Heterogeneous Spatial Quality for Omnidirectional Video

Hristina Hristova, Xavier Corbillon, Gwendal Simon, Viswanathan Swaminathan, Alisa Devlic

▶ **To cite this version:**

Hristina Hristova, Xavier Corbillon, Gwendal Simon, Viswanathan Swaminathan, Alisa Devlic. Heterogeneous Spatial Quality for Omnidirectional Video. MMSP 2018 : IEEE 20th International Workshop on Multimedia Signal Processing, IEEE, Aug 2018, Vancouver, Canada. 10.1109/MMSP.2018.8547114 . hal-01896283

HAL Id: hal-01896283

https://imt-atlantique.hal.science/hal-01896283v1

Submitted on 16 Oct 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.

Heterogeneous Spatial Quality for Omnidirectional Video

Hristina Hristova*, Xavier Corbillon*, Gwendal Simon*, Viswanathan Swaminathan†, Alisa Devlic‡
*IMT Atlantique †Adobe ‡Huawei

***Abstract*—A video with heterogeneous spatial quality is a video where some regions of the frame have a different quality than other regions (for instance, a better quality could mean more pixels and less encoding distortion). Such a quality-variable encoding is a key enabler of Virtual Reality application, with 360-degree videos. So far, the main technique that has been proposed to prepare spatially heterogeneous quality is based on the concept of tiling. More recently, Facebook has implemented another approach: the *offset projection* where more emphasis is put on a specific direction of the frame. In this paper, we study quality-variable 360-degree videos with two main contributions. First, we provide the theoretical analysis of the offset projection and show the impact of the parameter settings on the video quality. Second, we propose another approach which consists in preparing the 360-degree video from a Gaussian pyramid of downscaled and blurred versions of the video. We perform an evaluation of tiling, offset and Gaussian-based approaches in representative scenarios of heterogeneous spatial quality in 360-degree videos and highlight the main trade-off to consider when implementing these approaches.**

I. Introduction

To deliver 360° videos, the content providers implement *viewport-adaptive streaming* solutions [2, 13, 15], where the delivered video is characterized by *heterogeneous spatial quality*: some regions of the frame have a better quality than others [4]. The motivation is twofold: (i) display a high-quality video in the *viewport* of the client, and (ii) reduce the delivered bit-rate by encoding less information in the regions that are unlikely to be watched.

In a video with a spatially heterogeneous quality, each pixel is associated with a target quality, which ranges in a given ordered set. The rationale behind the mapping between quality and pixels, whether it comes from a statistical analysis of previous sessions [3, 11, 17] or from a content analysis [10], is out of the scope of this paper. We assume that the content provider has defined a small set of qualities and a set of *regions* consisting of contiguous pixels with the same quality. An ideal implementation of spatially heterogeneous quality in a 360° video is characterized by:

Precision The *visual quality* of a region in the frame reflects the specifications, both in terms of region boundaries and relative quality with respect to the quality in other regions.

Smoothness The pixels at the boundary of two contiguous regions enable a smooth transition between both regions.

Encoding Efficiency The bit-rate budget that is necessary to implement the spatially heterogeneous quality is low with respect to the obtained visual quality of the regions.

Universality The process of preparing the video can be implemented and widely deployed.

Requirement The resources (in particular, computing power and memory) that have to be provisioned to prepare the video are available in standard media servers.

To prepare heterogeneous spatial quality in videos, the concept of *tiling* has received the most attention. Tiling is offered by the motion-constrained tile sets (MCTS) feature in the High Efficiency Video Coding (HEVC) encoder [5, 9, 14, 19]. The key idea of tiling is to spatially cut the video frame into non-overlapping blocks of pixels, called tiles, and to encode them independently [12]. Before streaming the video, a quality for each tile is selected depending on its location and the current quality-region input specifications [16]. The drawbacks of tiling become evident when analyzed with regards to the aforementioned fundamental characteristics. First, the precision of tiling depends on the number of tiles. A large number of tiles comes at the price of a lower encoding efficiency [1] and an increased resource requirement (high signalling overhead and heavier content delivery network (CDN) management). Moreover, the visual quality changes abruptly at the boundary of two contiguous tiles. Finally, tiling has a low universality. So far, the only fast open-source HEVC encoder to implement tile encoding is *Kvazaar* [7]. Hence, despite being the most widely used method, tiling is not an ultimate solution, and the preparation of heterogeneous spatial quality in 360° videos is still an open research question.

