|  |  |
| --- | --- |
| **ITU – Telecommunications Standardization Sector**STUDY GROUP 21 Question C/16**Video Coding Experts Group (VCEG)**75th Meeting: 2-8 November 2024, Kemer, TR | Document VCEG-BW06-v1 |

|  |  |
| --- | --- |
| Question: | C/16 SG21 (VCEG) |
| Source: | **Stephan Wenger, (Tencent)** | Email: | swenger@global.tencent.com |
| Title: | **Proposed additional requirements for the next generation video coding standard** |
| Purpose: | Proposal |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

The authoring company believes that the draft requirements documented in the VCEG meeting report of Sapporo are an excellent starting point for future discussion. However, we also believe that for a successful next generation video coding standards, we need further refinements in at least the following directions:

* Neural network support in the loop.
* Source-coding based error resilience for ultra-low delay use cases, if those were to be considered.

# Introduction

It is, to us, currently not clear that the world needs another video coding standard addressing requirements substantially similar to those adequately addressable by VVC, plus additional coding efficiency at the cost of additional complexity. VVC adoption is slow, perhaps in part because even for VVC, the coding efficiency gain vs. complexity increase equation is not appealing to the market. That goes for the decoder but also, and especially, for the encoder. There is little evidence for mainstream video resolutions growing beyond 4k. Networks are getting faster, and storage cheaper. Energy consumption of the whole video ecosystem is not quite an issue yet, but with increasing environmental consciousness, it may well become one in the 2030 timeframe and beyond.

We think that a next generation video coding standard needs to have incentives beyond coding efficiency increase at impractical encoder complexity levels. The old story that encoder complexity doesn’t matter because it’s an implementation issue/product differentiator doesn’t hold anymore. The lack of fast adoption of both HEVC and VVC tell that tale.

As for novel applications or use cases or technologies that could possibly ensure a next generation video coding standard to see reasonably fast adoption, we see at least the following two directions that need, in our opinion, to be explored before formal standardization can start. i.e. during the ongoing exploration phase or in a CfE:

* Neural network support in the loop.
* Source-coding based error resilience for ultra-low delay use cases, if those were to be considered.

Both are elaborated upon in the following.

# Neural network support in the loop

Over the past two+ years, a large number of JVET experts have worked on neural network-based coding technologies. Coding efficiency gains are impressive. However, the computational complexity, using general purpose CPUs and our current testing methodology, is preventative for practical use. There are at least two options to deal with this situation.

The first option is to move the use of neural networks out of the loop into optional post-processing. Great progress has been made on this front with technologies such as the neural network post filter-related SEI messages. That said, with an out-of-loop approach, however it may be specified, one cannot in practice guarantee performing applications. Implementers can and will take shortcuts. Further, out-of-loop technologies can, and in in most cases are, specified such that they are agnostic to what goes on inside the loop, and hence agnostic to the video codec standard in the traditional sense. Relegating neural network support to out-of-loop designs may, therefore, not help in the adoption of a new video coding standard.

The second option is to use neural network-based coding technology in the loop. On a classic CPU-based system using classic complexity measurements, or in classic hardware designs, that has been shown to be prohibitively expensive. However, in the 2030 timeframe—the earliest we can reasonably expect adoption even if the most aggressive schedules proposed were implemented (and note that we are not supportive of such an aggressive timeframe!)—we believe that GPUs will be ubiquitous in target devices, from mobile phones up. Using GPU support, the use of neural networks seems much less an obstacle. Of course, devices and especially hardware decoders will probably have to consider major architecture changes; either including GPU functionality on a decoder chip or providing the high-speed data path between the GPU and the decoder chip. Software implementations, OTOH, can likely take advantage of close GPU/CPU interaction right away, and that has been demonstrated in the context of video decoding multiple times by multiple parties over the last few years.

Assuming the availability of a sufficiently powerful GPU sufficiently closely integrated with the CPU and other decoder hardware technologies changes the complexity equation fundamentally. This, we believe, is not reflected in our current test conditions (neither in JVET’s CE1 nor in CE2), nor is it trivial to do. Such a study probably will have to be conducted before, or the latest, during a call for evidence

# Ultra-low-delay/latency, and error resilience

Ultra-low delay (ULD) video coding seems to be one key application requirement mentioned by many. We define ULD as such a codec technology where decoding can complete for a subset of a reconstructed picture before the encoding of the same picture is concluded, assuming zero transmission delay.

Obviously, ULD makes sense only in the context of real-time transmission—for store-forward, it doesn’t matter whether the latency is sub-picture or a full picture, and in most cases, Group-of-Picture (GOP)-time-interval delays are acceptable and in practical use. ULD also loses some of its appeal as frame rates go up. Historically, at 15 fps, ULD-like coding was a necessity for a well-performing video conferencing system. At 240 Hz frame rate, ULD, by definition, provides benefits only in systems that care for an improvement of glass-to-glass latency below 1/240s or approximately 4 milliseconds. That’s not likely an issue for human consumption; hence ULD seems more likely relevant in the context of machine-to-machine video.

When considering real-time transmission at latencies in the single-digit millisecond range, the subject of error resilience immediately comes up. Even in LAN environments, demand-based retransmission in case of packet loss does not work. FEC may work but adds significant bitrate. It may be sensible to spend those FEC bits on source-coding based error resilience instead of FEC. If we were ambitious, we could venture into joint source-channel coding optimization, but that’s hard, and we are not sure JVET has the mandate or the expertise to conduct such a study.

Source coding-based error resilience, and companion technologies such as error concealment, were subject of much work in standards committees until 2005 or so. Not much work was observed thereafter. We currently don’t have an adequate toolchain for error resilience tests. Our reference decoders would most likely crash upon observance of a missing NAL unit, and certainly would not conceal errors well (or at all). The error patters used previously were taken from the Internet some 20 years ago and likely show fundamentally different traffic characteristics compared to what can be observed today. We could leverage work that was conducted by 3GPP, IETF, academics, and/or IEEE, but that would reflect only their network designs and preferences. In short, bootstrapping error resilience work is a major effort, and that effort should take place at the CfE stage or earlier, so to have evidence that our forthcoming design can address the relevant use cases.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_