|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  8th Meeting: San José, CA, USA, 1–10 February, 2012 | Document: JCTVC-H0069 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **APS Referencing** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ming Li and Ping Wu No. 50 Ruanjiandadao Road, ZTE R&D Center District II, Nanjing, China 210012  Stephan Wenger and Jill Boyce 433 Hackensack Ave., Hackensack, N.J. 07601, USA  David Flynn BBC R&D, Centre House, 56 Wood Lane, W12 7SB London, UK | Tel: Email: | Ming Li, +86-25-88014641 {li.ming42, ping.wu}@zte.com.cn  Stephan Wenger,  +1-415-713-5473 {stephan,jill}@vidyo.com   davidf@rd.bbc.co.uk |
| *Source:* | ZTE, Vidyo, BBC Research & Development | | |

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

# Abstract

Proposed is a partial update mechanism for parameter sets, particular for the adaptation parameter set, based on parameter set referencing. Parts of a parameter set can be overridden by another parameter set of the same type, that has the to-be-overridden parts populated with updated data, and is inserted into a linked list of parameter sets. Bitrate savings over a full update of a parameter set every time even when only a (small) part changes are, in the authors’ opinion, obvious, but not quantified in this document.

# Introduction and Problem Statement

HEVC is likely to contain a number of technologies that require picture-level adaptive parameters not lightweight enough to be replicated in every slice header. In compliance with widely agreed architectural considerations, such parameters are carried in parameter sets. However, some picture level parameters are unlikely to change frequently. Others are picture adaptive in at least some encoder implementations and, therefore, likely to change from picture to picture. Placing such parameters into an H.264/AVC style picture parameter set has disadvantages from a coding efficiency viewpoint, as the unlikely-to-change parameters need to be sent just to allow the sending of the likely-to-change parts. Such an issue did not exist in H.264/AVC, and a novel solution is required. The need for such a solution is widely understood and agreed at least among the delegates attending the BoG on APS in Geneva. As noted in the BoG report JCTVC-G1016 [1], “[…] It was agreed that we have an issue we called “partial update”, that will need to be addressed at some point in time […]”.

At the 6th JCT-VC Meeting (Torino), based on the contribution JCTVC-F747 [2], an Adaptation Parameter Set (APS) was developed and adopted. The APS is intended for picture-adaptive data, including parameters related to sample-adaptive offset (SAO), adaptive loop filter (ALF), deblocking filter (DPS), and scaling list (quantization matrix information). Conceivably, more parameters may be added in the future; for example, the reference picture ordering breakout has looked into proposals to convey the reference picture ordering information (ex MMCO) in the APS, weighted prediction information may be information suitable for the APS, and so on. The PPS is retained for non picture-adaptive data that still can change within a sequence. Both APS and PPS are referenced from the slice header.

Further available in the slice header are flags enabling certain tools, such as ALF, or SAO.

Although the APS provides an effective approach to share picture-adaptive information common at the slice level , there still exists a problem of duplication of the coded APS information when only a part of the APS parameters changes between pictures. For example, conceivably, loop filter parameters may be calculated only once every n pictures (as this is a very computationally intensive operation, and under the assumption that content characteristics change slowly), whereas SAO parameters may be generated more frequently. Another example would be that, assuming something like an IBBPBB picture structure, certain picture adaptive parameters are calculated by a non-real-time encoder only once per GOP and picture type. In either case, a part of an APS already available at the decoder may be re-usable (as it already contains the required information), whereas another part may be new.

