ESTIMATION OF OPTIMAL ENCODING LADDERS FOR TILED 360° VR VIDEO IN ADAPTIVE STREAMING SYSTEMS

Cagri Ozcinar, Ana De Abreu, Sebastian Knorr, and Aljosa Smolic

Trinity College Dublin (TCD), Dublin 2, Ireland.

ABSTRACT

Given the significant industrial growth of demand for virtual reality (VR), 360° video streaming is one of the most important VR applications that require cost-optimal solutions to achieve widespread proliferation of VR technology. Because of its inherent variability of data-intensive content types and its tiled-based encoding and streaming, 360° video requires new encoding ladders in adaptive streaming systems to achieve cost-optimal and immersive streaming experiences. In this context, this paper targets both the provider's and client's perspectives and introduces a new content-aware encoding ladder estimation method for tiled 360° VR video in adaptive streaming systems. The proposed method first categories a given 360° video using its features of encoding complexity and estimates the visual distortion and resource cost of each bitrate level based on the proposed distortion and resource cost models. An optimal encoding ladder is then formed using the proposed integer linear programming (ILP) algorithm by considering practical constraints. Experimental results of the proposed method are compared with the recommended encoding ladders of professional streaming service providers. Evaluations show that the proposed encoding ladders deliver better results compared to the recommended encoding ladders in terms of objective quality for 360° video, providing optimal encoding ladders using a set of service provider's constraint parameters.

Index Terms— 360° video, virtual reality, adaptive streaming, encoding ladder, optimization

I. INTRODUCTION

Recent years have witnessed a significant industrial investment in virtual reality (VR) technology that has motivated technical developments of graphic cards and head-mounted displays (HMDs) [1]. Currently, the video technology field is evolving toward providing immersive VR experiences using 360° video streaming. 360° video is captured with omnidirectional camera arrays and the individual camera views are projected onto a sphere. For backward-compatibility purposes with the existing video coding standards and streaming pipelines, the spherical videos are mapped onto a planar surface using projection techniques, such as equi-rectangular projection (ERP). ERP videos contain full panoramic 360° horizontal and 180° vertical views of the scene.

Fig. 1: Overview of the different formats and representations.

360° video streaming is significantly challenging owing to its resource-intensive encoding and storage requirements to cope with the very high resolution of its representation. As the VR end-user can only view the field of view (FoV) of the display device (*e.g.,* HMD, smartphone, tablet or laptop), called viewport, very high resolution of 360° video (*e.g.,* 8K×4K ERP) is required for transmission in order to achieve high-quality and seamless video streaming experiences. To reduce both the bitrate consumption of the end-user and the visual distortion of the viewport, 360° video frames can be divided into self-decodable regions [2], [3], namely, tiles.

To deliver the tiled 360° videos to the end-user devices, adaptive streaming systems, such as MPEG-dynamic adaptive streaming over HTTP (DASH) [4], provide smooth 360° video streaming experiences, but still require high encoding and storage costs for the tiled 360° video. The *spatial relationship description* (SRD) [5] can be used with DASH systems where the 360° video stream is divided into tiles. In the SRD, each 360° video is divided into a set of tiles that includes different *bitrate levels* of the tiled video. Different bitrate levels share the same video content but are encoded using various settings, such as the resolution and the target bitrate for encoding. Each different version is called a *representation*, and a set of representations for the video content forms the *encoding ladder* which is requested by the DASH client to play the tiled 360° video. However, encoding and accumulating a large combination of representations for each video content might cover a broad range of network bandwidths such that the end-users can request video streams of appropriate bitrates, and thus it requires high encoding and storage costs [6]. Fig. 1 illustrates the different stages from the spherical projection to the encoding ladder with the different representations of the ERP video.

To tackle this problem, cost-optimal encoding ladders are needed for service providers to deliver tiled 360° video content and satisfy network bandwidths. In fact, tiled 360°

video provides different rate-distortion (RD) performance compared to the traditional video content due to different characteristics of both. In particular, tiling affects the coding efficiency, because redundancy cannot be exploited over tiles. Furthermore, given its 2D projection for encoding (*e.g.,* ERP), each tile of the 360° video has a different level of contribution for the overall 360° video viewing quality due to stretching effects caused by the projection [7], [8]. To this end, new encoding ladder configurations are required for the tiled 360° videos to provide cost-optimal video streaming service for VR end-user devices.

Adaptive streaming systems must deal with issues of the delivery of the tiled 360° video from two different perspectives, the service provider and the client. Most recent work focused on the client's perspective [9]–[14] without considering the service providers' perspective. More clearly, they neither provide 360° video content-specific encoding ladders nor consider the resource costs of the content delivery network (CDN), which is a cloud-based video streaming system that delivers videos to the edge servers so as to effectively connect to the end-users. Given the different characteristic of the tiled 360° video content (*e.g.*, ERP and tile encoding), recommended encoding ladders for traditional videos [15], [16], that are currently used for adaptive streaming systems, might not achieve an acceptable quality of experience (QoE) [6], [17] for the tiled 360° video. Using such encoding ladders might also waste CDN resources and the end-users' bandwidth.

