Bitrate Reduction for Omnidirectional Video Streaming: Comparing Variable Quantization Parameter and Variable Resolution Approaches

Giuseppe Ribezzo
Politecnico di Bari
Bari, Italy
giuseppe.ribezzo@poliba.it

Luca De Cicco
Politecnico di Bari
Bari, Italy
luca.decicco@poliba.it

Vittorio Palmisano
Politecnico di Bari
Bari, Italy
vittorio.palmisano@poliba.it

Saverio Mascolo
Politecnico di Bari
Bari, Italy
saverio.mascolo@poliba.it

2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet) | 978-1-6654-3590-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/MEDCOMNET52149.2021.9501278

***Abstract*—Immersive and Extended Reality applications are getting increasingly popular due to the recent improvements and wide availability of Head Mounted Displays in the mass market. Among the many new immersive applications, Omnidirectional Video Streaming (OVS) – or 360° Video Streaming – is attracting the attention of both the industry and the research community. The increased immersivity of 360° videos comes at the cost of larger bandwidth requirements compared to classic 2D videos. To tackle the issue of reducing bandwidth requirements, tiling is a viable technique that allows to encode the portions of the 360° video most likely to fall outside of the users' viewport at lower quality using a higher quantization parameter. Tiling requires new encoders to be used which however do not have available hardware decoders in mobile devices yet. The variable resolution approach instead shrinks areas not falling in the region of interest to decrease the overall resolution and thus allowing bitrate reduction with any codec. This paper quantitatively compares the two approaches to find the trade-offs between achievable bitrate reduction and visual quality measured using the VMAF visual quality metric.**

***Index Terms*—Omnidirectional video streaming, DASH, Tiling, Rescaling, Visual Quality**

I. Introduction

Immersive multimedia for extended reality applications is becoming increasingly popular in many application fields such as, f.i., entertainment, e-learning, e-health, gaming. In this technological context, streaming of Omnidirectional Videos (OV) is rapidly gaining momentum and already counts for leading platforms such as YouTube and Facebook offering such immersive content to their users.

It is well-known that serving OVs requires a much higher network bandwidth compared to classical 2D videos to provide an equal level of visual quality to users. In fact, it has been shown that streaming a 360° video requires a network bandwidth of $\sim$400 Mbps to deliver a video quality similar to that of a full High Definition (HD) resolution 2D video [1]. Consequently, the design of compression techniques suitable for 360° videos is considered an important research topic with interests in both the academia and the video industry.

In OVs the whole 360° scene is captured by using an array of appropriately displaced cameras. A Head Mounted Display (HMD) is employed to render the OV allowing the user to freely explore the recorded environment by simply moving the head, thus augmenting considerably the immersiveness of the experience. At each given point in time, the HMD displays only a portion of the captured scene to the user, the so called *viewport*. Approximately, only 1/6 of the captured scene falls into the viewport [2]. Exploiting this observation, the *viewport-adaptive streaming* technique was proposed. The main idea behind these techniques is to deliver to the user a video having a maximal quality in the regions currently falling into the users' viewport, keeping the other regions with a lower quality (or not delivered at all in the extreme case).

This paper presents a comparison between two approaches for bitrate reduction of 360° video content: 1) the *variable quantization parameter* (VQP) strategy, that exploits the HEVC tiling feature to encode portions of the video which are unlikely to be of interest to the viewer at a lower quality; 2) the *variable resolution* (VRES) approach that shrinks the resolution of the portions outside the region of interest [2]. On one hand, it is expected that the VQP approach should lead to a better bitrate reduction since it exploits a specific feature of the encoder that is aware of how to spatially allocate the *encoding budget* bitrate in the video. On the other hand, the VRES approach has the merit of being extremely simple to be implemented, codec agnostic, and ready to be used in any modern mobile device. In this paper, we systematically assess the performance of these two approaches by quantifying the obtainable bitrate reductions and the corresponding visual quality using the VMAF metric. To the purpose, we employ the Kvazaar encoding open-source encoder and compare the two approaches over a dataset of nine 4K OVs.

The remainder of the paper is structured as follows: Section II provides a background on the existing bitrate reduction approaches used for OV contents; Section III describes the two considered bitrate reduction strategies; Section IV presents the methodology employed to assess the performance of the two approaches; Section V reports the obtained experimental results and, finally, Section VI concludes the paper.

II. Related work

MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH) is today the de-facto standard employed for streaming video contents. MPEG-DASH requires clients to dynamically adapt the video bitrate to the time-varying network bandwidth according to an *Adaptive BitRate* (ABR) control algorithm [3]. In the case of classical 2D streaming, the video content is encoded at different bitrate *levels* (or *representations*) l_i which form the *video levels set* $\mathcal{L} = \{l_1, l_2, \ldots, l_M\}$ ($l_i < l_{i+1}$) [4]. Each video level l_i is logically, or physically, divided into segments of constant duration. Then, the produced video segments are stored on a HTTP server. The client fetches the video segments by employing an HTTP connection: the ABR control algorithm dynamically selects the video level to be streamed at each segment download with the goal of maximizing the users perceived Quality of Experience (QoE) given the available bandwidth.

