Probabilistic Viewport Adaptive Streaming for 360-degree Videos

Zhimin Xu*, Xinggong Zhang*†‡, Kai Zhang§ and Zongming Guo*†
*Institute of Computer Science & Technology, Peking University, Beijing, P.R. China
§School of Computer Science, Beijing University of Posts & Telecommunications, Beijing, P.R. China
†Cooperative Medianet Innovation Center, Shanghai, P.R. China

***Abstract*—Recently, there has been a significant interest towards 360-degree virtual reality (VR) video. However, it is a big challenge for them to stream over Internet for huge bit-rates. In this paper, we have designed a novel viewport adaptive streaming scheme for 360-degree videos with probabilistic viewport prediction and optimal segments prefetching by Dynamic Adaptive Streaming over HTTP (DASH). In this way, continuous and smooth video playback, low viewport prediction error and high PSNR are obtained. To avoid head-movement prediction error, a probabilistic viewport prediction model is proposed, which leverages the probability distribution of user's orientation. Further, an optimal segments prefetching method is implemented. Finally, we also implement our method in a real system. The numerous experiment results have demonstrated that the proposed method has achieved significant performance gains compared with the existing methods. Our related work also win the Runner-up in *ICME 2017 DASH-IF Grand Challenge: Dynamic Adaptive Streaming over HTTP*.**

I. Introduction

With the increasing demand for better user experience in interactive online virtual reality (VR) applications, it has becomes one of the most significant impediment how to deliver high bit-rates' VR video over Internet. Due to the widely use of Dynamic Adaptive Streaming over HTTP (DASH) [2], viewport adaptive streaming, which only deliver viewport-dependent part of content to users, is a promising way to deliver 360-degree video to the end users [3].

Recently, there are many works for 360-degree video adaptive streaming. Such as [4–7], all of which are tile-based [8] viewport adaptive streaming. Due to the critical requirements of low Motion-to-Photon latency in VR video browsing, client need to prefetch video segments. However, it is hard to predict user' head movement accurately [9]. Though tile-based streaming is flexible in video content request, wrong prediction may arise black area in user's Field of View (FOV), reducing the quality of user experience (QoE). In [10], the author develop fixation prediction networks, which use some sensor- and content-related features to solve prediction problem. However the networks are time-consuming. What's more, tile-based streaming is need support by *x265/HEVC* [11]. Another work is about asymmetrical projection-based streaming [12], which is support by *x264/AVC* [13]. This method won't produce black area, but wrong prediction may lead to remarkable quality decline. How to streaming different viewport-dependent segment is a big challenge.

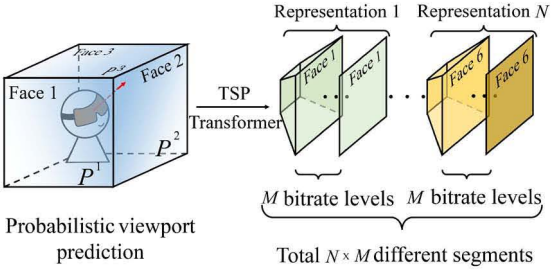


Fig. 1. System Diagram of Probabilistic Viewport Adaptive Streaming

In this paper, we apply Qualcomm's Truncated Pyramid Projection (TSP) [14] geometry which is compatible with MPEG OMAF. By which, we can avoid black area problem in tile-based streaming method as this format video contain all 360 degree content, and provide higher video quality in viewport. Meanwhile, which is support by *x264/AVC*, and we can easily achieve it in most of video players. To cope with the prediction error and viewport-dependent segment selection problem, we have designed a novel optimal viewport adaptive streaming scheme for 360-degree video, which leverages the probability distribution of user's orientation, and prefetches segments by maximizing the expected quality. In this way, the client may choose one or more viewport dependent representations to download, which avoid viewport quality decline when prediction error is large. By our proposed method, low stall, low viewport error and high viewport PSNR are obtained.

To verify our proposed method, we implement it in a real system by modifying standard MPD format in MPEG-DASH P1 [15], and extensive experiments are carried under controlled test-bed and real Internet trace. The novelty and performance can be summarized into two-folds:

- A small playback buffer and probabilistic viewport prediction model reduce stall and viewport error observably.
- A probabilistic viewport adaptive streaming is designed to provide high expected quality, and improve the viewport PSNR more than 30% compared with Equirectangular Projection (ERP) method [16] and more than 7% compared with TSP method.

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.

