Content-Aware Optimization of Tiled 360° Video Streaming Over Cellular Network

Saurabh S Dange, Shashwat Kumar, Antony Franklin A.
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad
Telangana, India
{cs19mtech11028@iith.ac.in, cs15resch11011@iith.ac.in, antony.franklin@cse.iith.ac.in}

***Abstract*—The proliferation of 360° video content and easy availability of head-mounted devices propelled the popularity of 360° video streaming on various platforms, including smartphones. However, the high bandwidth requirement of 360° video streaming impedes its further growth, specifically in bandwidth constraint cellular networks. Some viewport adaptive 360° video streaming methods have been proposed recently to reduce the bandwidth requirement by spatially reducing the transmitted data. However, these methods require accurate prediction of the future viewport, and a prediction error can significantly impair the user's Quality of Experience (QoE). In this work, we took a different approach and propose a content-aware solution for tiled 360° video streaming that does not require viewport predictions. In this work, we exploit the fact that the background scene does not contain prominent details and can be transmitted at a lower bit rate without affecting the user's QoE. The proposed approach uses the saliency map to identify prevalent tiles of the 360° video through machine learning-based classification using the Kernel Density Estimation (KDE). First, the tiles are classified based on their saliency information, and then bitrates are assigned to the tile clusters. The results show that the proposed approach reduces the bandwidth requirement by 50% while delivering the equivalent user's QoE.**

***Index Terms*—360° video streaming, Saliency Map, Bitrate Allocation**

I. Introduction

While 360° videos were introduced as early as 1900 [1] by Disney, it gained vast popularity recently with the introduction on major streaming platforms such as Facebook and YouTube. The 360° video provides an immersive experience, which can not be achieved through planar video. In 360° video, a spherical view is presented to create an immersive experience, where a viewer can rotate his/her head, with 3-degree of freedom (yaw, pitch, and roll), to view the scene in any direction. The size of 360° videos are larger than the regular videos, which requires higher bandwidth for transmission. This is the biggest challenge for 360° video streaming, and it is imperative to address this issue to enable high-quality 360° video streaming over bandwidth-constrained wireless networks, specifically cellular networks.

In 360° video streaming, a user views only a small part, called Field-of-View (FoV), of the whole spherical 360° view in a video frame. The tile-based streaming [2], [3], [4] exploits this characteristic to reduce the bandwidth requirement of 360° video streaming by either omitting the Out-of-Sight (OOS) tiles or transmitting them in lower quality. To accomplish it, the 360° video segments are spatially divided into tiles, which can be encoded and transmitted independently. In the tile-based 360° video streaming, the bandwidth requirement is reduced by transmitting only the tiles which overlap with the user's FoV [5], [3]. However, the client needs to buffer some future video segments for smooth playback. Therefore, viewport adaptive tiled 360° video streaming requires the prediction of the user's future viewports. Existing solutions employ various ways to deal with the OOS tiles in a video segment. While some methods only fetch the FoV [5] tiles, others transmit OOS tiles in the lowest quality [3] or gradually decrease the quality of OOS tiles subject to their Euclidean distance from the FoV.

Although existing methods significantly reduce the bandwidth requirement of 360° video, unsolicited prediction errors would adversely impact the user's Quality of Experience (QoE). When there is a viewport prediction error, a viewer observes the blank tiles in the FoV or experiences a re-buffering, which is notably detrimental to the user's QoE. To overcome this challenge, we propose a different approach that uses the content features which does not require FoV prediction to reduce the bandwidth requirement. The proposed solution exploits the fact that the viewers do not pay much attention to the non-salient region in the video frames, and streaming quality of these parts does not significantly affect the video quality [6]. The proposed content-aware bitrate selection solution assigns the bitrate to individual tiles on the basis of saliency information and reduces the bandwidth requirement without affecting the user's QoE. The tiles containing numerous salient points are assigned a higher bitrate, while the tiles with lesser salient points are streamed in the lower bit rate to attain a better user's QoE. First, using Kernel Density Estimation (KDE), the tiles are segregated into priority classes based on the number of salient points, then the bitrates are assigned to these classes. In contrast to the existing works, which primarily use the relative position of OOS tiles for bitrate assignment, the proposed approach uses the content features to assign tile bitrate irrespective of location. The results show that the proposed method significantly reduces the bandwidth requirement of 360° video streaming without exposing it to frequent re-buffering due to tile miss on prediction error.

