & Implementation - Volume 3*, NSDI'06, pages 12–12, Berkeley, CA, USA, 2006. USENIX Association.

[16] A. BjÄorc. *Numerical Methods for Least Squares Problems*. SIAM: SIAM: Society for Industrial and Applied Mathematics, 1996.

[17] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in cooperative environments. *SIGOPS Operating Systems Review*, 37(5):298–313, October 2003.

[18] Y.-C. Chang, T. Carney, S. A. Klein, D. G. Messerschmitt, and A. Zakhor. Effects of temporal jitter on video quality: Assessment using psychophysical methods. In *Proceedings of the SPIE: Human Vision and Image Processing*, 1998.

[19] M. Chen. Design of a virtual auditorium. In *Proceedings of the ninth ACM international conference on Multimedia*, MULTIMEDIA '01, pages 19–28. ACM, 2001.

[20] M. Chesire, A. Wolman, G. M. Voelker, and H. M. Levy. Measurement and analysis of a streaming media workload. In *Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems - Volume 3*, USITS'01, pages 1–12. USENIX Association, 2001.

[21] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper and Row, 1990.

[22] M. Csikszentmihalyi. *Finding Flow: The Psychology of Optimal Experience*. Basic Books, 1997.

[23] Y. Cui and K. Nahrstedt. Layered peer-to-peer streaming. In *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, NOSSDAV '03, pages 162–171. ACM, 2003.

[24] R. Cutler, Y. Rui, A. Gupta, J. Cadiz, I. Tashev, L.-w. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: a meeting capture and broadcasting system. In *Proceedings of the tenth ACM international conference on Multimedia*, MULTIMEDIA '02, pages 503–512. ACM, 2002.

[25] M. J. G. Cynthia LeRouge and A. R. Hevner. Quality attributes in telemedicine video conferencing. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 6 - Volume 6*. IEEE, 2002.

[26] F. D. Davis. Perceived usefulness, ease of use, and usage of information technology: a replication. *MIS Quarterly*, 13(3):319–340, 1989.

[27] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw. User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, 35(8):982–1003, 1989.

[28] A. Dean and D. Voss. *Design and Analysis of Experiments*. Springer, 1998.

[29] A. Eichhorn, P. Ni, and R. Eg. Randomised pair comparison: an economic and robust method for audiovisual quality assessment. In *Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video*, NOSSDAV '10, pages 63–68. ACM, 2010.

[30] K. E. Finn, A. Sellen, S. Wilbur, K. Finn, A. J. Sellen, and S. B. Wilbur. *Video-mediated Communication*. Lawrence Erlbaum Associates, 1997.

[31] M. Fishbein and I. Ajzen. *Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research*. Addison-Wesley, 1975.

[32] M. Forte and G. Kurillo. Cyberarchaeology - experimenting with teleimmersive archaeology. In *16th International Conference on Virtual Systems and Multimedia*, VSMM '10, 2010.

[33] S. Fussell, R. Kraut, and J. Siegel. Coordination of communication: Effects of shared visual context on collaborative work. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, CSCW '00, pages 21–30. ACM, 2000.

[34] S. R. Fussell, L. D. Setlock, and R. E. Kraut. Effects of head-mounted and scene-oriented video systems on remote collaboration on physical tasks. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '03, pages 513–520, 2003.

[35] L. Gautier and C. Diot. Design and evaluation of MiMaze, a multi-player game on the internet. In *Proc. of IEEE Multimedia Systems Conference*, pages 233–236, 1998.

[36] G. Gescheider. *Psychophysics: The Fundamentals*. Psychology Press, 3rd edition, 1997.

[37] J. A. Ghani and S. P. Deshpande. Task characteristics and the experience of optimal flow in human-computer interaction. *The Journal of Psychology*, 128(4):381–391, 1994.

[38] R. Gifford. Environmental psychology. *Encyclopedia of Human Behavior*, 1999.