To overcome the limitations of tiling, in this paper we propose two new approaches that implement heterogeneous spatial quality in 360° videos:

Gaussian Pyramid Composition An input video is processed pixel-wise using multiple decreasing qualities, arranged in a Gaussian pyramid. For each pixel in the output video, the content provider picks a pixel from a Gaussian level with respect to the expected quality at this pixel. Such a pixel-wise approach has never been explored for preparing heterogeneous spatial quality in 360° videos.

Formal Study of Offset Projection The idea consists in modifying the sphere-to-plane projection before video encoding [20]. The content provider applies a patch to the projection to map more spherical pixels to a given area on the planar image. Despite the interest in this approach, the impact of the offset parameters on the visual quality after video encoding has never been formally studied.

We study these two approaches by introducing their mathematical principles, by analyzing their impact on the video visual quality, and by comparing them to the traditional tiling approach. In essence, this paper shows that three approaches exist to implement heterogeneous spatial quality in 360° videos: by leveraging encoder distortion (the tiling approach), by blurring images (the Gaussian approach), and by unequal pixel sampling (the offset approach). We reveal their respective advantages and weaknesses as well as open perspectives for this research challenge.

II. Definitions

The input of the content provider is a 360° spherical video, which is captured by an omnidirectional camera (either two fish-eye cameras, or a set of multiple stitched cameras). The content provider aims to generate a video, which is projected into a two-dimensional rectangular area (in this paper, we focus on the *equirectangular* projection but our results can be extended to other projections) and is then encoded with regards to some specific characteristics related to heterogeneous spatial quality.

The concept of heterogeneous spatial quality is defined by two input parameters. First, we define $\mathcal{Q}$ as an ordered set of qualities. This set is not formally associated with any precise measurable scale. Following the definition from MPEG [6], the quality is a rather vague indicator. We thus consider that there exist Q qualities in $\mathcal{Q}$, denoted q_i for $i \in \{0, 1, \ldots, Q-1\}$, where q_0 is the lowest quality, and q_{Q-1} is the best quality.

Second, we define a set of *regions*, which are sets of contiguous pixels in a frame. The idea behind the concept of quality-region is the following. The content provider considers that the pixels in a given region have approximately the same probability to be displayed in the viewport, so they must be encoded at the same quality, i.e. the higher the probability to be displayed in the viewport, the higher the quality. Corbillon et al. [4] have considered a single region per video and have represented this region as a compact area on the sphere (i.e. spherical rectangle). Here, we do not restrict the regions to spherical rectangles. On the contrary, we assume that the regions may have any shape. We denote the set of regions by $\mathcal{R}$. The regions do not overlap and cover the whole frame. The quality of a region $R \in \mathcal{R}$ is a spatial function denoted $q(R) \in \mathcal{Q}$. The qualities of any two regions in $\mathcal{R}$ can be either different or the same.

III. Gaussian Pixel-Wise Encoding

A. Background on Gaussian Pyramids

Image pyramids are multi-scale representations of an image. In particular, the Gaussian pyramid has been widely used to carry out various image processing tasks. The Gaussian pyramid is constructed using a function, denoted $downSample(\cdot)$. The function $downSample\,(I, f(\cdot))$ performs two operations as follows: (i) blurs an input image I using the spatial weighing function $f(\cdot)$ and (ii) reduces both dimensions of image I by a scale of 2, i.e. $(W_1, H_1) = (W/2, H/2)$, where $W \times H$ is the original resolution of I . The blurring is carried out by convolving the input image I with the function $f(\cdot)$, also referred to as the *blurring kernel*. To generate the Gaussian pyramid of image I, I is convolved with a Gaussian kernel, i.e. each pixel of the blurred image $G^{(1)}$ is computed as a weighted average of the pixels of I in a 5×5 neighbourhood, as follows:

$$G^{(1)}(i,j) = \sum_{x=-2}^{2}\sum_{y=-2}^{2} f(x,y)I(2i+x, 2j+y) \quad (1)$$

$$f(x,y) = \omega(x)\omega(y) \quad (2)$$

$$\omega(\cdot) = (0.25 - p/2, 0.25, p, 0.25, 0.25 - p/2), \quad (3)$$

where the parameter $p \in [0.3, 0.6]$ controls the shape of the Gaussian kernel $f(\cdot)$. Once the blurred image $G^{(1)}$ is computed using Section III-A, the function $downSample\,(\cdot, f(\cdot))$ downsamples image $G^{(1)}$ by rejecting even rows and even columns, which reduces the resolution of the resulting image, $g^{(1)}$, in half. Both the Gaussian blur and function $downSample(\cdot)$ can then be applied to the resulting image $g^{(1)}$ in order to compute a second scale of the input image I. The sequence of images $G = \{g^{(0)}, g^{(1)}, \ldots, g^{(N)}\}$, where $g^{(0)} = I$ and $g^{(i)} = downSample(g^{(i-1)}, f(\cdot))$ for $i \in (1, N]$, denotes the Gaussian pyramid of image I. The resolution of each image in the sequence G is twice as small as its predecessor.

