



(51) International Patent Classification:  
*H04N 19/52* (2014.01)

(21) International Application Number:

PCT/CN2019/130712

(22) International Filing Date:

31 December 2019 (31.12.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2018/126066

31 December 2018 (31.12.2018) CN

PCT/CN2019/070636

07 January 2019 (07.01.2019) CN

PCT/CN2019/071159

10 January 2019 (10.01.2019) CN

(71) Applicants: **BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD.** [CN/CN]; Room B-0035, 2/F, No. 3 Building, No. 30, Shixing Road, Shijingshan Dis-

trict, Beijing 100041 (CN). **BYTEDANCE INC.** [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) Inventors: **ZHANG, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **LIU, Hongbin**; Jinritoutiao Post Office, China Satellite Communications Tower, No. 63, Zhichun Road, Haidian District, Beijing 100080 (CN). **XU, Jizheng**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **WANG, Yue**; Jinritoutiao Post Office, China Satellite Communications Tower, No. 63, Zhichun Road, Haidian District, Beijing 100080 (CN).

(74) Agent: **LIU, SHEN & ASSOCIATES**; 10th Floor, Building 1, 10 Caihefang Road, Haidian District, Beijing 100080 (CN).

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM,

(54) Title: PARSING METHOD OF DISTANCE INDEX IN MERGE WITH MVD

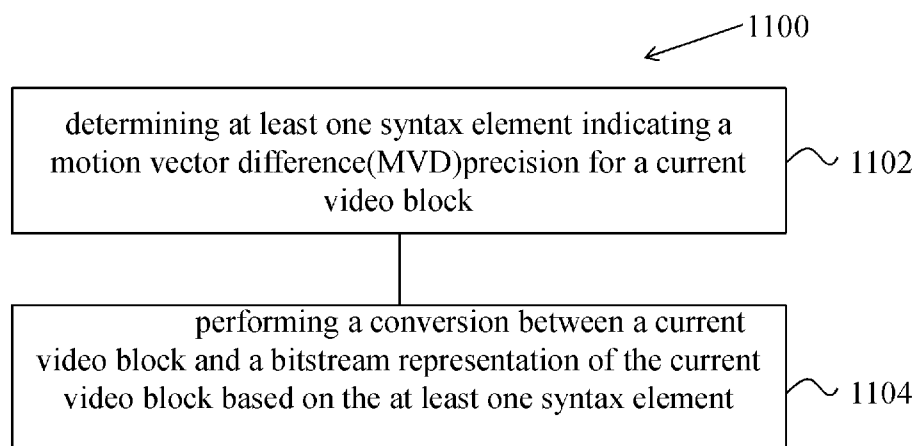


FIG. 11

(57) Abstract: Disclosed are systems, methods and devices for video processing. An exemplary method for video processing includes: determining at least one syntax element indicating a motion vector difference (MVD) precision for a current video block, the MVD precision indicating a distance between a motion vector of the current video block and a base candidate selected from a merge candidate list; performing a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element; MVD precisions allowed for the current video block are classified as entries into a plurality of subsets each is assigned with a subset index, and the at least one syntax element includes a first syntax element indicating the subset index and a second syntax element indicating a distance index (DI) which indicates an entry in a subset with the subset index.



AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, 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, ST, 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).

**Declarations under Rule 4.17:**

— *of inventorship (Rule 4.17(iv))*

**Published:**

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

## **PARSING METHOD OF DISTANCE INDEX IN MERGE WITH MVD**

### **TECHNICAL FIELD**

**[001]** Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefits of International Patent Application No. PCT/CN2018/126066, filed on December 31, 2018, International Patent Application No. PCT/CN2019/070636, filed on January 7, 2019, and International Patent Application No. PCT/CN2019/071159, filed on January 10, 2019. The entire disclosures thereof are incorporated by reference as part of the disclosure of this application.

**[002]** The present document relates to video and image coding and decoding.

### **BACKGROUND**

**[003]** Digital video accounts for the largest bandwidth use on the internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, it is expected that the bandwidth demand for digital video usage will continue to grow.

### **SUMMARY**

**[004]** The present document discloses video coding tools that, in one example aspect, improve the signaling of motion vectors for video and image coding.

**[005]** In one aspect, there is disclosed a method for video processing. The method comprises: determining at least one syntax element indicating a distance index(DI) for a current video block, wherein the at least one syntax element is not binarized with a truncated unary code; and performing a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element.

**[006]** In one aspect, there is disclosed a method for video processing. The method comprises: determining at least one syntax element indicating a motion vector difference(MVD)precision for a current video block, wherein the MVD precision indicates a distance between a motion vector of the current video block and a base candidate selected from a merge candidate list; performing a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element; wherein MVD precisions allowed for the current video block are classified as entries into a plurality of subsets each is assigned

with a subset index, and the at least one syntax element comprises a first syntax element indicating the subset index and a second syntax element indicating a distance index(DI) which indicates an entry in a subset with the subset index.

**[007]** In an aspect, there is disclosed an apparatus in a video system, the apparatus comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of examples described as above.

**[008]** In an aspect, there is disclosed a computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of examples described as above.

**[009]** In yet another example aspect, the above-described method may be implemented by a video encoder apparatus or a video decoder apparatus that comprises a processor.

**[0010]** In yet another example aspect, these methods may be embodied in the form of processor-executable instructions and stored on a computer-readable program medium.

**[0011]** These, and other, aspects are further described in the present document.

## **BRIEF DESCRIPTION OF DRAWINGS**

**[0012]** FIG. 1 shows an example of simplified affine motion model.

**[0013]** FIG. 2 shows an example of affine motion vector field (MVF) per sub-block.

**[0014]** FIG. 3A-3B show 4 and 6 parameter affine models, respectively.

**[0015]** FIG. 4 shows an example of motion vector predictor (MVP) for AF\_INTER.

**[0016]** FIG. 5A-5B show examples of candidates for AF\_MERGE.

**[0017]** FIG. 6 shows an example of candidate positions for affine merge mode.

**[0018]** FIG. 7 shows an example of distance index and distance offset mapping.

**[0019]** FIG. 8 shows an example of ultimate motion vector expression (UMVE) search process.

**[0020]** FIG. 9 shows an example of UMVE search point.

**[0021]** FIG. 10 is a flowchart for an example method for video processing.

**[0022]** FIG. 11 is a flowchart for another example method for video processing.

**[0023]** FIG. 12 shows an example of a hardware platform for implementing a technique described in the present document.

