(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 September 2015 (24.09.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/139183 A1

(51) International Patent Classification: *H04N 7/24* (2011.01)

(21) International Application Number: PCT/CN2014/073547

(22) International Filing Date: 17 March 2014 (17.03.2014)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **MEDIATEK INC.** [CN/CN]; No.1, Dusing Road 1st, Science-Based Industrial Park, Hsin-Chu, Taiwan 300 (CN).

(72) Inventors: **LIN, Jian-Liang**; No.133, Neipi Rd., Nanjian Vil. Su'ao Township, Yilan County, Taiwan 270 (CN). **CHEN, Yi-Wen**; 8F.,No.66, Ln.211, Yongkang Rd., Fengyuan Dist., Taichung City, Taiwan 420 (CN). **ZHANG, Xianguo**; Room 202, Unit 4, Building 45, Zuojiazhuang Beili, Chaoyang District, Beijing 100028 (CN). **ZHANG, Kai**; A1101, Dongfang Yujinxiang Building, No. 216 Tangli Road, Chaoyang District, Beijing 100013 (CN). **AN, Jicheng**; No. 44, Zhanlanguan Road, Xicheng District, Beijing 100037 (CN). **HUANG, Han**; No.3 Shangyuancun, Haidian District, Beijing 100044 (CN).

(74) Agent: **BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.**; 16th Fl., Block A, Corporate Square, No.35 Jinrong Street, Beijing 100033 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) Title: METHOD OF SIGNALING OF DEPTH-BASED BLOCK PARTITIONING MODE FOR THREE-DIMENSIONAL AND MULTI-VIEW VIDEO CODING

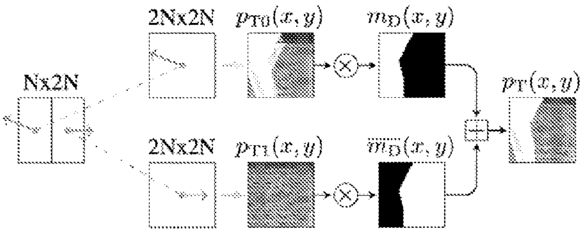

Fig. 3

(57) Abstract: They are methods for the DBBP mode signaling in multi-view and 3D video coding. It is proposed to always signal both DBBP flag and partition flag without dependence. It is also proposed to signal the DBBP flag before the partition flag. When the DBBP mode is enabled, the transmission of the partition flag is skipped.

WO 2015/139183 A1

METHOD OF SIGNALING OF DEPTH-BASED BLOCK PARTITIONING MODE FOR THREE-DIMENSIONAL AND MULTI-VIEW VIDEO CODING

TECHNICAL FIELD

[0001]The invention relates generally to Three-Dimensional (3D) video processing. In particular, the present invention relates to method for depth-based block partitioning (DBBP) signaling in 3D video coding.

BACKGROUND

[0002]Three-dimensional (3D) television has been a technology trend in recent years that is targeted to bring viewers sensational viewing experience. Multi-view video is a technique to capture and render 3D video. The multi-view video is typically created by capturing a scene using multiple cameras simultaneously, where the multiple cameras are properly located so that each camera captures the scene from one viewpoint. The multi-view video with a large number of video sequences associated with the views represents a massive amount data. Accordingly, the multi-view video will require a large storage space to store and/or a high bandwidth to transmit. Therefore, multi-view video coding techniques have been developed in the field to reduce the required storage space and the transmission bandwidth. A straightforward approach may simply apply conventional video coding techniques to each single-view video sequence independently and disregard any correlation among different views. Such straightforward techniques would result in poor coding performance. In order to improve multi-view video coding efficiency, multi-view video coding always exploits inter-view redundancy. The disparity between two views is caused by the locations and angles of the two respective cameras.

[0003]To reduce the inter-view redundancy, disparity-compensated prediction (DCP) has been used as an alternative to motion-compensated prediction (MCP). MCP refers to an inter-picture prediction that uses already coded pictures of the same view in a different access unit, while DCP refers to inter-picture prediction that uses already coded pictures of other views in the same access unit, as illustrated in Fig. 1. The three-dimensional/multi-view data consists of texture pictures (110) and depth maps (120). The motion compensated prediction is applied to texture pictures or depth maps in the temporal direction (i.e., the horizontal direction in Fig. 1). The disparity compensated prediction is applied to texture pictures or depth maps in the view direction (i.e., the vertical direction in Fig. 1). The vector used for DCP is termed disparity vector (DV), which is analog to the motion vector (MV) used in MCP.