In the case of OVs, the content generation pipeline differs significantly from the one employed for classical 2D videos. First of all, 360° cameras capture spherical scenes (the omnidirectional scenes) that need to be projected onto the 2D plane (the projected scene) in order to be encoded. However, the part of the video which is currently visualized by the user (the user *viewport*) is roughly one-sixth of the entire projected video resolution. To make a concrete example, in order to deliver a video content with a viewport resolution of 1080p, the projected video resolution has to be larger than 6480p (that is, a video resolution larger than 8K ultra HD). The encoding of such a large resolution video at high quality might result in a too large video bitrate. As a consequence, streaming the encoded video at full resolution entails a remarkable waste of network bandwidth.

A starting point in the OV encoding research field is represented by the Facebook pyramidal projection [5] proposal: the idea was to project the 3D spherical scene onto the different sides of a pyramidal 3D object. The base of the pyramidal 3D-object, presenting less distortion, keeps the portion of the video with the most interesting content. The other video portions are mapped on the sides of the pyramid. Different video portions can be chosen as the base of the pyramid, enabling *viewport-adaptivity*. Then, the pyramid is unfolded and mapped onto a 2D plane. The client is free to choose the most suitable version according to the user viewport. This technique presents several encoding inefficiencies that impact the resulting quality and the achievable bitrate reduction [6]. Other proposals exploiting the idea of assigning a higher pixel density at regions with interesting video contents were the *barrel layout* and the *offset projection* mapping [7]. The barrel layout consists of manipulating the standard *EquiRectangular Projection* (ERP) in such a way to produce a pseudo-cylindrical projection. This is obtained by cropping an area of around 25% from the top and the bottom of the ERP video. The central area is vertically manipulated to increase the pixel density. The top and bottom areas are reprojected to form the top and bottom sides of a cylinder. The *offset projection* modifies the standard *CubeMap Projection* (CMP) by properly adding an offset to the pixels before being projected on the cube face, increasing the pixel density for the front faces. With respect to the pyramidal projection, this approach has proven to be more efficient. In [8], the authors provide a comprehensive analysis of the pseudo-cylindrical projections, pointing out the inefficiencies of their usage with standard video codecs. Based on this analysis, the authors propose two methods that improve the compression performance of both intra-frame and inter-frame coding of pseudo-cylindrical panoramic content. The theoretical basis and some preliminary results for the *offset projection* are exposed in [9].

A performance evaluation of the 3D-to-2D projection methods is provided in [10]. In this work, several of the most commonly used projection functions are tested against different encoder implementations. Performance is measured in resulting quality, output bitrate, and encoding efficiency. The results reveal that ERP grants the best resulting quality/bitrate ratio, while CMP shows better encoding efficiency. It is worth to remark here that the 3D-to-2D projection functions must be applied before encoding, requiring to modify the existing camera hardware and software to be efficient.

A different approach not requiring modifications to the existing encoders – named *divide-and-conquer* – is investigated in [11]. In summary, the idea is to divide the 360° scene in different spatial portions (*slice*). Each *slice* is then encoded independently and packaged separately in a different bitstream. Only the one falling in the current user's field of view (FoV) is delivered to the user. The advantage of this approach is the implementation simplicity, however, the drawback is that a RoI may span multiple slices, requiring one (hardware or software) decoding process *per slice* running on the client device. Moreover, the client has to download in parallel each slice composing the RoI, making the adaptive streaming algorithm considerably more complex. To solve this issue, in [12] the authors take advantage of the HEVC tiling feature to implement a divide-and-conquer approach. The HEVC tiling feature allows to identify different spatial regions in a video and to set encoding parameters specific for that region. The resulting bitstream can be decoded with a single decoder instance at the client-side. Moreover, the authors in [13] and [14] propose a HEVC tile-based 360° streaming framework as an Android application.

In [15], the authors use a multi-scale technique to add viewport-adaptivity to the 360° video and evaluate the proposed approach with respect to the offset projection and the tiling technique. The results show similar quality performances for multi-scale and tiling approaches, outperforming the offset strategy. However, the proposed multi-scale encoding strategy is quite complex and can be hardly used for realtime streaming. In [16] the Region of Interest (RoI) concept is exploited to provide an encoder-agnostic technique for reducing the bitrate requirements of the 360° video. In particular, the goal is reached by properly downsampling the spatial regions outside the identified RoI. More details on this approach are given in the next section.

