|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, CH, 21-30 November, 2011 | Document: JCTVC-G1016r1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **JCT-VC break-out report: Adaptation Parameter Set issues** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Report | | |
| *Author(s) or Contact(s):* | **Stephan Wenger**  Vidyo |  | [stewe@stewe.org](mailto:jill@vidyo.com) |
| *Source:* | BoG | | |

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

# Abstract

A break-out group meeting was held Saturday, Nov. 26 2011 from 8:00 through 10:10. Approximately 20 delegates were present.

# Mandates

The mandates of the BoG are:

* Review and summarize input documents G122, G220, G295, G330, G332, G 566, G658
  + Presented by proponents if present
  + Review of input document otherwise
* Make recommendations to JCT-VC on disposal of input documents
* Identify other open issues in APS design, including need for harmonization (if time available)

# Recommendations

Based on contributions G.122 and G.220, it was agreed to recommend removing CABAC from the APS. It was further agreed to recommend removal of any byte alignment syntax elements in the APS, as they are not needed anymore. WD text is included in G.122.

While it may not be in the mandate of the BoG to recommend anything beyond APS related proposals, the BoG nevertheless agreed to recommend not to include henceforth any CABAC coded syntax elements “above” the slice header (inclusive). It was agreed that we have an issue we called “partial update”, that will need to be addressed at some point in time. “Partial update” is to be understood that from one picture to another picture, *parts* of the information carried in APS can change without transmission of a full APS carrying both the changed and unchanged information. There are two very similar proposals addressing this issue at this meeting (G295 and G332), both of which rely on multiple APS references in the slice header. The proponents of G295 and G332 agreed to prepare a combined proposal for consideration by JCT-VC. It was also remarked that document E309 addresses the same problem in a different way, avoiding multiple references to parameter sets in the slice header. It was further remarked that a solution according to G295/G332 could co-exist with a solution based on E309, as they have different tradeoffs in terms where the bits are spent (slice header vs. additional NAL units). The BoG did not agree on a solution for this problem, but recommends that JCT-VC considers the unified proposal at this meeting.

It was agreed to recommend to JCT-VC to include a tool that allows MTU size matching (akin slices) for parameter sets and especially for the APS, if there is an expectation that a parameter set may become larger than commonly used MTU sizes. It was not clear to the participants whether we have this problem at this point in time. It would be good if this can be clarified by the larger group.

The group discussed the allegedly different error resilience properties of APS and other parameter sets, and to which extent that should affect the selection of data to be carried in the APS. It was noted that, due to its strict timing requirements, APSs are likely to be conveyed in-band and synchronous to the rest of the bitstream (at least at picture and perhaps even at slice level), and, therefore, are less easily protectable than other parameters sets. Conceivably, this could be a reason to limit the data included into an APS to such data whose loss is not catastrophic to the decoding process. However, BoG came to the conclusion that this argument is not very compelling. First, from a compliance viewpoint, the APS includes data that is required for the decoding process. Decoding data in the absence of the correct APS is an incompliant operation, with all its consequences. Second, it is very hard to define “catastrophic”. ALF parameters were claimed to be one examples where a loss is not catastrophic—if you loose those, you would still get a somewhat appealing picture out of your decoder. However, it’s probably not difficult to design an evil bitstream which includes ALF parameters that render that bitstream useless if ALF not applied properly. Accordingly, the BoG recommends to work under the assumption that an APS is available at decoding time of a slice it references that APS, and expect system layer support to ensure sufficiently reliable transport of the APS for the application in question.

It was agreed that the APS is the right place for quantization matrices, as proposed in 295, 330, and 658, certainly if we can get the “partial update” problem under control.

It was agreed that, for the RPS related syntax elements, what is currently in the PPS should stay in the PPS, and what is currently in the slice header to be moved to APS. Ye-Kui agreed to write drop-in text for the larger group and the editors. The group agreed to include this document into the zip file of this report, and it is incorporated by reference into this report.

# List of AHG related input documents / detailed notes

### G.122 VLC for high level syntax (ALF and SAO parameters)

This contribution proposes always using VLC for high level syntax elements; specifically, ALF and SAO parameters are coded with VLC even when CABAC is selected as the entropy coder. These parameters include **alf\_cu\_control\_flag, alf\_cu\_control\_max\_depth, alf\_length\_cu\_control\_info, alf\_cu\_flag, sao\_split\_flag, sao\_flag\_cb, sao\_flag\_cr, sao\_type\_idx, sao\_offset**. This change has negligible impact on coding efficiency in HM-4.0 under common conditions (0.0 % AI-HE, 0.1% RA-HE, 0.1% LD-HE).

Not presented by proponent, but reviewed by group and agreed.

### G.220 Non-CE8: Pure VLC for SAO and ALF

In HM-4.0-dev-miscs, SAO and ALF parameters in APS and CU-level ALF-on/off flags in slice header can be coded by CABAC. No other syntax elements in APS and slice header can be coded by CABAC. In this contribution, it is proposed to use pure VLC for SAO and ALF and to remove byte alignment bits for CABAC in APS. Simulation results reportedly show 0%, 0.1%, 0.2% and 0.2% coding efficiency gains for HE-AI, HE-RA, HE-LDB, and HE-LDP, respectively, when the APS coding is changed from CABAC to VLC. Simulation results also show no coding efficiency impact when the slice header coding is changed from CABAC to VLC.