The rest of the paper is structured as follows. The existing works which are related to the present study are reviewed in

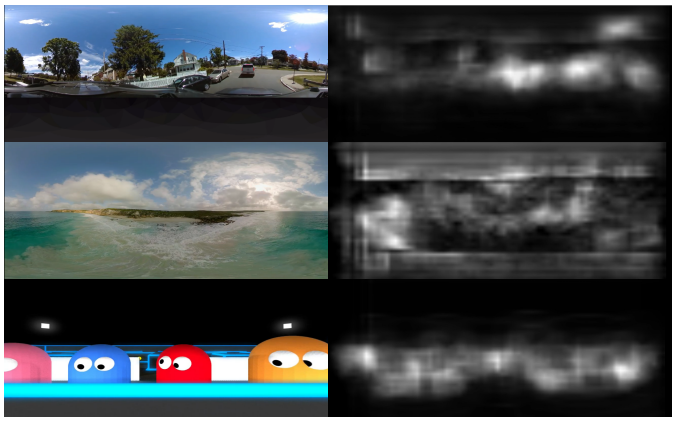


Fig. 1: 360° Video frames and their corresponding saliency maps, indicating a correlation between the details in the frame and the salient points.

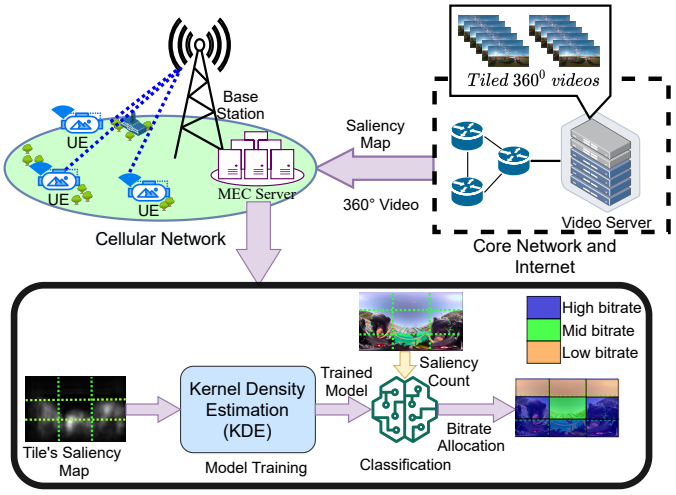


Fig. 2: System diagram of the proposed solution being deployed at MEC server on the base station.

Section II. The details of the proposed solution are discussed in Section III, explaining the proposed bitrate allocation of tiles. The implementation details and results are presented in Section IV. Finally, the work is concluded in Section V.

II. Related Work

Le Feuvre [7] proposed an approach for adaptive streaming of 360° videos using tiling, which uses the Spatial Representation Description (SRD) feature of DASH [8]. SRD enables the spatial division of video segments into numerous tiles, enabling the quality selection of a specific part of the frame independently through tiling. Various viewport prediction-based approaches [5], [3], [2] take advantage of the fact that when the users watch 360° videos, they only see the small area of the frame at a time, also called the viewport. Flare [3] only fetches the part of the frame which is inside the viewport. Rubiks [5] proposes an approach that can be implemented on the smartphone as traditional methods are computationally heavy for the smartphones. Parima [2] uses past viewports of users along with the trajectories of prime objects to predict future viewports. Viewport-based techniques stream viewport in the highest quality even if only a small part of viewport contains the region of interest. The proposed approach tries to save bandwidth by identifying a region of interest even in the viewport to save bandwidth. Multi-access Edge Computing (MEC) plays a crucial role in offloading the heavy computational load of the client system on the nearest edge server. Lo [9] proposes an edge-assisted 360° video streaming system, which leverages edge servers to render viewports of 360° videos for viewers. The edge server identifies the region of interest and assigns priorities to the video chunks for transmission based on current bandwidth availability.

