TILE-BASED QOE-DRIVEN HTTP/2 STREAMING SYSTEM FOR 360 VIDEO

Zhimin Xu[1], *Yixuan Ban*[4], *Kai Zhang*[4], *Lan Xie*[1], *Xinggong Zhang*[1,2,3], *Zongming Guo*[1,2], *Shengbin Meng*[5], *and Yue Wang*[5]

[1] Institute of Computer Science & Technology, Peking University, P.R. China

[2] Cooperative Medianet Innovation Center, Shanghai, P.R. China

[4] Beijing University of Posts & Telecommunications, Beijing, P.R. China

[5] Beijing Bytedance Nertwork Technology Co., Ltd, Beijing, P.R. China

ABSTRACT

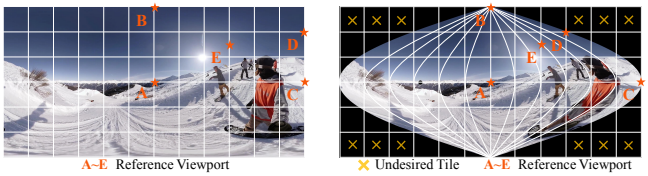
Recently, the 360-degree video has become a hot topic in multimedia area. However, the requirements of high bitrate, low Internet interactive latency and high perceived quality limit its further applications. So in this paper, we design a tile-based QoE-driven HTTP/2 streaming system for 360 video. It uses a new projection method, Sinusoidal Projection (SP), to reduce the bitrate of tiles. A novel cross-user's behavior learning method is also used to predict viewpoint. To improve bandwidth utilization, it pushes multiple tiles in one request by HTTP/2. Besides, by using a QoE-driven framework, our approach can significantly improve users' perceived quality. The numerous experiment results have demonstrated the efficiency of the proposed system. Compared with the legacy methods, the transmission bitrate drops about 17%, the viewport prediction accuracy improves 30%, the Viewport-PSNR improve 22% and the transmission latency drops about 30%.

Index Terms— HTTP/2, tile-based, Quality of Experience (QoE), k-push, HEVC, Sinusoidal Projection (SP)

1. INTRODUCTION

With the increasing demand for better user experience in interactive online virtual reality (VR) applications, how to deliver such high bitrates video over the Internet has become one of the urgent problems. With the development of High Efficiency Video Coding (HEVC), tile-based adaptive streaming [1–3] has become an ideal way to deliver 360-degree video by only delivering the dependent tiles covered by user's viewport.

However, even adopting the advanced tile-based adaptive streaming, there are so many challenges ahead of us. Firstly, *traditional tile-based Equirectangular Projection (ERP) video format exitsts a large amount of redundant data especially on poles.* Secondly, as is known to all, due to the critical requirements of low Motion-to-Photon latency in VR video browsing, client needs to predict user's viewpoint to prefetch the segments after $10 \sim 100$ seconds. However, *it is hard to predict user's head movement accurately.* It is still a real challenge for 360 video streaming, especially for long-term viewpoint prediction. Thirdly, given a available bandwidth, *how to allocate appropriate bitrates for each tile* and maximize perceived quality could be a fatal problem as well. Finally, delivering tiles dependently over the Internet will introduce *plenty of HTTP requests and response,* which will increase streaming latency and drop bandwidth utilization, especially when the Round-Trip Time (RTT) is large.



(a) ERP. (b) Sinusoidal projection (SP).

Fig. 1: Comparation of ERP and SP.

In this paper, we designed a tile-based QoE-driven HTTP/2 streaming system for 360 video, which address all aforementioned challenges. Our contributions are as follows:

- Sinusoidal Projection (SP) [4, 5]: we adopt a pseudo-cylindrical equal-area map projection, which is a lossless projection, and can significantly reduce the transmitted number of tiles and bitrates up to 17%, especially on poles area compared with traditional ERP format.
- Viewport prediction: we propose a learned-based viewport prediction approach [6], which utilizes cross-users' watching behaviors to improve the accuracy of viewport prediction up to 30%.
- Bitrate allocation: we propose a QoE-driven model to allocate bitrates for each tile [7]. The experiment shows that our approach improve Viewport-PSNR up to 22%.
- HTTP/2 k-push: we employ HTTP/2 push mechanism to push tiles set, all spatial tiles within one segment, upon a request. Compared to HTTP/1.1, it can improve

This work was supported by National Natural Science Foundation of China under contract No. 61471009 and Beijing Culture Development Funding under Grant No.2016-288.