‡Corresponding author. E-mail: zhangxg@pku.edu.cn

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

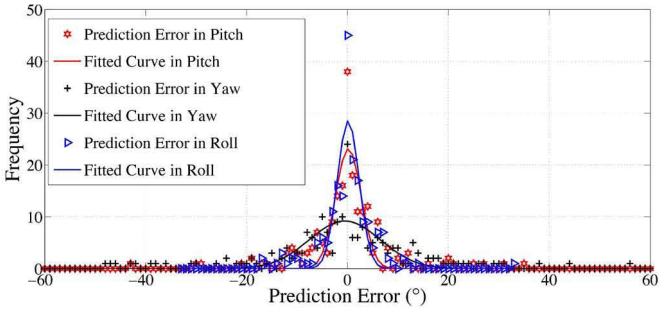


Fig. 2. Prediction Error Gaussian Distribution when $\Delta t = 3$s

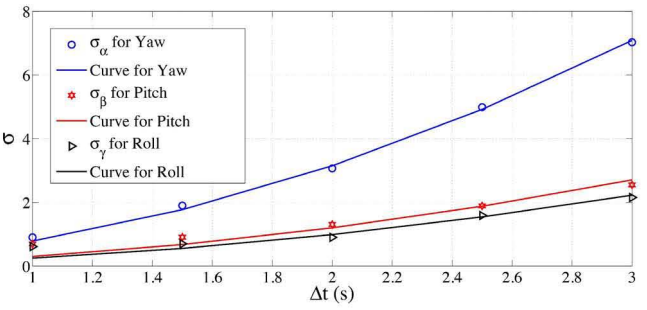


Fig. 3. Variation of σ_α, σ_β and σ_γ

What is noteworthy is that our related work also win the Runner-up in *ICME 2017 DASH-IF Grand Challenge: Dynamic Adaptive Streaming over HTTP* [1] in Hong Kong.

II. Model and Design

In this section, we develop a probabilistic viewport prediction model, and an optimal expected quality framework.

As shown in **Fig. 1**, we apply Qualcomm's Truncated Pyramid Projection (TSP) geometry, and transform 360-degree videos into N different viewport-dependent representations. For each viewport-dependent representation, it is further encoded into M bitrate version and partitioned into segments with same duration.

To predict user's orientation, we project a 360-degree video into a Cube, and observe the viewing probability of the six faces, as shown in **Fig. 1**. Assume a user sits inside a Cube, we predict the viewport from user's historical head-movement traces. Each point in the faces is assigned to some gazing probabilities. By averaging the points, we obtain the viewing probability of the faces.

After knowing the viewing probability of each face, we can calculate the expected quality of these TSP representations. One or more TSP viewport-dependent representations are prefetched according to the criteria of maximal expected quality.

A. Problem Formulation

Let $i \in \{1...N\}$ denote viewport-dependent representations and $j \in \{1...M\}$ denote bitrate levels. We define $r_{i,j}$ as the bitrate of segment(i,j), and $k \in \{1...K\}$ as the Cube's K faces on the sphere as show in **Fig. 1**. In this problem, we want to find the set of streaming segments, $\boldsymbol{X} = \{x_{i,j}\}$, while $x_{i,j} = 1$ denotes the segment of i-th viewport at j-th bitrate level is selected for streaming and $x_{i,j} = 0$ otherwise. We let $Q^k_{i,j}$ denotes the average quality on the k-th face, and the P^k denotes the viewing probability of the k-th face for the segment of i-th viewport at j-th bitrate level.

At each download step, our objective is to maximize the expected quality under the total bitrate budget R and segment selection constraints. Therefore, our optimization problem can be formulated as:

$$\begin{aligned} \max_{\boldsymbol{X}} \quad & \sum_{k=1}^{K} P^k \cdot \max\{x_{i,j} \cdot Q^k_{i,j}\} \\ \text{s.t.} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j} \cdot r_{i,j} \le R, \\ & \sum_{j=1}^{M} x_{i,j} \le 1,\ x_{i,j} \in \{0,1\},\ \forall i. \end{aligned} \tag{1}$$

The first constraint in the optimization problem restricts the total bitrate of selected segments, which can be calculated by any rate adaptation algorithm [17]. Then, the second constraint gives the restriction on $x_{i,j}$. Obviously, it only needs to select at most one bitrate level of each viewport representation.

From (1), we may choose one or more viewport-dependent representations to download. Thus client can playback befitting segment, which avoid poor quality when the user's orientation prediction is error. By solving the optimization problem (1), the probabilistic viewport adaptive streaming system can select the best segments to provide high user's QoE.