III. Bitrate Reduction Techniques

In this Section, we briefly describe the two considered bitrate reduction approaches. Figure 1 shows the pipeline used to produce the OV content employed in our performance evaluation which is divided into four parts.

In the *RoI selection* phase (marked with ❶) an algorithm detects a higher interest area spanning 120° horizontally. The algorithm used to select the most interesting areas can be a general *content-aware* algorithm based on saliency map, such as the one described in [17]. This way a number N of views can be produced, each one centered at a specific RoI.

Next, the *Projection* phase (marked with ❷) projects the entire 3D sphere of a view onto a 2D plane using the equirectangular projection format. Referring to Figure 1 notice that each area of the 360° video, i.e. the RoI, the region at the Left (L) and at its Right (R), corresponds to a vertical strip of the same horizontal resolution res_0 in the ERP projection. To clarify, res_0 is the horizontal resolution that is always rendered in the user's viewport.

The *encoding* (marked with ❸) and *decoding* (marked with ❹) phases differ depending on the approach used to reduce the bitrate. In the following, we separately describe the two

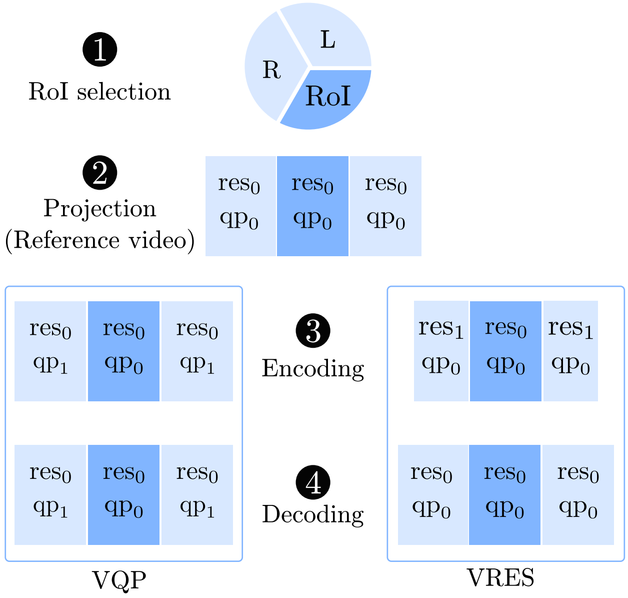


Figure 1: Pipeline used to produce the OV content

considered bitrate reduction approaches and summarize their main advantages and drawbacks.

A. Variable Quantization Parameter (VQP) approach

Let us start by describing the *encoding* phase in the case of the *Variable Quantization Parameter* (VQP) approach, shown in the left branch of Figure 1. In the encoding phase the resolution of the three regions is kept unchanged to res_0. Each region is mapped to a different HEVC tile and, by enabling the *Motion Constrained Tile Set* (MCTS) feature, the decoding process is fully parallelizable. The encoder quantization parameter is set to qp_0 in the RoI region, whereas the regions outside the RoI (L and R) are encoded at a higher quantization parameter equal to $\mathrm{qp}_1 = \mathrm{qp}_0 + \Delta\mathrm{qp}$, which decreases the encoding budget used for those regions. In the decoding phase, no particular operation is needed to be performed in the case of the VQP approach: decoding is performed in parallel by a single HEVC decoding instance for all the three downloaded tiles.

This approach allows performing server-side storage optimization techniques, f.i. by enabling packaging of different tiles being performed on-the-fly on demand as proposed in [17]. Nevertheless, a drawback of this approach is that it is strictly dependent on the HEVC codec which requires specific hardware support for decoding that is not widely available in the mobile market at the moment.

B. Variable Resolution (VRES) approach

The *Variable RESolution* (VRES) approach is shown in the right branch of Figure 1. In this case, the encoding phase requires that the two regions outside the RoI are shrunk horizontally from a resolution res_0 to a lower resolution res_1. Next, the resulting rescaled video is encoded at a quantization parameter equal to qp_0 applied to all the ERP video. After the video is decoded at the client, the two regions outside the RoI are upscaled from res_1 back to the original horizontal resolution res_0.

With respect to the VQP approach, this technique has interesting features: 1) it is independent of the employed codec, 2) it can be efficiently handled by hardware decoders at the client-side, 3) it can use well-established and mature algorithms (interpolation, filtering, etc.) to improve the resulting video quality. Nevertheless, server storage consumption can be high if RoI selection phase is not appropriately tuned.