[3]Corresponding author. E-mail: zhangxg@pku.edu.cn

978-1-5386-1737-3/18/$31.00

throughput up to 45% and reduce transmission latency about 30%, especially in high RTT mobile networks.

2. SYSTEM DESIGN AND MODEL

2.1. Sinusoidal Projection

As shown in Fig.1, in ERP, the vertical and horizontal lines represent longitude and latitude separately, which stretches the sphere onto a planar especially on poles. On the contrast, Sinusoidal Projection (SP) is a equal-area projection, which reduces a large of redundant area.

The projection is defined as:

$$\begin{aligned} x &= (\lambda - \lambda_0) \cdot \cos(\varphi) \\ y &= \varphi \end{aligned} \tag{1}$$

where φ is the latitude, λ is the longitude, and λ_0 is the central meridian. In SP, the black tiles without any information don't need to be delivered at all.

2.2. KNN-based Viewport Prediction

We adopt a KNN-based viewport prediction approach which exploits cross-users' watching behaviors to improve the accuracy of traditional linear regression (LR) [6]. As the region of interest in one video are similar for users, it is possible to exploit other users' viewing history to predict viewpoint.

Specifically, the client first implements LR method to find a possible fixation O_r. Then, other user's viewing history are exploited to find K fixations O_f nearest to O_r by K-Nearest Neighbors algorithm. Tiles' viewing probability is proportional to their viewing times. $L_i(O) = 1$ means tile-i is viewed while $L_i(O) = 0$ otherwise. To model that LR's prediction accuracy decreases rapidly as time horizon extends, we assign different weight to these fixations. Specifically, O_r's prediction weight is $w_r = \frac{1}{\delta}$. As for other K cross-users' fixations, we assign a constant weight $w_f = 1$. Tile's viewing probability V_i can be formulated as:

$$\begin{aligned} V_i &= w_r \cdot L_i(O_r) + \sum_{k=1}^{K} w_f \cdot L_i(O_f^k) \\ &= \frac{1}{\delta} \cdot L_i(O_r) + \sum_{k=1}^{K} L_i(O_f^k) \end{aligned} \tag{2}$$

Then, we can obtain each tile's viewing probability p_i.

2.3. Bitrate Allocation

We propose a QoE-driven bitrate allocation approach aiming at providing high video quality [7]. Specifically, the client side should firstly estimate available bandwidth for video streaming [7]. Then each tile's bitrates could be calculated based on *viewport prediction* results mentioned above immediately. One thing should be noticed that due to the spatial

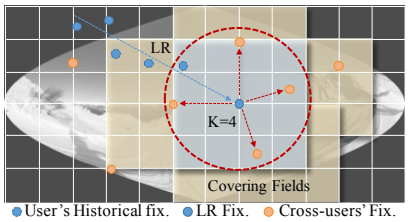


Fig. 2: KNN-based viewport prediction

partition of 360-degree videos, different tiles even encoded in the same bitrates could exist quality inconsistency, which makes picking tiles only by bitrates inadvisable.

Specifically, the original planar video are firstly divided into N tiles in raster-scan order spatially and cropped into continues segments temporally with the same duration T. After that, each segment should be encoded into M rate levels waiting for downloads on the server side. To derive the optimal selecting tile sets, we denote $i \in \{1...N\}$ and $j \in \{1...M\}$ as the tile index and rate level seperately. Then for certain tile i in j-th rate level, $r_{i,j}$ and $d_{i,j}$ represent the actual bitrates and quality distortion compared with original videos. Besides, we denote $\boldsymbol{X} = \{x_{i,j}\}$ as the choosing results, where $x_{i,j} = 1$ means the tile is selected and $x_{i,j} = 0$ otherwise.

