Received: 28 May 2019 | Revised: 31 May 2019 | Accepted: 31 May 2019

DOI: 10.1002/itl2.118

LETTER

WILEY

Reducing the network bandwidth requirements for 360° immersive video streaming

Luca De Cicco | **Saverio Mascolo** | **Vittorio Palmisano** | **Giuseppe Ribezzo**

Dipartimento di Ingegneria Elettrica e dell'Informazione (DEI), POLIBA, Politecnico di Bari, Bari, Italy

Correspondence
Luca De Cicco, Dipartimento di Ingegneria Elettrica e dell'Informazione (DEI), Politecnico di Bari, Via Orabona, 4, 70125 Bari, Italy.
Email: luca.decicco@poliba.it

Funding information
Italian Ministry of Economic Development (MISE), F/050136/01/X32

Virtual Reality/Augmented Reality applications are getting increasingly popular and new services are emerging in a diverse set of fields such as entertainment, art, e-health, and smart factories. 360° videos are an advancement of classical two-dimensional videos giving the user the impression of being immersed at the center of a scene that can be explored freely and dynamically by simply turning the head to a given region of interest (RoI). Providing a high Quality of Experience to users when streaming 360° videos over the Internet is particularly challenging due to the very high bandwidth requirements. In this letter, a scaling technique to reduce bandwidth requirements to stream omni-directional videos is presented. An experimental investigation of the proposed approach has shown that it is possible to obtain a reduction of the required bitrate up to around 50% while gracefully degrading visual quality far from the RoI.

KEYWORDS
augmented reality/virtual reality, experimental performance evaluation, immersive video streaming, MPEG-DASH

1 | INTRODUCTION AND BACKGROUND

Virtual Reality/Augmented Reality applications are steadily increasing their popularity, thanks to the improvements and the penetration of cheap Head Mounted Displays in the consumer market. In such a context, the ability to stream 360° videos, or omni-directional video contents, is a key enabling technology for several emerging applications such as immersive cinema, social media, and immersive health care. Its technological relevance is testified by the fact that leading platforms such as YouTube and Facebook are already delivering 360° videos to their users. Nevertheless, these new services pose numerous new issues, among it is worth to mention: (a) the standardization of new video formats; (b) the design of new algorithms to be used to select the video bitrate conforming to the *MPEG Dynamic Adaptive Streaming over HTTP* (MPEG-DASH or DASH) standard; (c) the design of compression techniques suitable for 360° videos. In particular, the last issue is very challenging since it has been shown that streaming a 360° video would require a network bandwidth of approximately 400 Mbps to deliver a video quality similar to that of a full high-definition (HD) resolution two-dimensional (2D) video.[1]

Recently, different approaches have been proposed to reduce the required network bandwidth to stream the content which are summarized in the following. One popular design strategy employed today is to stream to the user only a portion of the video, the one falling in the current user's field of view (FoV), that is, the region of interest (RoI). One way to implement this approach is using the *slicing* technique which divides the video into several portions which are encoded and stored separately in different bitstreams. The advantage of this approach stems from its implementation simplicity. The drawback is that a RoI may span multiple slices, each one requiring one decoding process running on the client device. Consequently, this solution cannot be easily implemented in mobile devices. Moreover, the client has to parallelly download the slices composing the RoI, making the adaptive streaming algorithm considerably more complex.

A new approach not requiring separate decoding process is *tiling,* a concept which has been introduced in High Efficiency Video Coding (HEVC) and recently also considered for DASH-compliant video delivery systems.[2,3] This approach requires the video to be spatially divided into several *tiles* which are encoded independently and possibly stored into a single bitstream. Spatial relationship between different tiles can be embedded into bistream using either (a) the Omnidirectional Media Application Format [4] extension or (b) integrated into the Media Presentation Descriptor employing Spatial Relationship Descriptors. Either way, the client can decide to request a subset of the available tiles (the ones composing the RoI) and a single process is able to decode the received compressed bitstream. However, tiling does not allow varying the resolution of the representations, but only their bitrate. As a consequence, the resolution of each tile must remain constant, that is, the tile grid cannot change across representations.[2] Another limitation is that tiling efficiency decreases when increasing the number of tiles.[5] Most importantly, when sudden changes of the viewpoint occur, video segments of new tiles should be quickly downloaded and rebuffering events might occur in the case those segments are not downloaded in time. Finally, encoders supporting tiling are still at the experimental stage while hardware decoding of these new formats is not widely available in the mobile ecosystem yet.