[0004]3D-HEVC is an extension of HEVC (High Efficiency Video Coding) that is being developed for encoding/decoding 3D video. One of the views is referred to as the base view or the independent view. The base view is coded independently of the other views as well as the depth data. Furthermore, the base view is coded using a conventional HEVC video coder.

[0005]In 3D-HEVC, a hybrid block-based motion-compensated DCT-like transform coding architecture is still utilized. The basic unit for compression, termed coding unit (CU), is a 2N×2N square block, and each CU can be recursively split into four smaller CUs until the predefined minimum size is reached. Each CU contains one or multiple prediction units (PUs). The PU size can be 2N×2N, 2N×N, N×2N, or N×N. When asymmetric motion partition (AMP) is supported, the PU size can also be 2N×nU, 2N×nD, nL×2N and nR×2N.

[0006]In the depth-based block partitioning (DBBP) mode, an arbitrarily shaped block partitioning for the collocated texture block is derived based on a binary segmentation mask computed from the corresponding depth map. Each of the two partitions (resembling foreground and background) is motion compensated and afterwards merged based on the depth-based segmentation mask.

[0007]A single flag is added to the coding syntax to signal to the decoder that a block uses DBBP for prediction. When current coding unit is coded with the DBBP mode, the corresponding partition size is set to SIZE_2Nx2N and bi-prediction is inherited.

[0008]A disparity vector derived from the DoNBDV process is applied to identify a corresponding depth block in a reference view. The corresponding depth block has the same size as current texture block. When the depth block is found, a threshold is calculated based on the average of all depth pixels within the corresponding depth block. Afterwards, a binary segmentation mask m_D (x,y) is generated based on depth values and the threshold. When the depth value located at the relative coordinator (x, y) is larger than the threshold, the binary mask m_D (x,y) is set to 1. Otherwise, m_D (x,y) is set to 0. An example is shown in Fig. 2.

[0009]For each of the two decoded motion parameters a 2Nx2N motion compensation is performed. The resulting prediction signals p_T0 (x,y) and p_T1 (x,y) are combined using the DBBP mask m_D (x,y), as depicted in Fig. 3. The combination process is defined as follows:

p_T (x,y) = p_T0 (x,y), if m_D (x,y)=1

= p_T1 (x,y), otherwise

[0010] Whether the DBBP mode is used is signaled on coding unit as shown in Table 1. In currently design, the DBBP flag is conditionally signaled depended on a transmitted partition mode. The flag is signaled only when the transmitted PartMode equals to 2NxN partition.

Table 1

coding_unit(x0, y0, log2CbSize , ctDepth) {	**Descriptor**
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if((CuPredMode[x0][y0] != MODE_INTRA || log2CbSize = = MinCbLog2SizeY) && !predPartModeFlag)	
part_mode	ae(v)
if(depth_based_blk_part_flag[nuh_layer_id] && PartMode = = PART_2NxN)	
dbbp_flag[x0][y0]	u(1)
if(sdcEnableFlag)	
sdc_flag[x0][y0]	ae(v)
if(CuPredMode[x0][y0] = = MODE_INTRA) {	
if(PartMode = = PART_2Nx2N && pcm_enabled_flag && log2CbSize >= Log2MinIpcmCbSizeY && log2CbSize <= Log2MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned( ))	
pcm_alignment_zero_bit	f(1)
pcm_sample(x0, y0, log2CbSize)	
} else {	
pbOffset = (PartMode = = PART_NxN) ? (nCbS / 2) : nCbS	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset) {	
if(vps_depth_modes_flag[nuh_layer_id])	
intra_mode_ext(x0 + i , y0+ j , log2CbSize)	
if(dim_not_present_flag[x0 + i][y0 + j])	
prev_intra_luma_pred_flag[x0 + i][y0 + j]	ae(v)
}	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
if(dim_not_present_flag[x0 + i][y0 + j]) {	

