THE XVC VIDEO CODEC – A REVOLUTIONARY SOFTWARE-DEFINED VIDEO COMPRESSION FORMAT

Jonatan Samuelsson, Per Hermansson

Divideon, Sweden

ABSTRACT

Efficient video compression is a key technology component that enables high quality media services across different platforms and connection types. Several different video codecs have been used since the start of the TV digitalization and more efficient compression methods are constantly being developed. For mobile video streaming services, it is important use the network resources as efficiently as possible, but compression efficiency is not the only factor that determines which video codec is best suited to be used. There needs to be a clear and reasonable licensing scheme, the encoding complexity needs to be at a manageable level, but most importantly, the receivers must be able to support decoding of the codec. The xvc codec is a software-defined video compression format which delivers unprecedented compression performance, is available with a single reasonable license and with a light-weight decoding process that can be run in software on today's mobile phones and tablets.

INTRODUCTION

Video streaming applications are growing in volume and popularity and an increasing amount of video services are consumed over mobile networks. High performing video codecs are vital in order to utilize network capacity as efficiently as possible and to deliver the best possible quality across various connection types. Due to high and uncertain licensing costs, the HEVC codec 1 has been unable to serve this market and is so far not deployed in many mobile streaming applications. Some are looking to completely royalty-free alternatives such as AV1 2 to fill this gap, but the licensing situation around AV1 is unclear and due to technical compromises it is also unclear if AV1 will be able to match the compression performance of HEVC at reasonable computational complexity levels. A different approach, which was first introduced during the IBC week 2017, and which is gaining more and more interest in the media streaming industry is the xvc codec. The xvc codec is software-defined and has been designed to enable efficient software encoders and decoders. The publicly available xvc reference software includes an xvc decoder, capable of realtime decoding of FullHD video on smartphones and tablets. The software-focused approach makes it possible to deploy enhancements and novel compression tools on devices already in the market today, without having to wait for years until new decoding hardware is available and then wait for more years until the consumer base has swapped out their legacy devices. This paper presents background information on the state of video codec licensing and lays out a description of the xvc licensing framework, constructed to significantly improve the current video codec licensing situation. The paper provides technical details on of how the xvc codec is constructed to enable an efficient, flexible and extendable software compression system. By the end of this paper, results are provided, reporting on the bitrate savings offered by xvc relative to HEVC and AV1.

BACKGROUND

The Advanced Video Coding standard (AVC, also known as H.264) 3 was developed in a joint project between MPEG (ISO/IEC JTC 1/SC 29/WG 11) and VCEG (ITU-T Q.6/SG 16), and after it had been released in 2003, it became successful and widely deployed within just a few years. Most of the patent holders made their standard essential patents available through the MPEG LA patent pool and the royalty rates were set at a level that enabled the industry to add support for AVC in different video based applications, services and devices. When the High Efficiency Video Coding standard (HEVC, also known as H.265) was released in 2013, as a result of a new collaboration between the same standards groups, many expected a similar arrangement and yet another widely-spread, successful codec. The reality is that adoption of HEVC has been quite modest, especially in the mobile video streaming area, and for HEVC there is not just one, but three patent pools, and a large number of patent holders that do not make their patents available through the patent pools. Figure 1 shows an illustration of some of the organizations that have declared to have patents which would be required to be used to implement HEVC. The figure is based on public information available from the patent databases of ISO 4, IEC 5, and ITU-T 6 and the patent pools of MPEG LA 7, HEVC Advance 8, and Velos Media 9. It should be noted that the figure only includes a subset of the organizations that have made declarations, and it does not account for the possibility that some HEVC essential patents may have been transferred from one organization to another.

