

© 2016 by Chien-nan Chen. All rights reserved.

SEMANTICS-AWARE CONTENT DELIVERY FRAMEWORK
FOR 3D TELE-IMMERSION

BY
CHIEN-NAN CHEN

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, 2016

Urbana, Illinois

Doctoral Committee:

Professor Klara Nahrstedt, Chair & Director of Research

Professor Roy H. Campbell

Associate Professor Indranil Gupta

Associate Professor Cheng-Hsin Hsu, National Tsing Hua University

Abstract

3D Tele-immersion (3DTI) technology allows full-body, multimodal interaction among geographically dispersed users, which opens a variety of possibilities in cyber collaborative applications such as art performance, exergaming, and physical rehabilitation. However, with its great potential, the resource and quality demands of 3DTI rise inevitably, especially when some advanced applications target resource-limited computing environments with stringent scalability demands. Under these circumstances, the tradeoffs between 1) resource requirements, 2) content complexity, and 3) user satisfaction in delivery of 3DTI services are magnified.

In this dissertation, we argue that these tradeoffs of 3DTI systems are actually avoidable when the underlying delivery framework of 3DTI takes the *semantic information* into consideration. We introduce the concept of semantic information into 3DTI, which encompasses information about the three factors: environment, activity, and user role in 3DTI applications. With semantic information, 3DTI systems are able to 1) identify the characteristics of its computing environment to allocate computing power and bandwidth to delivery of prioritized contents, 2) pinpoint and discard the dispensable content in activity capturing according to properties of target application, and 3) differentiate contents by their contributions on fulfilling the objectives and expectation of user's role in the application so that the adaptation module can allocate resource budget accordingly. With these capabilities we can change the tradeoffs into synergy between resource requirements, content complexity, and user satisfaction.

We implement semantics-aware 3DTI systems to verify the performance gain on the three phases in 3DTI systems' delivery chain: capturing phase, dissemination phase, and receiving phase. By introducing semantics information to distinct 3DTI systems, the efficiency improvements brought by our semantics-aware content delivery framework are validated under different application requirements, different scalability bottlenecks, and different user and application models.

To sum up, in this dissertation we aim to change the tradeoff between requirements, complexity, and satisfaction in 3DTI services by exploiting the semantic information about the computing environment, the activity, and the user role upon the underlying delivery systems of 3DTI. The devised mechanisms will enhance the efficiency of 3DTI systems targeting on serving different purposes and 3DTI applications with different computation and scalability requirements.

To my dear family.

Acknowledgments

First and foremost, I would like to express my deep gratitude to my advisor, Professor Klara Nahrstedt, for giving me the opportunity to join the MONET family and the fascinating TEEVE project, for giving me the freedom to go for things that excited me, for the valuable discussions and comments on my publications, and for the non-stop support, belief, and encouragement. I will be forever grateful for having Klara as my advisor.

I also sincerely thank my other committee members, Professor Indranil Gupta, Professor Roy Campbell, and Professor Cheng-Hsin Hsu 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.

Many thanks to my colleagues in the tele-immersion project, particularly Zhenhuan Gao, Raoul Rivas, Ahsan Arefin, Pengye Xia, Aadhar Jain, and Sabrina Schulte for offering generous assistance in the development of my works. I would like to thank the present and past members of Multimedia Operating Systems and Networking research group, including Haiming Jin, Hongyang Li, Phuong Nguyen, and many others who have provided valuable suggestions and helps in the years. Sincere appreciation is extended to Lynette Lubben, Andrea Whitesell, and Mary Beth Kelly who helped me in administrative matters.

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 Professor Polly Huang at National Taiwan University. I am deeply grateful to her for 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. In particular, I would like to thank the AC Crew, the 304/307, Welly Chen, the Taiwanese Student Association, and many others, who made the life in the corn field fun and memorable.

I would like to express my earnest gratitude to my family for their support, without which none of my achievements would have been possible. I dedicate this thesis to my parents, my fiancé Charlotte, and my sister. 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 (NeTS-0520182), the Center for Integration of Medicine and Innovative Technology, Deutsche Telecom, Saburo Muroga Endowed Fellowship, and the Ministry of Education of Taiwan.

Table of Contents

List of Abbreviations	vii
1. Introduction	1
1.1 3D Tele-Immersion	1
1.2 Motivation.....	5
1.3 Challenges	6
1.4 Our Approach	8
1.5 Dissertation Contributions	12
2. Literature Review	15
2.1 Resource Adaptors in 3DTI.....	15
2.2 Scalability Demand in 3DTI	16
2.3 Quality Demand in 3DTI	16
2.4 Accessibility Demand in 3DTI	17
3. Semantics-Aware Content Delivery Framework.....	18
3.1 3DTI System 1: Activity-Aware Adaptive Capturing	19
3.2 3DTI System 2: Amphitheater	20
3.3 3DTI System 3: Cyber-Physiotherapy	21
4. A3C: Activity Semantics in Capturing Phase	23
4.1 Introduction	23
4.2 System Architecture	25
4.3 Activity Classification Module	26
4.4 Morphing-Based Frame Synthesis Module.....	30
4.5 Quality Demand Module.....	34
4.6 Evaluation	38
4.7 Conclusion	42
5. Amphitheater: User Semantics in Dissemination Phase	44

5.1	Introduction	44
5.2	System Model	46
5.3	User Model	48
5.4	Stream Delivery Model	52
5.5	Forest Construction	54
5.6	Forest Adaptation	58
5.7	Evaluation	62
5.8	Conclusion	68
6.	CyPhy: Activity and Environment Semantics in Receiving Phase	70
6.1	Introduction	70
6.2	Use Case Model	72
6.3	System Model	73
6.4	Content Archiving Feature	76
6.5	Content Streaming Feature	82
6.6	Experiment Settings	86
6.7	Evaluation	87
6.8	Conclusion	93
7.	Conclusion	95
7.1	Dissertation Achievements	95
7.2	Lesson Learned	97
7.3	Future Works	98
	References	100
	Appendix	110

List of Abbreviations

3DTI	3-Dimensional Tele-Immersion
A3C	Activity-Aware Adaptive Capturing
ACR-HR	Absolute Category Rating with Hidden Reference
AI	Artificial Intelligence
ANOVA	Analysis of Variance
API	Application Programing Interface
AQoS	Application Quality of Service
CDN	Content Distribution Network
CF	Contribution Factor
CPU	Central Processing Unit
CyPhy	Cyber Physiotherapy
DASH	Dynamic Adaptive Streaming over HTTP
DES	Discrete Event Simulator
EHR	Electrical Healthcare Record
EMG	Electromyography
FLANN	Fast Library for Approximate Nearest Neighbors
FPS	Frames per Second
GB	Gigabyte
GOP	Group of Pictures
GPU	Graphics Processing Unit
GW	Gateway
HP	Hierarchical Priority
HTTP	Hypertext Transfer Protocol
I/O	Input / Output
IoT	Internet of Things

IP	Internet Protocol
IPTV	Internet Protocol Television
JNGD	Just Noticeable Degradation
JSON	JavaScript Object Notation
JUADG	Just Unacceptable Degradation
LoM	Level of Motion
LTS	Long Term Support
MB	Megabyte
MBFS	Morphing-Based Frame Synthesis
MJPEG	Motion JPEG (Joint Photographic Experts Group)
MOS	Mean Opinion Score
MP3	MPEG (Moving Picture Experts Group) Audio Layer III
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
NLP	Natural Language Processing
P2P	Peer to Peer
PC	Personal Computer
PESQ	Perceptual Evaluation of Speech Quality
PSNR	Peak Signal to Noise Ratio
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random Access Memory
RB	Residual Bandwidth
RGB-D	Color (Red, Green, Blue) plus Depth
RMSE	Root Mean Square Error
RSF	Ratio of Synthesized Frames
SSIM	Structural Similarity Index

SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TFAN	Triangle Fan-based compression
TV	Television
VoD	Video on Demand
VoIP	Voice over Internet Protocol
VR	Virtual Reality

1. Introduction

1.1 3D Tele-Immersion

1.1.1 Background

We have come a long way to finally reach the alien technology in 1978: the time when Clark Kent finally saw his birth father's giant floating head in his *3D video message*. As we have witnessed since 1990s, from VoIP (voice over internet protocol) to VoD (video-on-demand) to video conferencing to telepresence, the media of information delivery over digital network becomes more and more immersive along with the development of wider-band connections, higher-resolution I/O devices, and cheaper storage entities. Yet, it is until the recent decade has the 3D visual communication finally begin to come into reality. 3D Tele-immersion (3DTI) technology allows full-body, multimodal interaction among geographically dispersed users, which opens a variety of possibilities in cyber collaborative applications. Individual user of a 3DTI site is captured by an array of 3D cameras surrounding her user space along with other application-specific sensors (Figure 1.1). The captured 3D models with 360 degree coverage of participating users are put into a shared virtual space. With the synchronization between actions of a user and her 3D model, the physical user spaces are synchronized with the shared virtual space, which creates the ultimate as-if-being-there immersion for all users (Figure 1.2).

The advent of 3DTI is a giant leap of improvement from 2D-visual-plus-audio-based multimedia delivery. One obvious reason is the third dimension brings the delivered content closer to our everyday experience in the three-dimensional world we live in. The more remarkable breakthrough, however, is its enhancement on users' ways to express and interact. While conventional 2D systems deliver mere verbal and visual contents, 3DTI enables kinetic and physical user interaction with its omni-directional motion capturing and synchronized virtual space. In other words, people are intrigued by 3D communication not because they only want to *talk* in 3D. It is the cyber-physical expression/interaction and the various possible applications/activities it could deliver in which we see its potentials. During the past years, impressive applications has been developed with many existing 3DTI prototypes such as remote education and training, hazard scene investigation, industrial and architectural cyber-collaboration, and remote healthcare assessment (Figure 1.2) [Sheppard 2008][Bajcsy 2009][Wu 2009][Sadagic 2013].

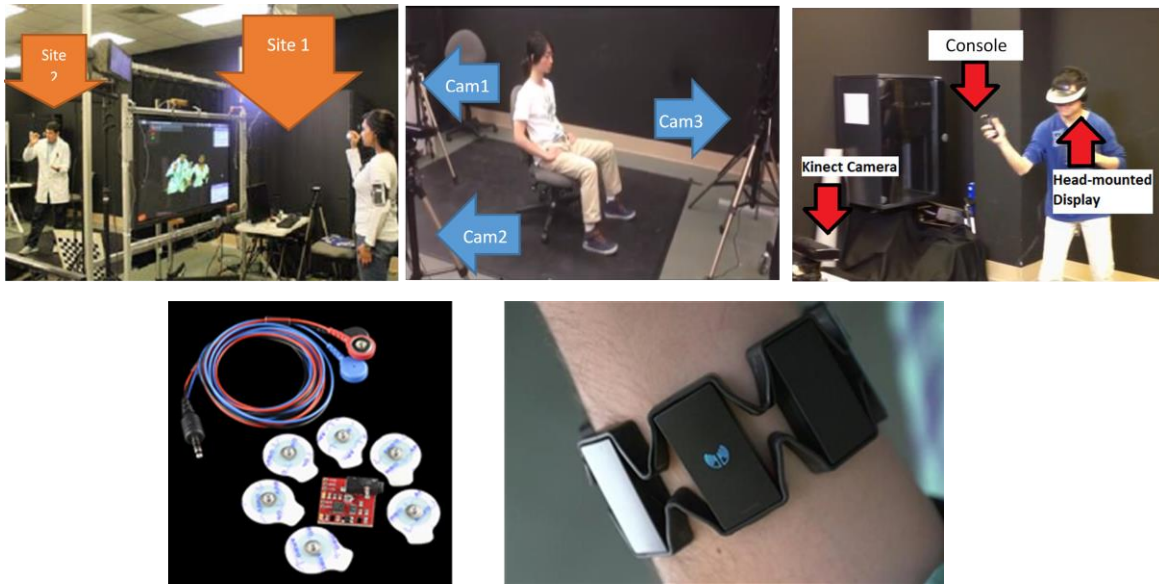


Figure 1.1: I/O device in 3DTI site.



Figure 1.2: Shared virtual space and various collaborative activities of 3DTI.

1.1.2 System Model

A complicated delivery chain of 3DTI content lies behind the media-enriched cyber collaborations to cope with real-time constraint of interaction, processing complexity of graphic rendering, and heavy load of data transmission. As illustrated in Figure 1.3, delivery chain of 3DTI can be broken down into three phases: capturing, dissemination, and receiving.

- **Capturing Phase.** A 3DTI application session is captured by 3D camera array, microphone, and other application-specific sensors. Action of user in an immersive site is captured and digitized into full-body, multi-modal stream bundles which contains heterogeneous data streams. The captured bundle is processed at a local (i.e., resides in the same 3DTI site) gateway machine, which handles the transmission of streams to other remote entities.
- **Dissemination Phase:** On dissemination of the captured content, streams captured by multiple 3DTI sites are exchanged and shared between all participating sites via the P2P (peer-to-peer) overlay network formed by all gateway machines. Based on the scale of user size, a central session manager may be coordinating the content dissemination tree structure by matching 3DTI sites as sender/receiver pairs in content sharing.
- **Receiving Phase:** Depending on different application types, the receiver of 3DTI content can be a 3DTI site, which aims for immediate playout on stream arrival; or it can be a storage entity, which archives the arrived streams, pending offline playback requests (from 3DTI sites). For content playout/playback, multiple rendering devices such as displays, speakers and other application-specific output devices are used in a 3DTI site. The gateway renders all 3D visual streams (i.e., 3D models of users) together and projects them into one virtual space (Figure 1.2).

1.1.3 User and Application Models

Users in a 3DTI application session can be categorized into two types: immersive (site) users and non-immersive (site) users. Immersive users are the content producers of the application session. Their actions are captured by their immersive sites and their 3D models are put into the shared virtual space. Non-immersive users are pure observers in the application session. They do not involve in the interaction in the virtual space so the non-immersive sites do not require capturing devices (e.g., camera array) to be installed.

Application model of 3DTI can be categorized into two types: synchronous 3DTI applications and asynchronous 3DTI applications. Synchronous application model allows

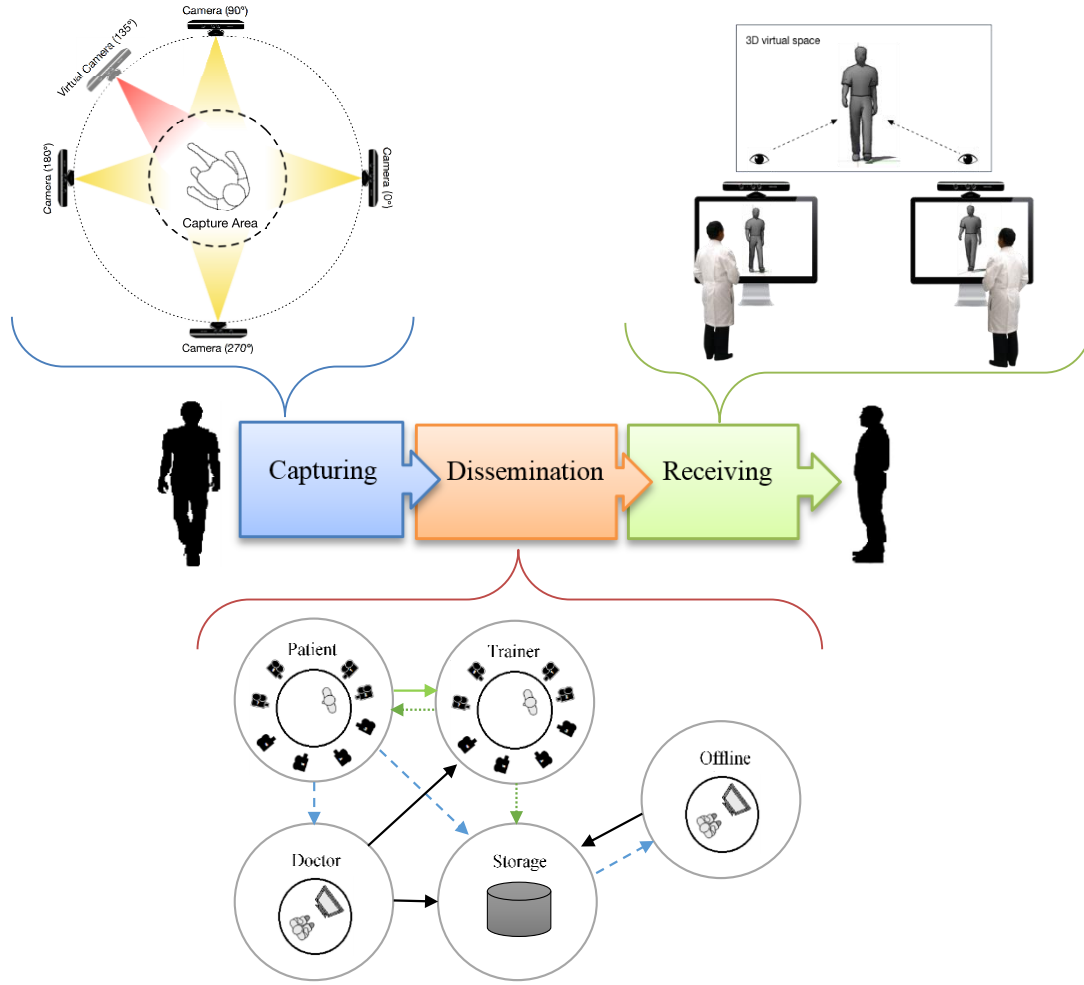


Figure 1.3: Delivery chain of 3DTI.

immersive users to interact with each other in the shared virtual space. If non-immersive users are involved in the application session, they can observe the interaction as it happens (e.g., live streaming). Asynchronous application model, on the other hand, provides on-demand services to users with a storage entity. The captured 3DTI contents from immersive users are uploaded to the storage entity. Later, other immersive users can stream the stored content from the storage and interact with the recorded 3D model in a virtual space. Non-immersive users can also stream the recordings for passive, offline viewing purpose.

Combining different users and application models enables 3DTI to carry applications of different goals and purposes. For interactive communication like conferencing, all participating users can be immersive under the synchronous application model. For live streaming applications like live exergame broadcasting, both immersive (players) and non-immersive (audience) users are involved synchronously. On-demand applications like

rebroadcasting of sport events involve only non-immersive subscribers viewing the recorded content under asynchronous application model.

1.2 Motivation

With the great potential of the immersive content delivered, the resource and quality demands of 3DTI rise inevitably, especially when more and more advanced applications target resource-limited computing environments (e.g., mobile or home environments) with stringent scalability requirements (e.g., multi-site interaction or large-scale broadcasting). Under these circumstances, the tradeoffs between: 1) resource requirements, 2) content complexity, and 3) user satisfaction in delivery of 3DTI content are magnified. Intuitively, when resource budget is low and content complexity is high, user's perceptual quality will be sacrificed. For example, watching live broadcasting 3DTI sport events on limited bandwidth budget incurs bad viewing experience. On the other hand, when the computing resource is scarce and the service quality is demanded, the system can only deliver low complexity activities. For example, a non-immersive 3DTI site running on a smart phone over 3G network will not support tele-medical applications.

In this dissertation, we argue that these tradeoffs of 3DTI systems are actually avoidable when the underlying delivery chain of 3DTI takes *semantic information* into consideration. We introduce the concept of semantic information into 3DTI, which encompasses information about the three factors: *computing environment*, *activity*, and *user role* in 3DTI systems. With semantic information, 3DTI systems are able to 1) identify the characteristics of its computing environment (e.g., resource budget) to allocate computing power and bandwidth to delivery of prioritized contents, 2) pinpoint and discard the dispensable, unnoticeable data in content capturing according to properties of target activities (e.g., motion level), and 3) differentiate contents by their contributions on fulfilling the objectives of user's so that the adaptation module can allocate available resource accordingly. With these capabilities we can change the tradeoffs into synergy between (resource) requirements, (content) complexity, and (user) satisfaction in 3DTI systems. For example:

- **In content capturing**, when delivering activity involving only little user movement (e.g., storytelling), the system could tune down the temporal resolution of video capturing to save extra bandwidth without degrading perceptual quality, and vice versa.

- **In content dissemination**, when the preferences of viewer towards each 3DTI performer is acquired by the system, streams can be prioritized on their delivery to reach maximum service satisfaction under bounded bandwidth.
- **In content receiving**, when targeting on mobile devices with limited power and display resolution, computation-intensive rendering functions should be automatically offloaded.

To sum up, in this dissertation, our goal is to bridge the gap between high-level semantics and low-level data delivery. We want to change the tradeoffs between requirements, complexity, and satisfaction in 3DTI services by exploiting the semantic information about environment, activity, and user role upon the underlying 3DTI systems. Thus, our dissertation statement is as follows:

Semantic gap between human-physical systems and multimedia cyber-systems will be bridged efficiently by injecting environmental and user-activity semantic information into the multimedia-cyber-systems.

By bridging the semantic gap, the efficiency of resource utilization will be improved, and hence will render 3DTI a feasible vehicle for carrying advanced applications that involve quality and scalability demands on devices with diverse computational capabilities.

1.3 Challenges

1.3.1 Capturing Phase and Its Challenges

Equipped with 3D camera array, microphone, and other application-specific sensors, a 3DTI site captures the activity of its user and digitizes it to enable full-body, free-viewpoint experience. However, as pointed out by many previous works [Yang 2010][Mekuria 2013][Xia 2013] on 3DTI capturing interfaces, bitrate of the captured content bundle (i.e., the aggregation of heterogeneous content streams) is oftentimes too high to be supported by common networking environments. An empirical calculation provided in [Yang 2010] reports a 300 Mbps bitrate for visual streams in 3DTI with minimum quality setting (320x240 resolution, 10 fps). A more modern hardware setting in interface proposed in [Mekuria 2013] introduces an even higher 1,032 Mbps bitrate with Kinect cameras (assuming 640x480 resolution, 30 fps) without compression. Yet, advanced applications of 3DTI such as event broadcasting [Arefin 2012], remote healthcare [Sonnenwald 2014] [Han 2015], and mobile communication [Shi 2012] all picture modest CPU/GPU and low bandwidth budget in their computing environments in order to enable them on home or

mobile devices. Thus, in the capturing phase of 3DTI, it is necessary to identify the dispensable details in the content bundle to reduce the total bitrate to a manageable level.

1.3.2 Dissemination Phase and Its Challenges

On dissemination of the captured content, stream bundles captured by multiple performer sites (i.e., immersive sites) are exchanged and shared between all viewer sites (i.e., all participating sites regardless of whether they are immersive or non-immersive) via P2P network. With the received content, a synchronous, shared virtual space is rendered locally in each site. In this process, the first challenge comes from 3DTI's multi-view characteristic. The multi-source (multi-performer), multi-content (multi-camera) dissemination becomes a content distribution forest construction problem in the P2P overlay network formed by all participating sites. Multiple viewers subscribing to multiple streams from multiple performers introduce massive bandwidth consumption, especially when the user number scales up. Platform for 3DTI broadcasting envisioned in [Arefin 2012] aims to serve 1,000 concurrent viewers. The system is estimated to consume up to 6 Gbps outbound bandwidth of the content sources. The second challenge is efficient delivery of the multi-view content. Streams produced by all performer sites are not equally important in a viewer's rendering. [Yang 2010] first identifies the relationship between a camera stream's shooting angle and its contribution to the field of view of a viewer. The work validates the importance of content differentiation in the dissemination phase. However, in multi-site settings [Wu 2008][Arefin 2012], Yang's dissemination only brings marginal (5%) improvement on scalability. Thus, in the dissemination phase of 3DTI, we need an efficient prioritization scheme to enable efficient utilization of the overlay network to create a manageable environment for large-scale applications.

1.3.3 Receiving Phase and Its Challenges

Depending on the application model, a 3DTI site may receive the content bundle for immediate display (for synchronous interactive applications) or for storage and later retrieval (for asynchronous on-demand viewing). For these different purposes, the challenges of receiving phase lie in 1) efficient storage, 2) review summary generation, and 3) adaptive streaming. On efficient storage, [Mekuria 2013] proposed a mesh-based compression scheme for 3DTI content which reached 1:10 compression ratio. However, this is still far from being comparable to video codecs for conventional 2D content (e.g., 1:100 for MPEG-1). On review summary generation, [Jain 2013] proposed a metadata analysis module in their 3DTI system for activity recognition. Yet, training and tuning phases of the module make it impractical for most applications in home environment. On adaptive rendering, previous 3DTI

clients are all designed for specific computing environments (mobile: [Shi 2012], PC: [Xia 2013]) and platforms (Windows: [Xia 2013], Linux: [Yang 2010], Android: [Shi 2012]). However, in modern use cases, many applications require the flexibility to run under different resource limitations (e.g., power and bandwidth). Thus, in the receiving phase of 3DTI, we need an adaptive receiving client and a database entity with efficient storage and retrieval features.

1.4 Our Approach

To tackle the identified challenges to enable higher scalability and to fulfill quality demand of advanced 3DTI applications, we introduce the concept of semantic information, which contains information about environment, activity, and user role. We argue that, a 3DTI system should be aware of the semantic information of these three factors throughout its delivery chain in order to avoid the tradeoff situations between resource requirements, content complexity, and user satisfaction. In the following, we first introduce the scope of the three factors and the definition of semantic information. Next, we provide an example to demonstrate the gain we will get from a semantics-aware content delivery framework.

1.4.1 Semantics-Aware Content Delivery Framework

Semantic information covers the information about environment, activity, and user role. Thus, we start from formalizing the scope of these three factors.

- **Environment:** The computing environment that a 3DTI application targets to utilize for delivering 3DTI contents. For example, a remote healthcare application may target home computing environment for its patient sites; whereas a performance broadcasting application will target studio with dedicated network infrastructure and computing devices for its performer sites.
- **Activity:** The expected activities that will happen in the physical user space of a 3DTI application. For example, in exergaming applications, fast-paced, gross-motor activities involving whole user body are expected; while in physical rehabilitation applications, user (patient) activity is more predictable and expected to be repetitive and focusing on particular (injured) body parts.
- **User role:** The objectives and expectation of a particular user towards the 3DTI application. For example, performers and the audience are two different user roles in a live broadcasting application. Due to the different roles, these different types of users will have different requirements, tolerances, and priorities to contents.

With the scopes formalized, we define semantic information as:

“The high level information about computing environment, user activity, and user role; from which we can infer the context of the scenario and feed it into adaptive system modules that will configure the content delivery chain to maximize system efficiency.”

Figure 1.4 depicts our semantics-aware content delivery framework. On the highest semantic level, we collect information about environment, activity, and user roles; on the lowest system level, there exist the tradeoff relationships between resource requirement, content complexity, and user satisfaction. Our goal is to fill up the gap in between by devising semantics-aware modules and include them into 3DTI systems of different application-level purposes. These awareness modules take the context inferred from various semantic information as input, and configure the underlining content delivery chain accordingly.

Semantic information of the environment, i.e., the *environment semantics*, is determined by the target environment of a 3DTI system (e.g., home, mobile, or dedicated facilities). From its computing environment, we can infer the connection type (e.g., 3G, home cable network, or dedicated optical connection), the computation budgets (e.g., CPU rate, RAM size) and the I/O interfaces (e.g., number of cameras and their resolutions) of a 3DTI system. This helps the underlying system adapts its content complexity to avoid wasting unnecessary resource, or offloads/un-offloads computation intensive functions to more powerful entities in the delivery chain (Chapter 6).

Semantic information of the activity, i.e., the *activity semantics*, is determined by the activity expected in the application (e.g., lecture, gaming, or physical exercise). From the expected activity, we can infer the range of movement (e.g., gross-motor body exercise or fine-motor handcraft), the level of motion (e.g., interactive action gaming or slow motion rehabilitation), and the focus of movement (e.g., facial expression in storytelling or injured body part in physical inspection). These properties of activities directly affect the bitrate, the motion vector, and the resolution of the captured content. Thus, with activity semantics, adaptive compression (Chapter 6) and content capturing (Chapter 4) becomes possible, which help us achieve efficient delivery.

Semantic information of the user role, i.e., the *user semantics*, is determined by the objectives and expectation of user towards the 3DTI system. From the user role, we can infer the priorities of heterogeneous streams (e.g., vital sensing streams to doctors, or audiovisual sensing streams to patients) and priorities of content sources (e.g., an audience user’s preference towards different performers). With this information, the controller entity in the

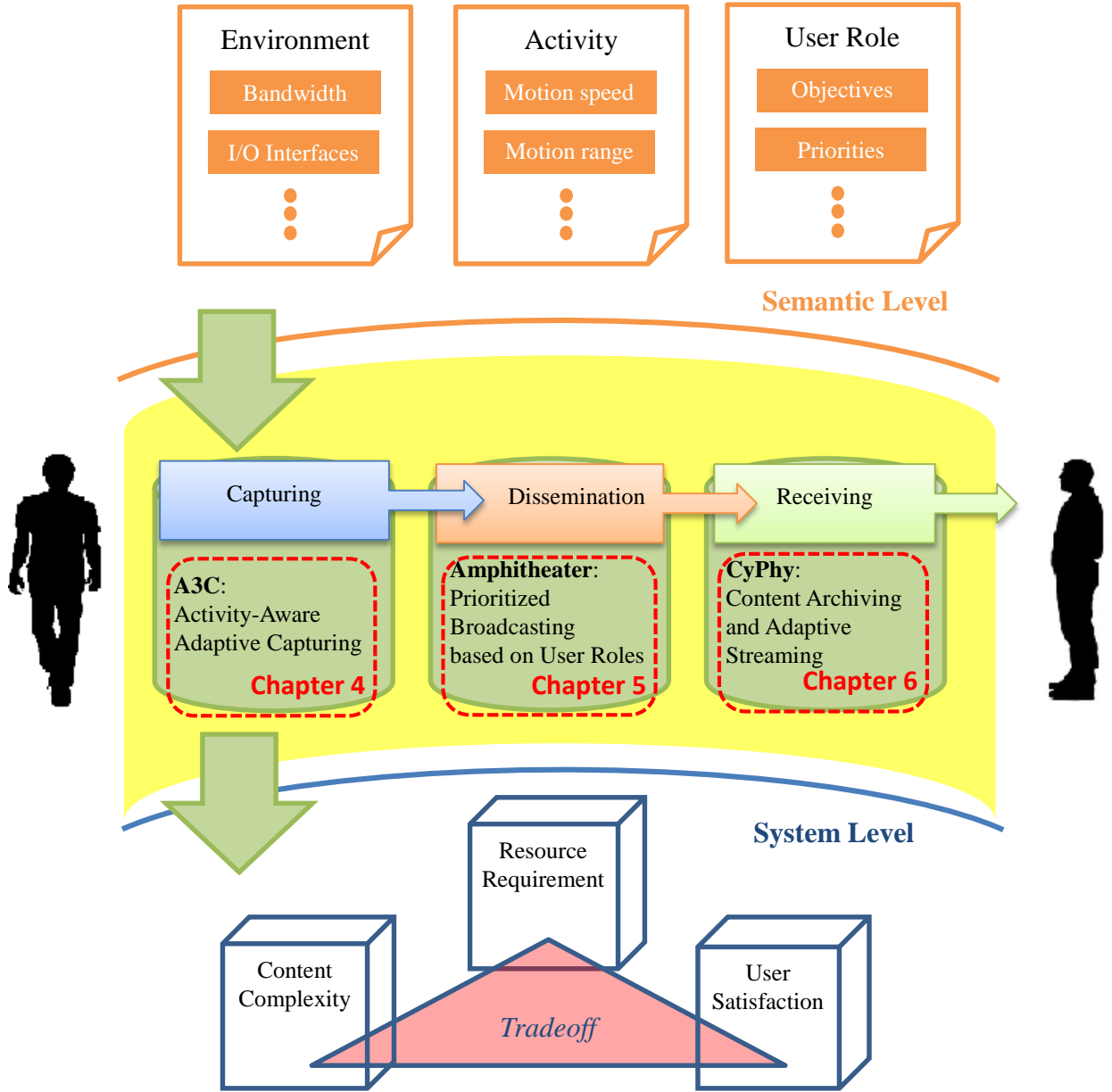


Figure 1.4: Semantic-aware content delivery framework.

3DTI system can construct a content dissemination network which allocates networking resources unevenly to tailor the semantic needs of different user roles without exceeding the available resource constraint (Chapter 5).