A different class of approaches leverages a *scaling technique* to reduce the bitrate to encode the projected 360° video.[6,7] In Reference 6, the authors propose a technique exploiting the downscaling operation to realize a specific mixed-resolution packing for 360° video streaming. The bitrate reduction here is achieved by varying the used quantization parameter to encode each tile, while the downscaling operation is exploited to rearrange some portion of the 360° video properly. In Reference 7, a Gaussian pyramid projection mapping technique is conceived to provide viewport-adaptivity. Once the RoI is identified, the Gaussian pyramid is implemented by halving the resolution recursively in the areas around the RoI.

In this letter, a scaling technique to reduce bandwidth requirements to stream omni-directional videos is presented, having the following main features: (a) it achieves bitrate reductions by aggressively reducing the horizontal resolution of the areas outside the main RoI; (b) it employs an approach that is encoder-agnostic; (c) it can be easily adopted using standard technologies already available in the vast majority of mobile platforms and devices.

Differently from the approach proposed in Reference 7, in this letter the bitrate reduction is obtained by applying downscaling homogeneously to the regions outside the RoI. Such an approach makes the implementation considerably simpler, a particularly important aspect to empower live streaming of 360° videos.

2 | PROPOSED APPROACH

The de facto standard employed today in the industry is the MPEG-DASH which allows clients to dynamically adapt the video bitrate to the time-varying network bandwidth. The video content is stored on a standard HTTP server and a client fetches the video by employing an HTTP connection. The video content is encoded at different bitrate *levels* (or *representations*) which form the *video levels set* $\mathcal{L} = \{l_1, l_2, \ldots, l_M\}$ with $l_i < l_{i+1}$.[8] At the client, a control algorithm dynamically selects the video level to be streamed at each segment download.

The content generation in the case of omni-directional videos differs significantly from the one employed for classical 2D videos. In particular, 360° cameras capture a spherical scene (the omni-directional video) that need to be projected onto the 2D plane (the projected video) in order to be encoded. It is worth to mention that only a small portion (roughly one-sixth of the video resolution) of the projected video falls in the users' viewport, that is, the part of the video which is currently visualized by the user. To make a concrete example, in order to deliver a video content with a viewport resolution of 1080p, the video resolution of the projected video has to be larger than 6480p that is a resolution larger than 8K ultra HD. Indeed, the encoding of such a large resolution video at high quality might result in a too large video bitrate. Consequently, streaming the encoded projected video at full resolution entails a remarkable waste of network bandwidth.

The idea to reduce the video bitrate is that only the parts of the scene that are considered to be a RoI should be encoded at a high quality. In this letter, RoI has been defined as the portion of the video falling in the current user's viewport at a given time. As such, the RoI changes over time and depends on the scene and on the user's behavior during the video playback.

In a nutshell, the idea is to generate from one projected video a number N of versions, the *views*, that encode RoIs at different positions. All the views constitute the *views set* $\mathcal{V} = \{v_1, \ldots, v_N\}$. Now each view $v_i \in \mathcal{V}$ is encoded into M video representations at different bitrates l_j (and resolutions) forming the video *level set* $\mathcal{L} = \{l_1, \ldots, l_M\}$. At the end of this procedure, the DASH Server stores and indexes a set of representations $\mathcal{R} = \mathcal{V} \times \mathcal{L}$ composed of $N \cdot M$ files. In the following, details of the methodology proposed to produce the views v_i will be provided.

Without loss of generality, the original uncompressed scene has been considered being produced in Equirectangular Projection (ERP) format, which is by far the most popular output format for 360° cameras. Notice that this is not a limitation since any other format is in principle supported by using format adapters filters.* In order to produce the different views v_i it is required to first manipulate the original ERP video. To the purpose, the RoI has been identified as the *spherical lune* (a slice of the sphere)