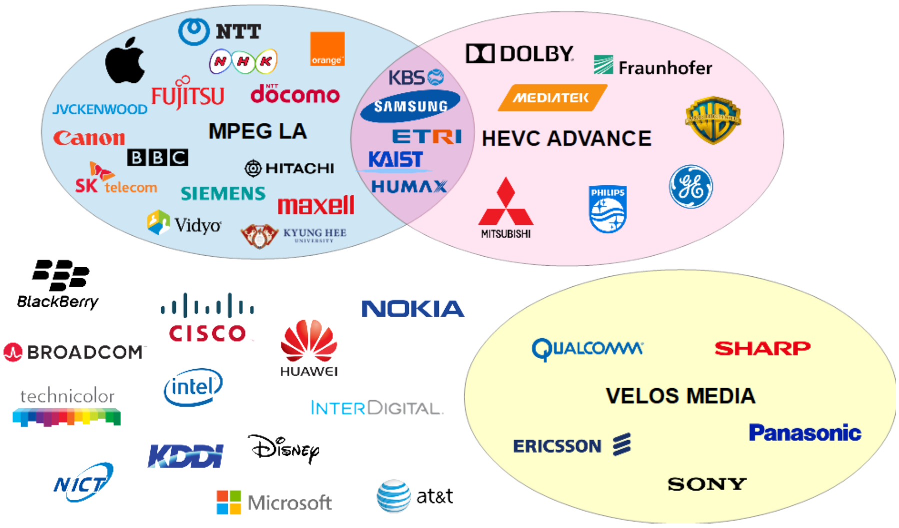

Figure 1 – A subset of the organizations that have declared to hold HEVC essential patents.

Alliance for open media and AV1

In September 2015, the Alliance for Open Media (AOM) was formed with Amazon, Cisco, Google, Intel, Microsoft, Mozilla and Netflix as founding members 10. The focus has been to deliver a high performing, royalty-free video codec and it is well understood that the formation of AOM to a large extent was a reaction to the expensive and uncertain licensing conditions associated with HEVC. The AV1 codec was built using Google's VP10 codec as a starting point (which was an unfinished extension of VP9) and with technology added from Mozilla's Daala codec and Cisco's Thor codec. Since the formation of AOM, several additional organizations have joined the group, including Apple, Arm, Facebook, IBM and Nvidia. The AV1 codec was completed and released in April 2018 11. The AV1 codec has been constructed with the ambition of not infringing on patented technology for which the patent holder would not be willing to make those patents available under royalty-free licensing terms. Existing, known methods have been avoided or redesigned in order to keep AV1 royalty free. How successful the AV1 codec eventually will become might to a large degree depend on the answers to two very fundamental questions: Has the AOM development team been able to successfully avoid existing video coding patents? And how much has that avoidance cost in terms of compression performance and computational overhead? The results presented in this paper may provide a little bit of guidance for the latter of those two questions.

The xvc video codec

The xvc video codec is developed by the software video compression company Divideon and was released in its first version in September 2017. The xvc codec has been developed primarily based on known technology that has been included in AVC or HEVC or that has been evaluated in the context of the Joint Video Exploratory Team (JVET). The codec has been designed with a specific focus on enabling efficient software implementations, but hardware considerations have also been taken into account. The reference software of xvc is publicly available and can be accessed from xvc.io 13 .

The xvc video codec is being developed with an ambition of being an independent, alternative and complementary video codec, offering a middle-way between the existing options represented by HEVC and AV1. Compared to HEVC, xvc offers better compression performance and a clearer and more manageable licensing situation with a single reasonable license available with compelling licensing terms. Compared to AV1, xvc offers better compression performance and lower computational complexity.

The xvc software-defined conformance definition

In order to guarantee interoperability between different implementations of video encoders and decoders, there needs to exist definitions of conformance. Conformance definitions have conventionally been expressed in a standard text or a specification document. In xvc, the conformance definitions relates to the reference software and is expressed as follows:

- A bitstream is a conforming xvc bitstream if and only if the current version of the reference xvc decoder successfully decodes the bitstream and returns “Conformance verified.”.
- An encoder is a conforming xvc encoder if and only if it produces conforming xvc bitstreams.
- A decoder is a conforming xvc decoder if and only if it produces identical output as the current version of the xvc reference decoder for all conforming xvc bitstreams.