Video	Youtube ID
Boomerang	r-qmDDi8S5I
FighterJet	NdZ02-Qenso
UniversalStudiosFlorida	Js_Jv5EzOv0
Tahiti360	7gjR60TSn8Q
KITZ360	KS9S1Hgx2co
WhiteLions360	14O7AxqjiVY
WildDolphins	BbT_e8lWWdo
GirlGroup360	NxIRVul10CA
MaldivesVR360	MgJITGvVfR0

Table I: The video catalog

IV. Methodology

Table I shows the video catalog fetched from YouTube and used for the performance comparison. All the considered videos have a resolution of 3840×1920 and a framerate of 30 fps which means that $\mathrm{res}_0 = 1280\,\mathrm{px}$. To consider the settings commonly used for online streaming, the *Group of Pictures (GoP)* parameter was fixed to 150, which means that a key-frame is generated every 5 seconds. The visual quality assessment between the manipulated video and the reference one has been obtained by using the visual quality metric *Video Multi-Method Assessment Fusion* (VMAF) [18] which has proven to be effective for 360° videos [19].[1] The visual quality assessment for each video has been carried out as described in the following. For each video in the catalog that is assumed as the reference video, a manipulated copy has been produced according to the considered bitrate reduction strategy, namely VQP and VRES. Both the manipulated and the reference video have been segmented at the GoP boundaries, producing a chunk set with chunks duration equal to 5 seconds. It is worth noting that the GoP structure has been applied is such a way to produce chunks time-aligned between the reference and the manipulated video. An area, centered at a configurable *yaw angle* α and 120° wide horizontally, has been cropped for each chunk from both the manipulated and the reference video chunk set. The two cropped areas have been compared using the VMAF visual quality metric to produce a score. The yaw angle α has been set to vary in the set $\{-120, -100, -90, ..., 90, 100, 120\}$ to cover the entire 360° field of view. It is important to notice that $\alpha = 0$ corresponds to the case in which only the RoI (that is never degraded) is framed in the viewport. The extreme case where the user frames in the viewport only degraded content corresponds to either $\alpha = -120°$ or $\alpha = 120°$. To clarify, these extreme cases represent situations in which the viewport is framing the region to the left (right) of the RoI that is marked with an L (R) in Figure 1.

The VRES and VQP approaches have been tested leveraging the tiling feature implemented by the *kvazaar* encoder [20]. The *kvazaar* encoder allows to set the grid

[1] Notice that we also have computed SSIM scores which however prove less expressive compared to the ones obtained using VMAF and therefore are not discussed in this paper.

to be used to divide the video in tiles. To comply with the rationale used in [2], a 3-column grid as been applied each having horizontal resolution equal to $\mathrm{res}_0 = 1280\,\mathrm{px}$. The `--mv-constraint frametilemargin` option ensures that the encoder operations are fully parallelizable for each tile, by managing the HEVC MTCS feature. Furthermore, in the case of the VQP approach, the encoder allows to specify the variation of the quantization parameter ($\Delta\mathrm{qp}$) to be applied to each tile with respect to a baseline quantization parameter qp_0. In the experiments, the $\Delta\mathrm{qp}$ varies in the set $\{5, 10, 15, 20\}$ for VQP. Notice that the lower $\Delta\mathrm{qp}$ the lower is the expected bitrate reduction.

The VRES approach implements the bitrate reduction strategy as described in [2]. Again, to provide a fair performance evaluation, the same encoder, i.e., *kvazaar* is used to encode the same video catalog. In this case, the `--mv-constraint frametilemargin` option has been left unset. The VRES approach has been tested with four different downscaled resolutions res_1, namely 1080p, 720p, 480p, 240p.

As already mentioned above, videos encoded with the VRES approach need to upscale the encoded video to the original resolution in order to be decoded and played back by the user. Such an operation is performed through an interpolator filter. For this purpose, in this work we have employed the bicubic interpolator made available by the FFMPEG suite.

To investigate the relationship between the obtainable bitrate reduction and the resulting video quality, the encoder has been set in *constant quality* (CQ) mode. When configured in this mode, the encoder is free to vary the output bitrate to reach the set output video quality. In this work, the $--\mathrm{qp}$ parameter has been chosen to output a visually lossless video quality. Moreover, it is worth to remark here that we are interested in bitrate reduction capability of the algorithms, not on absolute output bitrate. As reported in [2], the $--\mathrm{qp}$ value has been set equal to 22, i.e., for VRES the whole video is compressed with $\mathrm{qp}_0 = 22$. In the case of VQP the RoI is encoded at $\mathrm{qp}_0 = 22$, whereas the regions falling outside of the RoI are encoded with a quantization parameter equal to $\mathrm{qp}_0 + \Delta\mathrm{qp}$.

Finally, the obtained dataset comprises around 64,000 VMAF scores obtained by analyzing a total of around 88 hours of video content. Also notice that the entire duration of the videos has been analyzed.