Our work aims to improve the performance of adaptive 360° video streaming systems, providing guidelines for the design of optimal 360° VR video streaming systems using tiles. To this end, we focus on the configuration of cost-optimal encoding ladders in adaptive streaming systems by considering both the provider's and client's perspective and develop an encoding ladder estimation method for tiled 360° video streaming, which is the main contribution of this work. To the best of our knowledge, such encoding ladder estimation method has not been studied yet. The proposed method deals with minimizing the distortion of the observed tiled 8K×4K ERP video content on the client side while reducing the resource costs on the service provider side, such as storage capacity utilization and computational costs for encoding. In this context, we categorize the given 360° videos using their extracted features of encoding complexity, estimate their visual distortion based on the developed distortion model, and calculate the resource costs using the proposed cost models. The cost-optimal encoding ladder configuration problem is then solved using the formulated integer linear programming (ILP) algorithm by considering practical constraints. Our evaluations show that the proposed cost-optimal encoding ladders using a set of service provider's constraint parameters achieve better results compared to the recommended encoding ladders in terms of objective quality for 360°.

The remainder of this paper is organized as follows. Related work is detailed in Section II. Then, the proposed system model is presented in Section III. Experiments to demonstrate the performance of our proposed method are presented in Section IV. Finally, Section V concludes this paper with a summary and future work.

II. RELATED WORKS

To define the most suitable encoding ladder for traditional video, an unique encoding ladder for each given video content is generated for instance by the engineers at Netflix using the brute-force search algorithm [16]. In their research work, each quality-resolution pair was plotted for a given content at each bitrate level. An upper convex hull of its RD curve was then selected to define the encoding ladder. Their approach is very effective concerning QoE for traditional video content. However, it is neither cost-optimal in the sense of resource consumption of a CDN nor content-specific and optimized for tiled 360° videos.

Similarly, academic researchers demonstrated that the previously defined fixed encoding ladders such as Apple's and Netflix's one-size-fits-all schemes [15], [16], have critical weaknesses for traditional video content as described in [18]. Here, the authors defined an optimal encoding ladder for each video category to improve the performance of adaptive streaming for traditional videos. The problem was formulated as an optimization algorithm to find the best bitrate ladder for the given videos by considering the characteristics of a set of end-users in a given database without considering encoding and storage costs. The results have shown, however, that the fixed encoding ladders cannot provide the best objective quality for given traditional videos and clients' bandwidth.

Most recent work focused on 360° video streaming solutions using tiles in order to optimize the quality on the client side [9]–[14]. The authors in [9] proposed a new adaptive streaming system based on tiling, integration of the DASH standard and a viewport-aware bitrate level selection method. In [10], an adaptive bandwidth-efficient 360 VR video streaming system using a divide and conquer approach was presented. The work is based on a dynamic viewport-aware adaptation technique using tiles, derived from a hexaface sphere, and the DASH standard. Similar to the previous work, the authors of [11] also propose a viewport-adaptive video delivery system using tiles (cube maps) and different video representations that differ by their bitrate and different scene regions. Additionally, in [12], high-resolution video content is transmitted in tiled fashion using fixed rectangular tiles. The authors in [13] presented a bandwidth efficient adaptive 360° video streaming system. The work in [14] described the bandwidth problem of 360° video, and suggested to use tile-based streaming. Furthermore, their work described the principles of adaptive streaming of 360° video using tiles and evaluated their system with respect to bitrate

overhead, bandwidth, and quality requirements. However, none of these works are dealing with cost-optimal encoding ladders on the service provider's side to reduce storage capacity utilization and computational costs.

III. PROPOSED SYSTEM MODEL

We consider a cloud-based video-on-demand 360° video streaming pipeline for VR as depicted in Fig. 2. Each captured 360° spherical video is mapped to the ERP representation in 8K×4K resolution for encoding purposes at the source node. The media platform divides each ERP video into N tiles and estimates an unique cost-optimal encoding ladder. Each tile is then encoded at various bitrate levels using multiple encoders with estimated cost-optimal encoding ladder parameters. Then, the generated bitstreams are divided into a set of chunks with equal playback duration, encapsulated by the packaging node and eventually stored on the origin server. Each stored content is then deployed to the CDN, where the bitstreams are efficiently distributed to the VR end-users through the edge servers.

Each end-user device contains the tiled DASH-VR player [9] to communicate with the edge servers and to request individual tiles with appropriate bitrate levels and resolutions from the encoding ladder depending on the bandwidth availability of the network. For adaptive streaming purposes, a set of tiles is encoded at the media platform using different encoding settings. More precisely, let v be an 8K×4K ERP 360° video in the set of videos $\mathcal{V}$. Each v is split into N tiles, each tile j, $j \in \mathcal{T}$, is then encoded at a different bitrate b_j and resolution $r_j = w_j \times h_j$. Hence, the quadruple (v,j,b,r) corresponds to a representation of the video $v \in \mathcal{V}$ for the tile $j \in \mathcal{T}$, encoded at a target bitrate $b \in \mathcal{B}$ and spatial resolution $r \in \mathcal{R}$. Note that v, j, b, and r are integer values and represent the indices of their corresponding sets.

In this context, encoding and accumulating all combinations of the quadruple (v,j,b,r) might be very expensive for service providers. Therefore, a cost-effective optimization is required in order to minimize the service provider's resource costs while providing cost-optimal and high quality 360° video streaming experience.

For this aim, the proposed estimation method contains four main parts: classification of the content type, distortion modeling, cost modeling, and problem formulation. First, we extract spatial and temporal features (f_{spa} and f_{tmp}) of the v-th video to classify its content type as described in subsection III-A. Then, we perform an automatic estimation procedure for the encoding ladder using distortion and cost models for the tiled v-th video as detailed in subsections III-B and III-C, respectively. Again, in this encoding ladder estimation process we consider both the client side (quality distortion) and service provider side (resource costs). Finally, we formulate the cost-optimal estimation problem for the encoding ladder by applying certain practical constraints, which is eventually solved using the proposed ILP algorithm as described in subsection III-D.