if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
mpm_idx[x0 + i][y0 + j]	ae(v)
else	
rem_intra_luma_pred_mode[x0 + i][y0 + j]	ae(v)
}	
intra_chroma_pred_mode[x0][y0]	ae(v)
}	
} else {	
if(PartMode == PART_2Nx2N)	
prediction_unit(x0, y0, nCbS, nCbS)	
else if(PartMode == PART_2NxN) {	
prediction_unit(x0, y0, nCbS, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS, nCbS / 2)	
} else if(PartMode == PART_Nx2N) {	
prediction_unit(x0, y0, nCbS / 2, nCbS)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS)	
} else if(PartMode == PART_2NxnU) {	
prediction_unit(x0, y0, nCbS, nCbS / 4)	
prediction_unit(x0, y0 + (nCbS / 4), nCbS, nCbS * 3 / 4)	
} else if(PartMode == PART_2NxnD) {	
prediction_unit(x0, y0, nCbS, nCbS * 3 / 4)	
prediction_unit(x0, y0 + (nCbS * 3 / 4), nCbS, nCbS / 4)	
} else if(PartMode == PART_nLx2N) {	
prediction_unit(x0, y0, nCbS / 4, nCbS)	
prediction_unit(x0 + (nCbS / 4), y0, nCbS * 3 / 4, nCbS)	
} else if(PartMode == PART_nRx2N) {	
prediction_unit(x0, y0, nCbS * 3 / 4, nCbS)	
prediction_unit(x0 + (nCbS * 3 / 4), y0, nCbS / 4, nCbS)	
} else { /* PART_NxN */	
prediction_unit(x0, y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
}	
}	
}	
cu_extension(x0, y0)	
if(!cu_skip_flag[x0][y0]) {	
if(!pcm_flag[x0][y0] && !sdc_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]))	
rqt_root_cbf	ae(v)
if(rqt_root_cbf) {	
MaxTrafoDepth = (CuPredMode[x0][y0] == MODE_INTRA ? (max_transform_hierarchy_depth_intra + IntraSplitFlag) : max_transform_hierarchy_depth_inter)	
transform_tree(x0, y0, x0, y0, log2CbSize, 0, 0)	
}	
}	
}	
}	

depth_based_blk_part_flag[layerId] equal to 0 specifies that depth based block partitioning is not used for the layer with nuh_layer_id equal to layerId. depth_based_blk_part_flag[layerId] equal to 1 specifies that depth based block partitioning might be used for the layer with nuh_layer_id equal to layerId. When not present, the value of depth_based_blk_part_flag[layerId] is inferred to be equal to 0.

SUMMARY

[0011] In the current design (HTM-10.0), the DBBP flag is conditionally signaled depended on a transmitted partition mode. This invention proposes to improve the DBBP mode signaling in 3D video coding.

[0012] Other aspects and features of the invention will become apparent to those with ordinary skill in the art upon review of the following descriptions of specific embodiments.

BRIEF DESCRIPTION OF DRAWINGS

The invention can be more fully understood by reading the subsequent detailed description and examples with references made to the accompanying drawings, wherein:

[0013] Fig. 1 illustrates an example of three-dimensional/multi-view coding, where motion compensated prediction (MCP) and disparity compensated prediction (DCP) are used.

[0014] Fig. 2 illustrates an example of segmentation mask generation.

[0015] Fig. 3 illustrate an example of DBBP merging process.

DETAILED DESCRIPTION

[0016] The following description is of the best-contemplated mode of carrying out the invention. This description is made for the purpose of illustrating the general principles of the invention and should not be taken in a limiting sense. The scope of the invention is best determined by reference to the appended claims.

[0017] The basic concept of the invention is to improve the DBBP mode signaling in 3D video coding. Several methods are illustrated as below:

[0018] When current coding unit is coded with the DBBP mode, the corresponding partition size is set to SIZE_2N×2N and bi-prediction is inherited. Therefore, in one embodiment proposed in this invention, the DBBP flag is proposed to be signaled only when the transmitted partition mode is 2Nx2N partition mode. An example of signaling the DBBP flag when the partition mode is 2N×2N is shown in Table 2.

