IBC2023

BEST OF BOTH WORLDS - VVC ENABLES OPEN-GOP CODING IN ADAPTIVE STREAMING WORKFLOWS

M. Alvarez-Mesa[1], B. Bross[2], C. C. Chi[1], C. Feldmann[3],
S. Sanz-Rodriguez[1], A. Wieckowski[2]

[1] Spin Digital, DE and [2] Fraunhofer HHI, DE and [3] Bitmovin, AT

ABSTRACT

Over-The-Top adaptive streaming technology has become a popular method for delivering high-quality video content over the internet, adjusting the video quality based on the user's internet connection speed and device capabilities. It uses multiple bit-rates encoding of video content, where the video is divided into smaller segments of varying bit-rates and resolutions. Due to codec constraints, the segments had to be coded in a so-called closed GOP configuration while in broadcast, a more efficient open GOP is widely used. The emerging Versatile Video Coding (VVC) standard allows the use of the more efficient open GOP coding approach in adaptive streaming as well. In this paper, the integration of open GOP coding in a cloud transcoding and a live encoding adaptive streaming application are described and discussed. In addition, an informal subjective test confirmed the benefits of the proposed open GOP technique, showing that subjective quality improves considerably compared to closed GOP coding.

INTRODUCTION

Random access points (RAPs) are very important in video entertainment applications. They refer to the specific points within a coded video stream where a viewer can begin playback without having to wait for the entire stream to load. This is particularly important in broadcast to tune-in or switch channels as well as in adaptive streaming, where video streams are often divided into smaller segments and delivered dynamically based on the viewer's bandwidth and device capabilities.

In video coding, a group of pictures (GOP) define hierarchical referencing structures between RAPs. A RAP is always characterised by an intra-picture predicted frame and modern video codecs often use multiple GOPs in between. To avoid confusion, this paper uses the term GOP for these smaller groups and the term intra period to refer to the distance between two RAPs. Traditionally, GOPs at RAPs were "closed", i.e. the inter-picture prediction of a codec cannot reference pictures from GOPs before the RAP. This reduces the coding efficiency because it restricts the temporal redundancies to be exploited. More recent standards, e.g. High Efficiency Video Coding (HEVC), facilitate so-called "open" GOP coding for higher compression efficiency. In the broadcast world, open GOPs are already widely used. In the adaptive streaming world, closed GOPs are used for random access as well as for switching a rendition e.g., spatial resolution or bit-rate. When switching spatial resolutions, open GOP inter-picture referencing between GOPs is prohibited by legacy codecs as spatial scaling is required. The most recent Versatile Video Coding (VVC) standard introduces a functionality called reference picture resampling

(RPR) to address that shortcoming. In addition to that, VVC encoder restrictions prevent unpleasant visual artefacts, which can be caused by open-GOP resolution switching. More details on HEVC and VVC can be found in a detailed overview by Bross et al (1).

In this paper, we first review the state-of-the-art in adaptive streaming using closed GOPs and their shortcomings. After that, a short description of how VVC enables open GOP adaptive streaming using reference picture resampling and certain encoder constraints is given. Before concluding this paper, we describe the benefits and challenges of integrating open-GOP encoding in a highly scalable cloud transcoding solution as well as in a live encoding workflow.

STATE-OF-THE-ART IN ADAPTIVE STREAMING

Adaptive streaming, e.g. with HTTP Live Streaming (HLS) or Dynamic Adaptive Streaming over HTTP (DASH), and state-of-the-art video codecs is restricted to closed GOPs. In this section, we explain the rationale behind that restriction, differences between open and closed GOP and their relation to the intra period for random access points. After that we discuss the shortcomings of closed GOP adaptive streaming.

Open GOP and Closed GOP

Modern video compression standards since MPEG-2 allow the encoding of video pictures either by exploiting spatial redundancy with intra-picture prediction (**I**) or by exploiting both spatial and temporal redundancy with inter-picture prediction. Inter-frames can be either predictive (**P**), using data from one previously decoded picture for temporal prediction of a block, or Bipredictive (**B**), averaging data from up to two previously decoded pictures.

The frequency with which the I pictures are inserted in a video bitstream is referred to as the **intra period**. RAPs are typically created using I-frames, which can be coded independently and allow the decoder to start decoding the video sequence. The intra period is defined by the application. For broadcast, the intra period is typically set to 1 second to minimise tune-in and channel switching latency. However, for streaming applications, a longer Intra Period of 2 to 4 seconds can be used to improve compression efficiency, reducing the amount of data needed to deliver high-quality video content.

