A Multi-User 360-Video Streaming System

for Wireless Network

Haowei Cao  
 Shanghai Jiao Tong University  
 Shanghai, China  
 chw0239@163.com

Jialiang Lu  
 Shanghai Jiao Tong University  
 Shanghai, China  
 jialiang.lu@sjtu.edu.cn

Nan Zong   
 Shanghai Jiao Tong University  
 Shanghai, China  
 cisnazong@sjtu.edu.cn

ABSTRACT

With the rapid development of Virtual Reality technology and its hardware, 360-degree video is becoming a new form of media which arouses the interest of public. In the past few years, many 360-degree video delivery schemes are proposed, but there hasn’t been a standard solution which can perfectly overcome the difficulties caused by network latency and bandwidth limit. In this paper, we consider the context of a wireless network consisting of a base station and several users. We proposed a 360-degree video delivery and streaming scheme which serves multiple users simultaneously while optimizing the global bandwidth consumption. The system will predict the head movement of users using machine learning algorithm, and extract the visible portion of the video frame for transmission. The core contribution of the scheme is that it will recognize the conjoint viewport of multiple users, and then optimize the global bandwidth consumption by arranging the transmission of conjoint viewport over the public channel of the wireless network. The results prove that the proposed scheme can effectively reduce global bandwidth consumption of the network with relatively simple configuration.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*VRCAI’19, November, 2019, Brisbane, Australia*

© 2019 Copyright held by the owner/author(s).

CCS CONCEPTS

• Human-centered computing → Virtual reality • Networks → Network measurement

KEYWORDS

Virtual Reality, 360-degree video, Machine learning, Head movement prediction, Network resource management

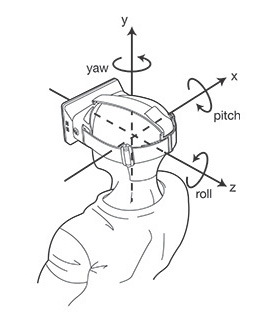
ACM Reference format:

Haowei Cao, Jialiang Lu and Nan Zong. 2019. A Multi-User 360-Video Streaming System for Wireless Network. In *Proceedings of ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI '19). ACM, Brisbane, Australia*

**1** INTRODUCTION

Virtual Reality technology has become increasingly popular. With the development of VR hardware and software, many new VR applications emerged. One important and popular VR product is 360-degree video, which is also known as immersive video, panoramic video, or spherical video.

360 videos are usually captured with omnidirectional cameras. They can be watched using head-mounted displays(HMDs) or VR headsets. When watching a 360-degree video, the viewer can freely control the viewing direction by turning the head. More specifically, the VR device will capture the head orientation of the viewer, which is generally represented by three Euler angles – pitch, yaw, and roll, which correspond to rotation around the X, Y, Z axes. Having obtained the head orientation of the viewer, the video playback system will extract the corresponding visible portion of the video frame based on the field of view(FoV) parameters which define the extent of the viewing area.



**Figure 1:** Euler angles

360-degree video enable users to interact with the content of the video and enjoy smoother viewing experience. However, the delivery of 360-degree video is much more challenging than regular video. Because 360-degree video records a spherical surface around the viewing point, a full frame of a 360-degree video will contain 4-6 times of the number of pixels in one regular video frame, thus consumes much more bandwidth during video transmission. This increase will significantly burden the network traffic, especially over wireless networks which have limited bandwidth. While it is possible to relieve the network pressure by improving video encoding algorithm or using bit-rate adaptive delivery scheme[1,2], a more effective strategy is to avoid the transmission of the invisible portion of the video, as only about 20% area of the 360-degree video frame is displayed in front of the viewer. In this way, instead of downloading the entire video frame, the viewer only fetches parts that are visible to him/her. Meanwhile, the transmission itself takes time, thus the viewer will receive the packet demanded after certain delay. In order to keep the consistency between viewer’s head orientation and the transmitted video portion, the content provider or server need to predict the viewer’s head orientation.

While the idea of head movement prediction has already been proposed and adopted to reduce network traffic during 360-degree video delivery[3,4,6,7,8], the research community still lacks a 360-degree video delivery scheme designed for wireless network in a multi-user context.

In this paper, we will present our contribution by proposing a multi-user 360-degree video streaming scheme which exploits the underlying similitude of users’ reaction towards the video content by recognizing the conjoint viewport of multiple users – the portion of video which is visible to multiple users. The video server will predict the head orientation of multiple users, as well as their conjoint viewport patterns. Then, for each user’s unique visible portion, the server will transmit the data one-by-one to the user as usual. For the conjoint viewport which is shared by multiple users, the server will transmit the data over the public channel to all the target users, e.g., the multicasting mode over LTE network against the uni-casting mode. By doing so, the server avoids sending the chunks of conjoint viewport several times, and the global traffic over the network is thus reduced.