It should be noted that the conformance definition relates to the current version of the reference software. This is in alignment with the desire to continuously evolve xvc and ensure that improved compression tools can be introduced over time. As far as it is practically possible, new versions of xvc decoders will be able to decode old versions of xvc bitstreams.

The xvc licensing framework

The developers of the xvc codec strongly believes in open, transparent and collaborative projects for developing interoperable technical systems. In some areas, it might be possible to perform such developments with royalty-free deliverables, without compromising on the performance of the result and without reducing the incentive for innovation. However, in the video compression area, it has been very clear that the most successful standards and the largest collaborative efforts have been around royalty-bearing standards such as MPEG-2 Video and AVC.

The xvc codec is a royalty-bearing codec with a one-stop shop license from which patent holders can receive reward in relation to their share of the total number of registered xvc patents. The xvc license is publicly available at xvc.io 13 and includes the specific details of what type of usage that is free of charge and what type of usage is associated with a fee. In general it can be said that the fee is based on the number of active xvc instances and that there are no content fees in the xvc license. The xvc license is intended to fully cover all rights needed for implementing, using and distributing xvc compatible implementations. If, at any point, a third party organization requests additional fees or additional licenses related to xvc then it is possible to report such requests to Divideon so that the technology in question can either become covered by the xvc license or be removed from the xvc codec.

TECHNOLOGY IN XVC

The basic building blocks of the compression technology in xvc resemble to a large extent the technology used in other modern video codecs such as AVC, HEVC and AV1. In summary it can be said that xvc is a block-based hybrid (inter/intra) codec that operates on raw pictures of YUV samples and compresses them to a NAL (network abstraction layer) unit structured bitstream. Each picture in a video sequence is divided into rectangular blocks of samples of size up to 64x64 samples, which are predicted from samples in the same picture (intra prediction) or samples in previously coded pictures (inter prediction). Residuals are transformed using non-square transforms and the coded symbols are compressed using a context-adaptive binary arithmetic coder. Block boundaries are filtered using a deblocking filter.

Functionalities and features

The xvc codec includes a slim layer of high level syntax, used to signal properties of the compressed video and information related to the reference picture structure. An xvc bitstream consists of one or more segments, which is an independently decodable set of pictures that starts with an Intra picture. Each segment starts with a segment header that includes information about the compressed video (such as resolution and chroma format) and which coding tools were used to compressed the segment. The xvc codec also includes a unique framework for handling open-GOP intra pictures in Adaptive BitRate (ABR) scenarios, a functionality that is described in more detail below. The xvc codec has support for several different chroma formats (Monochrome, 4:2:0, 4:2:2, and 4:4:4), several different bit-depths (8, 10, and 12), resolutions up to and beyond 8K, wide colour gamut, and high dynamic range (including HLG and PQ).

New coding tools

The xvc codec is an independent codec developed from scratch, but using technologies that resemble technologies in for example AVC and HEVC. In general, xvc is more similar to HEVC but there are several clear differences. One of the most significant differences is that xvc uses non-square coding units for which both prediction and transform is applied. Thus, xvc does not contain any separate trees of prediction units and transform units, as is the case with HEVC.