III-A. Classification of the content type

To classify the content type from a given set of content types $\mathcal{O}$, spatial f_{spa} and temporal f_{tmp} complexity features are extracted from the videos. As each video v has different RD performances at various resolutions, we can identify two sources of video distortion: spatial down-sampling and quantization. As a down-sampled version of v suffers from spatial information loss, the level of information loss depends on the *spatial complexity* of each video, which is one of the encoding complexity features. Moreover, the high-resolution version of a given v requires a larger amount of bits to reduce its visual distortion. Compared to its low-resolution version, the high-resolution version has a higher sensitivity for unpredictable motions, which requires further residuals to avoid visual distortions. Since predicted residuals are compressed through quantization which results in quality distortions, *temporal complexity* is the second encoding complexity feature. The content type o of each video is then determined from a given $\mathcal{O}$ by classification using the extracted two complexity features.

To extract the feature set $\mathcal{F} = \{f_{spa}, f_{tmp}\}$, we use the constant rate factor (CRF) encoding. The CRF encoding, unlike the constant quantization parameter (QP)-based encoding, has the QPs slightly varied across the time based on the scene complexity, action, and motion. For instance, when a scene contains a lot of action and motion, a higher compression can be applied by raising the QP in order to save bitrates. Therefore, the feature set $\mathcal{F}$ can be extracted from the CRF encoded stream to identify the encoding complexity of each v. For this purpose, the average size of I- and P- frames can be used as main indicators to determine the complexity features. As also demonstrated in [19], the size of I-frames expresses the spatial complexity of each v. Thus, we use the normalized version of the I frame sizes to estimate f_{spa} for a given video. As the average size of P frames characterizes the amount of residual bits, we use the ratio of the size of P frames over the size of I-frames as the indicator for f_{tmp}.

III-B. Distortion modeling

To model the distortion of a given v, we model two sources of artifacts, the compression and spatial scaling artifacts, of the tiled 360° video using its content type and encoding resolution. Both artifacts, which are the most important distortions that deteriorate QoE, are driven by the encoding target rate and the adaptation of the video resolution to the target resolution. With the aim of reducing search complexity, we generate a continuous distortion model for each content type, as the given parameter space is too large for a the brute-force search algorithm (*e.g.,* Netflix's work in [16]). To this end, we derive a distortion function by

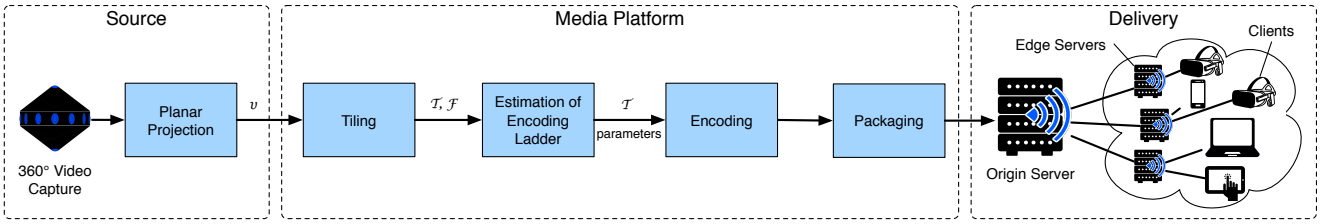

Fig. 2: Schematic diagram of a cloud-based video streaming pipeline for VR which includes source, media platform, and delivery of the tiled 360° video content.

fitting the two-term power series model using the following fit function:

$$FT_{ogB} = k_{og} Z_B^{\Omega_{og}} + \Phi_{og}, \tag{1}$$

where k, Ω, and Φ are fitting parameters used in the curve fitting operation for the o-th content type, $o \in \mathcal{O}$ and $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{O}|}\}$, of the g-the resolution, $g \in \mathcal{G}$ and $\mathcal{G} = \{g_1, g_2, \ldots, g_{|\mathcal{G}|}\}$, at the tiled ERP video bitrate B. Note that Z is the value of the total bitrate of the tiled 360° video in terms of *Mbps* (*i.e.,* total bitrate of the ERP video recomposed of the tiles with bitrate B). These parameters for the proposed distortion model, shown in Table I, were found using the curve fitting operator. Note that index number of o and g are listed in ascending order of their size. The target resolution size is 8K×4K. For the sake of simplicity and also a lack of variety of 8K 360° video content types, we only distinguish between three content types and resolutions. Each row and column number of the fitting parameters in the table represents a different content type and resolution, respectively.

To better reflect the distortion of the 360° video at the clients' side, we estimate the distortion, caused by the mapping of the spherical content onto the planar surface of the devices (*spherical distortion*), of the tiled 360° video as a target value in the curve fitting using the *weighted-to-spherically-uniform mean square error* (WS-MSE) [8]. WS-MSE measures the spherical surface using a non-linear weighting in the MSE calculation. Such weights are calculated using the stretching ratio of the area that is projected from the planar surface to the spherical surface. The noise power for the i-th representation of the j-th tile, d_{ij}, can be formulated as follows:

$$d_{ij} = \frac{\sum_{x \in W} \sum_{y \in H} \left((t_j(x,y) - \tilde{t}_{ij}(x,y))^2 q_j(x,y) \right)}{\sum_{x \in W} \sum_{y \in H} q_j(x,y)}, \tag{2}$$

where $W \times H$ is the resolution of the reconstructed version of the ERP 360° video. Note that x and y denote the pixel coordinates of the ERP video, t and $\tilde{t}$ stand for the original (*i.e.,* uncompressed) and reconstructed versions of the j-th tile and $q_j(x,y)$ represents the weighting intensity in (x, y) of the weight distribution of the ERP for t_j which can be calculated according to [8] with:

$$q_j(x,y) = cos \frac{(y + 0.5 - H/2)\pi}{H}. \tag{3}$$

III-C. Cost modeling

In this subsection, we develop cost models for the cloud-based video streaming system in order to minimize the resource costs for encoding workload and storage capacity utilization at the service providers' side.