B. Probabilistic Model of Viewport Prediction

1) Viewport Prediction: In viewport adaptive streaming, since the client need to prefetch video segments which will likely be viewed by the user in the future, viewport prediction is requisite to estimate the user's head orientation.

We denote the user's orientation (Euler angle), as yaw (α), pitch (β) and roll (γ) and leverage Linear Regression (LR) model to do prediction. We denote t_0 as the current time of system. By using the historical samples, we apply Least Squares Method (LSM) [18] to calculate the trends of head movements. We denote the slope of the trend over yaw, pitch and roll as v_α, v_β and v_γ. Therefore, the Euler angle after Δt (the Δt is same as the current buffer length) can be predicted using Linear Regression model, $\hat{\alpha}(t_0 + \Delta t) = v_\alpha \cdot \Delta t + \alpha(t_0)$, $\hat{\beta}(t_0 + \Delta t) = v_\beta \cdot \Delta t + \beta(t_0)$, and $\hat{\gamma}(t_0 + \Delta t) = v_\gamma \cdot \Delta t + \gamma(t_0)$.

2) Prediction Error: It is well known that any head movement prediction method is inaccuracy. We use one real head movement trace, and plot the prediction error of the LR method in **Fig. 2**. It is noticed that the prediction error follows Gaussian Distribution, such as $e_\alpha \sim \mathcal{N}(\mu_\alpha, \sigma^2_\alpha)$. By curve fitting, mean μ_α and standard deviation σ_α can be learned.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

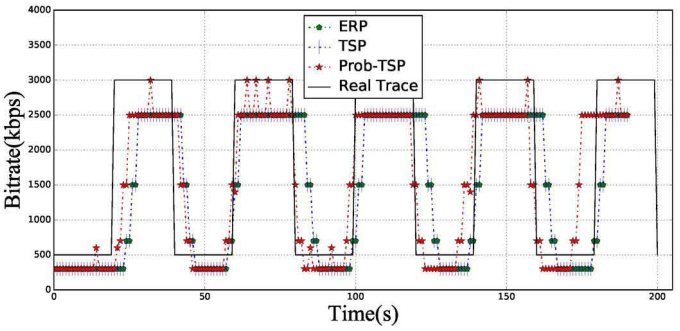


Fig. 4. Prefetching Bitrates under Long-term Bandwidth Variations

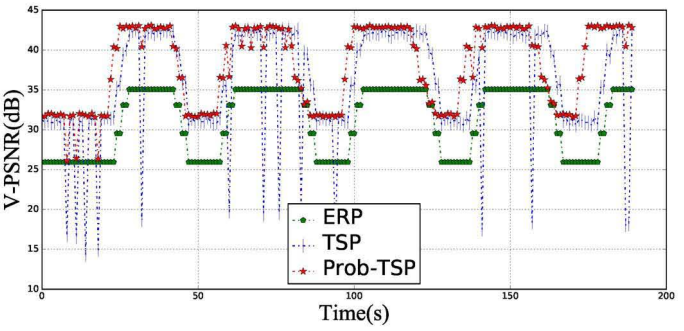


Fig. 5. V-PSNR under Long-term Bandwidth Variations

Therefore, the probability distribution of head movement Euler angle can be derived as:

$$\begin{cases} P_{\text{yaw}}(\alpha) = \frac{1}{\sigma_\alpha\sqrt{2\pi}}\exp(-\frac{(\alpha-(\hat{\alpha}+\mu_\alpha))^2}{2\sigma_\alpha^2}), \\ P_{\text{pitch}}(\beta) = \frac{1}{\sigma_\beta\sqrt{2\pi}}\exp(-\frac{(\beta-(\hat{\beta}+\mu_\beta))^2}{2\sigma_\beta^2}), \\ P_{\text{roll}}(\gamma) = \frac{1}{\sigma_\gamma\sqrt{2\pi}}\exp(-\frac{(\gamma-(\hat{\gamma}+\mu_\gamma))^2}{2\sigma_\gamma^2}). \end{cases} \quad (2)$$

Moreover, it is a reasonable assumption that long-term prediction would result into larger error deviation. Thus, we also plot the value of standard deviations against prediction duration Δt in **Fig. 3**. It is obvious that the standard deviation of prediction error is *strictly increasing* with prediction duration. By fitting the data we obtain the function between them as $\sigma_\alpha = \delta_\alpha \cdot (\Delta t)^2$, $\sigma_\beta = \delta_\beta \cdot (\Delta t)^2$ and $\sigma_\gamma = \delta_\gamma \cdot (\Delta t)^2$.