It is suggested that finding a generic solution for partial parameter set updates is preferable over a specific solution that addresses only those technologies currently on the table. Being future-proof on architectural features like this is hardly ever wrong, even if it comes at some (manageable) cost—sub-optimal efficiency, complexity, or otherwise. Historically, parameter sets have grown over time, and we are unaware of any reason to believe that this is changing in HEVC. A solution should scale with this anticipatable growth

# Solution

To address the problem of a partial parameter set update, a referencing approach is proposed. This proposal is a refined version of the attachment to JCTVC-G1016 [1] that was developed in Geneva. While this proposal is concerned only with the APS, the technology proposed would work with all other currently envisioned parameter set types (see also document JCTVC-H0071 on the subject of integrating APS data into the PPS). In the following, we make reference only to the APS for convenience.

The APS is subdivided into a number of groups of syntax elements, each associated with a certain coding technology (such as ALF, or SAO). Each of these groups is preceded by a flag indicating their respective presence. For example, the alf\_params() are preceded by an aps\_adaptive\_loop\_filter\_data\_present\_flag. \_, aps\_deblocking\_filter\_flag, aps\_sample\_adaptive\_offset\_flag

The APS also includes a conditional reference to another APS. A ref\_aps\_flag signals the presence of a reference ref\_aps\_id referred to by the current APS. With this link mechanism, a linked list of multiple APSs can be created. The decoding process during APS activation uses the reference in the slice header to address the first APS of the linked list. Those groups of syntax elements for which the associated flag (such as the aps\_adaptive\_loop\_filter\_data\_present\_flag) is set, are decoded from the subject APS. After this decoding, the linked list is followed to the next linked APS (if any—as indicated by ref\_aps\_flag equal to 1). Only those groups which were not signaled as present previously, but are signaled as present in the current APS, are decoded from the current APS. The mechanism continues along the list of linked APSs until one of three conditions are met: (1) all required groups (as indicated by SPS, PPS, or profile/level) have been decoded from the linked APS chain, (2) the end of the list is detected, and (3) a fixed, probably profile-dependent, number of links have been followed—the number could be as small as one. If there are any groups that are not signaled as present in any of the linked APSs, the related decoding tool is not used for this picture.

Condition (2) prevents circular referencing loops. The complexity of the referencing mechanism is further limited by the finite size of the APS table.

As already agreed in JCT-VC, the slice header further contains a set of flags, one for each APS-related coding tool, which can enable/disable the tool for the subject slice. These flags have no relationship with the activation process. Parsing independency between the APS activation and the flags is thereby maintained.

Figure 1 shows an example.

When encoding picture 0, a simple encoder which wants to update, for example, both the sao\_params() and alf\_params(), selects an APS currently not used by the decoding process from the table (in the example of figure 1, it’s saying APS(1)), places sao\_params() and alf\_param() into it, sets the aps\_sample\_adaptive\_offset\_data\_present\_flag and aps\_adaptive\_loop\_filter\_data\_present\_flag, and clears aps\_scaling\_list\_data\_present\_flag in APS(1). Further, it sets the ref\_aps\_flag to 1, and ref\_aps\_id to 0, e.g. APS(0), which contains the appropriate scaling list parameters other than the sao\_params() and alf\_params(). In doing so, the new APS in inserted at the start of the APS linked list and, according to the algorithm described, sao\_aparam() and alf\_params() are taken from the new APS(1) while scaling list\_param() from older APS(0) in the APS linked list.

If, for the next picture, (picture 1), only new SAO parameters were required to be updated, the encoder has several choices. It can, for example, select an APS not currently used for the decoding process (here APS(2)), populate it with SAO parameters, and insert it at the start of the list as already described. After doing so, the linked list would have grown by one APS element, as shown in Figure 1. However, it can equally place the SAO parameters into the APS that contained the SAO parameters before, i.e. APS(1) that was once referred to by the slices in picture 0, without changing ref\_aps\_flag, ref\_aps\_id and the loop filter data. In this case, the length of the APS reference chain remains the same as before. Note that, in some cases, the encoder may have no other choice but using the second mechanism, because the number of entries and the maximum size of the linked list are limited.

