|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  10th Meeting: Strasbourg, FR, 18–24 October 2014 | Document: JCT3V-J0109 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Update of SEI message in AVC/PDAM1** | | |
| *Status:* | Input Document of JCT-3V | | |
| *Purpose:* | Draft Text | | |
| *Author(s) or Contact(s):* | Takanori Senoh NICT  Peng Yin Dolby Laboratories, Inc.  Ying Chen Qualcomm  Miska M. Hannuksela Nokia  Jens-Rainer Ohm RWTH Aachen University  Gary J. Sullivan Microsoft | Email:  Email:  Email:  Email:  Email:  Email: | senoh@nict.go.jp  pyin@dolby.com  cheny@qti.qualcomm.com  miska.hannuksela@nokia.com  ohm@ient.rwth-aachen.de  garysull@microsoft.com |
| *Source:* | Editors | | |

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

# Abstract

This draft reports a revised version of the AVC/PDAM1: MFC plus depth and additional SEI message according to the PDAM1 voting comments from Spain and US.

© ISO/IEC 2013 – All rights reservedISO/IEC 14496-10:2012/PDAM 5 63Part 10: Advanced Video Coding, AMENDMENT 5: Multi-resolution frame compatible stereo codingInformation technology — Coding of audio-visual objectsÉlément introductif — Élément central — Partie 10: Titre de la partieInformation technology — Coding of audio-visual objects — Part 10: Advanced Video Coding, AMENDMENT 5: Multi-resolution frame compatible stereo codingE2013-02-15(30) CommitteeISO/IECISO/IEC J20125AmendmentInternational Standard2013 ISO/IEC 14496‑ISO/IEC 14496‑10ISO/IEC 14496-10/PDAM 5  Coding of audio, picture, multimedia and hypermedia informationInformation technology11291 2見出し 2見出し 1,Heading U,h1,H1,H11,Œ©o‚µ 1,?co??E 1,뙥,?c,?co?ƒÊ 1,?,Œ    02 STD Version 2.1c230   4C:\Users\shinji\_w\AppData\Roaming\Microsoft\Templates\STD\14496-10\_PDAM5.doc            ISO/IEC JTC 1/SC 29 N

Date:   2014-10-18

ISO/IEC 14496-10:201x/DAM 1

ISO/IEC JTC 1/SC 29/WG 11

Secretariat:

Information technology — Coding of audio-visual objects — Part 10: Advanced Video Coding, AMENDMENT 1: Multi-Resolution Frame Compatible Stereoscopic Video with Depth Maps, and Additional Supplemental Enhancement Information

*Élément introductif — Élément central — Partie 10: Titre de la partie*

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

**Copyright notice**

This ISO document is a working draft or committee draft and is copyright-protected by ISO. While the reproduction of working drafts or committee drafts in any form for use by participants in the ISO standards development process is permitted without prior permission from ISO, neither this document nor any extract from it may be reproduced, stored or transmitted in any form for any other purpose without prior written permission from ISO.

Requests for permission to reproduce this document for the purpose of selling it should be addressed as shown below or to ISO's member body in the country of the requester:

[Indicate the full address, telephone number, fax number, telex number, and electronic mail address, as appropriate, of the Copyright Manger of the ISO member body responsible for the secretariat of the TC or SC within the framework of which the working document has been prepared.]

Reproduction for sales purposes may be subject to royalty payments or a licensing agreement.

Violators may be prosecuted.

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 14496‑10:201x was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technolog — Coding of audio-visual objects — Part 10: Advanced Video Coding, AMENDMENT 1: Multi-Resolution Frame Compatible Stereoscopic Video with Depth Maps, and Additional Supplemental Enhancement Information

