Joint Collaborative Team on 3D Video Coding Extension Development of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11
2nd Meeting: Shanghai, CN, 13–19 Oct. 2012

Document: JCT3V-B0164
M27008

Title: **3D-HEVC HLS on depth definition**

Status: Input Document

Purpose: Proposal

Author(s) or Contact(s): Shinya Shimizu[1], Olgierd Stankiewicz[2], Shiori Sugimoto[1], Hideaki Kimata[1], Krzysztof Wegner[2], Marek Domański[2]

Tel: +81-46-859-2703
Email: shimizu.shinya@lab.ntt.co.jp
ostank@multimedia.edu.pl

Source: [1]NTT Corporation, [2]Poznań University of Technology

---

Abstract

This contribution proposes a definition of depth representation information in order to support many kinds of multiview video plus depth map other than those used in the development of MV-HEVC and 3D-HTM. The proposed depth representation information includes four kinds of representation types: disparity, an inverse of Z value, Z value and non-linearly transformed disparity. Since in MV-HEVC no coding tool requires understanding of the definition of the depth, this contribution proposes to define depth representation information as a SEI message.

Such a SEI message has already been adopted to 3D-AVC Draft Text [4] and 3D-ATM.

1 Introduction

There is no definition of the depth in either text of [1] pr [2]. However, it seems necessary to inform what kind of information is transmitted in order to utilize in the appropriate manner. We extend the definition proposed in [3] to include some other popular depth representations. In this contribution, we would like to discuss the better place where the depth representation type defined. In addition, we'd also like to discuss the depth-range-based weighted prediction.

2 Definition of Depth - Depth representation type

In [3], two depth representation types are proposed to be defined: 1) an inverse of Z value and 2) disparity. The first one is the most familiar type since depth information obtained by using pixel-based estimation is normally defined with this type. The depth information from Microsoft KINECT is also based on this type. Although depth information in the second type can be derived by scaling those in the first one, it might be very useful for disparity-based display systems like multiview auto-stereoscopic display. In addition to these types, this contribution proposes to define distance, i.e. Z value itself as the third type. This type of depth representation is used in the Time-On-Flight (TOF) camera system and the computer graphic. Therefore, this contribution proposes to define a new syntax element, *depth_representaion_type*, to specify three depth representation types.

In [4], also an another depth representation type is proposed - nonlinearly transformed disparity. Such approach has been found to be efficient for 3D-video coding and become one of coding tools in 3D-ATM [4].

Besides the kind of the depth information, it is also necessary to specify some additional information. As proposed in [3], if the depth representation type is based on disparity, it is necessary to indicate the base view which the displacement is derived from. If the depth representation type is either an inverse of Z value or Z value itself, the definition of Z-axis is necessary. Considering the acquisition method, it is

normal to define Z-axis is defined as the optical axis of the own camera/sensor. However, it is also popular to share one axis among multi-camera in order to make it easier to process depth information on the 3D space although direct acquisition of such information is impossible. Therefore, this contribution proposes to define two new syntax elements, *all_views_equal_flag* and *depth_representaion_base_view_id*. *all_views_equal_flag* specifies whether a common base view is used or not. *depth_representaion_base_view_id* specifies disparity base view when the depth representation type is disparity or z-axis when the depth representation type is either an inverse of Z value or Z value.

The proposed syntax is as follows:

depth_represention_information(payloadSize) {	C	**Descriptor**
depth_representation_type	5	ue(v)
all_views_equal_flag	5	u(1)
if(all_views_equal_flag = = 0){		
num_views_minus1	5	ue(v)
numViews = num_views_minus1 + 1		
}else{		
numViews = 1		
}		
for(i = 0; i < numViews; i++) {		
depth_representation_base_view_id[i]	5	ue(v)
}		
if (depth_representation_type==3) {		
depth_nonlinear_representation_num_minus1		ue(v)
depth_nonlinear_representation_num = depth_nonlinear_representation_num_minus1+1		
for(i = 1; i <= depth_nonlinear_representation_num; i++)		
depth_nonlinear_representation_model[i]		ue(v)
}		
}		

depth_representation_type specifies the representation definition of luma pixels in coded frame of depth views as specified in Table I-XX. In Table I-XX, disparity specifies the horizontal displacement between two texture views and Z value specifies the distance from a camera.

Table I-XX Definition of depth_representation_type

depth_representation_type	Interpretation
0	Each luma pixel value in coded frame of depth views represents an inverse of Z value normalized in range from 0 to 255
1	Each luma pixel value in coded frame of depth views represents disparity normalized in range from 0 to 255
2	Each luma pixel value in coded frame of depth views represents Z value normalized in range from 0 to 255
3	Each luma pixel value in coded frame of depth views represents nonlinearly mapped disparity, normalized in range from 0 to 255.
Other values	Reserved for future use

all_views_equal_flag equal to 0 specifies that depth representation base view may not be identical to respective values for each view in target views. all_views_equal_flag equal to 1 specifies that the depth representation base views are identical to respective values for all target views.

num_views_minus1 shall be equal to the value of the syntax element num_views_minus1 in the 3DVC sequence parameter set for the coded video sequence. The value of num_views_minus1 shall be in the range of 0 to 1023, inclusive.

depth_representaion_base_view_id[i] specifies the view identifier for the NAL unit of either base view which the disparity for coded depth frame of i-th view_id is derived from (depth_representation_type equal to 1 or 3) or base view which the Z-axis for the coded depth frame of i-th view_id is defined as the optical axis of (depth_representation_type equal to 0 or 2).

depth_nonlinear_representation_num_minus1 + 2 specifies the number of piecewise linear segments for mapping of depth values to a scale that is uniformly quantized in terms of disparity.

depth_nonlinear_representation_model[i] specifies the piecewise linear segments for mapping of depth values to a scale that is uniformly quantized in terms of disparity.

NOTE – When depth_representation_type is equal to 3, depth view component contains nonlinearly transformed depth samples. Variable DepthLUT [i], as specified below, is used to transform coded depth sample values from nonlinear representation to the linear representation – disparity normalized in range from 0 to 255. The shape of this transform is defined by means of line-segment-approximation in two-dimensional linear-disparity-to-nonlinear-disparity space. The first (0, 0) and the last (255, 255) nodes of the curve are predefined. Positions of additional nodes are transmitted in form of deviations (depth_nonlinear_representation_model[i]) from the straight-line curve. These deviations are uniformly distributed along the whole range of 0 to 255, inclusive, with spacing depending on the value of nonlinear_depth_representation_num.

Variable DepthLUT[i] for i in the range of 0 to 255, inclusive, is specified as follows.

```
depth_nonlinear_representation_model[ 0 ] = 0
depth_nonlinear_representation_model[depth_nonlinear_representation_num + 1 ] = 0
for( k=0; k<= depth_nonlinear_representation_num; ++k )
{
    pos1 = ( 255 * k ) / (depth_nonlinear_representation_num + 1 )
    dev1 = depth_nonlinear_representation_model[ k ]
    pos2 = ( 255 * ( k+1 ) ) / (depth_nonlinear_representation_num + 1 ) )
    dev2 = depth_nonlinear_representation_model[ k+1 ]

    x1 = pos1 - dev1
    y1 = pos1 + dev1
    x2 = pos2 - dev2
    y2 = pos2 + dev2

    for ( x = max( x1, 0 ); x <= min( x2, 255 ); ++x )
        DepthLUT[ x ] = Clip3( 0, 255, Round( ( ( x - x1 ) * ( y2 - y1 ) ) ÷ ( x2 - x1 ) + y1 ) )
}
```

3 Conclusion

The contribution proposed a definition of depth representation type.

We recommend discussing the right place to define the depth representation type.

4 Patent rights declaration

NTT 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).

Poznań University of Technology 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).

References

[1] Gerhard Tech, Krzysztof Wegner, Ying Chen, Sehoon Yea, "3D-HEVC Test Model 1", JCT3V-A1005, Stockholm, Sweden, 2012.

[2] Gerhard Tech, Krzysztof Wegner, Ying Chen, Miska Hannuksela, " MV-HEVC Working Draft 1", JCT3V-A1004, Stockholm, Sweden, 2012.

[3] Sally Hattori, and Yoshitomo Takahashi, "3D-AVC HLS: Modifications on Depth Acquisition Information SEI message", ISO/IEC JTC1/SC29/WG11 MPEG2012/M24730, April 2012, Geneva, Switzerland.

[4] M. Hannuksela, Y. Chen, T. Suzuki (Editors), "3D-AVC Draft Text 3", JCT3V-A1002, Stockholm, Sweden, 2012.