Schwarz et al (2) have shown that re-arranging pictures into a so-called **GOP** to obtain a hierarchical referencing structure can provide some significant coding efficiency gain. Figure 1 shows an example of two such GOPs with 8 pictures. No picture from GOP #2 references a picture from the previous GOP #1. This break of the temporal referencing structure is referred to as **closed GOP**. In case of random access, the last I picture and all previous B pictures from GOP #2 can be decoded and displayed independent of the previous pictures from GOP #1.

Figure 2 on the other hand illustrates an **open GOP** structure. Here the pictures from GOP #2 reference pictures from GOP #1 that precedes the associated I picture. In case of random access at the last I picture, all following B pictures in coding order (10-16) need to be skipped because they depend on the previous pictures (0-8).

To signal that, HEVC introduced different picture types (1). For closed GOP, the instantaneous decoding refresh (IDR) picture indicates that the decoder is reset including the decoded picture buffer and the random access decodable leading (RADL) picture type is used to mark pictures that do not reference pictures that precede the associated IDR picture in output order, i.e. are decodable in case of random access (Figure 1). For open

GOP, the constraint random access (CRA) picture type retains the reference pictures in the decoded picture buffer and the random access skipped leading (RASL) picture type is used to mark pictures that do reference pictures that precede the associated CRA picture in display order, i.e. needs to be skipped in case of random access (Figure 2). VVC uses the same picture types to signal closed and open GOP.

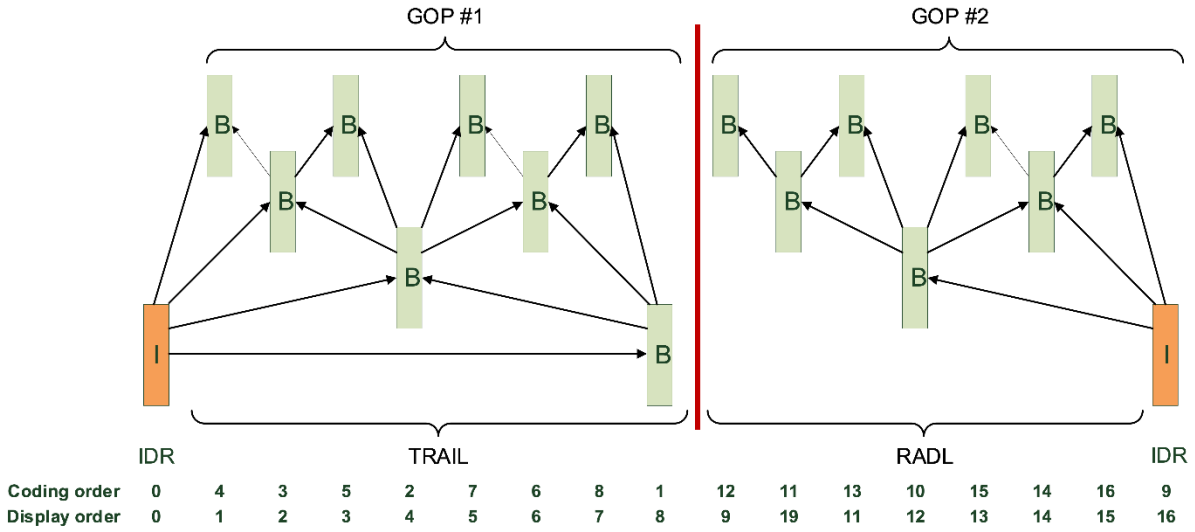

Figure 1 – Closed GOP coding structure with instantaneous decoding refresh (IDR) intra (I) picture and random access decodable leading (RADL) inter (B) pictures.