III-C1. Encoding cost

The encoding cost is one of the most expensive computing costs which usually occurs on the cloud servers and which heavily depends on the video resolution. To calculate encoding costs, we consider the *broken-line model* where the same cost is defined for similar resolutions. To this end, we extend the cost calculation model used by the Amazon cloud service [20] in order to consider broad range of resolution sizes. The encoding cost c^e can be described for the j-th tile of the i-th representation as follows:

$$c_{ij}^e = \begin{cases} \mu_e, & r_{ij} \leq 720p \\ 2\mu_e, & 720p < r_{ij} \leq 1080p \\ 4\mu_e, & 1080p < r_{ij} \leq 4K \\ 8\mu_e, & 4K < r_{ij} \leq 8K \end{cases} \tag{4}$$

where μ_e is a constant term for the encoding cost defined by the service provider and r_{ij} is the resolution of the j-th tile in the i-th representation.

III-C2. Storage cost

Additionally, large storage capacity is required to store all encoded tiles with different representations for adaptive streaming on the server. The storage cost depends on the data size of the tiled 360° video which is located on the server. Considering a linear cost model where the cost is proportional to the data size of each tiled 360° video stream, the storage cost c^s for the j-th tile of the i-th representation can be described as follows:

$$c_{ij}^s = \mu_s bs_{ij}, \tag{5}$$

where μ_s is a constant term for storage cost defined by the service provider and bs_{ij} is the estimated data size of the

Resolution $\mathcal{G}$		g_1						g_2						g_3					
Model		Distortion			Data size			Distortion			Data size			Distortion			Data size		
		k	Ω	Φ	k	Ω	Φ	k	Ω	Φ	k	Ω	Φ	k	Ω	Φ	k	Ω	Φ
	o_1	1809	-0.6959	5.649	0.7613	0.9901	52.54	4002	-0.7558	2.723	0.8005	0.9859	52.25	1829	-0.5587	-3.266	0.8264	0.9846	214.9
Content type $\mathcal{O}$	o_2	220.1	-0.3583	6.447	0.6467	1.003	29.36	191.9	-0.2763	-5.728	0.6078	1.009	71.15	480.6	-0.3643	-5.728	0.5654	1.015	269
	o_3	820.4	-0.4702	6.2	0.6631	1.001	10.69	643	-0.3825	-2.625	0.6691	1	17.46	616.9	-0.2837	-23.78	0.5943	1.012	203.8

Table I: Curve fitting parameters for the proposed distortion and data size estimation models.

j-th tile in the i-th representation. The data size for each j tile is estimated using the curve fitting technique similar to the one used for Eq. (1). Parameters for the equation, shown in Table I (Data Size), were found using the curve fitting operator.

III-D. Problem formulation

In order to obtain the cost-optimal encoding ladder $\mathcal{L}^*$ for a given video, a set of representations for $\mathcal{L}^*$ is chosen from the set of the estimated representation $\mathcal{L}$ that minimize both the total spherical quality distortion of tiles and the total resource cost of the cloud-based streaming system. For this purpose, we formulate the problem as an optimization problem using the following practical constraints:

(I) **Bandwidth:** In the proposed system, we consider that the encoding ladder needs to cover a set of given network bandwidth profiles $\mathcal{P} = \{p_1, p_2, \ldots, p_{|\mathcal{P}|}\}$ with their minimum B^{min} and maximum B^{max} bandwidth ranges.

(II) **Computational and storage costs:** We set limits for the encoding and storage costs which are the maximum allowed computational cost C^{max} and storage cost S^{max} of the streaming system.

(III) **Encoding rate:** The bitrate levels of the representations should be spaced between each other by the minimum step size τ.

Our objective is to provide a low-quality distortion encoding ladder for a given tiled v at minimum resource costs by considering the above described constrains. Thus, we formulate the optimization problem as follows:

$$\mathcal{L}^* : \underset{\mathcal{L}}{\operatorname{argmin}} \sum_{i \in \mathcal{L}} \sum_{p \in \mathcal{P}} \left(\gamma c_i + (1-\gamma) d_i\right) a_{ip} \tag{6}$$

with

$$c_i = \sum_{j \in \mathcal{T}} (c^e_{ij} + c^s_{ij}) \quad c_i \in \mathcal{P} \tag{7}$$

and

$$d_i = \sum_{j \in \mathcal{T}} d_{ij}, \tag{8}$$

where c_i and d_i are the total resource cost and quality distortion for the i-th representation, respectively. In order to have a trade-off between c_i and d_i, we introduce a predefined constant $\gamma \in [0, 1]$ to be assigned by the service-provider. To cover a wide range of network bandwidths, we introduce a set of network bandwidth profiles in the problem definition. The decision variable $a_{ip} = \{0, 1\}$ indicates if the i-th bitrate level for the p-th profile of a set of network bandwidth profiles $\mathcal{P}$ is included *or* excluded in the encoding ladder for a given v.