Content-aware adaptive streaming has been explored previously in the context of planar video compression. These methods identify Region of Interest (ROI) using saliency map and adjust Quantization Parameter (QP) of video chunks during encoding. Hadizadeh *et al.* [10] presents a saliency-aware video compression method that enhances the saliency of the ROI region after compression and decreases the saliency of the OOS region. Zhu *et al.* [11] propose a rate-distortion calculation method to choose the pattern and guide the allocation of bits during the video compression process. Sun *et al.* [12] presents a content-aware rate control scheme for High Efficiency Video Coding (HEVC) based on static and dynamic saliency detection. It allocates specific QP to some parts of the frame using saliency importance as a variable. These ROI-based techniques try to compress the video by analyzing video content to distribute bitrate for video compression. Unlike these works, which deal with video compression, we propose a solution to reduce the bandwidth requirement of tiled 360° video streaming by utilizing the saliency map for tiles prioritization through clustering and bitrate allocation.

III. Proposed Approach

As discussed earlier, 360° video streaming requires high bandwidth, which is difficult to attain in wireless networks. In this work, we propose a solution that reduces the bandwidth requirement while delivering high-quality tiled 360° videos. Viewport adaptive streaming techniques reduce the bandwidth requirement by omitting the OOS tiles while streaming or by allocating them a lower bitrate, but it demands an accurate viewport prediction. 360° video streaming requires very low latency for an adequate user experience, and if a user abruptly rotates her/his head, it may lead to blank tiles in the user's FoV, which adversely affects the user experience. Furthermore, viewport adaptive streaming techniques require real-time data feed from the user device for viewport prediction, restricting its application. This work proposes a server-side solution that does not require the user's past viewport data and only depends on the saliency map of the video.

The proposed solution is considered to be deployed on Mobile Edge Computing (MEC) architecture, which provides

the computation and storage resources at the network edge, to minimize the network latency. Fig. 2 displays the system architecture of the proposed solution where the MEC server is responsible for fetching videos, threshold selection using KDE, bitrate allocation, and video recreation at the base station. The proposed solution takes the video saliency map as input and identifies the influence of each tile on user-perceived quality using the salient points. Employing a fixed-size tiling, the video frame is spatially divided into multiple tiles, which can be streamed independently by the server. We utilize the Spatial Relationship Description (SRD) feature of MPEG-DASH [13] to enable the tiled 360° video streaming, which spatially divides the video segments into the tiles. The working of the proposed solution is described in Algorithm 1.

Algorithm 1: Saliency Based Bitrate Allocation

Input: $Height,\ Width, saliencyMap,$
$Tile\ Size\ (M \times N), Frame-rate\ (F),$
$Duration\ (T), Quality\ levels\ (Q)$

1 $to\_grayscale\ (saliencyMap_i), \forall i \in T$;
2 $mean \leftarrow \frac{1}{NMT} \sum_{i=0}^{T} \sum_{j=0}^{Height} \sum_{k=0}^{Width} saliencyMap_{i,j,k}$;
3 **for** $i \in [1,T], j \in [1,M], k \in [1,N]$ **do** `/* pixel value normalization */`
4 **if** $saliencyMap_{i,j,k} >= mean;$
5 **then**
6 $saliencyMap_{i,j,k} \leftarrow 1$;
7 **else**
8 $saliencyMap_{i,j,k} \leftarrow 0$;
9 **end**
10 **end**
11 **for** *each frame in saliencyMap* **do** `/* caclulating the frequency of salient points */`
12 Divide frame into $N \times M$ tiles ;
13 **for** $i \in [1,T], j \in [1,M], k \in [1,N]$ **do**
14 $saliencyMatrix_{i,j,k} = count(saliencyMap_{i,j,k})$;
15 **end**
16 **end**
17 $i \leftarrow 1$;
18 **while** $i <= T$ **do** `/* compressing the lookup for each segment */`
19 $finalSaliencyMatrix_{i,j,k} = max_{j \in [i,i+F]}\ saliencyMatrix_{i,p,q}$;
20 $i \leftarrow i + F + 1$;
21 **end**
22 $clusters \leftarrow KDE.fit(flatten(finalSaliencyMatrix))$ `/* training KDE */`;
23 **for** $i \in [1, Q-1]$ **do** `/* bitrate allocation */`
24 $threshold_i \leftarrow minima(cluster_{i+1})$;
25 **end**
26 Allocate bitrate using derived thresholds;