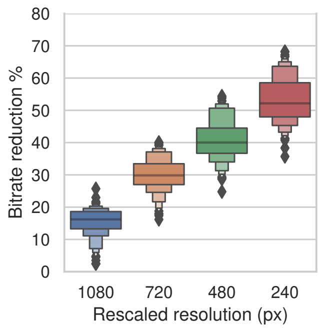
In Table II we summarize the toolchain and parameters used to carry out the performance evaluation.

V. Results

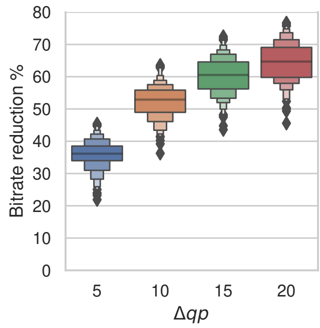
This section presents the obtained results and it is organized as follows. We first show the impact of the parameters used by the two approaches on the obtained bitrate reduction (Section V-A). Then, we compare the overall visual quality obtained by each of the considered approaches as a function of the position of the users' head (Section V-B). We next delve into investigating how the video content impacts the differences between the visual quality obtained by the VRES and VQP approaches V-C.

Table II: The settings employed to carry out the performance evaluation

Encoder	kvazaar v1.3.0
Multimedia tool	FFMPEG v4.2.1
Frame rate	30 fps
GoP	150 frames
Segment Duration	5 s
Quantization parameter qp_0	22
Parameters for the areas outside the ROI	**VQP**: $\Delta\mathrm{qp} \in \{5, 10, 15, 20\}$
	VRES: $\mathrm{res}_1 \in \{1080p, 720p, 480p, 240p\}$ with Bicubic interpolation for upscaling



(a) VRES



(b) VQP

Figure 2: Bitrate reduction (%)

A. Bitrate reduction

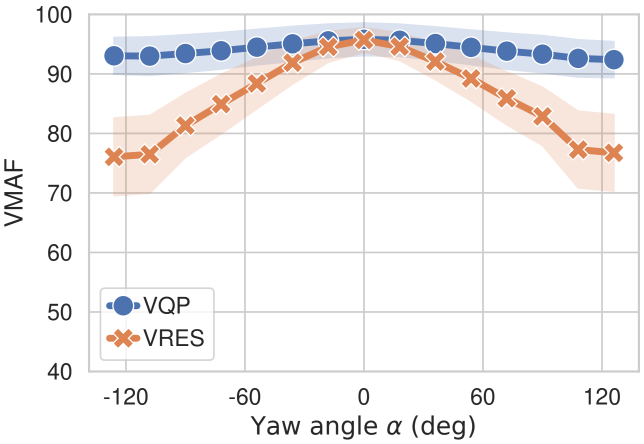
We start our investigation by considering the efficiency in terms of bitrate reduction of VRES and VQP schemes as a function of their respective parameters. In particular, for VRES the rescaled resolution res_1 varies in {1080p, 720p, 480p, 240p}, whereas in the case of VQP $\Delta\mathrm{qp}$ varies in {5, 10, 15, 20}.

Figure 2a and Figure 2b compare the overall percentage of bitrate reduction which can be obtained by both VRES and VQP considering the whole video catalog. The results are shown in a box plot which captures the variability of the results with respect to different videos and segment in the video.

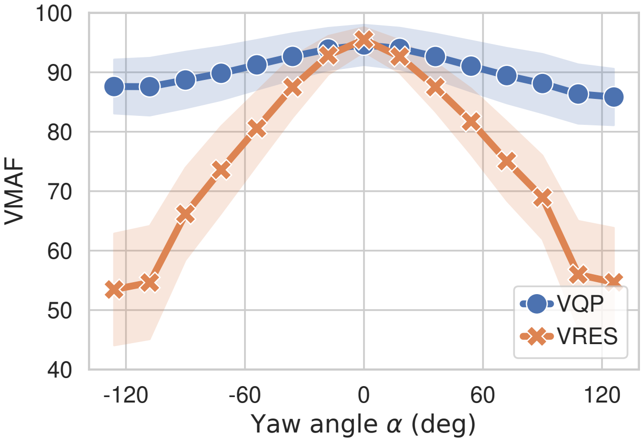
Figure 2a shows that, in the VRES case, as the rescaled resolution decreases from 1080p to 240p, the obtained bitrate reduction increases from a median value of around 15% up to 52% quite linearly. Regarding the VQP approach, Figure 2b shows that the impact of $\Delta\mathrm{qp}$ on bitrate reduction is more pronounced as this parameter increases. In particular, $\Delta\mathrm{qp} = 5$ already provides a median bitrate reduction of around 36%, and rapidly increases to 52% for $\Delta\mathrm{qp} = 10$ which is exactly equal to the maximum bitrate reduction obtained in the case of the VRES approach when the rescaled resolution is set to 240p. Also, comparable median bitrate reductions are obtained for $\Delta\mathrm{qp} = 5$ (corresponding to ~36%) and for rescaled resolution 480p (corresponding to 40%).