## DETAILED DESCRIPTION

**[0024]** The present document provides various techniques that can be used by a decoder of video bitstreams to improve the quality of decompressed or decoded digital video. Furthermore, a video encoder may also implement these techniques during the process of encoding in order to reconstruct decoded frames used for further encoding.

**[0025]** Section headings are used in the present document for ease of understanding and do not limit the embodiments and techniques to the corresponding sections. As such, embodiments from one section can be combined with embodiments from other sections.

### **[0026] 1. Summary**

**[0027]** This patent document is related to video coding technologies. Specifically, it is related to motion compensation in video coding. It may be applied to the existing video coding standard like HEVC, or the standard (Versatile Video Coding) to be finalized. It may be also applicable to future video coding standards or video codec.

### **[0028] 2. Introductory comments**

**[0029]** Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM). In April 2018, the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC.

#### **[0030] 2.1 Affine motion compensation prediction**

**[0031]** In HEVC, only translation motion model is applied for motion compensation prediction (MCP). While in the real world, there are many kinds of motion, e.g. zoom in/out, rotation, perspective motions and the other irregular motions. In the JEM, a simplified affine transform motion compensation prediction is applied. As shown FIG. 1, the affine motion field of the block is described by two control point motion vectors.

**[0032]** The motion vector field (MVF) of a block is described by the following equation:

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w}x - \frac{(v_{1y} - v_{0y})}{w}y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w}x + \frac{(v_{1x} - v_{0x})}{w}y + v_{0y} \end{cases} \quad (1)$$

**[0033]** Where  $(v_{0x}, v_{0y})$  is motion vector of the top-left corner control point, and  $(v_{1x}, v_{1y})$  is motion vector of the top-right corner control point.

**[0034]** In order to further simplify the motion compensation prediction, sub-block based affine transform prediction is applied. The sub-block size  $M \times N$  is derived as in Equation 2, where  $MvPre$  is the motion vector fraction accuracy (1/16 in JEM),  $(v_{2x}, v_{2y})$  is motion vector of the bottom-left control point, calculated according to Equation 1.

$$\begin{cases} M = clip3(4, w, \frac{w \times MvPre}{\max(abs(v_{1x} - v_{0x}), abs(v_{1y} - v_{0y}))}) \\ N = clip3(4, h, \frac{h \times MvPre}{\max(abs(v_{2x} - v_{0x}), abs(v_{2y} - v_{0y}))}) \end{cases} \quad (2)$$

**[0035]** After derived by Equation 2, M and N should be adjusted downward if necessary to make it a divisor of w and h, respectively.

**[0036]** To derive motion vector of each  $M \times N$  sub-block, the motion vector of the center sample of each sub-block, as shown in FIG. 2, is calculated according to Equation 1, and rounded to 1/16 fraction accuracy.

**[0037]** After MCP, the high accuracy motion vector of each sub-block is rounded and saved as the same accuracy as the normal motion vector.

#### **[0038] 2.1.1 AF\_INTER mode**

**[0039]** In the JEM, there are two affine motion modes: AF\_INTER mode and AF\_MERGE mode. For CUs with both width and height larger than 8, AF\_INTER mode can be applied. An affine flag in CU level is signaled in the bitstream to indicate whether AF\_INTER mode is used. In this mode, a candidate list with motion vector pair  $\{(v_0, v_1) | v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}\}$  is constructed using the neighbour blocks. As shown in FIG. 4,  $v_0$  is selected from the motion vectors of the block A, B or C. The motion vector from the neighbour block is scaled according to the reference list and the relationship among the POC of the reference for the neighbour block, the POC of the reference for the current CU and the POC of the current CU. And the approach to select  $v_1$  from the neighbour block D and E is similar. If the number of candidate list is smaller than 2, the list is padded by the motion vector pair composed by duplicating each of the AMVP candidates. When the candidate list is larger than 2, the candidates are firstly sorted according to the consistency of the neighbouring motion vectors (similarity of the two motion

vectors in a pair candidate) and only the first two candidates are kept. An RD cost check is used to determine which motion vector pair candidate is selected as the control point motion vector prediction (CPMVP) of the current CU. And an index indicating the position of the CPMVP in the candidate list is signaled in the bitstream. After the CPMVP of the current affine CU is determined, affine motion estimation is applied and the control point motion vector (CPMV) is found. Then the difference of the CPMV and the CPMVP is signaled in the bitstream.

**[0040]** FIG. 3A shows an example of a 4-parameter affine model. FIG. 3B shows an example of a 6-parameter affine model.

**[0041]** In AF\_INTER mode, when 4/6 parameter affine mode is used, 2/3 control points are required, and therefore 2/3 MVD needs to be coded for these control points, as shown in FIG. 3A. In an example, it is proposed to derive the MV as follows, e.g.,  $mvd_1$  and  $mvd_2$  are predicted from  $mvd_0$ .

$$\begin{aligned} mv_0 &= \overline{mv}_0 + mvd_0 \\ mv_1 &= \overline{mv}_1 + mvd_1 + mvd_0 \\ mv_2 &= \overline{mv}_2 + mvd_2 + mvd_0 \end{aligned}$$

**[0042]** Wherein  $\overline{mv}_i$ ,  $mvd_i$  and  $mv_i$  are the predicted motion vector, motion vector difference and motion vector of the top-left pixel ( $i = 0$ ), top-right pixel ( $i = 1$ ) or left-bottom pixel ( $i = 2$ ) respectively, as shown in FIG. 3B. Please note that the addition of two motion vectors (e.g.,  $mvA(xA, yA)$  and  $mvB(xB, yB)$ ) is equal to summation of two components separately, that is,  $newMV = mvA + mvB$  and the two components of newMV is set to  $(xA + xB)$  and  $(yA + yB)$ , respectively.

#### **[0043] 2.1.2 Fast affine ME algorithm in AF\_INTER mode**

**[0044]** In affine mode, MV of 2 or 3 control points needs to be determined jointly. Directly searching the multiple MVs jointly is computationally complex. A fast affine ME algorithm is proposed and is adopted into VTM/BMS.