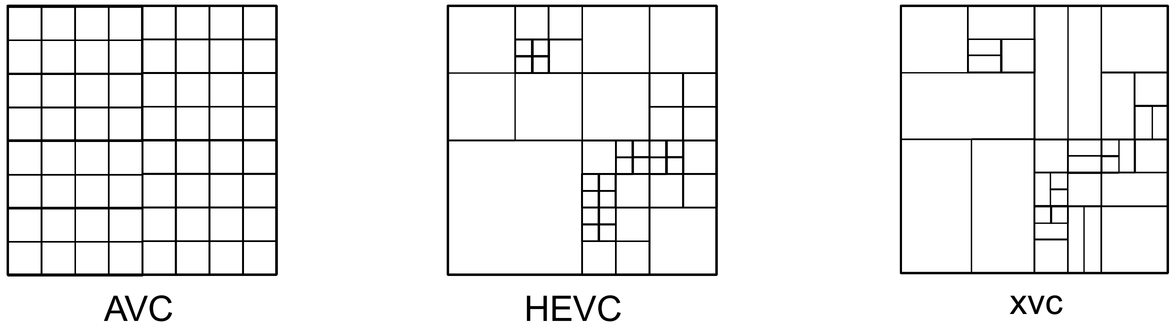

Figure 2 – Example block structure in AVC, HEVC and xvc.

The xvc codec includes extensions to technologies in HEVC in several areas, for example 67 intra prediction modes instead of 35, but there is also a significant number of new coding tools for which there is no corresponding technology in HEVC, such as:

- Adaptive motion vector precision – where the precision of the motion vectors are signalled to allow for more efficient signalling of long and integer motion vectors.
- Affine motion prediction – where individual motion vectors are calculated and applied for each 4x4 sub-block of a coding unit which makes it possible to better represent non-translational motions such as rotation and zoom.
- Cross component prediction – where chroma samples are predicted from luma samples using a linear model.
- Transform selection – where a set of different transforms with different characteristics are evaluated to determine which transform most efficiently represents the residual of a specific coding unit.
- Local illumination compensation – where a linear model is used to account for local offsets of sample values when predicting from a reference pictures, which is particularly useful for representing changes in lighting conditions of an object or a scene.

Open-GOP Intra pictures in Adaptive BitRate (ABR) streaming applications

In ABR streaming applications, the same video sequence is encoded in multiple different representations typically with a wide variety of bitrates and resolutions. The xvc codec includes support for switching between different encoded video formats (for example resolutions) by upsampling videos of lower resolution to a predefined higher resolution. This makes it possible to guarantee smooth video play-out in ABR applications with no need of re-initializing the decoder or having to run several decoder instances in parallel. A unique feature of the xvc codec is its ability to support open-GOP intra pictures in ABR streaming applications. Open-GOP (group of pictures) intra pictures is a concept that is used frequently in broadcast applications but that has not previously been applied in ABR streaming applications. An open-GOP intra picture is a picture that provides random access into a bitstream and at the same time allows for pictures that precede the intra picture in output order to use the intra picture for prediction. In other words, pictures that follows the open-GOP intra picture in output order cannot predict from any picture that precedes the open-GOP intra picture in output order, but there may be pictures that precedes the open-GOP intra picture in output order and predicts both from the open-GOP intra picture and from earlier pictures from before the intra picture. Pictures that predict from the open-GOP intra picture must naturally follow the open-GOP intra picture in coding order in order to use the open-GOP intra picture for prediction. Figure 3 illustrates the difference between closed-GOP intra pictures and open-GOP intra pictures. Open-GOP intra pictures offers significantly better compression performance than closed-GOP intra pictures both in PSNR but even more in subjective quality since annoying "intra pumping" effects can be reduced and or even completely removed.