In the next sections, we shall employ the established couples of parameters that provide similar bitrate reductions,

[2]https://github.com/ultravideo/kvazaar

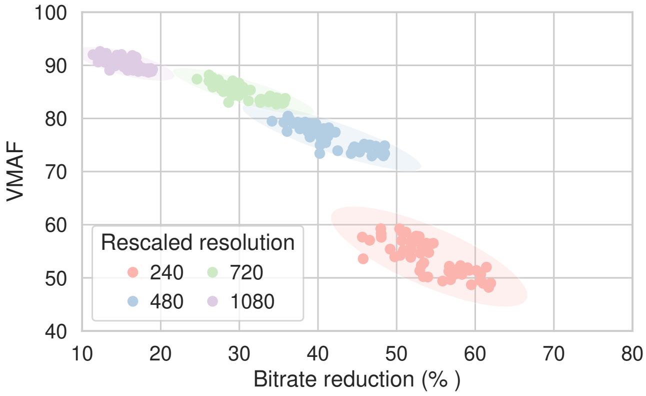


(a) scaled resolution 480p, $\Delta\text{qp} = 5$

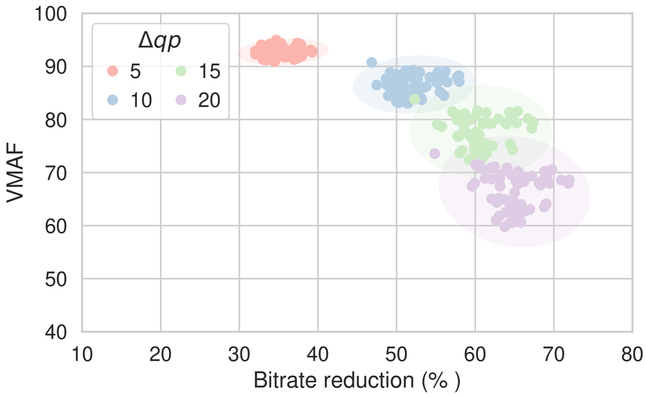


(b) scaled resolution 240p, $\Delta\text{qp} = 10$

Figure 3: VMAF as a function of the user's head yaw angle α



(a) VRES



(b) VQP

Figure 4: Worst case Visual Quality vs Bitrate reduction trade-off

i.e. $(5, 480\text{p})$ and $(10, 240\text{p})$, to compare the corresponding visual quality obtained.

B. *Visual quality as a function of the user's head position*

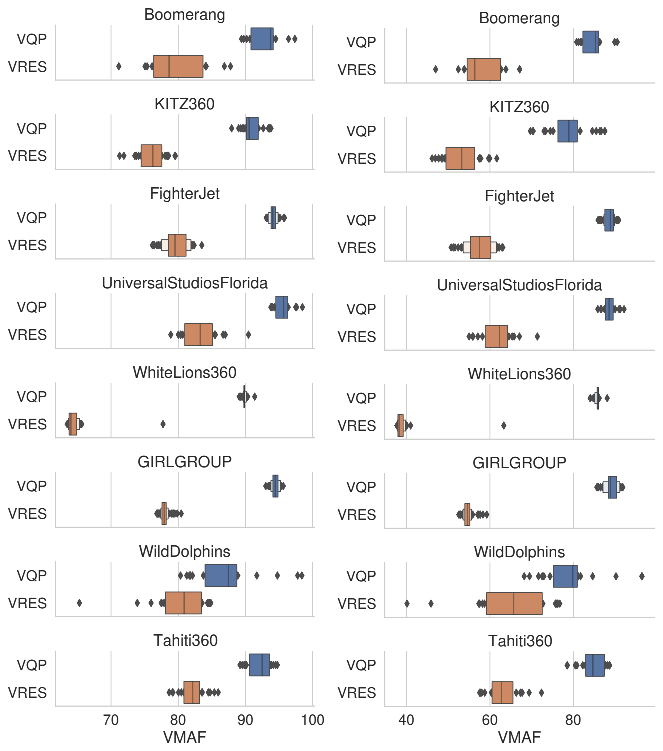
We are now interested in comparing the visual quality obtained by VRES and VQP when they offer comparable bitrate reductions. To the purpose, for each video we collect the VMAF score measured when the users' head is positioned at a certain yaw angle α with respect to the center of the RoI. Recall that, $\alpha = 0$ corresponds to the case in which the viewport only frames the 120°-wide area that is non distorted. As α moves away from the RoI, larger and larger degraded portions of the video will fall in the users' viewport and the visual quality is expected to decrease. In the case of VQP, the degradation is due to the higher quantization parameter used to encode the content outside the RoI, in the VRES case, the degradation is due to the downscaling and upscaling operations described in Section III and IV.