Table 2

coding_unit(x0, y0, log2CbSize , ctDepth) {	**Descriptor**
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if((CuPredMode[x0][y0] != MODE_INTRA || log2CbSize = = MinCbLog2SizeY) && !predPartModeFlag)	
part_mode	ae(v)
if(depth_based_blk_part_flag[nuh_layer_id] && PartMode = = PART_2Nx2N)	
dbbp_flag[x0][y0]	ae(v)
if(sdcEnableFlag)	
sdc_flag[x0][y0]	ae(v)

[0019]In another embodiment, the DBBP flag is proposed to be signaled only when the transmitted partition mode with the longest binarization codeword is transmitted. As an example shown in Table 3, the DBBP flag is signaled when the transmitted partition mode equal to AMP xxx partition mode. (xxx could be one of the following AMP partitioning modes: PART_2N×nU, PART_2N×nD, PART_nL×2N and PART_nR×2N).

Table 3

coding_unit(x0, y0, log2CbSize , ctDepth) {	**Descriptor**
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	

if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if((CuPredMode[x0][y0] != MODE_INTRA || log2CbSize = = MinCbLog2SizeY) && !predPartModeFlag)	
part_mode	ae(v)
if(depth_based_blk_part_flag[nuh_layer_id] && PartMode = = PART_AMPxxx)	
dbbp_flag[x0][y0]	ae(v)
if(sdcEnableFlag)	
sdc_flag[x0][y0]	ae(v)

[0020] In yet another embodiment of the present invention, the transmission of the DBBP flag doesn't not depend on transmitted partition mode. The DBBP flag is always signaled as long as the DBBP mode is enabled in current coding slice or picture. For example, when the flag depth_based_blk_part_flag[layerId], which is used to specified whether the depth based block partitioning is used or not for current layer, is true, the DBBP is always signaled as an example shown in Table 4.

Table 4

coding_unit(x0, y0, log2CbSize , ctDepth) {	**Descriptor**
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if((CuPredMode[x0][y0] != MODE_INTRA || log2CbSize = = MinCbLog2SizeY) && !predPartModeFlag)	
part_mode	ae(v)
if(depth_based_blk_part_flag[nuh_layer_id])	
dbbp_flag[x0][y0]	ae(v)
if(sdcEnableFlag)	
sdc_flag[x0][y0]	ae(v)

[0021] Since the mask for motion compensation and the partition for motion storage in DBBP are implicitly derived based on the corresponding depth block, the transmission of partition mode can actually be skipped when it is DBBP mode. In another embodiment of the present invention, the DBBP flag is transmitted before the partition flag. When the DBBP mode is enabled, the transmission of the partition flag is skipped. An example is shown in Table 5.

Table 5

coding_unit(x0, y0, log2CbSize , ctDepth) {	**Descriptor**
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if(depth_based_blk_part_flag[nuh_layer_id])	
dbbp_flag[x0][y0]	ae(v)
if((CuPredMode[x0][y0] != MODE_INTRA || log2CbSize = = MinCbLog2SizeY) && !predPartModeFlag && ! **dbbp_flag**[x0][y0])	
part_mode	ae(v)
if(sdcEnableFlag)	
sdc_flag[x0][y0]	ae(v)

[0022] In yet another embodiment of the present invention, the DBBP flag and the partition flag are both transmitted. When the DBBP mode is enabled, the partition flag is actually used to represent the partition for motion information storage in DBBP and is not used to represent the partition of motion compensation. The examples are shown in Table 4 and Table 6.

Table 6

coding_unit(x0, y0, log2CbSize , ctDepth) {	**Descriptor**
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	

else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if(depth_based_blk_part_flag[nuh_layer_id])	
dbbp_flag[x0][y0]	ae(v)
if((CuPredMode[x0][y0] != MODE_INTRA || log2CbSize = = MinCbLog2SizeY) && !predPartModeFlag)	
part_mode	ae(v)
if(sdcEnableFlag)	
sdc_flag[x0][y0]	ae(v)

[0023] In yet another embodiment of the present invention, the DBBP flag is transmitted when current CU is not predicted by an intra mode. When the current CU is intra predicted, dbbp_flag is not transmitted. The examples are shown in Table 7 and Table 8.