In this dissertation, we devise three 3DTI system instances (Chapter 3) following the semantics-aware content delivery framework. The three representative systems each has its own application-level purposes and requirements which emphasize the efficiency in different phases in the delivery chain of 3DTI content. Therefore, together they validate the necessity of semantics-awareness and demonstrate the scalability and quality improvements

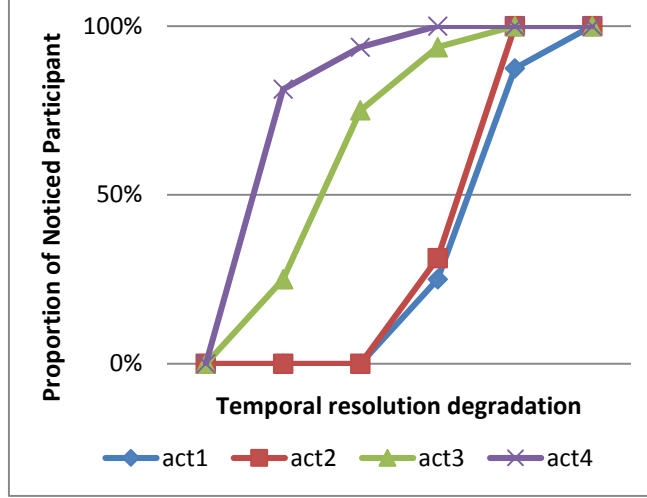


Figure 1.5: Noticeability of temporal resolution degradation of different activities.

we can gain from semantics-aware-modules in each delivery phase.

1.4.2 Example: QoE-Bitrate Balancing

Quality of Experience (QoE) is generally defined as the level of satisfaction of a user towards an application or a service [ITU 2008a][Wu 2009]. It is acquired by user study and interviews and is often time being quantified by mean opinion score (MOS), which is a subjective score rated by user in Likert scale. Intuitively, a system that renders high MOS is desirable yet expensive because the price of high QoE is usually high resource demand. In delivery of multimedia services, a naïve way to ensure high QoE is to ramp up the capturing rate (frame per second, i.e., FPS) of content. This incurs high temporal resolution, which brings high QoE; but also incurs high bitrate. While ramping up the bitrate to ensure QoE is feasible for low (bandwidth) cost services (e.g., VoIP), for 3DTI the regular demanded bitrate is already approximating the outbound bandwidth budget for most home computing environment. Therefore, this tradeoff between satisfaction (QoE) and resource (bitrate) restricts the application of 3DTI in previous works.

However, if we take semantic information into the picture, we find the relationship between QoE and bitrate is not invariant. According to our observations from user studies [Wu 2011][Schulte 2014], the *activity semantics* plays an important role in determining the relationship. Figure 1.5 shows the relationship between QoE and temporal resolution of four different activities. The x-axis is the degradation on temporal resolution (hence lower bitrate), and the y-axis is the percentage of users noticing the degradation (i.e., giving a decreased MOS). As we can see from the figure, users have different tolerance and sensibility towards different activities. After a closer look at these activities we find that the level of motion (LoM) of the target activity is a decisive element. Activity 1 and 2 (storytelling and lecturing)

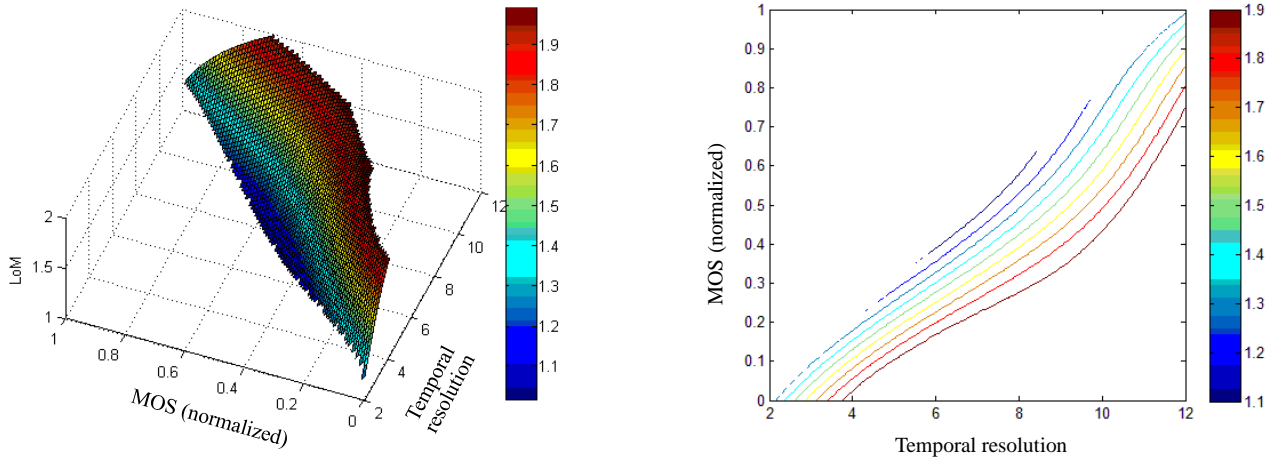


Figure 1.6: Three-dimensional model of QoE-Temporal resolution relationship.

involve very few body movements; whereas activity 3 (training exercise) involves slow gross-motor movements, and activity 4 (exergaming) involves face-paced, gross-motor movements with user moving rapidly in the physical space.

For 3DTI systems, detecting LoM in the user space is relatively easy comparing to a traditional 2D video studio setting. For special purpose systems, extra sensor in the user space can help the system understand the characteristic of current user activity. For instance, in a medical 3DTI system, vital sensors (heartbeat, sweat, respiratory) are used to indicate the rapidness of motion; in a gaming system, motion sensing consoles are attached to user to capture movements. For general purpose systems, user’s skeleton movement can be extracted from RGB-D (color plus depth) visual content captured by 3D camera array, which also contributes as a dependable hint at LoM.

Hence, with activity semantics taken into consideration, a three-dimensional semantic model which describes the relationship between QoE and bitrate can be complied, as shown in Figure 1.6. The colors in the figure indicates different LoM. The other two axes represent QoE (MOS) and bitrate (temporal quality), respectively. By including this activity-semantics-model into our adaptive compression module of the content capturing mechanism, we achieve 43%~87% bitrate reduction on content capturing. The processed content is evaluated by 92 human users by crowdsourcing evaluation engine. 72% of the participants cannot perceive the quality difference between the bitrate-reduced version and the original version [Chen 2013b].

1.5 Dissertation Contributions

The most important contribution of this dissertation is a holistic content delivery

framework that comprehends different semantics in multiple aspects (user, activity, and environment) and reflects the awareness in resource management and quality control to achieve higher service scalability and quality. Semantics information is addressed in the design, implementation, validation, and evaluation throughout the development of the 3DTI systems in this dissertation. By introducing semantics-awareness to distinct 3DTI systems, the performance gains are validated under different application requirements, different scalability bottlenecks, and different user and application models.

As modern multimedia services embrace more comprehensive multimodality, higher interactivity, and more flexible scalability, the underlying multimedia system becomes more and more complicated and resource-demanding with all type of sensors. We believe that the semantics-aware delivery framework anticipates the requirements of various advanced multimedia systems by providing a general, holistic solution on the balancing between content complexity, resource usage, and user satisfaction. Although in this dissertation we develop the semantic framework with concentration on 3DTI, we are convinced by its generality that the same framework is applicable to content delivery of other advanced multimedia applications such as multi-on-body-camera systems and ubiquitous sensing in intelligent homes.

The other contributions of this dissertation are as follows:

- To our best knowledge, we devise the most comprehensive semantic framework for 3DTI in term of the coverage phases in content delivery. Our semantics-aware modules exist in capturing, dissemination, and receiving phases of the underlying system, which result in a system-wise synergy towards resource efficiency.
- We are the first to consider semantic information in full-aspect with 3DTI systems. We include confound semantic factors regarding user (role, preferences, view), activity (range, speed, posture, position, number of participants, repetitiveness), and environment (network capability, computing capability, power limitation) into the design and development of our 3DTI systems.
- To our best knowledge, the semantics-aware content delivery framework is the first 3DTI framework that is general enough to encompass 3DTI applications with different requirements, demands, use cases, user models, and efficiency bottlenecks. The generality of the framework is validated by the diverse 3DTI system instances implemented following the purposed framework throughout this dissertation.
- We design and implement the first general purpose 3DTI capturing system which leverages the motion characteristics of user activity to mask the perceptual

degradation of 3DTI content. Our approach is able to adapt the temporal quality of 3D visual content in a way that reduces considerable amount of data while maintaining the same visual service quality. (Chapter 4)

- We devise the first 3DTI dissemination network construction and adaptation mechanisms that consider not only application-level semantics but also user-level semantics that address the user role and preferences. (Chapter 5)
- We improve the previous broadcasting platform of 3DTI content by introducing semantic awareness to the dissemination system. Both scalability and application level quality of service are substantially improved by our design. (Chapter 5)
- We design and implement the first total solution for 3DTI asynchronous telehealth application, which not only include basic store and play features, but encompasses recording, uploading, archiving, review recommendation, adaptive streaming, and workload offloading of the rehabilitation session. (Chapter 6)
- We design the first 3D video codec that exploits the activity semantics of rehabilitation exercise to enhance the compression efficiency. With more efficient archiving, the devised system can serve larger user scale with activity awareness. (Chapter 6)
- We are the first to implement DASH-compatible multimodal-plus-3D content streaming. This enables quality adaptation based on semantic information related to offline viewer's computing environment, which makes 3DTI a more feasible medium to be carried on inferior networking infrastructure and restricted computing device. (Chapter 6)

2. Literature Review

2.1 Resource Adaptors in 3DTI

2.1.1 Adaptors Based on Application Layer Semantics

Before 3DTI, adaptation schemes of multimedia services focused on individual streams. In other words, correlation and differentiation among streams were overseen. This implied waste of resources on delivering less important data to the application layer. In view of the problem, [Yang 2006a] and [Yang 2010] introduced application layer semantics to the design of adaptation scheme in 3DTI in order to achieve efficient content dissemination. However, the schemes introduced extra complexity to data sharing in the overlay network. Because every link in the overlay network was transmitting different parts of the stream bundle [Agarwal 2010] (a combination of streams containing different sensing data that are highly correlated), the topology of the network became a crucial factor which decides the efficiency of content delivery. A heuristic solution based on genetic algorithm are proposed in [Arefin 2013]. However, the computational complexity was inevitably raised with the adoption of the scheme.

2.1.2 Adaptor Based on Psychophysics

The effective end-to-end transport of delay-sensitive data has been long a subject of study in interactive multimedia services. The goal of [Huang 2012] was to provide human-centric adaptation on media playout scheduling in 3DTI. The authors investigated the mappings between Quality of Experience (QoE) and Quality of Service (QoS) metrics (e.g., end-to-end delay, PESQ, FPS) to find the suitable adaptation for gross-motor and fine-motion user activities. Different from our purpose, the work focused on resource allocation among streams rather than overall reduction of resource consumption. In [Wu 2011], the authors exploited the limits upon human visual system to balance between spatial resolution (the color-plus-depth level-of-details) and framerate. Because of physiological limitations, users were not able to tell the differences between certain levels of graphical degradation. In light of this nature, two QoE thresholds: Just Noticeable Degradation (JNDG) and Just Unacceptable Degradation (JUADG) were identified. With the two thresholds, the authors were able to adapt the resource consumption without degrading the service quality.

2.2 Scalability Demand in 3DTI

There are many IPTV frameworks such as [Zhang 2005] and [Gopalakrishnan 2011] that aimed for large scale dissemination of conventional 2D video streams. However, none of them considered 360 degree view dynamics with multi-source composition. Therefore, the challenges are different from ours. The 4D TeleCast framework which targeted a large scale content multicasting of 3DTI was proposed in [Arefin 2012]. In order to accommodate the hundred-scale audience group, Arefin et al. proposed to allocate the viewer sites into different classes (layers) with different delay service qualities. The higher the service class, the fewer hops there were between the viewer site and the source in the P2P network. The heavy burden of content dissemination not only came from bandwidth and delay requirements, but also because they tried to support the randomness of the free-viewpoint characteristic. Viewers in the hundred-scale audience could all have their distinct views. This made the effectiveness of content sharing very unstable in the dissemination network. Thus, the 4D TeleCast had to sacrifice part of the audience by giving them delayed content and requires an additional content distribution network to support the service. In [Yang 2010], a mesh topology was used to deliver the content from each producer site to one another. The number of participating site therefore became very restricted. Later, [Wu 2008] proposed to alleviate the workload via a randomized admission under a pub-sub model. Yet, the algorithm failed to consider the role of the users and focused only on differentiation at stream level.

2.3 Quality Demand in 3DTI

2.3.1 Telehealth Interfaces for Physiotherapy

In recent years, physiotherapy interfaces have been proposed in robotics and sensor research areas to provide inputs for telehealth. In [Dowling 2014], the authors developed a robotic rehabilitation system to treat musculoskeletal conditions. They captured EMG (electromyography) and skeleton signals of patient's arm movement and fed the captured data to a remote robot arm. The robot reproduced the movement of patient so a therapist could give prescriptions remotely. In [Gonzalez 2014], an interface combining Kinect camera and Wii balance board was proposed. It captured the center of mass position to determine patient's physical stability. Based on determined stability, the interface provided visual feedback to patients during their self-supervised rehabilitation. In [Han 2015] and [Kurillo 2014], the authors used Kinect camera to analysis the 3D reachable workspace of patients

with upper body conditions. The interface provided real-time visual feedback indicating range of motion and generated workspace analysis as 3D images which were sent to a therapist for diagnosis.

2.3.2 Cyber Collaborations through 3DTI

3DTI systems aim towards multi-purpose, multi-sites, and multimodality to enable a wide variety of user activities [Kurillo 2013][Schulte 2014][Fuchs 2014]. In [Sadagic 2013] and [Sonnenwald 2014], 3DTI is proposed to be the medium for training and simulation in critical/hazard domains like military training and emergency healthcare. Educational 3DTI application like archeology is also proposed in [Forte 2010] and [Xia 2013]. In [Chen 2014], 3DTI platform for performance broadcasting is proposed. The authors envision performer crew to be physically dispersed and interact remotely in the virtual world.

2.4 Accessibility Demand in 3DTI

2.4.1 3DTI Content Archiving

Due to the high bitrate of 3DTI, various archiving schemes for compression and content analysis were proposed. In [Chen 2013b], compression module based on frame synthesis was proposed to lower the bitrate of 3DTI systems. In [Chen 2013a], the module was paired with activity recognition to achieve dynamic bitrate adaptation. Other compression schemes for mesh-based 3DTI content were proposed in [Mekuria 2014] and [Mamou 2009], which concentrated on independent 3D image compression without inter-frame coding. Analysis on 3DTI data using metadata was proposed in [Jain 2013]. The authors achieved high activity detection accuracy via metadata analysis to avoid computationally expensive deep content analysis.

2.4.2 3DTI Content Delivery

Delivery of 3DTI content is not trivial due to its bandwidth consumption and real-time requirement. In [Chen 2015], the authors propose a prioritization scheme for 3DTI in bandwidth limited environment. Streams are prioritized based on their shooting angles and viewer's preferences. In [Hamza 2014], a DASH-based offline streaming for 3D streams is proposed. The authors develop adaptation mechanism based on quality balancing between two requested streams and allocate bandwidth accordingly to achieve a better quality of experience.

3. Semantics-Aware Content Delivery Framework

As we showed previously in Figure 1.4, the purpose of the semantics-aware content delivery framework is to bridge between the semantic level and the system level to solve the performance tradeoffs. Yet, for different 3DTI systems with different application purposes, the efficiency bottleneck exists in different phases in the delivery chain of 3DTI. Therefore, the solutions to these distinct bottlenecks require the framework to exploit semantic information of different, and in some cases more than one, semantic aspects.

In this chapter, we introduce three 3DTI systems (Figure 3.1) to demonstrate the mapping from our general framework to specific system bottlenecks, performance requirements, semantic aspects, and the tradeoff relationships between resource requirement, content complexity, and user satisfaction. The three representative systems each has its own application-level purposes and requirements which emphasize the efficiency in different phases in the delivery chain. Therefore, together they validate the necessity of semantics-awareness and demonstrate the scalability and quality improvements we can gain from semantics-aware-modules in each delivery phase.

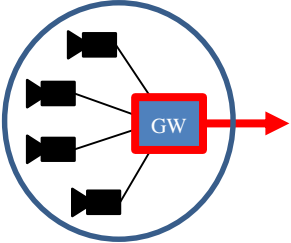
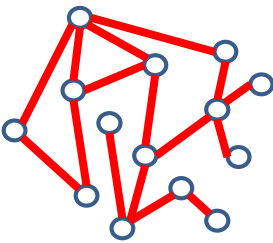
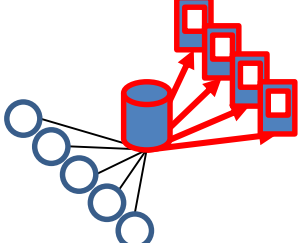
	A3C (Chapter 4)	Amphitheater (Chapter 5)	CyPhy (Chapter 6)
System Purpose	General activity capturing	Live broadcasting	On-demand streaming and Large-scale archiving
Efficiency Bottleneck	Capturing phase 	Dissemination phase 	Receiving phase 
Target Semantics	Activity semantics	User semantics	Activity and Environment semantics

Figure 3.1: The three 3DTI systems following the semantics-aware content delivery framework.

These three distinct 3DTI system and their semantics-aware designs are the core of this dissertation and will be detailed in proceeding Chapter 4, 5, and 6. Semantics information will be the core of the design, implementation, validation, and evaluation throughout the development of these 3DTI systems.

3.1 3DTI System 1: Activity-Aware Adaptive Capturing (A3C)

System purpose. The A3C (Activity-Aware Adaptive Capturing) system aims to provide immersive users a content compression mechanism for more efficient activity capturing. The system serves both synchronous and asynchronous application models and does not presume any activity type it serves. The goal of A3C is to reduce the captured stream bitrate to alleviate the outbound bandwidth requirement of gateway machine in an immersive site. With more efficient bandwidth usage on content capturing phase, 3DTI applications can utilize more limited computing environment as its immersive user site (e.g., home environment), and can acquire more sophisticated multi-modal capturing (i.e., more camera coverage and more application-specific sensors).

Efficiency Bottleneck. The bottleneck in content capturing phase is the outbound network interface of immersive sites. The outbound bandwidth is saturated by the multiple camera streams to support 3DTI’s multi-view feature. A 3D camera stream alone produces visual content in 10 Mbps scale. To alleviate the bandwidth consumption, previous 3DTI system can either 1) reduce the number of cameras, or 2) reduce the resolution of the cameras, but either method degrades the perceptual service quality.

Target semantics and awareness modules. From user studies in [Wu 2011] and [Schulte 2014], we see that 3DTI users have very different noticeability on quality degradation when participating in different user activities. Thus in A3C we target the activity semantics. As the core of A3C, we introduce an activity-aware adaptation module which consists of an activity classification model and a quality demand model. The first model exploits the enriched 3D visual content and application-specific motion sensing data to analyze the motion characteristics of current user activity. The second model, taking the detected activity from the first model as input, configures the compression ratio of our 3DTI compression mechanism on-the-fly. This design balances between bandwidth saving and perceptual quality based on activity semantics, which achieves substantial bandwidth saving without incurring noticeable quality degradation.

Tradeoff factors explored. To verify our design, we evaluate A3C via objective compression ratio as well as subjective user study. Objective evaluation focuses on

bandwidth savings (resource requirement factor) under different activity sessions with different motion characteristics (content complexity factor). Subjective evaluation is conducted by in-lab close-up interviews with synchronous application users and large-scale crowdsourcing feedback collected from asynchronous application users (user satisfaction factor).

3.2 3DTI System 2: Amphitheater

System purpose. Due to the high bandwidth demand in content exchange across 3DTI sites, previous applications of 3DTI are restricted with a small group of users [Yang 2010]. In order to promote 3DTI to applications with more flexible scalability, we devise the 3DTI Amphitheater: a live broadcasting platform which renders a shared virtual space that targets a hundred-scale user group. Users in the 3DTI Amphitheater are divided into two groups: immersive performers and non-immersive audience. Immersive performers interact on the virtual stage. Non-immersive audiences only passively observe the interaction among the performers.

Efficiency Bottleneck. The bottleneck in dissemination phase in such large-scale application is about the shared P2P overlay among users. The bandwidth depletion problem in the overlay for 3DTI is more severe than IPTV due to 1) the omni-directional view feature, and 2) distributed performers. When a user chooses to see a performer from an angle which no camera is shooting from, streams captured from more than one adjacent cameras need to be merged together to create the requested view. This implies that, unlike IPTV, users of 3DTI requests more than one camera streams from an immersive site. Also, they make the same request towards every site in the performer crew. On top of the bandwidth depletion, there is the synchronization issue across all received streams before they can be rendered together into one virtual scene. This can create delay to the responsiveness of user join and view changing requests.

Target semantics and awareness modules. We tackle the scalability challenge by 1) stream differentiation and 2) site differentiation based on user semantics. We argue that, to a particular viewer, not all streams are equally important. First, based on the user-chosen viewing position and direction, not all streams contribute equally to the requested view. We address this as the view-based priority of streams. Second, the importance of streams produced by a particular performer depends on her role in the application session. We address this as the role-based priority of streams. The two types of user-semantics-based stream priorities help the session manager differentiate the stream requests and plan the

dissemination network accordingly. Bandwidth in the overlay network is allocated to the dissemination of high priority stream requests. Low priority requests which have smaller effects to the viewer's service quality are rejected upon bandwidth limitation.

Tradeoff factors explored. We verify the performance of 3DTI Amphitheater by simulation with real-world network topology and the configurations of our 3DTI platform. The verification is two-fold. First, we evaluate the effectiveness of our semantic stream prioritization by examining the application quality of service (AQoS) of user sites (user satisfaction factor). Second, we investigate the bandwidth saving brought by content dissemination scheme in the amphitheater. We compare its bandwidth usage (resource requirement factor) and its sustainable performer crew size (content complexity factor) with previous multi-site/broadcasting 3DTI platforms [Nahrstedt 2011] [Arefin 2012] without semantic awareness.

3.3 3DTI System 3: Cyber-Physiotherapy (CyPhy)

System purpose. As a platform for investigating semantics-awareness in the receiving phase, we implement an asynchronous 3DTI system for physiotherapy session reviewing: the CyPhy (Cyber-Physiotherapy) system. The envisioned use case of CyPhy is as follows. As part of the prescription given in the end of at-clinic face-to-face meeting between therapist and patient, a "CyPhy kit" will be provided to the patient. The kit includes devices for the patient to set up a light-weighted 3DTI recording studio at home. On a daily basis, CyPhy will stream to the patient a pre-recorded exercise demonstration prescribed by the therapist. Patient will follow the video to conduct correct therapeutic exercises and have this rehabilitation session recorded with the CyPhy kit. The recorded session is upload to an electronic health record (EHR) cloud to be archived. Recorded sessions will be played out by the therapist whenever and wherever she is available on mobile devices or PC. Therapist can supervise the correctness of patient's moves by viewing the streamed content bundle and provide professional feedbacks.

Efficiency Bottleneck. At the system level, the application scenario involves two different content-receiving entities that bear the efficiency bottlenecks: 1) the EHR cloud, and 2) non-immersive therapist sites. Since the purpose of the EHR cloud is record archiving, its efficiency bottleneck is the large scale of incoming content bundle. Note that because the application model is asynchronous, the uploading of recorded contents do not come at once, so traffic congestion is not a challenging issue as in previous two systems. Instead, challenges in archiving come from 1) efficient storage, and 2) stored content analysis and

recommendation. As for non-immersive doctor sites, the purpose of content receiving is for immediate playback. Thus, the bottleneck is about 1) the network capability and 2) the 3D rendering capability of the receiving device.

Target semantics and awareness modules. On efficient archiving in the EHR cloud, we exploit the activity semantics of rehabilitation exercise to customize a compression scheme for CyPhy. The likeliness of everyday rehabilitation activities is exploited to enable inter-3D-video coding. Furthermore, the inter-coded result provides metadata that helps the system identify anomalous user activities in the stored 3DTI content. This in turn aids the auto summarization and review recommendation for offline viewers (e.g., therapists). On efficient non-immersive site streaming, we devise offloading mechanism based on environment semantics. Depending on both networking and computation capability of the computing environment, CyPhy adapts its content complexity by 1) adjusting content quality based on DASH [ISO 2014] standard, 2) offloading computationally expensive rendering to the server side, and 3) providing multiple view changing features with different levels of elaborations.

Tradeoff factors explored. With CyPhy, we focus on resource requirement factor and user satisfaction factor. Since CyPhy targets a very specific 3DTI service, content complexity factor is rather constant under its rehabilitation scenario. We generate a series of rehabilitation session recordings to evaluate the compression ratio (resource requirement factor) and the resulting visual quality (user satisfaction factor). We further compare the compression result with 3D video codec which has no activity awareness to verify CyPhy's performance gain from semantic information. For adaptive asynchronous streaming, the environment-semantics-aware modules: DASH-based streaming and renderer offloading, are tested under varying available bandwidth and computing power limitations (resource requirement factor) to evaluate the smoothness of content playback under constant user view change (user satisfaction factor).

4. A3C: Activity Semantics in Capturing Phase

4.1 Introduction

While most commercial 3D capturing systems are specialized for sole purpose (e.g., Kinect/Xbox for gaming), the development of 3DTI platforms is aiming towards multi-purpose [Arefin 2013][Chen 2013c], multi-sites [Wu 2008], and multi-modal [Huang 2011] in order to enable a variety of user activities including e-learning [Vasudevan 2011], remote therapy [Nahrstedt 2012], collaborative art performance [Sheppard 2008], and interactive gaming [Wu 2010]. When these applications become more and more demanding on quality and multimodality of content captured, the bandwidth consumption of the total content bundle, i.e., the aggregation of all homogeneous (e.g., cameras streams) and heterogeneous (e.g., audio and motion sensing streams) contents, inevitably rises. This rising demand on outbound bandwidth makes computing environments with moderate networking capability (e.g., regular home environment) infeasible to support immersive 3DTI site which handles the capturing of content. Hence, use case of advanced 3DTI applications is hindered by this bandwidth restriction.

In view of this outbound bandwidth bottleneck in the capturing phase of 3DTI's content delivery, we investigate deeper into the relationship between resource usage, activity semantics, and the quality of experience (QoE). We build an efficient Activity-Aware Adaptive Capturing (A3C) system which reduces the bandwidth consumption of immersive sites without incurring perceptible quality degradation by exploiting the activity semantics. A3C focuses on bitrate reduction of visual content in its capturing since visual streams (i.e., streams captured by the 3D camera array) contribute to most of the bitrate (more than 90%) of the content bundle. The basic idea of A3C is illustrated in Figure 4.1. In the content-capturing 3DTI site, A3C selects certain number of frames based on motion characteristics of the current user activity. These selected frames are removed from the visual stream before the stream is sent out through the network interface of capturing site. Thus, the outbound bandwidth demand can be reduced. When the content-receiving 3DTI site receives the visual stream, it amends the removed frames via our Morphing-Based Frame Synthesis (MBFS) technique to restore the viewing quality.

Challenges behind the workflow of A3C depicted in Figure 4.1 is three-folds. First,

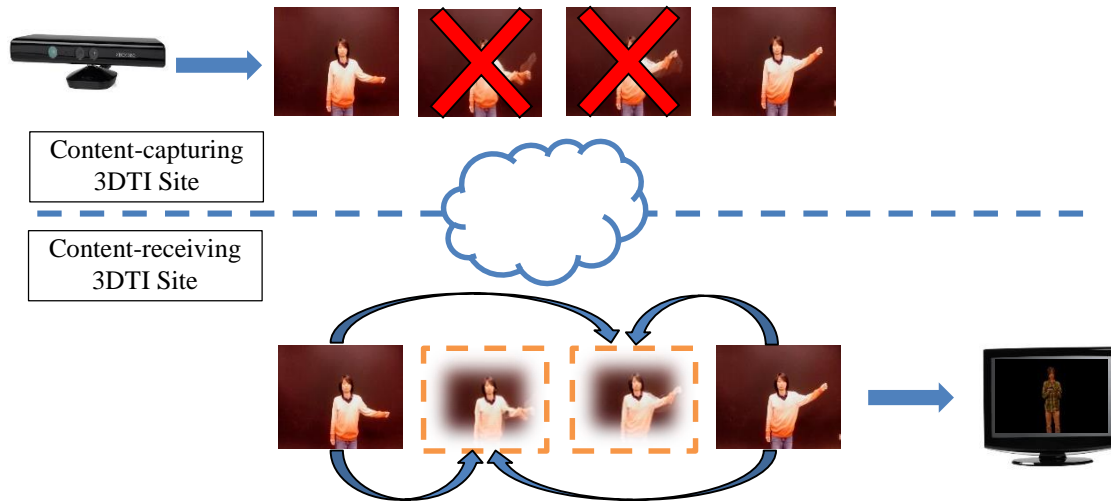


Figure 4.1: Basic idea behind A3C.

A3C system has to be activity-semantics-aware in the sense that it needs to be able to classify current activity in the user space from sensing data acquired by the input interface (e.g., on-body accelerometers) of the 3DTI application. To achieve this, A3C requires a real-time *activity classification model* which takes motion sensing data as input to classify current activity into pre-defined activity types on-the-fly. Second, after acquiring the activity type, A3C needs to utilize this information to select the number of frames to be removed from the captured stream. While removing more frames incurs more reduction on the stream bitrate, it also compromises the visual quality of the stream since the synthesized frames (created in frame amendment at the receiving 3DTI site) might have inferior quality. Thus, A3C needs a *quality demand model* which balances the bitrate reduction and the quality degradation according to user's sensibility and tolerance in current activity. Last, in the content-receiving site of A3C system, the frame rate of the received stream needs to be restored by amending the removed frames. This frame amendment process is done by our *MBFS module*, which exploits properties of regular 3DTI visual frames to enable restoration of missing frames via graphical morphing.

Mapping to semantics-aware framework. As a 3DTI system targeting on general activity capturing, A3C follows our semantics-aware content delivery framework to exploit activity semantics to solve the efficiency bottleneck in the content capturing phase (Figure 4.2 and Figure 1.4). Semantics-awareness modules of A3C include the activity classification module, the quality demand module, and the frame amendment module. These components in A3C will be detailed in later sections respectively followed by a series of subjective evaluation experiments involving real human users participating in 3DTI activities with

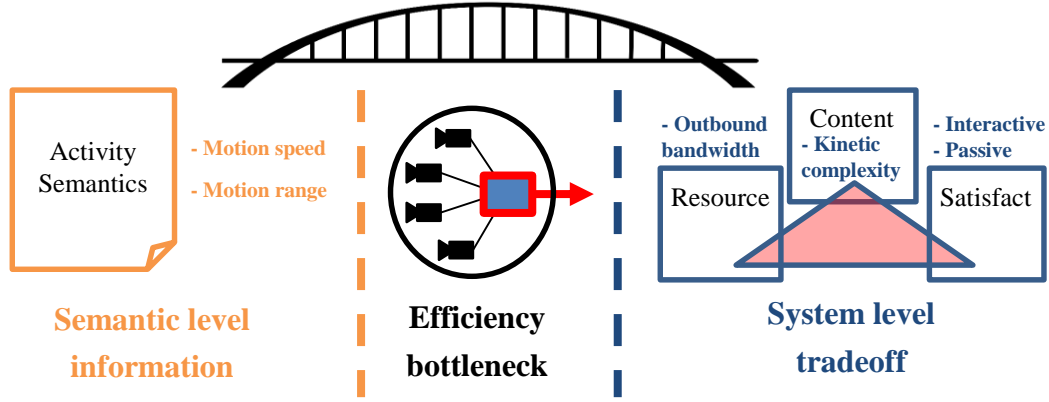


Figure 4.2: Mapping from semantics-aware content delivery framework to A3C.

different motion levels and application/user models. The effectiveness of semantic-awareness is validated by the experiment results on its improvement on system-level tradeoffs between 1) bandwidth savings of the outbound network interface (resource requirement factor), 2) kinetic complexity of visual content (content complexity factor), and 3) service quality of synchronous/asynchronous applications (user satisfaction factor).

4.2 System Architecture

The workflow of A3C's stream processing can be broken down into three steps: activity classification, frame removing, and frame amendment (Figure 4.3). The first two steps are handled by activity classification module and quality demand module in the content-capturing 3DTI site, respectively; and the last step is handled by the MBFS module in the content-receiving 3DTI site.