*In 7.3.2.1.1, replace the syntax table with the following:*

|  |  |  |
| --- | --- | --- |
| seq\_parameter\_set\_data( ) { | **C** | **Descriptor** |
| **profile\_idc** | 0 | u(8) |
| **constraint\_set0\_flag** | 0 | u(1) |
| **constraint\_set1\_flag** | 0 | u(1) |
| **constraint\_set2\_flag** | 0 | u(1) |
| **constraint\_set3\_flag** | 0 | u(1) |
| **constraint\_set4\_flag** | 0 | u(1) |
| **constraint\_set5\_flag** | 0 | u(1) |
| **reserved\_zero\_2bits** /\* equal to 0 \***/** | 0 | u(2) |
| **level\_idc** | 0 | u(8) |
| **seq\_parameter\_set\_id** | 0 | ue(v) |
| if( profile\_idc = = 100 | | profile\_idc = = 110 | |  profile\_idc = = 122 | | profile\_idc = = 244 | | profile\_idc = = 44 | |  profile\_idc = = 83 | | profile\_idc = = 86 | | profile\_idc = = 118 | |  profile\_idc = = 128 | | profile\_idc = = 138 | | profile\_idc = = 139 | |  profile\_idc = = 134 | | profile\_idc = = 135) { |  |  |
| **chroma\_format\_idc** | 0 | ue(v) |
| if( chroma\_format\_idc = = 3 ) |  |  |
| **separate\_colour\_plane\_flag** | 0 | u(1) |
| **bit\_depth\_luma\_minus8** | 0 | ue(v) |
| **bit\_depth\_chroma\_minus8** | 0 | ue(v) |
| **qpprime\_y\_zero\_transform\_bypass\_flag** | 0 | u(1) |
| **seq\_scaling\_matrix\_present\_flag** | 0 | u(1) |
| if( seq\_scaling\_matrix\_present\_flag ) |  |  |
| for( i = 0; i < ( ( chroma\_format\_idc  !=  3 ) ? 8 : 12 ); i++ ) { |  |  |
| **seq\_scaling\_list\_present\_flag[**i ] | 0 | u(1) |
| if( seq\_scaling\_list\_present\_flag[ i ] ) |  |  |
| if( i < 6 ) |  |  |
| scaling\_list( ScalingList4x4[ i ], 16,   UseDefaultScalingMatrix4x4Flag[ i ]) | 0 |  |
| else |  |  |
| scaling\_list( ScalingList8x8[ i − 6 ], 64,  UseDefaultScalingMatrix8x8Flag[ i − 6 ] ) | 0 |  |
| } |  |  |
| } |  |  |
| **log2\_max\_frame\_num\_minus4** | 0 | ue(v) |
| **pic\_order\_cnt\_type** | 0 | ue(v) |
| if( pic\_order\_cnt\_type = = 0 ) |  |  |
| **log2\_max\_pic\_order\_cnt\_lsb\_minus4** | 0 | ue(v) |
| else if( pic\_order\_cnt\_type = = 1 ) { |  |  |
| **delta\_pic\_order\_always\_zero\_flag** | 0 | u(1) |
| **offset\_for\_non\_ref\_pic** | 0 | se(v) |
| **offset\_for\_top\_to\_bottom\_field** | 0 | se(v) |
| **num\_ref\_frames\_in\_pic\_order\_cnt\_cycle** | 0 | ue(v) |
| for( i = 0; i < num\_ref\_frames\_in\_pic\_order\_cnt\_cycle; i++ ) |  |  |
| **offset\_for\_ref\_frame[** i **]** | 0 | se(v) |
| } |  |  |
| **max\_num\_ref\_frames** | 0 | ue(v) |
| **gaps\_in\_frame\_num\_value\_allowed\_flag** | 0 | u(1) |
| **pic\_width\_in\_mbs\_minus1** | 0 | ue(v) |
| **pic\_height\_in\_map\_units\_minus1** | 0 | ue(v) |
| **frame\_mbs\_only\_flag** | 0 | u(1) |
| if( !frame\_mbs\_only\_flag ) |  |  |
| **mb\_adaptive\_frame\_field\_flag** | 0 | u(1) |
| **direct\_8x8\_inference\_flag** | 0 | u(1) |
| **frame\_cropping\_flag** | 0 | u(1) |
| if( frame\_cropping\_flag ) { |  |  |
| **frame\_crop\_left\_offset** | 0 | ue(v) |
| **frame\_crop\_right\_offset** | 0 | ue(v) |
| **frame\_crop\_top\_offset** | 0 | ue(v) |
| **frame\_crop\_bottom\_offset** | 0 | ue(v) |
| } |  |  |
| **vui\_parameters\_present\_flag** | 0 | u(1) |
| if( vui\_parameters\_present\_flag ) |  |  |
| vui\_parameters( ) | 0 |  |
| } |  |  |

*In 7.3.2.1.3, replace the syntax table with the following:*

|  |  |  |
| --- | --- | --- |
| subset\_seq\_parameter\_set\_rbsp( ) { | **C** | **Descriptor** |
| seq\_parameter\_set\_data( ) | 0 |  |
| if( profile\_idc = = 83 | | profile\_idc = = 86 ) { |  |  |
| seq\_parameter\_set\_svc\_extension( ) /\* specified in Annex G \*/ | 0 |  |
| **svc\_vui\_parameters\_present\_flag** | 0 | u(1) |
| if( svc\_vui\_parameters\_present\_flag = = 1 ) |  |  |
| svc\_vui\_parameters\_extension( ) /\* specified in Annex G \*/ | 0 |  |
| } else if( profile\_idc = = 118 | | profile\_idc = = 128 | |  profile\_idc = = 134 ) { |  |  |
| **bit\_equal\_to\_one** /\* equal to 1 \*/ | 0 | f(1) |
| seq\_parameter\_set\_mvc\_extension( ) /\* specified in Annex H \*/ | 0 |  |
| **mvc\_vui\_parameters\_present\_flag** | 0 | u(1) |
| if( mvc\_vui\_parameters\_present\_flag = = 1 ) |  |  |
| mvc\_vui\_parameters\_extension( ) /\* specified in Annex H \*/ | 0 |  |
| } else if( profile\_idc = = 138 | | profile\_idc = = 135) { |  |  |
| **bit\_equal\_to\_one** /\* equal to 1 \*/ | 0 | f(1) |
| seq\_parameter\_set\_mvcd\_extension( ) /\* specified in Annex I \*/ |  |  |
| } else if( profile\_idc = = 139 ) { |  |  |
| **bit\_equal\_to\_one** /\* equal to 1 \*/ | 0 | f(1) |
| seq\_parameter\_set\_mvcd\_extension( ) /\* specified in Annex I \*/ |  |  |
| seq\_parameter\_set\_3davc\_extension( ) /\* specified in Annex J \*/ | 0 |  |
| } |  |  |
| **additional\_extension2\_flag** | 0 | u(1) |
| if( additional\_extension2\_flag = = 1 ) |  |  |
| while( more\_rbsp\_data( ) ) |  |  |
| **additional\_extension2\_data\_flag** | 0 | u(1) |
| rbsp\_trailing\_bits( ) | 0 |  |
| **}** |  |  |