Table 7

coding_unit(x0, y0, log2CbSize , ctDepth) {	**Descriptor**
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if(depth_based_blk_part_flag[nuh_layer_id] && CuPredMode[x0][y0] != MODE_INTRA)	
dbbp_flag[x0][y0]	ae(v)
if((CuPredMode[x0][y0] != MODE_INTRA || log2CbSize = = MinCbLog2SizeY) && !predPartModeFlag)	
part_mode	ae(v)
if(sdcEnableFlag)	

Table 8

coding_unit(x0, y0, log2CbSize , ctDepth) {	**Descriptor**
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbS = (1 << log2CbSize)	

if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbS, nCbS)	
else {	
if(slice_type != I)	
pred_mode_flag	ae(v)
if(depth_based_blk_part_flag[nuh_layer_id] && CuPredMode[x0][y0] != MODE_INTRA)	
dbbp_flag[x0][y0]	ae(v)
if((CuPredMode[x0][y0] != MODE_INTRA || log2CbSize = = MinCbLog2SizeY) && !predPartModeFlag && ! dbbp_flag[x0][y0])	
part_mode	ae(v)
if(sdcEnableFlag)	
sdc_flag[x0][y0]	ae(v)

[0024] In the above examples, since the DBBP would implicitly divide current coding blocking into multiple partitions (e.g. 2 partitions or 4 partitions), we need to ensure the PU level syntax will be parsed correctly according to the number of partitions.

[0025] To solve this issue, some semantic constraint could be added to the "part_mode" as shown below. One example is given as below, for a CU with dbbp_flag equal to 1, its partition mode is set to a partition type with multiple (2 or 4) partitions (e.g. PART_2N×N, PART_N×2N, PART_N×N, or AMP partitions). In the below example, we modify the semantic description of "part_mode" in the SPEC of 3D-HEVC and the modified parts are highlighted as bold font.

I.7.4.9.4 Coding quadtree semantics (in 3D-HEVC Draft Text 3, JCT3V-G1001)

part_mode specifies partitioning mode of the current coding unit. The semantics of part_mode depend on CuPredMode[x0][y0]. The variables PartMode and IntraSplitFlag are derived from the value of part_mode and partPredIdc as defined in Table I-2. When dbbp_flag is equal to 1, PartMode is explicitly derived as PART_2N×N.

Table I-2 – Name association to prediction mode and partitioning type

CuPredMode[x0][y0]	**part_mode**	partPredIdc	**IntraSplitFlag**	**PartMode**
MODE_INTRA	0	-	0	PART_2Nx2N
	1	-	1	PART_NxN
MODE_INTER	0	-	0	PART_2Nx2N
	1	0	0	PART_2NxN
	1	1	0	PART_2NxN
	1	2	0	PART_Nx2N
	2	0	0	PART_Nx2N
	2	1	0	PART_2NxnU
	2	2	0	PART_nLx2N
	3	0	0	PART_NxN
	3	1	0	PART_2NxnD
	3	2	0	PART_nRx2N
	4	-	0	PART_2NxnU
	5	-	0	PART_2NxnD
	6	-	0	PART_nLx2N
	7	-	0	PART_nRx2N

[0026]Another way to solve this issue is to add condition to the parsing stage. In the below example 1a-b, when DBBP divides current block into two partitions, it may need to parse two PU level syntax. In example 1-b, instead of combining the dbbp flag with the partition PART_2N×N (else if(PartMode == PART_2N×N || dbbp_flag[x0][y0]))), the dbbp flag could also be combined with other partition mode with two partitions such as PART_N×2N or one of the AMP partition.

[0027]In the example 2, when DBBP divides current block into four partitions, it may need to parse four PU level syntax.

Example 1a

if(PartMode = = PART_2Nx2N)	
prediction_unit(x0, y0, nCbS, nCbS)	
else if(dbbp_flag[x0][y0]) {	
prediction_unit(x0, y0, nCbS, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS, nCbS / 2)	
}else if(PartMode = = PART_2NxN) {	
prediction_unit(x0, y0, nCbS, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS, nCbS / 2)	
} else if(PartMode = = PART_Nx2N) {	
prediction_unit(x0, y0, nCbS / 2, nCbS)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS)	