**2** RELATED WORKS

On major video platforms such as YouTube and Facebook, 360-degree video is encoded into the same video format as regular video, e.g., H.264 format in an MP4 container. However, each frame of a 360-degree video captures the scene in every direction from a unique point. There are several possible ways to project a spherical surface onto a planar rectangular frame，e.g., equirectangular panorama and cube map. Many researches have been made to compare different projection methods and improve them[9,10]. It is proved that cube map can reduce a certain amount of bandwidth consumption while providing the same quality of viewing experience. Meanwhile, they will not be discussed in this paper as we will focus on viewport-adaptive transmission approaches.

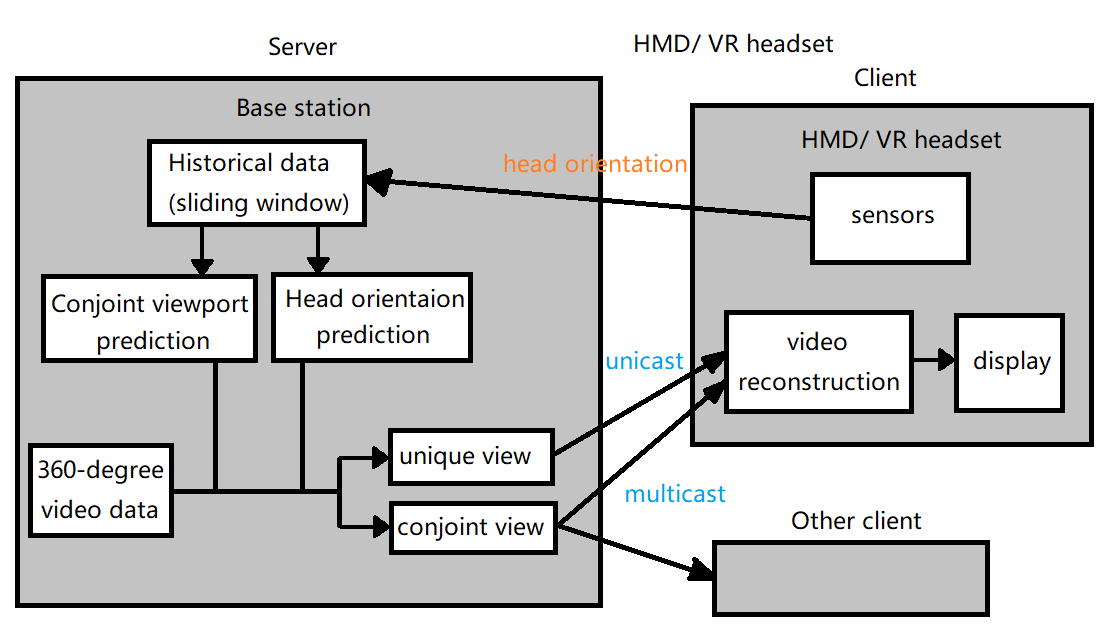
When watching a 360-degree video, a viewer is facing a certain direction at every moment, and his/her viewport covers only about 20% of the surrounding sphere. To avoid the transmission of unused portion, viewport-adaptive delivery schemes have been proposed to reduce bandwidth consumption[3-5]. In [3], each 360-degree video frame is pre-segmented into multiple smaller chunks called tiles, viewer only requests tiles that overlap with his/her viewport. In [5], each video is pre-processed to generate several representations with different bit-rate and quality, viewer will request for his/her viewport the version of high quality, and for invisible region the version of lower quality.

As the future viewport of user is unknown to the video server , the server has to predict user’s head orientation. In [4] and [3], the feasibility of head movement prediction is discussed and tested, and the results show that head movement is predictable in short term (0-0.5s). In [3] it is proved that by using a simple regression model such as linear regression, the prediction error of head orientation can already be limited within 10 degrees in each dimension for 95% of the test samples. This kind of error can easily be compensated by slightly enlarging the field of view downloaded. Other regression models such as ARMA and neural networks are also used to predict head movement.

Apart from works mentioned above, there are several other techniques that contribute to the research of 360-degree video. [11-13] proposed video bit-rate adaptation based on network bandwidth. Adaptation based on Quality of Experience is also studied[7,14,8]. These ideas could be useful in future works related to optimization of 360-degree video.