*Add the following equation, with appropriate equation numbering in sequence, into subclause I.7.4.2.1.4, just before the semantics of applicable\_op\_depth\_flag[ i ][ j ][ k ], and increment subsequent equation numbers in Annex I to account for the presence of the additional equation.*

The variable AllViewsPairedFlag is derived as follows:

AllViewsPairedFlag = 1  
for( i = 1; i <= num\_views\_minus1; i++ )  
 AllViewsPairedFlag = ( AllViewsPairedFlag & depth\_view\_present\_flag[ i ] & (I-xx)  
 texture\_view\_present\_flag[ i ] )

*Add a new subclause I.10.1.2 “MFC Depth High profile” as follows:*

**I.10.1.2 MFC Depth High profile**

Bitstreams conforming to the MFC Depth High profile shall obey the following constraints:

– The base view bitstream as specified in clause I.8.5.4 shall obey all constraints of the High profile specified in clause A.2.4, and all active sequence parameter sets shall fulfill one of the following conditions:

– profile\_idc is equal to 77 or constraint\_set1\_flag is equal to 1,

– profile\_idc is equal to 100.

– The sub-bitstream of stereoscopic texture bitstream, as specified in clause I.8.5.5, shall obey all constraints of the MFC High profile specified in clause H.10.1.3 and all active MVC sequence parameter sets shall fulfill one of the following conditions:

– profile\_idc is equal to 134,

– profile\_idc is equal to 100,

– profile\_idc is equal to 77 or constraint\_set1\_flag is equal to 1.

– Only I, P, and B slice types may be present.

– NAL unit streams shall not contain nal\_unit\_type values in the range of 2 to 4, inclusive.

– Arbitrary slice order is not allowed.

– Picture parameter sets shall have num\_slice\_groups\_minus1 equal to 0 only.

– Picture parameter sets shall have redundant\_pic\_cnt\_present\_flag equal to 0 only.

– When frame\_mbs\_only\_flag is equal to 1 in an active sequence parameter set for a texture view, frame\_mbs\_only\_flag shall be equal to 1 in the active sequence parameter set for the depth view having the same view\_id.

– When frame\_mbs\_only\_flag is equal to 0 in an active sequence parameter set for a depth view, mb\_adaptive\_frame\_field\_flag shall be equal to 0.

– MVCD sequence parameter sets for the depth views shall have chroma\_format\_idc equal to 0 only.

– MVCD sequence parameter sets shall have bit\_depth\_luma\_minus8 equal to 0 only.

– MVCD sequence parameter sets shall have bit\_depth\_chroma\_minus8 equal to 0 only.

– MVCD sequence parameter sets shall have qpprime\_y\_zero\_transform\_bypass\_flag equal to 0 only.

– For each access unit, the value of level\_idc for all active view MVCD sequence parameter set RBSPs shall be the same as the value of level\_idc for the active MVCD sequence parameter set RBSP.

– The level constraints specified for the MFC Depth High profile in clause I.10.2 shall be fulfilled.

Conformance of a bitstream to the MFC Depth High profile is indicated by profile\_idc being equal to 135.

Decoders conforming to the MFC Depth High profile at a specific level shall be capable of decoding all bitstreams in which both of the following conditions are true:

1. All active MVCD sequence parameter sets have one or more of the following conditions fulfilled:

– profile\_idc is equal to 135,

– profile\_idc is equal to 138,

– profile\_idc is equal to 134,

– profile\_idc is equal to 128,

– profile\_idc is equal to 118 and constraint\_set5\_flag is equal to 1,

– profile\_idc is equal to 100,

– profile\_idc is equal to 77 or constraint\_set1\_flag is equal to 1.

1. All active MVCD sequence parameter sets have one or more of the following conditions fulfilled:

– level\_idc or (level\_idc and constraint\_set3\_flag) represent a level less than or equal to the specific level,

– level\_idc[ i ] or (level\_idc[ i ] and constraint\_set3\_flag) represent a level less than or equal to the specific level.

