Archiving and Delivery of 3DTI Rehabilitation Sessions

Shannon Chen, Zhenhuan Gao, and Klara Nahstedt

Department of Computer Science, University of Illinois at Urbana-Champaign

{cchen116, zgao11, klara}@illinois.edu

ABSTRACT

In this paper we present CyPhy: a cyber-physiotherapy system that brings daily rehabilitation to patient's home with supervision from trained therapist. With its archiving and delivery features, CyPhy is able to 1) capture and record RGB-D and physiotherapy-related medical sensing data streams in home environment; 2) provide efficient storage for rehabilitation session recordings; 3) provide fast metadata analysis over stored sessions for review recommendation; 4) adaptively deliver rehabilitation session under different networking capabilities; 5) support smooth viewpoint changing during 3D video streaming with scene rendering schemes tailored for devices with different bandwidth and power limitations; and 6) provide platform-independent streaming client for various mobile and PC environments.

1. INTRODUCTION

As the population continues to age, the need for physical therapy (physiotherapy) services increases accordingly due to high correlation between age and physical conditions. In a recent report by the US Department of Labor, the demand of physiotherapy services is expected to further increase by 36% from 2012 to 2022 [1]. However, analysis of data compiled from online job search engine [2] also shows that the providers (clinics, hospitals, certified trainers) of physiotherapy still largely concentrates in only urban areas. Thus, although new physiotherapy workforce is well expected because of the growing need, the demand in rural areas is still unlikely to be fulfilled. These observations indicate that, *in the foreseeable future, the massive growth of elderly patients who suffered from physical conditions will become a great challenge for current physiotherapy providers due to their highly uneven geographical distribution.*

Physiotherapy, among other treatments, is a practice whose effectiveness relates more closely to the geographical proximity of therapy provider and patient. Due to the fact that a major part of the treatment relies heavily on long-term supervised rehabilitation, patients in in urban areas are more motivated to participate more regularly in therapy sessions. In rural areas, it is difficult for patients to conduct rehabilitation sessions on a regular basis under clinical supervision. Unsupervised rehabilitation, however, deprives the feedback and progress monitoring by trained professionals and hence has constrained effectiveness. Oftentimes patient even falsely interprets the pictorial paper-based instructions prescribed by the therapist and wasted weeks on inaccurate exercises in unsupervised rehabilitation.

Overcoming the geographical distance and bringing healthcare to patient's home, the concept of Telehealth [3] is proposed. Through online video inquiry, telehealth systems allow professionals to diagnose their patients despite of geographical separation. However, traditional telehealth medium is insufficient for physiotherapy. Physiotherapy involves wide-range body movement of patient, and accordingly, dynamic perspective (i.e., viewpoint) of the therapist. Thus, conventional 2D-video-chat-like telehealth medium is inadequate for delivering physiotherapy practice. *New multimedia cyber-physical environments, with more comprehensive representations and more dynamic points of interests, must be provided to physiotherapy providers*.

3D Tele-immersion (3DTI) technology allows full-body, multimodal content delivery among geographically dispersed users. In 3DTI, user's 3D model will be captured by multiple RGB-D (color plus depth) cameras surrounding user's body. In addition, various sensors (e.g., motion sensors, medical sensors, wearable gaming consoles, etc.) specified by the application will be included to deliver a multimodal experience. This opens a wide variety of possibilities for cyber-collaboration [4][6]. Among them, cyber-physiotherapy is an advanced healthcare application that can benefit greatly from the 3DTI framework due to its 360° coverage of user's body. 3DTI can enable preliminary remote injury assessment while the patient is at home and the therapist is at the clinic; or a patient can do tele-rehabilitation with a pre-recorded 3D training video while a therapist follows up offline on the patient's progress. Hence, we present CyPhy: a 3DTI **Cy**ber-**Phy**siotherapy system which exploits the richness of healthcare semantics encompasses correlated multimodal information, and health-related users' attention to deliver 3D multimodal content between therapist and patient.

We notice that the interactions between physiotherapy providers and patients can actually be separated into two phases: injury assessment and rehabilitation. While the two have equal importance towards patient's recovery, they consume very different amount of time throughout the therapy. For example, in treatment for adhesive capsulitis (i.e., frozen shoulder), injury assessment contributes to biweekly meetings of 20 minutes in average between therapist and patient; while rehabilitation contributes to therapeutic exercise of 30 minutes every day. Therefore, in view of the highly unbalanced time proportions (10 min/week vs. 3.5 hours/week), *the scope of this paper focuses on realizing at-home rehabilitation with asynchronous progress assessment and supervision by trained professionals.*

To enable remote asynchronous rehabilitation, this paper covers two major features and their challenges of CyPhy: *archiving* and *delivery* of rehabilitation sessions. Archiving a rehabilitation session serves the purpose of offline supervising, patient progress tracking, and health record keeping. Due to the large number of patients, therapist is not expected to be on-call 24/7 to monitor patient's exercise online. Instead, therapist will follow up and check the patient's recorded session periodically. Thus, the rehabilitation is done at home, being recorded and archived. Delivery of rehabilitation sessions, on the other hand, serves two purposes. First, CyPhy delivers demonstration 3D video of therapeutic exercise recorded and prescribed by the therapist to patient's home so that patient can follow correct exercise in her daily rehabilitation. Second, therapist needs to be able to playback the rehabilitation sessions which a patient recorded at home to evaluate her progress. While the first purpose assumes viewer (patient) will be receiving the session at home with stable internet connection and power, viewer of the second purpose (therapist) may be on the move, receiving patient's recordings with a mobile device under limited bandwidth and power supply resources.

