Viewport-Driven Rate-Distortion Optimized Scalable Live 360° Video Network Multicast

Ridvan Aksu
The University of Alabama
Tuscaloosa, AL 35487

Jacob Chakareski
The University of Alabama
Tuscaloosa, AL 35487

Viswanathan Swaminathan
Adobe Research
San Jose, CA

***Abstract*—Virtual Reality (VR) technologies enable remote scene 360° video immersion experiences. The growing popularity of VR increases the demand for 360° video delivery over the Internet. Compared to regular videos, 360° videos are characterized by an enormous data volume and spatial user navigation. User interactivity is activated by head movements and changes the spatial portion of a video viewed by the user, hence making only a small portion of the video essential at a time. Therefore, streaming the full video at high quality causes suboptimal use of the bandwidth. The above properties of 360° videos require novel streaming techniques in order to maintain high Quality of Experience (QoE). Live 360° multicast has not being studied yet, due to the emerging nature of 360° video, and it represents one of the most promising applications. Relative to on-demand single user 360° video streaming, it presents new challenges such as handling the interactions of multiple users simultaneously, while dynamically encoding the live 360° video on the fly.**

We propose a novel scalable multicast live 360° video streaming framework. It comprises a rate-distortion analysis that captures the fidelity-rate trade-offs of 360° videos, an optimization formulation to assign data rates to spatial video regions, and a scalable 360° video data representation. A 2-3 dB of quality gain for lower bandwidth classes and 3-4 dB for higher bandwidth classes are observed over a conventional method that streams the monolithic content at uniform quality. Our work shows that for live 360° multicast achieving quality levels close to on-demand streaming is possible despite the lack of information about future content and user navigation actions.

I. Introduction

Developments in VR display and omnidirectional capture technologies enabled the 360° video format. Omnidirectional cameras or camera rigs can capture 360° videos that provide a 360° look around of the surrounding scene and Head Mounted Display (HMD) allows a remote users to experience them. However, streaming a 360° video using state-of-art standards has numerous challenges.

For any given video frame, HMD displays a viewport, which represents only a small portion of the entire 360° video panorama, while the rest of it is not viewed. Using traditional video streaming techniques results in a bandwidth waste since a viewport comprises of less than 1/6 of the entire 360°. In addition, most broadband networks are not capable of streaming a 360° video in a good quality. Several studies have been carried out on improving 360° video delivery. Although these studies improve streaming on-demand 360° video, new challenges arise in the case of live 360° multicast.

The work of J. Chakareski and R. Aksu has been supported in part by NSF award CCF-1528030 and a research gift from Adobe Systems.

Recently, CNN started its VR news service [1] and live streamed the 2017 total solar eclipse. For widespread commercial live 360° multicast applications, there are additional obstacles to overcome. First, there are no pre-encoded 360° video regions in different qualities to send users with various demands as in the case of on-demand video. Additionally, there is no prior history of user navigation actions that can be used for viewport prediction, given that the video is being streamed for the first time. To overcome these challenges, we propose a novel scalable live 360° multicast framework.

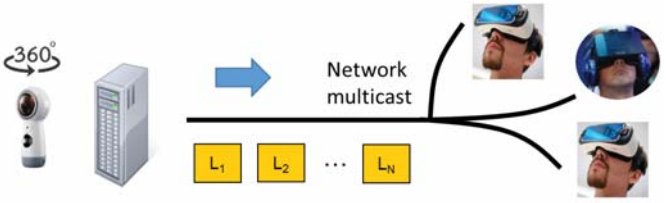


Fig. 1: Multicast 360° live streaming: N layers are streamed.

In this study, we use equirectangular tiling to partition a 360° video into smaller spatial tiles so that the viewport can be processed individually from the rest of the video. We analyze the rate-distortion characteristics of the partitioned video tiles as introduced in our recent work [2], to develop an analytical model that captures the effect of tile data rates on the reconstructed 360° video quality. Moreover, two methods for tile navigation likelihood prediction are explored based on the user head movements: (I) a neural network, trained on the different video frames and head traces, (II) and a pyramidal prediction method. Based on their network bandwidth, users are distributed into different multicast classes. Predicted tile navigation likelihoods of class users and rate-distortion characteristics of tiles are used to assign video qualities to every tile such that the aggregate 360° video quality is delivered to each class. The assigned qualities are then used to encode the tiles in a scalable fashion such that every scalable layer captures the network bandwidth difference between classes. A user in class k receives the tiles in scalable layers up to L_k from the N streamed layers as illustrated in Figure 1.