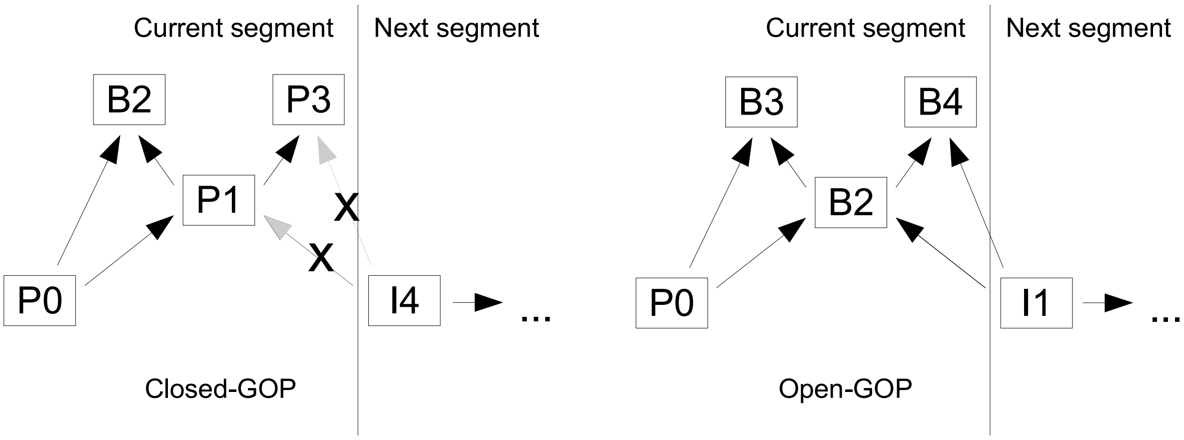

Figure 3 – Closed-GOP intra picture and open-GOP intra picture

Support for open-GOP intra pictures is enabled in xvc through the use of a concept called *tail pictures* and a syntax element called `buffer_flag`. The term tail picture is used to denote any picture that follows the last temporal layer 0 picture in a segment in output order and precedes the first temporal layer 0 picture in the next segment (i.e. the intra picture) in output order. If the intra picture in the next segment is an open-GOP intra picture, the tail pictures need to be decoded after the intra picture. In conventional codecs such as AVC and HEVC, those tail pictures would need to be signalled in the bitstream after the intra picture. This means that if open-GOP intra pictures would be applied to conventional codecs in ABR applications, the tail pictures would reside in the next segment even though they use pictures from the current segment for prediction. In conventional codecs, it is not defined how those tail pictures should be handled if a switching between different representations has occurred for example when the resolution is different in the different representations. If those pictures are simply discarded, there would be an annoying glitch in the play-out of the video. This is the reason why open-GOP intra pictures are not used with conventional codecs in ABR streaming applications.

In xvc, the tail pictures are signalled before the open-GOP intra picture in the bitstream as illustrated in Figure 4. The syntax element `buffer_flag` in the picture header is used to indicate if a picture is a tail picture that should be buffered by the decoder. If the `buffer_flag` in the picture header is set equal to one, the decoder will buffer the data for that picture in compressed form until it has received and decoded the next picture with the `buffer_flag` set equal to 0, i.e. the open-GOP intra picture. If a switching operation occurs between different representations, the tail pictures for the current segment will be available to the decoder regardless of the properties of the representation that is switched to. If the resolution (or some other video format property such as bitdepth or chroma format) is different in the two representations, a resampled version of the open-GOP intra picture will be generated to be used for prediction by the tail pictures. It can be noted that the sample values of the resampled reference picture generally does not exactly match the sample values of the reference picture that was used when encoding the tail pictures. There is therefore a risk of mismatch and visible degradations (local drift) within the tail pictures. However, such degradations are typically non-intrusive and encoders can include methods that minimizes the risk for visible degradations. In reality, such degradations are visible only in very rare cases, while the visual benefit of deploying open-GOP intra pictures can be clearly seen and appreciated across different bitrates and different content types. Figure 4 shows where tail pictures are signalled in xvc and in other codecs, respectively.