Therefore, in order to serve archive and delivery challenges, our approaches (and hence our contributions in this paper) in the CyPhy system are as follows

A. Rehabilitation session archiving feature:

1. Effective capturing and recording of RGB-D and physiotherapy-related medical sensing data streams in home environment.
2. Efficient storage of rehabilitation session recordings in an electronic health record (EHR) cloud in order to sustain long term records of large patient group.
3. Fast metadata analysis over stored rehabilitation sessions to provide recommendations upon session reviewing.

B. Rehabilitation session delivery feature:

4. Adaptive, DASH[7]-like 3D session delivery mechanism for rehabilitation session streaming to viewers (therapist, patient) with different network capabilities.
5. Smooth viewpoint change during 3D video streaming with scene rendering schemes tailored to devices with different bandwidth and power limitations.
6. Platform-independent 3D streaming client for viewer's mobile and PC operating environments.
7. Validation of the integrated CyPhy System.

We evaluate the performance of our CyPhy prototype with rehabilitation sessions recorded in our lab. We use Kinect cameras to capture RGB-D and audio content and a microcontroller with electrodes as medical sensor to capture electromyography (EMG) pulse of muscle. We generate a series of rehabilitation session recordings as our dataset to test CyPhy's archiving feature. Results show that our storage compression scheme achieves 1:1255 compression ratio, which is a x1.73 gain from MPEG-TS [9] currently used for offline streaming. In addition, the codec provides metadata for fast preliminary analysis which will substantially speed up the process of patient's abnormal behavior detection. For session delivery, CyPhy is able to offload the computation-intensive rendering function and saves 75% bandwidth and 25% power for mobile clients. For adaptive streaming, CyPhy is able to adjust its streaming quality according to available bandwidth as well as user's viewpoint. Our prototype achieves zero playback interruption while maintaining high bandwidth utilization.

The rest of this paper is structured as follows. In Section 2 we review some related works on telehealth, 3DTI, and offline delivery. In Section 3 we define the use case model for CyPhy. In Section 4 the system architecture of CyPhy is presented. In Section 5 and 6 we introduce the archiving and delivery features of CyPhy. In Section 7 we present the settings for evaluation experiments which give the results in Section 8. Finally in Section 9 we conclude this paper.

2. RELATED WORKS

Telehealth interfaces for physiotherapy. In recent years, physiotherapy interfaces have been proposed in robotics and sensor research areas to provide inputs for telehealth. In [10], the authors develop a robotic rehabilitation system to treat musculoskeletal conditions. They capture EMG and skeleton signals of patient's arm movement and feed the captured data to a remote robot arm. The robot will reproduce the movement of patient so a therapist can give prescriptions remotely. In [11], an interface combing Kinect camera and Wii balance board is proposed. It captures the center of mass position to determine patient's physical stability. Based on determined stability, the interface provides visual feedback to patients during their self-supervised rehabilitation. In [5][27], the authors use Kinect camera to analysis the 3D reachable workspace of patients with upper body conditions. The interface provides real-time visual feedback indicating range of motion and generates workspace analysis as 3D images which are sent to a therapist for diagnosis.

Cyber collaborations through 3DTI. 3DTI systems aim towards multi-purpose, multi-sites, and multimodality to enable a wide variety of user activities [8][13][21]. In [16] and [29] 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 [4] and [18]. In [15], 3DTI platform for performance broadcasting is proposed. The authors envision performer crew to be physically dispersed and interact remotely in the virtual world.

3DTI content archiving. Due to the high bitrate of 3DTI, various archiving schemes for compression and content analysis were proposed. In [14] and [17], compression module based on frame synthesis is proposed to lower the bitrate of 3DTI systems. In [18], the module is paired with activity recognition to achieve dynamic bitrate adaptation. Other compression schemes for mesh-based 3DTI content are proposed in [19][20], which concentrate on independent 3D image compression without inter-frame coding. Analysis on 3DTI data using metadata is proposed in [22]. The authors achieve high activity detection accuracy via metadata analysis to avoid computationally expensive deep content analysis.

3DTI content delivery: Delivery of 3DTI content is not trivial due to its bandwidth consumption and realtime requirement. In [15], 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 [23], 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 to their delivery accordingly to achieve a better quality of experience.

3. 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. In the end of the assessment, prescription will be made, which contains instructions for patient to follow in her rehabilitation. In the rehabilitation phase, patient conducts daily therapeutic exercises prescribed by the therapist under supervision of trained medical staff if feasible. 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 supersized rehabilitation with trained professionals, they would have to travel back and forth between home and clinic on a daily basis. CyPhy is proposed to relieve patients from at-clinic rehabilitation and replace it by at-home, supervised rehabilitation session. Note that *CyPhy system in this paper 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 1.) 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 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 also provide as references in evaluation of patient's recovery progress.