We denote the inverse function of $downSample(\cdot)$ as $upSample(\cdot)$. The function $upSample(g^{(i+1)}, f(\cdot))$ first upscales the image $g^{(i+1)}$ to dimensions (W_i, H_i), such that $(W_i, H_i) = (2W_{i+1}, 2H_{i+1})$ ($W_i \times H_i$ being the resolution of the Gaussian level i) by injecting zero even rows and columns. Then, the resulting upscaled image G_{i+1} is convoluted with a Gaussian kernel $f(x, y)$, identical to the one used in function $downScale(\cdot)$, as follows:

$$G^{(i+1)}(i,j) = 4\sum_{x=-2}^{2}\sum_{y=-2}^{2} f(x,y)G_{i+1}\left(\left[\frac{i-x}{2}\right], \left[\frac{j-y}{2}\right]\right), \quad (4)$$

where $[a]$ denotes the nearest integer value to a.

B. Pixel-Wise Video Composition

We propose a multi-scale spatial approach for implementing heterogeneous spatial quality in 360° videos based on the Gaussian multi-scale representation. It consists in two processes: an offline process to construct the Gaussian pyramid, and an online process to compose the video version with heterogeneous spatial quality.

Gaussian Pyramid Construction. For a given input video, we compute the sequence $\{g^{(i)}\}_{i=0}^{N}$, where each element $g(i)$ contains the i^{th} layer of the Gaussian pyramid, constructed for all frames of the video. We refer to $\{g^{(i)}\}_{i=0}^{N}$ as the Gaussian pyramid of the original 360° video. The levels of the Gaussian pyramid represent the degradation of the video quality in $N + 1$ multi-scale levels. Then, we upscale each level $g^{(i)}$ to the resolution of the original video. Each level $G^{(i)}$ of the upscaled Gaussian pyramid is the

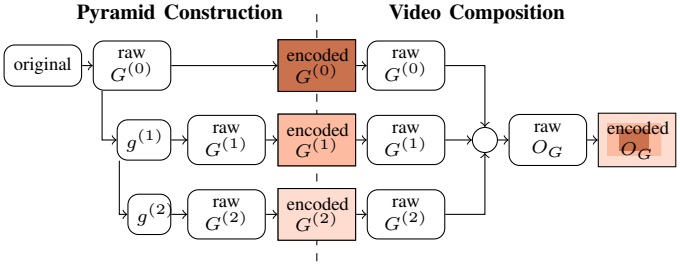
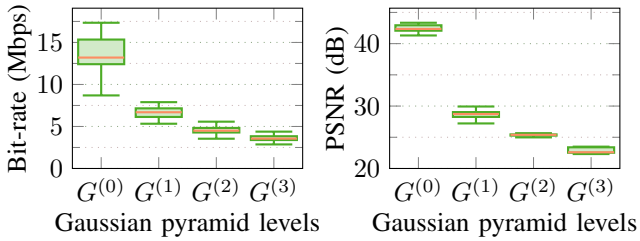


Figure 1: Construction of the Gaussian pyramid (left part) and video composition (right part).



(a) Bit-rate **(b)** Distortion in PSNR

Figure 2: The bit-rate and distortion distributions for the first four levels of the upscaled Gaussian pyramid.

result of upscaling and convolving with the Gaussian kernel i times. This process, illustrated in Figure 1 (left part), leads to a certain amount of redundant information, which is exploited by the encoder to better compress the upsampled videos. To analyze the bit-rate gain, we chose three representative 1 min-long 360° equirectangular videos [3] and we computed their upscaled Gaussian pyramids. Then, we encoded each level of the upscaled Gaussian pyramids using a constant Quantization Parameter (QP) value. In Figure 2, we present the distributions of the video bit-rates for each Gaussian level as well as the respective quality distortions. We show the efficiency of the Gaussian approach for generating videos at variable qualities and bit-rates by applying basic pixel operations.

Create Quality-Variable Video. The upscaled Gaussian pyramid $\{G^{(i)}\}_{i=0}^{N}$ is then exploited to create quality-variable videos as illustrated in Figure 1 (right part). We use a linear optimization to map the qualities, provided by the function $q(R)$, to a given upscaled level $G^{(i)}$. It takes into account the overall video bit-rate budget and chooses the most appropriate Gaussian levels with respect to this budget. Let $\phi(\cdot)$ be a function that, for every pixel, maps $q(R)$ to optimal Gaussian levels given a bit-rate budget B.