As for those limits, we suggest that eight entries in an APS table are appropriate. For the length of the linked list, two elements is the minimum that allows the mechanism to work, and anything larger than three appears to us to be overkill. The table size needs to be profile-independent so to preserve parsing independency between SPS and APS, but the maximum length of the linked list can and probably should be profile-dependent and scale with the number of tools with APS data structures allowed in the profile.

Another special case in Figure 1 is that tool parameters are not available after parsing an APS. Suppose that when coding picture 2, the scaling list data is unchanged, sao\_params() is newly derived, and ALF is not applied. Moreover, the encoder has decided that the data in APS(2) will not be needed in the future any more. Therefore, the encoder generates a new APS with an used APS ID of 2 to overwrite the older APS(2), where there are new SAO parameters but no loop filter data. ref\_aps\_flag is set to 1 and ref\_aps\_id to 0. Therefore, no ALF parameters will be obtained by parsing this APS link, and the encoder sets the slice header adaptive\_loop\_filter\_flag, to 0. Note that an encoder must always guarantee that when scaling list, SAO and/or ALF are used in coding a slice, the corresponding tool parameters will be available after parsing referred APS.



Figure 1. Example of APS referencing

# Modified Syntax and Semantics

The modifications of syntax and semantics for the proposed design are listed in the table below.

Table 1 Modifications Summary

|  |  |
| --- | --- |
| APS | Adding one flag indicating whether this APS refers to an existing APS. If true, adding one APS ID signaling the reference APS.  Adding three new elements signaling whether scaling list, SAO, ALF parameters are present in this APS;  Removing SAO, ALF ON/OFF control flags, i.e. aps\_sample\_adaptive\_offset\_flag and aps\_adaptive\_loop\_filter\_flag. |
| Slice Header | Adding an APS ID presentation condition for scaling list.  Adding scaling list, SAO and ALF ON/OFF flags to specify the applications of these tools in decoding this slice. |

Note that to separate the presence flags from the control flags, the two ON/OFF flags, i.e. aps\_sample\_adaptive\_offset\_flag and aps\_adaptive\_loop\_filter\_flag, are removed from APS, as previously agreed in Geneva. The slice level ON/OFF control flags are signaled in slice header. The modified APS syntax based on the WD document [3] is presented below.

|  |  |  |
| --- | --- | --- |
| aps\_rbsp( ) { | Descriptor | |
| **aps\_id** | ue(v) | |
| **aps\_ref\_flag** | u(1) |
| if( aps\_ref\_flag ) |  |
| **aps\_ref\_idx** | ue(v) |
| **aps\_scaling\_list\_data\_present\_flag** | u(1) |
| if( aps\_scaling\_list\_data\_present\_flag ) |  |
| scaling\_list\_param( ) |  |
| **aps\_deblocking\_filter\_data\_present\_flag** | u(1) |
| if( aps\_deblocking\_filter\_data\_present\_flag ) { |  |
| **disable\_deblocking\_filter\_flag** | u(1) | |
| if( !disable\_deblocking\_filter\_flag ) { |  |
| **beta\_offset\_div2** | se(v) | |
| **tc\_offset\_div2** | se(v) | |
| } |  | |
| **aps\_sample\_adaptive\_offset\_data\_present\_flag** | u(1) |
| if( aps\_sample\_adaptive\_offset\_data\_present\_flag ) { |  |
| sao\_param( ) |  |
| } |  |
| **aps\_adaptive\_loop\_filter\_data\_present\_flag** | u(1) |
| if( aps\_adaptive\_loop\_filter\_data\_present\_flag ) |  |
| alf\_param( ) |  |
| } |  |
| **aps\_extension\_present\_flag** | u(1) |
| if( aps\_extension\_present\_flag ) |  |
| while( more\_rbsp\_data( ) ) |  | |
| **aps\_extension\_data** | u(1) |
| rbsp\_trailing\_bits( ) |  | |
| } |  | |

The modified slice header syntax is presented below.