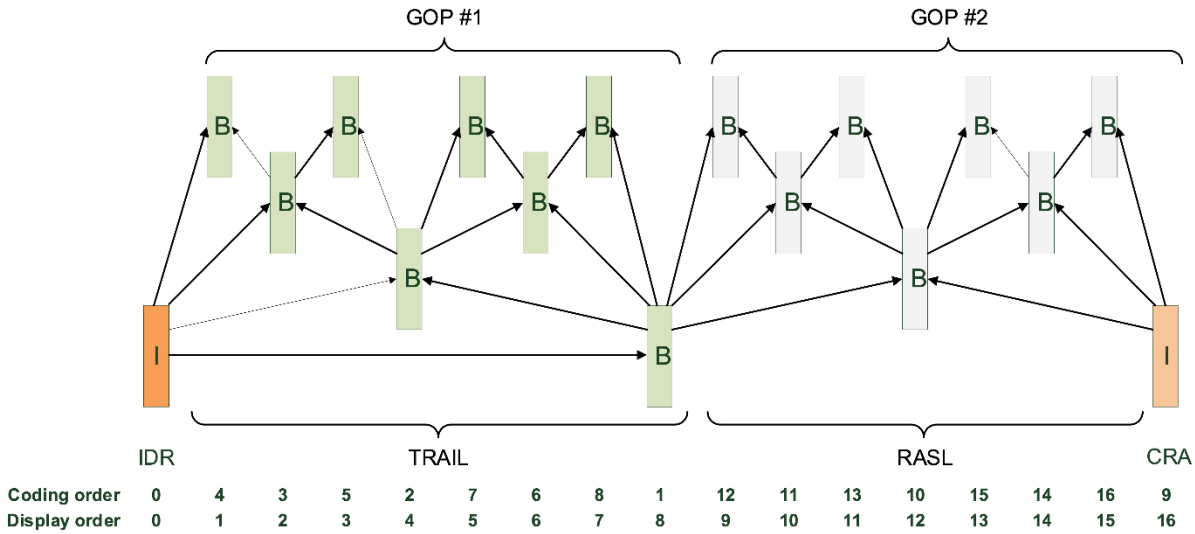

Figure 2 – Open GOP coding structure with constraint random access (CRA) intra (I) picture and random access skipped leading (RASL) inter (B) pictures.

Shortcomings of Closed GOP Adaptive Streaming

Shortcomings of the closed GOP approach for adaptive streaming have been analysed by Skupin et al (3). In terms of objective performance, it is reported that using open GOP can

provide 8.5% and 2% bit-rate savings for the same Peak Signal to Noise Ratio (PSNR) for 1s and 4s segments length respectively.

The reported bit-rate savings by using open GOP are based on PSNR values averaged over all pictures. However, in a closed GOP structure, the errors are not equally distributed throughout the video but mostly concentrated at the random access switching points, i.e. around the intra pictures. This can lead to unpleasant artefacts, i.e. a so-called temporal pumping effect can be observed at switching point. This is because closed GOP breaks the motion-compensated prediction which results in different distortion patterns. This is even worse when switching to a higher quality rendition.

VVC IMPROVING ADAPTIVE STREAMING WITH OPEN-GOP ENCODING

In the previous section, the benefits of the open GOP approach as well as the reason why it is not used in adaptive streaming have been explained. In order to enable open GOP at switching points in adaptive streaming, the most recent VVC standard comprises a tool named reference picture resampling (RPR). This, in combination with encoder constraints, enables open GOP in adaptive streaming as detailed in the following subsections.

