Optimal Set of 360-Degree Videos for Viewport-Adaptive Streaming
Xavier Corbillon, Alisa Devlic, Gwendal Simon, Jacob Chakareski

To cite this version:

HAL Id: hal-01767558
https://hal.archives-ouvertes.fr/hal-01767558
Submitted on 16 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Optimal Set of 360-Degree Videos for Viewport-Adaptive Streaming

Xavier Corbillon
IMT Atlantique & IRISA, France
xavier.corbillon@imt-atlantique.fr

Gwendal Simon
IMT Atlantique & IRISA, France
gwendal.simon@imt-atlantique.fr

ABSTRACT
With the decreasing price of Head-Mounted Displays (HMDs), 360-degree videos are becoming popular. The streaming of such videos through the Internet with state of the art streaming architectures requires, to provide high immersion feeling, much more bandwidth than the median user’s access bandwidth. To decrease the need for bandwidth consumption while providing high immersion to users, scientists and specialists proposed to prepare and encode 360-degree videos into quality-variable video versions and to implement viewport-adaptive streaming. Quality-variable versions are different versions of the same video with non-uniformly spread quality: there exists some so-called Quality Emphasized Regions (QERs). With viewport-adaptive streaming the client, based on head movement prediction, downloads the video version with the high quality region closer to where the user will watch. In this paper we propose a generic theoretical model to find out the optimal set of quality-variable video versions based on traces of head positions of users watching a 360-degree video. We propose extensions to adapt the model to popular quality-variable version implementations such as tiling and offset projection. We then solve a simplified version of the model with two quality levels and restricted shapes for the QERs. With this simplified model, we show that an optimal set of four quality-variable video versions prepared by a streaming server, together with a perfect head movement prediction, allow for 45% bandwidth savings to display video with the same average quality as state of the art solutions or allows an increase of 102% of the displayed quality for the same bandwidth budget.

KEYWORDS
360-degree Video, Omnidirectional Video, Quality Emphasized Region, Viewport Adaptive Streaming

1 INTRODUCTION
Offering high-quality virtual reality immersion by streaming 360-degree videos on the Internet is a challenge. The main problem is that most of the video signal information that is delivered is not displayed. Indeed, the Head-Mounted Displays (HMDs) that are used for immersion show a viewport, which represents a small fraction of the whole 360-degree video. Typically, to extract a 4K (3840 × 2160 pixels) video viewport from the whole 360-degree video, the stream should be at least a 12K (11520 × 6480 pixels) video, from which most information is ignored by the video player.

A solution researchers are exploring to limit the waste of bandwidth is to prepare and stream 360-degree videos such that their quality is not homogeneous spatially [6, 11, 19, 23]. Instead the quality is better at the expected viewport positions than in the rest of the video frame. Two main concepts that support this solution are (i) encoding of quality-variable videos, which can be based on tiling [28], scalable coding [4, 26], and offset projections [30]; and (ii) implementation of viewport-adaptive streaming, which is to signal the different quality-variable versions of the video, to predict viewport movements, and to make sure that a given user downloads the quality-variable video such that the quality is maximum at her viewport position.

The design of efficient viewport-adaptive streaming systems requires the understanding of the complex interplay between the most probable viewport positions, the coding efficiency, and the resulting Quality of Experience (QoE) with respect to the traditional constraints of delivery systems such as bandwidth and latency. MPEG experts have proposed the concept of quality region, which is a rectangular region defined on a sphere, characterized by a quality level ranging from 1 to 100. The main idea is that the content provider determines some quality regions based on offline external information (e.g., content analysis and statistics about viewport positions), and then prepares multiple quality-variable versions of the same 360-degree video based on these quality regions.

We provide in this paper a theoretical analysis of this concept of quality regions for 360-degree videos. We present optimization models to determine the optimal quality regions, subject to a population of clients, the number of quality-variable video versions, and the bandwidth. We aim at maximizing the video quality displayed in the client viewports by identifying (i) the location of the quality region, (ii) their dimensions (or area size), and (iii) the quality inside and outside the regions. Our model enables content providers to prepare 360-degree videos based on the analytics of the head movements collected from the first content consumers. Using a dataset of real head movements captured on an HMD, we study an optimal set of video versions that are generated by our algorithms and evaluate the performance of such optimal viewport-adaptive streaming. We demonstrate
that, for a given overall bit-rate video, the video quality as perceived by the user improves by 102% on average.

2 RELATED WORK

2.1 Quality-Variable Videos Implementation

In the literature, we distinguish two approaches to implement quality-variable 360-degree videos. We give a brief introduction to these approaches in the following, while providing more details in Sections 3.2 and 3.3 on how our model applies to these approach.