The rest of the paper is organized as follows. We review related work in Section II. In Section III, the components of the proposed system framework are discussed. Section IV develops a problem formulation that finds the optimal tile qualities that result in maximizing the aggregate expected viewport 360° quality across all classes. Section V demonstrates the performance of our framework. Finally, Section VI discusses future work and concludes the paper

II. Related Work

360° video streaming has various challenges due to its omnidirectional nature. According to De Simone et al. challenges through the omnidirectional video communication chain are distortions during the capture, projection distortion, rendering the viewport, and losses during the encoding and streaming [3]. In terms of projection, equirectangular projection (ERP) is a popular approach [2, 4, 5] due to its simpler implementation. Other types of projections are also investigated e.g. cubemap [6], non-uniform [7], and resolution-defined [8].

Probabilistic prediction of the viewport is proposed in several works [4, 5]. Qian et al. and Nasrabadi et al. use linear regression to predict the viewport [9, 10]. We used navigation action history of previous users to predict the viewport in our recent work [2].

Another approach for maintaining high viewport quality is using Scalable Video Coding (SVC) [11]. Nasrabadi et al. proposed using SVC in 360° videos and stream all video tiles in base layer quality while viewport tiles in enhancement layers [10]. According to He et al. SHVC causes only a negligible loss in quality compared to HEVC while reducing the bit stream size around 87% [12]. Multicasting of 360° videos are studied by Ahmadi et al. [4].

There are various implementations in terms of QoE assessment. According the Chen et al. planar PSNR calculation causes distortion and can be addressed with various implementations [13]. However, De Simone et al. states that there are still limitations of proposed QoE assessment methods [3].

Despite the recent popularity and potential of 360° videos there is not an agreed standard on live 360° multicast. We propose an analytical and scalable solution to multicast the live 360° videos to fill this gap.

III. System models

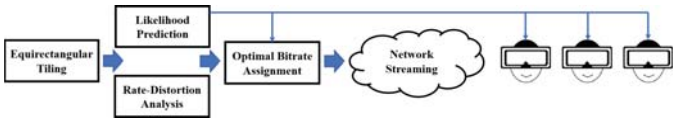


Fig. 2: System model.

A. Overview

Users' viewports change over time in accordance with user head movements. To achieve a satisfactory QoE at the user end, pixels within the viewport should be in high quality. Also, delivering all the pixels outside the viewport in high quality limits the network bandwidth and redundant. However, two challenges arise in the case of solely viewport transmission. Firstly, encoding pixels in irregular shapes is not a suitable approach in state-of-art encoding technologies, so residual pixels outside the viewport are unavoidable. Additionally, precisely determining the viewport is not trivial since the server cannot receive the precise viewport location at all times. If a portion of the viewport is failed to be transmitted, that results in a sudden drop of QoE and causes VR sickness.

The proposed system is composed of the following steps: First, recorded video frames are partitioned into equirectangular tiles. Then, rate-distortion characteristics of tiles are analyzed. Meanwhile, user navigation actions are used to assign tile likelihoods of being in viewport. Using these characteristics and likelihoods optimal tile bitrates are assigned for each scalable user class. Finally, calculated bitrates are assigned to the tiles as quality enhancements in scalable fashion (Figure 2).

B. Video Tiling

A 360° video panorama is between 4-8 times of a user viewport. Partitioning a 360° video into $N \times M$ equirectangular tiles allows treating each video tile as individual video blocks. Individual blocks help to isolate the viewport and achieve an overall non-uniform video quality. Once determined, viewport tiles can be encoded in high quality and the rest are encoded in low quality, allowing an efficient usage of bandwidth.

We partitioned our videos into medium sized 6×4 tiles. Larger tiles would benefit less in terms of bandwidth gains and smaller tiles would generate larger manifest files and decreases the encoding gain which are out of scope of this paper. Figure 3 shows the video tiling in a video panorama.