4. SYSTEM ARCHITECTURE

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

4.1 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 1) includes four Kinect cameras[1] (with optional tripods) and a compression suit embedded with a microcontroller and EMG sensors (Figure 2). With the CyPhy kit, patient is expected to be able to set up the light-weighted rehabilitation home studio without any technical background.

Setting of the studio is borrowed from MobileTI [24]. Inside the kit patient will find a mat with standing position and shooting direction of cameras marked on it. The four cameras are set to be surrounding the patient's exercise area (a 2m diameter circle), 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 [25] and iterative closest point alignment [26]. 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. The issue is not addressed in our current prototype since we are only interested in gross motor movements. Minor noise incurred by interference does not degrade our graphical quality by large. As a future work, amendment module proposed by Maimone and Fuchs [33] can be included in CyPhy to solve the problem.

Figure 1. CyPhy kit: Kinects, compression suit, mat, and extra electrodes.

Figure 2. Compression suit with EMG sensors and TESSEL.

The compression suit in CyPhy kit is provided so that patient will wear it during the rehabilitation session. The suit avoids occlusion by normal clothing and allows 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 inside of the suit so that they can be deployed easily by patients themselves. The EMG pulses picked up by the sensors are collected by a microcontroller, TESSEL [28], 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 PC of patient's need 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. We will come back to details on streaming in Section 6 and on recording in Section 5.1. Uploading of the recorded session is straight forward since the CyPhy kit only generates 30 minute content every day. Thus, uploading can be done in the background with low bandwidth consumption. This avoids CyPhy from interfering with other networking applications in the home environment.

4.2 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 then 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. Since delivery and rendering of free viewpoint video can be considerable burdens for mobile devices, CyPhy client is adaptive in two aspects. First, it conducts adaptive streaming based on DASH [7] mechanism for 3D content. According to the available bandwidth, it adaptively requests content of different bitrates. Second, it automatically offloads part of the rendering function to the server (EHR cloud) when the therapist is using a mobile device with power limitations. In Section 6 we introduce more details regarding our design on adaptive delivery.

4.3 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 3). Machines in the first tier are archive nodes which keep long-term rehabilitation recording of patients. They provide compression as well as metadata analysis of the stored sessions. Results of

---

[1] If patient's PC does not have enough computing power to support four Kinects, additional laptops can be included in the CyPhy kit. For synchronization across machines, see [32].

metadata analysis provide information on anomaly detection and recommendation on session reviewing. Details on session archiving will be revisited in Section 5.

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, task for CDN nodes concentrates on delivery of rehabilitation sessions. This includes handling requests from viewer as a DASH server, and point-cloud-to-scene rendering offloaded by mobile clients. Details on session delivery will be revisited in Section 6.

5. SESSION ARCHIVING

Archiving of patient's rehabilitation involves three features 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.

5.1 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. 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 using an open source EMG project [28] provided by TESSEL. The captured signals are sent in JSON format to patient's PC, where they are bundled with the other contents.

The different streams in our bundle need to be synchronized with each other to provide correct information to therapists. Since RGB-D, skeleton, and audio are captured by devices connected to the same PC[1], synchronization between them is done by standard buffering and alignment [30]. To synchronize EMG signals with others, we adopted the audio-based synchronization scheme used in [31]. The idea is to have devices to be synchronized equipped with audio sensors and make them synchronize their sampling to a coordinating periodic audio tone. The frequency of the tone is set out of the audible range so that it will not be interfered by background noises. In our scenario, the audio tone will be emitted by patient's PC, and the TESSEL on patient's body has an ambient sensing module to pick up audio signals. The EMG signals collected by TESSEL, are tagged with the synchronization signals. When they arrive at the PC, the EMG signals are bundled with the rest of the contents according to their tags.

5.2 Content Compression

The total bitrate of a raw data bundle is in 1100 Mbps magnitude. This implies that, without compression, each patient will upload 247 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. 99.94% of raw data in the content bundle is contributed by RGB-D videos hence we concentrate on compression of visual streams in this paper. Audio stream is compressed with simple MP3 codec and skeleton/EMG streams stay in their raw data format.

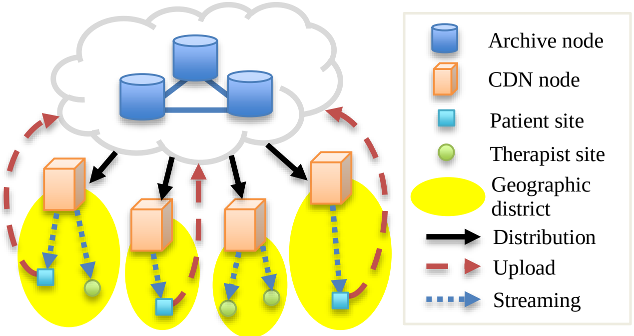


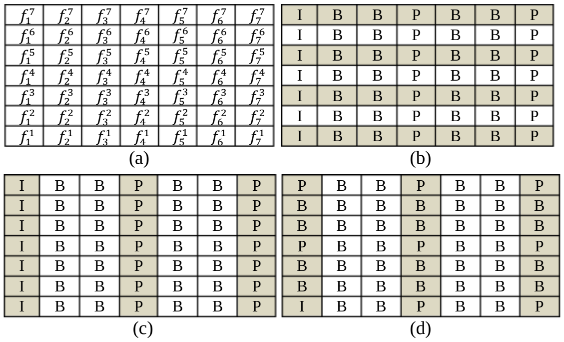
Figure 3. Two-tier architecture of EHR cloud.