**[0045]** The fast affine ME algorithm is described for the 4-parameter affine model, and the idea can be extended to 6-parameter affine model.

$$\begin{cases} x' = ax + by + c \\ y' = -bx + ay + d \end{cases} \quad (3)$$

$$\begin{cases} mv_{(x,y)}^h = x' - x = (a - 1)x + by + c \\ mv_{(x,y)}^v = y' - y = -bx + (a - 1)y + d \end{cases} \quad (4)$$

**[0046]** Replace  $(a - 1)$  with  $a'$ , then the motion vector can be rewritten as:

$$\begin{cases} mv_{(x,y)}^h = x' - x = a'x + by + c \\ mv_{(x,y)}^v = y' - y = -bx + a'y + d \end{cases} \quad (5)$$

**[0047]** Suppose motion vectors of the two controls points (0, 0) and (0, w) are known, from Equation (5) we can derive affine parameters,

$$\begin{cases} c = mv_{(0,0)}^h \\ d = mv_{(0,0)}^v \end{cases} \quad (6)$$

**[0048]** The motion vectors can be rewritten in vector form as:

$$MV(p) = A(P) * MV_C^T \quad (7)$$

**[0049]** Wherein

$$A(P) = \begin{bmatrix} 1 & x & 0 & y \\ 0 & y & 1 & -x \end{bmatrix} \quad (8)$$

$$MV_C = [mv_{(0,0)}^h \ a \ mv_{(0,0)}^v \ b] \quad (9)$$

**[0050]** P = (x, y) is the pixel position.

**[0051]** At encoder, MVD of AF\_INTER are derived iteratively. Denote  $MV^i(P)$  as the MV derived in the *i*th iteration for position P and denote  $dMV_C^i$  as the delta updated for  $MV_C$  in the *i*th iteration. Then in the (*i*+1)th iteration,

$$\begin{aligned} MV^{i+1}(P) &= A(P) * ((MV_C^i)^T + (dMV_C^i)^T) \\ &= A(P) * (MV_C^i)^T + A(P) * (dMV_C^i)^T \\ &= MV^i(P) + A(P) * (dMV_C^i)^T \end{aligned} \quad (10)$$

**[0052]** Denote  $Pic_{ref}$  as the reference picture and denote  $Pic_{cur}$  as the current picture and denote  $Q = P + MV^i(P)$ . Suppose we use MSE as the matching criterion, then we need to minimize:

$$\begin{aligned} \min \sum_P \left( Pic_{cur}(P) - Pic_{ref}(P + MV^{i+1}(P)) \right)^2 \\ = \min \sum_P \left( Pic_{cur}(P) - Pic_{ref}(Q + A(P) * (dMV_C^i)^T) \right)^2 \end{aligned} \quad (11)$$

**[0053]** Suppose  $(dMV_C^i)^T$  is small enough, we can rewrite  $Pic_{ref}(Q + A(P) * (dMV_C^i)^T)$  approximately as follows with 1th order Taylor expansion.

$$Pic_{ref}(Q + A(P) * (dMV_C^i)^T) \approx Pic_{ref}(Q) + Pic_{ref}'(Q) * A(P) * (dMV_C^i)^T \quad (12)$$

**[0054]** Wherein  $Pic_{ref}'(Q) = \left[ \frac{dPic_{ref}(Q)}{dx} \ \frac{dPic_{ref}(Q)}{dy} \right]$ . Denote  $E^{i+1}(P) = Pic_{cur}(P) - Pic_{ref}(Q)$ ,

$$\begin{aligned} \min \sum_P \left( Pic_{cur}(P) - Pic_{ref}(Q) - Pic_{ref}'(Q) * A(P) * (dMV_C^i)^T \right)^2 \\ = \min \sum_P \left( E^{i+1}(P) - Pic_{ref}'(Q) * A(P) * (dMV_C^i)^T \right)^2 \end{aligned} \quad (13)$$

**[0055]** we can derive  $dMV_C^i$  by setting the derivative of the error function to zero. Then can then calculate delta MV of the control points (0, 0) and (0, w) according to  $A(P) * (dMV_C^i)^T$ ,

$$dMV_{(0,0)}^h = dMV_C^i[0] \quad (14)$$

$$dMV_{(0,w)}^h = dMV_C^i[1] * w + dMV_C^i[2] \quad (15)$$

$$dMV_{(0,0)}^v = dMV_C^i[2] \quad (16)$$

$$dMV_{(0,w)}^v = -dMV_C^i[3] * w + dMV_C^i[2] \quad (17)$$

**[0056]** Suppose such MVD derivation process is iterated by n times, then the final MVD is calculated as follows,

$$fdMV_{(0,0)}^h = \sum_{i=0}^{n-1} dMV_C^i[0] \quad (18)$$

$$fdMV_{(0,w)}^h = \sum_{i=0}^{n-1} dMV_C^i[1] * w + \sum_{i=0}^{n-1} dMV_C^i[2] \quad (19)$$

$$fdMV_{(0,0)}^v = \sum_{i=0}^{n-1} dMV_C^i[2] \quad (20)$$

$$fdMV_{(0,w)}^v = \sum_{i=0}^{n-1} -dMV_C^i[3] * w + \sum_{i=0}^{n-1} dMV_C^i[2] \quad (21)$$

**[0057]** As an example, e.g., predicting delta MV of control point (0, w), denoted by  $mvd_1$  from delta MV of control point (0, 0), denoted by  $mvd_0$ , now actually only  $(\sum_{i=0}^{n-1} dMV_C^i[1] * w, -\sum_{i=0}^{n-1} -dMV_C^i[3] * w)$  is encoded for  $mvd_1$ .

### **[0058] 2.1.3 AF\_MERGE mode**

**[0059]** When a CU is applied in AF\_MERGE mode, it gets the first block coded with affine mode from the valid neighbour reconstructed blocks. And the selection order for the candidate block is from left, above, above right, left bottom to above left as shown in FIG. 5A. If the neighbour left bottom block A is coded in affine mode as shown in FIG. 5B, the motion vectors  $v_2$ ,  $v_3$  and  $v_4$  of the top left corner, above right corner and left bottom corner of the CU which contains the block A are derived. And the motion vector  $v_0$  of the top left corner on the current CU is calculated according to  $v_2$ ,  $v_3$  and  $v_4$ . Secondly, the motion vector  $v_1$  of the above right of the current CU is calculated.