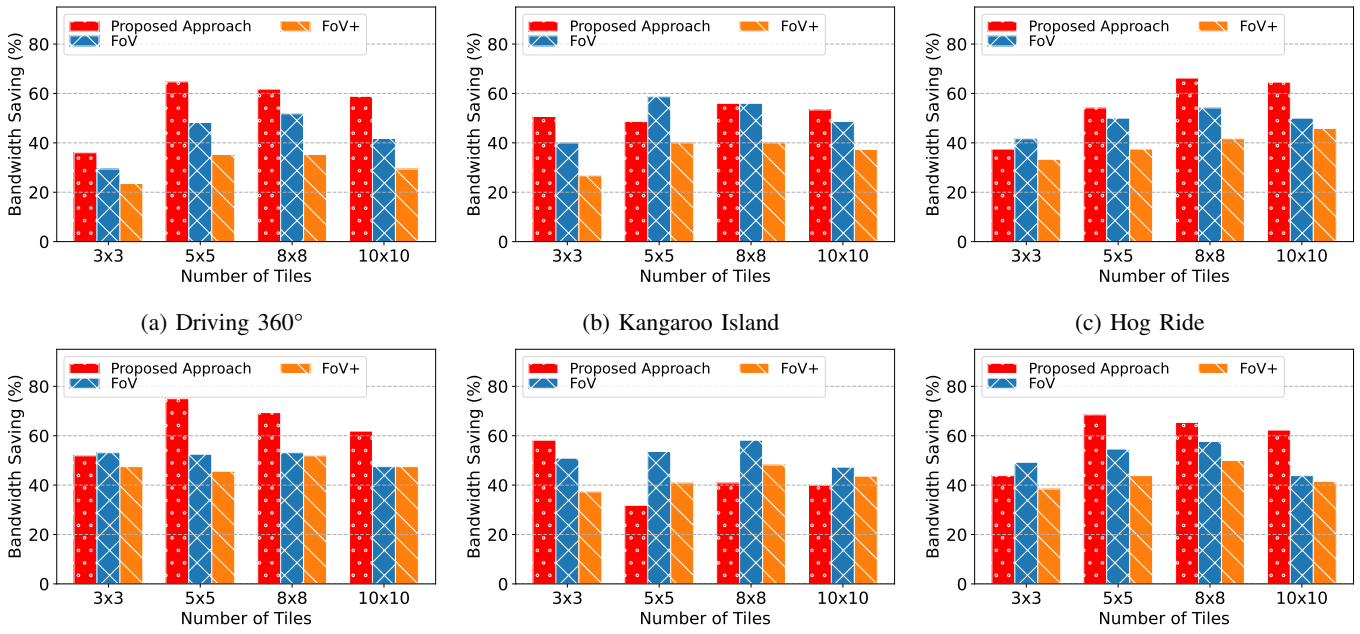
A. Threshold Selection for Bitrate Mapping

The saliency map of the video signals the salient points across the frame which represents the significant features in the scene. The pixel value in the saliency map ranges from 0 to 255 and indicates its importance. The density of such salient points determines the region's importance in the video frame. The wide range of pixel values generate mixed signals in classification; therefore, pre-processing is required before computing the salient point density in different tiles. In $Line-3\ to\ 6$ of Algorithm 1, the saliency map is converted to a binary image using thresholding with a mean pixel value. The number of saliency points is counted for each tile in a video from the pre-processed binary saliency map. In Algorithm 1, $Line-13\ to\ 15$ describes the counting process of salient points from the segmented video tiles. A 3-D array $saliencyMatrix$ of size $[M, N, T \times F]$ is created, where $M \times N$ denotes the total number of tiles, T denotes the duration of the video, and F denotes the video's frame rate. To allocate the bitrates to all the tiles in each 1 $Sec.$ video segment, $saliencyMatrix$ is further compressed into a 3-D array of size $[M, N, T]$. The range of the final saliency counts is very wide and depends on the video resolution. Moreover, saliency counts are neither equally distributed nor unique. Therefore, we use KDE to prioritize the tiles for bitrate allocation by classifying them into different clusters based on salient points.