Since yaw, pitch and roll are independent of each other, the probability distribution of user's orientation can be obtained from **Equ. 2**, i.e. $P_E(\alpha,\beta,\gamma) = P_{\text{yaw}}(\alpha) \cdot P_{\text{pitch}}(\beta) \cdot P_{\text{roll}}(\gamma)$.

From the above probability of user's orientation, we can obtain the viewing probability of Cube's faces as shown in **Fig. 1**. We define a spherical point as (φ,θ). Since these spherical point could be distributed among K different TSP's faces, we define $\boldsymbol{L_k}(\varphi,\theta)$ as the set of points within the k-th face. For simple, we let the viewing probability of k-th face P^k equals to the average probability of orientations in $\boldsymbol{L_k}(\varphi,\theta)$ as:

$$P^k = \frac{1}{|\boldsymbol{L_k}(\varphi,\theta)|} \cdot \sum_{(\alpha,\beta,\gamma)\in\boldsymbol{L_k}(\varphi,\theta)} P_E(\alpha,\beta,\gamma) \quad (3)$$

III. System and Experiment

In this section, we firstly present the implementation of probabilistic viewport adaptive streaming system. To evaluate the performance of the proposed method, we also carry out extensive simulation experiments under various head movement traces and network conditions.

A. System Implementation

The system consists of media production, HTTP server and client. In media production, it contains the three key components, *TSP Asymmetrical Projector* transform a ERP video into TSP format by *360tools* [19], *Encoder* partitions and encodes each viewport video by *x264* [13] and *MP4Box* [20] and *Viewport-dependent MPD Generator* is designed to support viewport adaptive streaming by adding *@longitude* and *@latitude* attribute corresponding to standard representation definition in MPEG-DASH P1 [15]. For client implementation, we implement the video player based on the open source available MPEG-DASH *dash.js* [21] player and the open source *eleVR Web Player* [22] for 360-degree videos and implement our proposed probabilistic viewport adaptive streaming scheme. The client consists of the eight components, *MPD Parser, Bandwidth Estimator, Buffer Controller, Orientation Prediction, Viewport Adaptation, QoE-driven Optimizer, Rendering and Head Position Acquisition.*

B. Experimental Setup

In the experiments, we imitate user's head motion by embedding real user's head movement trace into the *dash.js* [21] player and actively manipulate the network conditions to observe how different schemes react to the network fluctuations. Specifically, we examine the performance on video sequence and 5 user's head movement traces on this video, which are generously provided by AT&T [9]. The sequence is about 3 minutes long with the resolution 2880×1440 in ERP format. For each viewport-dependent representation, it is further partitioned into segments with same duration 1 second. The bitrate levels of each segment are set as {300kbps, 700kbps, 1500kbps, 2500kbps, 3500kbps}. The video codec is the widely used open source encoder *x264* [13].

We select three typical 360-degree video streaming methods as the comparisons, they use the same buffer-based rate-adaptation method and bandwidth estimation method, ERP, which treats 360-degree video streaming as ordinary video [16], TSP, which uses Linear Regression method to predict future viewport and requests corresponding segments [23]. But it doesn't apply probabilistic viewport prediction. This is the baseline of probabilistic viewport adaptive streaming. Prob-TSP, which is the proposed method with probabilistic viewport prediction and optimal segments selection. It also uses the same Linear Regression in TSP method.In performance comparison, we also take three measurement metrics into consideration, stall, viewport PSNR (V-PSNR) [24], which directly indicates the quality of content in the user's viewport, viewport error, which indicates the incorrect times of prefetching segments due to viewport prediction error.

C. Experiment Results and Analysis

1) Experiments with Long-term Bandwidth Variations: We evaluate the three methods under the case of long-term bandwidth variations (from 500kbps to 3000kbps, each bandwidth