**3** SYSTEM STRUCTURE AND PRINCIPLE

Our proposed system is targeted at wireless network which consists of a base station and several users. The system is running on both server side(base station) and client side(user device). Below is an overview of the system architecture.



**Figure 2:** System architecture

The user should use certain HMDs or VR headsets to watch the video. In order to capture the head movement of the user, the HMD or headset should be equipped with sensors, gyroscopes, or accelerometers. Current state-of-art HMDs are able to record in real-time with high accuracy the rotation of user’s head, which are denoted by three Euler angles, pitch, yaw, and roll. The client side will periodically send head orientation information to the server side.

On the server side, the base station of the network will collect users’ head orientation information and store it in a sliding window. Machine learning methods will then be applied to predict users’ future head orientation based on previous records.

**3.1** Head orientation prediction

Assume that the head orientation is updated *n* times every second, and the sliding window stores data in the past *Δt* second, so there are *n* tuples of 3 angles in the sliding window:

(1)

where corresponds to three Euler angles.

The objective of head movement prediction is thus to predict the head orientation after *δ* second: . A regression problem is then formulated:

Given previous head orientations , predict future head orientation

Some existing schemes used linear regression(LR) model to solve the problem, which turned out to give considerable results when the time interval *δ* is below 0.5s. This is coincident with the wireless network condition in our context. Given that the base station is nearby, the round-trip time between server and client is normally limited under 50ms. Taking into account the processing delay, butter time and data transmission time, the total time interval between the request and the response is about 20-300ms, which is within the valid range of prediction.

In our scheme, in addition to linear regression, we use Long Short-Term Memory(LSTM), a recurrent neural network which is widely used to process sequences of data. A basic LSTM is composed of a cell and three gates. The cell stores some hidden states which act as the memory of LSTM, and the gates control the transition of the states. With this structure, LSTM can handle the exploding and vanishing gradient problems which are often encountered when processing sequences, thus is better at dealing with time series data than other traditional RNNs. The user’s head orientation information is a sequence of angle vectors, so we consider that LSTM is likely to have better performance than linear regression.

**3.2** Conjoint viewport prediction

Besides predicting users’ future head orientation, the server will also predict the conjoint viewport of users. Here, we define conjoint viewport as the region of video frame which is visible to multiple users at a given time. In existing schemes, the video data is transferred from server to each client separately, ignoring the content of transferred data, and as a result, the part of conjoint viewport will be transferred several times, which causes unnecessary waste of bandwidth.

In our scheme, the server will predict whether multiple users have conjoint viewport. Here, we assume that a conjoint viewport exists if the differences of yaw and pitch between the users are less than a certain threshold *θ*. The input data is the head orientation of all the users. This is thus a binary classification problem:

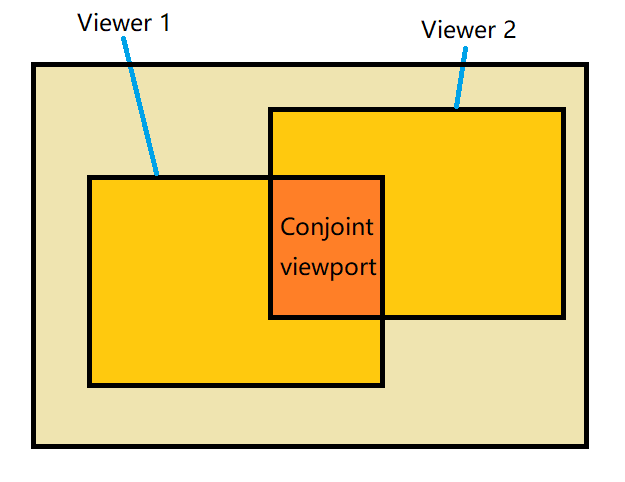
Given previous head orientation of multiple users:

, predict whether they have conjoint viewport at *t + δ*

It is also a problem with sequential features, so we also choose to use LSTM to solve it.

If a conjoint viewport is found, the server will calculate size and orientation of the conjoint viewport. Then, the server will use public transmission mode such as multicasting over cellular network to transfer the conjoint viewport data chunks. In this way, the conjoint part will be transferred only once in the network, which reduces the global network traffic. As for each user’s unique viewing portion, the server will transfer it in the usual way.

Upon receiving data from server, the client will decode and reconstruct the video frame according to associated metadata which stores orientation information about the transferred conjoint and private view. Finally, the client will extract the visible portion according to its current true orientation and display it to user.



**Figure 3:** Conjoint viewport

**4** IMPLEMENTATION AND EVALUATION

In order to train and evaluate the machine learning models in our system, we used the 360-degree video head movement dataset provided by [5]. The dataset contains the head movements of 63 viewers watching 7 different 360-degree videos. The videos are about 60s-100s long in time duration. The head orientation of viewer is captured 30 times every second and is stored in the form of quaternion. We processed the dataset and converted quaternion to Euler angles which is used in our models.