After counting all the salient points, a one-dimensional data array is created consisting of the saliency counts for all the tiles. Saliency count data contains values ranging from $0\ to\ 150,000$, which also depends on the video resolution. Bitrate allocation requires some definite rules, but the calculated saliency count matrix does not provide any pattern or apparent distribution because of independent values. The proposed solution uses 1-D clustering to classify the data and enabling the bitrate allocation decisions. 1-D KDE is employed in the proposed solution for clustering a list of saliency counts. KDE uses bandwidth as a smoothing parameter, which eventually decides the number of clusters for the given data. Bandwidth is identified empirically to get the required number of clusters from the data in the proposed approach. All the clusters generated by KDE have minima and maxima values in which minima is used as a threshold to classify saliency counts for bitrate allocation as described in $Line-24$ of Algorithm 1.

B. Bitrate Selection

After a lookup for the video bitrate allocation is created that helps in determining the quality of every tile during streaming. The proposed method does not differentiate between the FoV region and the OOS region in bitrate allocation because all the tiles in FoV do not necessarily contain the salient regions. Since, only a few tiles, with some object, building, or vehicle, in the FoV may contain the salient regions. Existing works stream the entire FoV in the highest possible quality without considering the content in individual tiles, resulting in bandwidth wastage. The proposed solution considers the video content and accordingly makes the bitrate selection for each tile, including the FoV tiles. For example, if there are



(a) Driving 360° (b) Kangaroo Island (c) Hog Ride

(d) Roller Coaster (e) SFR (f) Shark Dive

Fig. 3: Bandwidth reduction for different videos in the dataset [4] and effect of change in tile size.

four tiles in the FoV and only one tile contains the salient points then the proposed approach allocates a higher bitrate to this tile while the rest of the tiles are allocated lower bitrates. Moreover, the tiles with salient points outside FoV are also assigned a higher bit rate, ensuring a better QoE on a sudden change in the user's viewport.

IV. Results and Discussion

This section discusses the dataset used for evaluating the proposed method, tools used, and the system configuration of the testbed.

1) Dataset: To evaluate the proposed solution, a 360° video viewing dataset [4] is used. This dataset contains ten different 360° videos from YouTube under three categories; (i) CG, fast-paced,(ii) Natural Image, fast-paced (iii) Natural Image, slow-paced. Initial 60 $Sec.$ of the video is extracted from the original video to use for the playbacks, and each video is in 30 fps amounting to 1800 $frames$ in each playback trace. Salient regions are identified per frame using a Convolutional Neural Network (CNN) and then stitched back to make a 60 $Sec.$ saliency video.

2) Tools: For HEVC encoding and tiling, Kvazzar [14] encoder is used, whereas FFMPEG [15], which provides various options for trimming and calculating the PSNR of the videos, is used for pre-processing the videos and adjusting attributes. The MP4Box tool from GPAC [16], which provides various tools to manage the tiled 360° videos, is used for dashing the encoded videos, tilling, and stitching back the video segments. It also provides mp4client, which can play multi-track 360° videos without aggregating all video tracks. If required, a multi-track video can be converted to a single-track video using MP4Box. A system with Intel®Xeon®Processor E5-2630 v4 processor and 64 GB RAM is used for the priority and bitrate allocation process.