Fig. 3: Tiling of 360° video.

C. Tile Likelihoods

Position of the viewport is strictly controlled by the user's head movements. Receiving the viewport position by using real-time head trace data is not feasible as discussed by Qian et al [9]. In order to achieve a high QoE, viewport should be predicted in advance.

User QoE depends on the quality of the tiles that are within the actual viewport. Presence of a tile in the viewport can vary. Spatially, tiles with larger area in the viewport should receive higher quality. Temporally, a tile can be present in the viewport only a few consecutive frames. Since consecutive frames are encoded together for efficiency, for a Group of Picture (GOP) presence of tiles can change. So, each tile should be encoded in a quality that is proportional to its fraction of the likelihood of presence during the GOP. Let $w_{k,j}$ denote the surface area of the viewport occupied by the tile k at frame j. Distorted nature of the ERP stretches the area of the tiles closer to the poles. Normalized likelihood in each frame is calculated using $\bar{w}_{k,j} = w_{k,j} / \sum_k w_{k,j}$ to account for this distortion. Finally, we sum these normalized likelihoods for all the frames in a GOP.

Although using non-uniform quality across the viewport might result in undesirable quality variance, determining tile

qualities with respect to rate-distortion characteristics helps smoothening it. Also, tiles with very small portion in the viewport do not considerably affect the quality variance.

D. Likelihood Prediction

1) Pyramidal Approach: Since determining the actual viewport is not possible for each frame, we can use the initial tile of a GOP as a basis for viewport prediction. [14] states that users stay within 1 *orthodomic* distance (i.e. the angular distance from the center) from viewport center 90% of time in one second. This shows us that in 90% of the time, viewport stays around the tiles surrounding the center tile.

Since a viewport is expected to be around the center tile for the following GOP, we can assign likelihood of being in the viewport to the tiles symmetrically. We implemented *pyramidal* approach introduced by Ahmadi et al [4]. In this approach, tile likelihoods decrease gradually from the center tile to outwards linearly like a pyramid. So, the quality of the possible viewport tiles is kept high.

2) Neural Network Approach: Compared to the static nature of the pyramidal approach, a neural network can actively predict the tile likelihoods for the GOP t_i given the head trace of the user in the GOP t_{i-1}. The network generates a curve fitting from the head navigation trace of a time chunk to the tile likelihoods of the next chunk. Although contents of different videos can be unrelated, given the chunks are small, it is very likely that similar head movements observed in the time chunks t_{i-1} lead to similar tile likelihoods for the time chunk t_i.

The prediction works as follows: The neural network reads the yaw and pitch data of the viewport center for all frames in the GOP t_{i-1} as the input. Then it outputs the likelihood of tiles in the GOP t_i. Since fitting is a non-linear operation, some tiles are assigned with negative or very small likelihoods. We set the likelihoods smaller than a threshold to 0. Threshold is defined as 5% of the total likelihood in a GOP. This helps saving the bandwidth by ignoring the tiles that are unlikely to end up in the viewport.

The neural network consists of 15 hidden layers. There are 64 inputs (32 sets of yaw and pitch, per frame) and 24 outputs (likelihood of 6×4 tiles) for the network (Figure 4). The network is trained using Bayesian regularization.

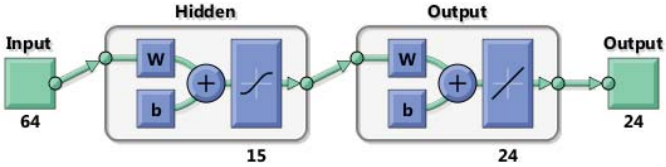
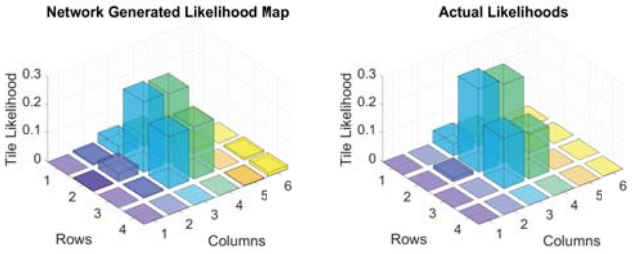


Fig. 4: Neural network to predict viewport likelihoods.