**Tile-based Approach.** The motion-constrained tiles are contiguous fractions of the whole frame, which can be encoded/decoded independently and can thus be seen as separated sub-videos. The concept of tiling is part of the High Efficiency Video Coding (HEVC) standardized decoder [18] and is considered as a key supporting technology for the encoding of quality-variable video versions. The tile-based approach has been developed for other multimedia scenarios where end-users consume only a fraction of the video, especially in navigable panorama [10, 22, 25]. This approach has been recently extended to meet the demand of virtual reality and 360-degree video systems. In a short paper, Ochi et al. [20] have sketched a solution where the spherical video is mapped onto an equirectangular video, which is cut into 8x8 tiles. Zare et al. [28] provide more details on the encoding performance of tiling when applied on projected frames. This study demonstrates the theoretical gains that can be expected by a quality-variable implementation of 360-degree video. More recently, Hosseini and Swaminathan [11] proposed a hexaface sphere-based tiling of a 360-degree video to take into account projection distortion. They also present an approach to describe the tiles with MPEG Dynamic Adaptive Streaming over HTTP (DASH) Spatial Relationship Description (SRD) formatting principles. Quan et al. [21] also propose the delivery of tiles based on a prediction of the head movements. Their main contribution is to show that the head movements can be accurately predicted for short segment sizes by using standard statistical approaches. Le Feuvre and Concolato [16] and Concolato et al. [5] studied an optimal viewport-adaptive streaming selection algorithm based on different heuristically defined quality-variable versions of 360-degree videos. In this paper, we focus on an optimization model to generate quality-variable video versions for viewport-adaptive streaming that maximize the quality inside users’ viewports when number of video versions available to the user is limited. To the best of our knowledge, nobody studied before us optimal parameters to generate limited number of quality-variable versions for 360-degree videos.

**Projection-Based Approach.** This approach, which has been proposed by Kuzyakov [14] and is currently implemented in practical systems [30], takes profit from the geometrical projection. Indeed, since state-of-the-art video encoding are based on two-dimensional rectangles, any 360-degree video (captured on the surface of a sphere) needs to be projected onto a two-dimensional surface before encoding. Scientists have been studying spherical projection onto maps for centuries. The most common projections are equirectangular, cube map, and pyramid [6, 27]. The main idea introduced by Kuzyakov [14] is to leverage a feature of the pyramid, projection: the sampling of pixels from the spherical surface to the two-dimensional surface is irregular, which means that some parts of the spherical surface get more distortion after the projection than others. Depending on the position of the base face of the pyramid, the projection, and consequently the video encoding, is better for some parts of the spherical surface. A refined approach based on geometrical projections is the offset projection [30], where a constant directed vector is applied during the projection to change the pixel sampling in the sphere domain while keeping the same pixel sampling (resolution) in the projected domain. It results in a better quality encoding near the “offset direction” and a continuously decreasing qualities for viewports far from this direction.

2.2 Viewport-Adaptive Streaming

Several researchers have concomitantly studied solutions to stream 360-degree videos based on the same principle as in rate-adaptive streaming [5, 6, 15, 16, 21]. A server splits the video into different segments which duration typically vary between 1 s to 10 s. Each segment is then encoded into different representations each representation having different size (in byte) and having different quality distribution. A client decides, thanks to an adaptation algorithm using local information and predictions, which video representation (or set of representations) to download, to match the available bandwidth budget and the future position of the user viewport.

Zhou et al. [30] studied a practical implementation of viewport-adaptive streaming made for the Oculus HMD, and showed that the occlus’ implementation is not efficient: 20% of the bandwidth is wasted to download video segments that are never used. Le Feuvre and Concolato [16] and Concolato et al. [5] studied practical implementation of tile-based quality-variable 360-degree videos viewport-adaptive streaming. Corbillon et al. [6] studied an optimal viewport-adaptive streaming selection algorithm based on different heuristically defined quality-variable versions of 360-degree videos. In this paper, we focus on an optimization model to generate quality-variable video versions for viewport-adaptive streaming that maximize the quality inside users’ viewports when number of video versions available to the user is limited. To the best of our knowledge, nobody studied before us optimal parameters to generate limited number of quality-variable versions for 360-degree videos.

2.3 Regions of Interest

Our work has also some common roots with the literature on Region of Interest (RoI) in video delivery. The human vision system can only extract information at high resolution near the fovea, where the gaze focuses its attention; the vision resolution decreases with eccentricity. Within the same video picture, it is common that most users focus their gaze on some specific regions of the picture, named RoI. Researchers have studied saliency map, which measures the gaze location of multiple users watching the same video. The goal is to extract RoI and, if possible, to corroborate RoI with picture structures to enable automatic RoI prediction [3, 9]. However, the concept of saliency map should be revisited with 360-degree videos, because the head movement is the prevailing factor to determine
the attention of users. To the best of our knowledge, the relation between gaze-based saliency map and head movements in HMD has not been demonstrated.