Presented by proponent, reviewed by group and agreed.

### G.295 Non-CE4 Subtest3 : Extension of Adaptation Parameter Sets syntax for Quantization matrix

In WD4, Adaptation Parameter Sets(APS) is defined for ALF and SAO parameters. APS has a parameter “aps\_id” which is referred to in the slice header. Quantization matrix in HEVC was proposed in JCTVC-F362/F475. In those proposals, quantization matrices are transmitted as new NAL unit. This contribution propose,

1. Extension of APS syntax for Quantization matrix.
2. Instead of aps\_id, new parameters: aps\_sao\_id, aps\_alf\_id and aps\_qmatrix\_id are defined.
3. In slice header, aps\_sao\_id, aps\_alf\_id and aps\_qmatrix\_id are signaled individually.

Point 1: agreed.

Point 2 and 3: there is a general agreement that some form of a “partial update” or “partial reference” mechanism is needed. No agreement over the mechanism proposed in the document. Proponents of G.295 and G332 will work towards a unified solution for consideration in JCT-VC.

### G.330 Syntax elements in adaptation parameter set

This document discusses what syntax elements to be included into the Adaptation Parameter Set (APS). Currently, the APS may include ALF and SAO parameters. In this document, inclusion of other syntax elements, namely 1) reference picture list construction related syntax elements; 2) weighted prediction related syntax elements; 3) decoded picture buffer management related syntax elements; and 4) quantization matrices table, in APS is discussed.

1. reference picture list construction related syntax elements ONLY in APS??? It was remarked that RPS list could change from slice to slice, so slice header may also be an option. Or both.
2. Weighted prediction is the same in nature as 1), so also open.
3. RPS syntax elements. Some stuff in PPS, and what’s currently in the slice header to be moved to APS. Ye-Kui to write drop-in text for the editor.
4. Already agreed to be in APS

### G.566 Syntax Refinements for SAO and ALF

The Adaptation Parameter Set (APS) syntax structure has been adopted during the 6th JCT-VC meeting in Torino to be used in the conveyance of parameter sets of Adaptive Loop Filter (ALF) and Sample Adaptive Offset (SAO). However APS is a very recent syntax structure whose performance and usage needs to be improved and refined. In the current proposal 3 issues related to APS and ALF/SAO syntax are addressed. The proposed solutions provide on average 0.2% coding gain for low-bitrate applications, improve parsing robustness and reduce latency of HEVC.

Flags are reinserted enabling/disabling ALF and SAO per slice. It was claimed that doing so has advantages for parallel processing and for error resilience due to parsing independency in the direction slice header to APS. The group found this compelling. However, it was also noted that, as the number of tools that keep information in the slice header grows, so probably will grow the number of bits (at least if we desire an architecturally clean design).

It was further claimed that enabling/disabling ALF/SAO allows to selective use either tool with an existing APS (without further APS transmission) even in the case where, for example APS parameters are adequate but SAO parameters are not. The group is not chartered to decide on the slice-adaptive switching on/off of SAO and ALF. However, from an APS high level syntax perspective, there is no reason to believe that such a functionality would create a problem. It was noted, if APS updates were allowed between slices, the mentioned functionality could also be addressed through APS updates.

BoG recommendation: adopt reinserted enabling/disabling ALF and SAO flag in slice header. Allow different values in APS and slice header (suggested restriction: if flag is “0” in APS , it cannot be “1” in slice header).

### G.332 Multiple Adaptation Parameter Sets Referring

In this document, a multiple adaptation parameter sets (APSs) referring approach is proposed. Since APS tends to carry the coding parameters which are more likely to be changeable from picture to picture, or even from slice to slice, therefore, such a referring design is proposed to be at slice layer. In a slice layer encoding/decoding, multiple APS IDs can be made available, and each coding tool, such as SAO or ALF, *etc*., is initialized by activating only once in one APS of the several referred ones. Thereby, the encoder is allowed to code SAO and/or ALF parameters by directly utilizing whole or partial information already presented in the existing APSs. This facilitates flexible configuration of the coding tools and saves bits when re-using the former tool parameters while coding the current slice. The proposed scheme can also serve both the case when parameters change from slice to slice, such as in video coding tool with Weighted Prediction, and the case when parameters intend to be unchanged for the whole sequence.

Roughly comparable to Sony proposal (G295) in that the burden of selecting parts of the APS are moved to the slice header. It was remarked that this proposal may be a bit more future-proof than the Sony proposal.

### G.658 Quantization matrices in fragmented APS

In this document, it is proposed that the quantization matrices are signaled in the Adaptation Parameter Set (APS), with predictions between quantization matrices inside the same NAL unit being allowed. To solve the problem that an APS can be larger than the Maximum Transmission Unit (MTU) size, it is proposed that the APS may be fragmented into multiple NAL units, each of which can be independently parsed and applied.

It was agreed to recommend to JCT-VC to include a tool that allows MTU size matching (akin slices) for parameter sets and especially for the APS, if there is an expectation that a parameter set may become larger than commonly used MTU sizes. It was not clear to the participants whether we have this problem at this point in time. It would be good if this can be clarified by the larger group.