3D compression gains its popularity since as early as 1998. Mekuria *et al.* [19] provide comparison on compression ratio and computation overheads of 3D codecs up to 2013. Since most of these works inherit from computer graphics studies rather than image processing, they choose to format 3D visual content with triangle meshes but point clouds. This design decision brings substantiate bitrate saving for still 3D image because it significantly lowers down the number of vixel units. Empirically, a point cloud containing 300,000 points can be represented by 70,000 triangle vertices without major quality loss [19]. However, compression ratio of this approach is still far from being comparable to video codecs for conventional 2D video. According to Mekuria's survey, the 1:10 compression ratio brought by TFAN [20] and Mekuria *et al.* [19] is the finest result among other mesh-based schemes. However, compression ratio of simple MPEG-1 [34] codec is well known to be in 1:100 magnitude. The reason behind this barrier is that these mesh-based schemes do not exploit inter-frame likeliness 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 [35] compression for 2D videos, which only have 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, 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 mock codec with MPEG-TS [9], which is a popular 2D video codec adopted by offline streaming standards (e.g., DASH). By encoding RGB and D streams separately with MPEG-TS, data size of a 30 minutes rehabilitation session is lowered down from 247 GB to 575 MB.

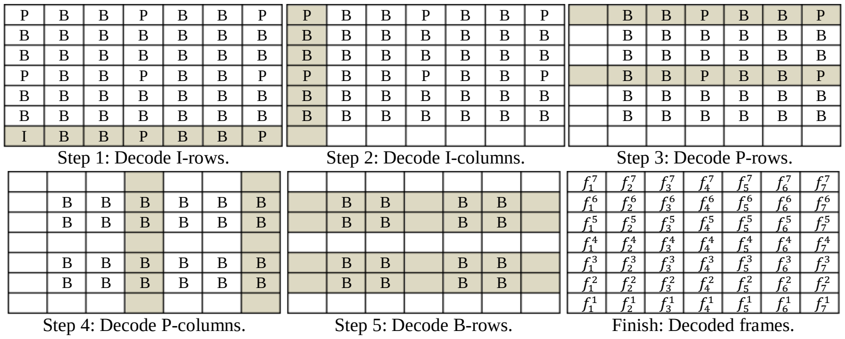
In view of the effectiveness of inter-frame coding, we propose to further exploit the similarity between *videos* rather than just frames. We observe that rehabilitation recordings of a patient have the following properties:

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 our 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.



(a) (b) (c) (d)

Figure 4. Inter-video encoding scheme of CyPhy.



Step 1: Decode I-rows. Step 2: Decode I-columns. Step 3: Decode P-rows. Step 4: Decode P-columns. Step 5: Decode B-rows. Finish: Decoded frames.

Figure 5. Inter-video decoding process of CyPhy.

4. In the recording, therapist only cares about the patient but not about the background. Thus, background in the video can be removed with the help of depth information.

Combining the observations, we know that visual contents recorded in adjacent rehabilitation sessions of a patient will be very similar to each other. Therefore, we propose a compression scheme specifically tailored for archiving rehabilitation recordings by compressing multiple videos recorded in time proximity (e.g., in the same week) 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 4. In Figure 4a, 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 MPEG encoding on each video, frames will be coded into I-, P-, and B-frames as illustrated in Figure 4b. While our pervious mock codec with MPEG-TS stops at this point, we propose to further encode each column of I- and P-frames together (Figure 4c). Since patient is expected to have consistent moves in every video, vertical adjacent frames in a column also share great similarity. Figure 4d shows the final coded frames after column-wise inter-frame coding. We can see 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, we expect a substantial, extra compression gain.

The proposed 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 is illustrated in Figure 5. 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, 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. We will come back to more details on file arrangement in CDN node in Section 6.1.

5.3 Metadata Analysis

Being able to analyze human activities in multimedia recordings is essential for automatic recommendation, summary generation, and anomaly event detection. 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 (e.g., 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 deep analysis on recorded content computationally expensive and time consuming. A broad range of previous research has been devoted to "deep analysis" on media data to identify human activity, which often requires computation-intensive computer-vision-based analysis [36][37] and/or pre-knowledge of possible activity categories [17][18].

In CyPhy, we propose to analyze metadata of the archived recordings, coded by our proposed compression scheme, to provide preliminary anomaly (i.e., patient's abnormal behavior) analysis. Metadata analysis in 3DTI content is first proposed by Jain *et al.* [22], 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 deep analysis. Rather, its result is provided as preliminary hint to help speed up the following deep analysis step. For example, assume we have a slow deep 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, deep 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 deep analysis module only need to go through 10% of the dataset, suggested by metadata analysis, to locate 50 anomalies (Figure 6).