|  |  |
| --- | --- |
| slice\_header( ) { | Descriptor |
| **lightweight\_slice\_flag** | u(1) |
| if( !lightweight\_slice\_flag ) { |  |
| **slice\_type** | ue(v) |
| **pic\_parameter\_set\_id** | ue(v) |
| **aps\_id** | ue(v) |
| if (scaling\_list\_enabled\_flag) |  |
| **scaling\_list\_flag** | u(1) |
| if (sample\_adaptive\_offset\_enabled\_flag) |  |
| **sample\_adaptive\_offset\_flag** | u(1) |
| if (adaptive\_loop\_filter\_enabled\_flag) |  |
| **adaptive\_loop\_filter\_flag** | u(1) |
| **……** |  |
| } |  |

7.4.2.5 Adaptation parameter set RBSP semantics:

Adaptation parameter sets supports partial updating utilizing the aps\_id aps\_ref\_flag and aps\_ref\_idx syntax elements.

**aps\_id** identifies the adaptation parameter set that is referred to by the slice header or by the ref\_aps\_id in another adaptation parameter set. The value of aps\_id shall be in the range of 0 to 7, inclusive.

**aps\_ref\_flag** equal to 1 specifies that this adaptation parameter set references another adaptation parameter set. aps\_ref\_flag equal to 0 specifies that this adaptation parameter set does not reference any other adaptation parameter set.

**aps\_ref\_idx** identifies a subordinate adaptation parameter set for use in construction process of the active adaptation parameter set.

**aps\_ref\_idx** specifies that the previous adaptation parameter set with aps\_id equal to aps\_ref\_idx is referenced by this adaptation parameter set. When aps\_ref\_idx equal to aps\_id, copying rather than referencing of adaptation parameter set data is performed.

On receipt of an adaptation parameter set RBSP, the following actions are performed:

* + If aps\_ref\_flag is equal to 1 and aps\_ref\_idx is equal to aps\_id, or there is no previous knowledge of an adaptation parameter set with the same aps\_id, the contents of the RBSP shall be known as APS[ aps\_id ].
  + Otherwise, the contents of the received adaptation parameter set RBSP is merged with the previous version according to mergeAPS( APS[ aps\_id ], aps\_rbsp ), as described in subsection XX [7.4.1.2.1 ?].

**aps\_scaling\_list\_data\_present\_flag** equal to 1 specifies that scaling list parameters are present in this adaptation parameter set, equal to 0 specifies that scaling list parameters are not present in this adaptation parameter set.

**aps\_deblocking\_filter\_data\_present\_flag** equal to 1 specifies that the deblocking filter parameters are present in this adaptation parameter set, equal to 0 specifies that the deblocking filter parameters are not present in this adaptation parameter set.

**aps\_sample\_adaptive\_offset\_data\_present\_flag** equal to 1 specifies that SAO parameters are present in this adaptation parameter set, equal to 0 specifies that SAO parameters are not present in this adaptation parameter set.

**aps\_adaptive\_loop\_filter\_data\_present\_flag** equal to 1 specifies that the ALF parameters are present in this adaptation parameter set, equal to 0 specifies that the ALF parameters are not present in this adaptation parameter set.

**aps\_extension\_present\_flag** equal to 0 specifies that no aps\_extension\_data\_flag syntax elements are present in the adaptation parameter set RBSP syntax structure. aps\_extension\_flag shall be equal to 0 in bitstreams conforming to this Recommendation | International Standard. The value of 1 for aps\_extension\_present\_flag is reserved for future use by ITU-T | ISO/IEC. Decoders shall ignore all data that follow the value 1 for aps\_extension\_present\_flag in an adaptation parameter set NAL unit.

**aps\_extension\_data\_flag** may have any value. It shall not affect the conformance to profiles specified in this Recommendation | International Standard.