*In J.7.3.2.1.5, replace the syntax table with the following:*

|  |  |  |
| --- | --- | --- |
| seq\_parameter\_set\_3davc\_extension( ) { | **C** | **Descriptor** |
| if( NumDepthViews > 0 ) { |  |  |
| **3dv\_acquisition\_idc** | 0 | ue(v) |
| for( i = 0; i < NumDepthViews; i++ ) |  |  |
| **view\_id\_3dv**[ i ] | 0 | ue(v) |
| if( 3dv\_acquisition\_idc ) { |  |  |
| depth\_ranges( NumDepthViews, 2, 0 ) |  |  |
| vsp\_param( NumDepthViews, 2, 0  ) |  |  |
| } |  |  |
| **reduced\_resolution\_flag** | 0 | u(1) |
| if( reduced\_resolution\_flag ) { |  |  |
| **depth\_pic\_width\_in\_mbs\_minus1** | 0 | ue(v) |
| **depth\_pic\_height\_in\_map\_units\_minus1** | 0 | ue(v) |
| **depth\_hor\_mult\_minus1** | 0 | ue(v) |
| **depth\_ver\_mult\_minus1** | 0 | ue(v) |
| **depth\_hor\_rsh** | 0 | ue(v) |
| **depth\_ver\_rsh** | 0 | ue(v) |
| } |  |  |
| **depth\_frame\_cropping\_flag** | 0 | u(1) |
| if( depth\_frame\_cropping\_flag ) { |  |  |
| **depth\_frame\_crop\_left\_offset** | 0 | ue(v) |
| **depth\_frame\_crop\_right\_offset** | 0 | ue(v) |
| **depth\_frame\_crop\_top\_offset** | 0 | ue(v) |
| **depth\_frame\_crop\_bottom\_offset** | 0 | ue(v) |
| } |  |  |
| **grid\_pos\_num\_views** | 0 | ue(v) |
| for( i = 0; i < grid\_pos\_num\_views; i++ ) { |  |  |
| **grid\_pos\_view\_id**[ i ] | 0 | ue(v) |
| **grid\_pos\_x**[ grid\_pos\_view\_id[ i ] ] | 0 | se(v) |
| **grid\_pos\_y**[ grid\_pos\_view\_id[ i ] ] | 0 | se(v) |
| } |  |  |
| **slice\_header\_prediction\_flag** | 0 | u(1) |
| **seq\_view\_synthesis\_flag** | 0 | u(1) |
| } |  |  |
| **alc\_sps\_enable\_flag** | 0 | u(1) |
| **enable\_rle\_skip\_flag** | 0 | u(1) |
| if( !AllViewsPairedFlag ) { |  |  |
| for( i = 1; i <= num\_views\_minus1; i++ ) |  |  |
| if( texture\_view\_present\_flag[ i ] ) { |  |  |
| **num\_anchor\_refs\_l0**[ i ] | 0 | ue(v) |
| for( j = 0; j < num\_anchor\_refs\_l0[ i ]; j++ ) |  |  |
| **anchor\_ref\_l0**[ i ][ j ] | 0 | ue(v) |
| **num\_anchor\_refs\_l1**[ i ] | 0 | ue(v) |
| for( j = 0; j < num\_anchor\_refs\_l1[ i ]; j++ ) |  |  |
| **anchor\_ref\_l1**[ i ][ j ] | 0 | ue(v) |
| } |  |  |
| for( i = 1; i <= num\_views\_minus1; i++ ) |  |  |
| if( texture\_view\_present\_flag[ i ] ) { |  |  |
| **num\_non\_anchor\_refs\_l0**[ i ] | 0 | ue(v) |
| for( j = 0; j < num\_non\_anchor\_refs\_l0[ i ]; j++ ) |  |  |
| **non\_anchor\_ref\_l0**[ i ][ j ] | 0 | ue(v) |
| **num\_non\_anchor\_refs\_l1**[ i ] | 0 | ue(v) |
| for( j = 0; j < num\_non\_anchor\_refs\_l1[ i ]; j++ ) |  |  |
| **non\_anchor\_ref\_l1**[ i ][ j ] | 0 | ue(v) |
| } |  |  |
| } |  |  |
| } |  |  |

*Add the following at the end of subclause J.7.4.2.1.5:*

For num\_anchor\_refs\_l0[ i ], anchor\_ref\_l0[ i ][ j ], num\_anchor\_refs\_l1[ i ], anchor\_ref\_l1[ i ][ j ], num\_non\_anchor\_refs\_l0[ i ], non\_anchor\_ref\_l0[ i ][ j ], num\_non\_anchor\_refs\_l1[ i ], and non\_anchor\_ref\_l1[ i ][ j ], the semantics specified in subclause H.7.4.2.1.4 is applied with the substitution of texture view component for view component. When num\_anchor\_refs\_l0[ i ], anchor\_ref\_l0[ i ][ j ], num\_anchor\_refs\_l1[ i ], anchor\_ref\_l1[ i ][ j ], num\_non\_anchor\_refs\_l0[ i ], non\_anchor\_ref\_l0[ i ][ j ], num\_non\_anchor\_refs\_l1[ i ], and non\_anchor\_ref\_l1[ i ][ j ] are not present, they are inferred to have the same values as the respective syntax elements in the seq\_parameter\_set\_mvcd\_extension( ) syntax structure in the same subset\_seq\_parameter\_set\_rbsp( ) syntax structure that also contains this seq\_parameter\_set\_3davc\_extension( ) syntax structure.