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

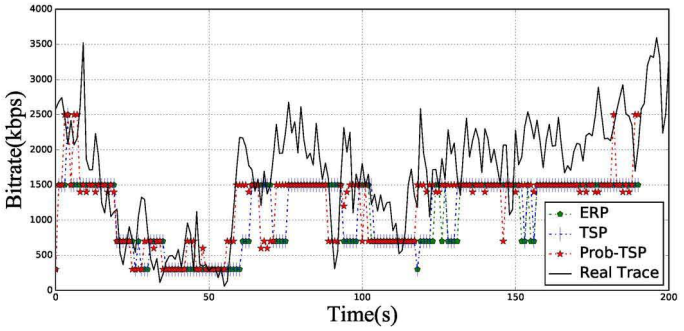


Fig. 6. Prefetching Bitrates under Real Internet Trace

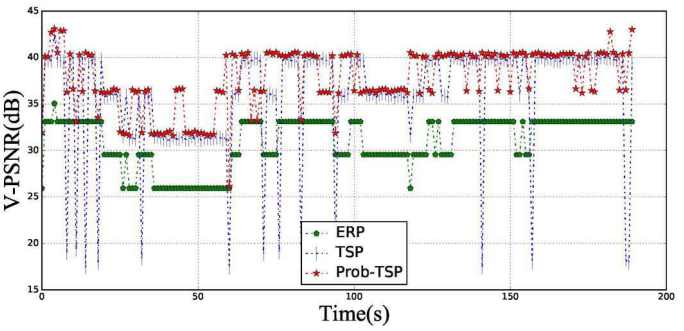


Fig. 7. V-PSNR under Real Internet Trace

TABLE I
PERFORMANCE ON LONG-TERM BANDWIDTH VARIATIONS

Methods	ERP	TSP	Prob-TSP
Average Bandwidth (kbps)	1452.87	1453.09	1503.15
Stall (Times)	6	8	7
Viewport Error (Times)	0	14	4
Average V-PSNR (dB)	24.45	35.07	37.32

TABLE II
PERFORMANCE ON REAL INTERNET TRACE

Methods	ERP	TSP	Prob-TSP
Average Bandwidth (kbps)	1084.29	1109.42	1176.96
Stall (Times)	4	4	3
Viewport Error (Times)	0	14	4
Average V-PSNR (dB)	24.72	35.06	37.73

sustain 20 seconds). We select one trace from the five head-movement traces randomly, and replay it for all of them.

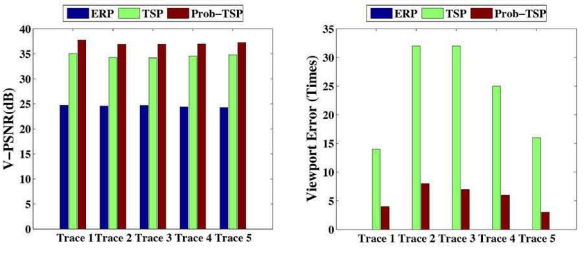
The prefetching bitrates are shown in **Fig.4**. All methods demonstrate the similar responsiveness to bandwidth variations, because they adopt the same buffer-based rate-adaptation method. It is also noticed that the proposed Prop-TSP sometimes achieves a few higher bitrates than the others. It is because the proposed Prop-TSP would prefetch two representations with different viewport if the expected quality is maximized. As show in **Fig.5**, ERP has the lowest viewport quality because it delivers whole 360-degree picture, while the rest twos are viewport-dependent streaming. TSP demonstrates the same quality as Prob-TSP when the viewport prediction is accurate. But when it is not, the proposed method demonstrates its merits. It achieves higher V-PSNR due to the proposed probabilistic viewport prediction method. The average performances of bitrates, stall times, viewport error and V-PSNR are shown in **Table.I**. The proposed Prob-TSP outperforms others significantly.

2) Experiments with Real Internet Trace: Secondly, we evaluate the three methods with a real Internet trace, which selected from [25]. Furthermore, we also repeat one same head movement trace from all methods.

Fig.6 shows the bit-rates of prefetching segments. All methods show almost same performance since they have the same rate adaptation algorithm. At some points, the proposed Prop-TSP achieves higher bit-rate with the same reason as **Section.III-C1**. **Fig.7** demonstrate the viewport quality with V-PSNR. Our Prob-TSP method is able to compensate for big variation on head movement using our optimization formula and reduce the viewport prediction error. As show in **Table.II**, our Prob-TSP method also achieved highest bandwidth utilization, low stall, lower viewport error and higher average V-PSNR. Besides, an smoothing V-PSNR variation is provided.

3) Experiments with Different Head Movement Trace: We also evaluate the three methods with five different head movement traces, to observe the effects of viewport prediction method. For each experiment, we use the same real Internet.

For different head movement trace, Prob-TSP method also achieved better performance as show in **Fig.8**. The ERP method achieved the lowest average V-PSNR under different 5 head movement traces, and our Prob-TSP method obtained the highest average V-PSNR more than 7% compared with TSP method.Our Prob-TSP method achieved lower viewport error compared with the TSP method, which further validate the efficiency of the proposed Prob-TSP viewport-adaptation method.