The video composition, prepared at the server side, is computed from the upscaled Gaussian pyramids for each pixel (u, v) as follows:

$$O_G(u,v) = G^{(\phi(q(x,y),B))}(u,v), \tag{5}$$

where $\{O_G\}$ is the output video with spatially-varying quality.

C. Analysis

The Gaussian-based pixel-wise video composition approach requires the server to first compute the $N+1$ levels of the Gaussian pyramid and encode them (for optimal storage). This process can be done only once, as the pyramid can then be re-used to create any number of different quality-variable videos. The preparation of a quality-variable video requires that the server decodes $\min(N+1, |\mathcal{R}|)$ Gaussian pyramid levels, performs the spatial operation to compose the spatially heterogeneous video, and encodes it. To sum up, the number of operations per video composition is: (i) decoding - $\min(N+1, |\mathcal{R}|)$ times; (ii) one spatial video composing; (iii) one video encoding.

IV. Offset projection

The *offset transformation* [8] is a bijective sphere-to-sphere transformation, which increases the quality of a 360° video near an emphasized direction $\vec{\mathbf{b}}$. When two pixels on the sphere are close to (resp. far from) the emphasized direction $\vec{\mathbf{b}}$, the offset transformation decreases (resp. increases) the angular distance between the two pixels. The *offset projection* is a composition of a sphere-to-plane projection with an offset transformation. First, the spherical image is distorted using the offset transformation, then the distorted spherical image is projected to a plane using a sphere-to-plane projection. The offset projection maps on the plane more pixels close to $\vec{\mathbf{b}}$ and fewer pixels far from $\vec{\mathbf{b}}$. The spatial sampling (thus, the quality) of the video decreases when the angular distance to $\vec{\mathbf{b}}$ increases.

A. Theory

The offset projection has been experimentally studied by Zhou et al. [20] in the case of the cube-map projection. It is characterized by the following parameters: a projection function $f : \mathbb{R}^3 \to \mathbb{R}^2$, an emphasized direction $\vec{\mathbf{b}}$, and an offset *amplitude* α. The projection function f can be any sphere-to-plane projection. The emphasized direction $\vec{\mathbf{b}}$ is a unit vector, pointing to the direction of space emphasized by the offset projection, and α is a real value in $[0, 1)$.

A sphere-to-plane projection maps a spherical pixel, characterized by a unit vector $\vec{\mathbf{a}}$ pointing from the origin of the sphere to the pixel, to a point (u, v) on the plane.

The offset transformation $F : \mathbb{R}^3 \to \mathbb{R}^3$ distorts the sphere by mapping the vector $\vec{\mathbf{a}}$ to the vector $\left(\vec{\mathbf{a}} + \alpha\vec{\mathbf{b}}\right)/\left\|\vec{\mathbf{a}} + \alpha\vec{\mathbf{b}}\right\|$. The offset projection $g : \mathbb{R}^3 \to \mathbb{R}^2$ is then given as follows:

$$g(\vec{\mathbf{a}}) = f \circ F(\vec{\mathbf{a}}) = f\left(\left(\vec{\mathbf{a}} + \alpha\vec{\mathbf{b}}\right)/\left\|\vec{\mathbf{a}} + \alpha\vec{\mathbf{b}}\right\|\right) \tag{6}$$

In Equation (6), both projection functions f and g transform a given viewing direction $\vec{\mathbf{a}}$ into planar coordinates (u, v). The inverse offset projection g^{-1}, transforming coordinates (u, v) into a viewing direction $\vec{\mathbf{a}}$, is defined as follows:

$$g^{-1}(u,v) = F^{-1} \circ f^{-1}(u,v) \tag{7}$$

where f^{-1} is the plane-to-sphere projection, and F^{-1} is:

$$F^{-1}(\vec{\mathbf{a}}) = \left(\vec{\mathbf{a}} \cdot \alpha\vec{\mathbf{b}} + \sqrt{(\vec{\mathbf{a}} \cdot \alpha\vec{\mathbf{b}})^2 - \alpha + 1}\right)\vec{\mathbf{a}} - \alpha\vec{\mathbf{b}} \tag{8}$$

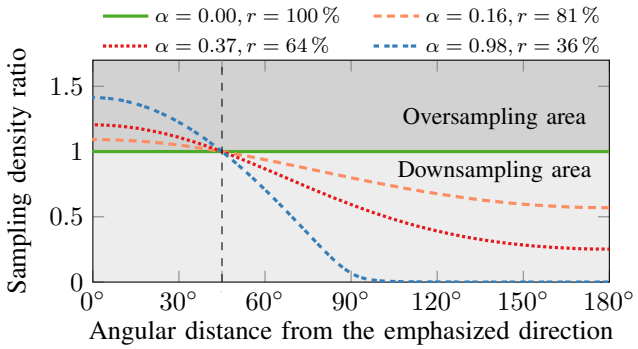


Figure 3: Sampling ratio $\sigma_{\alpha,r}/\sigma_{\vec{0},1}$ for the equirectangular offset projection for $\vec{b} = (1, 0, 0)$ at constant latitude $\frac{\pi}{2}$.

with $\cdot$ being the inner vector product in $\mathbb{R}^3$.