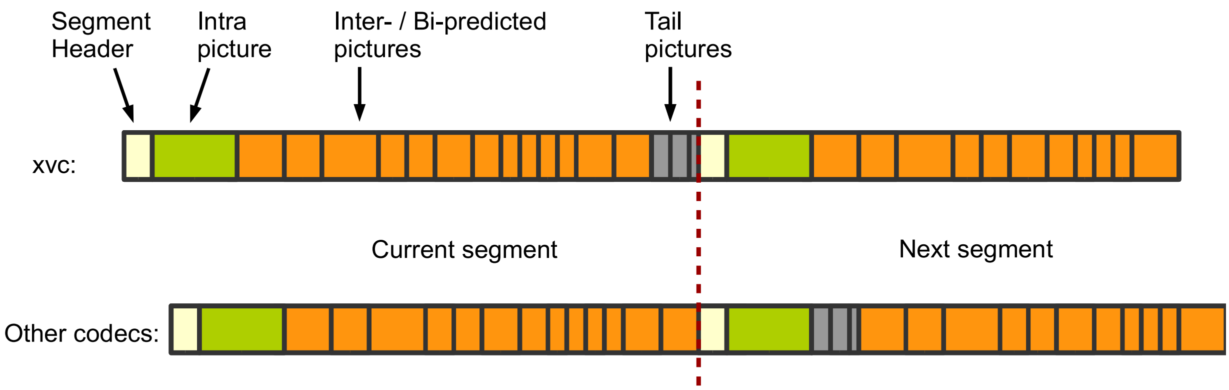

Figure 4 – Tail pictures in xvc compared to other codecs.

Restriction flags

The different coding tools in xvc can be enabled or disabled directly from the bitstream using a set of restriction flags in the segment header. In total, there are over 70 different restriction flags, and for each of the coding tools represented by these restriction flags, a simple fallback method is used when the tool is disabled. In many cases, the fallback method is just to bypass a processing step, e.g. not perform filtering of reference samples for intra prediction. In other cases the fallback method is a more basic method for performing a processing step, e.g. the fallback method for planar intra prediction is DC intra prediction. The restriction flags provide a powerful mechanism for quickly and dynamically turning off the use of a specific coding tool without the need for synchronized replacement of existing decoders – the existing decoders already knows how to handle the case when the coding tool is disabled. This makes the xvc codec much better suited to deal with potential IPR and licensing conflicts. The restriction flags are also a key component in the version handling scheme described below, which allows for a continuously evolving codec with smooth transitions between versions.

Version handling

The xvc codec has a powerful scheme for version handling, which allows the codec to evolve over time, with additions of new functionality and more advanced coding tools. All xvc bitstreams include, in the beginning of each segment header, one syntax element that represents the major xvc version and one syntax element that represents the minor xvc version.

The major xvc version is increased when non-backwards compatible changes are introduced, typically when new technology is added to xvc. When a new major version of xvc has been released, all existing decoders need to be updated over a period of time in order to support decoding of bitstreams of the new major version. Existing bitstreams do not have to be updated when the major version of xvc is increased, since new decoders will include support for decoding bitstreams with lower major version.

The minor xvc version is increased when backwards compatible changes are introduced. In practice, this occurs when technology is being disabled through setting their restriction flag equal to one. When a new minor version of xvc is released, existing decoders do not need to be updated immediately, since they already have support for decoding bitstreams in which the disabled technology is turned off. However, the reference decoder will be updated after some period of time to no longer accept bitstreams of the old minor xvc version. This is an action motivated by the fundamental principle that xvc will not include technology that cannot be licensed under reasonable terms, and thus support for such technology must be removed from the reference software. Eventually, all existing bitstreams have to be updated when the minor version of xvc is increased since the new version of the reference decoder will reject bitstreams with too low minor version when support for that technology has been removed from the reference software. In practice, it is expected to be extremely rare that the minor version has to be increased, but the framework for how to handle it is in place to ensure that there is always a deployable and licensable version of xvc available, and that no patent holder (or "patent troll") would be able to block the codec from being used altogether. There has been no occasions of tool removal and no minor version increments since the release of the first version of xvc in September 2017.

The separation of major version and minor version makes it possible to always allow for a transition period when changes are made to the xvc codec. In the case of a major version increase, new bitstreams will only be created once decoders have been updated to support the new version. In the case of a minor version increase, new decoders will not replace old decoders until existing bitstreams have been updated to use the new minor version. By applying this scheme, there is always a smooth upgrade-path between versions so that the codec can evolve over time without causing any interoperability problems.