Equation (6) minimizes both the overall distortion of the tiled 360° video and resource costs of the cloud-based streaming system and is subject to the following constraints:

$$B^{min}_p \leq b_i a_{ip} \leq B^{max}_p \quad \forall i \in \mathcal{L} \text{ and } \forall p \in \mathcal{P}, \tag{9}$$

$$\sum_{i \in \mathcal{L}} a_{ip} = \lfloor \frac{M \Lambda_p}{\sum_{p \in \mathcal{P}} \Lambda_p} \rfloor \qquad \forall p \in \mathcal{P}, \tag{10}$$

$$\sum_{p \in \mathcal{P}} a_{ip} \leq 1 \qquad \forall i \in \mathcal{L}, \tag{11}$$

$$\sum_{i \in \mathcal{L}} \sum_{p \in \mathcal{P}} s_i a_{ip} \leq S^{max}, \tag{12}$$

$$\sum_{i \in \mathcal{L}} \sum_{p \in \mathcal{P}} c_i a_{ip} \leq C^{max}, \tag{13}$$

$$\frac{b_i a_{ip}}{b^*_n} \geq \tau, \quad \forall i \in \mathcal{L}, \ \forall n \in \mathcal{L}^* \text{ and } \forall p \in P. \tag{14}$$

Equation (9) addresses Constraint (I) for each p. Equation (10) sets the maximum number of representations in the encoding ladder for the p-th profile based on its weighting factor Λ and the total number of representations M in the encoding ladder. The weighting factor Λ for each network profile is shown in Table II. The constraint of Equation (11) avoids the selection of the same representation for each profile. Additionally, Equations (12) and (13) satisfy Constraint (II) by ensuring that encoded videos for estimated encoding ladders cannot exceed S^{max} and C^{max}. Equation (14) satisfies Constraint (III) by ensuring that the target bitrate of each selected representation n in the $\mathcal{L}^*$ is spaced by a minimum step size τ.

IV. EXPERIMENTAL RESULTS

In this section, we investigate the performance of the proposed encoding ladder estimation method by comparison with the one-size-fits-all schemes [15], [16], [21] for the tiled 360° video, and evaluate the proposed method under several service provider's constraints.

IV-A. Setup

We use as the following six 8K×4K resolution 360° ERP video test sequences: $\mathcal{V}$ = {*Train*, *Stitched_left_Dancing360_8K*, *Basketball*, *KiteFlite*, *ChairLift*, *SkateboardInLot*} [22]–[24]. Each $v \in \mathcal{V}$ was split into $N = 10$ tiles which was obtained as an optimal number in our previous research work in [9]. The encoded

bitrate for each tile is equally distributed by dividing the *target bitrate* to the N tiles. Their encoding complexity features and assigned content types are shown in Table III, which was estimated using the described method in the Section III-A. Three content types in the set, $\mathcal{O} = \{o_1, o_2, o_3\}$, were used to classify the videos using the estimated complexity features. The *Train*, *Basketball*, and *ChairLift* sequences were used to model the curve fitting function in Equation (1) and we evaluate our method using the *Stitched_left_Dancing360_8K*, *KiteFlite*, and *SkateboardInLot* video sequences. Further, three different resolutions $\mathcal{G} = \{3072 \times 1536, 4096 \times 2048, 8192 \times 4096\}$ in the encoding ladders and four different bandwidth profiles p were used as defined in Table II with minimum B^{min} and maximum B^{max} bandwidth ranges, and Λ for each bandwidth profile.

Profiles:	p_1	p_2	p_3	p_4
B^{min} (*Mbps*)	1	3	15	25
B^{max} (*Mbps*)	4	20	30	40
Λ	0.25	0.25	0.25	0.25

Table II: Network bandwidth profiles.

We focus on the browser-based video streaming use-case which is one of the core experiments in the ongoing standardization activity [25]. Since AVC is the only implemented decoder in current available browsers which can support HMDs, we apply the H.264/AVC standard in our experiments. In this context, we encoded videos using the FFmpeg software (*ver.* N-85291) [26] with two-pass and 200 percent constrained variable bitrate encoding configurations. At this stage, it is important to mention that our proposed method is video codec agnostic; it can be easily utilized with different video coding standards.

Sequence	f_{spa}	f_{tmp}	$\mathcal{O}$
Train	0.977	0.065	o_1
Stitched_left_Dancing360_8K	0.884	0.110	
Basketball	0.843	0.090	o_2
KiteFlite	0.861	0.090	
ChairLift	0.789	0.212	o_3
SkateboardInLot	0.827	0.521	

Table III: Encoding complexity features and assigned content types for the used test sequences.

To evaluate our proposed method, the objective quality metrics WS-MSE and WS-PSNR [8] were utilized to calculate the quality performance of the 360° video. Further, three different one-size-fits-all encoding ladders (*i.e.,* Apple [15], Axinom [21], and Netflix [16]), which are recommended for traditional videos, were used as references to investigate the quality performance of our proposed method. Table IV shows three reference one-size-fits-all encoding ladders for their three ERP resolutions and four total target encoding rates. In the table, resolutions and target encoding rate were calculated by summation of each tile's resolution and target encoding rate, respectively.

IV-B. Performance evaluation

Encoding ladders for our proposed method have been estimated by solving the formulated ILP algorithm in Section III-D using Pyomo (*ver.* 5.0) [27]. We set μ_e and μ_s to 0.017 and 0.023, respectively. These cost values are same as the real cost values in [20].