**[0060]** After the CPMV of the current CU  $v_0$  and  $v_1$  are derived, according to the simplified affine motion model Equation 1, the MVF of the current CU is generated. In order to identify whether the current CU is coded with AF\_MERGE mode, an affine flag is signaled in the bitstream when there is at least one neighbour block is coded in affine mode.

**[0061]** In an example, which was planned to be adopted into VTM 3.0, an affine merge candidate list is constructed with following steps:

**[0062]** 1) Insert inherited affine candidates

**[0063]** Inherited affine candidate means that the candidate is derived from the affine motion model of its valid neighbor affine coded block. In the common base, as shown in FIG. 6, the scan order for the candidate positions is: A1, B1, B0, A0 and B2.

**[0064]** After a candidate is derived, full pruning process is performed to check whether same candidate has been inserted into the list. If a same candidate exists, the derived candidate is discarded.

**[0065]** 2) Insert constructed affine candidates

**[0066]** If the number of candidates in affine merge candidate list is less than MaxNumAffineCand (set to 5 in this contribution), constructed affine candidates are inserted into the candidate list. Constructed affine candidate means the candidate is constructed by combining the neighbor motion information of each control point.

**[0067]** The motion information for the control points is derived firstly from the specified spatial neighbors and temporal neighbor shown in FIG. 5B. CP<sub>k</sub> (k=1, 2, 3, 4) represents the k-th control point. A<sub>0</sub>, A<sub>1</sub>, A<sub>2</sub>, B<sub>0</sub>, B<sub>1</sub>, B<sub>2</sub> and B<sub>3</sub> are spatial positions for predicting CP<sub>k</sub> (k=1, 2, 3); T is temporal position for predicting CP<sub>4</sub>.

**[0068]** The coordinates of CP<sub>1</sub>, CP<sub>2</sub>, CP<sub>3</sub> and CP<sub>4</sub> is (0, 0), (W, 0), (H, 0) and (W, H), respectively, where W and H are the width and height of current block.

**[0069]** FIG. 6 shows an example of candidates position for affine merge mode

**[0070]** The motion information of each control point is obtained according to the following priority order:

**[0071]** For CP<sub>1</sub>, the checking priority is B<sub>2</sub>->B<sub>3</sub>->A<sub>2</sub>. B<sub>2</sub> is used if it is available. Otherwise, if B<sub>2</sub> is available, B<sub>3</sub> is used. If both B<sub>2</sub> and B<sub>3</sub> are unavailable, A<sub>2</sub> is used. If all the three candidates are unavailable, the motion information of CP<sub>1</sub> cannot be obtained.

**[0072]** For CP<sub>2</sub>, the checking priority is B<sub>1</sub>->B<sub>0</sub>.

**[0073]** For CP<sub>3</sub>, the checking priority is A<sub>1</sub>->A<sub>0</sub>.

**[0074]** For CP<sub>4</sub>, T is used.

**[0075]** Secondly, the combinations of controls points are used to construct an affine merge candidate.

**[0076]** Motion information of three control points are needed to construct a 6-parameter affine candidate. The three control points can be selected from one of the following four combinations ({CP<sub>1</sub>, CP<sub>2</sub>, CP<sub>4</sub>}, {CP<sub>1</sub>, CP<sub>2</sub>, CP<sub>3</sub>}, {CP<sub>2</sub>, CP<sub>3</sub>, CP<sub>4</sub>}, {CP<sub>1</sub>, CP<sub>3</sub>, CP<sub>4</sub>}). Combinations {CP<sub>1</sub>, CP<sub>2</sub>, CP<sub>3</sub>}, {CP<sub>2</sub>, CP<sub>3</sub>, CP<sub>4</sub>}, {CP<sub>1</sub>, CP<sub>3</sub>, CP<sub>4</sub>} will be converted to a 6-parameter motion model represented by top-left, top-right and bottom-left control points.

**[0077]** Motion information of two control points are needed to construct a 4-parameter affine candidate. The two control points can be selected from one of the following six combinations

({CP1, CP4}, {CP2, CP3}, {CP1, CP2}, {CP2, CP4}, {CP1, CP3}, {CP3, CP4}). Combinations {CP1, CP4}, {CP2, CP3}, {CP2, CP4}, {CP1, CP3}, {CP3, CP4} will be converted to a 4-parameter motion model represented by top-left and top-right control points.

**[0078]** The combinations of constructed affine candidates are inserted into to candidate list as following order:

{CP1, CP2, CP3}, {CP1, CP2, CP4}, {CP1, CP3, CP4}, {CP2, CP3, CP4}, {CP1, CP2}, {CP1, CP3}, {CP2, CP3}, {CP1, CP4}, {CP2, CP4}, {CP3, CP4}

**[0079]** For reference list X (X being 0 or 1) of a combination, the reference index with highest usage ratio in the control points is selected as the reference index of list X, and motion vectors point to difference reference picture will be scaled.

**[0080]** After a candidate is derived, full pruning process is performed to check whether same candidate has been inserted into the list. If a same candidate exists, the derived candidate is discarded.

**[0081]** 3) Padding with zero motion vectors

**[0082]** If the number of candidates in affine merge candidate list is less than 5, zero motion vectors with zero reference indices are insert into the candidate list, until the list is full.

## **[0083] 2.2 Affine merge mode with prediction offsets**

**[0084]** In an example, UMVE is extended to affine merge mode, we will call this UMVE affine mode thereafter. The proposed method selects the first available affine merge candidate as a base predictor. Then it applies a motion vector offset to each control point's motion vector value from the base predictor. If there's no affine merge candidate available, this proposed method will not be used.

**[0085]** The selected base predictor's inter prediction direction, and the reference index of each direction is used without change.

**[0086]** In the current implementation, the current block's affine model is assumed to be a 4-parameter model, only 2 control points need to be derived. Thus, only the first 2 control points of the base predictor will be used as control point predictors.

**[0087]** For each control point, a zero\_MVD flag is used to indicate whether the control point of current block has the same MV value as the corresponding control point predictor. If zero\_MVD flag is true, there's no other signaling needed for the control point. Otherwise, a distance index and an offset direction index is signaled for the control point.

**[0088]** A distance offset table with size of 5 is used as shown in the table below. Distance index is signaled to indicate which distance offset to use. The mapping of distance index and distance offset values is shown in FIG. 7.