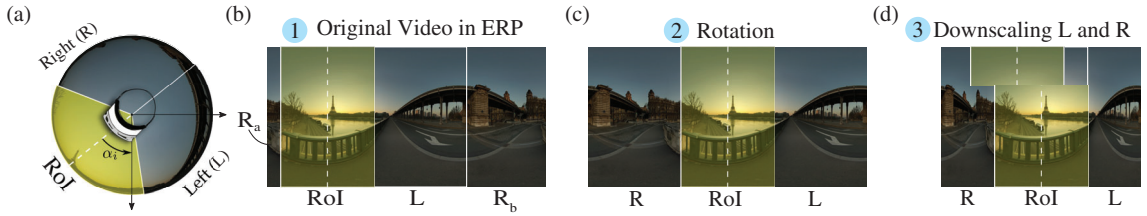

FIGURE 1 Approach to generate the *i*-th RoI representation

with a dihedral angle (ie, the FoV angular width) equal to 120°, centered at a particular yaw angle α_i. Let us consider Figure 1A that shows a user seen from above (the user is considered to be in the center of the sphere) with his head turned left so that his FoV is centered at a certain yaw angle α_i which falls into a specific RoI (the shaded area). The regions outside the RoI, namely the ones at its left (L) and its right (R), are divided into two spherical lunes of equal dihedral angle. Since the video is represented in ERP format, each spherical lune maps to a particular vertical strip of the video as shown in Figure 1B. The video in ERP format is manipulated in such a way that the RoI is always placed in the center of the frame as Figure 1C shows. The idea is to downscale the portions of the video outside the RoI, which are less likely to be in the user's FoV, to reduce the required encoding bitrate as shown in Figure 1D.

The choices made for the design of our strategy are motivated in the following. First, the RoI has been identified as the spherical lune because more complex strategies (such as ones employing spherical sectors[7,9]) may introduce inefficiencies into the intra-frame operations, leading to higher bitrate requirements.[2,10] Moreover, maintaining the RoI at the center of the frame—applying a rotation before downscaling—allows to better exploit the motion compensation algorithm by keeping the continuity between the scaled and nonscaled areas.[2,10] Finally, the usage of the *downscaling* operation—instead of *HEVC tiling*—is due to the fact that this technique is (a) independent of the employed codec, (b) can be efficiently handled by hardware decoders at the client-side, (c) can use well-established algorithms (interpolation, filtering, etc.) to improve the resulting video quality.

3 | METHODOLOGY

The content generation mechanism proposed in Section 2 has been implemented using a filter chain using FFMPEG[††]. A video catalog composed of 10 benchmark videos having a 4K resolution (ie, 3840×2048, 30 frames per second) has been produced. The videos were selected to produce a catalog sufficiently representative of different video categories and features. To investigate the relationship between the obtainable bitrate reduction and the resulting video quality, each video in the catalog has been encoded using the reference FFMPEG H.264 encoder (*libx264*[†††]) with a *Constant Rate Factor* (CRF) parameter equal to 20 before applying the downscaling algorithm.

As described in Section 2, for each view $v_i \in \mathcal{V}$ the regions outside the RoI are downscaled in order to reduce the encoding bitrate. The *downscale factor d* is defined as the ratio between the width of the downscaled video and the original video width w, that is, $d = (2w_d + w_{RoI})/w$, where w_d is the width of the downscaled regions outside the RoI and w_{RoI} is the width of the RoI. Since the catalog is composed of video with a resolution equal to 3840×2048, the resolution of each of the three vertical strips in which the video is divided (left, RoI, right) is equal to 1280×2048. The downscaled width w_d varies in the set {240px, 480px, 720px, 1080px}. The Group of Pictures parameter has been set equal to 60 frames, a typical setting commonly used by video streaming services. In order to quantitatively assess the video quality between the manipulated video (upscaled to the original resolution) and the original video in the benchmark video catalog, the structural similarity (SSIM) FFMPEG filters has been employed.

4 | CONSIDERED SCENARIOS

The experimental evaluation has been carried out into two scenarios: (a) the CRF *scenario*, in which the encoding parameters have been set to constant video quality, aiming at showing the impact of the conceived mechanism on the overall video quality; (b) the average bitrate (ABR) *scenario*, in which the encoding parameters have been set to produce a constant ABR, to explore the relationship between the proposed mechanism and video quality with a constraint on average video bitrate.