The first two modules in the capturing site do not handle the visual content stream directly. Instead, the activity classification module take motion sensing data as input, extracts motion features from it, and feeds them into a classification model based on SVM (support vector machine). The output of SVM is the activity type of current user activity. This information is passed on to the second quality demand module. The core of the quality demand module is a mapping model from activity type to the suitable frame removing ratio. For example, frame removing ratio of 1:2 means A3C will remove one in every two consecutive frames in the stream. This mapping is built upon subjective experiments on users' noticeability and tolerance. Before the visual stream passes through the outbound network interface, frames are removed according to the ratio decided by the quality demand module to reduce the bitrate. As the modified visual stream arrived at the receiving 3DTI site, it is processed by the MBFS module. The module uses morphing to synthesize the removed frames and brings back the frame rate of the stream to its original frame capturing rate. This

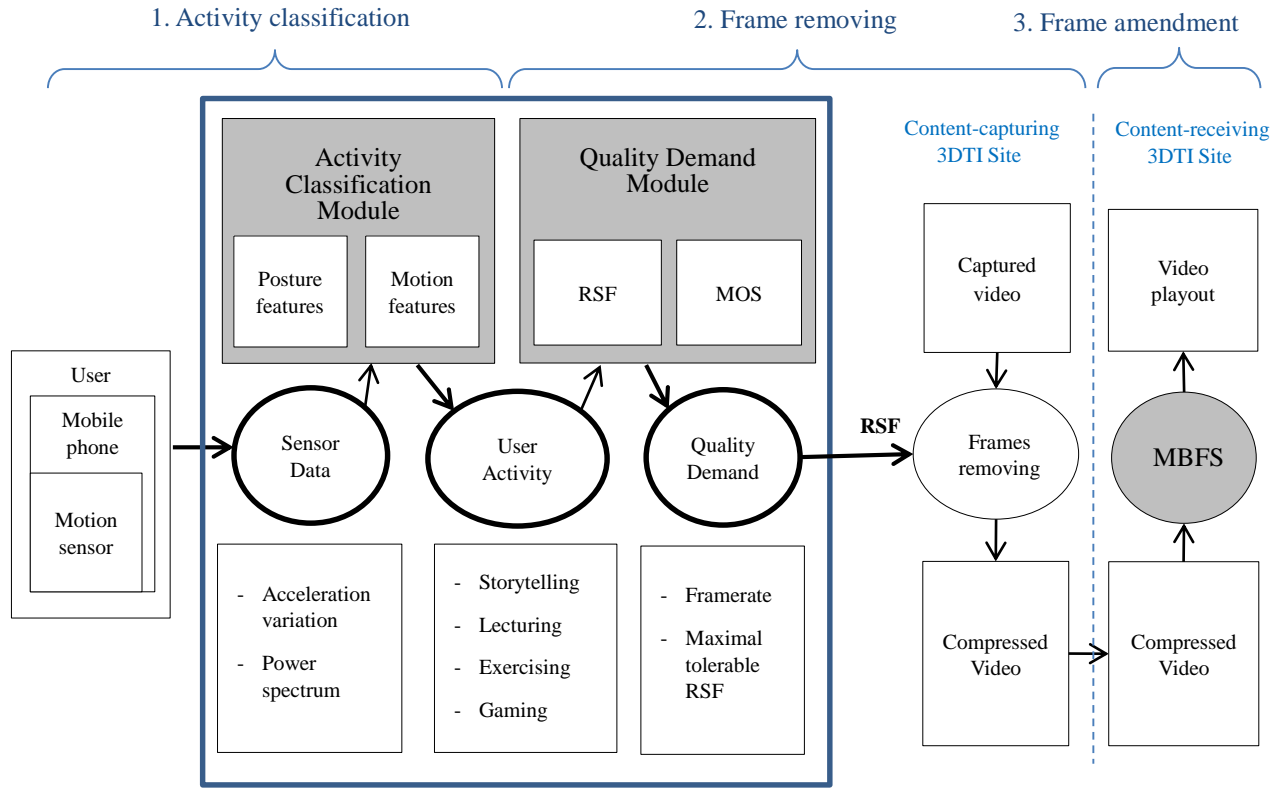


Figure 4.3: Architecture of A3C.

final process restores the smoothness of playback quality of 3DTI applications. In the rest of this chapter, we introduce the design and development of activity classification module and MBFS module first, and then we introduce the quality demand module.

4.3 Activity Classification Module

In this section, we introduce the activity classification module. First, we introduce the set of activity types we focus on and their motion characteristics. Second, we introduce the input sensing data that help our module discern the activities. Last, we introduce the SVM-based classification model that predicts the activity.

4.3.1 Activity Types

Common user activities in 3DTI environment include e-lectures, exercise training [Vasudevan 2011], and action gaming [Wu 2010]. Each of the activities has its motional and postural uniqueness. Thus, by monitoring the readings of accelerometer attached to user, the activities can be classified in real-time with a machine learning approach.

The activity types we are targeting and their motional/postural signatures are listed as

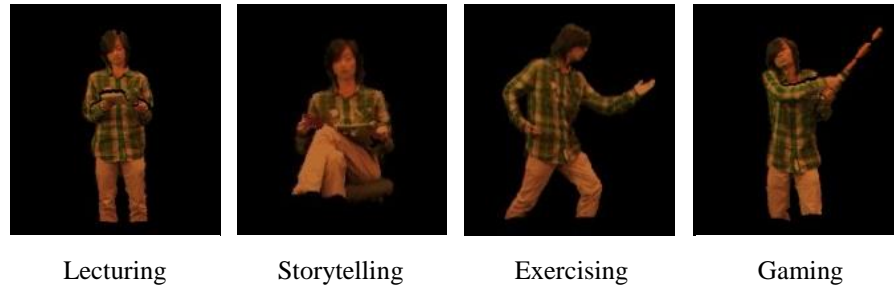


Figure 4.4: Target user activity types in the 3DTI space.

follows, ordered by their degree of motion from low to high (Figure 4.4).

- **Storytelling:** user is *sitting* in the center of the 3DTI environment with most of his/her actions concentrate on facial area. Occasional hand movement (e.g., page turning) is expected.
- **Lecturing:** user is *standing* in the center of the 3DTI environment. Frequent facial movement is expected along with occasional gesture and body movement.
- **Exercising:** user is mimicking the moves of a remote trainer (physiotherapist). Slow and gross-motor movements of all body parts are expected.
- **Gaming:** both posture and position of the user in the 3DTI environment are changing rapidly. Fast and gross-motor movements are expected at all times.

Note that we name the activity types by specific activities only for the ease of discussion. Activity types are much general classification of specific activities. For example, both conferencing and storytelling will be classified to the “storytelling type”; and both physical rehabilitation and slow dancing will be classified as “exercising type”. Compound user activities that involve constant changing of motion speed and range during the session can also be described by the four activity types because our classification module predicts the activity type on-the-fly. It can classify the first minute of the session as one activity type and the next minute as another. The following quality demand module which takes the activity type as input (Figure 4.4) will react to the changing activity type on-the-fly accordingly.

4.3.2 Sensing Data

For many 3DTI applications, extra sensing devices other than 3D cameras and microphones are broadly included in their user interfaces. Medical applications [Han 2015], for example, have the richest multimodality on content capturing and many of the on-body sensing streams (e.g., sweat, temperature, respiration, and heartbeat) in the content bundle provide good hint at user’s body movement. For exergaming [Wu 2010] and general physical

training [Kurillo 2014], acceleration and inertia sensing nodes are attached to user's body as they interact, so that activity can be predicted upon accurate motion sensing data. While these application-specific sensing stream provide convenient input for our activity classification module, A3C however targets on *general* activity capturing. This means that we do not get the advantage from specialized sensors.

Therefore, A3C exploits the user's smart phone as the input device that collects sensing data. Immersive users of A3C install our app on their phone before the application session starts. The app monitors the accelerometers embedded in the smart phone and sends the data (i.e., tri-axis accelerations) stream to the local gateway machine where the activity classification module resides. By monitoring the orientation of the accelerometer, posture of the user can be inferred. Based on the sizes of modern smart phones that are equipped with motion sensors, we assume that the users are most likely to put them in their pants pockets when being asked to carry their phones on-body. Thus, the orientation of the device becomes a reliable hint at postural signatures.

As for motional signatures, since different speeds and changing frequencies of movements directly effects the sensing data, we analyze the variation of acceleration in the time domain as well as the power spectrum in the frequency domain to generate the feature vector for SVM. For both acceleration and power variation, we use sliding windows of different sizes to include the variation in a short history. We calculate the minimum, the maximum, the average, and the standard deviation of values inside a window and include them into the feature vector. In its formal form, the resulting feature vector is as follows:

$$\begin{aligned}
P(f) &= \mathcal{F}\{A(t)\} \\
N &= \{n_1, n_2, n_3, \dots, n_k\} \\
\chi_A(n) &= [\max_n(A(t)) \quad \min_n(A(t)) \quad \text{avg}_n(A(t)) \quad \text{std}_n(A(t))] \\
\chi_P(n) &= [\max_n(P(f)) \quad \min_n(P(f)) \quad \text{avg}_n(P(f)) \quad \text{std}_n(P(f))] \\
\chi &= [\chi_A(n)|_{\forall n \in N} \quad \chi_P(n)|_{\forall n \in N}]^T
\end{aligned}$$

$A(t)$ is the acceleration data stream and $P(f)$ is the power spectrum obtained by applying (discrete) Fourier transform. $n_1, n_2, n_3, \dots, n_k$ are k pre-defined window sizes. $\max_n, \min_n, \text{avg}_n, \text{std}_n$ are statistics based on the n most recent readings in the data stream. Finally, χ is the final feature vector to be fed into the classification model. By including features originated from different window sizes, we can analyze motion features in different granularities and time spans. This provides our classification model the knowledge of micro and macro view on current activity.

4.3.3 Classification Model

The classification model is the core of activity classification. It takes the feature vector derived from sensing data as input and output the predicted activity type. To achieve real-time classification, we adopt the support vector machine (SVM) in our implementation. Like any machine learning model, SVM need to be trained by a set of pre-labeled training data first. Our training data are collected from accelerometers of smart phones when they are carried by four human actors with different body structures. Additionally, to account the effect of different sizes and shapes of the pocket where the phone sits in, the data are collected when actors are asked to wear different lower-body clothing (pants, jeans, and basketball shorts). Each actor performs the four user activities (storytelling, lecturing, exercising, and gaming) with a Nexus 4 Android phone [Nexus 2012] recording their acceleration in their pants pockets for five minutes. The total length of recorded data is 200 minutes, containing four targeted user activities with 50 minutes data each.

In the training phase, the collected data is first transformed into feature vectors as we defined in the previous sensing data section. We use $N = \{30, 150, 300, 900\}$ as five window sizes to account for the activity variation within $\{1, 5, 10, 30\}$ seconds under the 30 Hz sampling rate of the accelerometers. The derived feature vectors are paired with labels that represents different activity types. Thus, the training data, in its formal form, becomes:

$$\{(\chi_1, \alpha_1), (\chi_2, \alpha_2), \dots, (\chi_m, \alpha_m)\}$$

α_i is the label for feature vector χ_i (i.e., χ_i is the feature vector collected when actor was performing activity α_i). The goal of model training is to let SVM learns a function that maps from feature vector to a binary output (i.e., $f: \chi \rightarrow \{+1, -1\}$) in the form of:

$$f(\chi) = \text{sgn}(w^T \chi - \Theta)$$

$$w \in \mathbb{R}^{|\chi|}, \Theta \in \mathbb{R}$$

where w is the weight vector (which has the same size as the feature vector) and Θ is the constant threshold.

Since we consider multiple labels (i.e., activity types), SVM employs the one vs. all-others strategy. In other words, for each of the four activities, a distinct function is learned to discern whether the input should be mapped to the activity or not. In the learning process, this strategy learns independent binary classifiers for each label L by treating a sample (χ_i, α_i) as a positive training sample (+1) only if $\alpha_i = L$ and negative (-1) for all other labels ($\alpha_i \neq L$).

Once the model has been trained, the activity classifier predicts the human activity

given a feature vector. Having learned independent functions (i.e., independent weight vectors w_j) for each label α_j , the model performs:

$$f(\chi) = \operatorname{argmax}_j (w_j^T \chi)$$

where χ is the feature vector to be classified, w_j is the weight vector learned for the j^{th} activity type, and $f(\chi)$ is the predicted class label.

4.4 Morphing-Based Frame Synthesis Module

In this section, we introduce our Morphing-Based Frame Synthesis (MBFS) module in the content-receiving 3DTI site. The purpose of this module is to restore the removed frames in the received visual stream in order to resume the framerate of video playout. In the following, we first introduce the morphing technique and then we introduce why and how we can adopt morphing in 3DTI video frame synthesis.

4.4.1 Graphical Morphing

Graphical morphing is a technique in computer graphics which gradually changes one still image to another according to matching features (Figure 4.5). The technique was first published in [Beier 1992] and became renowned when it was shown in Michael Jackson’s music video “Black or White” in 1991. The process of morphing can be broken down into three steps: marking feature line pairs, calculating pixels mapping, and rendering of morphed image.

In the first step, feature line pairs are marked up (oftentimes manually) between the two input images. Using Figure 4.5 as an example, a pair of feature lines can contain a line



Figure 4.5: An example of morphing. Photo credit: [FantaMorph]

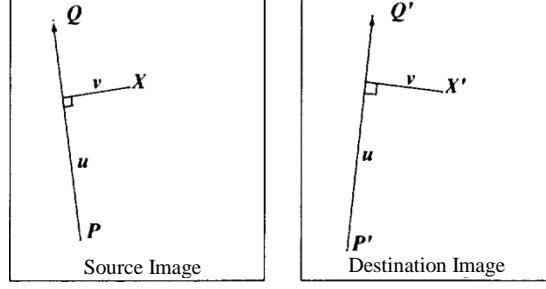


Figure 4.6: Pixel mapping calculation in morphing. Modified from [Beier 1992].

connecting the left to the right eyes of the lady and a line connecting the left to the right eyes of the leopard. More than one feature line pairs can be defined in the step. Since the feature line pairs are the references for the proceeding steps, more line pairs gives a finer definition of how the resulting morphed image should become. In the second step, the mapping of pixels in the two input images is calculated according to their relative position to the feature line pairs. The calculation of mapping can be formalized as follows. Given two input images (we call them source and destination images for the ease of discussion), a pair of feature lines (\overrightarrow{PQ} in the source image and $\overrightarrow{P'Q'}$ in the destination image) and a pixel (X) in the source image, the goal of calculation is to find the mapping pixel (X') in the destination image. The position of pixel (X') is calculated as follows:

$$u = \frac{\text{length of projection of } \overrightarrow{PX} \text{ on } \overrightarrow{PQ}}{\text{length of } \overrightarrow{PQ}} = \frac{\overrightarrow{PQ} \cdot \overrightarrow{PX}}{|\overrightarrow{PQ}|^2}$$

$$v = \frac{\text{signed distance between } X \text{ and } \overrightarrow{PQ}}{\text{length of } \overrightarrow{PQ}} = \frac{(\overrightarrow{PQ})^\perp \cdot \overrightarrow{PX}}{|\overrightarrow{PQ}|^2}$$

$$X' = P' + u \cdot \overrightarrow{P'Q'} + v \cdot (\overrightarrow{P'Q'})^\perp$$

An illustration of the calculation adopted from [Beier 1992] is provided in Figure 4.6. For each feature lines pair, a mapping can be calculated. The final mapping is derived by merging the mappings from each line pair via weight average. In the last step, one or more morphed images can be rendered easily since we have the mapping from all pixels in the source image to the pixels in the destination image. For a source pixel X and its mapping destination pixel X' , the morphed pixel (pixel in the morphed image) can be created with:

$$\begin{bmatrix} x_m \\ y_m \\ R_m \\ G_m \\ B_m \end{bmatrix} = \begin{bmatrix} x_s \\ y_s \\ R_s \\ G_s \\ B_s \end{bmatrix} + \alpha \cdot \begin{bmatrix} x_d \\ y_d \\ R_d \\ G_d \\ B_d \end{bmatrix}$$

where $\{x, y, R, G, B\}$ are the position and the color of the morphed pixel (m), source



Figure 4.7: Using morphing to synthesize video frames.

pixel (s), and destination pixel (d); and α is a ratio between 0 and 1.

4.4.2 3DTI Frame Synthesis

The natural relationship between the motion level of video content and the required framerate to sustain an enjoyable service quality is well known to be positive correlated. This means that for a high-motion activity (e.g., sport), the lowest acceptable framerate should be higher than a low-motion activity (e.g., conferencing). However, for conventional 2D videos, the limit of framerate adaptation is often defined by the capability of its capturing hardware (e.g., sampling rate of camera). Thus, under common scenarios, the adjustment (downgrading) of framerate is more of a sacrifice when the available bandwidth is insufficient. 3DTI videos, on the other hand, is more flexible on adaptation (both downgrading and upgrading) of framerate with frame synthesis via morphing. By taking two frames captured in 3DTI as input, our MBFS module uses graphical morphing to create additional synthesized frames between the two frames to boost up the framerate. An example is shown in Figure 4.7.

MBFS is not applicable to conventional 2D video frames because morphing causes distortion in the background (Figure 4.8). With depth information captured by 3D cameras, background can be easily removed from 3DTI frames before they are sent to the MBFS module. In addition, graphical properties of 3DTI videos which make MBFS feasible for 3DTI frame creation include:

Property I. Common subjects in 3DTI are human bodies, which have fair sizes that take up major portion of the scene. This makes auto-marking of feature line pairs

more accurate and efficient.

Property II. The number of subjects in one scene is restricted due to the interactive characteristic of the application and the size of the display. In most user activities [Vasudevan 2011], there will only be one user per 3DTI site, which allows us to make use of the skeleton information provided by the 3D camera (i.e., Kinect) in feature line marking.

Modern 3D cameras nowadays have sampling rates at 10~60 frames per second (FPS). This generates thousands of captured frames in a clip in minutes. Thus, the marking of feature line pairs (first step in graphical morphing) must be done automatically. MBFS adopts feature detection (SURF [Bay 2008]) and feature matching (FLANN [Muja 2009]) tools to mark the different locations of the same feature in two frames (Figure 4.9). Due to the graphical simplicity of 3DTI scene (Property I), the feature-based matching can provide a meaningful number of matching feature pairs. Given the matching features, a planar graph is built using Delaunay triangulation [Berg 2008] on the two input frames with the detected features as vertices (Figure 4.10). The edges connecting the same pair of features become the final feature line pairs for morphing. For rendering machines (i.e., gateway machine in the receiving 3DTI site) with inferior computation capability, the feature-based matching process may be too time-consuming to sustain real-time rendering. In such cases, the marking of feature line pairs can be done by directly adopting the skeleton, provided by Kinect (Property II), as feature lines. After MBFS acquires the feature line pairs, the rest of the morphing steps are straightforward. The calculation of pixel mapping and the rendering of the final morphed frame directly follow the formulae provided in the previous section.



Figure 4.8: Morphing on video frames without background removal causes distortion.



Figure 4.9: Feature matching for morphing.

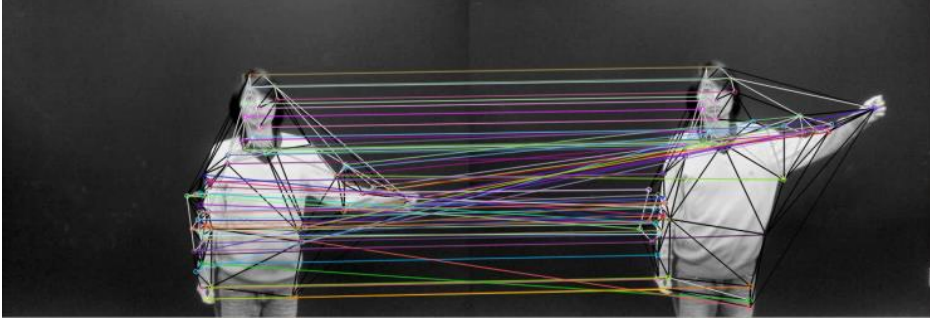


Figure 4.10: Line pairs marking for morphing.

4.5 Quality Demand Module

The quality demand module is the decision-making component of the entire A3C system. It takes the discerned activity type from its preceding activity classification module as input and decides the number of frames to be removed from the stream. The decision is made upon a mapping model between target activity and the maximum ratio of synthesized frame to be injected into a video without degrading the perceptual quality. In the following, we first introduce the effect of MBFS to the visual quality and then we introduce the development of the mapping model.

4.5.1 Effect of MBSF on Visual Quality

As we shown in Figure 4.7, the quality of the synthesized (i.e., morphed) frames tend to have inferior visual quality comparing to actual frames captured by the camera. This happens because of possible feature mismatch or lacking enough number of feature points in the automatic feature line pairs marking step. When the difference between the two input frames is large, the possibility of feature mismatch and lacking matching feature points rises. In Figure 4.11, we show the morphing results when the input frames are different (top row) versus when the two frames are similar (bottom row). We see that more artifacts are created in the former case and hence a worse visual quality then the latter. When a stream contains

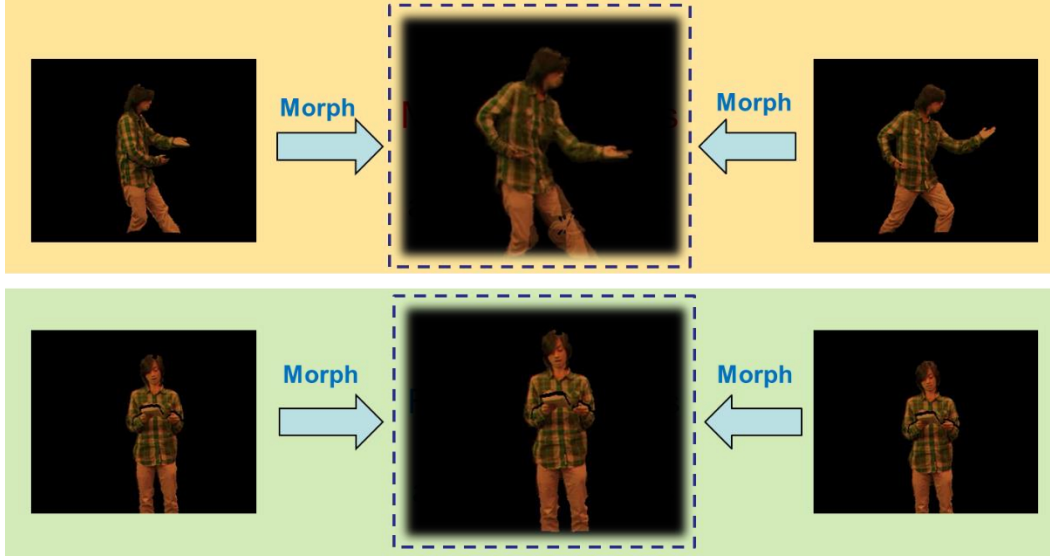


Figure 4.11: Morphed frame of a high motion activity (top row) contains more artifacts than low motion activity (bottom row).

more synthesized frames with perceivable artifacts, the visual quality of the service will be compromised. Since the difference between two consecutively captured frames are related to the motion level of user activity as we shown in Figure 4.11, A3C needs to be aware of the activity semantics. The amount of synthesized frames restored in the received video stream (which is equivalent to the amount of frames removed in the content-capturing 3DTI site) have to be dynamically adjusted according to the activity type. In the following section we introduce the activity-to-RSF (ratio of synthesized frames) mapping model that balances between bandwidth saving (i.e., frame removing) and perceptual quality.

4.5.2 Activity-to-RSF (Ratio of Synthesized Frames) Mapping Model

While MBFS gives A3C system the ability to reduce the framerate at the capturing 3DTI site to save transmission bandwidth, if we do it regardless of the activity semantics, perceptual quality can be severely degraded with high motion activities. Therefore, we want to find the noticeability threshold of the maximum ratio of synthesized frames (RSF) with different 3DTI user activities. We define RSF formally as:

$$RSF \equiv \frac{|synthesized\ frames|}{|received\ actual\ frames| + |synthesized\ frames|}$$

Since in A3C we use MBFS only to restore the removed frames, by its definition, from the content-capturing 3DTI site's point of view, RSF is also equivalent to:

$$RSF = \frac{|removed\ frames|}{|captured\ frames|} \approx bandwidth\ reduction\ rate$$

To acquire the noticeability thresholds of RSF to different activities, we conduct a subjective experiment. We invite real users to rate 3DTI recordings of different activities with different amount of frames replaced by synthesized ones to simulate the rendering result of A3C.

For each of the four user activity types targeted, a 30 seconds clip is recorded. The actor in all videos is the same male with a 5'7" height. For lecturing and storytelling scenarios, the actor is speaking to the camera in his standing/sitting position. For exercising, the actor is learning Tai-Chi exercise in slow motion. For gaming scenario, the actor is participating in a lightsaber fencing match with another remote party (Figure 1.2).

From each of the original clips we produce four variations with different RSF values. We replace actual frames in the clips with synthesized ones with RSF set at 0.00 (zero synthesized frames), 0.10 (1:10), 0.14 (1:7), and 0.25 (1:4). By definition of RSF, these configurations result in bandwidth reduction rate of 0%, 10%, 14%, and 25%, respectively.

We recruit 15 participants (5 females and 10 males) to view and rate the visual quality of the clips. The score are given on a Likert scale of 1 to 5 following the MOS (Mean Opinion

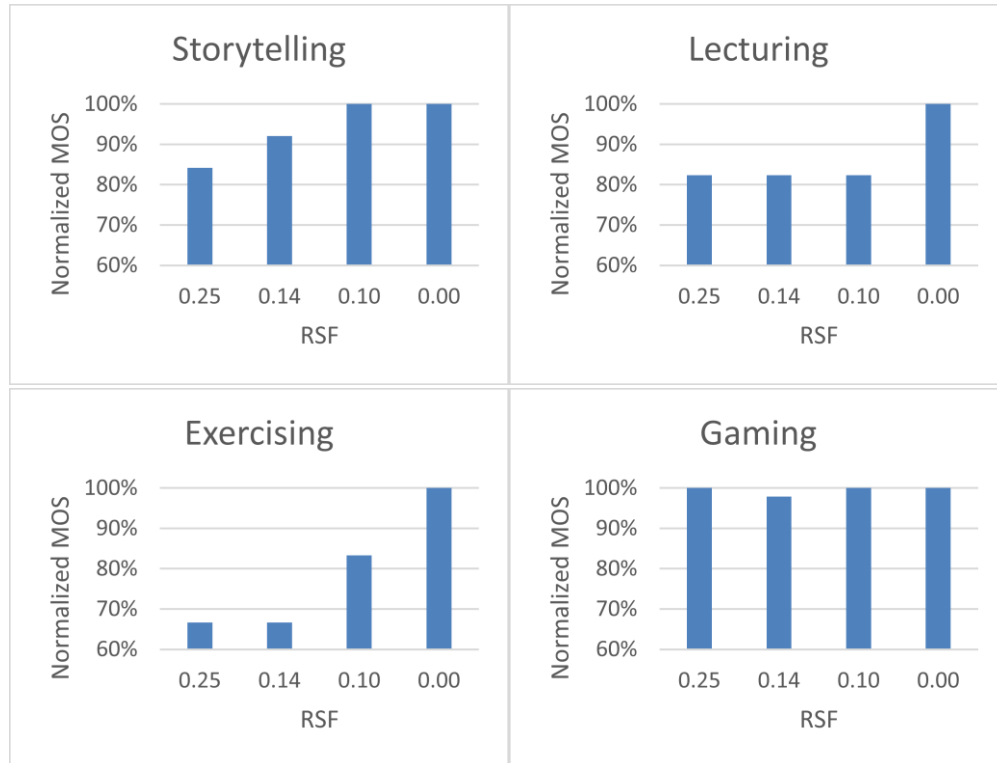


Figure 4.12: Normalized MOS under different RSF.

Score [ITU 2008a]) standard. The maximum, average, and minimum ages of the participants are 50, 26.5, and 21, respectively.

To calibrate biased scores due to fatigue, we take the Absolute Category Rating with Hidden Reference (ACR-HR) [ITU 2008b] approach. The main idea of ACR-HR is to play a reference clip (the original untampered recording) before each test clip so that the participants can adjust the score of each test clip based on the reference. This arrangement is kept secret from the participants to minimize the effect of anticipation. The total length of our experiment sequence is 20 minutes.

Results of the experiment are plotted in Figure 4.12. The x-axis is the RSF value and the y-axis is the normalized MOS given by the participants. Using these results, our mapping model between activity and RSF is built as shown in Table 4.1.

Table 4.1 RSF thresholds for different activity types when targeting different QoE.

QoE Target	Storytelling	Lecturing	Exercising	Gaming
100%	RSF = 0.10	RSF = 0.00	RSF = 0.00	RSF = 0.25
90%	RSF = 0.14	RSF = 0.00	RSF = 0.00	RSF = 0.25
80%	RSF = 0.25	RSF = 0.25	RSF = 0.10	RSF = 0.25

4.5.3 Subjective Experiment Result Analysis

In the storytelling activity, we can see that the synthesized frames blend in successfully. This means that the degradation on spatial resolution caused by synthesized frames is imperceptible to the participants. In a storytelling scene, the limited movement of the subject's body minimizes the difference between frames, which makes the morphing result more authentic. In addition, the size of the subject is smaller due to the sitting posture, making the degradation even harder to be noticed.

The influence of motion on the prominence of visual degradation and morphing artifacts can be seen in the comparison between lecturing and exercising activities. Again, the low motion (lecturing) of the video content lowers the bar of synthesizing an authentic frame. On the other hand, the gross body movements in the high motion scene (exercising) make the viewer concentrate more on the details of the subject's body. This makes the spatial degradation introduced by synthesized frames more detectable and hence bring down the MOS.

Following the same logic, the gaming scenario ought to be the most vulnerable one to degradation from MBFS. However, we discover that the relationship between user’s demand on visual quality and motion is more complicated. Counterintuitively, the scores of the gaming scenario are the highest ones among all user activities. The cause of this phenomenon is revealed in the feedbacks of the participants. When asked about noticeable degradations in the gaming clips, participant W.C. (male, age 21) stated that *“The unpredictable rapid moves of the subject make it hard to concentrate on the details”, “All the jumping around and the waving of the stick (lightsaber)... there are too many things going on in the scene.”* Even when the participants do notice the differences, they do not necessarily see them as drawbacks. Participant Z.G. (male, age 22) described the degradation as *“really cool special effects”*. Z.G. further explained that *“The motion blur of the lightsaber and the subject makes the fencing more exciting and enjoyable than the other clips.”* From the participants’ feedback we see that, when the speed of motion surpasses a certain level, the noticeability of MBFS degradation drops rapidly, making the tolerance of the degradation even higher than low motion activities.

4.6 Evaluation

The evaluation of A3C is three-folds. First, we evaluate the accuracy of activity classification. Since activity semantics is the important input for our decision making module (i.e., the quality demand module) in A3C, the accuracy determines the performance of the whole system. Second, we evaluate the performance of A3C in an immersive (i.e., interactive participation) 3DTI application. We setup a 3DTI gaming testbed and invite real users to experience the game play experience with the A3C system. Second, we evaluate the performance of A3C in a non-immersive (i.e., passive observation) 3DTI application. We adopt the crowdsourcing experiment method and setup a public viewing website for our 3DTI recordings. User feedbacks are collected online to validate the effectiveness of A3C. In both interactive and passive 3DTI application evaluations, we focus on bandwidth saving as well as service quality in our experiments.

4.6.1 Activity Classification Accuracy

The accuracies of classification on each user activity are listed in Table 4.2. The evaluation is done by 10-fold cross-validation on the sensing data compiled previously in the Section 4.3.2. The overall accuracy of classification is 91.5%.

We can see from Table 4.2 that lecturing and exercising activities have the lowest accuracy (yet still higher than 80%). The fact that the false positives of the two happen when

users are actually performing the other activity implies our classification module is prone to confuse the two activities together. This is due to the fact that during exercising activity, the subject occasionally needs to stop his/her move and stands still to observe the demonstration of the trainer. According to the sensor data, this situation becomes identical to the lecturing activity. Another observation is the perfect accuracy of discerning the storytelling activity. Due to its unique postural signature, the activity classification module can easily tell storytelling apart from the other activities by examining the orientation of the phone.

Table 4.2 Accuracy of user activity classification

True \ Inferred	Storytelling	Lecturing	Exercising	Gaming
Storytelling	100%	0.0%	0.0%	0.0%
Lecturing	0.0%	84.8%	14.4%	0.8%
Exercising	0.0%	7.6%	87.6%	4.8%
Gaming	0.0%	0.2%	6.2%	93.6

4.6.2 Immersive Interactive Evaluation

Experiment Testbed. We implement a virtual fencing game as our experiment testbed (Figure 4.13). The game includes two 3DTI user gaming sites connected within the campus network. Each site contains one Kinect camera to capture the 3D scene. The 3D data of each player is transmitted to the other site and rendered in the virtual world. Player in each site wears a head-mounted display embedded with accelerometers and sees from her first person's perspective in the virtual world. A sword is rendered in the virtual world in the hand of the player and the objective of the game is to hit the remote player (the opponent) with

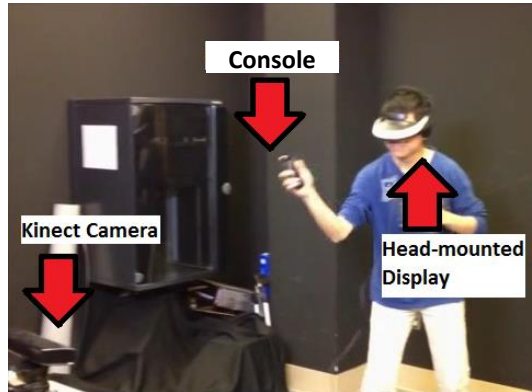


Figure 4.13: Hardware interface of TEEVE Endpoint.