Figure 5 shows the comparison of tile likelihoods generated by the neural network (5a) and actual likelihoods (5b) for a random GOP. Actual likelihoods are calculated using the actual head traces, and are used to train the network. Compared to actual likelihood map, neural network assigns excess likelihood to some extra tiles. Although these extra tiles use the bandwidth, majority of the weight is in the actual viewport tiles with a smoother distribution.



(a) Likelihoods by Network (b) Actual Tile Likelihoods

Fig. 5: Comparison of tile likelihoods.

E. Rate-Distortion Analysis

We employed Quantization Parameter (QP) of the codec to alter the video quality. Analytical dependency between QP and the bitrate of a given tile helps using the QP parameter to decide the video size. Similarly, an analytical dependency between video bitrate and distortion allows us to determine the video quality given the bitrate of the said video. Using these two relationships we can determine the expected distortion using the QP variable in the optimization function in section IV-B.

In our previous work we have investigated the rate-distortion (R-D) and QP-bitrate (QP-R) relationships for the videos we have used [2]. In the study, we have shown that both characteristics can be interpolated as power law or exponential functions as in equations 1 and 2.

$$R = a_1 e^{-b_1 \,\mathrm{QP}} \quad \text{or} \quad R = a_2 \mathrm{QP}^{b_2}. \tag{1}$$

$$D = c_1 e^{-d_1 R} \quad \text{or} \quad D = c_2 R^{d_2}. \tag{2}$$

In the paper, we investigated the videos and showed that the QP-R relationship is an exponential function $R = a_1 e^{-b_1 \,\mathrm{QP}}$ and the R-D relationship is a power law function $D = c_2 R^{d_2}$. Although these relationships can vary for the video, analyzing the video characteristics results the correct form for each particular case. Determining the R-D and QP-R relationships for each video tile in real time is out of this paper's scope.

F. Scalable Video

Multicast allows one copy of the video data to be sent to many users simultaneously to save bandwidth on the server side and the Internet pipelines. In case of the 360° video, users are expected to have distinct viewports. Considering the various network bandwidth levels of the users, each user has a distinct optimal tile quality set. Using traditional approach results in one of two scenarios: Either users with higher bandwidth will receive lower quality tiles than their network bandwidth is capable of or users with limited bandwidth will try to stream high quality content and result in buffer.

For multicast, we classify the users based on their bandwidths to solve this problem. For each user class there is a scalable video level to increase the quality. Each level is comprised of tile qualities that are optimal for the corresponding user class. The base layer video has the optimal tile qualities that serves the minimum distortion for the lowest class users. Each enhancement layer has enhancement tiles for existing tiles and new tiles, based on the video quality. Thus, each

user class receives video layers for the corresponding class and lower layers. This allows the server to send copies of the video limited to the number of scalable layers instead of one copy for each user.

Quality of the tiles in each class is calculated using the optimization discussed in section IV. Since optimality of each class is dependent on the viewport of that class users, each layer serves for a minimum total distortion for the mentioned class.

IV. Optimization

A. Problem Setup

Given the variables that have been discussed so far, we can derive the problem of optimal bitrates for each layer. Constraints of the problem are the class bandwidth limits and the minimum and maximum QP levels.

Users are divided into classes based on their bandwidths. Class of a user determines the total number of scalable levels that user receives. Given the bandwidth requirements of each user class, bitrates of the tiles should be allocated to ensure the minimum aggregate distortion. Let there are *K* classes with increasing bandwidths $C_0, C_1, \ldots, C_{k-1}$. C_0 being the base layer bandwidth, each user is assigned to class k where the user's bandwidth is in between C_{k-1} and C_k. So, for each class *k*, the following inequality should hold:

$$\sum_t R_{t,k}(QP_{t,k}) \leq C_k,\ t = 1, \ldots, N \times M,\ \forall k. \tag{3}$$

Here $R_{t,k}(QP_{t,k})$ stands for the bitrate of tile *t* of class *k* for the corresponding QP value. Since we use scalable levels, $R_{t,k}$ stands for the cumulative rate of tile *t* for class *k*. Which means each $R_{t,k}$ inherits the bitrate of the lower class $R_{t,k-1}$.