[39] M. Gross, S. Wurmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, and S. Lang. blue-c: A spatially immersive display and 3D video portal for telepresence. *ACM Transactions on Graphics*, 22(3):819–827, July 2003.

[40] S. hack Jung and R. Bajcsy. Learning physical activities in immersive virtual environments. In *IEEE ICVS*, 2006.

[41] S. Hacker. *MP3: The Definitive Guide*. O'Reilly, 1st edition, 2000.

[42] M. Hefeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava. PROMISE: peer-to-peer media streaming using collectcast. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 45–54. ACM, 2003.

[43] D. L. Hoffman and T. P. Novak. Marketing in hypermedia computer-mediated environments: Conceptual foundations. *Journal of Marketing*, 60(3):50–68, 1996.

[44] M. Hosseini and N. D. Georganas. Design of a multi-sender 3D videoconferencing application over an end system multicast protocol. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 480–489. ACM, 2003.

[45] I. Howard. *Seeing in Depth. Volume 2: Depth Perception.* I Porteous, 2002.

[46] HP Halo. http://hp.com/halo/.

[47] C.-L. Hsu and H.-P. Lu. Why do people play on-line games? An extended TAM with social influences and flow experience. *Information Management*, 41(7):853–868, 2004.

[48] S. Iai, T. Kurita, and N. Kitawaki. Quality requirements for multimedia communication services and terminals - interaction of speech and video delays. In *Proceedings of Global Telecommunications Conferenc*, GLOBECOM '03. IEEE, 1993.

[49] Y. Ito and S. Tasaka. Quantitative assessment of user-level QoS and its mappings. *IEEE Transaction on Multimedia*, 7(3):572–584, June 2005.

[50] ITU. Information technology - digital compression and coding of continuous-tone still images - requirements and guidelines. In *Recommendation T.81*, 1992.

[51] ITU. Methodology for the subjective assessment of the quality of television pictures. In *Recommendation BT.500*, 2009.

[52] C. M. Jackson1, S. Chow2, and R. A. Leitch. Toward an understanding of the behavioral intention to use an information system. *Decision Sciences*, 28(2):357–389, 2007.

[53] A. Jaimes, N. Sebe, and D. Gatica-Perez. Human-centered computing: a multimedia perspective. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA '06, pages 855–864. ACM, 2006.

[54] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, 1981.

[55] M. Jain and C. Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput. *IEEE/ACM Transactions on Networking*, 11(4):537–549, August 2002.

[56] R. Jain. Quality of experience. In *IEEE Multimedia*, volume 11, pages 95–96, 2004.

[57] Z. X. Jin, Y. J. Zhang, X. Wang, and T. Plocher. Evaluating the usability of an auto-stereoscopic display. In *Proceedings of the 12th international conference on Human-computer interaction: interaction platforms and techniques*, HCI '07, pages 605–614. Springer-Verlag, 2007.

[58] T. Kanade, P. Rander, and P. Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(1):43–54, 1997.

[59] N. Kelshikar, X. Zabulis, J. Mulligan, K. Daniilidis, V. Sawant, S. Sinha, T. Sparks, S. Larsen, H. Towles, K. Mayer-Patel, H. Fuchs, J. Urbanic, K. Benninger, R. Reddy, and G. Huntoon. Real-time terascale implementation of tele-immersion. In *Proceedings of the 2003 international conference on Computational science*, ICCS '03, pages 33–42. Springer-Verlag, 2003.

[60] N. Kelshikar, X. Zabulis, J. Mulligan, K. Daniilidis, V. Sawant, S. Sinha, T. Sparks, S. Larsen, H. Towles, K. Mayer-Patel, H. Fuchs, J. Urbanic, K. Benninger, R. Reddy, and G. Huntoon. Real-time terascale implementation of tele-immersion. In *Proceedings of the 2003 international conference on Computational science*, ICCS '03, pages 33–42. Springer-Verlag, 2003.

[61] J. K. Kies. *Empirical methods for evaluating video-mediated collaborative work*. PhD thesis, Virginia Polytechnic Institute and State University, 1997.