In the case of CyPhy, our metadata is the size of predicted frames (P- and B-frames) in the column-wise inter-frame coding (Figure 4c). By the design of video frame types (I, P, and B), a predicted frame contains only the difference between itself and nearby frames. Thus, a larger difference contributes to a larger size. Since the column-wise inter-frame coding is exploiting the difference between adjacent rehabilitation sessions, we can find a positive correlation between size of predicted frames and the difference between RGB-D contents. Small predicted frame sizes imply high similarity between rehabilitation sessions, which means recovery progress between these days is modest and steady. Large predicted frame sizes imply low similarity, which can mean the frame captures patient injured or fell, or the recording system is being misused.

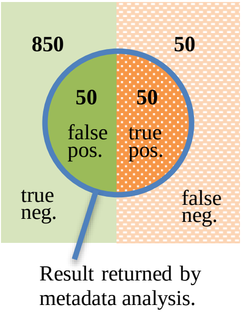


Figure 6. How metadata analysis aids deep analysis.

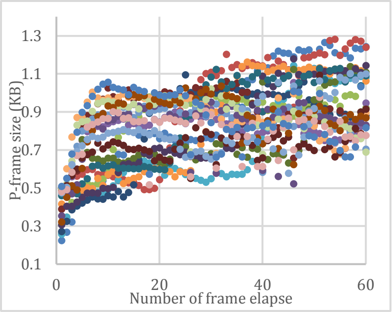


Figure 7. Correlation between P-frame size and content differences.

We conduct simple analysis on 2D video to verify the correlation between content differences and predicted frame size. From a raw 2D video we encode different versions of MPEG videos with 1) every frames in the raw sequence, 2) every two frames in the sequence, 3) every three frames…, etc. Intuitively, the larger the frame elapse we set in the encoding, the more the differences in adjacent frames in the encoded video. We plot the frame sizes of P-frames in Figure 7 with x-axis representing the number of frames elapsed in each video. We can see that, as expected, there is a clear trend between frames differences and P-frame sizes. The result of ANOVA test also shows significance with p<0.001.

6. SESSION DELIVERY

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

Challenges in rehabilitation session delivery in CyPhy are two-folds. 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 8, if viewer wishes to see from 30°, then streams captured by camera 1 (0°) and 2 (90°) needs 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 89° to 91° in Figure 8, the client software needs to subscribe to stream 3. Initiation delay for this subscription (time spent on buffering one RGB-D video segment of stream 3) will incur playback interruption. Yet, if we conduct naïve provisioning to subscribe to all four streams anytime in spite of viewer's current viewing angle, we can be wasting precious bandwidth in the home environment.

In summary, we propose the "EAR" requirements for session delivery of 3DTI:

- **E**fficient: Bandwidth used in delivery of a 3DTI session should be efficiently utilized.
- **A**daptive: Delivery system should offer best quality content under bandwidth and computing power constraints.

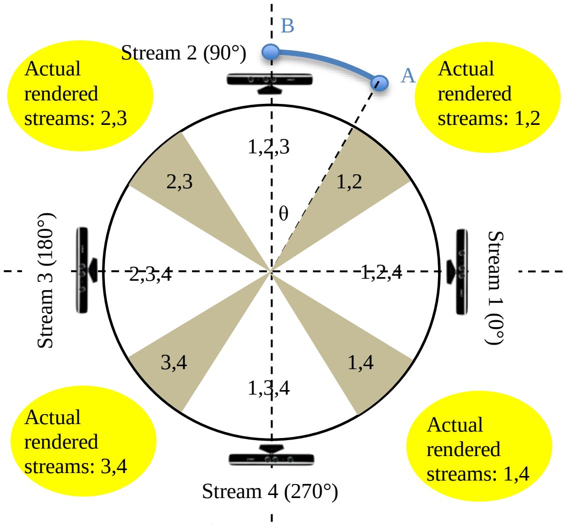


Figure 8. Safe zones and provision zones.

- **R**esponsive: 3DTI system should minimize the chance and length of playback interruption on view changing requests.

In the following, we introduce our implementation for the server (CDN nodes) and client (patient and therapist sites) of CyPhy that fulfill the EAR requirements.

6.1 Sever Design

We inherit the DASH standard in our 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 needs 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) 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 computation-expensive scene rendering to server. 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 in Figure 8.

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 server. The first option 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. 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 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, even for just 1°, 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 [23], 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 regular 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.

We implement a customized Node.js [38] HTTP server as our modified DASH server. The rendering functions including point cloud merging and scene extraction in the server are inherited from FreeViewer [39].

6.2 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 8, 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 to avoid initiation dely. The requested streams and the actual rendered streams in each zone are specified in Figure 8.

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 one segment of the newly requested stream or else interruption will happen. Using Figure 8 again as 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 (2θ) 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 90°/second 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 viewer changes from A to B. 1/3 seconds is the playback time of one RGB-D segment in our design thus naturally the buffering time of one segment is shorter than this.