The non-linear relationship of QP-R limits the available QP values. So, we introduce upper and lower bounds for QP in our optimization function.

$$R_{t,k}(QP_{max}) \leq R_{t,k} \leq R_{t,k}(QP_{min}),\ \forall t, k. \tag{4}$$

Our aim is to maximize the total quality of all users' viewports. In order to achieve that, we can minimize the aggregate distortion of the expected viewport of all users. $p_{t,i}$ is the likelihood of tile *t* for user *i* as discussed in section III-C. Using the likelihoods of all users results in an aggregate distortion of all user viewports.

Distortion of a tile $D_{i,t}$ can be expressed as a function of bitrate $R_{i,t}$ and bitrate of a tile *t* used in class *k* can be expressed as a function of the QP value $QP_{t,k}$ as discussed in section III-E. So, our target is to minimize $\sum_i \sum_t D_{t,i}(R_{t,i})p_{t,i}$ that will result in maximum aggregate viewport quality.

B. Optimization Formulation

Now that we have the constraints and the system model, we can formulate the optimization problem as follows:

$$\begin{aligned} \min_{\{R_{t,k}\}} \quad & \sum_i \sum_t D_{t,i}(R_{t,i})p_{t,i}, \\ \text{subject to:} \quad & \sum_t R_{t,k}(QP_{t,k}) \leq C_k,\ \forall t, k. \\ & R_{t,k}(QP_{max}) \leq R_{t,k} \leq R_{t,k}(QP_{min}),\ \forall t, k. \end{aligned} \tag{5}$$

In the objective function, *i* represents the user and *k* represents the class of users. So, since each user is in one of these classes, there are $R_{t,k}$ values for each tile of each class and $R_{t,i}$ corresponds to the bitrate of that user's class.

Since this is a convex optimization problem, it can be solved efficiently. Solving this optimization for each time chunk returns optimal tile bitrates for each class. Using the QP-R model, we can encode each level and generate a continuous QP value for each level. Then tiles are encoded in the nearest integer QP value and streamed accordingly.

V. Experiments

A. System Setup

We have tested our architecture using three popular 4K 360° videos from YouTube namely Coaster [15], Wingsuit [16], and Dolphin [17]. Using an Oculus Rift HMD device and OpenTrack software [18] we recorded a total of 111 head movement traces including yaw, pitch, and roll angles of the HMD direction with a time stamp for 3 videos. The frame rate of the HMD is 250 trace per second. We considered one trace data per video frame.

Videos are partitioned into 6×4 spatial tiles as discussed in section III-B and temporally composed of 1920 frames and 60 GOPs. We have shown the R-D and QP-R models of the videos in our previous study as power law and exponential functions respectively [2]. The process of generating the model is as follows: Tiles are encoded in 5 QP levels (22, 27, 32, 37, 42). Then bitrate and distortion of each tile are measured and R-D and QP-R relationships are extracted using Matlab.

User tile likelihoods are predicted using the pyramidal approach in Section III-D1 for *Pyr* and the neural network discussed in Section III-D2 for *NN*. For each video, the corresponding neural network is trained using the traces of the other two videos. Networks are trained using a windowed fashion where position vectors (yaw and pitch) of each consecutive 32 frames are used as the input (64 elements) and the tile likelihoods during the next 32 frames are used as the output (24 elements). Networks are trained using more than 172,000 samples per video. For the Pyramidal approach initial tile of each GOP is assigned with 0.25, adjacent 4 tiles are assigned with 0.125, and the diagonal 4 tiles are assigned with 0.0625 likelihood.

Two scalable layers and two corresponding user classes are used for our architecture: a base layer (*Base class*) and an enhancement layer (*Enhanced class*) with three and two users respectively. In a real scenario users can dynamically move between classes and total number of users can change however we decided to follow static user allocation for the sake of simplicity. We mimic scalable structure using High Efficiency Video Codec (HEVC) for this study. The tiles are encoded in different QP levels using x265 [19], a fast application of HEVC.