The workflow of the CRF scenario is described in details in Algorithm 1. In a nutshell, for each video, view, and considered downscale factor, the procedure described in Section 2 is carried out to produce downscaled video versions. For each downscaled

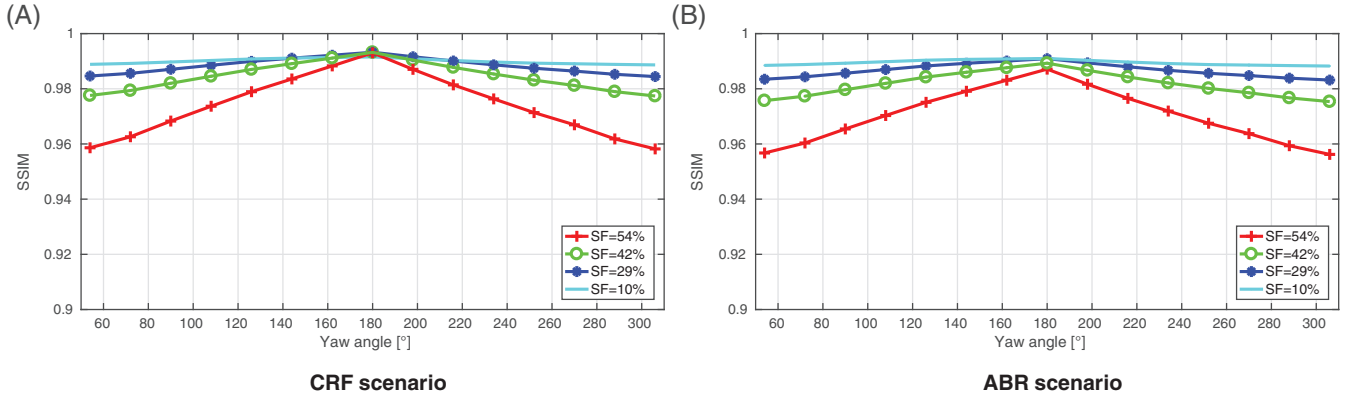

FIGURE 2 Average structural similarity (SSIM) function of the viewport yaw angle. A, CRF scenario. B, ABR scenario

video, the visual quality has been estimated by using the well-established SSIM metric, while the ABR has been measured in bit/s. At the end, for each downscale factor d the ABR reduction factor has been derived as $\hat{r}_d = \mathbb{E}[r_d^{(i)}]$.

Algorithm 1 Pseudo-code of CRF scenario

1: **for** each video **do**
2: **for** i=0 to N **do**
3: Generate i-th view v_i
4: Transcode the i-th view with CRF=20
5: Measure the average bitrate $b_o^{(i)}$
6: **for** each downscale factor d **do**
7: Downscale view v_i at factor d
8: Encode downscaled video with CRF=20
9: Measure the average bitrate $b_d^{(i)}$ and compute the bitrate reduction factor $r_d^{(i)} = b_d^{(i)}/b_o^{(i)}$
10: Upscale to the original resolution and measure $\text{SSIM}_d^{(i)}$ function of viewport yaw angle
11: **end for**
12: **end for**
13: **end for**

In the ABR scenario, the ABR reduction factors $\hat{r}_d$ have been employed to set for each view i a target ABR $\overline{b}_d^{(i)}$ equal to $(1 - \hat{r}_d)b_o^{(i)}$. The same workflow described in Algorithm 1 has been used but, instead of encoding the video using a CRF (line 8), the encoder has been set in ABR mode with a target bitrate $\overline{b}_d^{(i)}$. Such an approach has been considered because DASH systems typically produce video content by encoding videos in ABR mode to limit bitrate fluctuations.

5 | RESULTS

In the CRF scenario, the estimated video quality at different viewport yaw angles has been investigated for each considered downscale factor d. In CRF mode, the encoder is free to vary the output bitrate in order to reach a given video quality. As mentioned in the previous section, in this scenario a CRF equal to 20 has been considered.

Figure 2A shows the estimated video quality averaged over the considered videos. The figure shows that video quality is maximal at the center of the viewport (yaw angle equal to 180°) and it gracefully decreases when the angular distance from the center increases. Moreover, as expected, the slope of the video quality curves gets steeper when the scaling factor (SF) increases (ie, when the downscaled resolution decreases). Thus, when the SF is higher, the video quality degrades faster when the user moves away from the center of the RoI. Table 1 reports the ABR reduction $\hat{r}_d$ (expressed in percentage) measured for each of the considered downscale factors. The results show that the proposed approach provides a percentage bitrate reduction scaling almost linearly with the downscale factor d. Notice also that confidence intervals are quite tight, indicating that the proposed scheme is not content sensitive.