The attention-based video coding [2, 13, 17] is a coding strategy, which takes advantage of the gaze saliency prediction. The quantization parameters of the encoder are adjusted to allocate more bits near the different RoI and less bits farther away. A live encoder can perform attention-based video coding by using either feedback from a set of specific users or predicted RoI.

We revisit this approach to 360-degree videos in this paper. Our work is both to study per-segment RoI localization based on head movement information and to generate RoI-based encoded video representations. The creation of spherical quality-variable video versions based on head movement analysis enables viewport-adaptive streaming in the same manner that saliency map and attention-based video coding enable efficient video delivery on regular planar videos [9].

3 QUALITY-VARIABLE VIDEOS

We first introduce a model for quality-variable 360-degree videos and then provide some illustrations of this model on some implementation proposals.

3.1 Generic Model

**Spherical videos.** The unit sphere that underlies the 360-degree video is split into \( N \) non-overlapping areas that cover the full sphere. The set of areas is denoted by \( \mathcal{A} \). In essence, each area corresponds to the video signal projected on a given direction of the sphere. Let us denote by \( s_a \) the surface of an area \( a \) on the sphere and observe that the smallest possible surface \( s_a \) is the pixel (in which case the set \( \mathcal{A} \) is the full signal decomposition and \( N \) is the video resolution). However, video preparation processes are generally based on a video decomposition \( \mathcal{A} \) with larger surface \( s_a \), such as the concept of tiles in HEVC [18]. For the preparation of 360-degree videos, any decomposition of the video into \( \mathcal{A} \) can be considered if it respects that it covers the whole sphere, formally \( \sum_{a \in \mathcal{A}} s_a = 4\pi \).

**Area Quality.** The goal of a video encoder is to compress the information of the video signal corresponding to a given area \( a \) into a decodable byte-stream (lossy compression generating distortion when the video is eventually played). An encoder uses a compression algorithm with various parameter settings to encode the video. For a given encoder, the more compression due to the encoding settings, the more distortion in the decoded and played video. Using MPEG terminology, we use the generic term quality to express the settings of the encoding scheme on a given area, regardless of the used area encoding process. The number of different ways to encode areas is finite, which results in a set of available qualities \( Q \) for this encoder (typically the quality ranges from 1 to 100 in MPEG). The set \( Q \) is totally ordered with a transitive comparison function, noted with >.

We provide some natural notations: \( q_{\text{min}} \) (respectively \( q_{\text{max}} \)) is the lowest (respectively highest) possible quality for areas. The encoder processes an area \( a \in \mathcal{A} \) with a quality \( q \) to generate a byte-stream of size \( b_{a,q} \). Given the usual strictly increasing feature of the rate-distortion performance of video encoders, we get that if a quality \( q_1 \in Q \) is better than a quality \( q_2 \in Q \) (formally \( q_1 > q_2 \)), then we have \( b_{a,q_1} > b_{a,q_2} \), \( \forall a \in \mathcal{A} \).

**Viewport.** One of the peculiarities of 360-degree videos is that at a given time \( t \) a user \( u \) watches only a fraction of the whole video, which is generally called the viewport. The viewport displays only a subset of all the areas of the sphere. Let \( v_{u,t,a} \) be a real number equal to the ratio of the surface of area \( a \) that is inside the viewport of user \( u \) at time \( t \) and let \( v_{u,t,a} \) be the average value of \( v_{u,t,a} \) during all time \( t \) in a video segment: \( v_{u,t,a} = \frac{1}{T} \sum_{t} v_{u,t,a} \) with \( T \) the duration of the segment. With respect to the same definition of quality, we have that the average viewport quality during a video segment can be defined as being the sum of the qualities of all the areas that are visible in the viewports, formally \( \sum_{a \in \mathcal{A}} v_{u,t,a} \cdot r(a) \). In practice, the satisfaction of the user watching a viewport is more complex since it depends not only on the visible distortion of the different areas in the viewport but also on the possible effects that different levels of distortion on contiguous areas can produce. Nevertheless, for the sake of simplicity, and with regards to the lack of formal studies dealing with subjective satisfaction evaluation of multi-encoded videos, we consider here that the satisfaction grows with the sum of qualities of the visible areas.