(a) (b) (c)

Figure 9. Actor's normal movement (top row) versus movements with weights strapped on body (bottom row).

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 the viewer jumps within the same quadrant from a safe zone; or jumps within the same semicircle from a provision zone. Thus, the probability of interruption for non-offloading view jumps is only 58% (= 120°/360° × 270°/360° + 240°/360° × 180°/360°, assuming uniform distribution of starting and ending angles of jumps.)

We implement our client to be cross-platform with HTML5 and JavaScript. Client has normal DASH client functions in browsers using code modified from DASH-JS [40]. For non-offloading local rendering, we use WebGL and Three.js [41], and show the final rendered scene with HTML5 canvas element.

7. EXPERIMENT SETTINGS

7.1 Dataset

We have recorded a set of rehabilitation sessions in a home studio set as we specified in the patient site section. The set consists twelve recordings, imitating rehabilitation sessions a patient would conduct in twelve consecutive 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 9, the strapped weights incurs asymmetric standing posture and tilted movements, which are common symptoms seen in patients with physical conditions.

7.2 Testbed

Server Settings. We use a desktop PC equipped with 4-core CPU and 8GB RAM running Ubuntu 14.04 LTS to run our server. The server is driven by Node.js to server 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, 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 simulated 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. Both clients use Chrome to run our user interface written in HTML5 and JavaScript.

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

8. 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 delivery feature of CyPhy, which test the offloading and adaptive delivery, respectively.

8.1 Compression Effectiveness

We compress the rehabilitation session set recorded in our studio by our proposed scheme and MPEG-TS, respectively. On compression ratio, our scheme achieves a 1:1255 ratio on the dataset while MPEG-TS achieves 1:725. The difference implies that, with the same storage infrastructure, our scheme can sustain a patient group 1.73 times larger than MPEG-TS 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 PSNR and SSIM (structural similarity index) [42] tests against the raw frames. To get an estimation on the quality of experience (QoE), we use the empirical mapping from PSNR to MOS (mean opinion score) [43] for MPEG videos, reported in [44]. Differences between our proposed scheme and MPEG-TS are listed in Table 1. Although quality of our scheme is slightly worse (less than 10% difference) than MPEG-TS on PSNR and SSIM, the MOS difference (< 0.5) indicates that, statistically speaking (i.e., assuming normal distribution of individuals' sensitivity [45]), more than half of human viewers would not notice the quality difference between our scheme and MPEG-TS. 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 a 35mm average error over all compressed depth frames, which is acceptable for correctly presenting patient's exercise to therapists.

Table 1. Compression effectiveness of CyPhy

Compress Ratio	RGB Quality Relative to MPEG-TS			Depth RMSE
	PSNR	SSIM	MOS	
1:1255	-3.756	-0.0777	-0.4444	34.6 mm

8.2 Metadata Analysis

To evaluate the accuracy of the proposed metadata analysis to detect anomalies, we extract from each of the twelve recordings a 30 seconds long exercise that only involves shoulder rehabilitation (Figure 9a). Then, in this set of shoulder exercise recordings, we randomly inject 36 irrelevant exercises (Figure 9bc) 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.

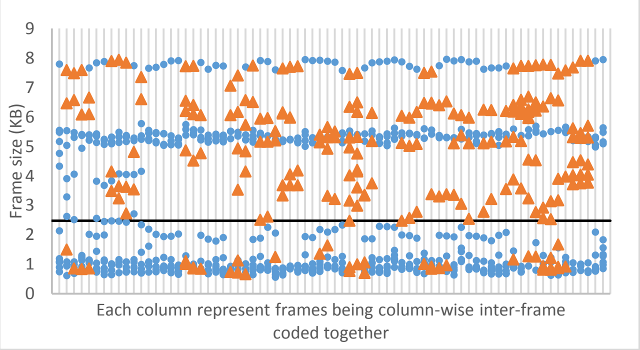


Figure 10. Frame size of anomaly (triangles) versus norm (dot).

In Figure 10 we plot the frame sizes of P-frames in the encoded set. Each column 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 a deep 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.

8.3 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 2.

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.

