**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG/M5XXXX**

**June 2020, Online**

|  |  |
| --- | --- |
| **Source** | **Interdigital** |
| **Status** | **Input document** |
| **Title** | **Per patch depth quantization** |
| **Author** | Bertrand Chupeau, Franck Thudor, Renaud Doré |

# Introduction

This contribution provides syntax for per patch depth quantization, in line with the fifth working draft [[N19212](http://wg11.sc29.org/doc_end_user/current_document.php?id=74671&id_meeting=182)]. This HLS enables depth quantization on a patch level [[m54419](http://wg11.sc29.org/doc_end_user/documents/131_OnLine/wg11/m54419-v1-m54419_CE112.zip) CE1.1.2].

It takes into account the feedbacks from CE1.1.2 m54419, but also the feedback from [m54176 CE1.2](http://wg11.sc29.org/doc_end_user/documents/131_OnLine/wg11/m54176-v1-m54176_CE1.zip) “geometry scaling” which has been said by the group to be correlated. We understand that the outcomes from the discussion – including on Gitlab - are the following:

* Instead of performing a depth quantization optimization per atlas, as proposed by m54176, it is cleaner to do it at the view level. There will be in next specification an update of quantization parameters per view, as a cleaner way to get Poznan m54176 objective improvements. A spec change update has been proposed by Bart on Gitlab cf [m54766](http://wg11.sc29.org/doc_end_user/documents/131_OnLine/wg11/m54766-v2-m54766dqupdates.zip).
* There is the perspective to get an even higher gain by quantizing depth per patch as proposed in Interdigital m54419 CE1.1.2.
  + Of course, it could be formally possible to artificially increase the number of virtual views and allocate patches to them (with the extreme situation of 1 patch to be specifically quantized = 1 new view) but this is not considered as a clean way.
  + It is observed that the specification could also be improved for the pdu\_depth\_start and pdu\_depth\_end parameters which are under-used today, and could be used to convey the per patch quantization parameters, and if done, would address the uniform depth patch implementation for the same price.
  + The group does not like to have a pair of floats at the patch\_data\_unit level.

Therefore, this contribution proposes to:

* Convey the low and end clipping values by pushing them at the view level in the depth quantization structure, under the names (**dq\_depth\_start, dq\_depth\_end)**
* Reuse the placeholders (**pdu\_depth\_start, pdu\_depth\_end**) now made available by placing here the pairs of minimum and maximum normalized disparity per patch (**pdu\_norm\_disp\_low, pdu\_norm\_disp\_high)**, when this mode is activated.
  + The name of this variable pair and the related semantics is changed.
  + The normalized disparity low and high at the view level is proposed to be maintained just because it could be useful for other means, otherwise it should be exclusive (this needs to be discussed)
  + The coding is no more float but is similar to what previously used for (**pdu\_depth\_start, pdu\_depth\_end**).
* In that situation, the newly introduced feature “uniform depth patches” are simply signaled by setting **pdu\_norm\_disp\_low = pdu\_norm\_disp\_high.**

# Proposal

## Modify clause 7.3.6.3.6

|  |  |
| --- | --- |
| depth\_quantization( v ) { | **Descriptor** |
| **dq\_quantization\_law**[ v ] | u(8) |
| if( dq\_quantization\_law[ v ] == 0 ) { |  |
| **dq\_norm\_disp\_low**[ v ] | fl(32) |
| **dq\_norm\_disp\_high**[ v ] | fl(32) |
| **dq\_patchwise\_depth\_quantization\_flag**[ v ] | u(1) |
| } |  |
| **dq\_depth\_occ\_threshold\_default**[ v ] | ue(v) |
| **dq\_depth\_start**[ v ] | u(v) |
| **dq\_depth\_end**[ v ] | u(v) |
| } |  |

## Modify clause 7.4.6.3.6

**dq\_patchwise\_depth\_quantization\_flag**[ v ] equal to 1 specifies that the minimum and maximum normalized disparity values used by the quantization law are defined on a per patch basis for the patches of the view with index equal to v. dq\_patchwise\_depth\_quantization\_flag[ v ] equal to 0 specifies that the minimum and maximum normalized disparity values used by the quantization law are dq\_norm\_disp\_low[ v ] and dq\_norm\_disp\_high[ v ], respectively, for all the patches of the view. When not present, the value of dq\_patchwise\_depth\_quantization\_flag is inferred to be equal to 0.

**dq\_depth\_ start**[ v ] specifies the start of the range of quantized depth values for the patches of the view with index equal to v.

**dq\_depth\_end**[ v ] specifies the end of the range of quantized depth values for the patches of the view with index equal to v.

## Modify clause 7.3.7.3

|  |  |
| --- | --- |
| patch\_data\_unit( p ) { | **Descriptor** |
| **pdu\_2d\_pos\_x**[ p ] | ue(v) |
| **pdu\_2d\_pos\_y**[ p ] | ue(v) |
| **pdu\_2d\_size\_x\_minus1**[ p ] | ue(v) |
| **pdu\_2d\_size\_y\_minus1**[ p ] | ue(v) |
| **pdu\_view\_pos\_x**[ p ] /\* new semantics for pdu\_3d\_pos\_x \*/ | u(v) |
| **pdu\_view\_pos\_y**[ p ] /\* new semantics for pdu\_3d\_pos\_y \*/ | u(v) |
| **pdu\_norm\_disp\_low**[ p ] /\* new semantics for pdu\_3d\_pos\_min\_z \*/ | u(v) |
| if( asps\_normal\_axis\_max\_delta\_value\_enabled\_flag ) |  |
| **pdu\_norm\_disp\_high**[ p ] /\* new semantics for pdu\_3d\_pos\_delta\_max\_z \*/ | u(v) |
| **pdu\_view\_id**[ p ] /\* new semantics for pdu\_projection\_id \*/ | u(v) |
| **pdu\_orientation\_index**[ p ] | u(v) |
| if( afps\_lod\_mode\_enabled\_flag ) { |  |
| **pdu\_lod\_enabled\_flag**[ p ] | u(1) |
| if( pdu\_lod\_enabled\_flag[ p ] > 0 ) { |  |
| **pdu\_lod\_scale\_x\_minus1**[ p ] | ue(v) |
| **pdu\_lod\_scale\_y\_idc**[ p ] | ue(v) |
| } |  |
| } | u(v) |
| if( asps\_point\_local\_reconstruction\_enabled\_flag ) |  |
| point\_local\_reconstruction\_data( p ) |  |
| if( asps\_miv\_extension\_flag ) |  |
| pdu\_miv\_extension( p ) |  |
| } |  |

## Modify Clause 7.4.7.3

**pdu\_norm\_disp\_low**[ p ] is used to derive the minimum normalized disparity value in meters-1 PduNormDispLow[ p ] for the patch with index equal to p.

PduNormDispLow[ p ] = (pdu\_norm\_disp\_low[ p ] << ath\_pos\_min\_z\_quantizer)

÷ (2gi\_geometry\_3d\_coordinates\_bitdepth\_minus1 + 1 -1)

**pdu\_** **norm\_disp\_high**[ p ] is used to derive the maxim normalized disparity value in meters-1 PduNormDispHigh[ p ] for the patch with index equal to p.

PduNormDispHigh[ p ] = (pdu\_norm\_disp\_high[ p ] << ath\_pos\_delta\_max\_z\_quantizer )

÷ (2gi\_geometry\_3d\_coordinates\_bitdepth\_minus1 + 1 -1)

## Modify clause H.4 Depth decoding process

(…)

MaxDepthSampleValue = 1 << (gi\_geometry\_nominal\_2d\_bitdepth\_minus1[ a ] + 1) – 1

bSz = AtlasPatchPackingBlockSize[ a ]

p = BlockToPatchMap[ y / bSz ][ x / bSz ]

if( p != -1 ) {

v = pdu\_view\_id[ a ][ p ]

ClampedDepthSample[ y ][ x ] = Clip(GeoFrame[ 0 ][ orderIdx ][ 0 ][ y ][ x ],

dq\_depth\_ start[ v ], dq\_depth\_end[ v ] )

DepthAtlasNormValue[ y ][ x ] = ClampedDepthSample[ y ][ x ] ÷ MaxDepthSampleValue

}

Then normalized disparity, NormDisp[ y ][ x ] , is derived as follows:

if( p != -1 ) {

v = pdu\_view\_id[ a ][ p ]

if( dq\_patchwise\_depth\_quantization\_flag[ v ] == 0 )  
 NormDisp[ y ][ x ] = dq\_norm\_disp\_low[ v ] + (dq\_norm\_disp\_high[ v ] – dq\_norm\_disp\_low[ v ])  
 \* DepthAtlasNormValue[ y ][ x ]

else

NormDisp[ y ][ x ] = PduNormDispLow[ a ][ p ] +

(PduNormDispHigh[ a ][ p ] – PduNormDispLow[ a ][ p ]) \* DepthAtlasNormValue[ y ][ x ]

}

(…)