**4.1** Head orientation model

For head movement prediction, we implemented two models, linear regression and LSTM. The LSTM model contains one layer with 128 hidden units, the learning rate is set to 0.0001. For both models, the input sliding window contains 60 tuples of , which refers to the head movement in the past 2 seconds. The models are trained to predict the head orientation after 100ms. To evaluate the performance of the models, we used two indicators: root mean square error, and failure rate which represents the percentage of prediction with an error larger than 10 degrees. The comparison of results of the two models is shown in Figure 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Linear regression | | | LSTM | | |
| Dimension | Roll | Yaw | Pitch | Roll | Yaw | Pitch |
| RMSE | 2.35 | 3.93 | 2.28 | 2.28 | 3.80 | 2.26 |
| Failure rate | 0.86% | 3.23% | 0.68% | 0.82% | 3.19% | 0.64% |

**Figure 4:** Head orientation prediction

We can see that the prediction in yaw dimension is more error-prone. It is easy to understand because viewer tends to turns head more frequently in yaw dimension to see the surrounding environment when watching a 360-degree video. It is also shown the two models have similar accuracy when predicting head orientation, while LSTM performs slightly better than linear regression. This is probably due to the fact that the head movement has a very strong linear correlation in a short period, which is more important than its sequential feature.

**4.2** Conjoint view model

For the prediction of conjoint viewport, we considered the case of two users sharing a similar view. Here, we used LSTM again to do the prediction. The input data is a sliding window containing 60 tuples of , which is the head movement of the two users in the past 2 seconds. The model is trained to predict whether the two users will have a valid conjoint viewport after 100ms. In order to estimate our model (which we call one-step method), we also implemented a naive method for comparison: the naive method will first predict the future head orientation of the users (which is already done in section 4.1), and then calculate differences between the predicted orientations to judge whether there will be a conjoint viewport. The results are shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| threshold θ | 30° | 40° | 50° | 60° | 70° |
| Accuracy (one-step) | 92.5% | 92.6% | 93.2% | 93.7% | 94.1% |
| Accuracy (naïve) | 91.7% | 91.8% | 92.2% | 92.5% | 93.0% |

**Figure 5:** Conjoint view prediction

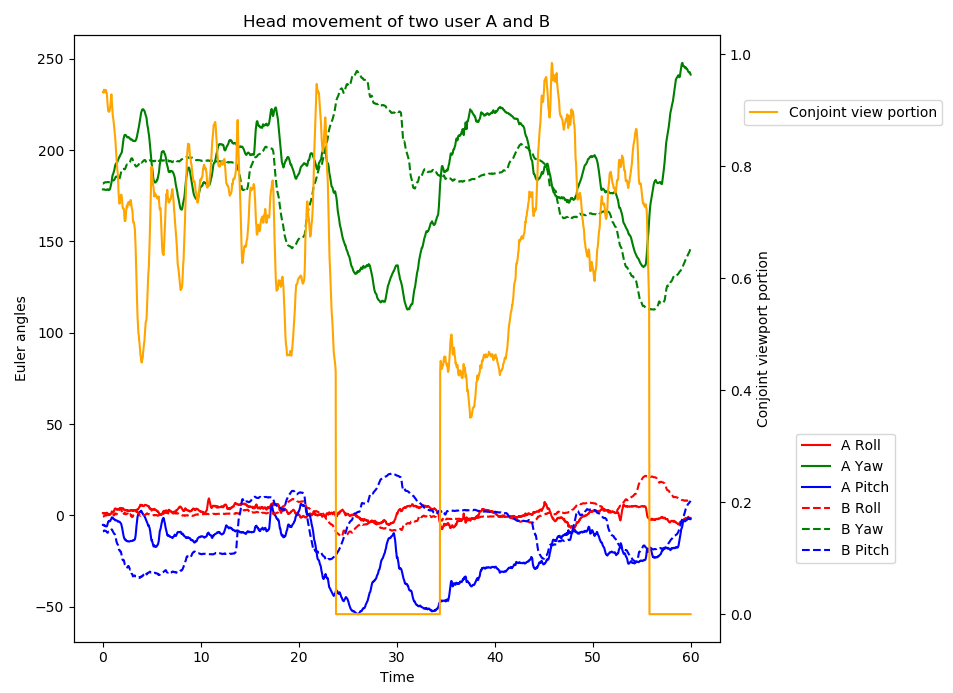
We can see that both methods achieve considerable results, and the one-step method is even better. Also, it should be noticed that a wrong prediction of the conjoint viewport will not degrade the viewing experience of user, because the conjoint viewport will only affect the mode of transmission.