*In Table D.1, add the following after “depth\_view\_info( payloadSize )”:*

|  |  |  |
| --- | --- | --- |
| else if( payloadType = = 56 ) |  |  |
| texture\_and\_depth\_view\_packing( payloadSize ) /\* specified in Annex J \*/ | 5 |  |

*Add a new subclause J.13.1.2 “Texture and depth view packing SEI message syntax” as follows:*

* + - 1. **Texture and depth view packing SEI message syntax**

|  |  |  |
| --- | --- | --- |
| texture\_and\_depth\_view\_packing( payloadSize ) { | **C** | **Descriptor** |
| **depth\_type** | 5 | ue(v) |
| if( depth\_type = = 0 ) { |  |  |
| **num\_residual\_texture\_views\_minus1** | 5 | ue(v) |
| **residual\_depth\_flag** | 5 | u(1) |
| **z\_flag** | 5 | u(1) |
| **intrinsic\_param\_flag** | 5 | u(1) |
| **rotation\_flag** | 5 | u(1) |
| **translation\_flag** | 5 | u(1) |
| if( z\_flag ) |  |  |
| for( i = 0; i <= num\_residual\_texture\_views\_minus1 + 1; i++ ) { |  |  |
| **z\_near\_sign\_flag[**i**]** | 5 | u(1) |
| **z\_near\_exponent[**i**]** | 5 | u(7) |
| **z\_near\_mantissa\_len\_minus1[**i**]** | 5 | u(5) |
| **z\_near mantissa[**i**]** | 5 | u(v) |
| **z\_far\_sign\_flag[**i**]** | 5 | u(1) |
| **z\_far\_exponent[**i**]** | 5 | u(7) |
| **z\_far\_mantissa\_len\_minus1[**i**]** | 5 | u(5) |
| **z\_far\_mantissa[**i**]** | 5 | u(v) |
| } |  |  |
| if( intrinsic\_param\_flag ) { |  |  |
| **prec\_focal\_length** | 5 | ue(v) |
| **prec\_principal\_point** | 5 | ue(v) |
| } |  |  |
| if( rotation\_flag ) |  |  |
| **prec\_rotation\_param** | 5 | ue(v) |
| if ( translation\_flag ) |  |  |
| **prec\_translation\_param** | 5 | ue(v) |
| for( i = 0; i <= num\_residual\_texture\_views\_minus1 + 1; i++ ) { |  |  |
| if( intrinsic\_param\_flag ) { |  |  |
| **sign\_focal\_length\_x**[ i ] | 5 | u(1) |
| **exponent\_focal\_length\_x**[ i ] | 5 | u(6) |
| **mantissa\_focal\_length\_x**[ i ] | 5 | u(v) |
| **sign\_focal\_length\_y**[ i ] | 5 | u(1) |
| **exponent\_focal\_length\_y**[ i ] | 5 | u(6) |
| **mantissa\_focal\_length\_y**[ i ] | 5 | u(v) |
| **sign\_principal\_point\_x**[ i ] | 5 | u(1) |
| **exponent\_principal\_point\_x**[ i ] | 5 | u(6) |
| **mantissa\_principal\_point\_x**[ i ] | 5 | u(v) |
| **sign\_principal\_point\_y**[ i ] | 5 | u(1) |
| **exponent\_principal\_point\_y**[ i ] | 5 | u(6) |
| **mantissa\_principal\_point\_y**[ i ] | 5 | u(v) |
| } |  |  |
| if( rotation\_flag ) |  |  |
| for ( j = 1; j <= 3; j++ ) /\* row \*/ |  |  |
| for( k = 1; k <= 3; k++ ) { /\* column \*/ |  |  |
| **sign\_r**[ i ][ j ][ k ] | 5 | u(1) |
| **exponent\_r**[ i ][ j ][ k ] | 5 | u(6) |
| **mantissa\_r**[ i ][ j ][ k ] | 5 | u(v) |
| } |  |  |
| if ( translation\_flag ) { |  |  |
| **sign\_t\_x**[ i ] | 5 | u(1) |
| **exponent\_t\_x**[ i ] | 5 | u(6) |
| **mantissa\_t\_x**[ i ] | 5 | u(v) |
| } |  |  |
| } |  |  |
| } |  |  |
| **}** |  |  |



*Add a new subclause J.13.2.2 “Texture and depth view packing SEI message semantics” as follows, with "J-xx" replaced with the appropriate table number in the sequence of tables present in the text.*

* + - 1. **Texture and depth view packing SEI message semantics**

The texture and depth view packing SEI message informs the decoder that view components of one output view contain samples of multiple distinct spatially packed constituent residual pictures. The related output view is not suitable to be displayed directly and may be converted to displayable views by using this SEI message. This information can be used after the decoder output to appropriately rearrange the samples to produce displayable views that are appropriate for display or other purposes (which are outside the scope of this Recommendation | International Standard).

The texture and depth view packing SEI message, when present, shall be associated with an IDR access unit. The information signalled in this SEI message applies to all the access units in the coded video sequence. When this SEI message is present, num\_anchor\_refs\_l0[ i ], num\_anchor\_refs\_l1[ i ], num\_non\_anchor\_refs\_l0[ i ] and num\_non\_anchor\_refs\_l1[ i ] shall be equal to 0 in the active SPS RBSP syntax structure for the coded video sequence and avc\_3d\_extension\_flag shall be 0 in all VCL NAL units of the coded video sequence.

NOTE – These constraints disable inter-view and inter-component prediction.

In such a coded video sequence, there are at least two output views, wherein the base view is coded as a displayable view and a non-base view is a residual view. A residual view is not a displayable view. Instead, it contains a sequence of difference pictures which are decoded view components of the residual view. Each difference picture is created by a frame packing arrangement containing up to four constituent residual pictures.

Each constituent residual picture represents a picture corresponding to a 2:1 decimation from the residual picture horizontally and vertically, starting from the top-left-most sample.