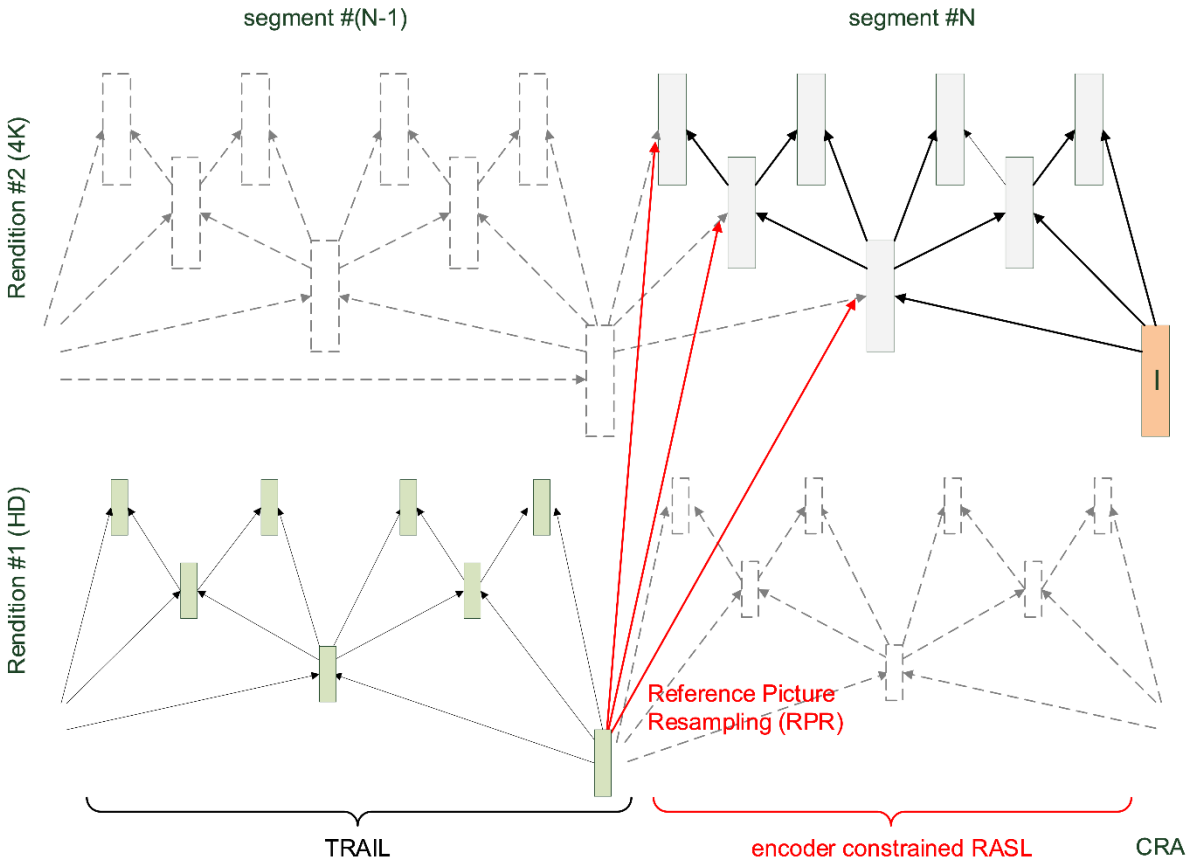

Figure 3 – Open GOP coding structure in adaptive streaming enabled by reference picture resampling (RPR) and constrained random access skipped leading (RASL) pictures.

Reference Picture Resampling in VVC

In adaptive streaming, when playing a video having a specific resolution, the client can decide to switch to a higher or lower resolution if bandwidth allows or requires it. Hence the pictures in the new segment have a different spatial resolution than the ones in the previous segment. Figure 3 illustrates such a switch from high definition (HD) to 4K using an open GOP coding structure. It can be seen that the 4K pictures from segment #N reference HD pictures from segment #(N-1). In VVC this is not an issue anymore because with RPR, the standard specifies a set of resampling filters that allow to upscale the HD pictures to 4K so that they can be referenced. In order to facilitate implementation especially on hardware, the scaling factor is restricted to be larger than or equal to 1/2 (2 times downsampling from the reference picture to the current picture), and less than or equal to 8 (8 times upsampling). In the example of Figure 3, 2 times upsampling is used.

In the extreme case of having to switch from a high resolution to a resolution more than 2 times lower, RPR cannot be used because of the aforementioned restriction. Hence, having a closed GOP fallback rendition for very low resolutions is required. However, the significant drop in bandwidth in this case would have such a severe impact on visual quality that open GOP would not bring any benefit anyway. A detailed description of the VVC high-level syntax including RPR is given by Wang et al (4).

VVC Constraints for Resolution Switching

While RPR in VVC solves the problem of referencing pictures from previous renditions having a different spatial resolution, one issue still remains. After switching a rendition with open GOP, the decoder-side reference picture differs from the encoder side. This drift occurs only in RASL pictures and it can cause visually unpleasant artefacts. These stem from coding tools that are sensitive to changes in the reference samples like decoder side motion vector refinement. One way to prevent this is constraining an encoder not to use these tools. Furthermore, these restrictions only apply to the RASL pictures as depicted in Figure 3 and it has been shown that the overall impact on coding efficiency is marginal with a worst case bit-rate increase of 0.65% (3).

In VVC, a constrained RASL encoding indication SEI message is defined to inform packagers, e.g. for DASH or HLS, that this set of encoding constraints is applied. That way, they can expect bitstream switching between bitstreams that represent the same source video content with fewer visually noticeable or visually annoying artefacts in the reconstructed sample values of the RASL pictures.