TABLE 1 Average bitrate reduction (with 95% confidence interval reported in the parentheses) for the considered downscaled resolutions in the case of CRF = 20

Downscale factor *d* (%)	Downscaled resolution (px)	Average bitrate reduction (%)
54	240	51.3 (48.0-54.7)
42	480	37.49 (34.1-40.8)
29	720	25.44 (21.9-28.9)
10	1080	7.90 (3.0-12.7)

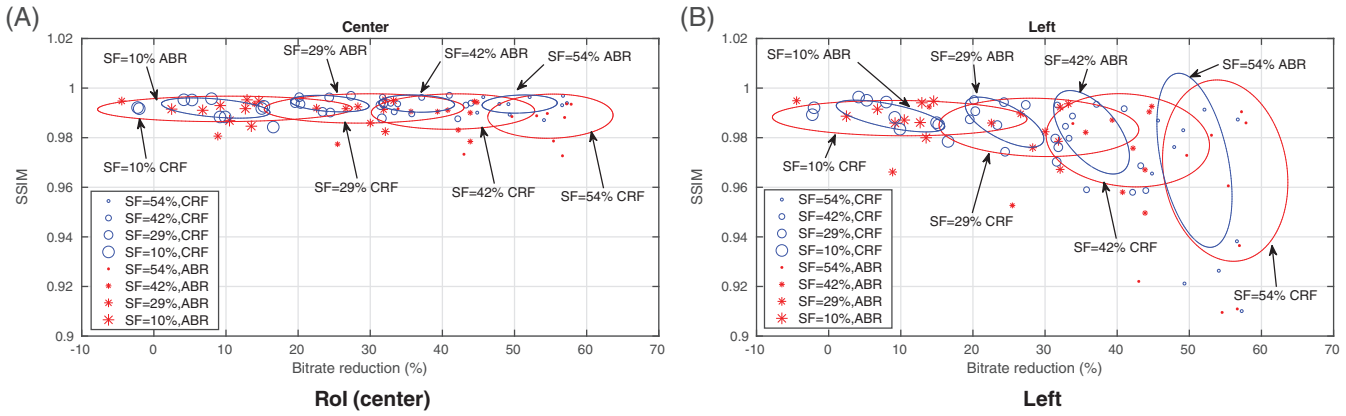

FIGURE 3 SSIM as a function of the bitrate reduction percentage. A, RoI (center). B, Left

In the ABR scenario, the impact on the video quality of the proposed bitrate reduction mechanism has been evaluated when a bitrate constraint is added as described in the previous section. Figure 2B shows that driving the encoder in ABR mode produces a smooth quality transition between lateral and RoI regions, gracefully degrading the video quality. Compared to the previous scenario, the video is only slightly affected. In particular, it has been found a maximum video quality loss in term of SSIM of around 0.005. This result indicates that the proposed content generation scheme performs satisfactorily also when the encoder is driven using a target ABR.

Figure 3A and B show scatter plots of the obtained SSIM against the measured bitrate reduction respectively when the yaw angle is at the center of the RoI or at the left[§§]. Each data point represents the obtained (bitrate reduction, SSIM) for one video of the catalog encoded at a particular SF and encoding strategy (CRF [° marker] or ABR [* marker]). In the figure, 90% confidence ellipses are reported for each considered (SF, encoding strategy) couple. Best results are obtained in the top right region of the figures (high SSIM and high bitrate reduction). Moreover, the smaller the confidence ellipses the more the strategy is insensitive to video diversity. Figure 3A shows that when the user's viewport at the center of the RoI, SSIMs are always very high and close to the maximum. However, the higher the SF, the higher the bitrate reduction, both in the case of the CRF and ABR cases. Let us now focus on Figure 3B showing the results when the user's viewport is at the left of the RoI, that is, where the video has been downscaled.

Results show that the maximum SF (52%) for both ABR and CRF provides the best results in terms of bitrate reduction, but the consequent SSIM degradation is not negligible. Moreover, confidence ellipses are larger in the SSIM direction which indicates a higher sensitivity to video content. The best overall results are obtained for a SF equal to 41% both for ABR and CRF: in particular, bitrate reductions are comparable to the ones obtained with SF of 52%, but the resulting SSIM is larger and confidence ellipses are considerably tighter. Therefore, the obtained results suggest that a SF equal to 42% provides the best trade-off between visual quality and bitrate reduction.