To derive the distortion function in Equation (1), we calculated the WS-MSE versus bitrate (in *Mbps*) performance graphs in Fig. 3 for each resolution of the videos *Train*, *Basketball*, and *ChairLift*. The results demonstrate the various performances due to the high diversity in video content characteristics. As can be seen in the figure, each content type has various content dependencies for each encoding resolution and bitrate. For instance, the *Train* sequence (content type o_1), which contains the lowest complex encoding features, achieves a low distortion score compared to content types o_2 and o_3. Because of such diversity, one-size-fits-all schemes, which are used by almost all research works, cannot provide cost-optimal and high-quality streaming performances for the tiled 360° videos.

Evaluation I: To evaluate the RD performance gain of our encoding ladder estimation solution, we compare our proposed method with three different recommended one-size-fits-all schemes of the streaming service providers. As these ladders were estimated without considering constraints, we set $\gamma = 0$ (in order to focus on distortion only) and exclude other constraints in equations between (9) and (13) for a fair comparison in this test.

Figure 4 shows the RD curves computed with average WS-PSNR for the *Stitched_left_Dancing360_8K*, *KiteFlite*, and *SkateboardInLot* sequences. The results show that our proposed method considerably increases the objective video quality (*i.e.,* WS-PSNR) compared to the one-size-fits-all schemes at all times. In particular, the proposed method demonstrates high bitrate savings between 10-30 *Mbps* bandwidth ranges for the content types o_1 and o_2. To this end, we notice that one-size-fits-all schemes provide high scores for the content type o_3 compared to their scores for content types o_1 and o_2.

Evaluation II: We further analyze the performance gain of our method using the Bjøntegaard metric [28] in Table V. This metric describes the distance between two RD curves. In this manner, the bitrate difference, *i.e.* BD-rate, was calculated in percentage averaged over the entire range. A negative BD-rate indicates a decrease of bitrate at the same quality. From the table, we can notice that the proposed method provides considerable bitrate savings compared to the recommended encoding ladders at the same bitrates.

Evaluation III: Finally, in the last set of evaluations, we consider a scenario where the constraints of S^{max} and C^{max} are *8000*, τ = 1.2, and M = 12. In this setup, we

Apple [15]		**Axinom** [21]		**Netflix** [16]	
Z (Mbps)	$W \times H$	Z (Mbps)	$W \times H$	Z (Mbps)	$W \times H$
45	8192 × 4096	45	8192 × 4096	43	8192 × 4096
30	8192 × 4096	30	8192 × 4096	30	4096 × 2048
20	4096 × 2048	21	4096 × 2048	23.5	4096 × 2048
11	3072 × 1536	12	3072 × 1536	17.5	3072 × 1536

Table IV: Recommended one-size-fits-all encoding ladders for traditional videos by service providers.

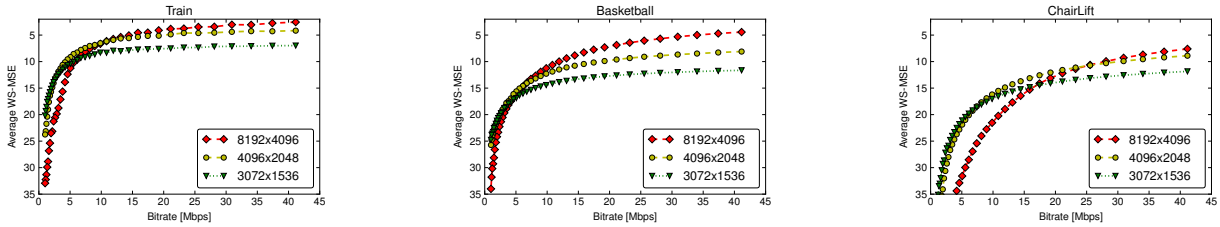

Fig. 3: Average WS-MSE - bitrate curves for sample 8K×4K ERP 360° videos with different content type.

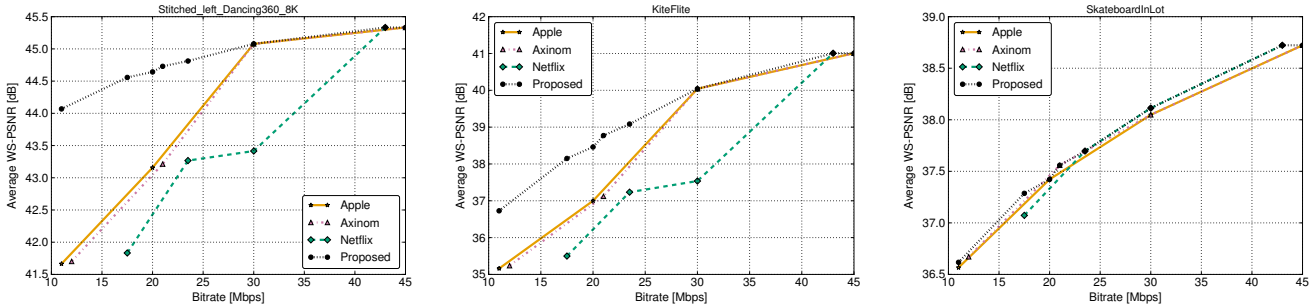

Fig. 4: Performance comparison using the RD curves computed with the average WS-PSNR.

Sequence v	**Streaming vendor**		
	Apple	Axinom	Netflix
Stitched_left_Dancing360_8K	-5.557	-5.885	-69.253
KiteFlite	-13.876	-14.436	-69.178
SkateboardInLot	-1.673	-1.701	-1.155

Table V: BD-rate saving (%) of the proposed method.

use the normalized difference of the total cost ΔCS and the distortion ΔDS (in terms of WS-MSE) in percentages for evaluation purpose. Table VI shows the results of the proposed encoding ladder estimation using resolution-bitrate pairs for $\gamma = 0$, $\gamma = 0.1$, and $\gamma = 0.5$.