The parameter i (with a value from 0 to num\_residual\_texture\_views\_minus1 + 1, inclusive) in the texture and depth view packing SEI message syntax table indicates the location of the constituent residual pictures in the non-base texture view as shown in Table J-xx. In this SEI message syntax table, i equal to 0 indicates the base texture view.

Let the output non-base texture view have a width equal to W and a height equal to H for the luma component. The texture and depth view packing SEI message shall not be present unless W is a multiple of 2 \* SubWidthC and H is a multiple of 2 \* SubHeightC.

**Table J**-**xx – Top-left corner coordinates of constituent residual pictures packed in a residual view of width W and height H.**

|  |  |
| --- | --- |
| i | Top-left corner coordinate |
| 1 | (0, 0) |
| 2 | (H/2, 0) |
| 3 | (0, W/2) |
| 4 | (H/2, W/2) |

**depth\_type** specifies the arrangement of how distinct spatially packed constituent residual pictures are packed into one view component of the non-base view. The values of depth\_type shall be equal to 0. Other values are reserved for future use by ITU-T | ISO/IEC and shall not be present in bitstreams conforming to this Specification. Decoders shall ignore texture and depth view packing SEI messages in which such other values are present.

**num\_residual\_texture\_views\_minus1** plus 1 specifies the number of constituent residual texture pictures packed into each texture view component of a non-base view. num\_residual\_texture\_views\_minus1 shall be in the range of 0 to 3, inclusive.

**residual\_depth\_flag** equal to 1 specifies that residual depth pictures are packed into the depth components of the non-base view, with a packing scheme that is identical to the one used for the residual texture pictures.

Each constituent residual depth picture of the non-base depth view component is associated with a constituent residual picture of the non-base texture view component in the same relative location. The number of the coded depth views in the coded video sequence shall be equal to 1 + residual\_depth\_flag. The depth components of the base view shall always be present, independent of the value of residual\_depth\_flag.

**z\_flag** equal to 1 indicates the presence of the syntax elements z\_near\_sign\_flag[ i ], z\_near\_exponent[ i ], z\_near\_mantissa\_len\_minus1[ i ], z\_near mantissa[ i ], z\_far\_sign\_flag[ i ], z\_far\_exponent[ i ], z\_far\_mantissa\_len\_minus1[ i ], and z\_far\_mantissa[ i ], for i ranging from 0 to num\_residual\_texture\_views\_minus1 + 1. z\_flag equal to 0 indicates that these syntax elements are not present.

**intrinsic\_param\_flag** equal to 1 indicates the presence of intrinsic camera parameter syntax elements. intrinsic\_param\_flag equal to 0 indicates that these syntax elements are not present.

**rotation\_flag** equal to 1 indicates the presence of rotation camera parameter syntax elements. rotation\_flag equal to 0 indicates that these syntax elements are not present. When rotation\_flag is 0, a default rotation camera parameter of a unit matrix value is inferred.

**translation\_flag** equal to 1 indicates the presence of horizontal translation camera parameter syntax elements. translation\_flag equal to 0 indicates that these syntax elements are not present.

**z\_near\_sign\_flag**[i] equal to 0 indicates that the sign of the nearest depth value of the i-th camera is positive. z\_near\_sign\_flag[ i ] equal to 1 indicates that the sign of the nearest depth value of the i-th camera is negative.

**z\_near\_exponent**[ i ] specifies the exponent part of the nearest depth value of the i-th camera. The value of z\_near\_exponent[ i ] shall be in the range of 0 to 27−2, inclusive. The value 27−1 is reserved for future use by ITU‑T | ISO/IEC. When z\_near\_exponent[ i ] is equal to 27−1, the value of z near[ i ] is unspecified.

**z\_near\_mantissa\_len\_minus1**[ i ] + 1 specifies the number of bits in the mantissa of the nearest depth value of the i-th camera. The value of z\_near\_mantissa\_len\_minus1 [ i ] shall be in the range of 0 to 31, inclusive.

**z\_near\_mantissa**[ i ] specifies the mantissa part of the nearest depth value of the i-th camera.

**z\_far\_sign\_flag**[i] equal to 0 indicates that the sign of the farthest depth value of the i-th camera is positive. z\_far\_sign\_flag[ i ] equal to 1 indicates that the sign of the farthest depth value of the i-th camera is negative.

**z\_far\_exponent**[ i ] specifies the exponent part of the farthest depth value of the i-th camera. The value of z\_far\_exponent[ i ] shall be in the range of 0 to 27−2, inclusive. The value 27−1 is reserved for future use by ITU‑T | ISO/IEC. When z\_far\_exponent[ i ] is equal to 27−1, the value of z far[ i ] is unspecified.

**z\_far\_mantissa\_len\_minus1**[ i ] + 1 specifies the number of bits in the mantissa of the farthest depth value of the i-th camera. The value of mantissa\_len\_minus1\_z\_far[ i ] shall be in the range of 0 to 31, inclusive.

**z\_far\_mantissa**[ i ] specifies the mantissa part of the farthest depth value of the i-th camera.

**prec\_focal\_length** specifies the exponent of the maximum allowable truncation error for focal\_length\_x[ i ] and focal\_length\_y[ i ] as given by 2−prec\_focal\_length. The value of prec\_focal\_length shall be in the range of 0 to 31, inclusive.