Finding the optimal tile qualities using *NN* and *Pyr* is as follows: First, R-D and QP-R relationship model is generated using 5 QP values as discussed earlier. Tile likelihoods of each

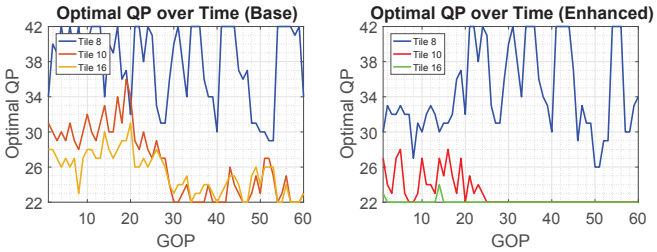
user is predicted using the corresponding method. Optimization constraints are then determined: 42 is chosen as the upper bound and 22 as the lower bound of QP. The final constraints are the class network bandwidths. For a fair comparison, the bitrate of the monolithic videos discussed below are used as bandwidth limits of classes. Finally, using optimization function, predicted likelihoods and calculated bandwidths are used the find minimum aggregate distortion.

We implemented *Monolithic* as our reference architecture which is based on streaming a monolithic 360° video encoded in constant QP value. For two user classes, we have selected QP values that result in 2-3 dB differences in viewport Y-PSNR. For each scenario, the bitrate of each video (*Base class* and *Enhanced class*) is calculated per GOP. And for the corresponding GOP, these bitrates are used as the bandwidth constraint of the respective class in our proposed architecture.

Finally, for the QoE assessment we used normalized Y-PSNR values. According to [13] regular 2D PSNR calculation introduces distortion due to 3D to 2D projection. So, we calculated the MSE of Y values for each viewport pixel and normalized it with the viewport size since it changes depending on the pitch angle and calculated the normalized Y-PSNR from there.

B. Optimal Tiles

Optimal $R_{t,k}$ for each class k and tile t is calculated using the equation 5 for each GOP. Using the QP-R relationship, optimal $QP_{t,k}$ values are calculated. Figure 6 compares the two classes of the Coaster video in terms of the QP values of tiles 8, 10, and 16 from figure 3. Overall QP in the *Enhanced class* is lower by 3-5 units than the *Base class* for the majority of the video. However, in some GOPs, the *Base class* and the *Enhanced class* are appointed with the same QP values. Between GOPs 20 and 40, tile 8 has the same trend in both cases indicating that increasing its quality in the *enhanced class* does not further improve the total quality without affecting other tiles or it is not a popular tile among the *Enhanced class* users. Tile 16 is assigned the minimum QP value in general for the *Enhanced class*. This shows that tile 16 is more likely to be in the viewport of *Enhanced class* users. Tile 10 becomes more popular in both cases and shows a 3 QP difference in between after GOP 25.

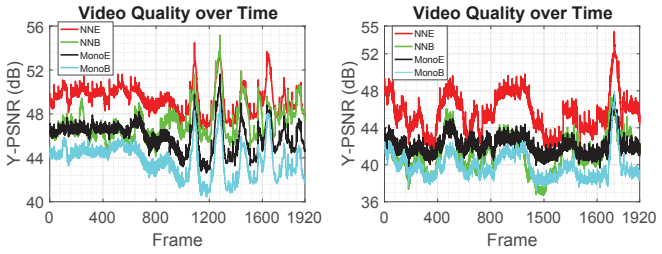


(a) Base Class (b) Enhanced Class

Fig. 6: Optimal QP of various tiles over time

C. Evaluation Results

The trend of PSNR over time is shown in figure 7 for the Coaster and Wingsuit videos. Here, *NN* indicates the neural network predicted optimal results, and *Mono* is the monolithic case. *E* and *B* are for the *Enhanced class* and *Base class* respectively. *Mono* case shows a smoother trend in the first half of the Coaster video as a result of temporally less varying bitrate than the second half. Overall, the base class of the *NN* is similar to the enhanced class of the *Mono*. In the Wingsuit video, in worse frames *NN* and *Mono* has very close results, but in better frames the difference increases above 4 dB. *Mono* in the Wingsuit video shows overall a smoother trend than the Coaster video, indicating a temporally smoother bitrate variation.