CLOUD TRANSCODING USE CASE

In this first use case we focus on a Video on Demand (VoD) use case where the full input is available offline at the time the encoding is started. This allows the Bitmovin encoder to operate on multiple parts of the input simultaneously and thus to scale horizontally across a pool of compute resources in the cloud. The Bitmovin encoder supports the open GOP encoding scheme through the integration of the VVenC VVC encoder software from Fraunhofer HHI. VVenC has been described by Wieckowski et al (5) and is publicly available on GitHub (6). The integration of this software into the Bitmovin encoding solution was already detailed by Wieckowski et al (7).

Integration with Bitmovin

The Bitmovin encoder operates on a cut and stitch approach. The overall encoding task is split up into individual encoding tasks based on the configured output format. E.g., for a

segmented DASH or HLS output with a configured segment size of 4 seconds, each of these segments represents one distinct encoding task. All encoding tasks are then scheduled and distributed over a cluster of virtual machines that run in the cloud of Amazon Web Services, Google Cloud Platform or Microsoft Azure. Subdividing the encoding into many small parts allows for excellent horizontal scalability. Furthermore, the smaller the individual tasks are, the more flexibly they can be scheduled onto the available virtual machines which allows for high utilisation of the whole cluster for the whole time of the encode which in turn minimises the overall encoding time. If the segments are long, scheduling is less efficient, and we observe a tailing effect at the later stages of encodes, where the encoding is not finished because a few longer running tasks are not completed yet.

We observed that the rate control of many core encoders performs better if the segments are longer. Longer segments also allow for lower overheads in the overall process (e.g., scheduling, upload, download, decoding, filtering etc.). With the introduction of Smart Chunking as described by Hafellner (8), we implemented a feature to merge multiple subsequent segment encoding jobs into larger tasks, which we will call an **encoding segment** (see Figure 4). The merged segments are encoded as one encoding task. The encoder is requested to place random access points where the bitstream is subsequently fragmented, multiplexed and uploaded to the output. As detailed above, the length of these segments is a tradeoff where longer segments result in improved coding performance, lower overhead, but also reduced scheduling efficiency.

For open GOP switching, Smart Chunking is a perfect fit. We can merge the encoding of multiple segments into bigger encoding segments where all but the first segment can be encoded with open GOP prediction. Since a new encoder must be started at the beginning of each encoding segment, we must begin the encode of each encoding segment with a closed GOP (with an IDR frame). As for Smart Chunking, this is a tradeoff. The longer the encoding segments are, the more GOPs are open and can benefit from higher coding efficiency and adaptive open GOP switching at segment boundaries. But longer segments also mean longer encoding time because of decreased scheduling efficiency and potentially higher cost of encoding.

Figure 4 – Example where 4 output segments are merged into one encoding segment. All but the first segment can be encoded using open GOP. Segment 5 must start with an IDR frame.

The practical implementation was straightforward as we already have VVenC integrated into the Bitmovin Encoder. The encoder for each encoding task is configured to use open GOP with the constraints for adaptive resolution switching enabled. We then enabled the Smart Chunking approach by configuring the encoder to place random access points exactly at the beginning of each sub-segment.

Experimental Results

In order to evaluate the impact of the longer encoding segments, we ran some encodings with various lengths. In the experiments, the size of each output segment is 4 seconds or 96 frames at 24 fps. We then merge 1, 4, 8 and 16 segments into encoding segments resulting in lengths of 4, 16, 32 and 64 seconds respectively. The VVenC encoder is

configured to place an IDR frame at the very beginning of the encoding segment and RAPs with constrained open GOP prediction at the start of the remaining segments. The intra period in this experiment is equal to the segment size (4s) and multiple renditions from 360p to 4k are encoded. The case with 1 segment consists of only closed GOP and serves as our reference for the measurement. The longer the encoding segments get, the more GOPs are open. For the 16s encoding segments, for example, there is one closed GOP segment followed by 3 open GOP segments. We measure the overall encoding time as well as the overall cloud cost of the encode.

Merged Segments	Relative Encoding time	Relative Cost
1	100%	100%
4	112%	107%
8	153%	119%
16	297%	188%

Table 1 – Relative cost and encoding time for short inputs (12-20 minutes) and varying number of segments merged into an encoding segment.

Table 1 shows that the overhead for the encoding can be quite significant for short encodings of 12-20 minutes. For the case of 16 merged segments, the overall encoding time almost triples. While the impact on the overall cost is lower, it is also increasing. For long inputs (60 minutes), we were able to observe very similar results for the relative encoding time while the relative cost increase is a bit lower.