(a) Average V-PSNR (b) Viewport Error

Fig. 8. Results on Five Different Head Movement Trace

IV. CONCLUTION

In this work, we have designed a novel probabilistic viewport adaptive streaming system for 360-degree videos and implemented it over *dash.js* [21] and *eleVR Web Player* [22]. The numerous experiment results have demonstrated that the proposed method has achieved significant performance gains compared with the existing methods. Through our approach low stall, low viewport error, and high V-PSNR obtained. What's more, our related work also win the Runner-up in *ICME 2017 DASH-IF Grand Challenge: Dynamic Adaptive Streaming over HTTP* [1] in Hong Kong.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

REFERENCES

[1] Z. Xu, L. Xie, X. Zhang, H. Hu, Y. Ban, and Z. Guo, "Optimal viewport adaptive streaming for 360-degree videos," in *IEEE International Conference on Multimedia and Expo Grand Challenge*. IEEE, 2017.

[2] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *ACM MMSys*. ACM, 2011, pp. 157–168.

[3] E. Kuzyakov and D. Pio, "Next-generation video encoding techniques for 360 video and vr," available online: https://code.facebook.com/posts/1126354007399553.

[4] M. Hosseini, "View-aware tile-based adaptations in 360 virtual reality video streaming," in *IEEE Virtual Reality*. IEEE, 2017, pp. 423–424.

[5] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "Hevc tile based streaming to head mounted displays," in *IEEE Annual Consumer Communications & Networking Conference*. IEEE, 2017, pp. 613–615.

[6] Y. Sanchez, R. Skupin, C. Hellge, and T. Schierl, "Spatio-temporal activity based tiling for panorama streaming," in *the Workshop on Network and Operating Systems Support for Digital Audio and Video, ACM*. ACM, 2017, pp. 61–66.

[7] R. Ju, J. He, F. Sun, J. Li, F. Li, J. Zhu, and L. Han, "Ultra wide view based panoramic vr streaming," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, ACM*. ACM, 2017, pp. 19–23.

[8] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using mpeg-dash," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 41.

[9] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[10] C. L. Fan, J. Lee, C. Y. Huang, K. T. Chen, and C. H. Hsu, "Fixation prediction for 360 video streaming in head-mounted virtual reality," in *the Workshop on Network and Operating Systems Support for Digital Audio and Video, ACM*. ACM, 2017, pp. 67–72.

[11] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. W., "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2013.

[12] Z. Chao, Z. Li, and Y. Liu, "A measurement study of oculus 360 degree video streaming," in *ACM on Multimedia Systems Conference*. ACM, 2017, pp. 27–37.

[13] "x264," available online: http://www.videolan.org/developers/x264.html, 2017.

[14] E. Kuzyakov, "End-to-end optimizations for dynamic streaming," available online: https://code.facebook.com/posts/637561796428084.

[15] ISO/IEC 23009-1, "mpeg-dash p1," available online: http://standards.iso.org/ittf/PubliclyAvailableStandards/c057623_ISO_IEC_23009-1_2012.zip.

[16] "Youtube live in 360 degrees encoder settings," available online: https://support.google.com/youtube/answer/6396222, 2017.

[17] C. Zhou, X. Zhang, and Z. Guo, "A control-theoretic approach to rate adaptation for dynamic http streaming," in *Visual Communications and Image Processing (VCIP)*. IEEE, 2012.

[18] Wikipedia, "Least squares method," available online: https://en.wikipedia.org/wiki/Least_squares.

[19] Vladyslav Zakharchenko, "360tools," available online: https://github.com/Samsung/360tools.

[20] GPAC, "Mp4box," available online: https://gpac.wp.imt.fr/mp4box.

[21] DASH Industry Forum, "dash.js," available online: http://github.com/DASH-Industry-Forum/dash.js.

[22] Andrea Hawksley, "elevr-web-player," available online: https://github.com/hawksley/eleVR-Web-Player.

[23] M. Inoue, H. Kimata, K. Fukazawa, and N. Matsuura, "Interactive panoramic video streaming system over restricted bandwidth network," in *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010, pp. 1191–1194.

[24] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *2015 IEEE ISMAR*. IEEE, 2015, pp. 31–36.

[25] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video streaming using a location-based bandwidth-lookup service for bitrate planning," in *ACM Trans. Multimedia Comput. Commun. Appl (TOMCCAP)*. ACM, 2012, p. 24.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE