|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  33rd Meeting: Macao, CN, 6–12 Oct. 2018 | Document: JCTVC-AG0021-v1 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Some errata items for HEVC and AVC** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | **Ye-Kui Wang** Huawei Technologies Co., Ltd. Telesis Court, San Diego CA 92121, USA  **Alexis M. Tourapis** Apple Inc. 1 Infinite Loop Cupertino, CA 95014, USA  **Gary Sullivan** Microsoft Corp 1 Microsoft Way Redmond, WA 90852 USA | Tel: Email: | +1-858-754-3673 [yekui.wang@huawei.com](mailto:yekui.wang@huawei.com)  [atourapis@apple.com](mailto:atourapis@apple.com)  +1 (425) 703-5308 [garysull@microsoft.com](mailto:garysull@microsoft.com) |
| *Source:* | Huawei Technologies Co., Ltd., Apple Inc., Microsoft Corp | | |

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

# Abstract

This contribution proposes two sets of changes, one on semantics of the frame packing arrangement SEI message and another on the definition and use of the square root function. Both sets apply to both HEVC and AVC.

# For AVC

## The first set of changes

*In 5.7, replace the following definition:*

Sqrt( x ) = 

*with the following:*

Sqrt( x ) returns the square root of x.

*In D.2.35.6.2, replace the following equation:*

lw = pictureWidth / 3  
lh = pictureHeight / 2  
w = Floor( hPos ÷ lw )  
h = Floor( vPos ÷ lh )  
tmpHorVal = hPos − w \* lw  
tmpVerVal = vPos − h \* lh  
hPos′ = −( 2 \* tmpHorVal ÷ lw ) + 1  
vPos′ = −( 2 \* tmpVerVal ÷ lh ) + 1  
if( w = = 1 && h = = 0 ) { /\* positive x front face \*/  
 x = 1.0  
 y = hPos′  
 z = vPos′  
} else if( w = = 1 && h = = 1 ) { /\* negative x back face \*/  
 x = −1.0  
 y = −vPos′  
 z = −hPos′  
} else if( w = = 2 && h = = 1 ) { /\* positive z top face \*/ (D‑XX)  
 x = −hPos′  
 y = −vPos′  
 z = 1.0  
} else if( w = = 0 && h = = 1 ) { /\* negative z bottom face \*/  
 x = hPos′  
 y = −vPos′  
 z = −1.0  
} else if( w = = 0 && h = = 0 ) { /\* positive y left face \*/  
 x = −hPos′  
 y = 1.0  
 z = vPos′  
} else { /\* ( w = = 2 && h = = 0 ), negative y right face \*/  
 x = hPos′  
 y = −1.0  
 z = vPos′  
}  
ϕ = Atan2( y, x ) \* 180 ÷ π  
θ =

*with the following (to use the defined function Sqrt( ) instead of the square root symbol):*

lw = pictureWidth / 3  
lh = pictureHeight / 2  
w = Floor( hPos ÷ lw )  
h = Floor( vPos ÷ lh )  
tmpHorVal = hPos − w \* lw  
tmpVerVal = vPos − h \* lh  
hPos′ = −( 2 \* tmpHorVal ÷ lw ) + 1  
vPos′ = −( 2 \* tmpVerVal ÷ lh ) + 1  
if( w = = 1 && h = = 0 ) { /\* positive x front face \*/  
 x = 1.0  
 y = hPos′  
 z = vPos′  
} else if( w = = 1 && h = = 1 ) { /\* negative x back face \*/  
 x = −1.0  
 y = −vPos′  
 z = −hPos′  
} else if( w = = 2 && h = = 1 ) { /\* positive z top face \*/ (D‑XX)  
 x = −hPos′  
 y = −vPos′  
 z = 1.0  
} else if( w = = 0 && h = = 1 ) { /\* negative z bottom face \*/  
 x = hPos′  
 y = −vPos′  
 z = −1.0  
} else if( w = = 0 && h = = 0 ) { /\* positive y left face \*/  
 x = −hPos′  
 y = 1.0  
 z = vPos′  
} else { /\* ( w = = 2 && h = = 0 ), negative y right face \*/  
 x = hPos′  
 y = −1.0  
 z = vPos′  
}  
ϕ = Atan2( y, x ) \* 180 ÷ π  
θ =

## The second set of changes

*In D.2.26 (Frame packing arrangement SEI message semantics), replace the semantics of frame\_packing\_arrangement\_type, including Table D.9, with the following:*

**frame\_packing\_arrangement\_type** identifies the indicated interpretation of the sample arrays of the output cropped decoded picture as specified in Table D-9.

When frame\_packing\_arrangement\_type is in the range of 0 to 4, inclusive, or equal to 7, each component plane of the output cropped decoded picture contains all samples (when field\_pic\_flag is equal to 0) or the samples corresponding to the top or bottom field (when field\_pic\_flag is equal to 1) of the samples of a frame packing arrangement structure.

**Table D‑9 – Definition of frame\_packing\_arrangement\_type**

|  |  |
| --- | --- |
| **Value** | **Interpretation** |
| 0 | The frame packing arrangement structure contains a "checkerboard" based interleaving of corresponding planes of two constituent frames as illustrated in Figure D‑1. |
| 1 | The frame packing arrangement structure contains a column based interleaving of corresponding planes of two constituent frames as illustrated in Figure D‑2 and Figure D‑3. |
| 2 | The frame packing arrangement structure contains a row based interleaving of corresponding planes of two constituent frames as illustrated in Figure D‑4 and Figure D‑5. |
| 3 | The frame packing arrangement structure contains a side-by-side packing arrangement of corresponding planes of two constituent frames as illustrated in Figure D‑6, Figure D‑7, and Figure D‑10. |
| 4 | The frame packing arrangement structure contains a top-bottom packing arrangement of corresponding planes of two constituent frames as illustrated in Figure D‑8 and Figure D‑9. |
| 5 | The component planes of the output cropped decoded pictures in output order form a temporal interleaving of alternating first and second constituent frames as illustrated in Figure D‑11. |
| 6 | The output cropped decoded picture constitutes a complete 2D frame (when field\_pic\_flag is equal to 0) or field (when field\_pic\_flag is equal to 1) without any frame packing. |
| 7 | The frame packing arrangement structure contains a tile format packing arrangement of corresponding planes of two constituent frames as illustrated in Figure D 12. |

*In D.2.26 (Frame packing arrangement SEI message semantics), remove the following NOTE and adjust the indices of the remaining NOTEs in the clause:*

NOTE 6 – frame\_packing\_arrangement\_type equal to 6 is used to signal the presence of 2D content (that is not frame packed) in 3D services that use a mix of 2D and 3D content. The frame\_packing\_arrangement\_type value of 6 should only be used with pictures that have field\_pic\_flag equal to 0.

# For HEVC

## The first set of changes

*In 5.8, replace the following definition:*

Sqrt( x ) = 

*with the following:*

Sqrt( x ) returns the square root of x.

*In D.3.41.2, replace the following equation:*

lw = pictureWidth / 3  
lh = pictureHeight / 2  
w = Floor( hPos ÷ lw )  
h = Floor( vPos ÷ lh )  
tmpHorVal = hPos − w \* lw  
tmpVerVal = vPos − h \* lh  
hPos′ = −( 2 \* tmpHorVal ÷ lw ) + 1  
vPos′ = −( 2 \* tmpVerVal ÷ lh ) + 1  
if( w = = 1 && h = = 0 ) { /\* positive x front face \*/  
 x = 1.0  
 y = hPos′  
 z = vPos′  
} else if( w = = 1 && h = = 1 ) { /\* negative x back face \*/  
 x = −1.0  
 y = −vPos′  
 z = −hPos′  
} else if( w = = 2 && h = = 1 ) { /\* positive z top face \*/ (D‑57)  
 x = −hPos′  
 y = −vPos′  
 z = 1.0  
} else if( w = = 0 && h = = 1 ) { /\* negative z bottom face \*/  
 x = hPos′  
 y = −vPos′  
 z = −1.0  
} else if( w = = 0 && h = = 0 ) { /\* positive y left face \*/  
 x = −hPos′  
 y = 1.0  
 z = vPos′  
} else { /\* ( w = = 2 && h = = 0 ), negative y right face \*/  
 x = hPos′  
 y = −1.0  
 z = vPos′  
}  
ϕ = Atan2( y, x ) \* 180 ÷ π  
θ =

*with the following (to use the defined function Sqrt( ) instead of the square root symbol):*

lw = pictureWidth / 3  
lh = pictureHeight / 2  
w = Floor( hPos ÷ lw )  
h = Floor( vPos ÷ lh )  
tmpHorVal = hPos − w \* lw  
tmpVerVal = vPos − h \* lh  
hPos′ = −( 2 \* tmpHorVal ÷ lw ) + 1  
vPos′ = −( 2 \* tmpVerVal ÷ lh ) + 1  
if( w = = 1 && h = = 0 ) { /\* positive x front face \*/  
 x = 1.0  
 y = hPos′  
 z = vPos′  
} else if( w = = 1 && h = = 1 ) { /\* negative x back face \*/  
 x = −1.0  
 y = −vPos′  
 z = −hPos′  
} else if( w = = 2 && h = = 1 ) { /\* positive z top face \*/ (D‑57)  
 x = −hPos′  
 y = −vPos′  
 z = 1.0  
} else if( w = = 0 && h = = 1 ) { /\* negative z bottom face \*/  
 x = hPos′  
 y = −vPos′  
 z = −1.0  
} else if( w = = 0 && h = = 0 ) { /\* positive y left face \*/  
 x = −hPos′  
 y = 1.0  
 z = vPos′  
} else { /\* ( w = = 2 && h = = 0 ), negative y right face \*/  
 x = hPos′  
 y = −1.0  
 z = vPos′  
}  
ϕ = Atan2( y, x ) \* 180 ÷ π  
θ =

## The second set of changes

*In D.3.16 (Frame packing arrangement SEI message semantics), replace the semantics of frame\_packing\_arrangement\_type, including Table D.8, with the following:*

**frame\_packing\_arrangement\_type** identifies the indicated interpretation of the sample arrays of the output cropped decoded picture as specified in Table D.8.

When frame\_packing\_arrangement\_type is equal to 3 or 4, each component plane of the output cropped decoded picture contains all samples (when field\_pic\_flag is equal to 0) or the samples corresponding to the top or bottom field (when field\_pic\_flag is equal to 1) of the samples of a frame packing arrangement structure.

Table D.8 – Definition of frame\_packing\_arrangement\_type

|  |  |
| --- | --- |
| **Value** | **Interpretation** |
| 3 | The frame packing arrangement structure contains a side-by-side packing arrangement of corresponding planes of two constituent frames as illustrated in Figure D.4, Figure D.5 and Figure D.8. |
| 4 | The frame packing arrangement structure contains a top-bottom packing arrangement of corresponding planes of two constituent frames as illustrated in Figure D.6 and Figure D.7. |
| 5 | The component planes of the output cropped decoded pictures in output order form a temporal interleaving of alternating first and second constituent frames as illustrated in Figure D.9. |

# Patent rights declaration(s)

**Huawei Technologies Co., Ltd., Apple Inc., and Microsoft Corp do not have any current or pending patent rights relating to the technology described in this contribution (to the extent of the personal awareness of the authors).**