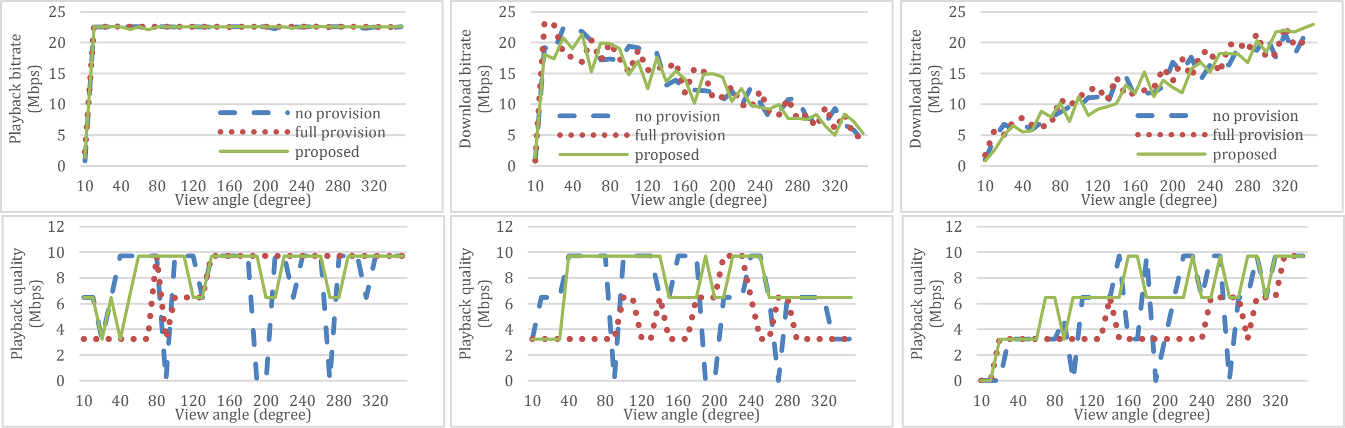
On power consumption, we see that the saving from offloading is not as much as streaming bandwidth. The reason is that our offloading feature only relieves the device from graphical rendering. Other power-consuming modules (e.g., screen backlight, session initiation) on the phone do not benefit from 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.

Table 2. Client resource consumption.

	Streaming bandwidth	**Power consumption**
Offload	1600 kbps	10.0% of full charge
No offload	6400 kbps	13.5% of full charge
Saving	**75%**	**26%**

8.4 Adaptive Delivery

Our DASH-based design in the server helps the delivery 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 delivery accounts both available bandwidth and user's



(a) Stable and abundant available bandwidth

(b) Available bandwidth depletion

(c) Available bandwidth recovery

Figure 11. Adaptive delivery of CyPhy.

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 delivery feature provides the first multi-stream 3DTI-on-demand (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 by Hamza and Hefeeda [23] and Su *et al.* [12]. However, their evaluations do not account dynamic view change during playback. Thus, due to lacking of existing subject for performance comparison, we provide two baseline 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 no 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 11 we plot the download bitrates and playback qualities 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 11 we plot the playback quality in each scenario. Here we use the total bitrate of the two RGB-D streams actually used in rendering (i.e., excluding the provisioned streams) to indicate playback quality. As we can see in Figure 11a, when the available bandwidth is ample, all three schemes reach the highest playback quality for most of the time. However, for no provisioning baseline, interruptions happen as the view angle changes across quadrants as expected, and cause the playback bitrate to drop to zero. The playback quality of our proposed 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 11b, we plot the playback qualities 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 playback qualities 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 proposed scheme achieves high playback quality as the no provisioning baseline, but without any interruption like the full provisioning baseline.

In Figure 11c, we see playback qualities 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 proposed scheme. But due to its heavy burden to download all four streams at any given time, it reaches high playback quality much later than ours.

In sum, although the utilization of available bandwidth are equally good for the three schemes, the gain of our proposed scheme reveals in its playback quality 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

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

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.

9. CONCLUSION

In view of the aging of our population and the uneven distribution of physiotherapy providers, in this paper we present CyPhy: a cyber-physiotherapy system that brings daily rehabilitation to patient's home with supervision from trained therapist. With its archiving feature, CyPhy provides efficient storage and analysis to keep long-term electronic health records for large patient group. With its DASH-based free-viewpoint delivery, CyPhy delivers recorded content to clients under different computing power and bandwidth constraints. We believe CyPhy is a new cyber medium with more comprehensive 3D/multimodal representation and more flexible interest point changing which helps physiotherapy providers prepare for upcoming challenges from the aging society.

10. REFERENCES

[1] Bureau of Labor Statistics, U.S. Department of Labor, Occupational Outlook Handbook, 2014-15 Edition

[2] Indeed. www.indeed.com

[3] American Telemedicine Association, Blueprint for Telerehabilitation Guidelines, 2010

[4] M. Forte and G. Kurillo, "Cyberarchaeology: Experimenting with teleimmersive archaeology", IEEE VSMM, 2010.

[5] J. J. Han, G. Kurillo, R. T. Abresch, E. de Bie, A. Nicorici, R. Bajcsy, Upper extremity 3D reachable workspace analysis in dystrophinopathy using Kinect, Muscle Nerve. 2015.

[6] S. Chen, Z. Gao, K. Nahrstedt et al., "3DTI Amphitheater: Towards 3DTI Broadcasting", ACM TOMM, 2015.

[7] ISO/IEC 23009-1, Information technology -- Dynamic adaptive streaming over HTTP (DASH), 2014.

[8] G. Kurillo and R. Bajcsy, "3D teleimmersion for collaboration and interaction of geographically distributed users", Virtual Reality, 2013.

[9] ISO/IEC 13818-1, Information technology -- Generic coding of moving pictures and associated audio information, 2013

[10] A.V. Dowling, O. Barzilay, Y. Lombrozo, A. Wolf, "An Adaptive Home-Use Robotic Rehabilitation System for the Upper Body", IEEE Journal of Translational Engineering in Health and Medicine, 2014.