*Table - Distance offset table*

| Distance IDX    | 0       | 1     | 2     | 3     | 4     |
|-----------------|---------|-------|-------|-------|-------|
| Distance-offset | 1/2-pel | 1-pel | 2-pel | 4-pel | 8-pel |

**[0089]** The direction index can represent four directions as shown below, where only x or y direction may have an MV difference, but not in both directions.

| Offset Direction IDX | 00 | 01 | 10 | 11 |
|----------------------|----|----|----|----|
| x-dir-factor         | +1 | -1 | 0  | 0  |
| y-dir-factor         | 0  | 0  | +1 | -1 |

**[0090]** If the inter prediction is uni-directional, the signaled distance offset is applied on the offset direction for each control point predictor. Results will be the MV value of each control point.

**[0091]** For example, when base predictor is uni-directional, and the motion vector values of a control point is MVP ( $v_{px}$ ,  $v_{py}$ ). When distance offset and direction index are signaled, the motion vectors of current block's corresponding control points will be calculated as below.

$$MV(v_x, v_y) = MVP(v_{px}, v_{py}) + MV(x\text{-dir-factor} * \text{distance-offset}, y\text{-dir-factor} * \text{distance-offset});$$

**[0092]** If the inter prediction is bi-directional, the signaled distance offset is applied on the signaled offset direction for control point predictor's L0 motion vector; and the same distance offset with opposite direction is applied for control point predictor's L1 motion vector. Results will be the MV values of each control point, on each inter prediction direction.

**[0093]** For example, when base predictor is uni-directional, and the motion vector values of a control point on L0 is MVP<sub>L0</sub>( $v_{0px}$ ,  $v_{0py}$ ), and the motion vector of that control point on L1 is MVP<sub>L1</sub>( $v_{1px}$ ,  $v_{1py}$ ). When distance offset and direction index are signaled, the motion vectors of current block's corresponding control points will be calculated as below.

$$MV_{L0}(v_{0x}, v_{0y}) = MVP_{L0}(v_{0px}, v_{0py}) + MV(x\text{-dir-factor} * \text{distance-offset}, y\text{-dir-factor} * \text{distance-offset});$$

$$MV_{L1}(v_{1x}, v_{1y}) = MVP_{L1}(v_{0px}, v_{0py}) + MV(-x\text{-dir-factor} * \text{distance-offset}, -y\text{-dir-factor} * \text{distance-offset});$$

**[0094]** **2.3 Ultimate motion vector expression**

[0095] In an example, ultimate motion vector expression (UMVE) is presented. UMVE is used for either skip or merge modes with a proposed motion vector expression method.

[0096] UMVE re-uses merge candidate as same as those included in the regular merge candidate list in VVC. Among the merge candidates, a base candidate can be selected, and is further expanded by the proposed motion vector expression method.

[0097] UMVE provides a new motion vector difference (MVD) representation method, in which a starting point, a motion magnitude and a motion direction are used to represent a MVD.

[0098] FIG. 8 shows an example of UMVE Search Process.

[0099] FIG. 9 shows examples of UMVE Search Points.

[00100] This proposed technique uses a merge candidate list as it is. But only candidates which are default merge type (MRG\_TYPE\_DEFAULT\_N) are considered for UMVE's expansion.

[00101] Base candidate index defines the starting point. Base candidate index indicates the best candidate among candidates in the list as follows.

*Table 1. Base candidate IDX*

| Base candidate IDX  | 0                   | 1                   | 2                   | 3                   |
|---------------------|---------------------|---------------------|---------------------|---------------------|
| N <sup>th</sup> MVP | 1 <sup>st</sup> MVP | 2 <sup>nd</sup> MVP | 3 <sup>rd</sup> MVP | 4 <sup>th</sup> MVP |

[00102] If the number of base candidate is equal to 1, Base candidate IDX is not signaled.

[00103] Distance index is motion magnitude information. Distance index indicates the pre-defined distance from the starting point information. Pre-defined distance is as follows:

*Table 2a. Distance IDX*

| Distance IDX   | 0       | 1       | 2     | 3     | 4     | 5     | 6      | 7      |
|----------------|---------|---------|-------|-------|-------|-------|--------|--------|
| Pixel distance | 1/4-pel | 1/2-pel | 1-pel | 2-pel | 4-pel | 8-pel | 16-pel | 32-pel |

[00104] The distance IDX is binarized in bins with the truncated unary code in the entropy coding procedure as:

*Table 2b: Distance IDX Binarization*

| Distance IDX |  | 0 | 1  | 2   | 3    | 4     | 5      | 6       | 7       |
|--------------|--|---|----|-----|------|-------|--------|---------|---------|
| Bins         |  | 0 | 10 | 110 | 1110 | 11110 | 111110 | 1111110 | 1111111 |

[00105] In arithmetic coding, the first bin is coded with a probability context, and the following bins are coded with the equal-probability model, a.k.a. by-pass coding.

[00106] Direction index represents the direction of the MVD relative to the starting point. The direction index can represent of the four directions as shown below.

Table 3. Direction IDX

| Direction IDX | 00  | 01  | 10  | 11  |
|---------------|-----|-----|-----|-----|
| x-axis        | +   | –   | N/A | N/A |
| y-axis        | N/A | N/A | +   | –   |

**[00107]** UMVE flag is signaled right after sending a skip flag or merge flag. If skip or merge flag is true, UMVE flag is parsed. If UMVE flag is equal to 1, UMVE syntaxes are parsed. But, if not 1, AFFINE flag is parsed. If AFFINE flag is equal to 1, that is AFFINE mode, But, if not 1, skip/merge index is parsed for VTM's skip/merge mode.

**[00108]** Additional line buffer due to UMVE candidates is not needed. Because a skip/merge candidate of software is directly used as a base candidate. Using input UMVE index, the supplement of MV is decided right before motion compensation. There is no need to hold long line buffer for this.

**[00109]** In current common test condition, either the first or the second merge candidate in the merge candidate list could be selected as the base candidate.