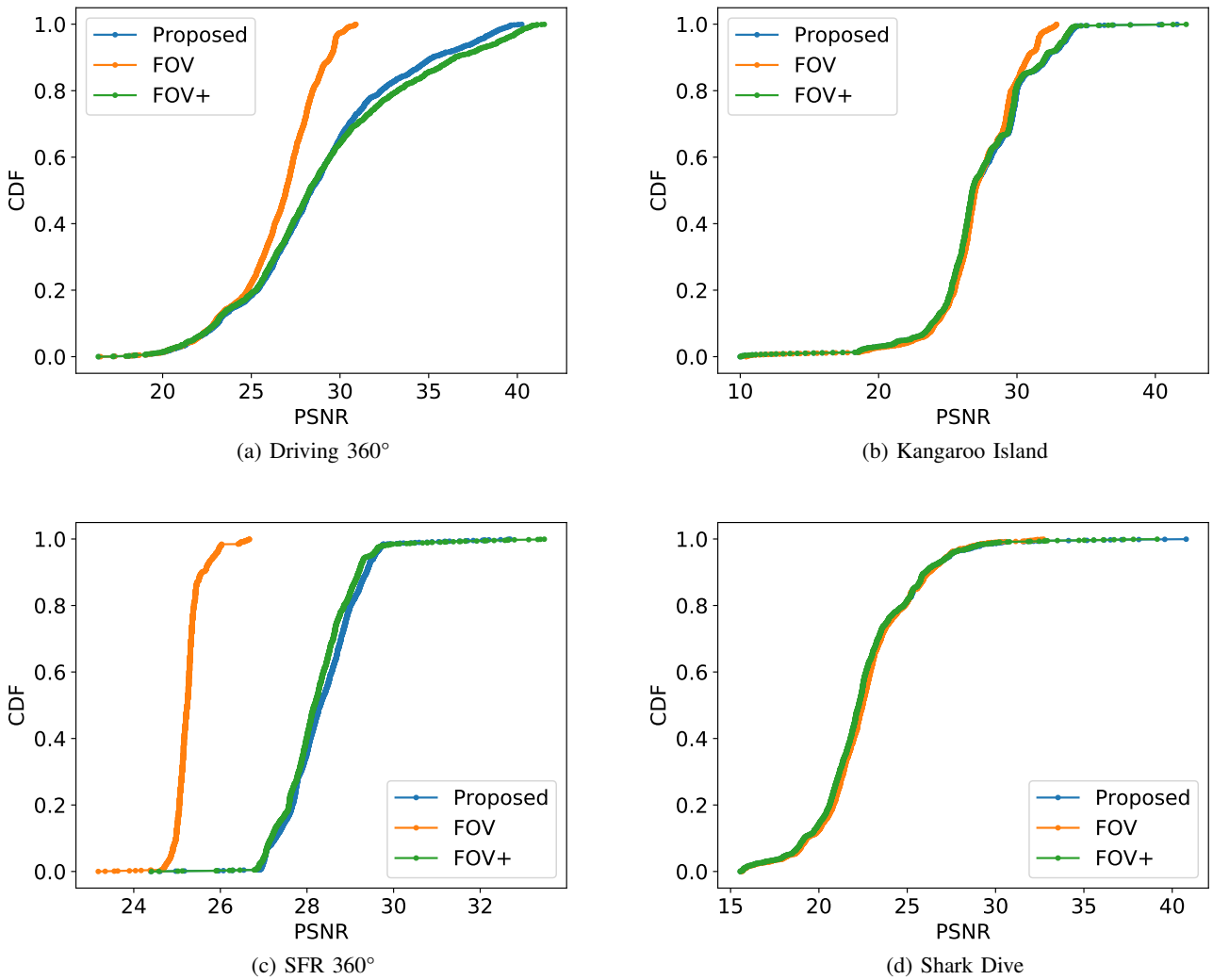
TABLE I: Reduction in video size (MB) using the proposed solutions for different tile sizes.