Version 2.0 of xvc

The second version of xvc was released in July 2018 [12] and includes among other things a royalty-free baseline profile and a dual-licensing scheme for the xvc reference software, with an open source LGPL option as well as a commercial option. The baseline profile consists of a pure subset of the coding tools available in xvc. Only 25 of the more than 70 different tools are used in the baseline profile which makes it less complex but also less efficient (typically around 15% higher bitrate). The second version of xvc offers full compatibility with all xvc version 1 bitstreams – completely aligned with the xvc versioning framework.

XVC APPLICATIONS

With the source code of xvc publicly available, it is easy to try it out and evaluate it for different applications and use cases. It is for example currently being evaluated by an OTT service provider with focus on streaming to mobile devices where bandwidth savings relative to their current AVC-based solution is highly desirable. The xvc codec has been tested with several different open source applications including ExoPlayer, FFmpeg, and VLC. These integrations have made it possible to verify consistent behaviour of the xvc decoder on various devices and platforms, including Android, iOS, Windows and Linux.

There is also an online demo of xvc available at the Divideon webpage 14. In that demo, a JavaScript version of the xvc decoder is used to highlight both the quality improvement relative to AVC but also the fact that the xvc decoding is so light-weight that decoding of a 640x360 video can be performed directly in JavaScript with no need for updating the browser or installing any third party plug-in.

BATTERY PERFORMANCE

For mobile streaming applications and other video services targeting mobile devices it is important that the battery consumption is not too high when running the video decoding in software. Divideon has performed internal testing of the first version of xvc to monitor how much the battery consumption increases when decoding xvc in software relative to decoding AVC in hardware. For resolutions lower than HD, the battery consumption difference is generally negligible. As an example, continuous playback of SD video encoded with xvc could run for more than 9 hours on a Samsung Galaxy S8+ while an AVC-encoded video could run for just a few minutes more. For video of 720p resolution there was a battery time reduction of 20% when using software xvc decoding relative to using hardware AVC decoding.

RESULTS

Results are presented for xvc relative to the reference software of HEVC (HM) and the AV1 software. The results are generated in alignment with two different testing documents; the NETVC Evaluation Methodology 15 that has been used during the development of AV1, and the Common Test Conditions from JCT-VC 16 that were used in the development of the HEVC codec. Since the focus of this paper is on mobile streaming applications, results are only provided for resolutions of 720p and lower. It should be noted that all codecs have been run with their slowest speed setting, which for xvc means -speed 0 (“placebo”). It is therefore significantly slower than HM but still more than 3 times faster than AV1. All results are with threading disabled for all codecs. It can be noted from these results that AV1 appears to be providing slightly better performance than HEVC (HM) at a complexity increase which corresponds to around 50 times longer encoding time.

Two modifications were applied relative to the JCT-VC test conditions in order to give as fair comparison as possible: Closed-GOP intra pictures were used for all codecs (since the

AV1 software does not seem to have encoder support for open-GOP intra pictures) and HM and xvc were allowed to use up to 3 reference pictures in each reference picture list in order to better match the number of reference pictures used by AV1.

		BD-rate of xvc vs. HM (%)			**BD-rate of xvc vs. AV1 (%)**		
Sequences	Resolution	PSNR-Y	PSNR-U	PSNR-V	PSNR-Y	PSNR-U	PSNR-V
Class of 360p seq.	360p	-26.08	-24.74	-28.75	-19.72	-9.56	-3.35
Class of 720p seq.	720p	-20.76	-30.04	-32.64	-13.28	-0.80	-4.43
Average		**-23.42**	**-27.39**	**-30.70**	**-16.50**	**-5.18**	**-3.89**
Encoding time			**11.9x**			**0.26x**	
Decoding time			**0.71x**			**0.61x**	

Table 1 - Bitrate savings of xvc for the low resolution sequences of the NETVC test conditions.