In conclusion, while open GOP adaptive streaming can be easily enabled in the cloud transcoding use case, there is a tradeoff between the number of segments that can benefit from it and the ability to scale the transcoding horizontally.

LIVE ENCODING USE CASE

This second use case focuses on live applications where the video is encoded in real-time and delivered over the open Internet. Adaptive streaming allows multiple end users with various device and network capabilities to access the content uninterruptedly. Unlike offline VoD applications where horizontal (e.g. segment-level) parallelization is possible, in the live streaming case the encoder has to rely on fine-grain parallelization (e.g. wavefronts, tiles, or frame-level) to be able to ensure real-time operation at the target frame rate and, at the same time, to satisfy low latency constraints (typically 1 to 2 seconds).

In addition, it is desirable that the encoder produces all the renditions in real-time on a single server, as this facilitates time synchronisation and packaging. In the overall live workflow the encoder receives the video and audio signals from the live production via SDI or TS-over-IP, encodes them in real-time with multiple renditions on a single server, packages the resulting bitstreams in HLS or DASH, and sends them to the CDN for final delivery to the end users.

Spin Digital has developed a VVC software encoder for UHD live adaptive bit-rate (ABR) applications that fulfil all above-mentioned requirements: live encoding in VVC for multiple renditions in a single server, low-latency encoding, packaging in HLS or DASH, and delivery to a CDN, Spin Digital (9).

VVC for Live ABR with Open GOP

Spin Digital Live VVC encoder has been enhanced with open GOPs and the proposed constraints to reduce error propagation at the switching points (3).

We performed some experiments to assess the impact of this new technique, where we used Spin Digital VVC live encoder to encode 11 4K-UHD 60 fps video clips in constant QP mode with QPs between 24 and 38. The GOP size (i.e., number of RASL pictures) was set to 16 frames and the intra period ranged from 16 to 128 frames. It should be noted that in live ABR streaming it is common to set the intra period with the same size as the segment size, and to define a relatively small segment size (2 to 4 seconds) in order to reduce the end-to-end latency. In live ABR streaming the segment size is a tradeoff between latency and upload efficiency: larger segments can be uploaded to the origin server more efficiently (less connection overhead) but result in longer latencies, while shorter segment sizes result in shorter latencies but are less efficient to deliver.

As can be seen in Table 2, constrained open GOP encoding results in minimal BD-rate losses (up to 0.10%) with respect to open GOP regardless of the intra period used. On the other hand, closed GOP encoding produces BD-rate losses in all cases, as expected. For example, for an intra period of 64 frames the BD-rate increase is 6.21%.

Intra Period [frames]	Closed GOP	Constrained Open GOP
128	3.16%	0.02%
64	6.21%	0.08%
16	22.76%	0.10%

Table 2 – YUV-PSNR BD-rate losses of closed GOP encoding and constrained open GOP encoding, both relative to open GOP encoding, for different intra periods.

The losses in compression efficiency become more significant as the intra period decreases due to the increasing presence of closed GOPs in the encoded video. The worst case occurs when the intra period has the same size as the GOP (i.e. 16 frames), in which the BD-rate loss can even exceed 20.0%. These results are in line with those presented in (3).

Real-time performance: Live ABR encoding on a single server

The VVC live encoder has been extensively optimised for the latest generation of CPU architectures in order to achieve the performance and compression levels required for live adaptive streaming applications. As a result, the encoder is able to process simultaneously, and in a single server, multiple renditions, including UHD, with significant compression efficiency gains over HEVC.

As an example, when running on a dual-socket server with two Intel Xeon Platinum 8368 CPUs (2x38 cores), the encoder was able to process 4K (2160p), 1080p and 720p, all of them in 10-bit at 60 fps, and to package them in HLS or DASH for streaming to a CDN.

Finally, the live encoder also supports the simultaneous and aligned mixture of open GOP and closed GOP renditions, which allows the receiver to quickly switch back a low-resolution rendition as a fallback, as mentioned in the previous section on reference picture resampling.

SUBJECTIVE VALIDATION

The new functionality of the VVC encoder has been validated in an informal subjective test. In this test, four 1-minute 8K (7680x4320px, 10-bit, 60 fps) clips were utilised, more specifically: *BerlinSeqs* (a concatenation of 6 10-second clips) from Fraunhofer HHI (10) *FollowCar* and *MC2* from Poznan Supercomputing and Networking Center (PSNC), Immersify (11); and one nature content from The Explorers (12) called *Teaser 1*.