(a) Coaster (b) Wingsuit

Fig. 7: 360° viewport video quality

Figure 8 and 9 compares the change of the average PSNR values of the viewport while the average bandwidth changing in Coaster video and the Wingsuit video respectively. *NNE* is the *Enhanced class* of our proposed architecture, and *NNB* is the *Base class*. *PyrE* and *PyrB* are the enhanced and the base class results of the pyramidal likelihood assignment respectively. Finally, *MonoE* and *MonoB* are for the monolithic case discussed in section V-A. *Enhanced class* here indicates the average of the 2 users in that class, and *Base class* is for the average of the 3 users of the *Base class*. Lines represent the average value PSNR value over the whole video for the average bandwidth of the scenario. In the x-axis, the first value is the average bandwidth allocated for the *Enhanced class*, and the value in the parenthesis is the average bandwidth of the *Base class*.

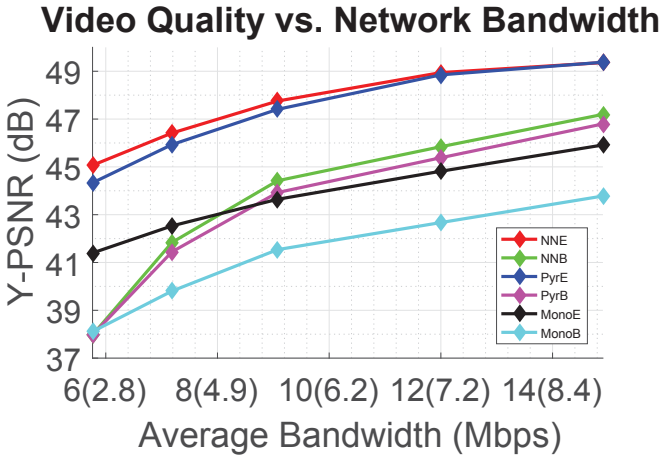


Fig. 8: Average video quality - Coaster

Enhanced and base classes of the *Monolithic* case shows 2-3.5 dB difference in average, showing an example of two different scalable levels. Difference between PSNR of two

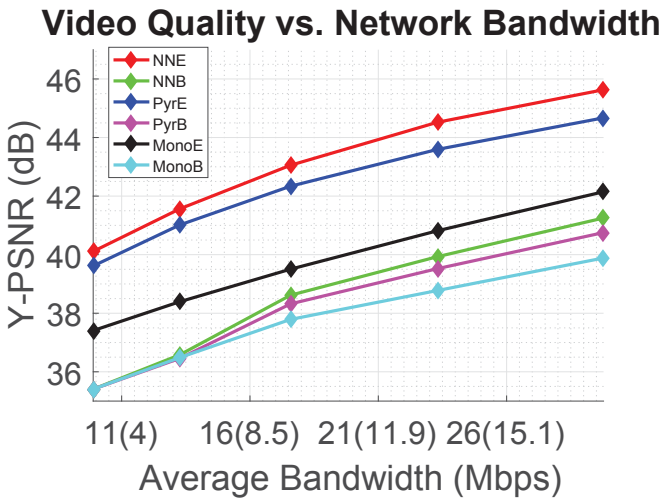


Fig. 9: Average video quality - Wingsuit

NN classes show a steady difference of 3-4 dB in higher bandwidths. This difference increases in lower bandwidth levels as lower bandwidths are only capable of streaming the lowest quality.

Gain of the *NN* from the *Mono* is around 3.5-4 dB in *Enhanced class* for the Coaster video and 2.5-3.5 dB in the Wingsuit video. In *Base class*, this gain is slightly less. Respectively 2-3 dB and 1-1.5 dB differences in higher bandwidth scenarios decays to 0 at lower bandwidth scenarios. So, proposed architecture leads to gains in both classes in regular network speeds while maintaining the needs of the multicast streaming.

The neural network slightly outperforms the pyramidal approach in both classes until it reaches the high bandwidth cap. A 0.5 dB difference is observed overall in two methods in the Coaster video and 1 dB in the Wingsuit video. This shows the overall advantage of the neural network.