6 | CONCLUSIONS

This letter proposed a codec-agnostic scheme to reduce network bandwidth requirements for immersive video streaming applications. In particular, the idea is to downscale the areas outside the RoI, that is, outside the current user's FoV, and to encode the resulting video. Then, the client decodes the video and upscales the video to the original resolution. The proposed scheme has been experimentally evaluated in order to measure the obtainable bitrate reductions. Moreover, the visual quality degradations

due to downscaling have been estimated through the SSIM metric. Results show that it is possible to obtain bitrate reductions up to 50% while gracefully degrading the visual quality of regions of the video falling outside the RoI.

NOTES

* https://trac.ffmpeg.org/wiki/RemapFilter

† https://ffmpeg.org/ffmpeg-filters.html

†† https://trac.ffmpeg.org/wiki/Encode/H.264

§ Results obtained for the region at the right of the RoI are very close to those shown in Figure 3B and are omitted due to space constraints.

ORCID

Luca De Cicco https://orcid.org/0000-0002-8900-175X

REFERENCES

1. Mangiante S, Klas G, Navon A, GuanHua Z, Ran J, Silva MD. VR is on the edge: how to deliver 360 videos in mobile networks. Paper presented at: *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*. Los Angeles, CA: ACM; 2017: 30–35.
2. Concolato C, Le Feuvre J, Denoual F, et al. Adaptive streaming of HEVC tiled videos using MPEG-DASH. *IEEE Trans Circuits Syst Video Technol*. 2018;*28*(8):1981-1992. https://doi.org/10.1109/TCSVT.2017.2688491.
3. Hosseini M, Swaminathan V. Adaptive 360 VR video streaming: divide and conquer. Paper presented at: 2016 IEEE International Symposium on Multimedia (ISM); San Jose, CA; 2016:107–110. https://doi.org/10.1109/ISM.2016.0028.
4. Skupin R, Sanchez Y, Wang YK, Hannuksela MM, Boyce J, Wien M. Standardization status of 360 degree video coding and delivery. Paper presented at: 2017 IEEE Visual Communications and Image Processing (VCIP). Saint Petersburg, FL: IEEE; 2017:1–4.
5. Sanchez Y, Skupin R, Schierl T. Compressed domain video processing for tile based panoramic streaming using hevc. Paper presented at: 2015 IEEE International Conference on Image Processing (ICIP); Quebec City, Canada; 2015:2244–2248. https://doi.org/10.1109/ICIP.2015.7351200.
6. Zare A, Aminlou A, Hannuksela MM. 6K effective resolution with 4K HEVC decoding capability for OMAF-compliant 360-degree video streaming. Paper presented at: *Proceedings of the 23rd Packet Video Workshop, PV '18*; Amsterdam, The Netherlands; 2018:72–77. https://doi.org/10.1145/3210424.3210425.
7. Hristova H, Corbillon X, Simon G, Swaminathan V, Devlic A. Heterogeneous spatial quality for omnidirectional video. Paper presented at: 2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP); Vancouver, Canada; 2018:1–6. https://doi.org/10.1109/MMSP.2018.8547114.
8. De Cicco L, Mascolo S, Palmisano V. Feedback control for adaptive live video streaming. Paper presented at: *Proceedings of 2nd ACM Conference on Multimedia Systems, MMSys '11*; San Jose, CA; 2011:145–156. https://doi.org/10.1145/1943552.1943573.
9. Corbillon X, Simon G, Devlic A, Chakareski J. Viewport-adaptive navigable 360-degree video delivery. Paper presented at: 2017 IEEE International Conference on Communications (ICC); Paris, France; 2017:1–7. https://doi.org/10.1109/ICC.2017.7996611.
10. Youvalari RG, Aminlou A, Hannuksela MM, Gabbouj M. Efficient coding of 360-degree pseudo-cylindrical panoramic video for virtual reality applications. Paper presented at: 2016 IEEE International Symposium on Multimedia (ISM); San Jose, CA; 2016:525–528. https://doi.org/10.1109/ISM.2016.0115.

How to cite this article: De Cicco L, Mascolo S, Palmisano V, Ribezzo G. Reducing the network bandwidth requirements for 360° immersive video streaming. *Internet Technology Letters*. 2019;2:e118. https://doi.org/10.1002/itl2.118