Figure 3a and Figure 3b compare the median visual quality and the standard deviation (shaded areas) measured using the VMAF score as a function of the yaw angle α. Let us start by considering Figure 3a which corresponds to the case in which VQP employs a $\Delta\text{qp} = 5$ to encode the regions outside the RoI and VRES downscales the horizontal resolution of the regions outside the RoI to 480p. In Section V-A, we have shown that those parameters provide a comparable median bitrate reduction of around 40%. Figure 3a shows that, as expected, as $|\alpha|$ increases larger and larger portions of degraded videos fall in the viewport and the measured VMAF decreases. Nevertheless, in the case of the VQP approach the quality degrades negligibly, whereas in the VRES case the VMAF drops from ~95 ($\alpha = 0$) to ~76 ($\alpha = 120°$)[3].

Figure 3b compares the case of VQP set with a $\Delta\text{qp} = 10$ and VRES set with a downscale resolution equal to 240p which corresponds to a median bitrate reduction of around 52% for both the approaches. The figure confirms that VQP is able to provide a graceful degradation of the visual quality obtaining a worst case VMAF score equal to ~85, whereas VRES achieves a worst case measured VMAF as low as ~52. This means that in the VRES case if users point their head to a region framing only distorted content the obtained visual quality is between "poor" and "fair" [18].

To complete this analysis, Figure 4a and Figure 4b show the worst case VMAF bitrate reduction trade-off achieved respectively by VRES and VQP obtained when $\alpha = 120°$. Each data point of the scatter plot represents one video chunk of a given video encoded with a specific parameter (differentiated by its color). The interesting insight that can be gathered from Figure 4b is that, in the case of VQP, increasing the Δqp parameter from 15 to 20 increases the bitrate reduction negligibly (as pointed out in Section V-A) at the price of a drastic decrease of the worst case visual quality from a median value of ~80 to ~65.

[3]According to VMAF authors a score equal to 70 can be mapped to a vote between "good" and "fair" [18].



(a) scaled resolution 480p, $\Delta\text{qp} = 5$ (b) scaled resolution 240p, $\Delta\text{qp} = 10$

Figure 5: Worst case Visual Quality vs Bitrate reduction trade-off for each video

In summary, VRES visual quality decreases faster when the user moves his head away from the RoI, whereas VQP gracefully degrades the visual quality. For VQP using a Δqp greater than 15 is not advisable.

C. *Visual quality as a function of video content*

In Section V-B, we have found that, in the worst case, the median difference between the VMAF score of VQP and VRES is equal to ~16 (~30) when the bandwidth reduction percentage is ~40% (52%) (see Figure 3a and Figure 3b). In this section, we are interested in investigating the sensitivity of the two bitrate reduction strategies to different video content. To the purpose, Figure 5a and Figure 5b compare the VMAF scores for each content of the video catalog (see Table I) in the worst case when the yaw angle is equal to 120°.

The figures show that in 7 out of 9 videos the VMAF scores do not differ significantly from the median value we have found in Section V-B. Nevertheless, the video *WhiteLions360* shows a remarkably lower VMAF score in the case the VRES strategy is used, whereas in the case of *WhiteDolphins* the VMAF scores are much closer with respect to the median case.

Figure 6 shows one frame extracted from the *WhiteLions360* at a yaw angle such that the left half of the frame belongs to the RoI, whereas the right half of the frame belongs to the distorted area outside of the RoI. The parameters employed for VRES and VQP lead to a 52% bandwidth reduction. By comparing the two frames it can be noticed that: i) in the VRES case the gaussian blur effect is clearly visible on the lion; this is due to the lossy process of downscaling the region outside the RoI from 1280p to 240p and then re-upscaling the video to the original resolution; ii) in the VQP case the frame is sharp also in the region where the higher quantization parameter is used (compare the field texture and the leaves of the tree); nevertheless, some artifacts affect the frame in the degraded region which are clearly visible on the lion's face and mane.

Figure 7 shows the same frame extracted by the *WildDolphins* video which shows that, due to of the lack of details of the texture of the sea in the original video, the blur effect peculiar of the VRES approach (Figure 7b) is not as evident as in the *WhiteLions360* video; this, together with some block boundary artifacts in the VQP compressed video (Figure 7a), results in a less pronounced VMAF difference as shown in Figure 5.