**[00110]** UMVE is known as Merge with MVD (MMVD).

#### **[00111] 2.4 Generalized Bi-prediction**

**[00112]** In conventional bi-prediction, the predictors from L0 and L1 are averaged to generate the final predictor using the equal weight 0.5. The predictor generation formula is shown as in Equ. (3)

$$P_{\text{TraditionalBiPred}} = (P_{L0} + P_{L1} + \text{RoundingOffset}) \gg \text{shiftNum}, \quad (1)$$

**[00113]** In Equ. (3),  $P_{\text{TraditionalBiPred}}$  is the final predictor for the conventional bi-prediction,  $P_{L0}$  and  $P_{L1}$  are predictors from L0 and L1, respectively, and  $\text{RoundingOffset}$  and  $\text{shiftNum}$  are used to normalize the final predictor.

**[00114]** Generalized Bi-prediction (GBI) is proposed to allow applying different weights to predictors from L0 and L1. The predictor generation is shown in Equ. (4).

$$P_{\text{GBi}} = ((1-w_1) * P_{L0} + w_1 * P_{L1} + \text{RoundingOffset}_{\text{GBi}}) \gg \text{shiftNum}_{\text{GBi}}, \quad (2)$$

**[00115]** In Equ. (4),  $P_{\text{GBi}}$  is the final predictor of GBI.  $(1-w_1)$  and  $w_1$  are the selected GBI weights applied to the predictors of L0 and L1, respectively.  $\text{RoundingOffset}_{\text{GBi}}$  and  $\text{shiftNum}_{\text{GBi}}$  are used to normalize the final predictor in GBI.

**[00116]** The supported weights of  $w_1$  is  $\{-1/4, 3/8, 1/2, 5/8, 5/4\}$ . One equal-weight set and four unequal-weight sets are supported. For the equal-weight case, the process to generate the final predictor is exactly the same as that in the conventional bi-prediction mode. For the true bi-

prediction cases in random access (RA) condition, the number of candidate weight sets is reduced to three.

**[00117]** For advanced motion vector prediction (AMVP) mode, the weight selection in GBI is explicitly signaled at CU-level if this CU is coded by bi-prediction. For merge mode, the weight selection is inherited from the merge candidate. In this proposal, GBI supports DMVR to generate the weighted average of template as well as the final predictor for BMS-1.0.

**[00118] 2.5 Adaptive motion vector difference resolution**

**[00119]** In HEVC, motion vector differences (MVDs) (between the motion vector and predicted motion vector of a PU) are signaled in units of quarter luma samples when `use_integer_mv_flag` is equal to 0 in the slice header. In VTM-3.0, a locally adaptive motion vector resolution (LAMVR) is introduced. In the JEM, MVD can be coded in units of quarter luma samples, integer luma samples or four luma samples. The MVD resolution is controlled at the coding unit (CU) level, and MVD resolution flags are conditionally signaled for each CU that has at least one non-zero MVD components.

**[00120]** For a CU that has at least one non-zero MVD components, a first flag is signaled to indicate whether quarter luma sample MV precision is used in the CU. When the first flag (equal to 1) indicates that quarter luma sample MV precision is not used, another flag is signaled to indicate whether integer luma sample MV precision or four luma sample MV precision is used.

**[00121]** When the first MVD resolution flag of a CU is zero, or not coded for a CU (meaning all MVDs in the CU are zero), the quarter luma sample MV resolution is used for the CU. When a CU uses integer-luma sample MV precision or four-luma-sample MV precision, the MVPs in the AMVP candidate list for the CU are rounded to the corresponding precision.

**[00122]** In arithmetic coding, the first MVD resolution flag is coded with one of three probability contexts: C0, C1 or C2; while the second MVD resolution flag is coded with a fourth probability context: C3. The probability context C<sub>x</sub> for the first MVD resolution flag is derived as (L represents the left neighbouring block and A represents the above neighbouring block):

**[00123]** If L is available, inter-coded, and its first MVD resolution flag is not equal to zero, x<sub>L</sub> is set equal to 1; otherwise, x<sub>L</sub> is set equal to 0.

**[00124]** If A is available, inter-coded, and its first MVD resolution flag is not equal to zero, x<sub>A</sub> is set equal to 1; otherwise, x<sub>A</sub> is set equal to 0.

**[00125]** x is set equal to x<sub>L</sub>+x<sub>A</sub>.

**[00126]** In the encoder, CU-level RD checks are used to determine which MVD resolution is to be used for a CU. That is, the CU-level RD check is performed three times for each MVD resolution. To accelerate encoder speed, the following encoding schemes are applied in the JEM.

**[00127]**     ○ During RD check of a CU with normal quarter luma sample MVD resolution, the motion information of the current CU (integer luma sample accuracy) is stored. The stored motion information (after rounding) is used as the starting point for further small range motion vector refinement during the RD check for the same CU with integer luma sample and 4 luma sample MVD resolution so that the time-consuming motion estimation process is not duplicated three times.

**[00128]**     ○ RD check of a CU with 4 luma sample MVD resolution is conditionally invoked. For a CU, when RD cost integer luma sample MVD resolution is much larger than that of quarter luma sample MVD resolution, the RD check of 4 luma sample MVD resolution for the CU is skipped.

**[00129]** In VTM-3.0, LAMVR is also known as Integer Motion Vector (IMV).

## **[00130] 2.6 Current picture referencing**

### **[00131] Decoder aspect:**

**[00132]** In this approach, the current (partially) decoded picture is considered as a reference picture. This current picture is put in the last position of reference picture list 0. Therefore, for a slice using the current picture as the only reference picture, its slice type is considered as a P slice. The bitstream syntax in this approach follows the same syntax structure for inter coding while the decoding process is unified with inter coding. The only outstanding difference is that the block vector (which is the motion vector pointing to the current picture) always uses integer-pel resolution.

**[00133]** Changes from block level CPR\_flag approach are:

- In encoder search for this mode, both block width and height are smaller than or equal to 16.
- Enable chroma interpolation when luma block vector is an odd integer number.
- Enable adaptive motion vector resolution (AMVR) for CPR mode when the SPS flag is on. In this case, when AMVR is used, a block vector can switch between 1-pel integer and 4-pel integer resolutions at block level.

### **[00134] Encoder aspect:**

**[00135]** The encoder performs RD check for blocks with either width or height no larger than 16. For non-merge mode, the block vector search is performed using hash-based search first. If there is no valid candidate found from hash search, block matching based local search will be performed.