Compared with the 4-5 dB gain of our previous work [2], a 3.5-4 dB gain in *Enhanced class* and a 2-3 dB gain in *Base class* shows that streaming live 360° video can be very close to the regular streaming considering the losses of scalable streaming.

VI. Conclusion

360° video delivery is an emerging topic in video communication. Live 360° multicast is one of the most promising application of it with additional challenges. In this work, we investigated live multicast considering its popularity in regular video communications. We predicted users' navigation actions with two prospective methods, analysed rate-distortion characteristics of video spatiotemporal video segments, and employed a scalable representation to deliver the 360° video regions in various quality levels. As a result, same content can be multicast to many users that are grouped by their network bandwidth constraints with a quality gain of 3 dB for low network bandwidth class users and 3.5-4 dB for high network bandwidth class users.

Correctly predicting user navigation actions is one of the challenges in 360° video communications. Considering the black-box nature of neural networks, further prediction improvements can be achieved. Using video content itself or exploiting the saliency conditions of the video [20] can lead to a better prediction method.

Using very small tiles can decrease the quality variance within the user viewport and increase the network bandwidth gain further by limiting the excessive high quality area outside the viewport. Commercial applications of very small tiles show that it can be possible to use very small tile without losing encoding efficieny [21].

References

[1] "CNN-VR." [Online]. Available: http://www.cnn.com/vr

[2] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, "Viewport-driven rate-distortion optimized 360 video streaming," in *Proc. IEEE ICC*, May 2018.

[3] F. D. Simone, P. Frossard, C. Brown, N. Birkbeck, and B. Adsumilli, "Omnidirectional video communications: new challenges for the quality assessment community," *VQEG eLetter*, vol. 3, no. 1, pp. 18–24, Nov 2017.

[4] H. Ahmadi, O. Eltobgy, and M. Hefeeda, "Adaptive multicast streaming of virtual reality content to mobile users," in *Thematic Workshops in ACM MM*, Mountain View, CA, Nov 2017.

[5] 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 *Proc. ACM MM*, Mountain View, CA, Nov 2017.

[6] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware adaptive 360° video streaming using tiles for virtual reality," in *Proc. IEEE ICIP*, Beijing, China, Sep. 2017.

[7] M. Xiao, C. Zhou, Y. Liu, and S. Chen, "Optile: Toward optimal tiling in 360-degree video streaming," in *Proc. ACM MM*, Mountain View, CA, Nov 2017.

[8] C. Dunn and B. Knott, "Resolution-defined projections for virtual reality video compression," in *IEEE VR*, March 2017, pp. 337–338.

[9] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. Workshop on All Things Cellular: Operations, Applications and Challenges*. New York City, NY: ACM, Oct. 2016.

[10] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using scalable video coding," in *Proc. ACM MM*, Mountain View, CA, Nov 2017.

[11] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sept 2007.

[12] G. He, J. Hu, H. Jiang, and Y. Li, "Scalable video coding based on user's view for real-time virtual reality applications," *IEEE Communications Letters*, vol. 22, no. 1, pp. 25–28, Jan 2018.

[13] Z. Chen, Y. Li, and Y. Zhang, "Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation," *Signal Processing*, vol. 146, pp. 66–78, 2018.

[14] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. IEEE ICC*, Paris, France, May 2017, pp. 1–7.

[15] "Mega coaster: Get ready for the drop (360 video)." [Online]. Available: https://youtu.be/-xNN-bJQ4vI

[16] "Wingsuit 360 degree video over Dubai." [Online]. Available: https://youtu.be/AX4hWfyHr5g

[17] "Wild dolphins VR / 360 video experience." [Online]. Available: https://youtu.be/BbT_e8lWWdo

[18] "Opentrack: Head tracking software." [Online]. Available: https://github.com/opentrack/opentrack

[19] "x265 HEVC encoder." [Online]. Available: http://www.x265.org

[20] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in VR: How do people explore virtual environments?" *IEEE TVCG*, 2018.

[21] R. Monnier, R. van Brandenburg, and R. Koenen, "Streaming UHD-Quality VR at realistic bitrates: Mission impossible?" May 2017. [Online]. Available: https://goo.gl/P4Hp8T