Figure 4.14: Game screen of the virtual fencing.

the weapon. Player's health point decreases as hit by one another. A player wins when her opponent's health point decreases to zero. Figure 4.14 shows a sample game screen.

Experiment procedure. Among the two gaming sites, only one of them has the A3C system enabled. We recruit seven real players to participate in the game in pairs. After a five-minute gameplay, we interview each player about their gameplay experience. After that, the players are asked to switch sites, play the game for another five minutes, and being interviewed again. The player do not know that the two sites have different system settings. The interview questions are listed in Table 4.3, which focus on the sensory immersion of gameplay experience proposed in [Ermi 2005]. The players are asked to answer the questions on a 5-point Likert scale.

Table 4.3: Interview Questions and Average Scores.

	A3C enabled	A3C disabled
Q1: Are you satisfied with the graphical resolution of the game?	3.3	3.3
Q2: How is the responsiveness of the game control?	4.4	4.1
Q3: Do you consider the game realistic?	3.4	3.7
Q4: Do you enjoy the immersive experience?	4.6	4.6

(Score: 5 being the most and 1 being the least)

Results and analysis. The average score to reach question is listed in Table 4.3. The result suggests that the gameplay experience within both user sites do not have significant difference. We run the two-way ANOVA test [Winer 1991] on the compiled score and the similarity of the two sites is supported by the statistic result ($F < 0.01$, $p = 1$). The bandwidth consumption of the user site with semantic modules enabled is always lower than or equal

to the other site during all game sessions. The highest bandwidth saving rate during the experiment is 25%. This shows that the semantic module can achieve resource saving without incurring noticeable quality degradation in interactive 3DTI application.

4.6.3 Non-immersive Crowdsourcing Evaluation

Experiment Testbed. We record a 60 second physical rehabilitation session with our delivery chain. The actor in the video follows the instruction in a publically available physical therapy demonstration video provided by a physical therapy provider [TenEase]. During the 60 second 3DTI recording various activities with different motion ranges, speeds, and postures are involved. From the feedback of our activity classification model, 22% of the video time is classified as *storytelling* (sit with low motion), 12% is classified as *lecturing* (stand with low motion), 33% is classified as *exercising* (low-motion gross-motor moves), and 33% is classified as *gaming* (high-motion gross-motor moves). Again, we create two versions of video clips from the recording: one processed by A3C, and the other is the raw recording.

Experiment procedure. Since the test content is asynchronous and non-immersive viewers, we adopt crowdsourcing methodology [Kittur 2008] to collect more user feedbacks. We set up a website (Figure 4.15) that plays the two versions side-by-side and simultaneously. Next to the video, there are some simple instruction sentences with a straightforward yes/no question:

*“This website consists of two clips playing side-by-side. Please watch them and then answer the following question: **Which of the clips has a better visual quality?**”*

Beneath the instruction, there are three buttons denoted “The one on the LEFT”, “The

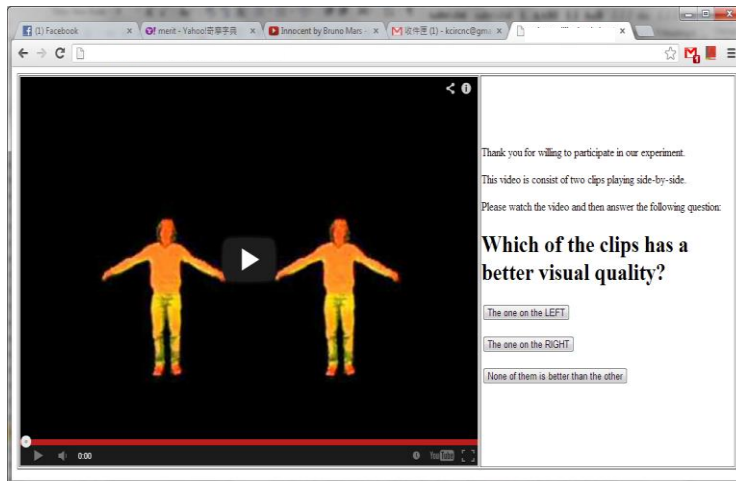


Figure 4.15: Crowdsourcing website.

one on the RIGHT”, and “None of them is better than the other”. When a viewer hits the button, she will be redirected to a thank you page; and the result will be collected in our webserver. The webpage is publically accessible and the link is advertised through social networks and through two mailing lists of University of Illinois and National Taiwan University.

Results and analysis. In one-month period, 147 users have visited the website and 92 have given valid feedbacks. Among the feedbacks, 18% of the viewers prefer the processed version, 54% cannot tell the difference, and 28% prefer the raw version. In result, 72% of the viewers do not perceive degradation caused by our delivery chain. In terms of resource saving, the processed version has 10% lower bandwidth consumption than the raw version. This shows that our semantics-aware adaptors save networking resource in content delivery without introducing perceivable negative effects to service quality of non-immersive 3DTI application as well.

4.7 Conclusion

In this chapter, we focus on solving the outbound bandwidth bottleneck issue of general content capturing 3DTI systems. We develop the A3C system which takes a morphing-based approach to synthesize frames in 3DTI videos. We further extend the technique to a quality metric: RSF, which affects the visual quality of a video with different levels of bandwidth saving. We combine the adaptation of RSF with motion characteristics of different activity semantics in the 3DTI space. With a machine learning (SVM) approach, the user activities can be classified in real-time based on the motion sensing data acquired from users’ mobile phones. Result shows that the level of motion of a user activity has significant influence on the prominence of RSF’s effect on QoE, and the relationship is not intuitively monotonic. Finally, by combining the activity classification module and the quality demand module, we build up a general 3DTI system for efficient content capturing, which automatically classifies the user activity and assigns suitable RSF to the production of the video stream.

Our contribution with A3C can be summarized as follows:

- Design and evaluation of the application of morphing technique in 3DTI video compression and enhancement.
- Devising the ratio of synthesized frames (RSF) metric which enables us to quantify the tradeoff between resource consumption and perceptual quality in 3DTI video production.

- Investigation of the relationship between the motion characteristics of user activities in 3DTI space and the noticeability of degradation on spatial quality.
- Implementation and validation of a semantics-aware adaptive content capturing system that saves fair portion of computing and networking resource without comparable degradation on QoE.

On the semantic level, the activity classification module grants A3C the awareness of activity semantics in the cyber-physical regime. Rather than relying on specialized motion sensors, A3C exploits the most commonly available motion sensor: user's smart phone as input interface to collect motion features of current user activity. The other A3C modules (i.e., quality demand module and MBFS module) configure themselves on RSF according to the acquired activity semantic so that the overall system can balance between resource saving and quality preservation.

On the system level, we evaluate A3C via objective resource saving as well as subjective user study. Objective evaluation focuses on bandwidth savings rate to address the resource requirement factor. The evaluation is done with activity sessions with different motion characteristics that incurs different visual complexities in the video streams to address the content complexity factor. Subjective evaluations that address the user satisfaction factor are conducted by in-lab close-up interviews with immersive application users and large-scale crowdsourcing feedback collection with non-immersive application users.

Through subjective experiments targeting immersive and non-immersive applications, A3C is proved to be able to save up to 25% networking resources without incurring perceptible degradation on service quality. Therefore, we conclude that A3C successfully bridges the gap between semantic and system level in the capturing phase of 3DTI's content delivery. With the semantics-awareness brought by A3C, advanced 3DTI applications which bear higher quality and multimodality demands can sustain more elaborate content capturing environment (i.e., capturing site with larger-scale camera array or more input sensors) with moderate networking capability.

5. Amphitheater: User Semantics in Dissemination Phase

5.1 Introduction

While we witness the thriving of live broadcasting video services such as [PPLive], [Ustream], and [Youtube], large-scale non-immersive audience is rarely included in the design of 3DTI applications. The main reason behind this is the resource limitation of the overlay P2P (peer-to-peer) network that is used to disseminate the content. Unlike conventional IPTV (internet protocol television), audience (i.e., non-immersive users) of 3DTI requires more than one 3D stream from more than one performer (i.e., immersive user) to render single application scene. Therefore, the bandwidth requirement of 3DTI content dissemination in large-scale broadcasting is magnitudes more than regular 2D video applications.

Targeting this resource bottleneck in the dissemination phase, in this chapter we introduce the Amphitheater system. The Amphitheater system is a media-enriched multi-view live broadcasting system that takes in user semantics and renders a shared virtual space which mimics an amphitheater in the real world (Figure 5.1). Users in the 3DTI Amphitheater are divided into two groups: performers and the audience. A user of an immersive site, or a *performer*, produces 3D streams by camera array in her physical user space. A 3D model of the performer will be constructed from the streams and placed on the virtual stage to interact with other performers. Users of non-immersive sites, or *the audience*, passively observe the interaction on virtual stage without any involvement. Every performer

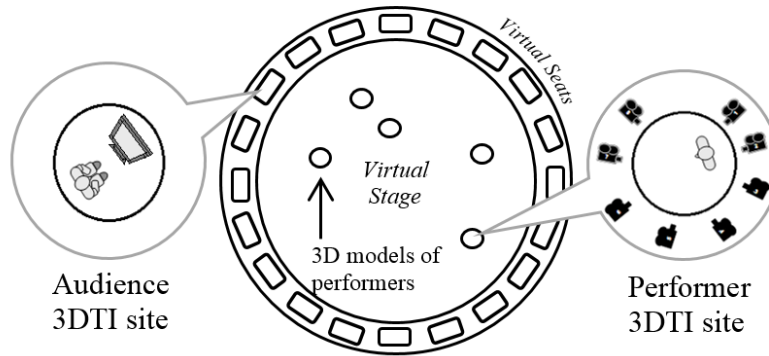


Figure 5.1: 3DTI Amphitheater.

and every audience is a *viewer* in the Amphitheater since they all need to view the virtual space from their different perspectives. Hence, in this chapter we use the terms *viewer* and *user* interchangeably. In the virtual amphitheater, a performer's standing position on the virtual stage is synchronized with her position in the physical space. The audience's position is fixed in pre-assigned virtual seat.

3DTI brings new challenges to the well-established IP-based live broadcasting framework. The first challenge comes from the free-viewpoint feature. The multi-source (multi-performer), multi-content (multi-camera) dissemination transforms 3DTI broadcasting into a forest construction problem in the P2P overlay network. Multiple participants subscribing to multiple streams from multiple performers incurs massive bandwidth consumption for the dissemination. The Amphitheater system tackles this problem by the design of virtual seats. Exploiting the semantics that audience users are evenly placed in virtual seats that surround the central virtual stage, Amphitheater system makes the aggregated view of the audience covers 360 degree perspective of a performer. This implies two advantages in content dissemination. 1) The stream subscriptions of adjacent audiences have substantial overlap. Via P2P sharing, the overlapped subscriptions can be fulfilled together. 2) For each stream capturing a particular angle of a performer, there exists at least one audience who subscribes to it and thus can aid its distribution as a hub. This alleviates the pressure on the limited outbound bandwidth of the source performer site.

The second challenge is efficient delivery of the multi-view content. Although the camera array captures a performer with an omnidirectional perspective, a user simply does not require all the streams since she can only fix her view on one side: when the viewer is looking at the front of a performer, the streams capturing the back do not contribute to her view. This leads us to differentiate streams further using user semantics. In the first tier of differentiation, we argue that not all cameras are equally important to a viewer. A more in-sync direction of the camera with the viewer's perspective produces a more contributive stream which deserves a higher delivery priority. We address this as the *view-based priority* of a stream request. In the second tier of stream differentiation, we argue that not all performers are equally important to a viewer. The audience may be more interested in the vocalist of a rock band, the diva of an opera, or the quarterback of a football team. We address this as the *role-based priority* of a stream to capture its importance based on the semantic relation between its viewer and its performer. Combining view-based and role-based priorities, we devise the hierarchical stream prioritization. Each user in the Amphitheater has her own hierarchical priority stated in her subscription request, which aids the construction of a more efficient content dissemination forest. The design helps the Amphitheater achieve

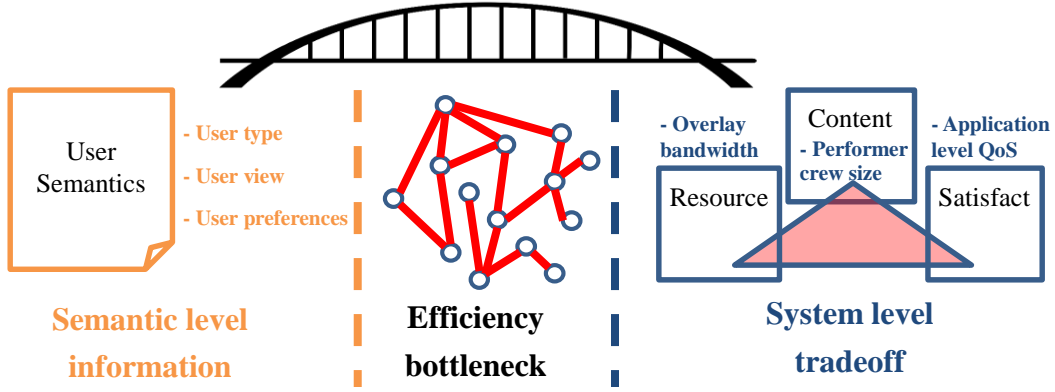


Figure 5.2: Mapping from semantics-aware content delivery framework to Amphitheater.

better service quality with larger user scale in a broadcasting application session.

Mapping to semantics-aware framework. As a 3DTI system targeting on large-scale live broadcasting, Amphitheater follows our semantics-aware content delivery framework to exploit user semantics to solve the efficiency bottleneck in the content dissemination phase (Figure 5.2 and Figure 1.4). Semantics-awareness of Amphitheater is reflected in two aspects in the construction of dissemination forest: user type differentiation (virtual seat design) and content differentiation (hierarchical priority). These components in Amphitheater will be detailed in later sections respectively followed by evaluation based on large-scale broadcasting session simulation. The effectiveness of semantic-awareness is validated by the experiment results on its improvement on system-level tradeoffs between 1) bandwidth usage in the P2P overlay network (resource requirement factor), 2) number for performers in the application session (content complexity factor), and 3) overall application level quality of service observed by both audience and performers (user satisfaction factor).

5.2 System Model

In the system model of Amphitheater system, multiple 3DTI sites collaborate together to produce and distribute the 3D visual content. Among all the participating sites, only a subset of the sites is producing the 3DTI video streams. Users within these sites have their 3D models projected into the virtual space where they interact with each other. In the rest of the sites, the users only passively observe the activity from the view they choose and do not have their 3D models built by the system. Together, these sites form a P2P overlay network that delivers the content streams. An example of a use case is illustrated in Figure 5.3a. Inside the virtual space depicted in the figure, only three performers have their 3D models created (P_1 , P_2 , and P_3). These performers actively interact with each other through their model on the virtual stage. Out of the stage there are two other audiences (A_1 and A_2) passively

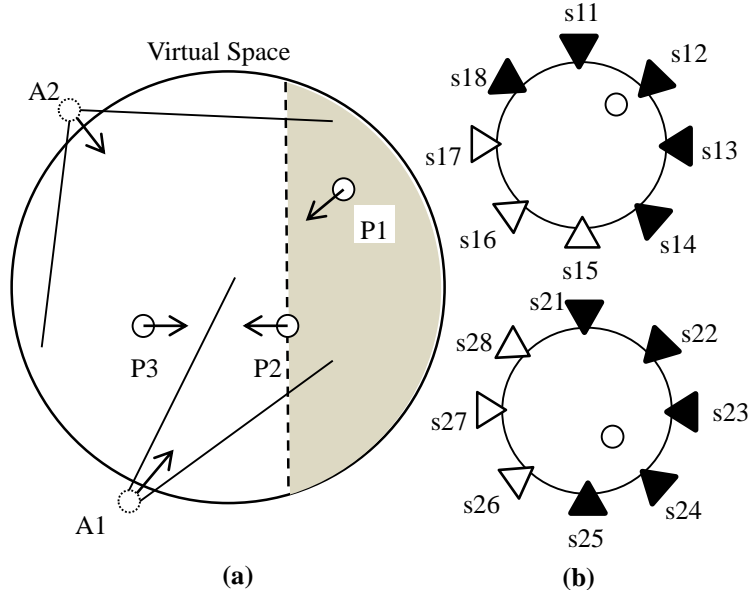


Figure 5.3: An illustration of (a) virtual space and (b) physical space.

observing the interaction from the view point (i.e., virtual seat) they get assigned. Their sites do not produce any 3D visual contents but only passively receive them from other sites. Based on the different roles of a participating site in a 3DTI session, we classify them into three types: performer sites (P_1 , P_2 , P_3), audience sites (A_1 , A_2), and a session manager (not shown in the figure). In the following, we introduce the system requirements and the assignments of each type.

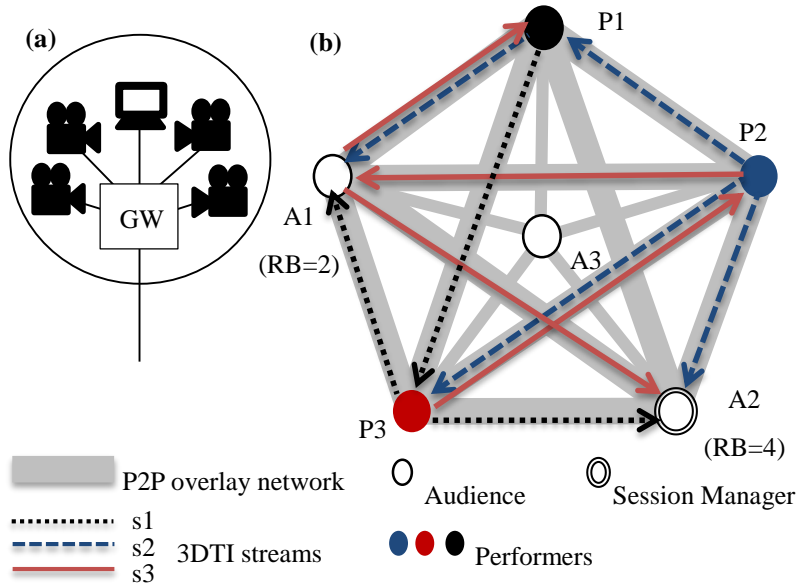


Figure 5.4: Content dissemination forest.

5.2.1 Performance Sites

A performer site is a producer of the 3DTI content during a session and also a consumer of the content produced by other performers. As mentioned, users in the performer sites are the performers that interact with each other. Thus, the hardware requirements of a performer site include a 3D camera array, a head mounted display, and a gateway (GW) machine (Figure 5.4a).

The 3D camera array consists of a group of 3D cameras that surround the user space (Figure 5.3b). Each camera captures a facet of the object in the physical user space. In real-time, the streams produced by the cameras are delivered to the gateway machine located in each participating site to render the 3D model of the performer. The gateway machines in the performer sites receive streams produced by all the performer sites and render out a consistent virtual space. According to the standing position and the view direction of performer in the physical space, the head mounted display connected to the gateway shows the relative view of the virtual space.

5.2.2 Audience Sites

The audience sites are observers during a 3DTI session. Users in audience sites passively view the performance without any involvement. Thus, the 3D camera array becomes optional within an audience site. The basic hardware requirement for the audience site only includes the gateway machine and a display. The gateway machines in audience sites collect the streams produced by the performer sites and render the 3D virtual stage space. The virtual stage space is shown by the display in the audience site, which can be either a conventional display or a head mounted one.

5.2.3 Session Manager

In addition to the 3DTI site, there is an independent session manager to examine the subscription requests sent from all the other participating sites and build the content dissemination network accordingly. The subscription request contains information to determine: 1) which streams the subscriber requires to render her view and 2) what the hierarchical priorities of the subscriber are. With these requests, the session manager can construct the dissemination network using our forest construction algorithm. In reply, the session manager will tell the subscriber from whom should it receive the required streams.

5.3 User Model

In this section we describe our user model in two parts. First, we introduce the characteristics of the Amphitheater and how its structure effects the subscription and

dissemination of streams. Second, we introduce the hierarchical stream prioritization, which addresses the user semantics by combining the view-based and the role-based priorities of a viewer.

5.3.1 Virtual Amphitheater

The virtual space constructed mimics an amphitheater, where the performers are interacting on the central virtual stage and the audiences are assigned with their own virtual seats which surround the stage and disperse evenly. Thus, the perspectives of a performer and an audience are different. While a user in a performer site may not be able to see the whole performer crew due to her standing position and view direction, the users in the audience site can always choose to see the whole virtual stage or to focus on part of the stage. As illustrated in Figure 5.3a, where the arrows indicate the view directions, performer P_2 can only see P_3 but not P_1 since P_1 is standing in her blind side (the grey area). As for A_1 and A_2 in the audience, A_2 chooses to see the whole stage while A_1 zooms-in on a particular part as if she sees through opera glasses.

In addition to the visibility of the performers, another factor that affects the user view is the relative position of a viewer-performer pair. At any given time, a viewer can only see a facet of a performer. Thus, the viewer site does not need all the streams originated from that performer site since half of the streams capture the opposite facet and do not contribute to the view. An example is provided in Figure 5.3, where the streams are denoted by the cameras in the performer sites (Figure 5.3b). According to the relative positions, performer P_3 only requires stream s_{15} , s_{16} , s_{17} from P_1 and stream s_{26} , s_{27} , s_{28} from P_2 .

The virtual seats fix the position of each audience. This design restricts the audiences from moving their viewpoints arbitrarily inside the whole virtual space (e.g., around the stage or even on stage) and it complies with the common sense in a real theater, where seats are pre-assigned and fixed during the performance. Furthermore, the design brings two advantages to the delivery of streams. First, it enhances the effectiveness of content sharing. For two audiences in adjacent seats, their views are very likely to overlap with each other by a fair portion. This implies that in stream delivery, the same stream is more likely to be subscribed by multiple sites, which makes the sharing of the content being able to save more bandwidth in the P2P network. For example, in Figure 5.3, while audience A_2 subscribes to stream s_{27} , s_{28} , s_{21} an audience A_2' sitting on her left possibly subscribes to s_{28} , s_{21} , s_{22} , hence a good portion of required streams can be shared between them.

Second, the surrounded arrangement of seats helps the distribution of streams from their source sites. Since the stage is surrounded by the audience, each and every side of the

performer's body must be looked at by some audiences at any given time. This implies that every stream (each capturing different sides of the performer) is likely to be subscribed by one or more audience sites at any time. In the dissemination network, these audience sites act like hubs that help the performer sites to distribute their streams to other performers. Often times the outbound bandwidth of a performer site is not enough to distribute its streams to all the other performers when the performer crew grows big. In that case, audience sites can help as hubs to relay those contents. The surrounded seat design raises the possibility of the existence of such hub audiences.

5.3.2 Hierarchical Stream Prioritization

In order to encompass user semantics in the construction of content dissemination network, the view-based priority and the role-based priority are both addressed in the hierarchical prioritization of streams. We introduce the three logical objects that we define in the hierarchy as follow (using Figure 5.3 as an example):

- **Stream.** A 3D video stream created by a camera. This is the basic content unit in the dissemination network. We denote a stream by s with a postfix number for identification, e.g., s_{21} .
- **View.** The set of streams that are created in the same performer site. We denote a view by v with a postfix number to identify the site, e.g., $v_2 = \{s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}, s_{27}, s_{28}\}$ is the view of performer P_2 in Figure 5.3.
- **Session.** The set of views in the Amphitheater. We denote a session by x , e.g., $x = \{v_1, v_2, v_3\}$.

As we discussed previously, there are two factors that affect the importance of a stream to a particular viewer. First, the view-based priority reflects the importance of a camera per site. If its shooting angle complies with the view angle of the viewer, then the stream it produced becomes more important to the viewer. Second, the role-based priority reflects the importance of performer per session. If a stream captures a performer in whom the viewer is more interested, then the stream is more important to the viewer.

To address these factors, each viewer would provide information to determine her own hierarchical priority in her subscription request. The hierarchical priority is represented as a sequence of numbers assigned to each element in a view or a session. For example, to address the view-based priority, a viewer may set her hierarchical priority as:

$$HP(v_2) = \{0, 0, 0, 0, \mathbf{3}, \mathbf{4}, \mathbf{3}, 0\}$$

$$v_2 = \{s_{21}, s_{22}, s_{23}, s_{24}, \mathbf{s_{25}}, \mathbf{s_{26}}, \mathbf{s_{27}}, s_{28}\}$$

This hierarchical priority (*HP*) states that, for this viewer, stream s_{26} is the most important stream among all streams in the set v_2 ; s_{25} and s_{27} come second; and the viewer does not care about $s_{21} \sim s_{24}$ and s_{28} . A larger number indicates higher importance of the stream. The method we apply to determine the numbers for view-based priority is modified from the contribution factor (*CF*) proposed in [Yang 2006b]:

$$CF = \vec{O}_l \cdot \vec{O}_u$$

where \vec{O}_l is the shooting direction of a camera, \vec{O}_u is the view direction, and *CF* is defined as their inner dot product. Thus, the result reflect part of the user semantics that relates to user's view. Our priority numbers are calibrated from the *CF* value by 1) treating non-positive *CF* as zero and 2) normalizing the numbers.

To address the role-based priority, the hierarchical priority is defined similarly, as a mapping from the views in a session to numbers, e.g.,

$$HP(x) = \{5, 3, 2\}$$

$$x = \{v_1, v_2, v_3\}$$

This states that, for the viewer who assigns this HP, the view that captures performer #1 (v_1) is more important than v_2 and v_3 ; and v_3 captures the least important performer. Again, the numbers are non-negative and they are normalized for the ease of computation. The determination of the numbers depends on the role of the viewer and the scenario. There are many ways to determine the role-based priority because it is affected by compound social, subjective, and objective user semantics. Here we provide three examples for three different roles of a viewer.

Viewer Role 1: Parent in the audience of a school play.

$$HP(x) = \{1, 8, 1\}$$

$$x = \{v_1, v_2, v_3\}$$

The priority numbers are subjectively assigned by the viewer (parent). Since v_2 is capturing the image of the viewer's child on stage, she assigns the highest priority to it and assigns the others kids with one.

Viewer Role 2: Player in a table tennis dual match.

$$HP(x) = \{10/3, 10/3, 10/3\}$$

$$x = \{v_1, v_2, v_3\}$$

In this case, the viewer is a performer. The priority numbers are uniform. For v_1 being the viewer's teammate and v_2, v_3 being the opponents, they have equal possibilities to change the game. Thus the role-based priority numbers are the same for all three views in the session.

Viewer Role 3: Participant of a cocktail party.

$$HP(x) = \left\{ \frac{1}{\text{dist}(\text{performer \#1})}, \frac{1}{\text{dist}(\text{performer \#2})}, \frac{1}{\text{dist}(\text{performer \#3})} \right\}$$

$$x = \{v_1, v_2, v_3\}$$

In this scenario all the participants (performers) interact with each other in a ball room (stage) that is monitored by the security (audience). The priority numbers are objectively determined based on the distance between two participants in the virtual environment. This way, the view of a performer who stands closer to the viewer will be given higher priority than the ones that are far away. Higher priority implies higher admission rate of stream requests and hence a higher quality of the performer's image. This complies with the sense in the real world, where it is hard to see clear of a further object but easy to see an object clearly when it is close to you.

The three examples show three possible ways to determine the role-based priority based on user semantics: 1) subjectively, 2) uniformly, and 3) objectively. Note that these are not the only methods but rather intuitive examples. Our design is compatible with different determination methods of role-based priority in different application scenarios and user roles.

5.4 Stream Delivery Model

Our Amphitheater system adopts the pub-sub (publish-subscribe) model [Eugster 2003] as its content dissemination paradigm. The model has three core components: publisher, subscriber, and broker. In the Amphitheater, the publishers are the performer sites; the subscribers are all the viewers (including both performer and audience sites); and the broker is the session manager. We introduce the message exchange among the three components as follows. An illustration of the process is shown in Figure 5.5.

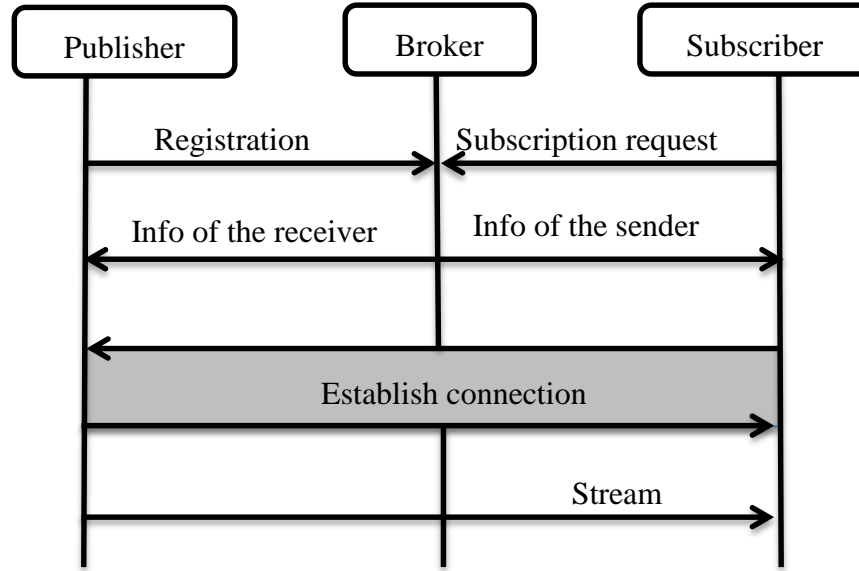


Figure 5.5: Protocol in the pub-sub model.

In the beginning of a session, the publishers register their cameras to the broker. The registration states the number of cameras that the performer owns and their shooting angles. The subscribers, on the other hand, submit their subscription requests to the broker, which contains the user type (performer or audience), the state (user position and view direction), and the interests (role-based priority). Whenever there is an update on the subscriber's information, it renews the subscription request and sends it to the session manager again. Registrations and subscriptions also contain the geographic location and the maximum inbound/outbound bandwidth of the sender. This information is used to estimate the propagation delay between sites and will be used in the construction of dissemination forest.

After the broker receives all the registrations from the publishers and all the subscription requests from the subscribers, it starts to translate them into stream requests. Since the broker knows 1) the positions and view directions of every viewer, and 2) the positions of each performer and the shooting angles of their streams, it can deduct which streams would be needed by a particular subscriber in order to construct her view. For example, in Figure 5.3, after the broker gets the positions of all participants (including all audiences and all performers), it will know that stream s_{26} of performer P_2 is needed by audience A_1 .

Furthermore, the broker would know how important a stream is to the particular viewer by considering its view-based and role-based priorities. Since the broker knows the position and view information of each viewer-performer pair, it can infer the view-based priority for each stream request. In Figure 5.3, for audience A_1 who has a 45 degree view

direction (we set 0 degree to be pointing to the right), the CF of s_{26} is $\cos(0^\circ) = 1.00$ because s_{26} also shoots with a 45 degree direction. CF of s_{25} for A_2 is $\cos(45^\circ) = 0.71$ because there is a 45 degree difference between A_2 's view direction and s_{25} 's shooting direction.

The determination of role-based priority, on the other hand, depends on how it is defined as we discussed earlier. Here, we use the three viewer roles in the discussion as examples. For the first two determination methods: subjective and uniform role-based priorities, the broker needs no extra computation to get the priority numbers since they are already specified in the subscription request (the interest field). As for the last objective determination method which is based on distance, the computation requires the positions of the viewer and the performers in the virtual space. This information is already stated in the subscription requests (the state field). Thus, the role-based priorities of all the subscribers are also known by the broker.

5.5 Forest Construction

After the stream requests and the hierarchical priorities are determined, the final job of the broker is to construct the content dissemination forest and then notifies the subscribers from whom should they receive the streams. The content dissemination forest decides the efficiency of stream delivery. Under the pressure of massive audience population, bandwidth-consuming streams, and delay bound, an efficient dissemination network would help preserving the service quality under the limited resource.

The objective for our forest construction algorithm is to construct a set of directed trees (a forest) in the P2P network among the participating sites (every performer and every audience site is included). Each tree connects all the subscribers who require the same stream with the publisher (performer) of that stream as the root. An example is provided in Figure 5.4b. In the figure we simplify the situation by assuming each performer only produces one stream. Under resource limitations, often times not all the stream requests can be admitted. Thus, construction of an efficient forest which preserves a low request rejection ratio is crucial.

Another important metric that reflects the efficiency of a forest is its resulting application quality of service (AQoS). AQoS is a weighted version of the admission ratio of stream requests. From hierarchical priorities, we know the importance of a particular stream to its subscriber. The more important the stream is, the higher its 'weight' will be. Thus, the AQoS metric essentially shows whether the forest construction algorithm can identify the

importance of the content and whether it can choose the less important ones to discard when there is not enough resource. In the following, we first formulate the forest construction problem and then introduce our forest construction algorithm. The construction algorithm handles the initial dissemination forest at the beginning of a session. Since our use case mimics a theater play in reality, we can assume that most of the passive users join the audience together from the beginning of the performance. Thus, the algorithm collects priorities from these initial audiences and constructs the initial dissemination forest. Another forest adaptation algorithm designed to deal with user churn (join and leave) after the session begins will be introduced afterwards.