**[00136]** In the hash-based search, hash key matching (32-bit CRC) between the current block and a reference block is extended to all allowed block sizes. The hash key calculation for every position in current picture is based on 4x4 blocks. For the current block of a larger size, a hash key matching to a reference block happens when all its 4x4 blocks match the hash keys in the corresponding reference locations. If multiple reference blocks are found to match the current block with the same hash key, the block vector costs of each candidates are calculated and the one with minimum cost is selected.

**[00137]** In block matching search, the search range is set to be 64 pixels to the left and on top of current block, and the search range is restricted to be within the current CTU.

**[00138] 2.7 Merge list design in one example**

**[00139]** There are three different merge list construction processes supported in VVC:

**[00140]** 1) Sub-block merge candidate list: it includes ATMVP and affine merge candidates. One merge list construction process is shared for both affine modes and ATMVP mode. Here, the ATMVP and affine merge candidates may be added in order. Sub-block merge list size is signaled in slice header, and maximum value is 5.

**[00141]** 2) Uni-Prediction TPM merge list: For triangular prediction mode, one merge list construction process for the two partitions is shared even two partitions could select their own merge candidate index. When constructing this merge list, the spatial neighbouring blocks and two temporal blocks of the block are checked. The motion information derived from spatial neighbours and temporal blocks are called regular motion candidates in our IDF. These regular motion candidates are further utilized to derive multiple TPM candidates. Please note the transform is performed in the whole block level, even two partitions may use different motion vectors for generating their own prediction blocks.

**[00142]** In some embodiments, uni-Prediction TPM merge list size is fixed to be 5.

**[00143]** 3) Regular merge list: For remaining coding blocks, one merge list construction process is shared. Here, the spatial/temporal/HMVP, pairwise combined bi-prediction merge candidates and zero motion candidates may be inserted in order. Regular merge list size is signaled in slice header, and maximum value is 6.

**[00144]** Sub-block merge candidate list

**[00145]** It is suggested that all the sub-block related motion candidates are put in a separate merge list in addition to the regular merge list for non-sub block merge candidates.

**[00146]** The sub-block related motion candidates are put in a separate merge list is named as 'sub-block merge candidate list'.

**[00147]** In one example, the sub-block merge candidate list includes affine merge candidates, and ATMVP candidate, and/or sub-block based STMVP candidate.

**[00148]** In this contribution, the ATMVP merge candidate in the normal merge list is moved to the first position of the affine merge list. Such that all the merge candidates in the new list (i.e., sub-block based merge candidate list) are based on sub-block coding tools.

**[00149]** An affine merge candidate list is constructed with following steps:

**[00150]** 1) Insert inherited affine candidates

**[00151]** Inherited affine candidate means that the candidate is derived from the affine motion model of its valid neighbor affine coded block. The maximum two inherited affine candidates are derived from affine motion model of the neighboring blocks and inserted into the candidate list. For the left predictor, the scan order is {A0, A1}; for the above predictor, the scan order is {B0, B1, B2}.

**[00152]** 2) Insert constructed affine candidates

**[00153]** If the number of candidates in affine merge candidate list is less than MaxNumAffineCand (set to 5), constructed affine candidates are inserted into the candidate list. Constructed affine candidate means the candidate is constructed by combining the neighbor motion information of each control point.

**[00154]** The motion information for the control points is derived firstly from the specified spatial neighbors and temporal neighbor shown in Fig. 7. CP<sub>k</sub> (k=1, 2, 3, 4) represents the k-th control point. A0, A1, A2, B0, B1, B2 and B3 are spatial positions for predicting CP<sub>k</sub> (k=1, 2, 3); T is temporal position for predicting CP4.

**[00155]** The coordinates of CP1, CP2, CP3 and CP4 is (0, 0), (W, 0), (H, 0) and (W, H), respectively, where W and H are the width and height of current block.

**[00156]** The motion information of each control point is obtained according to the following priority order:

**[00157]** For CP1, the checking priority is  $B2 \rightarrow B3 \rightarrow A2$ . B2 is used if it is available. Otherwise, if B2 is available, B3 is used. If both B2 and B3 are unavailable, A2 is used. If all the three candidates are unavailable, the motion information of CP1 cannot be obtained.

**[00158]** For CP2, the checking priority is  $B1 \rightarrow B0$ .

**[00159]** For CP3, the checking priority is  $A1 \rightarrow A0$ .

**[00160]** For CP4, T is used.

**[00161]** Secondly, the combinations of controls points are used to construct an affine merge candidate.

**[00162]** Motion information of three control points are needed to construct a 6-parameter affine candidate. The three control points can be selected from one of the following four combinations  $\{\{CP1, CP2, CP4\}, \{CP1, CP2, CP3\}, \{CP2, CP3, CP4\}, \{CP1, CP3, CP4\}\}$ . Combinations  $\{CP1, CP2, CP3\}, \{CP2, CP3, CP4\}, \{CP1, CP3, CP4\}$  will be converted to a 6-parameter motion model represented by top-left, top-right and bottom-left control points.

**[00163]** Motion information of two control points are needed to construct a 4-parameter affine candidate. The two control points can be selected from one of the two combinations  $\{\{CP1, CP2\}, \{CP1, CP3\}\}$ . The two combinations will be converted to a 4-parameter motion model represented by top-left and top-right control points.

**[00164]** The combinations of constructed affine candidates are inserted into to candidate list as following order:

**[00165]**  $\{CP1, CP2, CP3\}, \{CP1, CP2, CP4\}, \{CP1, CP3, CP4\}, \{CP2, CP3, CP4\}, \{CP1, CP2\}, \{CP1, CP3\}$

**[00166]** The available combination of motion information of CPs is only added to the affine merge list when the CPs have the same reference index.

**[00167]** 3) Padding with zero motion vectors

**[00168]** If the number of candidates in affine merge candidate list is less than 5, zero motion vectors with zero reference indices are insert into the candidate list, until the list is full.