To maximize user's quality while minimizing the spatial quality variance, the QoE problem can be formulated as [6]:

$$\begin{aligned} \min_{\boldsymbol{X}} \ & \Phi(\boldsymbol{X}) + \eta \cdot \Psi(\boldsymbol{X}) \\ \text{s.t.} \ & \sum_{i=1}^{N} \sum_{j=1}^{M} x_{i,j} \cdot r_{i,j} \le R, \\ & \sum_{j=1}^{M} x_{i,j} \le 1, \ x_{i,j} \in \{0,1\}, \ \forall i. \end{aligned} \tag{3}$$

Specifically, η represents the weight for quality variance while R representing the available bandwidth generated by rate adaptation algorithm, which is used to reduce rebuffering. The other limitation ensures each tile should be delivered in one rate in case of unnecessary waste.

2.4. HTTP/2 K-Push

The HTTP/2 Server Push feature, allows one server to push multiple segments with one request, is a mechanism designed for reducing web page load latency initially [8]. Leveraging this feature, *k-push* scheme is designed to stream 360-degree video, as the works in [3, 9, 10]. We also use HTTP/2 *k-push* mechanism to push a tiles set, which contains k tiles that compose a single temporal segment, to the client.

If the user need download K tiles in a certain FoV, it will send K requests in HTTP/1.1. Fortunately, HTTP/2 *k-push* can solve the request explosion problem, and improve network throughput, especially in high RTT network environment. In one push cycle, the client first determines the

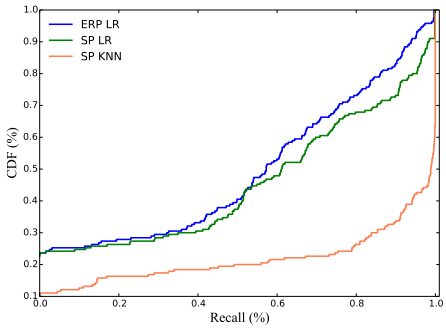


Fig. 3: Recall of viewport prediction

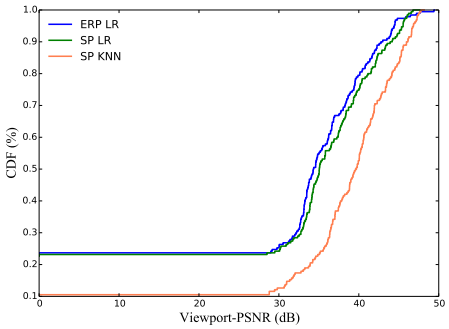


Fig. 4: CDF of Viewport-PSNR

tiles number K of next time. Then, the client sends one request with Push Directive to initiate a new *k-push* session and whereafter receives K tiles that concurrently pushed back by the HTTP/2 server. Compared to the HTTP/1.1, it only takes 1 request and at least $K-1$ RTTs are saved in one push cycle.

3. EXPERIMENTAL RESULTS AND ANALYSIS

3.1. System Setup

In our system, we develop our experiment by three components: (i). Media Encode Module, (ii). Server Module, (iii). Client Module. In Media Encode Module, we picked a video about skiing from [11], which contains head movement traces for 48 different users. To compare the performance on mapping format, we first mapped the source video into ERP and SP separately with resolution 2880×1440. Then, we used HEVC encoder to encode them into 6×12 equidistant tiles with five Quality Parameter (QP) levels $(22, 27, 32, 37, 42)$ as shown in Fig.1. Lastly, we encoded all tiles into 1sec segment. On Server Module, we developed two servers, an HTTP/2 server and an HTTP/1.1 server. The former supports the Server Push feature and will push the requested tiles together. The latter, on the contrary, will respond each tile's request independently. On Client Module, we develop a client supporting HTTP/2 push Directives. Besides, we also implement an OpenGL module, which supports ERP and SP rendering in dash player with FoV size $90^\circ \times 90^\circ$.