Slice header semantics

**scaling\_list\_flag** equal to 1 specifies that the scaling matrix is applied for the current slice, equal to 0 specifies that the scaling matrix is not applied for the current slice. The value of scaling\_list\_flag\_flag shall be the same for all slices in the current frame.

**sample\_adaptive\_offset\_flag** equal to 1 specifies that sample adaptive offset is applied for the current slice, equal to 0 specifies that sample adaptive offset is not applied for the current slice. The value of sample\_adaptive\_offset\_flag shall be the same for all slices in the current frame.

**adaptive\_loop\_filter\_flag** equal to 1 specifies that the adaptive loop filter is applied for the current slice, equal to 0 specifies that the adaptive loop filter is not applied for the current slice. The value of adaptive\_loop\_filter\_flag shall be the same for all slices in the current frame.

In semantics

##### Order of sequence, picture and adaptation parameter set RBSPs and their activation

This subclause specifies the activation process of picture, sequence and adaptation parameter sets for coded video sequences that conform to one or more of the profiles specified in Annex XX that are decoded using the decoding process specified in clauses XX.

NOTE 1 – The sequence, picture and adaptation parameter set mechanism decouples the transmission of infrequently changing information from the transmission of coded macroblock data. Sequence, picture and adaptation parameter sets may, in some applications, be conveyed "out-of-band".

An adaptation parameter set RBSP includes parameters that can be referred to by the coded slice NAL units of one or more coded pictures when at least one of sample\_adaptive\_offset\_enabled\_flag or adaptive\_loop\_filter\_enabled\_flag are equal to 1. Each adaptation parameter set may contain data relating to a class of parameters, each indicated by a presence flag. Each adaptation parameter set may be updated using a partial update procedure specified in XXX, forming the array APS[apsId]. A construction process is performed at the start of each coded picture to populate the active adaptation parameter set from a list of adaptation parameter set structures that have been previously received. The resulting active adaptation parameter set is only considered active for a single coded picture.

When a coded slice NAL unit references (using a particular value of aps\_id) an adaptation parameter set (with that value of aps\_id), the adaptation parameter set RBSP does not itself become activated, rather, the active adaptation parameter set is constructed using the coalesceAPS(aps\_id, apsRecursionLimit) function with the value of apsRecursionLimit of XX (Ed.: or as determined from the profile\_idc and level\_idc as indicated in Annex A.)

A picture parameter set RBSP includes parameters that can be referred to by the coded slice NAL units or coded slice data partition A NAL units of one or more coded pictures. Each picture parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one picture parameter set RBSP is considered active at any given moment during the operation of the decoding process, and the activation of any particular picture parameter set RBSP results in the deactivation of the previously-active picture parameter set RBSP (if any).

When a picture parameter set RBSP (with a particular value of pic\_parameter\_set\_id) is not active and it is referred to by a coded slice NAL unit or coded slice data partition A NAL unit (using that value of pic\_parameter\_set\_id), it is activated. This picture parameter set RBSP is called the active picture parameter set RBSP until it is deactivated by the activation of another picture parameter set RBSP. A picture parameter set RBSP, with that particular value of pic\_parameter\_set\_id, shall be available to the decoding process prior to its activation.

Any picture parameter set NAL unit containing the value of pic\_parameter\_set\_id for the active picture parameter set RBSP for a coded picture shall have the same content as that of the active picture parameter set RBSP for the coded picture unless it follows the last VCL NAL unit of the coded picture and precedes the first VCL NAL unit of another coded picture.

A sequence parameter set RBSP includes parameters that can be referred to by one or more picture parameter set RBSPs or one or more SEI NAL units containing a buffering period SEI message. Each sequence parameter set RBSP is initially considered not active at the start of the operation of the decoding process. At most one sequence parameter set RBSP is considered active at any given moment during the operation of the decoding process, and the activation of any particular sequence parameter set RBSP results in the deactivation of the previously-active sequence parameter set RBSP (if any).