For the cases of more than two users, the problem is more complex. One possible approach is to first apply the trained prediction model on each pair of users extracted from the whole set, then use the prediction result to divide the whole set into several subsets so that users in the same subset share a conjoint viewport. Currently we haven’t conducted a complete research on the cases of more than two users, it is to be completed in our future works.

**4.3** Bandwidth consumption estimation

In order to estimate the amount of bandwidth saved, we conducted a research on users’ head movement behavior. We chose 5 videos from the dataset, and we extracted the head movement of 10 users watching them. Then, for each pair of users, we calculated the portion of conjoint viewport during the playback: suppose that the field of view of each user is where and are the extent of viewport along yaw and pitch dimension, the differences of yaw and pitch between the two users are respectively and , then the portion of conjoint viewport can be estimated as:

An example head movement of two users and the portion of conjoint viewport are shown below, which prove that users tend to turn to similar viewing direction when watching the same video:



**Figure 6:** Conjoint view prediction

Given the portion of conjoint view during the playback, the percentage of bandwidth saved can be estimated as:

The average bandwidth saved for each video is shown in Figure 7:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| video | Driving | Paris | Rollercoaster | Timelapse | Elephant training |
| bandwidth saved | 15.4% | 13.8% | 28.7% | 9.6% | 20.3% |

**Figure 7:** Bandwidth saved

We observed that the performance of bandwidth optimization depends heavily on the content of the video. Meanwhile, the bandwidth is guaranteed to reduce by about 10% even in the worst case, which proves that the scheme is effective.

5 CONCLUSION

This paper presents a multi-user 360-degree video streaming system over wireless network. Based on head movement prediction, the system will recognize the conjoint viewport of multiple users using machine learning algorithm, and then transfer the content of conjoint viewport over the public channel of the wireless network to reduce global bandwidth consumption. The system uses LSTM to predict user’s head movement and conjoint viewport, which turns out to be effective and robust.

We show a two-user implementation in this paper. We are working on a further system implementation with multiple users. In such system users are divided into groups of same conjoint viewport, and the server transfers data to each group using separate multicastings. A multi-class classification model is used to get groups to this end.

REFERENCES

[1] Junchen Jiang, Vyas Sekar, and Hui Zhang. Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming. ACM CoNEXT, 2012.

[2] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. ACM SIGCOMM, 2014

[3] Feng Qian, Lusheng Ji, Bo Han, and Vijay Gopalakrishnan. 2016. Optimizing 360 video delivery over cellular networks. In Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges (ATC '16). ACM, New York, NY, USA, 1-6

[4] Y. Bao, H. Wu, T. Zhang, A. A. Ramli and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 2016, pp. 1161-1170

[5] Xavier Corbillon, Francesca De Simone, and Gwendal Simon. 360-Degree Video Head Movement Dataset. In Proceedings of ACM Multimedia System (MMSys) 2017

[6] Corbillon, Xavier & Devlic, Alisa & Simon, Gwendal & Chakareski, Jacob. (2016). Viewport-Adaptive Navigable 360-Degree Video Delivery.

[7] Reddy Gaddam V , Riegler M , Eg R , et al. Tiling in Interactive Panoramic Video: Approaches and Evaluation[J]. IEEE Transactions on Multimedia, 2016:1-1.

[8] Ghosh A , Aggarwal V , Qian F . A Rate Adaptation Algorithm for Tile-based 360-degree Video Streaming[J]. 2017.

[9] Next-generation video encoding techniques for 360 video and VR - Facebook Code\_files

[10] Under the hood\_ Building 360 video - Facebook Code\_files

[11] Junchen Jiang, Vyas Sekar, and Hui Zhang. Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming

[12] A. Hamza and M. Hefeeda, “Adaptive streaming of interactive free viewpoint videos to heterogeneous clients,” in Proceedings of the 7th International Conference on Multimedia Systems. ACM, 2016, p. 10

[13] J. Le Feuvre and C. Concolato. Tiled-based Adaptive Streaming using MPEG-DASH. In ACM MMSys, 20

[14] Pengye Xia and K. Nahrstedt. Object-level bandwidth adaptation framework for 3d tele-immersive system. 2013 IEEE ICME, pages 1–6, July 2013Conference Name:ACM Woodstock conferenceConference Short Name:WOODSTOCK’18Conference Location:El Paso, Texas USAISBN:978-1-4503-0000-0/18/06Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00