The results for xvc are generated with commit f0b3154, from 2018-02-28. The results for HM are generated with HM16.17 and for AV1 commit eede835, from 2018-04-25 is used for the JCT-VC test (with –kf-min-dist –kf-max-dist adjusted for the intra period and with --lag-in-frames=17) and commit 1a70994, from 2018-02-02, is used for the NETVC test. Both AV1 versions are run with --auto-alt-ref=2 --cpu-used=0 --passes=1 --threads=1.

		BD-rate of xvc vs. HM (%)			**BD-rate of xvc vs. AV1 (%)**		
Sequence	Resolution	PSNR-Y	PSNR-U	PSNR-V	PSNR-Y	PSNR-U	PSNR-V
BQSquare	240p	-11.51	-18.49	-24.95	-3.15	-7.49	-11.6
RaceHorses	240p	-11.41	-7.59	-7.61	-9.23	-4.66	-5.27
BasketballPass	240p	-13.62	-16.6	-10.44	-11.88	-8.65	-6.54
BlowingBubbles	240p	-7.73	-8.96	-12.86	-13.35	-9.43	-13.02
BQMall	480p	-12.84	-18.9	-19.23	-15.11	-12.99	-14.86
PartyScene	480p	-11.15	-14.94	-16.34	-6.38	-5.95	-7.06
RaceHorses	480p	-11.51	-4.93	-4.03	-4.2	-2.23	1.09
BasketballDrill	480p	-17.19	-17.73	-18.54	-10.81	-1.3	-3.28
Average:		**-12.12**	**-13.52**	**-14.25**	**-9.26**	**-6.59**	**-7.57**
Encoding time			**15.0x**			**0.31x**	
Decoding time			**0.70x**			**0.66x**	

Table 2 - Bitrate savings of xvc for the low resolution sequences of the JCT-VC test conditions.

SUMMARY

This paper has presented the software-defined video codec xvc which is particularly well suited for providing the highest video quality under the most challenging network conditions, such as in mobile video streaming applications. Comparisons to the other recent video codecs – HEVC and AV1 – show that xvc is capable of delivering reduced bitrate in the range of 10% to 25% for the same visual quality, with clearly lower computational complexity compared to AV1. The paper has also provided information about the xvc licensing framework and the xvc version handling, a powerful mechanism that enables a cutting-edge performance level and at the same time ensures that the xvc codec is licensable under reasonable and well-defined licensing terms.

REFERENCES

[1] ITU-T H.265 / ISO/IEC 23008-2 “High Efficiency Video Coding”

[2] Alliance for Open Media https://aomedia.org/

[3] ITU-T H.264 / ISO/IEC 14496-10 “Advanced Video Coding”

[4] ISO Standards and Patents site https://www.iso.org/iso-standards-and-patents.html

[5] IEC patents database http://patents.iec.ch/

[6] IPR in ITU Recommendations site http://www.itu.int/net4/ipr/search.aspx

[7] MPEG LA HEVC http://www.mpegla.com/main/programs/HEVC/Pages/Intro.aspx

[8] HEVC Advance https://www.hevcadvance.com/

[9] Velos Media http://velosmedia.com/

[10] https://aomedia.org/alliance-to-deliver-next-generation-open-media-formats/

[11] https://aomedia.org/the-alliance-for-open-media-kickstarts-video-innovation-era-with-av1-release/

[12] Divideon Releases xvc 2.0 - An Open-Source Video Codec with a Royalty-Free Baseline http://www.releasewire.com/press-releases/divideon-releases-xvc-20-an-open-source-video-codec-with-a-royalty-free-baseline-1005793.htm

[13] xvc website https://xvc.io/

[14] Divideon website https://www.divideon.com/

[15] Daede, T., Norkin, A., and I. Brailovsky, "Video Codec Testing and Quality Measurement", https://datatracker.ietf.org/doc/draft-ietf-netvc-testing/

[16] Sharman, K., Suehring, K., “Common test conditions” JCTVC-Z1100, Geneva, http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=10693