Table 3 shows detailed information about the tested clips, which also specifies their spatio-temporal characteristics in terms of spatial and temporal information (SI, TI), both defined in Recommendation ITU-R BT.500 (13).

	Producer	Type	Format	SI	TI
BerlinSeqs	Fraunhofer HHI	Footage	8Kp60 PQ	100.8 (med)	59.3 (low)
FollowCar	PSNC	Footage	8Kp59.94 SDR	150.8 (med)	113.3 (med)
MC2	PSNC	Footage	8Kp59.94 SDR	187.2 (high)	86.1 (low)
Teaser 1	The Explorers	Footage	8Kp50 PQ	101.6 (med)	52.7 (low)

Table 3 – Technical information of the 8K clips: producer, type of content, format, SI, TI.

For the sake of reproducibility of the tests and results, preconfigured renditions were created simulating a scenario with multiple resolution switches. In particular, the Spin Digital VVC live encoder was configured to use constrained open GOPs, 1-second intra period, and Constant Bit-rate (CBR) control. With these encoding settings, the video sequences were first encoded at different resolutions (2K, 4K, 5K, 6K, 8K) and bit-rates (see Table 4) and packed into HLS segments of 5 seconds each.

	2K	4K	5K	6K	8K
Resolution [pixels]	1920x1080	3840x2160	4800x2700	5568x3132	7680x4320
Bit-rate [Mbps]	7	15	20	28	38

Table 4 – Selected resolutions and bit-rates for the subjective test.

Then, several 1-minute VVC bitstreams were created by taking 12 5-second HLS segments of different resolutions (see Figure 5), so that the the RPR filter of the decoder could use the following up- and down-scaling factors in each dimension:

- Upscaling factors: 1.20, 1.25, 1.33, 1.50, 2.00, 4.00
- Downscaling factors: 1.20, 1.25, 1.33, 1.50, 1.60, 2.00, 4.00

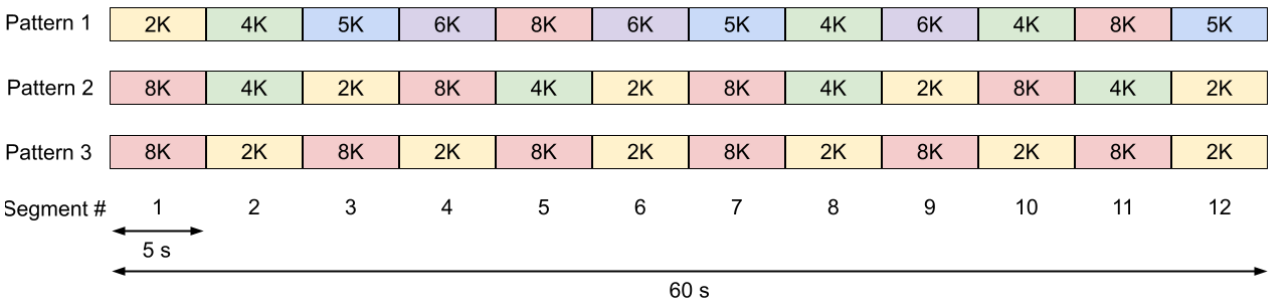

Figure 5 – Test patterns for video with different resolution switching points.

For comparison purposes, this workflow was repeated to also produce the HLS bitstreams encoded with closed GOPs.

The bitstreams were played back using the VVC decoder developed by Spin Digital, 2021 (14) and an 8K TV. The player was configured to upscale the decoded pictures to the native resolution of the TV.

The results of the subjective experiment are summarised next:

Pattern 1

In this bitstream, open GOP resolution changes with fractional RPR scaling factors were analysed. According to the subjective experiments, no visible coding artefacts were observed after the resolution changes in any of the combinations presented in the bitstream. No pumping artefacts were observed at the beginning of each intra period.

The closed GOP version showed periodic pumping artefacts, especially when the 2K rendition was displayed. It is worth noting that the pumping effect was not so critical at the resolution change, but in the subsequent intra periods.

Pattern 2

This second pattern includes resolutions that allow the RPR filter to use integer scaling factors: 2-fold downscaling and 4-fold upscaling. As in the first case, no coding errors were detected after the resolution change. Moreover, it was observed that the test clips become blurrier when switching from 8K to 4K and from 4K to 2K and sharper when changing from 2K to 8K, but without appreciable jumps in quality in the form of a pumping effect.