} else if(PartMode = = PART_2NxnU) {	
prediction_unit(x0, y0, nCbS, nCbS / 4)	
prediction_unit(x0, y0 + (nCbS / 4), nCbS, nCbS * 3 / 4)	
} else if(PartMode = = PART_2NxnD) {	
prediction_unit(x0, y0, nCbS, nCbS * 3 / 4)	
prediction_unit(x0, y0 + (nCbS * 3 / 4), nCbS, nCbS / 4)	
} else if(PartMode = = PART_nLx2N) {	
prediction_unit(x0, y0, nCbS / 4, nCbS)	
prediction_unit(x0 + (nCbS / 4), y0, nCbS * 3 / 4, nCbS)	
} else if(PartMode = = PART_nRx2N) {	
prediction_unit(x0, y0, nCbS * 3 / 4, nCbS)	
prediction_unit(x0 + (nCbS * 3 / 4), y0, nCbS / 4, nCbS)	
} else { /* PART_NxN */	
prediction_unit(x0, y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
}	

Example 1b

if(PartMode = = PART_2Nx2N)	
prediction_unit(x0, y0, nCbS, nCbS)	
}else if(PartMode = = PART_2NxN || **dbbp_flag**[x0][y0])) {	
prediction_unit(x0, y0, nCbS, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS, nCbS / 2)	
} else if(PartMode = = PART_Nx2N) {	
prediction_unit(x0, y0, nCbS / 2, nCbS)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS)	
} else if(PartMode = = PART_2NxnU) {	
prediction_unit(x0, y0, nCbS, nCbS / 4)	
prediction_unit(x0, y0 + (nCbS / 4), nCbS, nCbS * 3 / 4)	
} else if(PartMode = = PART_2NxnD) {	
prediction_unit(x0, y0, nCbS, nCbS * 3 / 4)	
prediction_unit(x0, y0 + (nCbS * 3 / 4), nCbS, nCbS / 4)	
} else if(PartMode = = PART_nLx2N) {	
prediction_unit(x0, y0, nCbS / 4, nCbS)	
prediction_unit(x0 + (nCbS / 4), y0, nCbS * 3 / 4, nCbS)	
} else if(PartMode = = PART_nRx2N) {	
prediction_unit(x0, y0, nCbS * 3 / 4, nCbS)	
prediction_unit(x0 + (nCbS * 3 / 4), y0, nCbS / 4, nCbS)	
} else { /* PART_NxN */	
prediction_unit(x0, y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
}	

Example 2

if(PartMode = = PART_2Nx2N)	
prediction_unit(x0, y0, nCbS, nCbS)	
else if(dbbp_flag[x0][y0])) {	
prediction_unit(x0, y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
}else if(PartMode = = PART_2NxN) {	
prediction_unit(x0, y0, nCbS, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS, nCbS / 2)	
} else if(PartMode = = PART_Nx2N) {	
prediction_unit(x0, y0, nCbS / 2, nCbS)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS)	
} else if(PartMode = = PART_2NxnU) {	
prediction_unit(x0, y0, nCbS, nCbS / 4)	
prediction_unit(x0, y0 + (nCbS / 4), nCbS, nCbS * 3 / 4)	
} else if(PartMode = = PART_2NxnD) {	
prediction_unit(x0, y0, nCbS, nCbS * 3 / 4)	
prediction_unit(x0, y0 + (nCbS * 3 / 4), nCbS, nCbS / 4)	
} else if(PartMode = = PART_nLx2N) {	
prediction_unit(x0, y0, nCbS / 4, nCbS)	
prediction_unit(x0 + (nCbS / 4), y0, nCbS * 3 / 4, nCbS)	
} else if(PartMode = = PART_nRx2N) {	
prediction_unit(x0, y0, nCbS * 3 / 4, nCbS)	
prediction_unit(x0 + (nCbS * 3 / 4), y0, nCbS / 4, nCbS)	
} else { /* PART_NxN */	
prediction_unit(x0, y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0, nCbS / 2, nCbS / 2)	
prediction_unit(x0, y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
prediction_unit(x0 + (nCbS / 2), y0 + (nCbS / 2), nCbS / 2, nCbS / 2)	
}	

[0028]The above description is presented to enable a person of ordinary skill in the art to practice the present invention as provided in the context of a particular application and its requirement. Various modifications to the described embodiments will be apparent to those with skill in the art, and the general principles defined herein may be applied to other embodiments. Therefore, the present invention is not intended to be limited to the particular embodiments shown and described, but is to be accorded the widest scope consistent with the principles and novel features herein disclosed. In the above detailed description, various specific details are illustrated in order to provide a thorough understanding of the present invention. Nevertheless, it will be understood by those skilled in the art that the present invention may be practiced.

[0029]Embodiment of the present invention as described above may be implemented in various hardware, software codes, or a combination of both. For example, an embodiment of the present invention can be a circuit integrated into a video compression chip or program code integrated into video compression software to perform the processing described herein. An embodiment of the present invention may also be program code to be executed on a Digital Signal Processor (DSP) to perform the processing described herein. The invention may also involve a number of functions to be performed by a computer processor, a digital signal processor, a microprocessor, or field programmable gate array (FPGA). These processors can be configured to perform particular tasks according to the invention, by executing machine-readable software code or firmware code that defines the particular methods embodied by the invention. The software code or firmware code may be developed in different programming languages and different formats or styles. The software code may also be compiled for different target platforms. However, different code formats, styles and languages of software codes and other means of configuring code to perform the tasks in accordance with the invention will not depart from the spirit and scope of the invention.

[0030]The invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described examples are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

1. Method to decide the enabling or disabling of DBBP mode comprising:

receiving input data associated with texture block of a current texture picture in a current dependent view;

receiving a depth block associated with the current texture block;

signaling a flag to indicate the enabling or disabling of the DBBP mode.

2. The method as claimed in claim 1, wherein the DBBP flag is signaled only when the partition mode is 2N×2N.

3. The method as claimed in claim 1, wherein the DBBP flag is signaled only when the partition mode is AMP xxx. (xxx could be one of the following AMP partitioning modes: PART_2N×nU, PART_2N×nD, PART_nL×2N and PART_nR×2N)

4. The method as claimed in claim 1, wherein the DBBP flag is always signaled and doesn't depend on the partition mode.

5. The method as claimed in claim 1, wherein the DBBP flag is transmitted before the partition flag, when the DBBP mode is enabled, the transmission of the partition flag is skipped.

6. The method as claimed in claim 1, wherein the DBBP flag and the partition flag are both transmitted, when the DBBP mode is enabled, the partition flag is actually used to represent the partition for motion information storage in DBBP and is not used to represent the partition of motion compensation.

7. The method as claimed in claim 1, wherein the DBBP flag is always signaled when the current CU is one inter-predicted CU and doesn't depend on the partition mode.

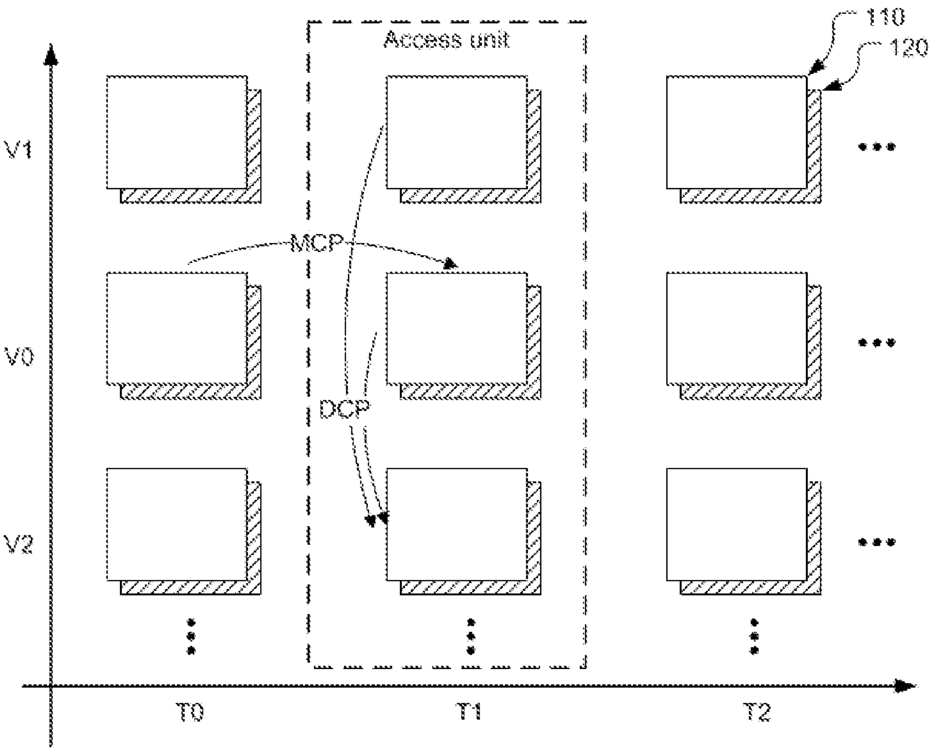

Fig. 1

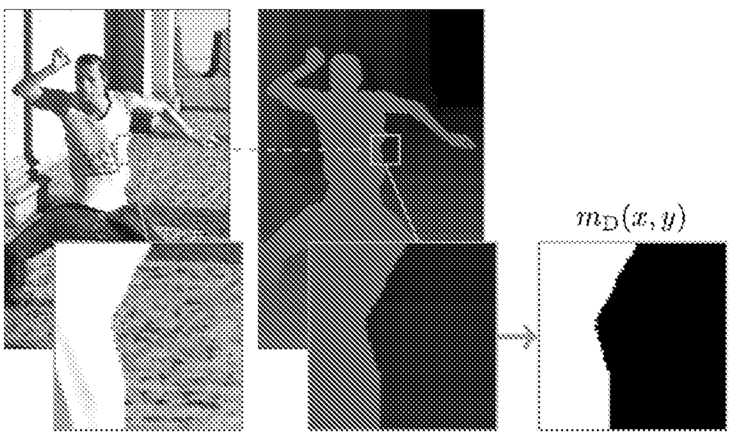

Fig. 2

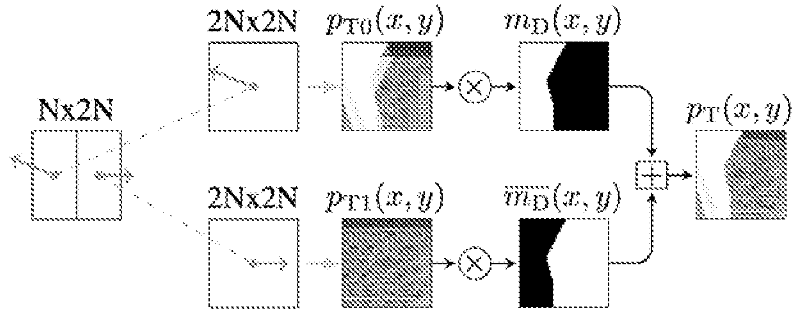

Fig. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2014/073547

A. CLASSIFICATION OF SUBJECT MATTER

H04N 7/24(2011.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI,CNPAT,WPI,EPODOC,GOOGLE:depth, signal, partitioning, three dimension, dbbp, flag, block, three, 3d, mode

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2014029086 A1 (MEDIATEK SINGAPORE PTE. LTD.) 27 February 2014 (2014-02-27) claims 1-9	1-7
A	US 2011222602 A1 (LG ELECTRONICS INC.) 15 September 2011 (2011-09-15) the whole document	1-7
A	WO 2013113134 A1 (NOKIA CORPORATION) 08 August 2013 (2013-08-08) the whole document	1-7
A	WO 2013138621 A1 (QUALCOMM INC.) 19 September 2013 (2013-09-19) the whole document	1-7

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier application or patent but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed
- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
- "&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 November 2014	**22 December 2014**

Name and mailing address of the ISA/CN	Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA(ISA/CN) **6,Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088 China**	**ZHANG,Tao**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)62414427**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2014/073547

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
WO	2014029086	A1	27 February 2014		Non	e	
US	2011222602	A1	15 September 2011	EP	2348732	A2	27 July 2011
				KR	20110093792	A	18 August 2011
				WO	2010053332	A2	14 May 2010
WO	2013113134	A1	08 August 2013		Non	e	
WO	2013138621	A1	19 September 2013	US	2013265388	A1	10 October 2013
				TW	201340724	A	01 October 2013
				US	2013242046	A1	19 September 2013
				WO	2013138593	A1	19 September 2013