[62] M. Koufaris. Applying the technology acceptance model and flow theory to online consumer behavior. *Information Systems Research*, 13(2):205–223, 2002.

[63] R. Kraut, S. Fussell, and J. Siegel. Visual information as a conversational resource in collaborative physical tasks. In *Human-Computer Interaction 18 (2003)*, pages 13–49, 2003.

[64] S. Kum, K. Mayer-Patel, and H. Fuchs. Real-time compression for dynamic 3D environments. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 185–194, 2003.

[65] G. Kurillo, T. Koritnik, T. Bajd, and R. Bajcsy. Real-time 3D avatars for tele-rehabilitation in virtual reality. In *Medicine Meets Virtual Reality Conference*, MMVR18, 2011.

[66] G. Kurillo, R. Vasudevan, E. Lobaton, and R. Bajcsy. A framework for collaborative real-time 3D teleimmersion in a geographically distributed environment. In *Proceedings of IEEE International Symposium on Multimedia*, ISM '08. IEEE, 2008.

[67] M. Lambooij and W. IJsselsteijn. Visual discomfort and visual fatigue of stereoscopic displays: A review. In *Journal of Imaging Science and Technology*, volume 53, 2009.

[68] J.-S. Lee, F. de Simone, Z. Z. N. Ramzan, E. Kurutepe, J. O. T. Sikora, and T. Ebrahimi. Subjective evaluation of scalable video coding for content distribution. In *Proceedings of ACM international conference on Multimedia*, MULTIMEDIA '10, pages 65–72. ACM, 2010.

[69] R. Lehman and S. Conceicao. Thinking, feeling and creating presence in the online environment: A learner's viewpoint. In *World Conference on Educational Multimedia, Hypermedia & Telecommunications*, ED-MEDIA '11. AACE, 2011.

[70] J. Lien, G. Kurillo, and R. Bajcsy. Skeleton-based data compression for multi-camera tele-immersion system. In *Proceedings of International Symposium on Visual Computing*, ISVC '07, pages 714–723, 2007.

[71] A. Mehrabian and J. A. Russell. *An approach to environmental psychology.* MIT Press, 1980.

[72] S. Moebs. A learner is a learner, is a user, is a customer - QoS based, experience-aware adaptation. In *Proceedings of the international conference on Multimedia*, MULTIMEDIA '08, pages 259–268, 2008.

[73] D. C. Montgomery. *Design and Analysis of Experiments.* Wiley, 5th edition, 2000.

[74] F. Mueller, S. Agamanolis, and R. Picard. Exertion interfaces: sports over a distance for social bonding and fun. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '03, pages 561–568. ACM, 2003.

[75] K. Nahrstedt and J. M. Smith. The QoS broker. *IEEE Multimedia Magazine*, 2(1):53–67, 1994.

[76] P. J. Narayanan, P. W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, 1998.

[77] T. P. Novak, D. L. Hoffman, and Y.-F. Yung. Measuring the customer experience in online environments: A structural modeling approach. In *Marketing Science*, volume 19, pages 22–42. INFORMS, 2000.

[78] H. Nysveen, P. E. Pedersen, and H. Thorbjørnsen. Intentions to use mobile services: Antecedents and cross-service comparisons. *Journal of the Academy of Marketing Science*, 33(3):330–346, 205.

[79] D. E. Ott and K. Mayer-Patel. A mechanism for TCP-friendly transport-level protocol coordination. In *Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*, pages 147–159. USENIX Association, 2002.

[80] D. E. Ott and K. Mayer-Patel. An open architecture for transport-level protocol coordination in distributed multimedia applications. *ACM TOMCCAP*, 3(3), 2007.

[81] M. Pinson and S. Wolf. Comparing subjective video quality testing methodologies. In *Visual Communications and Image Processing*, volume 5150, pages 573–582. SPIE, 2003.

[82] Polycom Telepresence. http://www.polycom.com/telepresence.