**prec\_principal\_point** specifies the exponent of the maximum allowable truncation error for principal\_point\_x[ i ] and principal\_point\_y[ i ] as given by 2−prec\_principal\_point. The value of prec\_principal\_point shall be in the range of 0 to 31, inclusive.

**prec\_rotation\_param** specifies the exponent of the maximum allowable truncation error for r[ i ][ j ][ k ] as given by 2−prec\_rotation\_param. The value of prec\_rotation\_param shall be in the range of 0 to 31, inclusive.

**prec\_translation\_param s**pecifies the exponent of the maximum allowable truncation error for t\_x[i] as given by 2−prec\_translation\_param. The value of prec\_ translation\_param shall be in the range of 0 to 31, inclusive.

**sign\_focal\_length\_x**[ i ] equal to 0 indicates that the sign of the focal length of the i-th camera in the horizontal direction is positive. sign\_focal\_length\_x[ i ] equal to 1 indicates that the sign of the focal length of the i-th camera in the horizontal direction is negative.

**exponent\_focal\_length\_x**[ i ] specifies the exponent part of the focal length of the i-th camera in the horizontal direction. The value of exponent\_focal\_length\_x[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU‑T | ISO/IEC. When exponent\_focal\_length\_x[ i ] is equal to 63, the value of focal length of the horizontal direction for the i-th camera is unspecified.

**mantissa\_focal\_length\_x**[ i ] specifies the mantissa part of the focal length of the i-th camera in the horizontal direction. The length of the mantissa\_focal\_length\_x[ i ] syntax element is determined as follows:

− If exponent\_focal\_length\_x[ i ] = = 0, the length is Max( 0, prec\_focal\_length − 30 ).

− Otherwise (0 < exponent\_focal\_length\_x[ i ] < 63), the length is Max( 0, exponent\_focal\_length\_x[ i ] + prec\_focal\_length − 31 ).

**sign\_focal\_length\_y**[ i ] equal to 0 indicates that the sign of the focal length of the i-th camera in the vertical direction is positive. sign\_focal\_length\_y[ i ] equal to 1 indicates that the sign of the focal length of the i-th camera in the vertical direction is negative.

**exponent\_focal\_length\_y**[ i ] specifies the exponent part of the focal length of the i-th camera in the vertical direction. The value of exponent\_focal\_length\_y[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU‑T | ISO/IEC. When exponent\_focal\_length\_y[ i ] is equal to 63, the value of focal length of the vertical direction is unspecified.

**mantissa\_focal\_length\_y**[ i ] specifies the mantissa part of the focal length of the i-th camera in the vertical direction.

The length of the mantissa\_focal\_length\_y[i ] syntax element is determined as follows:

– If exponent\_focal\_length\_y[ i ] = = 0, the length is Max( 0, prec\_focal\_length − 30 ).

– Otherwise (0 < exponent\_focal\_length\_y[ i ] < 63), the length is Max( 0, exponent\_focal\_length\_y[ i ] + prec\_focal\_length − 31 ).

**sign\_principal\_point\_x**[ i ] equal to 0 indicates that the sign of the principal point of the i-th camera in the horizontal direction is positive. sign\_principal\_point\_x[ i ] equal to 1 indicates that the sign of the principal point of the i-th camera in the horizontal direction is negative.

**exponent\_principal\_point\_x**[ i ] specifies the exponent part of the principal point of the i-th camera in the horizontal direction. The value of exponent\_principal\_point\_x[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. When exponent\_principal\_point\_x[ i ] is equal to 63, the value of principal point in the horizontal direction for the i-th camera is unspecified.

**mantissa\_principal\_point\_x**[ i ] specifies the mantissa part of the principal point of the i-th camera in the horizontal direction The length of the mantissa\_principal\_point\_x[i ] syntax element in units of bits is determined as follows:

– If exponent\_principal\_point\_x[ i ] = = 0, the length is Max( 0, prec\_principal\_point − 30 ).

– Otherwise (0 < exponent\_principal\_point\_x[ i ] < 63), the length is Max( 0, exponent\_principal\_point\_x[ i ] + prec\_principal\_point − 31 ).

**sign\_principal\_point\_y**[ i ] equal to 0 indicates that the sign of the principal point of the i-th camera in the vertical direction is positive. sign\_principal\_point\_y[ i ] equal to 1 indicates that the sign of the principal point of the i-th camera in the vertical direction is negative.

**exponent\_principal\_point\_y**[ i ] specifies the exponent part of the principal point of the i-th camera in the vertical direction. The value of exponent\_principal\_point\_y[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU-T | ISO/IEC. When exponent\_principal\_point\_y[ i ] is equal to 63, the value of principal point in the vertical direction for the i-th camera is unspecified.

**mantissa\_principal\_point\_y**[ i ] specifies the mantissa part of the principal point of the i-th camera in the vertical direction. The length of the mantissa\_principal\_point\_y[i ] syntax element in units of bits is determined as follows:

– If exponent\_principal\_point\_y[ i ] = = 0, the length is Max( 0, prec\_principal\_point − 30 ).

– Otherwise (0 < exponent\_principal\_point\_y[ i ] < 63), the length is Max( 0, exponent\_principal\_point\_y[ i ] + prec\_principal\_point − 31 ).