$Video \backslash Tiles$ Size	3x3	5x5	8x8	10x10	Original
Driving	11 MB	6 MB	6.5 MB	7 MB	17 MB
Kangaroo Island	7.4 MB	7.7 MB	6.6 MB	7 MB	15 MB
Pacman	4.3 MB	4.5 MB	5.4 MB	6.5 MB	10 MB
Hog Rider	15 MB	11 MB	8.1 MB	8.5 MB	24 MB
Roller Coaster	7.7 MB	4 MB	4.9 MB	6.1 MB	16 MB
SFR	4.6 MB	5.3 MB	6.5 MB	6.6 MB	11 MB
Shark Dive	7.3 MB	4.1 MB	4.5 MB	4.9 MB	13 MB

A. Prepossessing

Encoding: Videos are encoded using HEVC with motion constraints enabled. Every encoded video is first temporally divided into one-second segments, and then every segment is spatially divided in $M \times N$ tiles. For each T second video, Q bitrate versions for every tile of T segments are available.

Processing the Saliency Information: As shown in Fig. 1, saliency information of the videos is provided as an MP4 video file with highlighted salient regions. This MP4 file is a single channel video whose pixel values lie in $[0, 255]$. The mean of the smallest pixel and the highest pixel values are used to convert the pixel values to binary using thresholding as described in Algorithm 1. The videos and saliency maps are spatially divided into $M \times N$ tiles. After counting saliency points in all $M \times N$ tiles, a 3-D array of size $[M, N, F \times T]$ is created, where F is the frame rate of the video.



(a) Driving 360°

(b) Kangaroo Island

(c) SFR 360°

(d) Shark Dive

Fig. 4: Objective Quality Assessment (QA) using PSNR of the 360° videos in the dataset [4].

Since the videos are in 30 FPS, each one-second segment contains 30 $frames$, and the saliency map of the frames may differ. Because bitrate is allocated at the segment level, saliency information of frames within a segment must be considered jointly. The principle of max-pooling is applied to combine the saliency counts, where the maximum saliency count of a tile across F frames is taken. Even if there is only one frame for a particular tile in one segment that displays an important object, a high bit rate is allocated to that specific tile. Processing the saliency maps results in a 3-D array of size $[M, N, T]$, considering one-second segments, which contain the saliency count of every tile in each segment. The streaming server uses the computed final saliency count matrix as a look-up table while allocating bitrate to the respective video tiles.

B. Compared Methods

1) FoV: FoV-only [17] is a viewport-based approach that only fetches the FoV tiles. FoV-only [17] conducts the user trial to capture their head movements for different genres of the video and applies linear regression on past viewports to predict the viewport for short periods (e.g., $0.5\ Sec., 1\ Sec., and 2\ Sec.$).

2) FoV+: FoV+ [18] is another viewport-based approach. At time *t*, it predicts the viewport for time $t+T$ using collected movement traces of the user, where T denotes the prediction window. Based on the predicted FoV, it fetches the OOS tiles in the lowest quality to deal with the prediction error.

C. Evaluation Parameters

1) Bandwidth Requirement Reduction: While streaming any media over the wireless network, it is crucial to minimize bandwidth usage while providing the best user experiencing with minimum re-buffering. This parameter helps in analyzing the bandwidth requirement reduction capability of the proposed solution. The reduction in the size of the streamed video content is observed to assess the bandwidth usage.

2) Peak to Signal Noise Ratio (PSNR): PSNR is an objective video quality metric that uses Mean Square Error (MSE) to calculate the noise between two versions of the same video. PSNR is observed to assert that reduction in the bandwidth requirement is not degrading the observed video quality. During the evaluation, PSNR is calculated for all video frames.

D. Results and Analysis

1) Bandwidth Reduction: The proposed approach minimizes the bandwidth usage while maintaining the quality of the video. Data in Table I shows that the proposed solution reduces the bandwidth requirement by 55.06% compared to the source file. The results in Fig. 3 illustrate that the proposed approach provides at least 20% extra size reduction than FoV+ for all the videos in the dataset. Since FoV+ allocates higher bitrates to the FoV region, and the proposed method allocates a lower bitrate to less salient tiles in the whole frame to saves bandwidth. In some cases, FoV streaming-based method outperform the proposed solution where most of the tiles contain a significant amount of salient points and are assigned a higher bitrate.