To study how the offset projection continuously degrades the quality of the spherical image, we measure the sampling density of the projection on the sphere. The sampling density is a number that represents the number of pixels per surface unit on the sphere. It can be computed for any point on the sphere. If we consider only a finite number of pixels $W \times H$ on the plane, then the planar image can be split into $W \times H$ similar squares. A point p on the sphere is projected inside a unique square s_p on the plane. The sampling density at point p on the sphere can be approximated by the inverse of the surface of the square s_p, once s_p is projected back on the sphere. In what follows, $\sigma_{\alpha,r}$ denotes the sampling density function for the equirectangular offset projection with an amplitude α, an emphasized direction $\vec{b} = (1, 0, 0)$ and a resolution $r(W \times H)$. Hereafter, $r \in [0, 1]$ refers to the resolution ratio.

Increasing the amplitude α for a constant r increases the sampling near the emphasized direction and decreases it near the direction opposite $\vec{b}$. Put differently, when α increases, more pixels from the plane are assigned to spherical angles near the emphasized direction and less in the opposite direction. Figure 3 depicts the offset sampling density compared to the sampling density at the same position in the original video, *i.e.* it depicts $\sigma_{\alpha,r}/\sigma_{0,1}$. When $\alpha = 0$, the sampling density ratio is constantly equal to r. As illustrated by the curve for $r = 0.36$ in Figure 3, when α tends towards 1, the sampling density ratio tends toward 0 for points farther than 90° from $\vec{b}$, and tends towards $4r$ in the emphasized direction $\vec{b}$ (*i.e.* tends towards 1.44 for $r = 0.36$). The latter means that the offset projection with an amplitude $\alpha > 0$ cannot increase the sampling $\sigma_{0,r}$ in the emphasized direction more than four times the sampling for $\alpha = 0$.[1]

Furthermore, when $\alpha > 0$, the sampling decreases with the distance from the $\vec{b}$ and the transition between the emphasized and non-emphasized regions is smooth. Hereafter, we define the Quality Emphasized Region (QER) for the offset projection as being the zone of the video for which the sampling density ratio is at least equal to the original sampling ratio (in Figure 3, the original sampling ratio is constant and equal to 1). For a given resolution ratio r, there exists at most one value of α such that the angular size of the QER is equal to a given value. For instance, Figure 3 shows that for $r = 0.64$, a region of angular size 90° (i.e. 45° from $\vec{b}$) is a QER when $\alpha = 0.37$ (resp. $\alpha = 0.98$ for $r = 0.36$ and $\alpha = 0.16$ for $r = 0.81$).

[1]Due to page limitations, the demonstration is not in the paper.

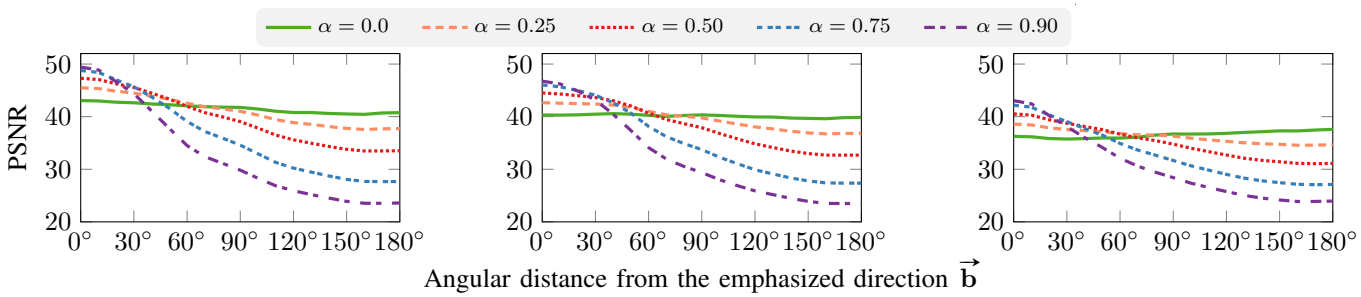
B. Experiments and analysis

We now evaluate the impact of r and α on the visual quality of the videos. So far, only Zhou et al. [20] have studied the quality distortion caused by the offset approach. The authors compare the video quality degradation of the offset cube projection to the quality of the original equirectangular video. Their experiments show that, for a given α, the offset cube projection produces videos with similar quality as a video at a higher resolution within a certain angular distance from the offset center. However, Zhou et al. do not elaborate on the choice of α and, more importantly, how this choice influences the quality of the offset videos. Hereafter, we present a more thorough analysis of the quality distortion, caused by the offset (equirectangular) projection given various video resolutions and various values of the offset amplitude.

We applied the offset equirectangular projection on three 360° videos using three resolution ratios, *i.e.* $r \in \{0.81, 0.64, 0.36\}$, and five amplitude values, i.e. $\alpha \in \{0, 0.25, 0.5, 0.75, 0.9\}$, for each resolution ratio r. To reduce the content-related bias in the quality measurement, we computed the offset projection *eight* times per video by applying different sphere rotations. To measure the distortion, introduced by the offset projection, we computed the Peak Signal to Noise Ratio (PSNR) between viewports from both the original equirectangular video and the offset videos. We extracted viewports at varying angular distances from the emphasized direction $\vec{b}$ to illustrate the relation between quality degradation and distance to the offset center.

Figure 4 shows the PSNR quality curves for each considered resolution ratio r and each amplitude α. The curves for $\alpha = 0$, i.e. when no offset is applied, are referred to as baselines. Each baseline measures the distortion caused by the resolution decrease (by r). The PSNR remains constant regardless of the distance to the spherical center. In contrast, for $\alpha > 0$, the video quality varies depending on the angular distance to the emphasized direction $\vec{b}$. The PSNR curves show that, for $\alpha > 0$, the quality of the viewports close to $\vec{b}$ is higher than the baseline quality. The farther we get from the center of emphasis, the higher the distortion.

Furthermore, for all resolution ratios r, the more we increase the amplitude α, the more we improve the video quality near the emphasized direction $\vec{b}$ and the more we degrade the quality near the opposite direction to $\vec{b}$. The plots in Figure 4 show that the quality curves intersect the baseline at smaller angles when α increases. For instance, the curve for $\alpha = 0.5$ intersects the baseline at 60°, whereas the curve for $\alpha = 0.9$ intersects the baseline at 40°. The latter means that the higher the amplitude α, the smaller the



(a) Resolution: 81 %

(b) Resolution: 64 %

(c) Resolution: 36 %

Figure 4: PSNR between viewports, extracted from the original equirectangular video, and viewports, extracted from the equirectangular offset projection for various values of the amplitude α and three resolution ratios r.

emphasized region. As shown in Figure 3, for high amplitude values, e.g. $\alpha = 0.98$, most of the samples are concentrated in a small region, centered at the emphasized direction $\vec{b}$, and there are less samples for regions away from the center of emphasis. This explains the rapid decrease in the quality when the distance from $\vec{b}$ increases.

V. Evaluation

In Table I, we summarize the main differences between the three approaches with regards to four out of the five main characteristics, introduced in Section I. Due to page limitations, the analysis of the computing and storage requirements for composing 360° videos as well as the user reaction to abrupt quality changes are left for future work. Here, we focus on the encoding efficiency by measuring the visual quality of the videos created using the three approaches, for a similar overall bit-rate.

	Tile	Offset	Gaussian
Precision	depends on nb. tiles	low	high
Smoothness	no	yes	no
Universality	encoding issue	yes	yes
Requirements	depends on nb. tiles	low	depends on nb. videos

Table I: Summary of main characteristics

Visual Quality Metric. We lack a metric to capture the heterogeneity of quality in a 360° video. To fill this gap, we created a new metric based on the Spherical Peak Signal to Noise Ratio (S-PSNR) Yu et al. [18]. The S-PSNR first maps each point on the sphere to a location in both the original and the encoded videos. Then, the pixel values, corresponding to these locations are used to determine the distortion between the two pixels. Finally, the errors for all spherical points are averaged to compute the S-PSNR. However, the concept of heterogeneous spatial quality in 360° videos takes its root from the fact that not all spherical pixels are equal (as some pixels have higher probability to be in the clients' viewports). To capture this heterogeneity, we propose to weigh the error of each pixel with respect to its expected quality. We thus use our quality function $q(R)$ to weigh the S-PSNR and obtain a new *weighted* metric, denoted WS-PSNR. The highest weight is assigned to spherical points which, after projection, lie in the region with the highest quality. The weights of the spherical points lying within the other quality regions decrease exponentially.

To sum up, the S-PSNR measures the average quality of the entire encoded video, whereas the WS-PSNR measures the quality with respect to the given quality regions. We combine both metrics to obtain an *interpolated* quality estimation, denoted IS-PSNR.

Quality Regions. We consider two configurations of quality regions (Figure 5), which represent common viewport movements [4]. The □-*shape* is common for a video where viewers stably focus on a single location, whereas the *V-shape* may appear when the object of interest is moving.

Video Preparation. We use three equirectangular 360° videos from a public dataset [3] (*rollercoaster*, *venice*, and *timelapse*). Their resolution is 3840×2160 pixels (4K). We use the Kvazaar encoder [7] with three bit-rate targets: 6 Mbps, 9 Mbps and 14 Mbps. Specific settings include:

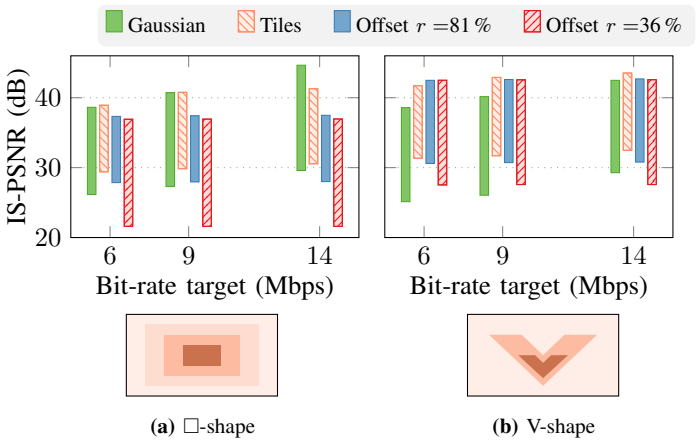
Tiles We set 8×8 tiles. We encode four tiled videos with four bit-rate targets (3 Mbps, 7 Mbps, 13 Mbps and 21 Mbps). We extract each tile from each tiled video. To obtain a quality-variable video matching the quality regions and the bit-rate target, we select each tile from one of the four qualities so that the overall bit-rate is close to the target.

Offset We chose two resolutions: 3686×1944 ($r = 81\,\%$ of the original video) and 2457×1296 ($r = 36\,\%$). The offset intensity α is set so that a 90° of the QER (resp. 50°) is oversampled for □-shape (resp. V-shape).

Gaussian The Gaussian pyramid is built by halving the video dimensions and applying a Gaussian kernel. The final video is composed from pre-selected upscaled Gaussian levels.

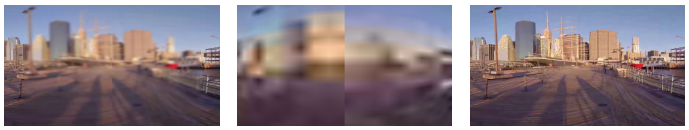
Result Analysis. In Figure 5a (resp. in Figure 5b), we show the IS-PSNR for the three bit-rate targets and for the □-shape (resp. V-shape) quality regions. The lowest and the highest values of the bars correspond to the S-PSNR and the WS-PSNR respectively. In Figure 6, we show snapshots of the back viewports and thus highlight the different approaches (blurring, distorting, and reducing bit-rate by encoding).

First, we observe that all approaches manage to prepare 360°



(a) □-shape **(b)** V-shape

Figure 5: IS-PSNR bars for two quality regions



(a) Back Gaussian **(b)** Back Offset 36 % **(c)** Back Tiles

Figure 6: Viewport in the non-QER for the 6 Mbps videos

videos with heterogeneous qualities. In each configuration, the visual quality in the emphasized regions, measured by the WS-PSNR), is higher than the visual quality, measured by the S-PSNR. The difference between both metrics is greater or equal to 10 dB, which is a significant gap.

The offset approach is relevant for low encoding bit-rates, but it does not benefit from extra bit-rates. Also, the offset is sensitive to the viewport location, with large quality range between the S-PSNR and the WS-PSNR, especially at low resolutions. To obtain a high quality in a large viewport despite the low resolution, the price we pay is a severely degraded quality in the direction opposite $\vec{\mathbf{b}}$, as epitomized in Figure 6. The Gaussian approach is also characterized by large quality ranges. The latter is a result of the significant quality gap between $G^{(0)}$ and $G^{(1)}$ (Figure 2). The gain, obtained by exploiting the blurred Gaussian levels, enables the encoder to maintain very high quality in the viewport without decreasing the resolution. The Gaussian approach never reaches the low quality levels, observed for the offset. The Gaussian approach is sensitive to the video bit-rate: the quality increases steadily with the increase of the bit-rate budget. Finally, the tiling approach appears more stable. It offers a consistent good quality in the best cases and the quality never reaches very low levels (unlike the offset). The back viewport in Figure 6 has also a good visual quality. However, tiling requires specific encoders, and the abrupt changes between tiles can degrade user's experience.