[11] A. Gonzalez, P. Fraisse, and M. Hayashibe, "Adaptive Interface for Personalized Center of Mass Self-Identification in Home Rehabilitation", IEEE Sensors Journal, 2014.

[12] T. Su, A. Javadtalab, A. Yassine et al., "A DASH-based 3D multi-view video rate control system", IEEE ICSPCS, 2014.

[13] S. Schulte, S. Chen, and K. Nahrstedt," Stevens' Power Law in 3D Tele-immersion: Towards Subjective Modeling of Multimodal Cyber Interaction", ACM Multimedia, 2014.

[14] S. Chen and K. Nahrstedt, "Activity-based Synthesized Frame Generation in 3DTI Video", IEEE ICME, 2013.

[15] S. Chen, K. Nahrstedt, and I. Gupta, "3DTI Amphitheater: A Manageable 3DTI Environment with Hierarchical Stream Prioritization", ACM Multimedia Systems, 2014

[16] A. Sadagic, M. Kolsch, G. Welch, C. Basu et al., "Smart Instrumented Training Ranges: Bringing Automated System Solutions to Support Critical Domain Needs," The Journal for Defense Modeling and Simulation, 2013.

[17] S. Chen and K. Nahrstedt, Impact of Morphing-based Frame Synthesis on Bandwidth Optimization for 3DTI Video, in IEEE International Symposium on Multimedia (ISM), 2013.

[18] S. Chen, P. Xia, and K. Nahrstedt, "Activity-Aware Adaptive Compression: A Morphing-based Frame Synthesis Application in 3DTI", ACM Multimedia (MM), 2013.

[19] R. Mekuria, M. Sanna, S. Asioli, E. Izquierdo, et al., "A 3D Tele-Immersion System Based on Live Captured Mesh Geometry", ACM Multimedia Systems, 2013.

[20] K. Mamou, T. Zaharia, and F. Prêteux, TFAN: A low complexity 3D mesh compression algorithm, Comp. Anim. Virtual Worlds, 20: 343–354, 2009.

[21] H. Fuchs, A. State, and J.-C. Bazin, "Immersive 3D Telepresence", IEEE Computer, 2014.

[22] A. Jain, A. Arefin, R. Rivas, C. Chen, and K. Nahrstedt, "3D Teleimmersive Activity Classification Based on Application-System Metadata", ACM Multimedia (MM), 2013.

[23] A. Hamza and M. Hefeeda, "A DASH-based Free Viewpoint Video Streaming System", ACM NOSSDAV, 2014.

[24] W. Wu, R. Rivas, A. Arefin, et al., MobileTI: A Portable Tele-Immersive System, in Proc. of ACM Multimedia, 2009.

[25] A. Priorov and A. Prozorov, "Methods of complete surface reconstruction through merging of point clouds according to stereo vision data", IEEE FRUCT16, 2014.

[26] LIBICP: C++ Library for Iterative Closest Point Matching. http://www.cvlibs.net/software/libicp/

[27] G. Kurillo, J. J. Han, A. Nicorici, R. Bajcsy, "Tele-MFAsT: Kinect-Based Tele-Medicine Tool for Remote Motion and Function Assessment", Stud Health Technol Inform. 2014

[28] TESSEL. https://tessel.io/

[29] D. Sonnenwald, H. Söderholm, G. Welch, et al., "Illuminating collaboration in emergency health care situations: paramedic-physician collaboration and 3D telepresence technology", Information Research, 2014

[30] Ralf Steinmetz and Klara Nahrstedt, "Multimedia Systems", Springer, 2004 edition

[31] P. Kannan, S. Venkatagiri, M. Chan, et al., "Low Cost Crowd Counting using Audio Tones", ACM Sensys, 2012.

[32] Z. Yang, W. Wu, K. Nahrstedt, et al., Enabling Multi-party 3D Tele-immersive Environments with ViewCast, ACM TOMM, 2009.

[33] A. Maimone and H. Fuchs. "Reducing Interference between Multiple Structured Light Depth Sensors Using Motion." IEEE Virtual Reality 2012.

[34] 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.

[35] RFC 2435, "RTP Payload Format for JPEG-compressed Video", 1998.

[36] W. Niu, J. Long, D. Han, et al., Human activity detection and recognition for video surveillance. IEEE ICME, 2004.

[37] J. Sung, C. Ponce, B. Selman, et al. Human activity detection from RGDB images. AAAI workshop on PAIR, 2011.

[38] Node.js. https://nodejs.org/

[39] Z. Gao, S. Chen, and K. Nahrstedt,"FreeViewer: A 3D Tele-Immersive Intelligent Director", ACM Multimedia, 2014.

[40] DASH-JS. http://www-itec.uni-klu.ac.at/dash/?page_id=746

[41] Three.js. http://threejs.org/

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, 2004.

[43] ITU-T. P.800.1: Mean Opinion Score terminology, 2003.

[44] J. Klaue, B. Rathke, and A. Wolisz, “EvalVid - A Framework for Video Transmission and Quality Evaluation”, Springer, Computer Performance Evaluation, 2003.

[45] P. Neri, “How inherently noisy is human sensory processing?” Psychonomic Bulletin & Review 2010.