VI. Conclusions

In this work, we have compared two bitrate reduction schemes for omnidirectional video content, namely the variable resolution (VRES) approach and the Variable Quantization Parameter (VQP) approach. For this purpose, we have used the VMAF metric to quantify the visual quality and we have measured the obtained bitrate reduction percentage by applying both VRES and VQP approaches to a catalog of nine benchmark 4K resolution omnidirectional videos. The obtained results have shown that comparing the two approaches at equal bitrate reduction percentage the VMAF score obtained by VRES is consistently lower than that of VQP. When the two approaches achieve a bitrate reduction percentage equal to 52%, the VQP obtains a VMAF scores higher up to 30 points compared to VRES. Nevertheless, at lower bitrate reductions, when the rescaled resolution is 480p, the VRES approach does not pay a remarkable quality loss and becomes a viable solution due to its implementation simplicity and due to the fact that it can be employed with any codec.

Acknowledgments

This work has been partially supported by the Italian Ministry of Economic Development (MISE) through the CLIPS project (no. F/050136/01/X32) and by the Italian Ministry of Education, Universities and Research (MIUR) through the MAIA project (no. ARS01_00353). Any opinions, findings, conclusions or recommendations expressed in this work are the authors and do not necessarily reflect the views of the funding agency.

References

[1] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, "Vr is on the edge: How to deliver 360 videos in mobile networks," in *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. ACM, 2017, pp. 30–35.

[2] G. Ribezzo, G. Samela, V. Palmisano, L. De Cicco, and S. Mascolo, "A dash video streaming system for immersive contents," in *Proc. ACM MMSys '18*, Amsterdam, June 2018, pp. 525–528.

[3] I. ISO, "23009–1: 2014: Information technology-dynamic adaptive streaming over http (dash)."

[4] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH)," in *2013 20th International Packet Video Workshop*, Dec 2013, pp. 1–8.

[5] Facebook developers, "Next-generation video encoding techniques for 360 video and vr," https://code.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/, Jan 2016, online; accessed 22-May-2019.

[6] K. K. Sreedhar, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications," in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 583–586.

(a) VQP, $\Delta\text{qp} = 10$

(b) VRES, rescaled resolution 240p

Figure 6: Frame extracted from the WhiteLions360 video

(a) VQP, $\Delta\text{qp} = 10$

(b) VRES, rescaled resolution 240p

Figure 7: Frame extracted from the WildDolphin video

[7] Facebook developers, "Enhancing high-resolution 360 streaming with view prediction," https://code.fb.com/virtual-reality/enhancing-high-resolution-360-streaming-with-view-prediction/, Apr 2017, online; accessed 22-May-2019.

[8] R. G. Youvalari, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Efficient coding of 360-degree pseudo-cylindrical panoramic video for virtual reality applications," in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 525–528.

[9] C. Zhou, Z. Li, and Y. Liu, "A measurement study of oculus 360 degree video streaming," in *Proc. ACM MMSys '17*, 2017, pp. 27–37.

[10] M. Jamali, F. Golaghazadeh, S. Coulombe, A. Vakili, and C. Vazquez, "Comparison of 3d 360-degree video compression performance using different projections," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. IEEE, 2019, pp. 1–6.

[11] M. Hosseini and V. Swaminathan, "Adaptive 360 vr video streaming: Divide and conquer," in *2016 IEEE International Symposium on Multimedia (ISM)*, Dec 2016, pp. 107–110.

[12] C. Concolato, J. Le Feuvre, F. Denoual, F. Maze, E. Nassor, N. Ouedraogo, and J. Taquet, "Adaptive streaming of hevc tiled videos using mpeg-dash," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1981–1992, 2018.

[13] J. He, M. A. Qureshi, L. Qiu, J. Li, F. Li, and L. Han, "Rubiks: Practical 360-degree streaming for smartphones," in *Proc. ACM MobiSys '18*, 2018, pp. 482–494.

[14] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proc. ACM MobiCom '18*, 2018, pp. 99–114.

[15] H. Hristova, X. Corbillon, G. Simon, V. Swaminathan, and A. Devlic, "Heterogeneous spatial quality for omnidirectional video," in *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, Aug 2018, pp. 1–6.

[16] L. De Cicco, S. Mascolo, V. Palmisano, and G. Ribezzo, "Reducing the network bandwidth requirements for 360 immersive video streaming," *Internet Technology Letters*, vol. 2, no. 4, p. e118, 2019.

[17] S. Rossi, C. Ozcinar, A. Smolic, and L. Toni, "Do users behave similarly in vr? investigation of the user influence on the system design," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 2, pp. 1–26, 2020.

[18] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," *The Netflix Tech Blog*, vol. 6, 2016.

[19] M. Orduna, C. Diaz, L. Munoz, P. Perez, I. Benito, and N. Garcia, "Video multimethod assessment fusion (vmaf) on 360vr contents," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 1, pp. 22–31, 2020.

[20] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. D. Hämäläinen, "Kvazaar: Open-source hevc/h.265 encoder," in *Proc. ACM MultiMedia '16*, 2016.