When a sequence parameter set RBSP (with a particular value of seq\_parameter\_set\_id) is not already active and it is referred to by activation of a picture parameter set RBSP (using that value of seq\_parameter\_set\_id) or is referred to by an SEI NAL unit containing a buffering period SEI message (using that value of seq\_parameter\_set\_id), it is activated. This sequence parameter set RBSP is called the active sequence parameter set RBSP until it is deactivated by the activation of another sequence parameter set RBSP. A sequence parameter set RBSP, with that particular value of seq\_parameter\_set\_id, shall be available to the decoding process prior to its activation. An activated sequence parameter set RBSP shall remain active for the entire coded video sequence.

NOTE 2 – Because an IDR access unit begins a new coded video sequence and an activated sequence parameter set RBSP must remain active for the entire coded video sequence, a sequence parameter set RBSP can only be activated by a buffering period SEI message when the buffering period SEI message is part of an IDR access unit.

Any sequence parameter set NAL unit containing the value of seq\_parameter\_set\_id for the active sequence parameter set RBSP for a coded video sequence shall have the same content as that of the active sequence parameter set RBSP for the coded video sequence unless it follows the last access unit of the coded video sequence and precedes the first VCL NAL unit and the first SEI NAL unit containing a buffering period SEI message (when present) of another coded video sequence.

NOTE 3 – If picture parameter set RBSP, sequence parameter set RBSP or adaptation parameter set RBSP are conveyed within the bitstream, these constraints impose an order constraint on the NAL units that contain the picture parameter set RBSP or sequence parameter set RBSP, respectively. Otherwise (picture parameter set RBSP, sequence parameter set RBSP or adaptation parameter set RBSP are conveyed by other means not specified in this Recommendation | International Standard), they must be available to the decoding process in a timely fashion such that these constraints are obeyed.

All constraints that are expressed on the relationship between the values of the syntax elements (and the values of variables derived from those syntax elements) in sequence parameter sets, picture parameter sets and adaptation parameter sets and other syntax elements are expressions of constraints that apply only to the active sequence parameter set, the active picture parameter set and the active adaptation parameter set. If any sequence parameter set RBSP is present that is not activated in the bitstream, its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise‑conforming bitstream. If any picture parameter set RBSP is present that is not ever activated in the bitstream, its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise-conforming bitstream. If any adaptation parameter set RBSP is present that is not ever activated in the bitstream, its syntax elements shall have values that would conform to the specified constraints if it were activated by reference in an otherwise-conforming bitstream.

During operation of the decoding process (see clause 0), the values of parameters of the active picture parameter set and the active sequence parameter set shall be considered in effect. For interpretation of SEI messages, the values of the parameters of the picture parameter set and sequence parameter set that are active for the operation of the decoding process for the VCL NAL units of the primary coded picture in the same access unit shall be considered in effect unless otherwise specified in the SEI message semantics.

# Conclusions

In order to facilitate flexible signalling of scaling list, SAO and ALF parameters in APS, an APS referencing approach is proposed. The recommendation is to adopt this approach to HEVC.

# References

1. Stephan Wenger, “JCT-VC break-out report: Adaptation Parameter Set issues”, JCT-VC Document JCTVC-G1016, November 2011, Geneva, Switzerland.
2. Stephan Wenger, Jill Boyce, Yu-Wen Huang, Chia-Yang Tsai, Ping Wu and Ming Li, “Adaptation Parameter Set (APS)”, JCT-VC Document JCTVC-F747, July 2011, Torino, Italy.
3. Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan and Thomas Wiegand, “WD5: Working Draft 5 of High-Efficiency Video Coding”, JCT-VC Document JCTVC-G1103\_d3, November 2011, Geneva, Switzerland.

# Patent rights declaration(s)

**ZTE Corporation may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

**Vidyo Inc. may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

**The British Broadcasting Corporation (BBC) may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**