5.5.1 Problem Formulation

The problem contains two constraints and two optimization goals.

Bandwidth Constraint. For each participating 3DTI site U , there is a limit on the inbound and outbound bandwidth of its local gateway machine, denoted as I_U and O_U . These limits in practice can be measured by various probing tools (e.g., Pathload [Jain 2003]). The total bandwidth consumed by streams received by the site must not exceed the limit I_U ; and the total bandwidth consumed by streams going out from the site (including those produced by the site itself and those relayed by the site) must not exceed the limit O_U .

Latency Constraint. To preserve the real-time property of 3DTI service among performer sites, an end-to-end latency bound D_I is placed on the content delivery to a performer site. As for audience sites, since the users are not interactive, the latency bound, denoted as D_P , is typically larger than D_I . The P2P overlay network among participating sites is a complete graph with a cost on each of its edges. The cost denotes the time delay for a stream to travel from one end to the other. These costs are estimated by the geographic locations of the sites by the empirical mapping provided in [Feldman 2007]. In forest construction, the total cost of a delivery route of a stream must not exceed the latency bound.

Minimizing the Stream Request Rejection Ratio. Any stream request that violates either one of the mentioned constraints will be rejected by the session manager site at forest construction. Since every request rejection implies potential degradation of the final service quality, our first goal of the construction is to minimize the number of rejected stream requests. Thus, we define the Request Rejection Ratio of streams as the number of rejected stream requests (N_R) over the total number of stream requests (N): N_R/N .

Maximizing the Application Quality of Service (AQoS). With the view-based and role-based priorities provided, we define the hierarchical priority (hp) of a stream request as the product of 1) the priority number of the stream in view-based priority, and 2) the priority

number of the view that contains the stream in role-based priority. For example, if for a particular viewer:

$$HP(v_2) = \{0, 0, 0, 0, 3, 4, 3, 0\}$$

$$v_2 = \{s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}, s_{27}, s_{28}\}$$

$$HP(x) = \{1, 8, 1\}$$

$$x = \{v_1, v_2, v_3\}$$

then the *hp* of stream s_{25} is $3 \times 8 = 24$. Note that if either of the priority numbers is zero, it means 1) this stream does not contribute to the view, or 2) the subscriber has no interest in the performer captured by this stream. In either case, the resulting zero *hp* signifies the session manager to ignore this stream request. We define the AQoS as:

$$AQoS \equiv \frac{\text{sum of } hp \text{ of the admitted requests in the session}}{\text{sum of } hp \text{ of all requests in the session}}$$

The resulting value is between zero and one, with one being the highest quality (i.e., the subscribers get all the streams they request), and zero being the least (i.e., none of the requested streams is admitted).

5.5.2 Initial Forest Construction

In [Wang 1996], Wang and Crowcroft prove that when a multicast routing problem is bound with two or more orthogonal constraints (in our case: bandwidth and latency), it becomes a NP-complete problem. Thus, we design a heuristic solution based on the hierarchical priorities of the stream requests. The main complication of our forest construction problem is two-folds: 1) Which request should be examined by the session manager first? 2) Whom should the subscriber receive the stream from when there are multiple holders of the requested content?

Order of Request Examination. The order of stream request examination decides the possibility for a specific request to be admitted. Intuitively, the first request being examined should always be admitted since the bandwidth of the overlay network has not been occupied by any other delivery. The later a request is examined indicates the higher the chance should it be rejected since the links could be occupied by preceding requests. Thus, we order the stream requests of all the subscribers by the *hp* of the requested stream from high to low. We examine the higher *hp* (high importance) stream request first to grant it a

higher probability to be admitted.

Selection of Sender Site. Since a stream is shared among the subscribers in the same tree via relay, there can be more than one site caching the same stream. For example, in Figure 5.4b, if A_3 has a request of stream s_1 , the potential sender sites are P_1 , P_3 , A_1 , and A_2 since they all hold s_1 . Our selection strategy can be broken down to three heuristics: similarity, residual bandwidth, and distance, ordered by the sequence of application.

Heuristic 1: Similarity. According to the user type (performer or audience) of the subscriber, it will be assigned to the sender of the same type if possible. This helps the performer sites receive streams from another performer instead of the audiences. The end-to-end latency among performers can be reduced because the performer sites will be closer to the root (the source performer) in dissemination trees with this heuristic. Since the audience sites would also receive streams from their own kind over the performer sites under this heuristic, the outbound bandwidth of performer sites can be reserved for other performers.

Heuristic 2: Residual Bandwidth. The subscriber will be assigned a sender with the most residual bandwidth. For an audience site, the residual bandwidth equals to its maximum outbound bandwidth O_U subtracted by the bandwidth consumed to relay streams. As for performer sites, the session manager has to make sure that all the requested streams can be sent out from its performer to at least one other site, or else no subscriber can receive this stream. Thus, the residual bandwidth is further deducted by a reserved bandwidth. The reserved bandwidth of a performer site is computed as the sum of bitrate of streams that fits all the following three conditions: 1) The stream is produced by the performer itself. 2) The stream is requested by at least one subscriber. 3) The stream has not been sent to any other site yet. This strategy is originally proposed in [Wu 2008], which guarantees that a requested stream can be disseminated before the outbound bandwidth of its producer is saturated.

Heuristic 3: Distance. On tie-breaking of the previous heuristic, the sender is set to be the one who is closer to the root (the performer of the stream) in the dissemination tree. This shortens the end-to-end delay.

We now use Figure 5.4b as an example to demonstrate the selection of sender site. We simplify the problem in this example by assuming the inbound and outbound bandwidths of all sites are able to sustain no more than four streams (assuming homogeneity of stream bitrates). The session involves six sites in total, which includes three performer sites (P_1 , P_2 , P_3) and three audience sites (A_1 , A_2 , A_3). The *RB* in the figure indicates residual bandwidth (in number of streams it can sustain) of the site. Following the order of request examination,

let the next three stream requests to be examined by the session manager to be “ A_3 requesting s_1 ”, “ A_3 requesting s_2 ”, and then “ A_3 requesting s_3 ”. Thus, by our selection strategy, A_2 will be the assigned sender for s_1 because it is the audience site (by Heuristic 1) with the most residual bandwidth (by Heuristic 2). After this assignment, the RB of A_3 becomes three. Next, A_2 will also be the sender for s_2 for the same reason and RB of A_3 becomes two. Finally, the sender of s_3 will be A_1 because now the two audience sites (by Heuristic 1) have the same residual bandwidth (by Heuristic 2) but A_1 is closer to the root of the tree that disseminates s_3 (by Heuristic 3).

5.6 Forest Adaptation

After the initial forest is constructed following the previous algorithm, audiences can still leave and join the Amphitheater at any time. Thus, in order to preserve the efficiency of resource utilization, two forest adaptation algorithms are devised to handle audience join and leave, respectively. Before we dive into details of the algorithms, a few notations and concepts must be introduced for the ease of explanation. First, we define notation

$$u_1 \xrightarrow{s} u_2$$

to denote that user 2 is downloading stream s from user 1. Note that, this relationship implies user 2 will never get a particular frame of stream s before user 1 does because each user only downloads a stream from one source.

Second, due to propagation delay, a user cannot get the newest frame captured by the producer instantaneously. Thus, there exists a “frame elapse” between the newly captured frame and the frame being played out at the audience site. For example, in Figure 5.6, a frame of stream s takes 100ms to travel from its producer (P_s) to user u ; a frame of stream t takes 150ms to travel from its producer (P_t) to user u . If new frames are captured every 50ms (i.e., at 20 FPS), it means that when the first frame of stream s finally arrives at u , that frame is actually captured two frames before the newest frame. Hence, we define frame elapse of stream s to user u as:

$$E_s(u) = \text{newest_frame_number} - \text{received_frame_number}$$

Thus, $E_s(u) = 3 - 1 = 2$ in Figure 5.6b. Note that all subscribed streams have to be synchronized at the audience site so that they can be rendered together. In Figure 5.6, user u subscribes to stream s and t . Consequently, user u has to wait until frames of s and t that were captured at the same time to arrive in its buffer before it can start rendering (Figure 5.6c).

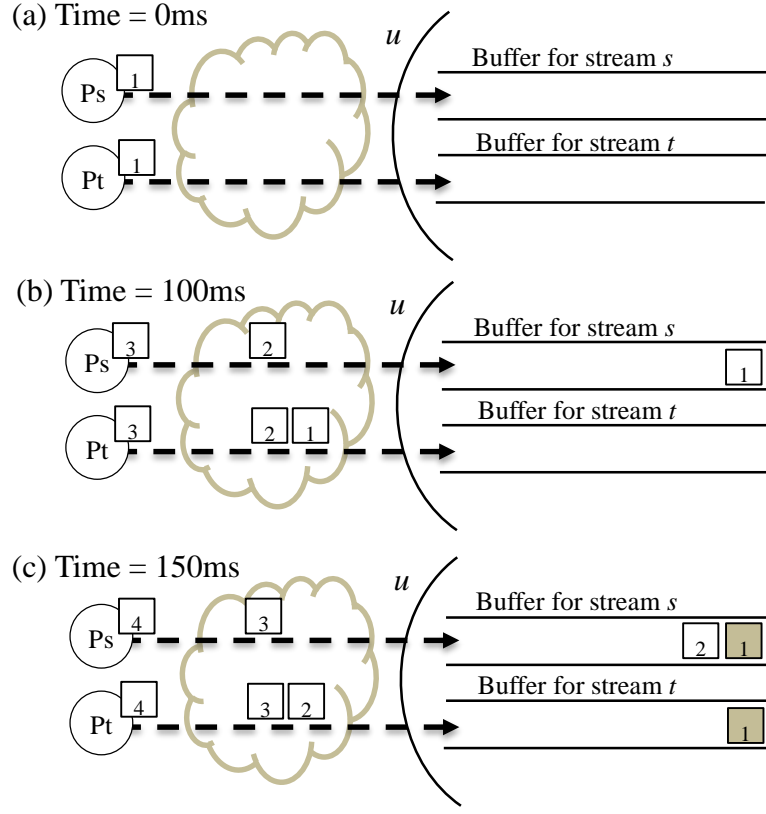


Figure 5.6: Stream synchronization.

Hence, we define the frame elapse of user u as:

$$E(u) = \max_{s \in S'} \{E_s(u)\}$$

where S' is the set of streams subscribed by user u . Thus, $E(u) = \max\{E_s(u), E_t(u)\} = \max(2, 3) = 3$ in Figure 5.6.

Now we are ready to introduce the adaptation algorithms for user join and leave. Pseudocodes of the algorithms are provided in the appendix. In the following we introduce the rationales of our designs.

5.6.1 Audience Join

From the example illustrated in Figure 5.6, we know that the delay between a join request and a frame is finally rendered consists two parts. The first part is propagation delay (0~100thms), which is inevitable and not controllable from the application layer. The second part is synchronization delay (100~150thms), which is the waiting time for the same frames

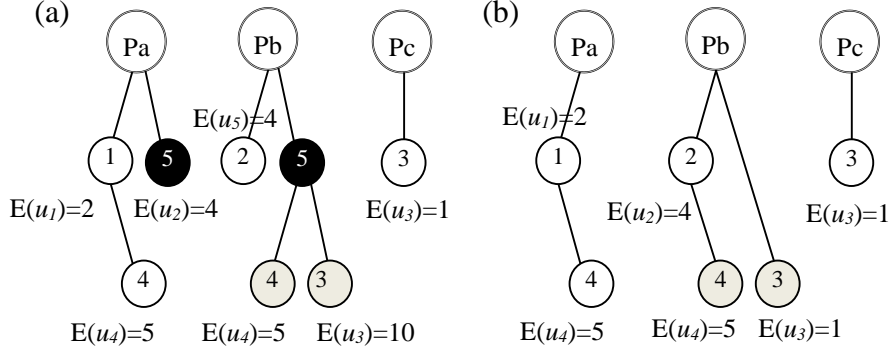


Figure 5.7: Forest adaptation.

of all subscribed streams to arrive. The synchronization delay is occurred by the multi-subscription property of 3DTI and can be minimized with proper assignment of content sources. In formal form, the synchronization delay is determined by

$$U = \{u_s \mid u_s \xrightarrow{s} u \ s \in S'\}$$

$$sync_delay = \{\max_{u_s \in U} E(u_s) - \min_{u_s \in U} E(u_s)\} / frame_rate$$

where u is the newly joined user, S' is the set of streams it subscribes to, and U is the set of sources that will provide the streams. Hence, the objective of the join algorithm is to minimize the difference between frame elapses of chosen sources. A naive way would be to let the new user always download streams directly from the producers. This way, user will always get the freshest frames and the frame elapses of sources are likely to be similar. Yet this method will soon saturate the outbound bandwidth of producer site. Thus, the new user should always download a stream from an existing user if possible.

The algorithm is comprised of two phases. The objective in the first phase is to ensure the freshness of content. We want the new user to be downloading the freshest content possible. We first select the user who holds the freshest content in each dissemination tree of each stream that the new user intends to subscribe. For example, in Figure 5.7, $S' = \{s_a, s_b, s_c\}$ and the dissemination trees are illustrated. The roots are producers and the other nodes are existing users. In this case, the users who hold the newest content in each tree would be u_1 (holding s_a), u_2 and u_5 (holding s_b), and u_3 (holding s_c). Note that the selected users must have available bandwidth to send stream to the new user. If not, we select the one with the second freshest content and so on. If there is no user in a tree who has available bandwidth to share, the new user will directly download from the producer.

Assume that in Figure 5.7a the bandwidth of u_5 is already saturated, thus the list becomes $\{u_1, u_2, u_3\}$.

The objective of the second phase is to minimize the synchronization delay. By definition, this equals to minimizing the difference between the largest and the smallest frame elapses among the chosen sources. Hence, we start from the source with the largest elapse in the list provided by the first phase, and gradually include new sources that minimize current synchronization delay until all the subscription requests are fulfilled. Using Figure 5.7a as an example, the list provided by phase one is $\{u_1, u_2, u_3\}$. We start from choosing u_3 as the source of s_c because $E(u_3) = 10$ is the largest elapse in the list. Second, we choose u_3 also as source of s_b because this minimizes the synchronization delay ($10 - 10 = 0$). Last, we choose u_4 to be the source of s_a because it is the subscriber of s_a that has an elapse closest to the other chosen sources. Thus, the final synchronization delay is $(10 - 5) / \text{frame_rate} = 250\text{ms}$.

5.6.2 Audience Leave

Audience leave event can be classified into normal leave or abnormal leave. Normal leave is when the leaving user notifies the session manager before it disconnects, so that the manager can handle the children of the leaving user in the dissemination trees. Abnormal leave is caused by unexpected termination of the audience site. In this case, the children have to detect the incident and notify the session manager to be reassigned a new parent. Detection of parent failure is done by standard heartbeat approach by treating the incoming frames as keep-alive messages.

For orphan reassignment, we want the transition to its new parent to be as seamless as possible without any interruption in the playout. Thus, we have to find a new parent who is holding content that, after propagation delay between orphan and itself, can continue with the same frame elapse as the orphan. When the audience group is small, the probability of finding such new parent is small. In the case that no suitable new parent exists, the orphan is reassigned to the producer.

Transition of orphan reassignment is illustrated from Figure 5.7a to 5.7b. Assume propagation delay will cause the frame elapsed to be increased by one (i.e., if $u_x \xrightarrow{s} u_y$ then $E_s(u_y) = E_s(u_x) + 1$). In Figure 5.7a, if u_5 leaves, it makes u_3 and u_4 orphans. After u_3 requests to be reassigned, it will download s_b directly from the producer since no existing user can continue its frame elapse. After reassigning u_4 , it will be downloading s_b from u_2 (Figure 5.7b).

5.7 Evaluation

The evaluation of our system is three-folds. First, we evaluate the overall performance of the Amphitheater with hundred-scale audience group. Second, we focus on the service quality of the performer sites. Third, we verify the effectiveness of virtual seat design on improving the efficiency of stream dissemination. To show the effectiveness of semantics-awareness, we compare Amphitheater with previous 3DTI systems ([Arefin 2012] and [Nahrstedt 2011]) in our evaluation. In the following sections, we first introduce the settings of our experiment testbed before we continue to the evaluations result analysis.

5.7.1 Simulation Settings

We evaluate the Amphitheater using a discrete event simulator (DES [Robinson 2004]). Following are the setting details of our simulation.

Network Settings. We adopt real-world topology from [Netmap] as the testbed of our simulation. Among the 1,092 real hosts distributed around the world in the Netmap database, we randomly picked 3 to 1,000 hosts to be our participating 3DTI sites. The host-to-host delay is estimated based on the geographic distance between them [Feldman 2007]. We set the maximum inbound (I_U) and outbound (O_U) bandwidths of a site to be random values in the range of 40~150 Mbps.

Site Settings. Each participating performer site is equipped with a camera array with eight 3D cameras shooting from octagonal positions around the user space. Each camera produces a 3D video stream with a 5~10 Mbps bitrate. In the simulation, the bandwidth consumption of a stream is set to be a random number in that range.

Latency Settings. The latency bound is set to be 100 ms (D_I) for performer sites and 5 s (D_P) for audience sites. Contents should be delivered with an end-to-end delay exceeding the bound will not be admitted by the session manager.

5.7.2 Broadcast with Hundred-Scale Audience

In this part of the evaluation, we evaluate the overall performance of the Amphitheater as a hundred-scale broadcast system of 3DTI. We investigate the effect of the size of performer crew and the size of the audience in the two parts of this experiment. The simulation results are compared with 4D TeleCast [Arefin 2012], a prior 3DTI dissemination system without semantic-awareness.

Performer Settings. On the circular stage of the Amphitheater, we randomly assign the standing positions and the view directions of the performers. The role-based priorities

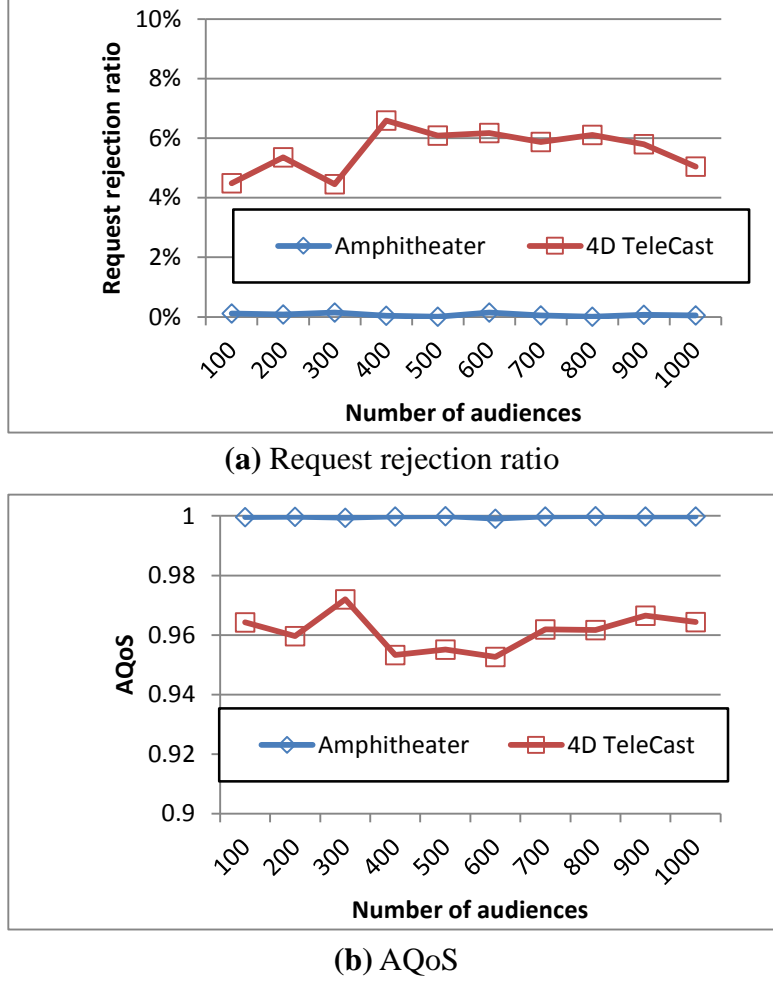
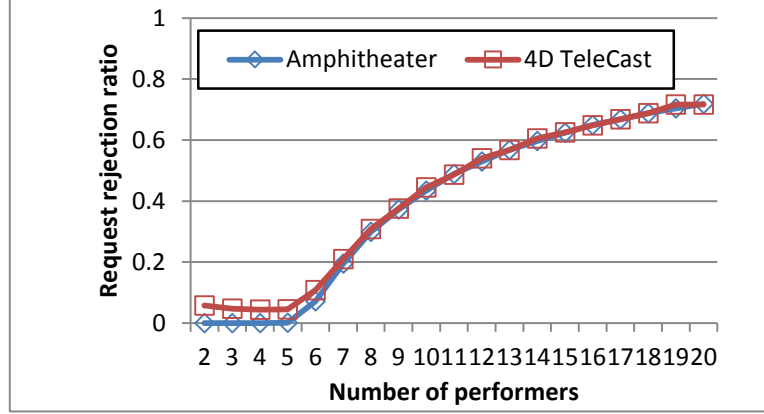


Figure 5.8: Broadcasting with fixed performer sites.

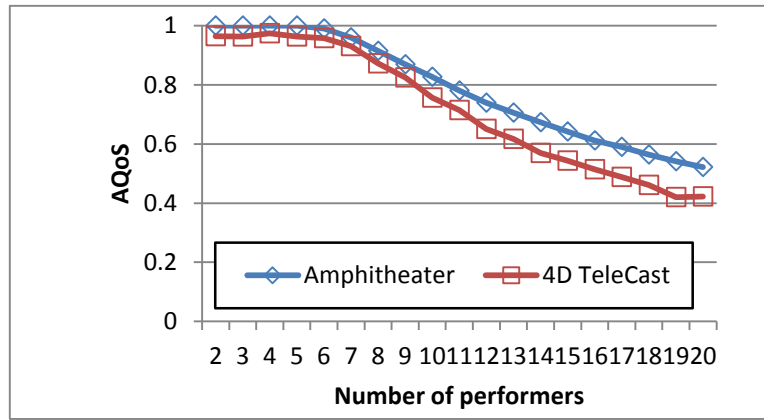
are set according to the distance between the viewer and the targeted performer.

Audience Settings. The audience is put in virtual seats, which surround the stage and are evenly dispersed. We set the views of the audiences to be pointing at the center of the stage and the field of view covers the whole stage (the same as audience A_2 depicted in Figure 5.3a.) The role-based priorities are set according to the ‘popularity’ of the targeted performer. In the performer crew, we set 25% of the performers as being ‘popular’, 50% of them being ‘average’, and 25% of them being ‘subordinate’. The role-priorities assigned by the audience are random variables. We set the average value of the assigned role-priorities to be the highest for popular performers; and the lowest for subordinate performers.

Simulation Results. In the first part, we fix the number of performer sites at five and gradually increase the total number of audience sites from 100 to 1,000. The results are plotted in Figure 5.8 with the performance of the Amphitheater being the blue-diamond curve and 4D TeleCast being the red-square curve. We can see that the number of the



(a) Request rejection ratio



(b) AQoS

Figure 5.9: Broadcasting with fixed audience sites.

audience sites does not affect the rejection ratio significantly. Intuitively, more sites participating in the session implies taller stream dissemination trees (since the outbound bandwidth is limited) and hence longer latency. However, since non-interactive audience has a much higher tolerance towards delay ($D_p = 5s$), the growth of the audience group does not increase the stream request rejection ratio in both systems. The rejection ratios are lower than 10% for both cases and the AQoS are higher than 0.9 due to the high request admission rate.

Comparing the performance of the two systems from Figure 5.8, we can see that even with abundant resource, 4D TeleCast still has a slightly inferior performance. Since user semantics (e.g., role-based priority) is not identified in 4D TeleCast, its scheduling algorithm assumes equal importance of all performer sites and will drop a viewer (by rejecting all of its stream requests) when it cannot receive at least one stream from each performer. On the other hand, since our examination order of stream requests addresses the user semantics (with role-based priority), if a viewer does not receive any stream from a particular performer,

it is because she has marked that performer as unimportant with a low role-based priority number. The all-or-nothing design of 4D TeleCast oversees the fine-grained hierarchical priority of streams. This contributes to an inferior performance when the user group grows. When the size of audience reaches 1,000, the Amphitheater sustains 1,010 more stream requests than 4D TeleCast with higher AQoS.

In the second part, we conduct another set of simulation with a fix number of audience (500 sites) and performer crews with different sizes (two to twenty). The results are shown in Figure 5.9. We can see from the figure that our forest construction algorithm is able to identify the important requests and to assign resource accordingly. With no significant difference ($< 6\%$) between the two systems on request rejection ratios, our algorithm is able to reach a higher AQoS. The improvement from adopting our algorithm grows with the increasing pressure of resource limitation. When the number of performers reaches twenty, the total bitrate of streams created by the whole performer crew is 1,600 Mbps, and our algorithm achieves 24% higher AQoS than 4D TeleCast.

5.7.3 Service Quality of Performers

In this part of the evaluation, we focus on the performer sites and evaluate their service qualities. The quality of a performance depends largely on the quality of interaction among the performers on stage. We compare our result with the forest construction algorithm used in the framework proposed in [Nahrstedt 2011], which assumes every participating site is immersive and hence every user (in a small user group) is a performer in their scenario. Thus, in the simulation, we accommodate the Amphitheater to this scenario by setting the auditorium to be empty. There are only interactive users in the session.

Performer Settings. The evaluation contains two parts. In the first part, we set the stage at a virtual play, which contains fewer than twenty performers, and the role-based priority is set according to their distance to the viewer in the virtual world. In the second part, we set the stage at a sport arena. Where fewer than ten performers (athletes) are in the arena and the role-based priority of a viewer is set to be uniform. In both scenarios, the performers are placed in random positions on the stage and each of them has a randomly set view direction.

Simulation Result. Since all participating sites are performer sites, intensive stream exchanges and hence massive bandwidth consumption in the overlay network are well expected. The results of virtual play and sport arena are presented in Figure 5.10 and 5.11, respectively.

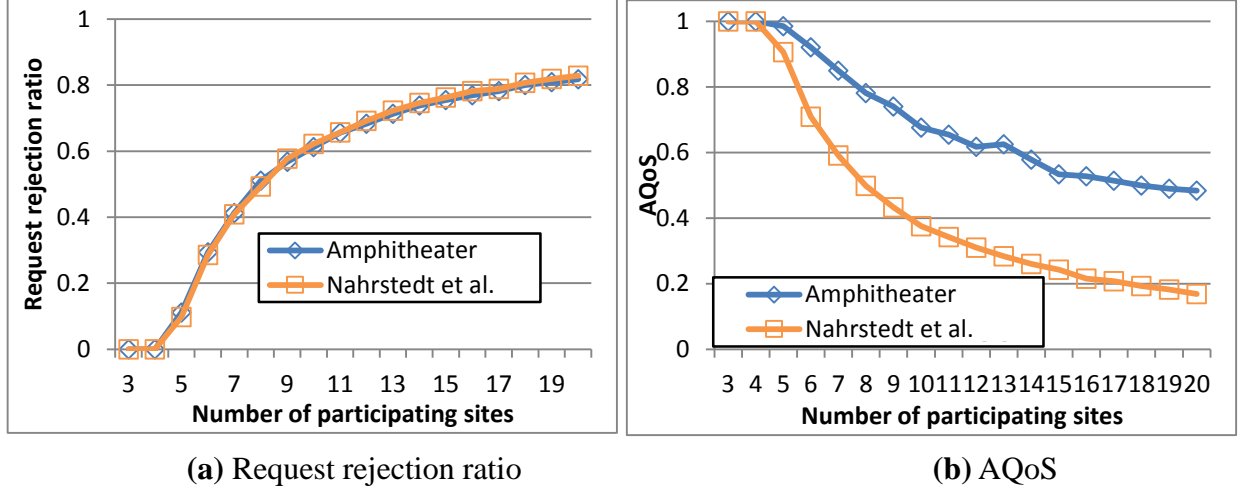


Figure 5.10: Simulation results of virtual play.

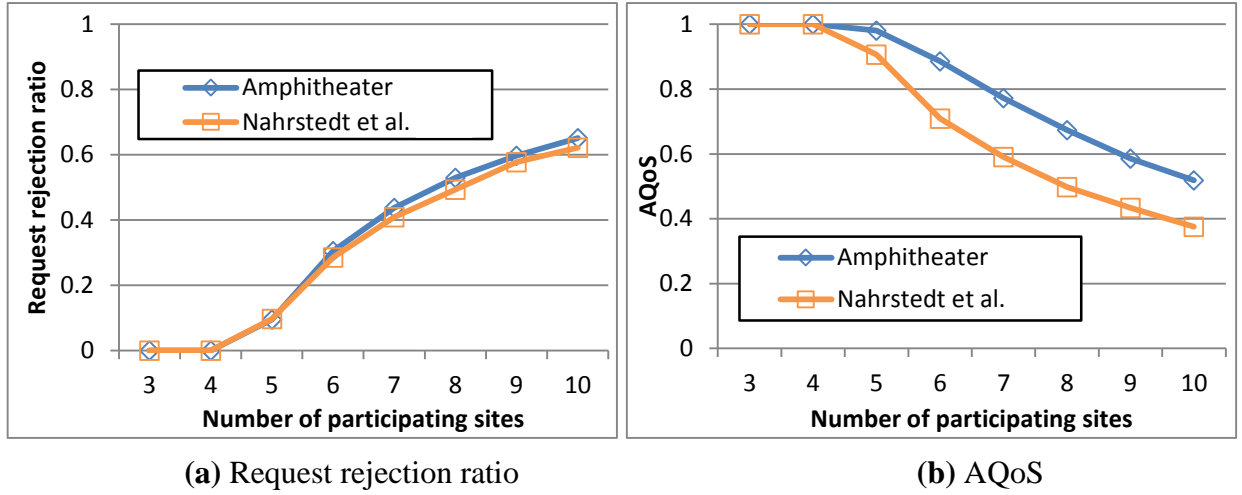


Figure 5.11: Simulation results of sport arena.

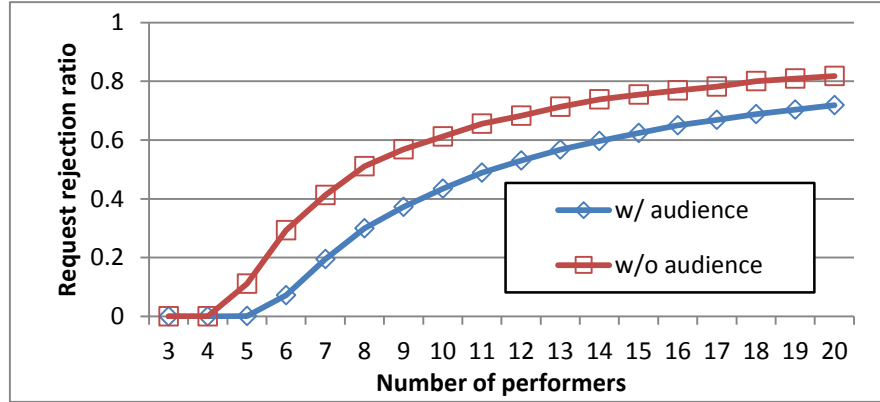
In Figure 5.10a and 5.11a, the request rejection ratios of the two algorithms are plotted against the number of participating sites with the results of the Amphitheater being the blue-diamond curve and [Nahrstedt 2011] being the orange-square curve. First, we can see the ratios increase along with the number of the participating sites in both cases. Under constant available networking resource, the increasing number of sites is introducing more stream requests that cannot be admitted due to the lack of bandwidth or due to violation of latency constraint. From the figure we can see that for cases with fewer than five participating sites, the network can sustain all of the requests addressed by the users. After that, the rejection ratio rises gradually. The rejection ratio is nearly 80% when the number of sites reaches twenty in the virtual play; and 60% when it reaches ten in the sport arena.

Second, comparing the performance of the two systems, we can see the resulting

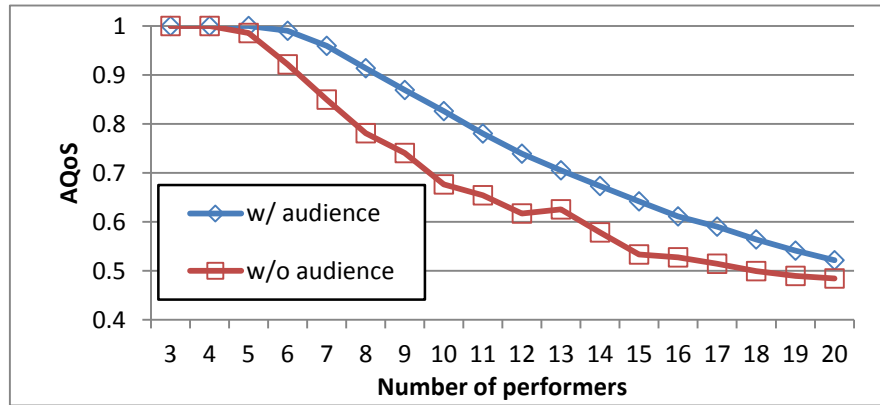
rejection ratios are very similar but the Amphitheater achieves a higher AQoS with the help of user semantics (Figure 5.10b, 5.11b). Since AQoS is actually a weighted version of the admission ratio of stream requests, the curves of AQoS have opposite shapes of rejection ratio. As the number of sites increases, AQoS drops intuitively. Comparing the two algorithms, we can see that our algorithm outperforms [Nahrstedt 2011] by a factor of x2.8 when the number of sites reaches twenty in the virtual play, and x1.4 when the number of sites reaches ten in the sport arena. This shows that, although the two systems has rejected the same amount of stream requests (Figure 5.10a and 5.11a), our Amphitheater is able to identify the semantically more important streams and assign higher priorities to them when the resource becomes scarce.

5.7.4 Effect of Virtual Seats

In the last part, we verify the effectiveness of virtual seat design in improving the efficiency of content dissemination. As we discussed in earlier sections, the surrounding seat arrangement is expected to help the dissemination of streams with content sharing and



(a) Request rejection ratio



(b) AQoS

Figure 5.12: Performance with and without audiences.

distribution. To verify these advantage brought by the virtual seats, we simulates two Amphitheatres. One with 500 audiences in the virtual seats, and one with zero audience. Other settings of this simulation are the same as the first part of the evaluation.

Simulation Results. We plot the result in Figure 5.12, where the blue-diamond curve stands for the Amphitheater with audience sites and the red-square curve stands for the Amphitheater without audience sites. We can see from the figure that the performance is generally better when there are audience sites participating the session. Since the audience sites do not produce content, they play the role of hubs in the dissemination network. Recall that in our algorithm, a performer site will turn to the audience sites to receive requested streams when the other performer sites are saturated. Thus, when there are audience sites in the session, this mechanism lowers the request rejection ratio of performer sites (and hence increases the AQoS) because more senders are provided to receive the stream from.

5.8 Conclusion

In this chapter we present the 3DTI Amphitheater, a live broadcasting system for dissemination of 3DTI content. In order to utilize the limited networking resource more efficiently so as to enhance the scalability of applications, we identify user semantics and utilizes it in the construction of the dissemination network. We design the hierarchical prioritization, which combines view-based and role-based priorities together to address user preferences and user view towards 3DTI content. With semantic awareness, networking resource in the P2P overlay can be utilized more efficiently and hence improving the broadcasting quality. The Amphitheater is tested by large-scale simulation based on real world network settings and configurations of real 3DTI system. Result shows that the Amphitheater outperforms prior 3DTI systems with an enhanced AQoS while sustaining the same hundred-scale audience group.

Contributions from our investigation in user-semantics-aware content dissemination system for 3DTI can be summarized as follows.

- Design and verification of a new live broadcast framework of 3DTI: the 3DTI Amphitheater, which makes use of the audience users as more efficient relay sites to aid the dissemination of 3D streams.
- Identification of role-based priority, which improves the utilization of limited bandwidth by giving higher priorities to the streams which are semantically more important to each user.

- Introducing the concept of hierarchical prioritization, which addresses the user semantics related to user preferences and user view. This fine-grains the differentiation of streams and improves the resulting AQoS by enabling a more efficient dissemination forest construction.

On the semantic level, user semantics including 1) user type (performer/audience), 2) user view (view-based priority), and 3) user preferences (role-based priority) take part in the construction of content dissemination forest built upon the P2P overlay network. Our forest construction algorithms make sure that networking resources are first allocated to subscription requests that are semantically more important. Via this semantic-based prioritization, both service quality and service scale are improved in the Amphitheater.

On the system level, the effectiveness of user-semantic-awareness is verified in quality and scalability aspects. First, to address the user satisfaction factor, the improvement on application quality of service brought by hierarchical stream prioritization is examined via simulation experiment involving hundreds of user sites. Second, to address the resource requirement factor, we stress-test the Amphitheater on user scalability under restricted bandwidth budget and find that our system is more capable than previous prototypes. To address the content complexity factor, in different simulation scenarios we vary the number of performer sites to increase the complexity and the volume of content produced during an application session.

Comparing to previous 3DTI systems that do not have semantic awareness, results show that the Amphitheater can sustain the same number of hundred-scale users with a decreased subscription rejection ratio (1,010 more subscriptions admitted while sustaining 1,000 users in simulation). On application quality of service, the identification of view-based and role-based priorities helps boost up the AQoS of the Amphitheater by a factor of 2.8 from previous system. Therefore, we conclude that the Amphitheater successfully bridges the gap between semantic and system level in the dissemination phase of 3DTI's content delivery. With the semantics-awareness brought by the Amphitheater, 3DTI becomes a more feasible medium for live broadcasting applications.

6. CyPhy: Activity and Environment Semantics in Receiving Phase

6.1 Introduction

In asynchronous application model, content of an application is sent to an intermediary storage entity after it is created by the content producer. The content is then archived in the storage entity, pending on-demand retrieval from the viewers. Although for asynchronous applications the delay constraint is relaxed because they do not target online interaction between users, scalability and accessibility problems still emerge in the receiving phase. In this chapter, we focus on a specific asynchronous 3DTI application: remote physical rehabilitation, to investigate the role of semantic information in the receiving phase of 3DTI content delivery. We devise a 3DTI system called CyPhy (Cyber-Physiotherapy) to enable in-home rehabilitation with offline supervision from physical therapists.

Following the aforementioned application model, the content-receiving entities in the system model of CyPhy fall into two types: the storage entity and the viewer sites. At the storage entity, the *archiving* feature of CyPhy has to include 1) efficient content compression and 2) content analysis for review recommendation and/or automatic summarization. These requirements are challenging the storage and computation capability of the storage entity when the scale of the application is large. In physiotherapy scenario, daily rehabilitation exercise is recorded at home by the patient and uploaded to an electronic health record (EHR) storage cloud. Since an EHR cloud will be serving tens of clinics and each clinic will be taking in hundreds of patients, we are looking at compression and analysis of 3DTI contents in a large scale.

To fulfill these requirements, we study the repetitiveness of rehabilitation exercise and exploit its activity semantics to achieve inter-video compression. Exploiting the similarity of daily exercise recordings, our compression scheme extracts and preserves only the difference between content recorded in temporal proximity to avoid wasting storage space on redundant data. On content analysis, we focus on the anomalous activity detection in the exercise recordings to help therapists pinpoint erroneous movements and injuries of patients. Exploiting the size of predicted frames in the compressed content, anomalous events can be identified efficiently via metadata analysis.

On the other hand, challenges at the 3DTI viewer sites in CyPhy are about the

streaming feature in 3DTI content receiving. The streaming feature of CyPhy has to be adaptive in regards of 1) the available bandwidth between storage entity and the viewer site, and 2) the rendering capability and power constraint of the viewer site. In physiotherapy scenario, therapists (i.e., viewers) are not expected to have stable connection and ample computing power when they review the exercise recordings. The reviewing can be done with tablets on the move or laptops on batteries. While this accessibility requirement is taken for granted in many 2D multimedia services, the bandwidth consumption and computation intensity of 3D rendering restrict the use case of asynchronous 3DTI applications.

To tackle these challenges, we exploit the environment semantics in the receiving phase. The 3DTI viewer sites of CyPhy is aware of the semantics in its computing environment so that the rendering quality as well as the rendering workload are adaptable to the current available bandwidth, computing capability, and energy budget. With the awareness of environment semantics, the adaptation mechanism of CyPhy includes 1) DASH-based [ISO 2014] 3D video streaming and 2) offload rendering. The former addresses environment semantics regarding bandwidth fluctuation. By making the streaming feature compatible to DASH standard, the quality and the bitrate of the streamed content becomes adaptable to networking resource to preserve the smoothness of reviewing. The latter addresses environment semantics regarding computation and power limitation of the viewer's device. When inferior computation environment is detected, CyPhy offloads the computationally intensive process (i.e., 3D rendering) to more powerful computing entity (e.g., the EHR cloud) in the system.

Mapping to semantics-aware framework. As a 3DTI system targeting on asynchronous content receiving, CyPhy follows our semantics-aware content delivery framework to exploit activity and environment semantics to solve the efficiency bottleneck in the content dissemination phase (Figure 6.1 and Figure 1.4). Semantics-awareness of

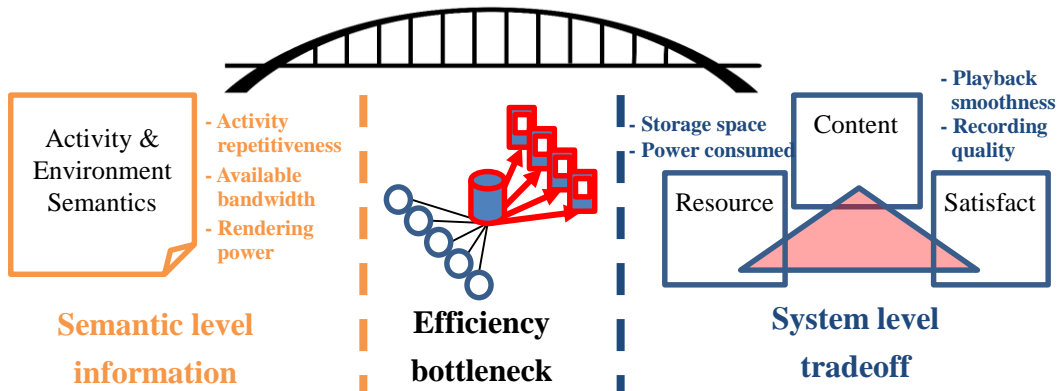


Figure 6.1: Mapping from semantics-aware content delivery framework to CyPhy.

CyPhy is reflected in both of the two major features in asynchronous receiving: content archiving (including compression and analysis) and content streaming (including adaptation to bandwidth and computing power). The designs of these features will be detailed in later sections respectively followed by evaluation on our CyPhy prototype with real exercise recordings. The effectiveness of semantic-awareness is validated by the evaluation results on its improvement on system-level tradeoffs between 1) compression ratio (resource requirement factor) and content quality (user satisfaction factor) on archiving feature; and 2) power/bandwidth consumption (resource requirement factor) and playback quality/smoothness (user satisfaction factor) on streaming feature. Due to the fact that CyPhy focuses on a specific type of content (rehabilitation exercise), the content complexity factor is rather constant at the system level.

6.2 Use Case Model

In traditional physiotherapy, both injury assessment and rehabilitation are conducted in a face-to-face manner at the clinic. A typical physiotherapy starts with a scheduled meeting between a patient and a therapist. During the meeting, therapist assesses patient's condition by physical examinations possibly with medical instruments. Prescription will be made in the end of the assessment, which contains instructions for patient to follow in her rehabilitation. During the rehabilitation phase, patient conducts daily therapeutic exercises prescribed by the therapist under supervision of trained medical staff. After a prescribed period of time in the rehabilitation phase, another meeting will be scheduled for the recovery progress to be evaluated.

This traditional procedure implies that, for patients to maximize their recovery speed via regular supervised rehabilitation with trained professionals, they would have to travel back and forth between home and clinic on a daily basis. CyPhy is designed to relieve patients from in-clinic rehabilitation and to replace it by in-home, supervised rehabilitation session. Note that CyPhy system has no intention to substitute injury assessment at the clinic because the process requires specialized medical instruments and in-person inquiries that involve haptic and kinetic measurements.

Aided by our CyPhy system, a novel physiotherapy procedure starts also with a scheduled meeting. However, as part of the prescription given in the end of the meeting, a "CyPhy kit" (Figure 6.2). will be provided to the patient. The kit includes required devices for the patient to set up a light-weighted recording studio at home. On a daily basis, CyPhy will stream to the patient a pre-recorded exercise demonstration 3D video prescribed by the



Figure 6.2: CyPhy kit: Kinects, compression suit, mat, and extra electrodes patches.



Figure 6.3: Compression suit with EMG sensors and TESSEL.

therapist. Patient will follow the video to conduct correct therapeutic exercises and have this rehabilitation session recorded with the CyPhy kit. After a rehabilitation session is recorded locally, CyPhy will upload the recording to patient's electronic health record (EHR) cloud to be archived. These recorded sessions will be played out by the therapist whenever and wherever she is available. Therapist and/or staff can supervise the correctness of patient's moves by viewing the streamed content bundle (including 3D video, skeleton, audio, and electromyography (EMG) pulse readings) and provide professional feedbacks. These recorded sessions are also used as references in the evaluation of patient's recovery progress.

6.3 System Model

The CyPhy system for remote rehabilitation comprises three major components in its system model: patient site, therapist site, and EHR cloud. The three components can be geographically separated and are connected via the Internet. In the following sections, we introduce their capabilities. We focus on functionality and performance of CyPhy. Security and privacy issues on health record keeping is not our scope at this point.

6.3.1 Patient Site

A patient site is constructed by combining the CyPhy kit given by the clinic, and patient's home PC, screen, speaker, and wireless network access point. Major hardware in the CyPhy kit (Figure 6.2) includes four Kinect cameras (with optional tripods) and a compression suit embedded with a microcontroller and EMG sensors (Figure 6.3). With the CyPhy kit, patient can set up the light-weighted rehabilitation home studio without any technical background.

Inside the kit, patient will find a mat with standing position and shooting direction of cameras marked on it (Figure 6.4). The four cameras are set to be surrounding the patient's exercise area, each placed 90 degrees apart from each other. For user friendliness, CyPhy does not require these cameras to be placed in perfect precision. Minor imprecisions in 3D point cloud capturing can be amended by automatic point cloud merging [Priorov 2014] and iterative closest point alignment [LIBICP]. Since we do not require the rehabilitation to be uploaded on-the-fly, overheads of these amending algorithms are tolerable. Another source of graphical imperfection can be infra-red interference between cameras. However, this only causes the precision to be downgraded within centimeters. Since in CyPhy we are interested in gross motor movements this minor noise incurred by interference does not incur perceptible degradation to the service quality [Narhstedt 2012].

The compression suit in CyPhy kit is for the patient to wear during the rehabilitation session. The suit avoids occlusion by normal clothing and allows the embedded EMG sensors to cling onto patient's skin. EMG sensors need to be attached to specific body spots without aid from medical staff in home environment. Thus, we embed the sensors on specified spots on the interior of the suit so that they can be deployed easily and correctly. The EMG pulses, picked up by the sensors, are collected by a microcontroller, [TESSEL],

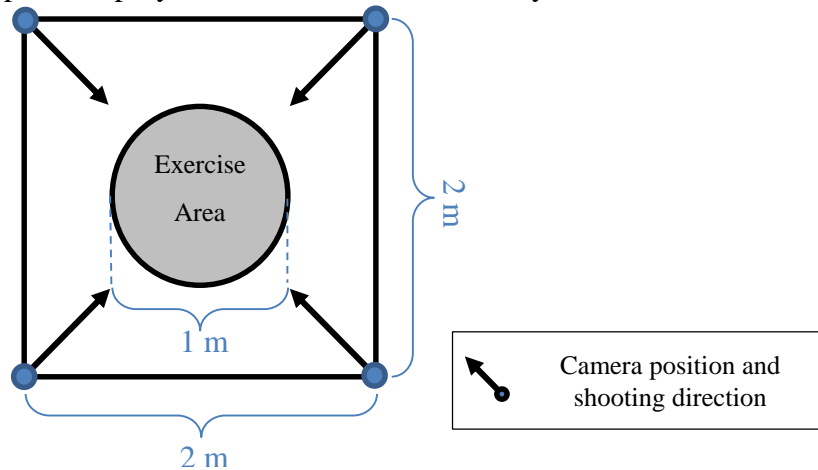


Figure 6.4: Position and shooting direction of cameras.

with WiFi capability. TESSEL is placed in a lower back pocket of the suit, wired with all on-body EMG sensors. It relays the EMG streams to patient's PC via WiFi so that they can be bundled with RGB-D and audio content captured by the camera array.

The patient's PC needs to be loaded with a CyPhy client which handles 1) streaming of demonstration 3D video of the prescribed exercise, 2) recording of the rehabilitation session, and 3) uploading the recorded session to CyPhy's EHR cloud. Uploading of the recorded session is straight forward since the CyPhy kit only generates 30 minute content every day. Thus, offline uploading can be done in the background with low bandwidth consumption. This avoids CyPhy from interfering with other networking applications in the home environment.

6.3.2 Therapist Site

A therapist site can be as simple as a PC, a laptop, or a tablet. When the therapist is available, she can request playback of rehabilitation sessions of her patient on her personal device. Content bundle of patient's rehabilitation will be streamed from the EHR cloud. The content bundle includes free viewpoint 3D video and skeleton streams, audio, and EMG readings during the whole session. The free viewpoint property helps therapist to review patient's exercise from all angles. A client software will be installed in the therapist site to handle content streaming with the DASH server in the EHR cloud.

6.3.3 EHR Cloud

The EHR cloud holds rehabilitation session recordings of patients. Multiple physiotherapy providers can share one EHR cloud. Thus, it is expected to be keeping rehabilitation recordings for hundreds of patients. EHR cloud has a two-tier architecture

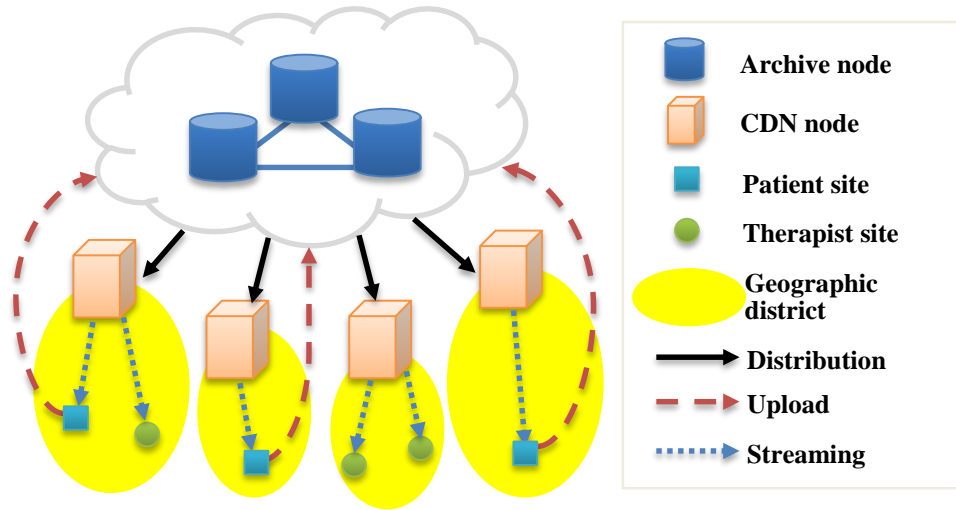


Figure 6.5: Two-tier architecture of EHR cloud.

(Figure 6.5). Machines in the first tier are archive nodes which keep long-term rehabilitation recording of patients. They provide video media compression as well as metadata analysis of the stored sessions.

Machines in the second tier are content distribution network (CDN) nodes. They share geographical proximity with viewers to ensure streaming efficiency. Unlike archive nodes, these CDN nodes only hold recordings which are more likely to be requested by their local viewers. Therefore, the primary task for CDN nodes concentrates on streaming rehabilitation sessions to viewers. This includes handling requests from viewer as a DASH [ISO 2014] server, and point-cloud-to-scene rendering offloaded by mobile clients.

6.4 Content Archiving Feature

Archiving of patient's rehabilitation involves three functions of our CyPhy system: multimodal bundle recording at the patient site; and recording compression and metadata analysis at the EHR archive nodes. In the following we detail our designs.

6.4.1 Recording Multimodal Bundle

The multimodal content bundle to be recorded during a rehabilitation session includes 1) RGB-D videos, 2) skeleton stream, 3) audio, and 4) EMG signals.

Kinect camera captures 640x480 RGB frames with D (depth) frames of the same resolution (Figure 6.6). Each pixel in the depth frame represents a depth value (distance between camera and the object) in the range between 0.4 and 4.5 m. Thus, we record RGB and D streams as two separated but synchronized videos. Skeleton stream contains 3D positions of patient's joints (i.e., shoulder, knee, hip, etc.) at every time instance. This information is extracted from RGB-D frames by Kinect API. EMG signals are captured and sent in JSON format to patient's PC, where they are bundled with the other contents.

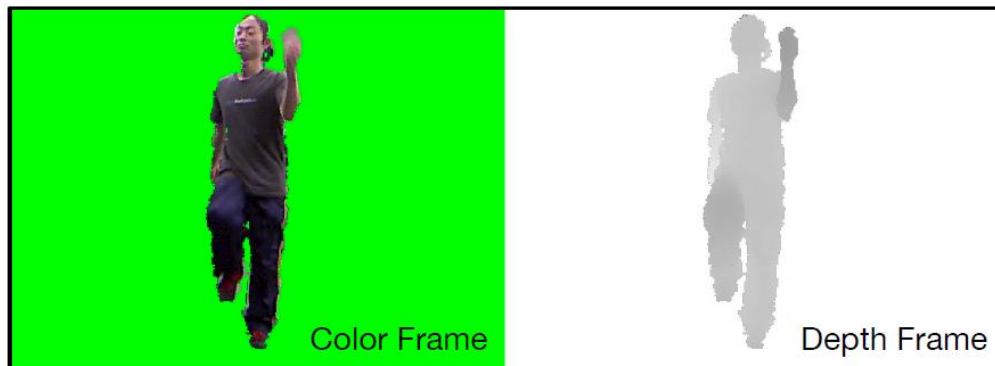


Figure 6.6: Frames captured by a RGB-D camera.

6.4.2 Content Compression

The total bitrate of a raw data bundle is in 1,100 Mbps magnitude. This implies that, without compression, each patient will upload around 250 GB of newly recorded rehabilitation session to the EHR cloud every day. Thus, an effective compression is essential since EHR cloud is designed to keep long-term rehabilitation records for hundred-scale patients. Since 99.94% of raw data in the content bundle is contributed by RGB-D videos, we concentrate on compression of visual streams. Audio stream is compressed with standard MP3 codec and skeleton/EMG streams stay in their raw data format.

3D compression gains its popularity since as early as 1998. [Mekuria 2013] provides comparison on compression ratio and computation overheads of 3D codecs up to 2013. Since most of these works inherit their approaches from computer graphics studies rather than from image processing, they choose to compress 3D visual content with triangle meshes instead of point clouds. This design decision brings substantial bitrate saving for still 3D images because it significantly lowers down the number of voxel units. Empirically, a point cloud containing 300,000 points can be represented by 70,000 triangle vertices without major quality loss [Mekuria 2013]. However, compression ratio of this approach is still far from being comparable to video codecs for conventional 2D videos. According to Mekuria's survey, the 1:10 compression ratio brought by TFAN [Mamou 2009] and [Mekuria 2013] is the finest result among mesh-based compression schemes. However, compression ratio of any MPEG [ISO 1993] codec is well known to be in 1:100 or higher magnitude. The reason behind this barrier is that these mesh-based schemes do not exploit inter-frame likeness of motion pictures. Frames in 3D video are coded by mesh-based schemes as independent, still images. Thus, mesh-based compression schemes are more akin to MJPEG [RFC 1998] compression for 2D videos, which have only 1:5 to 1:10 compression ratio due to no inter-frame coding.

A naïve approach to exploit existing inter-frame coding schemes is to directly adopt codecs from the MPEG family. Since we store RGB and D streams separately (Figure 6.6), RGB stream can be processed as regular 2D video. D stream can be processed as grey scale 2D video with 255 (white) representing the furthest distance and 0 (black) representing the nearest. To test the compression ratio of inter-frame coding, we implement a MPEG-based test codec. By encoding RGB and D streams separately like regular 2D frames, data size of a 30 minutes rehabilitation session is lowered down from 247 GB to 575 MB.

In view of the effectiveness of inter-frame coding, we further exploit the activity semantics to utilize the repetitiveness of rehabilitation exercise in video compression. We

observe that rehabilitation activities of a patient have the following properties:

1. Due to the fact that rehabilitation is a long term procedure, patient will be repeating the same exercise for many times.
2. Provided with the demonstration exercise video, patient will try to conduct exercise moves in consistent pace and motion range in every rehabilitation session with the video. In addition, since starting and ending time of recording is controlled by CyPhy client software, they will be consistent to the demonstration video.
3. Patient will be wearing the same compression suit provided in the CyPhy kit when she records her rehabilitation session.
4. In the recording, therapist only cares about the patient but not about the background. Thus, background in the video can be discarded with the help of depth information.

Combining the observations, we know that visual contents recorded in consecutive rehabilitation sessions of a patient will be very similar to each other. Therefore, we design a compression scheme specifically tailored for archiving rehabilitation recordings by compressing multiple videos recorded in temporal proximity (e.g., in the same week)

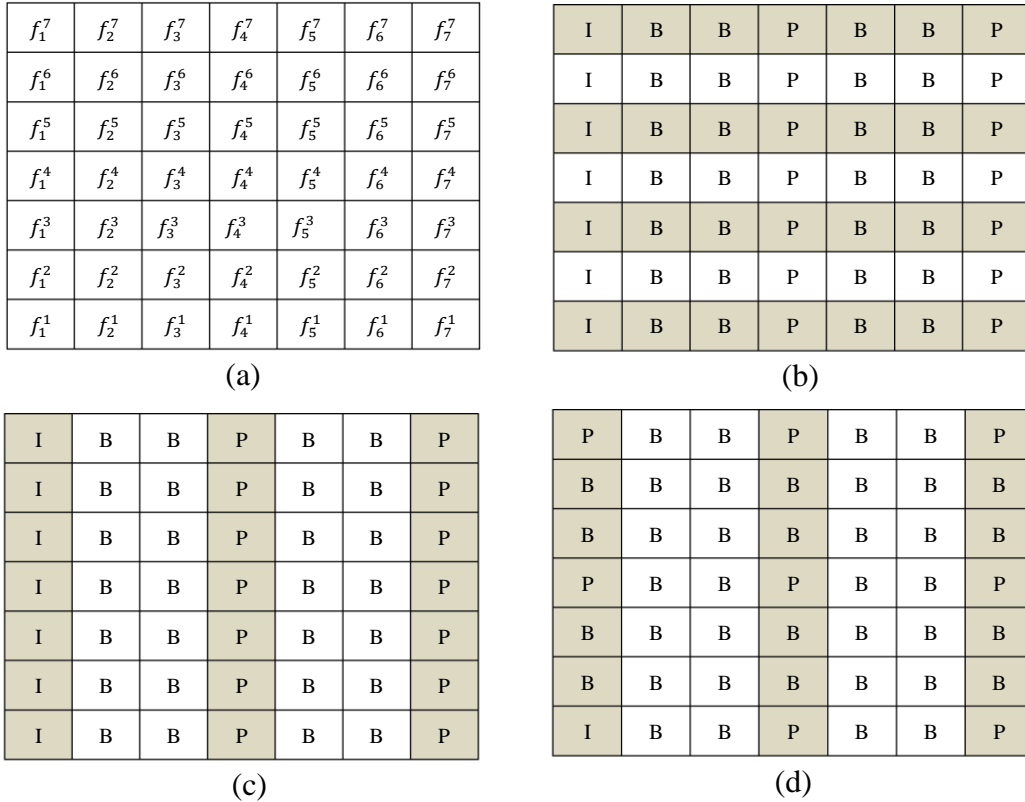


Figure 6.7: Inter-video encoding scheme of CyPhy.

together in order to exploit their similarity.

This idea is realized in our scheme by imposing inter-frame coding not only on adjacent frames in the same video, but also on videos recorded on adjacent days. An illustration of our approach is shown in Figure 6.7. In Figure 6.7a, we see the recorded raw video frames of each day can be arranged in a 2D array. Each row of frames represents frames in the same video, and adjacent rows are videos recorded on adjacent days. By applying inter-frame encoding on each video, frames will be coded into I-, P-, and B-frames

P	B	B	P	B	B	P
B	B	B	B	B	B	B
B	B	B	B	B	B	B
P	B	B	P	B	B	P
B	B	B	B	B	B	B
B	B	B	B	B	B	B
I	B	B	P	B	B	P

Step 1: Decode I-rows.

	B	B	P	B	B	P
	B	B	B	B	B	B
	B	B	B	B	B	B
	B	B	P	B	B	P
	B	B	B	B	B	B
	B	B	B	B	B	B

Step 3: Decode P-rows.

	B	B		B	B	
	B	B		B	B	
	B	B		B	B	
	B	B		B	B	

Step 5: Decode B-rows.

P	B	B	P	B	B	P
B	B	B	B	B	B	B
B	B	B	B	B	B	B
P	B	B	P	B	B	P
B	B	B	B	B	B	B
B	B	B	B	B	B	B

Step 2: Decode I-columns.

	B	B	B	B	B	B
	B	B	B	B	B	B
	B	B	B	B	B	B
	B	B	B	B	B	B

Step 4: Decode P-columns.

f_1^7	f_2^7	f_3^7	f_4^7	f_5^7	f_6^7	f_7^7
f_1^6	f_2^6	f_3^6	f_4^6	f_5^6	f_6^6	f_7^6
f_1^5	f_2^5	f_3^5	f_4^5	f_5^5	f_6^5	f_7^5
f_1^4	f_2^4	f_3^4	f_4^4	f_5^4	f_6^4	f_7^4
f_1^3	f_2^3	f_3^3	f_4^3	f_5^3	f_6^3	f_7^3
f_1^2	f_2^2	f_3^2	f_4^2	f_5^2	f_6^2	f_7^2
f_1^1	f_2^1	f_3^1	f_4^1	f_5^1	f_6^1	f_7^1

Finish: Decoded frames.

Figure 6.8: Inter-video decoding process of CyPhy.

as illustrated in Figure 6.7b. While our previous test codec stops here, we go on with inter-video encoding which further encode each column of I- and P-frames together (Figure 6.7c). Since patient is expected to have consistent moves in every video, vertical adjacent frames in a column also share great similarity. Figure 6.7d shows the final coded frames after inter-video coding. We can see that a large number of I-frames are replaced by P- and B-frames and a number of P-frames are replaced by B-frames. Thus, substantial extra compression gain is implied.

Our compression scheme, however, sacrifices efficiency on video playback. Note that, since multiple videos are encoded together, playback of one video may involve decoding of frames outside that video. Decoding steps of video are illustrated in Figure 6.8. We can see that, in order to decode one video, partial decoding of many other videos can be involved. This problem is solved by the two-tier design of our EHR cloud. In our design, this compression scheme is only used in the first tier archive nodes to achieve maximum compression ratio. Recordings in these nodes are for long-term record keeping. They are not expected to be retrieved frequently from these nodes because these records are archived for legal disputes or auditing and insurance purposes. For efficient, frequent recording playback (i.e., therapist reviewing the recorded rehabilitation), rehabilitation sessions are also stored in the second tier CDN nodes. In CDN nodes, multimedia streams are encoded independently and managed by a DASH server.

6.4.3 Metadata Analysis

Being able to analyze human activities in multimedia recordings is essential for reviewer recommendation and summary generation. In our physiotherapy scenario, these features are especially useful for back-tracking recovery history of one patient; or prioritizing supervision among multiple patients with anomaly detection. For example, setting higher review priority on patients who fell or were injured during exercise.

The large number of patients and their recorded rehabilitation sessions, and the large size of complex 3D/multimodal content bundle of CyPhy, however, make intrusive analysis (i.e., analysis on media content level involving computer vision tools and signal processing) on recorded content computationally expensive and time consuming. A broad range of previous research has been devoted to intrusive analysis on media data to identify human activity, which often requires intensive computation [Niu 2004][Sung 2011] and/or pre-knowledge of possible activity categories [Chen 2013b][Chen 2013a].

In CyPhy, we analyze metadata of the archived recordings, coded by our compression scheme, to provide preliminary anomaly (i.e., patient's abnormal behavior) analysis.

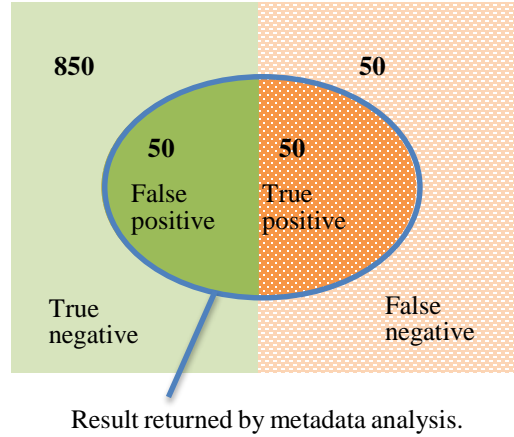


Figure 6.9: An illustration on how metadata analysis aids intrusive analysis.

Metadata analysis in 3DTI content is first proposed in [Jain 2013], where “metadata” is a set of features extracted from the running environment of 3DTI including CPU/memory usage, rendering time, and bandwidth consumption. These measurements are acquired without intrusive analysis on media content and hence are compiled and analyzed efficiently. Note that, metadata analysis does not fully replace intrusive analysis. Rather, its result is provided as a preliminary hint to help speed up the following intrusive analysis step. For example, assuming we have a slow intrusive analysis module with 100% precision and recall; and a fast metadata analysis module with 50% precision and recall. To perform anomaly detection on a dataset of size 1,000 with 100 evenly scattered anomalies, intrusive analysis alone need to go through 50% of the dataset to locate 50 of the anomalies. However, with pre-processing of fast metadata analysis over the dataset, the following intrusive analysis module only needs to go through 10% of the dataset, suggested by metadata analysis, to locate the same amount of anomalies (Figure 6.9).



Figure 6.10: Size of a predicted frame is affected by its difference from the reference.

In the case of CyPhy, our metadata is the size of predicted frames in the inter-video coding (Figure 6.7c). By the design of video frame types, a predicted frame contains only the difference between itself and its reference frames. Thus, a larger difference contributes to a larger size. An example is provided in Figure 6.10. When the predicted frame (P) is similar to its reference frame (I) as shown in Figure 6.10a, the size of the predicted frame is small because the difference is small. However, when the two frames are very different as shown in Figure 6.10b, the size of the predicted frame becomes larger. Exploiting this characteristic in video encoding, we can detect the anomalous movements by examining the size of predicted frames in inter-video compression. When a predicted frame in inter-video compression is large, it means that it is different from its adjacent video. This indicates that patient in that frame is doing a different movement from the movement she did in the videos recorded in adjacent days. This can mean patient injured or fell, or the recording system is being misused by the patient. These anomalous events are more interested in session reviewing for the therapists and thus should be marked up for summarization and review recommendation.

6.5 Content Streaming Feature

Streaming of 3DTI content is not trivial due to its free-viewpoint property. Unlike conventional 2D video, a viewer of 3DTI content can choose her view angle towards filmed

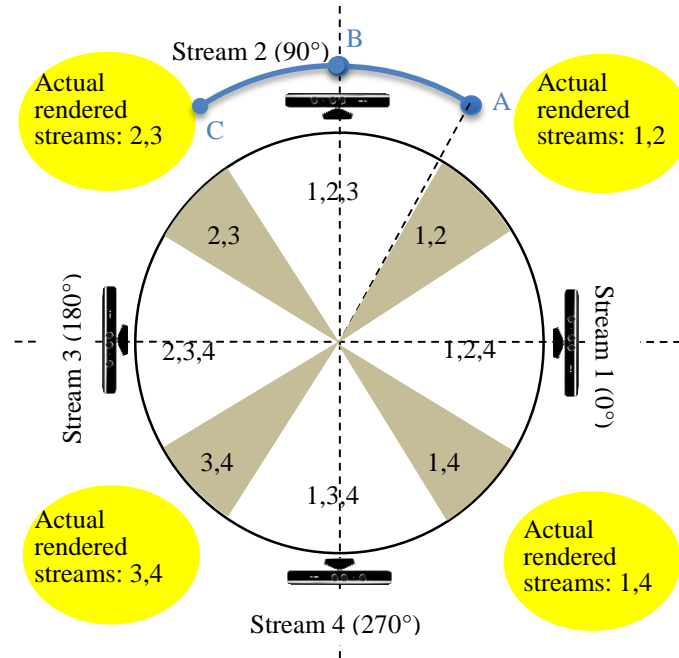


Figure 6.11: Safe zones and provision zones.

object arbitrarily during playback. Thus, we cannot directly adopt existing offline streaming standards for 3D videos in CyPhy.

There are two challenges in rehabilitation session streaming in CyPhy. The first challenge is rendering. To create a scene specified by viewer's viewing angle, CyPhy needs to merge two sets of point clouds (defined by two RGB-D streams captured by two Kinects) together and extract scene from the merged cloud. For example, in Figure 6.11, if viewer wishes to see from 30°, then streams captured by camera 1 (0°) and 2 (90°) need to be merged together to create the scene. This merging process involves computation-intensive graphic processing on-the-fly, which can bring substantial burden to power-limited portable devices. Since viewing angle ranges from 0° to 360° and is specified by viewer during playback, the rendering process cannot be done offline.

The second challenge is non-interruptive view change. Since view changing events may happen anytime during playback, an interruption may occur without provisioning. For example, when viewer changes her angle from A to C in Figure 6.11, the client software needs to switch from subscribing stream 1 and 2 to subscribing stream 2 and 3. Initiation delay for this subscription (time spent on buffering one RGB-D video segment of stream 3) will incur playback interruption.

6.5.1 Server Design

We inherit the DASH standard in our streaming server implementation. To realize adaptive streaming over HTTP, the idea behind DASH is to transcode different segments (chunks of stream) of equal duration but different qualities (i.e., different bitrates) to cater the connectivity of the different viewers. Each segment is a standalone multimedia clip that can be played independently. For example, a video segment can be a closed GOP (group of pictures). The server also keeps a manifest file, called MPD (media presentation description),

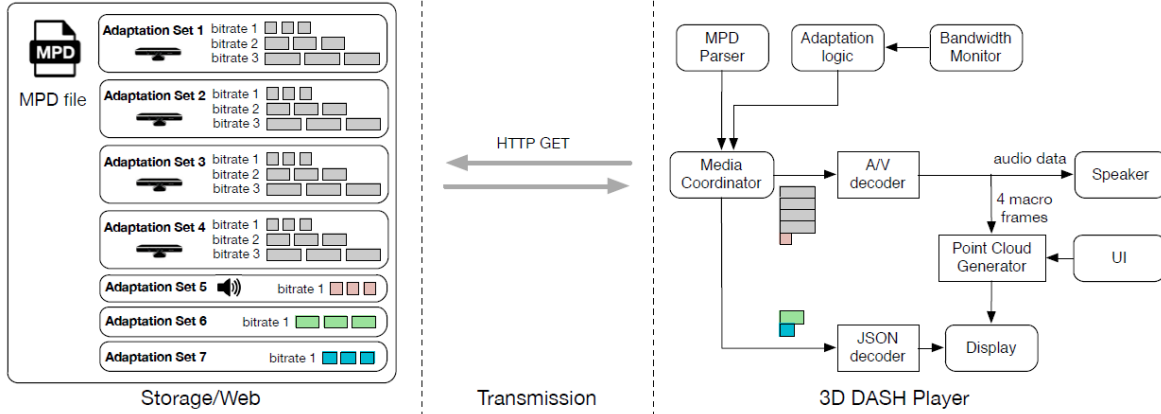


Figure 6.12: MPD file and the work flow of CyPhy player.

to list the segments it holds to the clients. A DASH client, based on its networking capability and user's preference, will request segments with suitable bitrate from the DASH server.

In our server, we exploit the use of MPD file to help client to offload computationally expensive scene rendering to the server (Figure 6.12). A MPD file in the CyPhy server will contain segments of 1) four RGB-D streams, 2) skeleton stream, 3) audio stream, and 4) EMG streams. For audio, skeleton, and EMG, a client may request all or part of them (e.g., audio stream with EMG data of the left shoulder). As for the four RGB-D streams, in the MPD file they will be marked with their shooting angles (0° , 90° , 180° , 270°) as illustrated in Figure 6.11.

When viewer chooses her view angle (e.g., 30°), her client has two options to send request, depending on whether it wants to offload the rendering to the server. The first option (Figure 6.13a) is to request existing streams in the MPD file (e.g., 0° and 90°). The server will then act like a regular DASH server to send the requested RGB-D segments. After receiving the segments, the client will conduct rendering by itself to create the scene. The second option is to offload the rendering to server (Figure 6.13b). In this case the client sends a request for a non-existing stream (30°). When the server does not find it in its MPD file, it will merge respective existing streams (0° and 90°) to create a segment that captures the scene, and then sends the rendered scene to the client.

Note that, while offloading the rendering process can save substantial computing

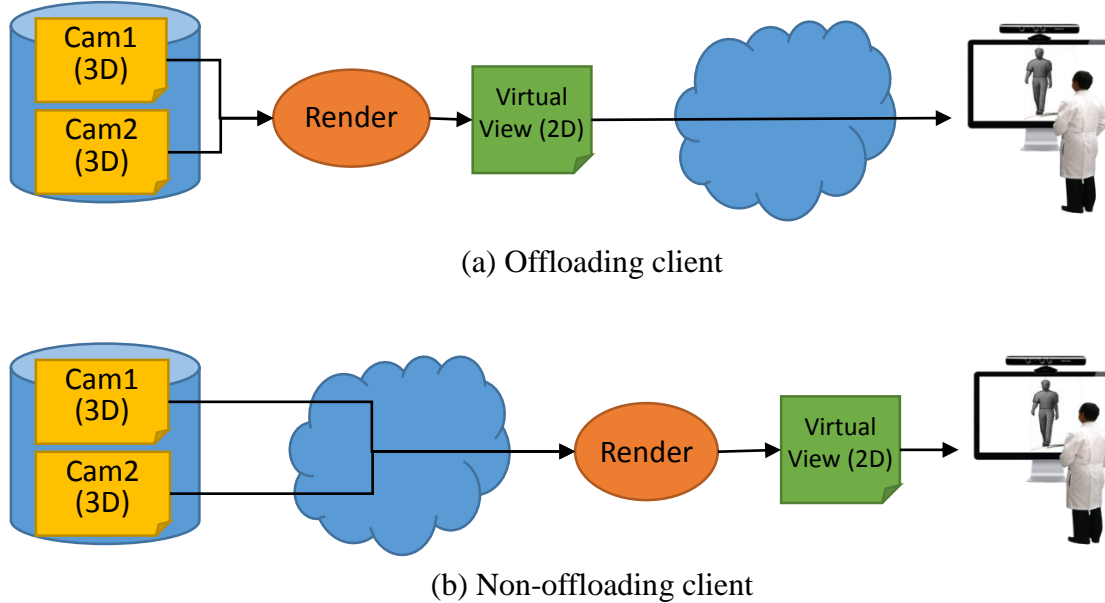


Figure 6.13: Offloading client and non-offload client. The former offload the merging of 3D scene to the server while the latter render the scene locally.

power and downloading bandwidth for the client, it will sacrifice the smoothness of session playback. We use our previous example where viewer intends to view from 30° to explain. If the client requests two existing streams (0° and 90°) without offloading, the viewer can change her view between 0° and 90° smoothly because the rendering can be done entirely locally at the client. However, if she chooses to offload view rendering to server, every time when the viewer decides to change a view, locally buffered content will become useless. A new request has to be sent and the viewer will have to endure the initiation delay during which the client stacks up its local buffer with the newly rendered stream. Therefore, the offloading option is set only to be used when client is run on mobile devices with limited resources.

Another concern on offloading is its scalability. As pointed out by Hamza and Hefeeda in [Hamza 2014], offloading rendering module to server can bring substantial computation burden when the client group is large. However, such concern is not applicable for CyPhy since it delivers healthcare sessions instead of social internet videos. Size of viewer group is restricted by doctor-patient confidentiality. Thus, offloading the rendering burden is feasible and worthwhile since it brings CyPhy's service to mobile devices.

6.5.2 Client Design

The client supports two kinds of view changing methods for viewers. The first method is to “jump” to the desired view angle. In the client user interface, the viewer can specify “ 45° ” and then the scene will switch directly to that viewpoint. The second method is gradual view change. With this method the viewer can drag the scene with a mouse or touch device to change view. This allows the viewpoint to change gradually for viewer to find her preferred view angle.

The gradual view changing method is only supported when the client does not offload the rendering function to the server. To achieve gradual view change without playback interruptions, a certain provisioning needs to be done by the client. As shown in Figure 6.11, the possible values of view angles are marked into safe zones (shaded areas) and provision zones (white areas). When user's view angle falls in the safe zones, the client will only request necessary streams (two streams) from the server. When user's angle falls in provision zones, it means that in the next time instance the viewer might move out of the current quadrant and hence new stream will be needed. Thus, the client requests an extra stream in the provision zone beforehand to avoid initiation delay. The requested streams and the actual rendered streams in each zone are specified in Figure 6.11.

Note that, the size of the provision zone decides the tradeoff between bandwidth

utilization and chance of playback interruption. During the in-between time when a viewer stays in a safe zone and when she moves to another quadrant, the client needs to finish buffering of at least one segment of the newly requested stream or else interruption will happen. Using Figure 6.11 again as an example, after viewer moves from angle A and passes angle B, she will require segment of stream 3 to render her scene. If the size of the provision zone is too small, the client would not have enough time to finish the buffering in time, which causes interruption. However, if the provision zone is too large, the client will be downloading three streams for most of the time when it actually only needs two to render each scene. To solve this issue, we limit the angle changing speed of gradual view changing method to $\phi = 90^\circ/s$ and set the size of provision zones to be 60° . This way, the client will have at least 1/3 seconds to buffer the new segment before the view passes B from A. 1/3 seconds is the playback time of one RGB-D segment in our design thus naturally the buffering time of one segment must be shorter than this.

On the other hand, the view jumping method is supported by both offloading and non-offloading clients. For offloading clients, every view change request is view jumping. It involves buffering of the new stream and causes delay on jumping events. For non-offloading clients, it does not have to buffer new streams when 1) the viewer jumps within the same quadrant from a safe zone; or 2) the viewer jumps within the same semicircle from a provision zone. Thus, the probability of interruption for non-offloading view jumps is 58% ($= 120^\circ/360^\circ \times 270^\circ/360^\circ + 240^\circ/360^\circ \times 180^\circ/360^\circ$, assuming uniform distribution of starting and ending angles of jumps).

6.6 Experiment Settings

6.6.1 Dataset

We have recorded a set of rehabilitation sessions in a home studio setting as we specified in the patient site section. The set consists of twelve recordings, imitating rehabilitation sessions a patient would conduct in twelve days. To imitate recovery progress of patient throughout the twelve sessions. We strap different amount of weights on actor's body. For example, to imitate shoulder injury, we strap 13 to 3 lb. on actor's left arm to imitate recovery progress from the first to the last day. As shown in Figure 6.14, the strapped weights incur asymmetric standing posture and tilted movements, which are common symptoms seen in patients with physical conditions.

6.6.2 Testbed

Server Settings. We use a desktop PC equipped with 4-core CPU and 8GB RAM

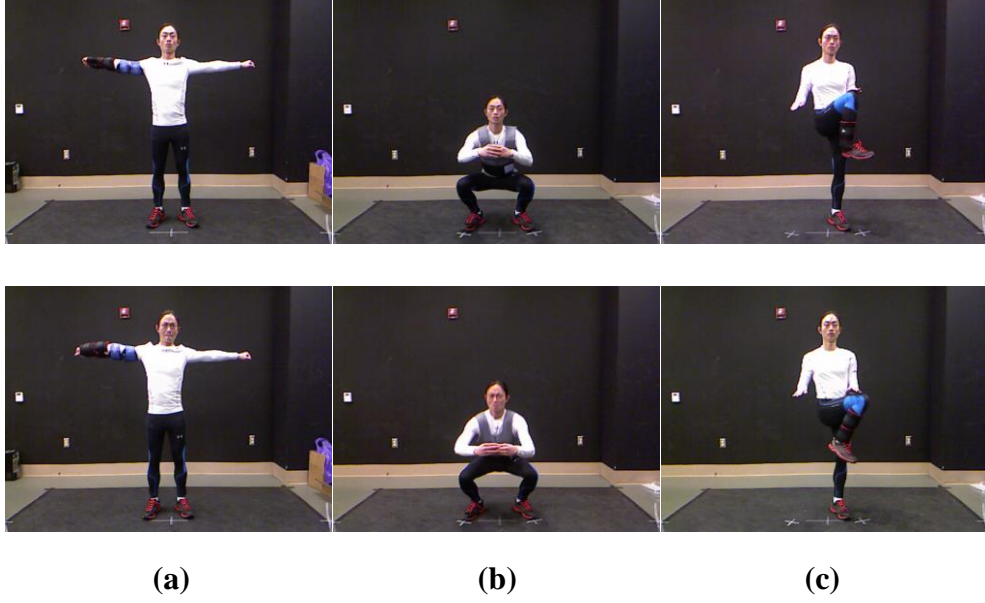


Figure 6.14: Actor’s normal movement (top row) versus movements with weights strapped on body (bottom row).

running Ubuntu 14.04 LTS to run our server. The server is driven by Node.js to serve both static and dynamic HTTP requests. When client offloads its rendering to server, it sends a dynamic request to initiate rendering on the server. When client is not offloading, the server acts like a static HTTP file server to send segments. On the server side, the recorded RGB-D streams are transcoded into two versions: high quality (4800 kbps) and low quality (1600 kbps). Each stream is segmented to comply with DASH standard.

Client Settings. To simulate client devices with different computing powers, we use a laptop equipped with 8-core CPU and 16GB RAM running Windows 8.1 as our non-offloading client; and a Nexus 4 phone running Android 5.0 as our offloading client.

Network Settings. We adopt a one-to-one server-client topology to do the experiment where we manually control the bandwidth using *tc* command on Linux. We set the maximum and minimum inbound bandwidth of a client to be 23 Mbps and 5 Mbps based on our observation in most home environments.

6.7 Evaluation

Our evaluation of CyPhy is four-folds. The first and second experiments focus on archiving feature of CyPhy, which test the compression and anomaly detection, respectively. The third and fourth experiments focus on streaming feature of CyPhy, which test the

offloading and adaptive streaming, respectively.

6.7.1 Compression Effectiveness

We compress the rehabilitation session set recorded in our studio by our compression scheme and standard MPEG, respectively. On compression ratio, our scheme achieves a 1:1255 ratio on the dataset while MPEG achieves 1:725. The difference implies that, with the same storage infrastructure, our scheme can sustain a 1.73 times larger patient group on electronic health record keeping.

The quality of our compression scheme is evaluated in two-folds. First, for color frames, we run the decoded frames on SSIM (structural similarity index) [Wang 2004] against the raw frames. To get an estimation on the quality of experience (QoE), we use the empirical mapping from PSNR to MOS for MPEG videos, reported in [Klaue 2003]. Differences between our scheme and MPEG are listed in Table 6.1. Although quality of our scheme is slightly worse (less than 10% difference) on SSIM, the MOS difference (< 0.5) indicates that, statistically speaking (i.e., assuming normal distribution of individuals' sensitivity [Neri 2010]), more than half of human viewers would not notice the quality difference. Second, for depth frames, we measure the root mean square error (RMSE) introduced by our compression scheme against the raw depth frames. Our results show an average error smaller than 3.5 cm over all compressed depth frames, which is acceptable for correctly presenting patient's exercise to therapists.

Table 6.1: Compression effectiveness of CyPhy

Compress Ratio	RGB Quality Relative to MPEG		Depth RMSE
	SSIM	MOS	
1:1255	-0.0777	-0.4444	34.6 mm

6.7.2 Metadata Analysis

To evaluate the accuracy of the metadata analysis to detect anomalies, we extract from each of the twelve recordings a 30 seconds long exercise that only involves shoulder rehabilitation (Figure 6.14a). Then, in this set of shoulder exercise recordings, we randomly inject 36 irrelevant exercises (Figure 6.14bc) as anomalies. Each injected exercise has length of 1 second (30 frames). This means that, within the original 10,800 frames of shoulder exercise, 10% of them are randomly replaced by anomalies.

In Figure 6.15 we plot the frame sizes of P-frames in the encoded set. Each column

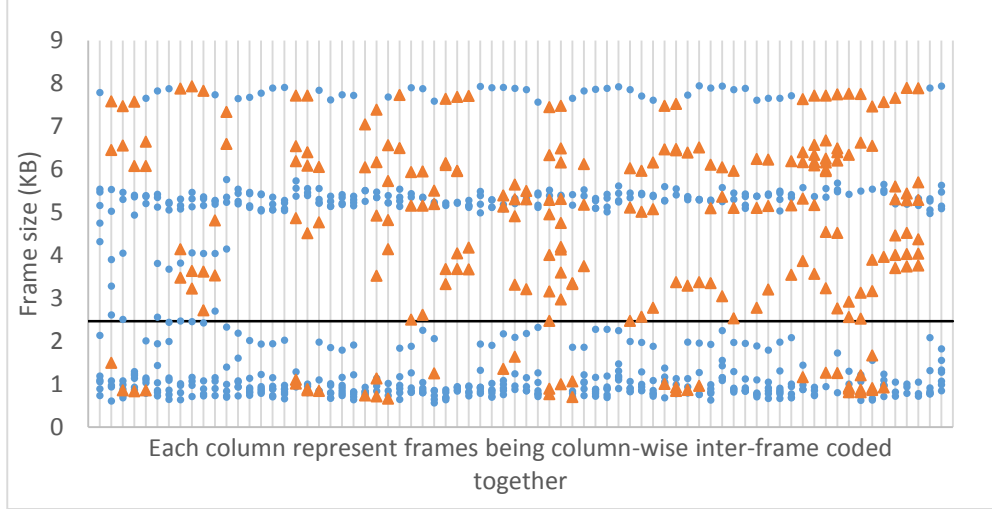


Figure 6.15: Frame size of anomaly (triangles) versus norm (dot).

of dots in the plot represent frames being column-wise inter-frame coded together. The blue round dots are original frames capturing shoulder exercise. The orange triangle dots are frames capturing anomalies. By setting a threshold at 2470 Byte (black line), we achieve an anomaly detection with 82.9% recall and 41.1% precision. This implies that, when being paired with an intrusive analysis module, our metadata analysis can help it discovers 80% anomalies when it only have to go through 20% dataset, which largely increase the speed of detection.

6.7.3 Client Offloading

In this section we evaluate the savings on streaming bandwidth and power consumption on viewer’s device with our offloading feature. We use the same Nexus 4 phone as the viewing device to watch a 30-minute-long recorded session with and without the offloading feature turned on. The results are listed in Table 6.2.

Table 6.2: Client resource consumption.

	Streaming bandwidth	Power consumption
Offload	1.56 Mbps	10.0% of full charge
No offload	6.25 Mbps	13.5% of full charge
Saving	75%	26%

On streaming bandwidth, non-offloading client needs to receive two RGB-D streams in order to render the chosen view. Yet, for offloading client, since the requested view is rendered by server, it only need to receive the final rendered frames (i.e., 2D scene that

complies with the requested view angle). Thus, the required bandwidth is substantially smaller than the non-offloading client.

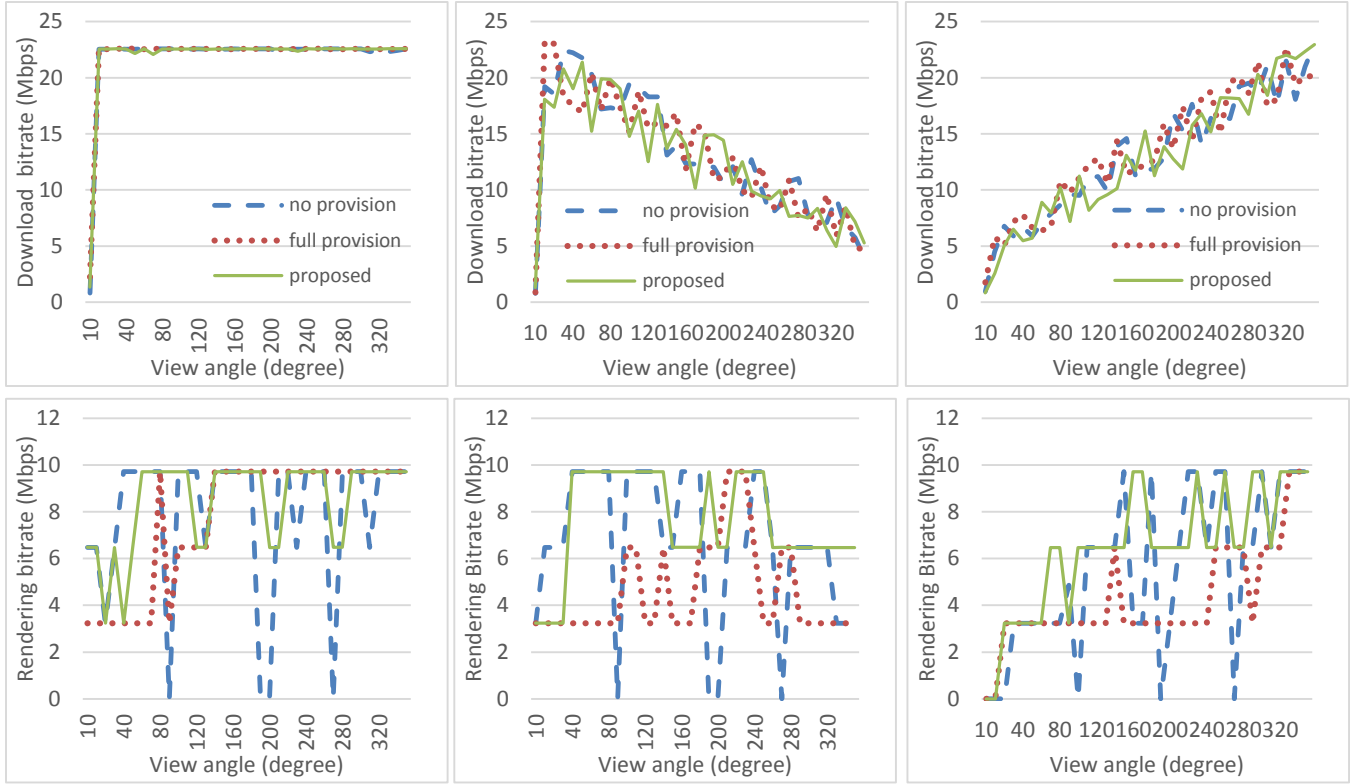
On power consumption, we see that the saving from offloading is not as much as streaming bandwidth. The reason is that our offloading feature only relieves the device from graphical rendering. Other power-consuming modules (e.g., screen backlight, session initiation) on the phone do not benefit from our offloading. Yet, our power saving still reaches 26%. We consider this a substantial improvement which makes CyPhy a more feasible service on mobile devices.

6.7.4 Adaptive Streaming

Our DASH-based design in the server helps the streaming of recorded sessions to be adaptive to 1) available bandwidth and 2) user view change. For offloading clients, the adaptation is more trivial since their view changing method is restricted to only view jumping. There is no discrimination of safe or provision zones for offloading clients and hence the adaptation only need to account current available bandwidth. Yet, for non-offloading clients, adaptive streaming accounts both available bandwidth and user's current view angle. As mentioned earlier, when user's view angle falls in provision zone, client needs to pre-buffer extra stream to avoid interruption. Therefore, in our evaluation experiment we focus on the more complicated case with non-offloading clients.

To test our server's reaction to bandwidth changing, we set up three different network condition scenarios. The first scenario is stable and abundant bandwidth. In this scenario the available bandwidth stays at 23 Mbps. This is the available bandwidth we observed in home environment with no other network applications running in the background. The second scenario is bandwidth depletion. In this scenario the available bandwidth drops from 23 to 5 Mbps. The third scenario is bandwidth recovery, in which the available bandwidth recovers from 5 to 23 Mbps. These last two scenarios simulate the effect of background traffic introduced by other network applications in the home environment. To account the adaptation due to user view change, the client in our experiment increase the view angle by 10° every second under all three network condition scenarios.

To our best knowledge, CyPhy's streaming feature provides the first multi-stream 3DTI-on-demand (i.e., RGB-D offline streaming) service which accounts both dynamic view change and bandwidth adaptation. Previous multi-stream video-on-demand systems which share similar objectives with ours are proposed in [Hamza 2014] and [Su 2014]. However, their evaluations do not account for dynamic view change during playback. Thus, due to lacking of existing subject for performance comparison, we provide two baseline



(a) Stable and abundant available bandwidth

(b) Available bandwidth depletion

(c) Available bandwidth recovery

Figure 6.16: Adaptive delivery of CyPhy.

adaptation schemes to show the performance gain from our adaptation scheme:

Baseline 1: No-Provisioning. This optimistic scheme downloads only the two RGB-D streams that will be used to render the current view. Since it downloads the minimal number of streams, each stream can be allocated more bandwidth and hence have better quality. Yet, as view angle change across quadrants, the playback will be interrupted due to lack of provisioning.

Baseline 2: Full-Provisioning. This conservative scheme downloads all four RGB-D streams anytime regardless of the current view angle. This scheme guarantees zero view-change interruption during playback. Yet, due to the large number of requested streams, each stream is allocated less bandwidth and hence have degraded quality.

In Figure 6.16, we plot the download bitrates and rendering bitrate of our adaptation scheme and the baselines, under the three network condition scenarios. As we can see from the download bandwidth plots (first row), all three adaptation schemes react to the change of available bandwidth and achieve good bandwidth utilization. This is well-expected

because, like any other offline streaming standards, our client adopts aggressive downloading. Whenever there is spare bandwidth, our client will utilize it to download ahead the current playback frame to stack up its stream buffer, regardless of provisioning or video bitrate. Therefore, our adaptation scheme and the baselines react similarly to available bandwidth change.

In the second row of Figure 6.16, we plot the rendering bitrate in each scenario, which is the total bitrate of the two RGB-D streams actually used in rendering (i.e., excluding the provisioned streams). As we can see in Figure 6.16a, when the available bandwidth is ample, all three schemes reach the highest rendering bitrate for most of the time. However, for no-provisioning baseline, interruptions happen as the view angle changes across quadrants as expected, and cause the rendering bitrate to drop to zero. The rendering bitrate of our scheme is the same as full-provisioning, except minor degradation when rendered streams change (i.e., view change across quadrants). Nevertheless, our scheme never incurs playback interruption like in the no-provisioning scheme.

In Figure 6.16b, we plot the rendering bitrates during available bandwidth depletion. Due to aggressive downloading, all three schemes are able to stack up their stream buffers in the beginning while the available bandwidth is still high. Thus, their rendering bitrates do not drop rapidly with the available bandwidth. Yet, we see the quality of full-provisioning baseline is the worst of the three because the bandwidth is not enough to download four streams in high quality. The quality of the no-provisioning baseline is the best if we omit the interruption periods. In this bandwidth depletion scenario, our scheme achieves high rendering bitrate as the no-provisioning baseline, but without any interruption like the full-provisioning baseline.

In Figure 6.16c, we see rendering bitrates of the three schemes rise with the available bandwidth. Occasionally the no-provisioning baseline has quality better than ours, but the interruptions makes it highly unstable. The quality of full-provisioning baseline rises steadily like our scheme. But due to its heavy burden to download all four streams at any given time, it reaches high rendering bitrate much later than ours.

In summary, although the utilization of available bandwidth is equally good for the three schemes due to the adoption of DASH standard, the gain of our scheme reveals in its rendering bitrate and its guaranteed smoothness (i.e., zero interruption during playback). To further quantify the gain of our adaptation scheme over the baselines, we define the *effectiveness of stream downloading* as

$$\frac{\text{amount of content begin rendered}}{\text{amount of content downloaded}}$$

Our scheme achieves 120% effectiveness comparing to no-provisioning baseline; and 168% effectiveness comparing to full-provisioning baseline, according to the results reported in the plots.

6.8 Conclusion

In this chapter we present CyPhy: an asynchronous 3DTI system targeting remote physical rehabilitation. Through the archiving requirements at its storage entity (i.e., EHR cloud) and its streaming requirements at its viewer sites, we investigate the usage of activity and environment semantics in the receiving phase of 3DTI content delivery. With its archiving feature, CyPhy provides efficient content compression and content analysis by exploiting the repetitiveness of daily rehabilitation exercise sessions. These advantages allow the CyPhy system to serve a larger user group with the same storage infrastructure of a conventional video storage entity with no semantic awareness. With its adaptive streaming feature, the bandwidth and computation demands for 3DTI content delivery become adjustable based on environment semantics. By adopting the DASH standard and the offloaded rendering, the capability of viewer's computing environment in various aspects including available bandwidth, computing power, and energy budget are considered in the streaming of recorded content. With this awareness of environment semantics, CyPhy is able to preserve the smoothness of reviewing session under available resource fluctuation and constant user view change.

In summary, our contributions with CyPhy system on semantics awareness are as follows.

A. On content archiving:

1. Effective recording of RGB-D and physiotherapy-related medical sensing data streams in home environment.
2. Efficient compression scheme based on activity semantics of rehabilitation session which enables the electronic health record (EHR) cloud to sustain long term record keeping for large patient group.
3. Fast metadata-based analysis over archived rehabilitation sessions to provide recommendations upon session reviewing.

B. On content streaming:

4. Adaptive, DASH-based 3D streaming mechanism for rehabilitation session to be delivered to viewers with different network capabilities.
5. Smooth viewpoint changing during 3D video streaming with scene rendering offloading scheme tailored to devices with different computation and power limitations.

C. Validation of the integrated CyPhy system.

On the semantic level, both activity semantics and environment semantics are utilized in the receiving of 3DTI content. Activity semantics of rehabilitation exercise (i.e. repetitiveness, same user outfit, indifference on video background) points to the similarity and synchronization of video contents. These properties allow the storage entity to increase its scalability with more efficient compression and analysis schemes. Environment semantics of viewer sites, which covers the budget of networking and computation resources, signifies the importance of quality/bitrate adjustment during reviewing session and a more balanced workload (i.e., rendering) allocation among computing entities.