3.2 Illustration: Offset Projections

To apply the implementation of offset projection as presented by Zhou et al. [30] to our model, we need to introduce some additional notations. Let \( 0 \leq \beta \leq 1 \) be a real number, which is the magnitude of the vector used by the "offset" projection. We denote by \( \theta \) the angular distance between the "offset direction" and a given point on the sphere. The variation of the sampling frequency compared to the frequency of the same projection without offset at angular distance \( \theta \) is:

\[
f(\theta) = \frac{1 + 2\beta + \beta^2}{1 + \beta} \left( 1 + \frac{\beta \cos(\theta) + 1}{1 + 2\beta \cos(\theta) + \beta^2} \right)
\]

If we denote by \( D(a_1, a_2) \) the angular distance between the centers of two areas \( a_1 \) and \( a_2 \), offset projections could be modeled by the set of version \( r \in R \) such as there exists \( a_{\text{offset}} \in \mathcal{A} \) such as \( \forall a \in \mathcal{A}, r(a) = f(D(a_{\text{offset}}, a)) \cdot r(a_{\text{offset}}) \).
3.3 Illustration: Tiling
We define the concept of tile and tiled partition to extend our model to tiled versions. A tile is a set of contiguous areas of \( \mathcal{A} \). A tiled partition \( \mathcal{T} \) of \( \mathcal{A} \) is a set of non-overlapping tiles that cover \( \mathcal{A} \). A tiled version using the tiled partition \( \mathcal{T} \) is a version \( r \in \mathcal{R} \) such that the quality is uniform on each tile of \( \mathcal{T} \). Formally, we have \( \forall r \in \mathcal{T}, |r(r)| = 1 \).

Note that in the tiled scenario, the service provider can generate a version for each tile individually without offering a version for the whole video. In this case, the client has to select separately a version for each tile to generate what we denote by a tiled version in our model. This differs from the other scenarios where the service provider is the one that decides which video version to generate.

4 VIEWPORT-ADAPTIVE STREAMING
An adaptive streaming model is modeled as being one client and one server, where the server offers \( J \) different versions of the video, and the client periodically selects one of these versions based on a version selection algorithm.

Server. The main question is to prepare \( J \) versions in \( \mathcal{R} \) among all the possible combinations of qualities and areas. In the practical 360-degree video streaming system described by Zhou et al. [30], the number of versions \( J \) is equal to 30, while the solution that is promoted by Niamut et al. [19] is to offer all the combinations of tiles (typically \( 8 \times 4 \)) and qualities (typically 3). In practice, a low number of versions \( J \) is suitable since it means less files to manage at the server side (96 files in the latter case) and less complexity in the choice of the version at the client side (more than 32 thousand combinations in the aforementioned case). The main variable of our problem is the boolean \( x_r \), which indicates whether the server decides to offer the version \( r \in \mathcal{R} \). Formally, we have:

\[
x_r = \begin{cases} 
1, & \text{if the server offers } r \in \mathcal{R} \\
0, & \text{otherwise} 
\end{cases}
\]

Since the server offers only \( J \) different versions, we have \( \sum_{r \in \mathcal{R}} x_r = J \). In the following, we restrict our focus on the case of a given overall bit-rate budget \( B \), which is a real number. The main idea is to offer several versions of the video meeting the same bandwidth requirement but with different quality distributions. All the versions have thus the same overall bit-rate "budget" but they differ by the quality of the video, which is better at some directions in the sphere than others.

Client. The version selection algorithm first determines the most suitable bit-rate, here \( B \), and then selects one and only one versions among the \( J \) offered versions for every segment of the videos, ideally the version that is the best match to user viewport. To simplify notations, we omit in the following the subscripts related to temporal segments, and we thus denote by \( y_{u,r} \), the binary variable that indicates that user \( u \) selects \( r \in \mathcal{R} \) for the video. Formally:

\[
y_{u,r} = \begin{cases} 
1, & \text{if the client } u \text{ selects } r \in \mathcal{R} \\
0, & \text{otherwise} 
\end{cases}
\]

Since the user selects only one offered versions, we have \( \sum_{r \in \mathcal{R}} y_{u,r} \cdot x_r = 1 \). We consider an ideal version selection algorithm and we thus assume that the client always selects the version that maximizes the viewport quality as previously defined, which is \( r \) such that \( \sum_a v_{u,a} \cdot r(a) \) is maximum.

4.1 Model Formulation
Our objective is to determine, for a given set of users who request the video at bit-rate \( B \), the \( J \) versions that should be prepared at the server side so that the quality of the viewports is maximum. In its most generic form, the problem can thus be formulated as follows.

\[
\max_{y_{u,r}} \sum_{u} \sum_{r \in \mathcal{R}} y_{u,r} \cdot \sum_a v_{u,a} \cdot r(a)
\]

Such that:

\[
\begin{align*}
\sum_a b_{u,a,r(a)} &= B \\
\sum_r x_r &= J \\
y_{u,r} &= 1 \\
y_{u,r} &\leq x_r
\end{align*}
\]

Note that with this formulation the problem is tractable.

5 PRACTICAL OPTIMIZATION MODEL
We take into account some practical additional constraints and some further hypothesis to formulate a tractable optimization problem, which meets key questions from content providers.

5.1 Practical Hypothesis
We first suppose that each area \( a \in \mathcal{A} \) in the whole spherical video has the same coding complexity. This means we suppose that for a given quality, the byte-stream size of a area is proportional to its size. We derive the concept of surface bit-rate, which expresses in Bps/m² the amount of data that is required to encode an area at a given quality. We obtain that \( b_{\text{max}} \) (respectively \( b_{\text{min}} \)) corresponds to the surface bit-rate for the maximum (resp. minimum) quality.

Second, we restrict our study to only two qualities per version. We follow in that spirit the MPEG experts in the Omnidirectional Media Application Format (OMAF) group [12], and notably we follow their recommendation to implement scalable tiled video coding such as HEVC Scalable Extension (SHVC) [4] for the implementation of quality-variable 360-degree video versions. It means that for each version we distinguish a Quality Emphasized Region (QER), which is the set of areas that are at the high quality noted \( q_{\text{er}} \), and the remaining areas, which are at the low quality \( b_{\text{out}} \). In the SHVC encoding, \( b_{\text{qer}} \) corresponds to the video signal with the enhancement layer, while \( b_{\text{out}} \) contains only the base layer. Let \( s_r \) be the overall surface of the areas that are in QER for a given version \( r \in \mathcal{R} \). The bit-rate constraints (1a) can thus be expressed as follow:

\[
s_r \cdot b_{\text{qer}} + (4\pi - s_r) \cdot b_{\text{out}} = B
\]

Third, we introduce a maximum gap between both qualities. The motivation is to prevent the video to have too visible quality changes between areas. This quality gap ratio, denoted by \( r_b \), can be defined as the maximum ratio that relate the qualities \( b_{\text{qer}} \) and
Finally, we define the QER as a rectangular region defined on the sphere as shown in Figure 1. We thus adopt the restriction that has been introduced in the MPEG OMAF [12] to delimit a so-called rectangular region on the sphere. We also adopt the same way to define the region by delimiting two small circles (angular distance \( \nu \text{Dim} \)), two great circles (angular distance \( h\text{Dim} \)) and the spherical coordinates of the region center is \((1, \theta, \phi)\).

In the following, we consider only video versions \( r \in R \) such that there exists \(-\pi \leq \theta \leq \pi, 0 \leq \phi \leq \pi, -\pi \leq h\text{Dim} \leq \pi, \) and \( 0 \leq \nu\text{Dim} \leq \pi \) such that for all area \( a \in A, \) if \( a \) is inside the rectangle characterized by \((\theta, \phi, h\text{Dim}, \nu\text{Dim})\), the bit-rate of \( a \) is \( b_{\text{qer}} \) otherwise it is \( b_{\text{out}} \). We denote such a version by \( r_{\theta, \phi, h\text{Dim}, \nu\text{Dim}} \).

5.2 Bit-Rate Computation

The objective function (1) imply that if two versions have a QERs containing the same areas, the optimal set of offered video versions can only contains the version that maximize the \( b_{\text{qer}} \) subject to the bit-rate constraint (2) and the ratio constraint (3).

In order to simplify the complexity of the model, we pre-computed the value of \( b_{\text{qer}} \) and \( b_{\text{out}} \) depending on the size of the QER \( s_r \). We identify four different cases depending on the size of the QER \( s_r \). For simplicity, we provide in the following the main ideas of the algorithm and put the details of the mathematical model in the Appendix of the paper.

We first combine the constraints given by the overall bit-rate budget with Equation (2) and the knowledge that \( h_{\text{min}} \leq b_{\text{out}} < b_{\text{qer}} \leq b_{\text{max}} \). There are two cases, depending on whether the QER is small or not:

- When the surface of the QER is small, i.e., \( s_r \leq \frac{B - 4\pi h_{\text{min}}}{b_{\text{max}} - b_{\text{min}}} \) (see in Appendix), the constraint on the maximum surface bit-rate prevails for \( b_{\text{qer}} \). The surface bit-rate inside the QER can be maximum. The bit-rate budget that remains after deducting the bit-rate in the QER is \( B - (s_r \cdot b_{\text{max}}) \). This remaining bit-rate budget is large enough to ensure that the surface bit-rate for the areas outside the QER is greater than \( b_{\text{min}} \). We obtain that \( b_{\text{qer}} \) is equal to \( b_{\text{max}} \) and \( b_{\text{out}} \) is derived as:

\[
b_{\text{out}} = \frac{B - (b_{\text{max}} \cdot s_r)}{4\pi - s_r} \tag{4}
\]

- When the surface of the QER is large, i.e., \( s_r \geq \frac{B - 4\pi h_{\text{min}}}{b_{\text{max}} - b_{\text{min}}} \), the constraint on the minimum surface bit-rate prevails. The surface bit-rate inside the QER cannot be \( b_{\text{max}} \), otherwise the remaining bit-rate that can be assigned to the video area outside the QER would not be large enough to ensure that \( b_{\text{out}} \) is greater than \( b_{\text{min}} \). Here, we first have to set \( b_{\text{out}} \) to \( b_{\text{min}} \) and then assign the remaining budget \( B - (b_{\text{min}} \cdot (4\pi - s_r)) \) to the QER area.

\[
b_{\text{qer}} = \frac{B - (b_{\text{min}} \cdot (4\pi - s_r))}{s_r} \tag{5}
\]

Next, we consider the quality gap ratio, which applies to both previously discussed cases:

- When the QER is small, setting \( b_{\text{qer}} = b_{\text{max}} \) and \( b_{\text{out}} \) to (4) can lead to not respect Equation (3). It occurs for any QER such that (see in Appendix):

\[
s_r \geq \frac{4\pi \cdot b_{\text{max}} - B \cdot r_b}{(1 - r_b) \cdot b_{\text{max}}}
\]

The surface bit-rate \( b_{\text{qer}} \) should be instead reset as \( b_{\text{qer}} = r_b \cdot b_{\text{out}} \). This constraint makes that some extra bit-rate are not assigned: \( s_r \cdot (b_{\text{max}} - r_b \cdot b_{\text{out}}) \). These extra bit-rates can thus be re-assigned to both \( b_{\text{qer}} \) and \( b_{\text{out}} \) (see in Appendix).

- When the QER is large, setting \( b_{\text{out}} = b_{\text{min}} \) and \( b_{\text{qer}} \) with Equation (5) can also lead to not respect Equation (3). It occurs for any QER such that:

\[
s_r \leq \frac{4\pi \cdot b_{\text{min}} - B}{(1 - r_b) \cdot b_{\text{min}}}
\]

Similarly as in the previous case, resetting \( b_{\text{qer}} \) with respect to the quality gap ratio leads to release of some extra bit-rates, which can be re-assigned to both \( b_{\text{out}} \) and \( b_{\text{qer}} \).

We represent in Figure 2 the algorithm with the four cases when it applies to standard settings of the overall bit-rate \( B \), the maximum surface bit-rate \( b_{\text{max}} \), the minimum surface bit-rate \( b_{\text{min}} \), and the quality gap ratio \( r_b \). Finally, we show in Figure 3 how the surface bit-rates are assigned depending on the surface \( s_r \) for a given parameter configuration (see in caption and in Section 6). Here the thin gray vertical lines correspond to the threshold at which the algorithm runs a different case.

6 EVALUATION – CASE STUDY

6.1 Settings

We used a custom-made C++ software publicly available on github. This software uses the IBM Cplex library to solve our optimization problem.

Dataset of Head Movements. We used the public head movement dataset that we recently extracted and shared with the community [7]. This dataset contains the head orientation of 59 persons watching, with a HMD, five 70-second-long 360-degree videos. In this paper we used the results from only two out of the five videos available: roller-coaster and diving. We selected those videos because users exhibit different behaviors while watching them: most users focus on a single RoI in the roller-coaster video.

\[1\] In some configurations, it is possible that some of the presented cases do not hold since the threshold for the cases can be negative, greater than \( 4\pi \), or interfering with a prevailing constraint. This however does not occur for the most common configuration parameters such that a quality gap ratio not too large and consistent values for both \( b_{\text{min}} \) and \( b_{\text{max}} \).

\[2\] https://github.com/xmar/adaptive-representation-viewpoint-viewport-adaptive-streaming

\[3\] http://dash.ipv6.enstb.fr/headMovements/
while people move their heads to explore the scene in the diving video.

<table>
<thead>
<tr>
<th>number of offered versions $J$</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>overall bit-rate $B$</td>
<td>12.56 Mbps</td>
</tr>
<tr>
<td>maximum surface bit-rate $b_{max}$</td>
<td>2.1 Mbps/m²</td>
</tr>
<tr>
<td>minimum surface bit-rate $b_{min}$</td>
<td>0.45 Mbps/m²</td>
</tr>
<tr>
<td>quality gap ratio $r_b$</td>
<td>3.5</td>
</tr>
<tr>
<td>number of areas $N$</td>
<td>400</td>
</tr>
<tr>
<td>video segment size</td>
<td>2 s</td>
</tr>
</tbody>
</table>

Table 1: Default evaluation settings

Content Provider Case Study. The default parameters are summarized in Table 1. The content provider generates up to $K = 4$ video versions and solves the optimization problem for every video segment (i.e., each video segment has its own set of versions). The parameters related to the bit-rates are similar as in Figure 3: a total bit-rate budget $B$ of 12.56 Mbps, a maximal surface bit-rate $b_{max}$ of 2.1 Mbps/m² and a minimal surface bit-rate $b_{min}$ of 0.45 Mbps/m². We restricted the positions of the center of the QER on the sphere to 17 possible latitudes and 17 possible longitudes. Moreover the angular distance $hDim$ and the angular distance $vDim$ can take 12 different values. We split the sphere into a total of $N = 400$ areas. We cut the videos of the dataset into 2 s long segments. We solved the optimization model independently for each video segment.

Figure 2: Algorithm for surface bit-rates in and out of the QER. The algorithm depends on the surface of the QER $s_r$. We show here the four different cases, for various surfaces (smallest to largest from left to right).

Figure 3: Surface bit-rates as a function of the QER surface. The overall video bit-rate $B$ is 12.56 Mbps, so the surface bit-rate for a uniform quality is 1 Mbps/m². The maximum surface bit-rate $b_{max}$ is 2.1 Mbps/m² while the minimum $b_{min}$ is 0.45 Mbps/m². Finally, the quality gap ratio $r_b$ is 3.

6.2 Theoretical Gains of Viewport-Adaptive Streaming

Our first goal is to evaluate the possible (theoretical) gains that the implementation of viewport-adaptive streaming can offer to the content providers. The gains can be evaluated from two perspectives: either the opportunity to save bandwidth while offering the video at the same level of quality as if the video was sent with uniform quality, or the opportunity to improve the quality of the video that is displayed at the client side for the same bit-rate as for a standard delivery. We computed the average surface bit-rate inside the viewport of the users (named visible surface bit-rate in the following) for different bit-rate budgets. The average visible surface bit-rate $b_{vqer}$ in the viewport during a segment can be formally written as follow, with $N_u$ the number of user:

$$b_{vqer} = \sum_{r,a} y_{u,r} \cdot \left( \sum_a v_{u,a} \cdot b_{r(a)} \cdot s_a \right) / N_u \cdot \sum_a v_{u,a} \cdot s_a$$  (6)

Figure 4 represents the mean average visible surface bit-rate for all segments of the two selected videos. The horizontal dashed line shows the average visible surface bit-rate for the bit-rate budget of 12.56 Mbps that is uniformly spread on the sphere, while the vertical dashed line indicates the quality for a constant bit-rate of 12.56 Mbps. We also represent the gains from the two aforementioned perspectives (either bit-rate savings or quality).

For a constant average quality inside the user viewports, the delivery of optimally generated QER versions enables 45%
We are interested in studying whether there exists a common
viewport-adaptive streaming system, that makes that the higher
bandwidth savings. For a constant bit-rate budget, the optimal
viewport-adaptive delivery enables an average increase of visible
surface bit-rate of 102%.

### 6.3 Video Content vs. Delivery Settings

We now study the settings of the viewport-adaptive streaming
systems, especially the parameters related to the number of different
versions (\(J\)) and the segment size (\(T\)). We compare the set of versions
that are generated by the optimal solver for both selected videos.
We are interested in studying whether there exists a common best-
practice setting to generate versions, regardless of the video content,
or whether each video should be prepared with respect to the
content by a dedicated process with its own setting. We show the
results computed separately for the roller-coaster and the diving
video. Recall that the roller-coaster video has a single static RoI and
most of the 59 users focus on it. On the contrary, the diving video
has multiple moving RoI, which most users alternatively watch.

Figure 5 represents the average visible surface bit-rate \(b_{\text{v,qer}}\) of
the optimal QER versions for each user and each video segment for
both videos: the roller-coaster video is in plain-green lines while the
diving video is in dashed-blue lines. The results are shown with a
box plot, with the 10th, 25th, 50th, 75th and 90th percentiles for the
30 segments watched by the 59 users of each video in an optimal
viewport-adaptive delivery system.

The viewport-adaptive streaming systems make that the higher
the number of QER versions offered by the content provider, the
better the average quality watched by the users because the set
of versions covers more user behaviors. However, we notice that
there exists a threshold value after which increasing the number of
versions does not significantly improve the quality of the viewport
of the users. This threshold depends on the video content. For the
roller-coaster video, the limit is four QER versions while this limit
is eight for the diving video. Please note that both threshold are
significantly lower than the thirty versions that are generated by
state-of-the-art viewport-adaptive delivery systems [14].

In Figure 7 we fix the number of QER versions to four and we
evaluate the impact of the segment size on the generated QER
versions. Like for Figure 5 the results are displayed with a box plot,
which follows the same color code.

The median quality decreases while the size of the segments
increases. Indeed, the higher the segments size, the wider are the
head movements of the users. But, similarly as in the number of
video versions, we notice that the median average displayed quality
for the diving video is more sensitive to the segment size than
for the roller-coaster video. For the latter, the quality decreases for
segments longer than 2 s while for the diving, the quality decreases
for segment longer than 1 s.

### 6.4 QER Dimensions vs. Overall Bit-rate

We study the main characteristics of the generated QER versions
with a focus on the impact of the global bit-rate budget on the
dimensions. We evaluate both the size of the QER inside each video
version and the shape of the QERs.

Figure 7 represents the cumulative density function (CDF) of
the surface of the QER inside each generated optimal version, for
different global bit-rate budget, for both video. The dashed vertical
black line represents the surface of the viewports of the users as it is
seen in the HMD.

The size of the QERs increases with the overall bit-rate budget.
If the bit-rate budget is small, the size of each QER is smaller than
the surface of the viewports. It means that no user has a viewport
with full quality everywhere. The optimal solver prefers here to
keep a high quality on an area that is common to the viewport of
many users. If we increase the available bit-rate budget, the surface
This paper investigates some theoretical models for the viewport within the QER.

When the provider receives enough analytic, he will be able to generate versions adapted to real user behavior on each video segment. Such functionality would be required in both the processed and the live video viewport-adaptive streaming. Additionally, in this paper we studied only a simplified version of the theoretical model with only two different levels of quality per versions. We plan to study smoother decreasing of the quality inside video versions.

**APPENDIX**

**Limits in the Optimal Bit-Rate Algorithm**

**Constraint on maximum and minimum bit-rate.** Let set \( b_{\text{qer}} = b_{\text{max}} \), which makes that \( s_r \cdot b_{\text{max}} \) bit-rate are used for the QER. The remaining bit-rate can be used to the non-QER: \( b_{\text{out}} = \frac{B - (s_r \cdot b_{\text{max}})}{4\pi - s_r} \). We know that \( b_{\text{min}} \leq b_{\text{out}} \). So:

\[
b_{\text{min}} \leq \frac{B - (s_r \cdot b_{\text{max}})}{4\pi - s_r}
\]

**Constraint on the quality gap ratio.** Let set \( b_{\text{qer}} = b_{\text{max}} \) and \( b_{\text{out}} \) be computed from Equation (4). However, for some \( s_r \), it can happen that \( r_b \cdot b_{\text{out}} \) is lower than \( b_{\text{max}} \):

\[
r_b \cdot \frac{B - b_{\text{max}} \cdot s_r}{4\pi - s_r} \leq b_{\text{max}}
\]

\[
s_r \geq \frac{4\pi b_{\text{max}} - r_b \cdot B}{(1 - r_b) b_{\text{max}}}
\]

**Extra bit-rate assignment.** In some cases, the algorithm obtains (at the step 3 in Figure 2) some so-called extra bit-rate, which comes from the quality gap ratio. This extra bit-rate must be assigned to both the QER and non-QER areas while still maintaining the constraints. Let \( E \) be the extra-bit-rate. Let \( y \) be the ratio of the extra bit-rate that is assigned to the non-QER areas. Let \( b_{\text{int}} \) be an intermediate surface bit-rate computed as in the step 1 in Figure 2. We have:

\[
b_{\text{out}} = b_{\text{int}} + y \cdot \frac{E}{4\pi - s_r}
\]

\[
b_{\text{qer}} = r_b \cdot b_{\text{int}} + (1 - y) \cdot \frac{E}{s_r}
\]
Given that the quality gap ratio is the prevailing constraint in the considered cases, $b_{qer} = r_b \cdot b_{out}$. We thus obtain:

$$r_b \cdot b_{int} + (1 - y) \cdot \frac{E}{s_r} = r_b \cdot \left( b_{int} + y \cdot \frac{E}{4\pi - s_r} \right)$$

$$y = \frac{4\pi - s_r}{4\pi + s_r} \cdot (r_b - 1)$$

REFERENCES