**sign\_r**[ i ][ j ][ k ] equal to 0 indicates that the sign of (j, k) component of the rotation matrix for the i-th camera is positive. sign\_r[ i ][ j ][ k ] equal to 1 indicates that the sign of (j, k) component of the rotation matrix for the i-th camera is negative.

**exponent\_r**[ i ][ j ][ k ] specifies the exponent part of (j, k) component of the rotation matrix for the i-th camera. The value of exponent\_r[ i ][ j ][ k ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU‑T | ISO/IEC. When exponent\_r[ i ][ j ][ k ] is equal to 63, the value of rotation matrix is unspecified.

**mantissa\_r**[ i ][ j ][ k ] specifies the mantissa part of (j, k) component of the rotation matrix for the i-th camera.

The length of the mantissa\_r[ i ][ j ][ k ] syntax element in units of bits is determined as follows:

– If exponent\_r[ i ] = = 0, the length is Max( 0, prec\_rotation\_param − 30 ).

– Otherwise (0 < exponent\_r[ i ] < 63), the length is Max( 0, exponent\_r[ i ] + prec\_rotation\_param − 31 ).

The rotation matrix R[ i ] for i-th camera is represented as follows:

 (J-XX)

**sign\_t\_x**[ i ] equal to 0 indicates that the sign of the horizontal component of the translation vector for the i-th camera is positive. sign\_t\_x[ i ] equal to 1 indicates that the sign of the horizontal component of the translation vector for the i-th camera is negative.

**exponent\_t\_x**[ i ] specifies the exponent part of the horizontal component of the translation vector for the i-th camera. The value of exponent\_t\_x[ i ] shall be in the range of 0 to 62, inclusive. The value 63 is reserved for future use by ITU‑T | ISO/IEC. When exponent\_t\_x[ i ] is equal to 63, the value of the translation vector is unspecified.

**mantissa\_t\_x**[ i ] specifies the mantissa part of the horizontal component of the translation vector for the i-th camera.

The length v of the mantissa\_t\_x[ i ] syntax element in units of bits is determined as follows:

– If exponent\_t\_x[ i ] = = 0, the length v = Max( 0, prec\_translation\_param − 30 ).

– Otherwise (0 < exponent\_t\_x[ i ] < 63), the length v = Max( 0, exponent\_t\_x[ i ] + prec\_translation\_param − 31 ).

The variable x, where x refers to a focal length, principal point, rotation, or translation is computed as follows, where e = exponent, s = sign, n = mantissa and v = mantissa\_len\_minus1 + 1:

– If 0 < e < 63, x = (−1)s \* 2e−31 \* (1 + n  2v).

– Otherwise (e is equal to 0), x = (−1)s \* 2−(30+v) \* n.

NOTE – The above specification is similar to what found in IEC 60559:1989, *Binary floating-point arithmetic for microprocessor systems*.

*Add a new subclause J.13.2.2.1 “Reconstruction of the displayable texture views (Informative)” as follows:*

**J.13.2.2.1 Displayable texture view reconstruction process (Informative)**

Inputs to this process are:

* decoded base view (texture and depth), decoded non-base view (texture and depending on the residual\_depth\_flag, depth)
* decoded SEI message (GVDViewSynth.cfg GVDCameraParam.txt and GVDCameraParam.CSV)

GVDViewSynth.cfg contains parameters for depth\_type, num\_residual\_texture\_views\_minus1, residual\_depth\_flag, and parameters for view synthesis. GVDCameraParam.txt contains intrinsic camera parameters (focal length, principal point co-ordinates) and extrinsic camera parameters (rotation matrix and x-translation). GVDCameraParam.CSV contains Znear, Zfar and x-translation.

Output of this process is a reconstructed displayable view (texture)

The base view is denoted as the 0-th view, corresponding to the 0-th camera, and the view B is denoted as the i-th view corresponding to the i-th camera. i is in the range of 1 through 4, inclusive. An informative process to reconstruct the displayable texture views from the constituent residual pictures in the non-base view as well as the pictures in the base view is summarized in the following paragraph for each view B which is packed as a constituent residual picture in the non-base view.

For i=1 to i <= num\_residual\_texture\_views\_minus1 + 1, repeat the following process.

1. If residual\_depth\_flag is equal to 1, the constituent residual depth picture associated with a view B is extracted out of the non-base depth view according to the value of i.
   1. The constituent residual depth picture is 1:2 upsampled horizontally and vertically to a residual depth view B.
2. By using the decoded SEI message, the base depth view is projected to the coordinates of the residual depth view B.
   1. Samples of residual depth view B which have smaller depth values are overwritten by the larger values of the projected depth samples.
   2. Remaining holes are inpainted.
   3. If num\_residual\_texture\_views\_minus1 + 2 is an even number, the base depth view is projected to the coordinates of the residual depth view B from its center location between base view and view B of i equal to 1.
3. The resulted depth view and the base depth view are upsampled to the same resolution of the base-texture view.
4. The constituent residual texture picture associated with view B is extracted from the non-base texture view and is 1:2 upsampled horizontally and vertically to a residual picture view B.
5. Using the resulted depth view B and the decoded SEI message, the base texture view is projected to the coordinates of the residual picture B.
6. The remaining holes are inpainted and output it as the displayable view B.