[83] X. Qin. Delayed consistency model for distributed interactive systems with real-time continuous media. *Journal of Software*, 13(6), 2002.

[84] M. E. R. Dahlhaus. *Causality and graphical models for time series.* University Press, Oxford, 2003.

[85] S. Raghavan, G. Manimaran, C. Siva, and R. Murthy. A rearrangeable algorithm for the construction of delay-constrained dynamic multicast trees. *IEEE Transaction on Networking*, 7:514–529, 1999.

[86] G. K. R. B. Ramanarayan Vasudevan, Edgar Lobaton, T. Bernardin, B. Hamann, and K. Nahrstedt. A methodology for remote virtual interaction in teleimmersive environments. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, MMSys '10, pages 281–292. ACM, 2010.

[87] A. Ranjan, J. P. Birnholtz, and R. Balakrishnan. An exploratory analysis of partner action and camera control in a video-mediated collaborative task. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, CSCW '06, pages 403–412. ACM Press, 2006.

[88] S. Schiesel. In the list of top-selling games, clear evidence of a sea change. In *http://nyti.ms/9rD0Eq*, Accessed 2010.

[89] O. Schreer, N. Brandenburg, S. Askar, and E. Trucco. A virtual 3D video-conferencing system providing semiimmersive telepresence: A real-time solution in hardware and software. In *Proceedings of eBusiness and eWork*, pages 184–190, October 2001.

[90] R. M. Sheppard, M. Kamali, R. Rivas, M. Tamai, Z. Yang, W. Wu, and K. Nahrstedt. Advancing interactive collaborative mediums through tele-immersive dance (TED): a symbiotic creativity and design environment for art and computer science. In *Proceeding of the 16th ACM international conference on Multimedia*, MULTIMEDIA '08, pages 579–588. ACM Press, 2008.

[91] D. D. Silva, W. Fernando, G. Nur, E.Ekmekcioglu, and S. Worrall. 3D video assessment with just noticeable difference in depth evaluation. In *17th IEEE International Conference on Image Processing*, ICIP '10, 2010.

[92] D. J. Simons and R. A. Rensink. Change blindness: Past, present, and future. *Trends in Cognitive Sciences*, 9(1), 2005.

[93] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand. 3D video and free viewpoint video - technologies, applications and mpeg standards. In *Multimedia and Expo, IEEE International Conference on*, volume 0 of *ICME '06*, pages 2161–2164. IEEE Computer Society, 2006.

[94] D. Sperber and D. Wilson. *Relevance: Communication and Cognition*. Wiley-Blackwell; 2nd edition, 1996.

[95] E. Steinbach, S. Hirche, J. Kammerl, I. Vittorias, and R. Chaudhari. Haptic data compression and communication. *IEEE SPM*, 2011.

[96] R. Steinmetz. Human perception of jitter and media synchronation. *Selected Areas in Communications, IEEE Journal on*, 14(1):61–72, 1996.

[97] J. Steuer. Defining virtual reality: dimensions determining telepresence. *Journal of Communication*, 42:73–93, 1992.

[98] S. Tasaka, H. Yoshimi, A. Hirashima, and N. Toshiro. The effectiveness of a QoE-based video output scheme for audio-video ip transmission. In *Proceedings of ACM international conference on Multimedia*, MULTIMEDIA '08, pages 259–268. ACM, 2008.

[99] H. Towles, W. chao Chen, R. Yang, S. Kum, H. F. N. Kelshikar, J. Mulligan, K. Daniilidis, H. Fuchs, C. C. Hill, N. K. J. Mulligan, L. Holden, B. Zeleznik, A. Sadagic, and J. Lanier. 3D tele-collaboration over Internet2. In *International Workshop on Immersive Telepresence*, 2002.

[100] H. Towles, S. uok Kum, T. Sparks, S. Sinha, S. Larsen, and N. Beddes. Transport and rendering challenges for multi-stream 3D tele-immersion data. In *NSF Lake Tahoe Workshop on Collaborative Virtual Reality and Visualization*, CVRV '03, 2003.