From the results, we observe that the lowest complex content, *i.e.,* content type o_1, increases its encoding resolution and decreases its target encoding rate at the range between $i = 2$ and $i = 10$ to reduce the total cost by considering cost and distortion tradeoffs using $\gamma = 0.1$ and $\gamma = 0.5$. On the other hand, we observe that the most complex content, *i.e.* content type o_3, decreases both its encoding resolution and target encoding rate in order to reduce the total cost by considering cost and distortion tradeoffs using the $\gamma = 0.1$ and $\gamma = 0.5$. Table VII reports the total cost saving and distortion gain with respect to different γ. Finally, we would like to mention that, the GNU linear programming kit (GLPK) for Pyomo was able to solve the formulated ILP algorithm in Section III-D using the calculated data in less than one minute on Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz with 32 GB of RAM.

V. CONCLUSIONS

This paper introduced a novel encoding ladder estimation method for tiled 360° video streaming systems, considering both the provider's and client's perspectives. To this end, the objective of our proposed method was to provide cost-optimal and enhanced video streaming experiences for VR end-users. The developed system included classification of the content type, distortion modeling, cost modeling, and problem formulation. The performance of our proposed method was verified in experimental evaluations. The results showed that our method achieved significant bitrate savings (especially for the content types o_1 and o_2) compared to the one-size-fits-all encoding ladders which are recommended by streaming service providers. Furthermore, the developed method can automatically find cost-optimal encoding ladders using several practical constraints, and provides efficient streaming service for tiled 360° video. As future work, we plan to extend our optimization framework by considering the number of tiles for a given content type and investigating

Sequence v	γ	Representation i											
		1	2	3	4	5	6	7	8	9	10	11	12
Stitched_left_Dancing360_8K	0.0	(g_1,1.47)	(g_1,1.78)	(g_1,2.15)	(g_1,3.8)	(g_1,4.6)	(g_1,5.6)	(g_2,10.84)	(g_2,13.11)	(g_2,15.87)	(g_2,28.11)	(g_3,34.01)	(g_3,41.15)
	0.1	(g_2,1.34)	(g_2,1.61)	(g_2,1.95)	(g_2,2.60)	(g_3,3.14)	(g_3,3.80)	(g_3,6.12)	(g_3,7.40)	(g_3,8.96)	(g_3,17.45)	(g_3,21.12)	(g_3,25.55)
	0.5	(g_2,1.00)	(g_2,1.21)	(g_2,1.47)	(g_2,2.36)	(g_3,2.86)	(g_3,3.46)	(g_3,6.12)	(g_3,7.40)	(g_3,8.96)	(g_3,17.45)	(g_3,21.12)	(g_3,25.55)
KiteFlite	0.0	(g_1,1.47)	(g_1,1.78)	(g_2,2.15)	(g_2,3.80)	(g_2,4.60)	(g_3,5.56)	(g_3,10.84)	(g_3,13.11)	(g_3,15.87)	(g_3,28.11)	(g_3,34.01)	(g_3,41.15)
	0.1	(g_1,1.47)	(g_1,1.78)	(g_2,2.15)	(g_2,3.80)	(g_2,4.60)	(g_3,5.56)	(g_3,6.73)	(g_3,8.14)	(g_3,9.85)	(g_3,17.45)	(g_3,21.12)	(g_3,25.55)
	0.5	(g_1,1.00)	(g_1,1.21)	(g_1,1.47)	(g_2,2.36)	(g_2,2.86)	(g_2,3.46)	(g_3,6.12)	(g_3,7.40)	(g_3,8.96)	(g_3,17.45)	(g_3,21.12)	(g_3,25.55)
SkateboardInLot	0.0	(g_1,1.47)	(g_1,1.78)	(g_1,2.15)	(g_1,3.80)	(g_1,4.60)	(g_1,5.56)	(g_2,10.84)	(g_2,13.11)	(g_2,15.87)	(g_2,28.11)	(g_3,34.01)	(g_3,41.15)
	0.1	(g_1,1.47)	(g_1,1.78)	(g_1,2.15)	(g_1,2.86)	(g_1,3.46)	(g_1,4.18)	(g_1,6.12)	(g_1,7.40)	(g_1,8.96)	(g_1,17.45)	(g_2,21.12)	(g_2,25.55)
	0.5	(g_1,1.21)	(g_1,1.47)	(g_1,1.78)	(g_1,2.36)	(g_1,2.86)	(g_1,3.46)	(g_1,6.12)	(g_1,7.40)	(g_1,8.96)	(g_2,17.45)	(g_2,21.12)	(g_2,25.55)

Table VI: Results of the proposed encoding ladder estimation for γ = 0, γ = 0.1, and γ = 0.5.

Sequence v	Δcost (%)		Δdistortion (%)	
	$\gamma = 0.1$	$\gamma = 0.5$	$\gamma = 0.1$	$\gamma = 0.5$
Stitched_left_Dancing360_8K	37.463	39.683	-13.628	-42.914
KiteFlite	33.165	39.206	-9.564	-25.326
SkateboardInLot	37.214	38.884	-8.977	-15.26

Table VII: Total cost saving and distortion gain with respect to γ=0.0.

the effect of total costs by evaluating the effects of the various constraint parameters using a larger set of video sequences.

VI. REFERENCES

[1] Augmented and Virtual Reality Market Report, "Augmented and virtual reality market expected to reach $59,511 million, globally, by 2022," https://www.alliedmarketresearch.com/press-release/augmented-and-virtual-reality-market.html, Apr 2017.