VI. Conclusion

In this paper, we study the preparation of heterogeneous spatial quality in 360° videos. We propose a novel method, based on the Gaussian pyramid, and we analyze the theoretical principles of the offset projection. We identify three families of approaches: a quality degradation by the encoder (tiling), a Gaussian kernel (Gaussian), and an unequal projection (offset projection). We compare the three approaches in terms of encoding efficiency.

This paper is the first one to formally study heterogeneous spatial quality in 360° videos. Each of the proposed approaches deserves a deeper analysis to better understand the impact of the settings on the overall performance. Then, the integration of the propositions in the global delivery chain of viewport-adaptive streaming solutions also deserves a deeper analysis with respect to the constraints of the services, including definition of quality regions, live requirements, and implementation in CDN.

References

[1] C. Concolato, J. Le Feuvre, F. Denoual, F. Maze, N. Ouedraogo, and J. Taquet. Adaptive streaming of hevc tiled videos using mpeg-dash. *IEEE Trans. on Circuits and Systems for Video Tech.*, 2017.

[2] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski. Viewport-adaptive navigable 360-degree video delivery. In *IEEE ICC*, 2016.

[3] X. Corbillon, F. De Simone, and G. Simon. 360-degree video head movement dataset. In *ACM MMSys*, 2017.

[4] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski. Optimal Set of 360-Degree Videos for Viewport-Adaptive Streaming. In *MM*, 2017.

[5] M. Hosseini and V. Swaminathan. Adaptive 360 VR video streaming based on MPEG-DASH SRD. In *IEEE ISM*, 2016.

[6] ISO/IEC 23000-20. Omnidirectional media application format (omaf) committe draft. MPEG, November 2017. ISO/IEC JTC1/SC29/W11.

[7] A. Koivula, M. Viitanen, A. Lemmetti, J. Vanne, and T. D. Hämäläinen. Performance evaluation of Kvazaar HEVC intra encoder on Xeon Phi many-core processor. In *IEEE GlobalSIP*, 2015.

[8] E. Kuzyakov. End-to-end optimizations for dynamic streaming. Blog, Feb 2017. https://code.facebook.com/posts/637561796428084.

[9] J. Le Feuvre and C. Concolato. Tiled-based Adaptive Streaming using MPEG-DASH. In *ACM MMSys*, 2016.

[10] J.-S. Lee, F. De Simone, and T. Ebrahimi. Efficient video coding based on audio-visual focus of attention. *Journal of Visual Communication and Image Representation*, 22(8):704–711, 2011.

[11] W. Lo, C. Fan, J. Lee, C. Huang, K. Chen, and C. Hsu. 360° video viewing dataset in head-mounted virtual reality. In *ACM MMSys*, 2017.

[12] K. M. Misra, C. A. Segall, M. Horowitz, S. Xu, A. Fuldseth, and M. Zhou. An overview of tiles in HEVC. *J. Sel. Topics Signal Proc.*, 7 (6):969–977, 2013.

[13] S. Petrangeli, V. Swaminathan, M. Hosseini, and F. De Turck. An HTTP/2-Based Adaptive Streaming Framework for 360 Virtual Reality Videos. *ACM MM*, 2017.

[14] Y. Sánchez, R. Skupin, and T. Schierl. Compressed domain video processing for tile based panoramic streaming using HEVC. In *IEEE ICIP*, 2015.

[15] K. K. Sreedhar, A. Aminlou, M. Hannuksela, and M. Gabbouj. Viewport-adaptive encoding and streaming of 360-degree video for virtual reality applications. In *IEEE ISM*, 2016.

[16] H. Wang, V.-T. Nguyen, W. T. Ooi, and M. C. Chan. Mixing Tile Resolutions in Tiled Video: A Perceptual Quality Assessment. In *ACM NOSSDAV*, 2014.

[17] C. Wu, Z. Tan, Z. Wang, and S. Yang. A dataset for exploring user behaviors in VR spherical video streaming. In *ACM MMSys*, 2017.

[18] M. Yu, H. Lakshman, and B. Girod. A Framework to Evaluate Omnidirectional Video Coding Schemes. In *IEEE ISMAR*, 2015.

[19] A. Zare, A. Aminlou, M. Hannuksela, and M. Gabbouj. HEVC-compliant tile-based streaming of panoramic video for virtual reality applications. In *ACM MM*, 2016.

[20] C. Zhou, Z. Li, and Y. Liu. A measurement study of occulus 360 degree video streaming. In *ACM MMSys*, 2017.