3.2. Performance of SP

Firstly, to demonstrate SP's feasibility, flexibility and effectiveness, we choose 5 representative reference viewports $A \sim E$ as shown in Fig. 1, which are spreading from prime meridian to international date line, from equator to poles. Constrained by the same parameters of bandwidth, encoding QP (at 22) and viewports, we calculate ERP and SP's needed tile numbers, bitrates and video quality represented by PSNR.

The results are listed in Table.1. Except for the viewport C, the SP can reduce the number of tiles within viewport tiles up to 12 in most cases. For point C, it request 20 tiles in SP as the 360-degree video's longitude curved. At the same time, SP only introduces slight quality dropping, which implies the SP is a better choice for 360-dgree video.

Table 1: Comparation of ERP and SP

Scheme		Tiles Number (tiles)	Tiles Total Size (kbits)	Viewport-PSNR (dB)
Point A	ERP	16	1856.7	45.85
	SP	16	1739.1	45.51
Point B	ERP	24	1821.9	51.84
	SP	16	1213.4	51.85
Point C	ERP	16	1505.4	48.35
	SP	20	2221.8	48.12
Point D	ERP	24	2170.4	50.30
	SP	22	1896.5	48.60
Point E	ERP	25	1196.1	50.75
	SP	13	580.8	49.65

3.3. Performance of KNN-based Viewport Prediction

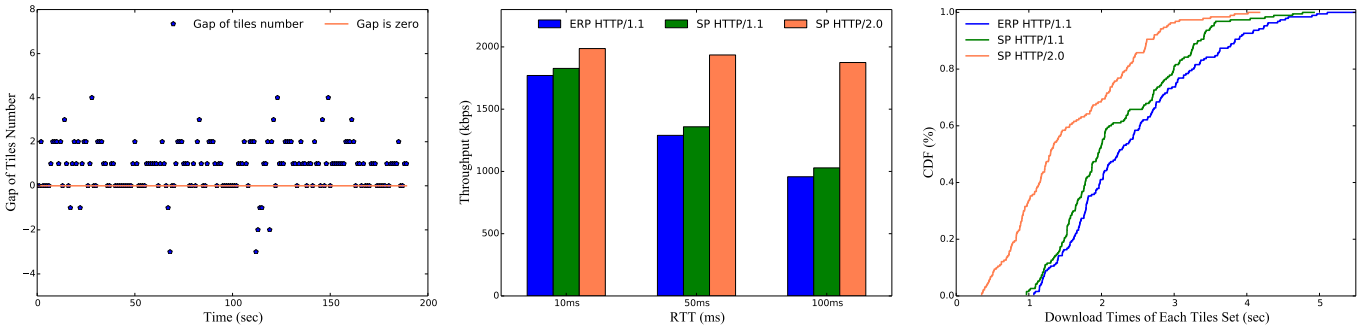
In this subsection, we compare our prediction method (KNN) with traditional LR method mapped in ERP and SP seperately with the same head trace under HTTP/1.1 with fixed bandwidth (2000kbps) and RTT (50ms).

To demonstrate the performance on prediction accuracy, we plot the Cumulative Distribution Function (CDF) of tile's recall for three methods (ERP with LR, SP with LR, SP with KNN). As shown in Fig.3, our method SP with KNN achieved the highest prediction accuracy obviously. Besides, the variation tendency for ERP with LR and SP with LR are closed to each other, which is because these two methods hold the same LR viewport prediction method. It also verifies the effectiveness and conclusiveness of KNN from another angle.

To demonstrate the efficiency of our QoE-driven HTTP/2 streaming system, we plot the CDF of Viewport-PSNR as shown in Fig.4. It shows that our method improve user's quality significantly. Because KNN-based prediction method has higher accuracy, it achieves higher viewport-PSNR than LR's.

3.4. Performance of HTTP/2.0

To compare the performance of ERP and SP methods under HTTP/1.1 and HTTP/2.0 seperately, we execute a experiment exploring their throughput under fixed bandwidth (2000kbps) with three different RTT (10ms, 50ms, 100ms).