[2] S. Heymann, A. Smolic, K. Mueller, Y. Guo, J. Rurainsky, P. Eisert, and T. Wiegand, "Representation, coding and interactive rendering of high-resolution panoramic images and video using MPEG-4," in *Panoramic Photogrammetry Workshop*, Berlin, Germany, Feb. 2005, pp. 24–25.

[3] C. Grunheit, A. Smolic, and T. Wiegand, "Efficient representation and interactive streaming of high-resolution panoramic views," in *2002 International Conference on Image Processing (ICIP)*, Sept. 2002, vol. 3, pp. III–209–III–212 vol.3.

[4] ISO/IEC 23009-1, "Information technology — dynamic adaptive streaming over HTTP (DASH) — part 1: Media presentation description and segment formats," Tech. Rep., ISO/IEC JTC1/SC29/WG11, 2014.

[5] O. A. Niamut, E. Thomas, L. D'Acunto, C. Concolato, F. Denoual, and S. Y. Lim, "MPEG DASH SRD: Spatial relationship description," in *7th International Conference on Multimedia Systems*. 2016, MMSys '16, pp. 5:1–5:8, ACM.

[6] J. L. Ozer, *Video Encoding by the Numbers: Metric-Based Encoding*, Doceo Publishing, 2016.

[7] A. De Abreu, C. Ozcinar, and A. Smolic, "Look around you: Saliency maps for omnidirectional images in vr applications," in *2017 International Conference on Quality of Multimedia Experience (QoMEX)*, Erfurt, May 2017.

[8] JVET, "AHG8: WS-PSNR for 360 video objective quality evaluation," Tech. Rep. JVET-D0040, JTC1/SC29/WG11, ISO/IEC, Chengdu, CN, Oct. 2016.

[9] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware adaptive 360 video streaming using tiles for virtual reality," in *2017 International Conference on Image Processing (ICIP)*, Sep 2017.

[10] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer!," in *2016 IEEE International Symposium on Multimedia (ISM)*, Sep 2016.

[11] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," *arXiv:cs.MM 1609.08042*, vol. cs.MM, no. 1609.08042, pp. 1–7, May. 2017.

[12] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using MPEG-DASH," in *7th International Conference on Multimedia Systems*, New York, NY, USA, 2016, MMSys '16, pp. 41:1–41:3, ACM.

[13] M. Graf, C. Timmerer, and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, New York, NY, USA, 2017, MMSys'17, pp. 261–271, ACM.

[14] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl, "Tile based HEVC video for head mounted displays," in *IEEE International Symposium on Multimedia (ISM)*, San Jose, CA, USA, Dec 2016, Accessed: 2017-1-16.

[15] Apple Developer, "General authoring requirements," https://developer.apple.com/library/content/technotes/tn2224/_index.html, Sep 2016, Accessed: 2017-6-04.

[16] Netflix Technology Blog, "Per-title encode optimization," http://techblog.netflix.com/2015/12/per-title-encode-optimization.html, Dec 2015, Accessed: 2017-4-27.

[17] M. N. Garcia, F. De Simone, S. Tavakoli, N. Staelens, S. Egger, K. Brunnstrm, and A. Raake, "Quality of experience and http adaptive streaming: A review of subjective studies," in *2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*, Sept 2014, pp. 141–146.

[18] L. Toni, R. Aparicio-Pardo, G. Simon, A. Blanc, and P. Frossard, "Optimal set of video representations in adaptive streaming," in *Proceedings of the 5th ACM Multimedia Systems Conference*, New York, NY, USA, 2014, MMSys '14, pp. 271–282, ACM.

[19] C. Chen, S. Inguva, A. Rankin, and A. Kokaram, "A subjective study for the design of multi-resolution abr video streams with the vp9 codec," *Electronic Imaging*, vol. 2016, no. 2, pp. 1–5, 2016.

[20] Amazon webservices, "Amazon elastic transcoder pricing," https://aws.amazon.com/elastictranscoder/pricing/, Jul 2017, Accessed: 2017-4-27.

[21] S. Saares, "General purpose media format," Tech. Rep. 7, Axinom, Germany, 09 2016.

[22] A. Abbas and B. Adsumilli, "Ahg8: New gopro test sequences for virtual reality video coding," Tech. Rep. JVET-D0026, JTC1/SC29/WG11, ISO/IEC, Chengdu, China, Oct 2016.

[23] E. Asbun, H. He, He. Y., and Y. Ye, "Ahg8: Interdigital test sequences for virtual reality video coding," Tech. Rep. JVET-D0039, JTC1/SC29/WG11, ISO/IEC, Chengdu, China, Oct 2016.

[24] G. Bang, G. Lafruit, and M. Tanimoto, "Description of 360 3D video application exploration experiments on divergent multiview video," Tech. Rep. MPEG2015/ M16129, ISO/IEC JTC1/SC29/WG11, Chengdu, CN, Feb. 2016.

[25] MPEG-DASH, "Descriptions of core experiments on DASH amendment," Tech. Rep. MPEG2016/ N16224, JTC1/SC29/WG, ISO/IEC, Geneva, Switzerland, June 2016.

[26] "VideoLAN," http://www.videolan.org/developers/x264.html, Feb 2017.

[27] W. E. Hart, C. Laird, J. Watson, and D. L. Woodruff, *Pyomo-optimization modeling in python*, vol. 67, Springer, 2012.

[28] G. Bjøtegaard, "Calculation of average PSNR differences between RD-curves (vceg-m33)," Tech. Rep. M16090, VCEG Meeting (ITU-T SG16 Q.6), Austin, Texas, USA,, Apr 2001.