On the system level, the effectiveness of semantic-awareness is verified in the tradeoffs between resource requirement and user satisfaction. Since CyPhy focuses on specify content type, the content complexity factor on system level is rather constant. In content archiving, the compression ratio addresses the resource requirement and the subjective/objective quality metrics (MOS/SSIM) address the user satisfaction. In content streaming, resource requirement is addressed by the saving on bandwidth and power on the viewer's device. User satisfaction is addressed by the rendering bitrate and the numbers of playback interruptions during the simulated reviewing session.

Results show that our storage compression scheme based on activity semantics achieves 1:1255 compression ratio, which is a x1.73 gain from MPEG-based compression commonly used for asynchronous applications. In addition, the compressed data provide metadata for fast detection in anomalous patient movement. On content streaming, we are able to offload the computation-intensive rendering function and saves 75% bandwidth and 25% power for mobile clients. By adopting the DASH standard, CyPhy is able to adjust its streaming quality according to available bandwidth as well as user's viewpoint and achieve zero playback interruption while maintaining high bandwidth utilization. Therefore, we conclude that CyPhy successfully bridges the gap between semantic and system level in the receiving phase of 3DTI's content delivery. With semantics-awareness, we demonstrate that both scalability and quality of service can be improved in asynchronous 3DTI application.

7. Conclusion

7.1 Dissertation Achievements

While 3DTI technology allows full-body, multi-view and multi-modal interaction among geographically dispersed users, the resource and quality demands of 3DTI rise inevitably, especially when advanced applications targeting resource-limited computing environments with stringent scalability requirements comes into the picture. Under these circumstances, the tradeoffs between resource requirements, content complexity, and user satisfaction in delivery of 3DTI services are magnified.

In this dissertation, we argue that these tradeoffs are avoidable when the underlying delivery chain of 3DTI takes the *semantic information* into consideration. We introduce the concept of semantic information into 3DTI, which encompasses information about the three factors: environment, activity, and user role in 3DTI systems. With semantic information, we devise semantics-aware modules which allow 3DTI systems to (1) identify the characteristics of its computing environment to allocate computing power and bandwidth to delivery of prioritized contents, (2) pinpoint and discard the dispensable content in activity capturing according to properties of target application, and (3) differentiate contents by their contributions on fulfilling the objectives and expectation of user's role in the application so that resource budgets can be allocated accordingly. We implement semantics-aware 3DTI systems that targets different use cases, different application models, and different efficiency bottlenecks to verify the performance gain from semantics-awareness on the three phases in 3DTI systems' delivery chain: capturing phase, dissemination phase, and receiving phase.

In the capturing phase: We combine user activity detection and 3DTI video content compression to reduce the resource usage in content-capturing (i.e., immersive) 3DTI user sites when the activity complexity is low. We achieve this by developing a morphing-based approach to synthesize frames in 3DTI videos and extending the technique to a quality metric: RSF, which affects both temporal and spatial resolution of a video with different levels of resource consumption. We combine the adaptation of RSF with motion characteristics of different activity semantics in 3DTI space. With a machine learning approach, the user activities can be classified in real-time. By combining the activity classification model and the quality demand model, we build up a user activity-aware adaptive capturing system for efficient content capturing, which automatically classifies the user activity and assigns suitable compression configurations that reduce 25% more networking resources

requirement without incurring perceptible degradation on visual quality.

In the dissemination phase: We devise a user-semantics-based stream prioritization method to tackle the challenge of high bandwidth demand in large scale 3DTI broadcasting. We argue that not all the streams in a 3DTI broadcasting session is equally important to a viewer. Content's importance to a viewer can be affected by user semantics including 1) view-based priority and 2) role-based priority. We present the 3DTI Amphitheater, a large scale broadcasting system for dissemination of 3DTI content. We identify the hierarchical prioritization of streams in the construction of the content dissemination forest and we validate the performance gain from semantic awareness with real world network settings and configurations of real 3DTI system. Result shows that the Amphitheater outperforms prior 3DTI systems with no semantics-awareness by boosting the AQoS while sustaining the same hundred-scale audience group.

In the receiving phase: We tackle challenges in the receiving phase of 3DTI applications including: (1) efficient compression, (2) fast content analysis, and (3) adaptive streaming and rendering. To approach them, we device an asynchronous 3DTI physiotherapy delivery framework: CyPhy, which provide solutions to all three aspects based on activity and environment semantics. On content archiving, we design a new inter-video compression scheme for 3D video archiving which exploit the activity semantics of physical rehabilitation. On content analysis for review summary generation, we analyze the metadata of inter-video coded frame sizes to detect anomaly activity events. This non-intrusive analysis shortens the detection time, which is a crucial improvement due to the large size and quantity of 3DTI recordings. On adaptive streaming and rendering, CyPhy references the environment semantics for bitrate adjustment and offloaded rendering. These designs allow the asynchronous 3DTI applications to adaptively adjust its streaming content complexity and provide different levels of user control to user devices of various computing capabilities.

In conclusion, in this dissertation we change the tradeoff between requirements, complexity, and satisfaction in 3DTI services by exploiting semantic information about the computing environment, the activity, and the user role upon the underlying delivery systems of 3DTI. Our mechanisms enhance the efficiency of 3DTI systems targeting on serving different purposes and 3DTI applications with flexible computation and scalability requirements. By the three 3DTI systems developed throughout this dissertation, we demonstrate the performance gain on scalability, quality, and accessibility of 3DTI services when semantic information of different aspects are taken into consideration by our design and implementation.

7.2 Lesson Learned

Through the three distinctive 3DTI systems, we learn some crucial guidelines in the development of a system that follows the semantic-aware content delivery framework. The first step towards a semantics-aware design is to identify the efficiency bottleneck of the system. Based on different purposes and use cases, we show that efficiency bottlenecks can exist in all phases in the delivery chain (capturing, dissemination, receiving). After the identification, in the second step one needs to view the application in the semantics level to find a cyber-physical property that could benefit the system objectives in a higher level. The third step is the most essential one, where one builds the semantics-awareness module that will take in the identified semantic information as input, and guide the underlying system to re-configure/adjust itself to cope with the needs and changes in the physical world. After the semantics-aware system is fully implemented, in the last evaluation step we need to go back to the system level to verify the performance gain in the tradeoff factors. In Table 7.1 we list the development steps for the three systems in this dissertation.

Table 7.1: Guidelines for developing a semantics-aware system.

		A3C	Amphitheater	CyPhy
Step 1: Identify efficiency bottlenecks from system objectives		Outbound bandwidth bottleneck of immersive sites	Resource bottleneck in the P2P overlay network	Computing power at system entities; connection between storage and viewer
Step 2: Find cyber-physical properties in the semantic level that leads to high level performance improvements		Motion level of user activities and user's tolerance to their quality degradation	User preferences, user view, and user role in an application session	Repetitiveness of activity; computing and networking resource budgets of user devices
Step 3: Design and implementation of semantics-aware modules that helps system adapt itself to different semantics		Activity classification module, quality demand module, and MBFS module	User-semantics-aware forest construction and forest adaptation algorithms	Compression and analysis modules; DASH-based streaming and offloaded rendering
Step 4: Verify the improvement on tradeoff factors in the lower system level	Resource requirement	Bandwidth saving	Performers outbound bandwidth	Storage space, analysis time, bandwidth/power savings
	Content complexity	Kinetic complexity	Performer crew size	Constant
	User satisfaction	User study	Application QoS	Compression quality; playback smoothness

7.3 Future Works

In this dissertation we validate the improvements on scalability, service quality, and accessibility brought by semantic information. While we use 3DTI systems to demonstrate these performance gain throughout this dissertation, we believe that the semantics-aware content delivery framework has the potential to encompass a wider variety of multimedia systems. In the following, we list three research directions we think will be promising extensions of this dissertation as future works.

Formalized semantics scripting. Towards generalization of the semantics-aware content delivery framework, we need a formalized scripting language to describe the dynamics in the cyber-physical regime to the digital computing entities. In this dissertation we see that the semantics information which helps improve the system efficiency is fetched from different cyber-physical aspects (activity, user, computing environment) in various forms (motion features, user preferences, computing capability, etc.). Thus, for further generalization and formalization, a descriptive script language akin to the semantics graphs used in AI and NLP areas would make the adoption of semantics-awareness for existing multimedia systems more convenient and expressive. In addition, since semantic graph and neural network belong to the same semantic network family [Sowa 1992], one may be able to exploit the existing deep learning frameworks to discover effective cyber-physical relations across semantic and system levels in the network in a more deterministic, non-empirical way.

Semantics-aware ubiquitous sensing. A mutual system property between 3DTI and ubiquitous sensing applications (e.g., IoT, intelligent home, self-driving cars, etc.) is their multimodality. Although in this dissertation we focus more on the data control over visual sensing contents, the prioritization and resource allocation towards the delivery of multiple heterogeneous and homogeneous streams share the same logic at the system level. Following the same semantics-aware content delivery framework, we believe that ubiquitous sensing applications can also benefit from utilizing cyber-physical properties in its sensing data delivery to improve the service quality. Semantic information can be the answers to questions in ubiquitous sensing including: 1) Which sensing data do user care more about in certain application scenario? 2) Which sensing data should have high priority in dissemination and analysis? 3) For sensing device with different restrictions in connection, power, and computing capability, how should the sensing system adapt? Should the computation be distributed to more capable entities? We believe our findings in semantic-aware 3DTI systems may be able to provide insights towards the answers.

Multi-lens systems. The advances in free-viewpoint TV, spherical scene capturing, and on-body multi-camera systems signify the thriving of research towards more elaborated multi-lens systems. Throughout the capturing, dissemination, and receiving phases of the system, performance issues (scalability, quality, and accessibility) similar to those encountered by our 3DTI systems can emerge due to the high resource demand on video delivery and rendering. Since the representation of visual content in 3DTI is also multi-view, we believe that most of the techniques developed in this dissertation can be adopted in the delivery frameworks of these new multi-lens systems. A successful example of adopting view-based user semantics (i.e., view-based priority) in the dissemination of spherical video has recently been announced by Facebook in its 360 video/VR feed feature [Facebook 2016]. We develop a transcoding filter with Facebook, which produces different version of spherical videos with enhanced quality in different viewports. So instead of pushing all content in maximum quality to the viewer, we can tailor the dissemination according to user's view to reach maximum service quality while preserving low bandwidth usage.

To sum up, we foresee many opportunities to apply the semantics-aware content delivery framework in the design and evaluation of more general multimedia systems. With the guidelines learned from the development of this dissertation, we expect application practitioners are able to systematically evaluate, design, implement, and verify a multimedia system centering on the semantics information.

References

- [Agarwal 2010] Pooja Agarwal, Raoul Rivas Toledano, Wanmin Wu, Klara Nahrstedt, and Ahsan Arefin, “Bundle of Streams: Concept and Evaluation in Distributed Interactive Multimedia Environments”, in Proceedings of IEEE International Symposium on Multimedia (ISM), 2010.
- [Arefin 2012] Ahsan Arefin, Zixia Huang, Klara Nahrstedt, and Pooja Agarwal, “4D TeleCast: Towards Large Scale Multi-Site and Multi-View Dissemination of 3DTI Contents”, in Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS), 2012.
- [Arefin 2013] Ahsan Arefin, Raoul Rivas, and Klara Nahrstedt, “Prioritized Evolutionary Optimization in Open Session Management for 3D Tele-Immersion”, in Proceedings of ACM Multimedia Systems Conference (MMSys), 2013.
- [ATA 2010] American Telemedicine Association, “Blueprint for Telerehabilitation Guidelines”, 2010.
- [Bajcsy 2009] Peter Bajcsy, Kenton McHenry, Hye-Jung Na, Rahul Malik, Andrew Spencer, Suk-Kyu Lee, Rob Kooper, and Mike Frogley, “Immersive Environments for Rehabilitation Activities”, in Proceedings of ACM international conference on Multimedia (MM), 2009.
- [Bay 2008] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc van Gool, “SURF: Speeded Up Robust Features”, Computer Vision and Image Understanding, Volume 110, Issue 3, Pages 346–359, Elsevier, 2008.
- [Beier 1992] Thaddeus Beier and Shawn Neely, “Feature-Based Image Metamorphosis”, in Proceedings of Computer graphics and Interactive Techniques (SIGGRAPH), 1992.
- [Berg 2008] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, “Computational Geometry: Algorithms and Applications”, Springer, Third Edition, 2008.

- [CBS 2004] “CBS to Use Enhanced Tape-Delay for Grammys”, 2004, <http://us.cnn.com/2004/SHOWBIZ/TV/02/03/grammys.tape.delay/index.html>
- [Chang 2011] Chih-Chung Chang and Chih-Jen Lin, “LIBSVM: a Library for Support Vector Machines”, *ACM Transactions on Intelligent Systems and Technology (TIST)*, Volume 2 Issue 3, 2011.
- [Chen 2013a] Shannon Chen, Pengye Xia, and Klara Nahrstedt, “Activity-Aware Adaptive Compression: a Morphing-Based Frame Synthesis Application in 3DTI”, in *Proceedings of ACM international conference on Multimedia (MM)*, 2013.
- [Chen 2013b] Shannon Chen and Klara Nahrstedt, “Impact of Morphing-Based Frame Synthesis on Bandwidth Optimization for 3DTI Video”, in *Proceedings of IEEE International Symposium on Multimedia (ISM)*, 2013.
- [Chen 2013c] Shannon Chen and Klara Nahrstedt, “Activity-Based Synthesized Frame Generation in 3DTI Video”, in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2013.
- [Chen 2014] Shannon Chen, Klara Nahrstedt, and Indranil Gupta, “3DTI Amphitheater: a Manageable 3DTI Environment with Hierarchical Stream Prioritization”, in *Proceedings of ACM Multimedia Systems (MMSys)*, 2014.
- [Chen 2015] Shannon Chen, Zhenhuan Gao, Klara Nahrstedt, and Indranil Gupta, “3DTI Amphitheater: Towards 3DTI Broadcasting”, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Volume 11, Issue 2s, 2015.
- [Chen 2016a] Shannon Chen, Zhenhuan Gao, and Klara Nahrstedt, “F.Live: Towards Interactive Live Broadcast FTV Experience”, in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

- [Chen 2016b] Shannon Chen, Zhenhuan Gao, Aadhar Jain, Klara Nahrstedt, Ahsan Arefin, and Raoul Rivas, “Metadata-Based Activity Analysis in 3D Tele-Immersion”, in Proceedings of IEEE International Conference on Multimedia Big Data (BigMM), 2016.
- [Coulouris 2011] George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair, “Distributed Systems: Concepts and Design,” Addison-Wesley, Fifth Edition, 2011.
- [Dowling 2014] Ariel V. Dowling, Ouriel Barzilay, Yuval Lombrozo, and Alon Wolf, “An Adaptive Home-Use Robotic Rehabilitation System for the Upper Body”, IEEE Journal of Translational Engineering in Health and Medicine, Volume 2, Pages 1-10, 2014.
- [Ermi 2005] Ermi Laura and Mäyrä Frans, “Fundamental Components of the Gameplay Experience: Analyzing Immersion”, in Proceedings of DiGRA International Conference: Changing Views: Worlds in Play, 2005.
- [Eugster 2003] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec, “The Many Faces of Publish/Subscribe”, ACM Computing Surveys (CSUR), Volume 35 Issue 2, 2003.
- [EyeVision 2001] EyeVision, 2001,
<http://www.ri.cmu.edu/events/sb35/tksuperbowl.html>
- [Facebook 2016] Facebook Engineering, “Encoding for 360 Video and VR”,
www.facebook.com/Engineering/videos/10153781047207200/
- [FantaMorph] FantaMorph, <http://www.fantamorph.com/index.html>
- [Feldman 2007] Dima Feldman and Yuval Shavitt, “An Optimal Median Calculation Algorithm for Estimating Internet Link Delays from Active Measurements”, Fifth IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON), 2007.
- [Forte 2010] Maurizio Forte and Gregorij Kurillo, “Cyberarchaeology: Experimenting with Tele-Immersive Archaeology”, in Proceedings of IEEE International Conference on Virtual Systems and Multimedia (VSMM), 2010.

- [Fuchs 2014] Henry Fuchs, Andrei State, and Jean-Charles Bazin, “Immersive 3D Telepresence”, IEEE Computer, Issue 7, Pages 46-52, 2014.
- [Gonzalez 2014] Alejandro González, Philippe Fraisse, and Mitsuhiro Hayashibe, “Adaptive Interface for Personalized Center of Mass Self-Identification in Home Rehabilitation”, IEEE Sensors Journal, Pages 2814 – 2823, 2014.
- [Gopalakrishnan 2011] Vijay Gopalakrishnan, Rittwik Jana, K. K. Ramakrishnan, Deborah F. Swayne, and Vinay A. Vaishampayan, “Understanding Couch Potatoes: Measurement and Modeling of Interactive Usage of IPTV at Large Scale,” in Proceedings of ACM Internet Measurement Conference (IMC), 2011.
- [Hamza 2014] Ahmed Hamza and Mohamed Hefeeda, “A DASH-based Free Viewpoint Video Streaming System”, in Proceedings of Network and Operating System Support on Digital Audio and Video Workshop (NOSSDAV), 2014.
- [Han 2015] Jay J. Han, Gregorij Kurillo, Richard Abresch, Evan Debie, Alina Nicorici, and Ruzena Bajcsy, “Upper Extremity 3D Reachable Workspace Analysis in Dystrophinopathy Using Kinect”, Muscle Nerve, Volume 52, Issue 3, Pages 344-55, 2015.
- [Horiuchi 2012] Toshiharu Horiuchi, Hiroshi Sankoh, Tsuneo Kato, and Sei Naito, “Interactive Music Video Application for Smartphones Based on Free-Viewpoint Video and Audio Rendering”, ACM International Conference on Multimedia (MM), 2012.
- [Huang 2011] Zixia Huang, Wanmin Wu, Klara Nahrstedt, Raoul Rivas, and Ahsan Arefin, “SyncCast: Synchronized Dissemination in Multi-Site Interactive 3D Tele-Immersion”, in Proceedings of ACM Conference on Multimedia Systems (MMSys), 2011.
- [Huang 2012] Zixia Huang and Klara Nahrstedt, “Perception-Based Playout Scheduling for High-Quality Real-Time Interactive Multimedia”, in Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2012.

- [ISO 1993] ISO/IEC 11172-2, “Information Technology: Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s”, 1993.
- [ISO 2014] ISO/IEC 23009-1, “Information Technology: Dynamic Adaptive Streaming over HTTP (DASH)”, 2014.
- [ITU 2008a] ITU-T Recommendation P.10/G.100, “Vocabulary for Performance and Quality of Service”, 2008.
- [ITU 2008b] ITU-T Recommendation P.910, “Subjective Video Quality Assessment Methods for Multimedia Applications”, 2008.
- [Jain 2003] Manish Jain and Constantinos Dovrolis, “End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput,” IEEE/ACM Transactions on Networking (TON), Volume 11, Issue 4, 2003.
- [Jain 2013] Aadhar Jain, Ahsan Arefin, Raoul Rivas, Chien-nan Chen, and Klara Nahrstedt, “3D Teleimmersive Activity Classification Based on Application-System Metadata”, in Proceedings of ACM Multimedia (MM), 2013.
- [Kittur 2008] Aniket Kittur, Ed H. Chi, and Bongwon Suh, “Crowdsourcing User Studies with Mechanical Turk”, in Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI), 2008.
- [Klaue 2003] Jirka Klaue, Berthold Rathke, and Adam Wolisz, “EvalVid - A Framework for Video Transmission and Quality Evaluation”, Computer Performance Evaluation: Modelling Techniques and Tools, Volume 2794, Pages 255-272, Springer, 2003.
- [Kurillo 2013] Gregorij Kurillo and Rezuna Bajcsy, “3D Tele-Immersion for Collaboration and Interaction of Geographically Distributed Users”, Virtual Reality, Volume 17, Issue 1, Pages 29-43, Springer, 2013.
- [Kurillo 2014] Gregorij Kurillo, Jay J. Han, Alina Nicorici, and Rezuna Bajcsy, “Tele-MFAsT: Kinect-Based Tele-Medicine Tool for Remote Motion and Function Assessment”, Studies in Health Technology and Informatics, Volume 196, Issue 2, Pages 15-21, 2014.

- [LIBICP] LIBICP: C++ Library for Iterative Closest Point Matching, <http://www.cvlibs.net/software/libicp/>
- [Mamou 2009] Khaled Mamou, Titus Zaharia, and Françoise Prêteux, “TFAN: A Low Complexity 3D Mesh Compression Algorithm”, *Computer Animation and Virtual Worlds (CASA): Special Issue, Volume 20, Issue 2-3*, 2009.
- [Mekuria 2013] Rufael Mekuria, Michele Sanna, Stefano Asioli, Ebroul Izquierdo, Dick C. A. Bulterman, and Pablo Cesar, “A 3D Tele-Immersion System Based on Live Captured Mesh Geometry”, in *Proceedings of ACM Multimedia Systems (MMSys)*, 2013.
- [Mekuria 2014] Rufael Mekuria, Michele Sanna, Ebroul Izquierdo, Dick C. A. Bulterman, and Pablo Cesar, “Enabling 3D Tele-Immersion with Live Reconstructed Mesh Geometry with Fast Mesh Compression and Linear Rateless Coding”, *IEEE Transactions on Multimedia (TMM)*, Volume 16, Issue 7, Pages 1809 – 1820, 2014.
- [Muja 2009] Marius Muja and David G. Lowe, “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration”, *International Conference on Computer Vision Theory and Applications (VISAPP)*, Springer, 2009.
- [Nahrstedt 2011] Klara Nahrstedt, Zhenyu Yang, Wanmin Wu, Ahsan Arefin, and Raoul Rivas, “Next Generation Session Management for 3D Tele-Immersive Interactive Environments,” *Multimedia Tools and Applications*, Volume 51, Issue 2, Pages 593-623, Springer, 2011.
- [Nahrstedt 2012] Klara Nahrstedt, “3D Tele-Immersion for Remote Injury Assessment”, in *Proceedings of International Workshop on Socially-Aware Multimedia (SAM)*, 2012.
- [Neri 2010] Peter Neri, “How Inherently Noisy is Human Sensory Processing?”, *Psychonomic Bulletin & Review*, Volume 17, Issue 6, Pages 802-808, 2010.
- [Netmap] Netmap, <http://www.caida.org/tools/visualization/mapnet>
- [Nexus 2012] Nexus, 2012, <http://www.google.com/nexus>

- [Niu 2004] Wei Niu, Jiao Long, Dan Han, and Yuan-Fang Wang, “Human Activity Detection and Recognition for Video Surveillance”, in Proceedings of IEEE International Conference on Multimedia and Expo (ICME), 2004.
- [PPLive] PPLive. <http://www.pptv.com/>
- [Priorov 2014] Andrew Priorov and Alexandr Prozorov, “Methods of Complete Surface Reconstruction through Merging of Point Clouds According to Stereo Vision Data”, in Proceedings of IEEE Conference of Open Innovations Association (FRUCT16), 2014.
- [RFC 1998] RFC 2435, “RTP Payload Format for JPEG-compressed Video”, 1998.
- [Robinson 2004] Stewart Robinson, “Simulation: The Practice of Model Development and Use”, Wiley, First Edition, 2004.
- [Sadagic 2013] Amela Sadagic, Mathias Kölsch, Greg Welch, Chumki Basu, Chris Darken, Juan P. Wachs, Henry Fuchs, Herman Towles, Neil Rowe, Jan-Michael Frahm, Li Guan, Rakesh Kumar, and Hui Cheng, “Smart Instrumented Training Ranges: Bringing Automated System Solutions to Support Critical Domain Needs,” The Journal of Defense Modeling and Simulation, Application, Methodology, Technology, Volume 10, Issue 3, Pages 327-342, 2013.
- [Schulte 2014] Sabrina Schulte, Shannon Chen, and Klara Nahrstedt, “Stevens’ Power Law in 3D Tele-Immersion: Towards Subjective Modeling of Multimodal Cyber Interaction”, in Proceedings of ACM Multimedia (MM), 2014.
- [Sheppard 2008] Renata M. Sheppard, Mahsa Kamali, Raoul Rivas, Morihiko Tamai, Zhenyu Yang, Wanmin Wu, and Klara Nahrstedt, Advancing interactive collaborative mediums through tele-immersive dance (TED): a symbiotic creativity and design environment for art and computer science, in Proceedings of ACM Multimedia (MM), 2008.

- [Shi 2012] Shu Shi, Klara Nahrstedt, and Roy H. Campbell, “A Real-Time Remote Rendering System for Interactive Mobile Graphics”, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Volume 8, Issue 3s, 2012.
- [Sonnenwald 2014] Diane H. Sonnenwald, Hanna Maurin Söderholm, Gregory F. Welch, Bruce A. Cairns, James E. Manning, and Henry Fuchs, “Illuminating Collaboration in Emergency Healthcare Situations: Paramedic-Physician Collaboration and 3D Telepresence Technology”, *Information Research*, Volume 19, Issue, 2, Paper 618, 2014.
- [Sowa 1992] John F. Sowa, “Semantic Networks”, in *Encyclopedia of Artificial Intelligence*, Wiley, second edition, 1992.
- [Su 2014] Tianyu Su, Abbas Javadtalab, Abdulsalam Yassine, Shervin Shirmohammadi, “A DASH-Based 3D Multi-View Video Rate Control System”, *IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2014.
- [Sung 2014] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena, “Human Activity Detection from RGBD Images”, *AAAI workshop on Pattern, Activity and Intent Recognition (PAIR)*, 2011.
- [Tanimoto 2011] Masayuki Tanimoto, Mehrdad Panahpour Tehrani, Toshiaki Fujii, and Tomohiro Yendo, “Free-viewpoint TV: A Review of the Ultimate 3DTV and its Technologies”, *IEEE Signal Processing Magazine*, Pages 67 – 76, 2011.
- [TEEVE] TEEVE, <http://cairo.cs.uiuc.edu/projects/teleimmersion/>
- [TenEase] TenEase, <http://tenease.com>
- [TESSEL] TESSEL, <https://tessel.io/>
- [Ustream] Ustream, <http://www.ustream.tv/>
- [Vasudevan 2011] Ramanarayan Vasudevan, Gregorij Kurillo, Edgar Lobaton, Tony Bernardin, Oliver Kreylos, Ruzena Bajcsy, and Klara Nahrstedt, “High Quality Visualization for Geographically Distributed 3D Tele-Immersive Applications”, *IEEE Transactions on Multimedia (TMM)*, Pages 573 – 584, 2011.

- [Wang 1996] Zheng Wang and Jon Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE Journal on Selected Areas in Communications (JSAC)*, Volume 14, Issue 7, Pages: 1228 – 1234, 1996.
- [Wang 2004] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Transactions on Image Processing*, Volume 13, Issue 4, Pages 600 – 612, 2004.
- [Winer 1991] Benjamin J Winer, Donald R Brown, and Kenneth M. Michels, "Statistical Principles in Experimental Design", McGraw-Hill, Thrid Edition, 1991.
- [Wu 2008] Wanmin Wu, Zhenyu Yang, Indranil Gupta, and Klara Nahrstedt, "Towards Multi-Site Collaboration in 3D Tele-Immersive Environments", in *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2008.
- [Wu 2009] Wanmin Wu, Ahsan Arefin, Raoul Rivas, Klara Nahrstedt, Renata Sheppard, and Zhenyu Yang, "Quality of Experience in Distributed Interactive Multimedia Environments: Toward a Theoretical Framework", in *Proceedings of ACM Multimedia (MM)*, 2009.
- [Wu 2010] Wanmin Wu, Ahsan Arefin, Zixia Huang, Pooja Agarwal, Shu Shi, Raoul Rivas, and Klara Nahrstedt, "I'm the Jedi! - A Case Study of User Experience in 3D Tele-Immersive Gaming", *IEEE International Symposium on Multimedia (ISM)*, 2010.
- [Wu 2011] Wanmin Wu, Ahsan Arefin, Gregorij Kurillo, Pooja Agarwal, Klara Nahrstedt, and Ruzena Bajcsy, "Color-Plus-Depth Level-of-Detail in 3D Tele-Immersive Video: A Psychophysical Approach", in *Proceedings of ACM Multimedia (MM)*, 2011.
- [Xia 2013] Pengye Xia and Klara Nahrstedt, "TEEVE Endpoint: Towards the Ease of 3D Tele-Immersive Application Development", in *Proceedings of ACM Multimedia (MM)*, 2013.
- [Yang 2006a] Zhenyu Yang, "A Multi-stream Adaptation Framework for Tele-Immersion", in *Proceedings of ACM Multimedia (MM)*, 2006.

- [Yang 2006b] Zhenyu Yang, Bin Yu, Klara Nahrstedt, and Ruzena Bajcsy, “A Multi-Stream Adaptation Framework for Bandwidth Management in 3D Tele-immersion”, in Proceedings of Network and Operating System Support on Digital Audio and Video Workshop (NOSSDAV), 2014., 2006
- [Yang 2010] Zhenyu Yang, Wanmin Wu, Klara Nahrstedt, Gregorij Kurillo, and Ruzena Bajcsy, “Enabling Multi-Party 3D Tele-Immersive Environments with ViewCast,” ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), Volume 6 Issue 2, 2010.
- [Youtube] Youtube Live, <http://www.youtube.com/>
- [Zhang 2005] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Tak-Shing Yum, “CoolStreaming/DONet: A Data Driven Overlay Network for Peer-to-Peer Live Media Streaming”, in Proceedings of IEEE International Conference on Computer Communications (INFOCOM), 2005.
- [ZLIB] Zlib, <http://www.zlib.net/>

Appendix

Here we list the pseudocodes of the dissemination forest adaptation algorithms and the pseudocode that handles user view change event.

	Algorithm 1: $UJoin(u, S')$
	Input: new user u , set of stream S' to subscribe
1:	<i>// Phase 1</i>
2:	$C \leftarrow \emptyset$
3:	For each $s \in S'$ do
4:	Find (u_s, s) such that
5:	(0) $E(u_s) + \varepsilon \geq E(u)$
	<i>// ε = additional elapse due to propagation</i>
6:	(1) u_s subscribes to stream s
7:	(2) u_s has available bandwidth
8:	(3) u_s has the smallest $E(u_s)$ under (0)(1)(2)
9:	If no such (u_s, s) exists do
10:	Let user u download stream s from the producer
11:	Remove s from S'
12:	Else do
13:	Add (u_s, s) to C
14:	End if
15:	End for
16:	If $C = \emptyset$ do
17:	Return <i>// u downloads all streams from producers</i>
18:	End if
19:	$(\check{u}_s, \check{s}) \leftarrow$ user \check{u}_s has the largest $E(\check{u}_s)$ in C
20:	Let user u download stream \check{s} from \check{u}_s
21:	Remove \check{s} from S'
22:	$I \leftarrow [E(\check{u}_s), E(\check{u}_s)]$ <i>//tight interval of sources' elapses</i>
23:	<i>// Phase 2</i>
24:	$C \leftarrow \emptyset$
25:	While $S' \neq \emptyset$ do
26:	For each $s \in S'$ do
27:	Find (u_s, s) such that
28:	(0) $E(u_s) + \varepsilon \geq E(u)$

29:	(1) u_s subscribes to stream s
30:	(2) u_s has available bandwidth
31:	(3) $E(u_s)$ is closest to interval I under (0)(1)(2)
32:	If no such (u_s, s) exists do
33:	Let user u download stream s from the producer
34:	Remove s from S'
35:	Else do
36:	Add (u_s, s) to C
37:	End if
38:	End for
39:	$(\tilde{u}_s, \tilde{s}) \leftarrow$ user \tilde{u}_s has the smallest $E(\tilde{u}_s)$ in C
40:	Let user u download stream \tilde{s} from \tilde{u}_s
41:	Remove \tilde{s} from S'
42:	Include $E(\tilde{u}_s)$ to I
43:	End While

Algorithm 2: $Reassign(o, S'_o)$ **Input:** orphan user o , set of streams S'_o without source

1:	For each $s \in S'_o$ do
2:	Find (u_s, s) such that
3:	(0) $E(u_s) + \varepsilon = E(o)$
4:	(1) u_s subscribes to stream s
5:	(2) u_s has available bandwidth
6:	If no such (u_s, s) exists do
7:	Let user o download stream s from the producer
8:	Else do
9:	Let user o download stream s from u_s
10:	End if
11:	End for

Algorithm 3: $VChange(u, S')$ **Input:** view changing user u , new set of stream S' to subscribe

1:	$O \leftarrow$ Children of user u
2:	For each $o \in O$ do
3:	$Reassign(o, S'_o)$
4:	End for
5:	User u terminates all the downloading
6:	$UJoin(u, S')$