It was also observed that video shots with a lot of spatial detail combined with relatively little motion were more sensitive to visible transitions in resolution than others with either a lot of motion (due to temporal masking) or little spatial complexity.

Pattern 3

This third experiment mainly focused on testing the RPR feature when the video resolution changes from 8K to 2K (4-fold downscaling), in order to verify whether coding artefacts or drifts were visible when the RPR downsampling factor is higher than the maximum allowed in the VVC standard (2-fold).

According to the subjective analysis, no coding errors were observed, demonstrating that the RPR downsampling factor could be increased to 4 or even 8 without negative impact on perceived quality. This would require adjusting the filter coefficients or filter length in order to increase the fidelity of resampled pictures and thus further reduce error propagation. This update could be proposed for future profiles or extensions of the VVC standard.

CONCLUSIONS

VVC allows the improvement of adaptive streaming by enabling open GOP encoding which allows, for a certain type of pictures (RASL) around the random access point, to reference pictures from a different representation of the same content. This does not only require a new feature in VVC called reference picture resampling but also certain encoder constraints to eliminate visually unpleasant artefacts. In this paper, we describe practical implementations of open GOP adaptive streaming for two applications. One is distributed cloud encoding for video on demand, where it has been shown that open GOP encoding can be combined with a smart chunking approach. The second one is live adaptive streaming, where a real-time encoder can apply open GOP to all encoded renditions and improve compression efficiency compared to closed GOP without negative side effects. An

informal subjective test demonstrated the subjective benefits of the proposed open GOP technique, showing that subjective quality improves considerably compared to closed GOP coding, and that there are no visible artefacts associated with open-GOP at resolution switching.

REFERENCES

1. Bross, B. et al, 2021. Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC). Proceedings of the IEEE, vol. 109, no. 9, pp. 1463-1493, Sept. 2021, doi: 10.1109/JPROC.2020.3043399.

2. Schwarz, H. et al, 2006. Analysis of Hierarchical B Pictures and MCTF. Proceedings of 2006 IEEE International Conference on Multimedia and Expo, Toronto, ON, Canada, 2006, pp. 1929-1932.

3. Skupin, R. et al., 2021. Open GOP Resolution Switching in HTTP Adaptive Streaming with VVC. Proceedings of 2021 Picture Coding Symposium. pp. 1-5.

4. Wang, Y. -K. et al, 2021. The High-Level Syntax of the Versatile Video Coding (VVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 10, pp. 3779-3800, Oct. 2021, doi: 10.1109/TCSVT.2021.3070860.

5. Wieckowski, A. et al., 2021. Vvenc: An Open And Optimized Vvc Encoder Implementation. Proceedings of 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shenzhen, China, 2021, pp. 1-2, doi: 10.1109/ICMEW53276.2021.9455944.

6. VVenC Software Repository on Github, https://github.com/fraunhoferhhi/vvenc

7. Wieckowski, A. et al., 2022. VVC in the cloud and browser playback: it works. Proceedings of the 1st Mile-High Video Conference (MHV '22). Association for Computing Machinery, New York, NY, USA, 19–24. https://doi.org/10.1145/3510450.3517305

8. Markus Hafellner - Bitmovin's Smart Chunking: the evolution of the split and stitch algorithm - https://bitmovin.com/smart-chunking-encoding/

9. Spin Digital, 2023. A VVC/H.266 Real-time Software Encoder for UHD Live Video Applications. Whitepaper, February 17, 2023. https://spin-digital.com/tech-blog/whitepaper-real-time-vvc-uhd-encoder-v2/

10. Fraunhofer HHI, 2022. Reference Sequences. 3GPP, 2022: https://dash-large-files.akamaized.net/WAVE/3GPP/5GVideo/ReferenceSequences/

11. Immersify, 2018. Content & Demos - Immersify. Immersify Project Website, 2018: https://immersify.eu/content-demos/

12. The Explorers, 2021. The Earth's first Inventory in High-Definition (4K/8K HDR). The Explorers Website, 2021: https://theexplorers.com/svod

13. ITU-R, 2019. Methodologies for the Subjective Assessment of the Quality of Television Images. Recommendation ITU-R BT.500-14, Oct. 2019.

14. Spin Digital, 2021. Spin Digital Announces a VVC 8K Decoder and Media Player. Spin Digital News, Jun. 2021: https://spin-digital.com/announcements/vvc-player/