[101] R. Vasudevan, G. Kurillo, E. Lobaton, T. Bernardin, O. Kreylos, R. Bajcsy, and K. Nahrstedt. High quality visualization for geographically distributed 3D teleimmersive applications. *IEEE Transaction on Multimedia*, 3(13):573–584, June 2011.

[102] V. Venkatesh, M. G. Morris, G. B. Davis, and F. D. Davis. User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3), September 2003.

[103] A. Vogel, B. Kerherve, G. von Bochmann, and J. Gecsei. Distributed multimedia and QoS: A survey. *IEEE Multimedia*, 2(2):10–19, 1995.

[104] Z. Wang and J. Crowcroft. QoS routing for supporting resource reservation. In *IEEE Journal on Selected areas in Communications*, 1996.

[105] S. Winkler and C. Faller. Perceived audiovisual quality of low-bitrate multimedia content. *IEEE Transaction on Multimedia*, 8(5):973–980, 2006.

[106] W. Wu, A. Arefin, Z. Huang, P. Agarwal, S. Shi, R. Rivas, and K. Nahrstedt. "I'm the Jedi!"- a case study of user experience in 3D tele-immersive gaming. In *Proceedings of IEEE International Symposium on Multimedia*, ISM '09, 2009.

[107] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang. Quality of experience in distributed interactive multimedia environments: Toward a theoretical framework. In *Proceedings of the international conference on Multimedia*, MULTIMEDIA '09, pages 481–490. ACM, 2009.

[108] W. Wu, Z. Yang, I. Gupta, and K. Nahrstedt. Towards multi-site collaboration in 3D tele-immersive environments. In *Proceedings of the 28th International Conference on Distributed Computing Systems*, ICDCS '08, pages 647–654. IEEE, 2008.

[109] W. Wu, Z. Yang, I. Gupta, and K. Nahrstedt. Towards multi-site collaboration in 3D tele-immersive environments. Technical Report UILU-ENG-2008-1726, University of Illinois at Urbana-Champaign, 2008.

[110] W. Wu, Z. Yang, and K. Nahrstedt. A study of visual context representation and control for remote sport learning tasks. In *World Conference on Educational Multimedia, Hypermedia & Telecommunications*, volume 2008 of *ED-MEDIA '08*, pages 1180–1189. AACE, 2008.

[111] S. Würmlin, E. Lamboray, and M. Gross. 3D video fragments: Dynamic point samples for real-time free-viewpoint video. Technical report, Institute of Scientific Computing, 2003.

[112] Z. Yang, Y. Cui, Z. Anwar, R. Bocchino, N. Kiyanclar, K. Nahrstedt, R. H. Campbell, and W. Yurcik. Real-time 3D video compression for tele-immersive environments. In *Proc. of SPIE/ACM Multimedia Computing and Networking*, MMCN '06, San Jose, CA, 2006.

[113] Z. Yang, W. Wu, K. Nahrstedt, G. Kurillo, and R. Bajcsy. Viewcast: View dissemination and management for multi-party 3D tele-immersive environments. In *Proceedings of ACM International Conference on Multimedia*, MULTIMEDIA '07, pages 882–891, 2007.

[114] Z. Yang, B. Yu, R. Diankov, W. Wu, and R. Bajcsy. Collaborative dancing in tele-immersive environment. In *Proceedings of ACM international conference on Multimedia*, 2006.

[115] Z. Yang, B. Yu, K. Nahrstedt, and R. Bajcsy. A multi-stream adaptation framework for bandwidth management in 3D tele-immersion. In *Proceedings of the international workshop on Network and operating systems support for digital audio and video*, NOSSDAV '06, pages 1–6, 2006.

[116] Z. Yang, B. Yu, W. Wu, R. Diankov, and R. Bajcsy. A study of collaborative dancing in tele-immersive environment. In *Proceedings of the 8th IEEE International Symposium on Multimedia*, ISM '06. IEEE, 2006.