2) Effect on the Video PSNR: To analyze the video quality, the PSNR value of the streamed video by the proposed solution is compared with the PSNR values of FoV and FoV+ approaches. The proposed solution achieves on average PSNR of $28.5\ dB$ for the complete video. Fig. 4 shows that the achieved PSNR values vary for different videos. In the proposed solution, if only a few tiles of the video segment have the salient point, then assigning the lower bitrate to non-salient tiles results in a lower PSNR value. In contrast, the video segments with uniformly distributed salient points attain a higher PSNR value. The proposed solution consistently outperforms the FoV approach, as FoV doesn't stream the OOS part of the frame. The FoV+ method provides similar PSNR results as compared to the proposed solution because of the transmission of OOS tiles. Although FoV+ provides similar PSNR values, the proposed solution attain significantly reduction in bandwidth.

V. Conclusion

This work introduces a content-aware adaptive bitrate selection solution for 360° video streaming, which doesn't consider the user's viewing pattern. The proposed solution determines the important video tiles using the saliency map which helps to overcome the dependency on the client feedback for the user's viewport. Though the viewport adaptive 360° video streaming solutions successfully reduce the bandwidth requirement, they endure QoE degradation where there is prediction error. The proposed solution overcomes this issue by utilizing the content features for bitrate selection through KDE-based tiles clustering. The results from our extensive experiments show that the proposed solution achieves up to 50% bandwidth reduction as compared to the legacy 360° video streaming. The proposed solution saves 15% more bandwidth while maintaining an average $28.5\ dB$ PSNR as compared to the existing solutions.

Acknowledgement

This work is partially funded by DST under "Autonomous driving enabling fog computing platform with edge cloud orchestration and edge analytics" project.

References

[1] [Online]. Available: https://en.wikipedia.org/wiki/Circle-Vision_360%C2%B0

[2] L. Chopra, S. Chakraborty, A. Mondal, and S. Chakraborty, "Parima: Viewport adaptive 360-degree video streaming," *arXiv preprint arXiv:2103.00981*, 2021.

[3] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '18. New York, NY, USA: Association for Computing Machinery, 2018.

[4] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360° video viewing dataset in head-mounted virtual reality," ser. MMSys'17. New York, NY, USA: Association for Computing Machinery, 2017.

[5] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han, "Rubiks: Practical 360-degree streaming for smartphones," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '18. New York, NY, USA: Association for Computing Machinery, 2018.

[6] Y. Tong, H. Konik, F. Cheikh, and A. Tremeau, "Full reference image quality assessment based on saliency map analysis," *Journal of Imaging Science and Technology*, vol. 54, no. 3, pp. 30 503–1, 2010.

[7] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using mpeg-dash," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16. New York, NY, USA: Association for Computing Machinery, 2016.

[8] T. Stockhammer, "Dynamic adaptive streaming over http –: Standards and design principles," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, ser. MMSys '11. New York, NY, USA: Association for Computing Machinery, 2011.

[9] W.-C. Lo, C.-Y. Huang, and C.-H. Hsu, "Edge-assisted rendering of 360° videos streamed to head-mounted virtual reality," in *2018 IEEE International Symposium on Multimedia (ISM)*, 2018.

[10] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 19–33, 2014.

[11] S. Zhu and Z. Xu, "Spatiotemporal visual saliency guided perceptual high efficiency video coding with neural network," *Neurocomputing*, 2018.

[12] X. Sun, X. Yang, S. Wang, and M. Liu, "Content-aware rate control scheme for hevc based on static and dynamic saliency detection," *Neurocomputing*, 2020.

[13] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim, "Mpeg dash srd: Spatial relationship description," in *Proceedings of the 7th International Conference on Multimedia Systems*, ser. MMSys '16. New York, NY, USA: Association for Computing Machinery, 2016.

[14] [Online]. Available: https://github.com/ultravideo/kvazaar

[15] [Online]. Available: https://www.ffmpeg.org/

[16] [Online]. Available: https://github.com/gpac/gpac

[17] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks." New York, NY, USA: Association for Computing Machinery, 2016.

[18] Y. Bao, T. Zhang, A. Pande, H. Wu, and X. Liu, "Motion-prediction-based multicast for 360-degree video transmissions," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2017.