(a) Gap of Tiles Number between the ERP and SP. (b) Throughput under different RTT (10ms, 50ms, 100ms). (c) CDF of Download Times for Each Tiles Set.

Fig. 5: Comparation of HTTP/1.1 and HTTP/2.0.

Furthermore, we calculated each tiles set's downloading time with the same head movement trace.

Firstly, we calculate the gap of tiles number between the ERP and SP methods as shown in Fig.5a. The results show that the SP method can reduce the necessary tiles number in most cases. However, there are some times that the SP method request more tiles, which becasue the user will seen region around point C (as shown in Fig.1) occasionally.

As shown in Fig.5b, the throughput of the SP method is little higer than the ERP method in HTTP/1.1. This is because the SP method sends less requests in most cases, which will improve throughput by saving unnecessary RTT. By using HTTP/2 k-push, the throughput of our method will improve up to 49% compared with the ERP method under HTTP/1.1 and 45% compared with the SP method under HTTP/1.1.

In Fig.5c, we calculate the CDF of download time for each tiles set during the same trace and RTT. It shows that HTTP/2.0 can save download time significantly because of saving request-response times by HTTP/2 k-push.

4. CONCLUSION

In this paper, we designed a tile-based QoE-driven HTTP/2 streaming system for 360 video by using a new projection method, Sinusoidal Projection (SP), a novel cross-user's behavior learning method (KNN) to predict viewpoint, and HTTP/2 k-push. We implemented it into a real system finally. The numerous experiment results demonstrated that the proposed method can achieve significant performance gain compared with the legacy methods. Through our approach, the transmission bitrate drops about 17%, the viewport prediction accuracy improves 30%, the Viewport-PSNR improve 22% and the transmission latency drops about 30%.

5. REFERENCES

[1] Alireza Zare, Alireza Aminlou, Miska M Hannuksela, and Moncef Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *ACM Multimedia (MM)*, 2016, pp. 601–605.

[2] Mario Graf, Christian Timmerer, and Christopher Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation," in *ACM on Multimedia Systems Conference*, 2017, pp. 261–271.

[3] Mengbai Xiao, Chao Zhou, Yao Liu, and Songqing Chen, "Optile: Toward optimal tiling in 360-degree video streaming," in *ACM Multimedia (MM)*, 2017, pp. 708–716.

[4] John Parr Snyder, *Map projections–A working manual*, vol. 1395, US Government Printing Office, 1987.

[5] Ramin Ghaznavi Youvalari, Alireza Aminlou, Miska M. Hannuksela, and Moncef Gabbouj, "Efficient coding of 360-degree pseudo-cylindrical panoramic video for virtual reality applications," in *IEEE International Symposium on Multimedia*, 2017, pp. 525–528.

[6] Y. Ban, L. Xie, Z. Xu, X. Zhang, and Z. Guo, "Cub360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming," in *Proc. IEEE International Conference on Multimedia & Expo (ICME)*, 2018, pp. 1–6.

[7] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360probdash: Improving qoe of 360 video streaming using tile-based http adaptive streaming," in *ACM Multimedia (MM)*, 2017, pp. 315–323.

[8] Christopher Mueller, Stefan Lederer, Christian Timmerer, and Hermann Hellwagner, "Dynamic adaptive streaming over http/2.0," in *IEEE International Conference on Multimedia & Expo (ICME)*, 2013, pp. 1–6.

[9] Stefano Petrangeli, Viswanathan Swaminathan, Mohammad Hosseini, and Filip De Turck, "An http/2-based adaptive streaming framework for 360 virtual reality videos," in *ACM Multimedia (MM)*, 2017, pp. 306–314.

[10] Mengbai Xiao, Chao Zhou, Viswanathan Swaminathan, Yao Liu, and Songqing Chen, "Bas-360: Exploring spatial and temporal adaptability in 360-degree videos over http/2," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2018.

[11] Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang, "A dataset for exploring user behaviors in vr spherical video streaming," in *ACM on Multimedia Systems Conference*, 2017, pp. 193–198.