**[00169]** **2.8 MMVD with affine merge candidate in an example**

**[00170]** For example the MMVD idea is applied on affine merge candidates (named as Affine merge with prediction offset). It is an extension of A MVD (or named as “distance”, or “offset”) is signaled after an affine merge candidate (known as the is signaled. The all CPMVs are added with the MVD to get the new CPMVs. The distance table is specified as

|              |   |   |   |   |   |
|--------------|---|---|---|---|---|
| Distance IDX | 0 | 1 | 2 | 3 | 4 |
|--------------|---|---|---|---|---|

|                 |         |       |       |       |       |
|-----------------|---------|-------|-------|-------|-------|
| Distance-offset | 1/2-pel | 1-pel | 2-pel | 4-pel | 8-pel |
|-----------------|---------|-------|-------|-------|-------|

**[00171]** In some embodiments, a POC distance based offset mirroring method is used for Bi-prediction. When the base candidate is bi-predicted, the offset applied to L0 is as signaled, and the offset on L1 depends on the temporal position of the reference pictures on list 0 and list 1.

**[00172]** If both reference pictures are on the same temporal side of the current picture, the same distance offset and same offset directions are applied for CPMVs of both L0 and L1.

**[00173]** When the two reference pictures are on different sides of the current picture, the CPMVs of L1 will have the distance offset applied on the opposite offset direction.

**[00174] 3. Examples of problems solved by the disclosed embodiments**

**[00175]** There are some potential problems in the design of MMVD:

- The coding/parsing procedure for UMVE information may be not efficient since it uses truncated unary binarization method for coding distance (MVD precision) information and fixed length with bypass coding for direction index. That is based on the assumption that the 1/4-pel precision takes the highest percentage. However, it is not true for all kinds of sequences.
- The design of the possible distances may be not efficient.

**[00176] 4. Examples of techniques implemented by various embodiments**

**[00177]** The list below should be considered as examples to explain general concepts. These inventions should not be interpreted in a narrow way. Furthermore, these techniques can be combined in any manner.

Parsing on distance index (e.g., MVD precision index)

1. It is proposed that the distance index (DI) used in UMVE is not binarized with the truncated unary code.
  - a. In one example, DI may be binarized with fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.
2. The distance index may be signaled with more than one syntax elements.
3. It is proposed to classify the set of allowed distances to multiple sub-sets, e.g., K sub-sets (K is larger than 1). A sub-set index (1<sup>st</sup> syntax) is firstly signaled, followed by distance index (2<sup>nd</sup> syntax) within the sub-set.
  - a. For example, **mmvd\_distance\_subset\_idx** is first signaled, followed by **mmvd\_distance\_idx\_in\_subset**.

- i. In one example, the **mmvd\_distance\_idx\_in\_subset** can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.
  - (i) Specifically, **mmvd\_distance\_idx\_in\_subset** can be binarized as a flag if there are only two possible distances in the sub-set.
  - (ii) Specifically, **mmvd\_distance\_idx\_in\_subset** is not signaled if there is only one possible distance in the sub-set.
  - (iii) Specifically, the maximum value is set to be the number of possible distance in the sub-set minus 1 if **mmvd\_distance\_idx\_in\_subset** is binarized as a truncated code.
- b. In one example, there are two sub-sets (e.g., K=2).
  - i. In one example, one sub-set includes all fractional MVD precisions (e.g.,  $\frac{1}{4}$ -pel,  $\frac{1}{2}$ -pel). The other sub-set includes all integer MVD precisions (e.g., 1-pel, 2-pel, 4-pel, 8-pel, 16-pel, 32-pel).
  - ii. In one example, one sub-set may only have one distance (e.g.,  $\frac{1}{2}$ -pel), and the other sub-set has all the remaining distances.
- c. In one example, there are three sub-sets (e.g., K=3).
  - i. In one example, a first sub-set includes fractional MVD precisions (e.g.,  $\frac{1}{4}$ -pel,  $\frac{1}{2}$ -pel); a second sub-set includes integer MVD precisions less than 4-pel (e.g. 1-pel, 2-pel) ; and a third sub-set includes all other MVD precisions (e.g. 4-pel, 8-pel, 16-pel, 32-pel).
- d. In one example, there are K sub-sets and the number of K is set equal to that allowed MVD precisions in LAMVR.
  - i. Alternatively, furthermore, the signaling of sub-set index may reuse what is done for LAMVR (e.g., reusing the way to derive context offset index; reusing the contexts, etc., al)
  - ii. Distances within a sub-set may be decided by the associated LAMVR index (e.g., AMVR\_mode in specifications).
- e. In one example, how to define sub-sets and/or how many sub-sets may be pre-defined or adaptively changed on-the-fly.

- f. In one example, the 1<sup>st</sup> syntax may be coded with truncated unary, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.
- g. In one example, the 2<sup>nd</sup> syntax may be coded with truncated unary, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.
- h. In one example, the sub-set index (e.g., 1<sup>st</sup> syntax) may be not explicitly coded in the bitstream. Alternatively, furthermore, the sub-set index may be derived on-the-fly, e.g., based on coded information of current block (e.g., block dimension) and/or previously coded blocks.
- i. In one example, the distance index within a sub-set (e.g., 2<sup>nd</sup> syntax) may be not explicitly coded in the bitstream.
  - i. In one example, when the sub-set only have one distance, there is no need to further signal the distance index.
  - ii. Alternatively, furthermore, the 2<sup>nd</sup> syntax may be derived on-the-fly, e.g., based on coded information of current block (e.g., block dimension) and/or previously coded blocks.
- j. In one example, a **first resolution bin** is signaled to indicate whether DI is smaller than a predefined number T or not. Alternatively, a **first resolution bin** is signaled to indicate whether the distance is smaller than a predefined number or not.
  - i. In one example, two syntax elements are used to represent the distance index. **mmvd\_resolution\_flag** is first signaled followed by **mmvd\_distance\_idx\_in\_subset**.
  - ii. In one example, three syntax elements are used to represent the distance index. **mmvd\_resolution\_flag** is first signaled followed by **mmvd\_short\_distance\_idx\_in\_subset** when **mmvd\_resolution\_flag** is equal to 0 or **mmvd\_long\_distance\_idx\_in\_subset** when **mmvd\_resolution\_flag** is equal to 1.
  - iii. In one example, the distance index number T corresponds to 1-Pel distance. For example, T = 2 with the Table 2a defined in VTM-3.0.

- iv. In one example, the distance index number  $T$  corresponds to 1/2-Pel distance. For example,  $T = 1$  with the Table 2a defined in VTM-3.0.
- v. In one example, the distance index number  $T$  corresponds to W-Pel distance. For example,  $T = 3$  corresponding to 2-Pel distance with the Table 2a defined in VTM-3.0.
- vi. In one example, the **first resolution bin** is equal to 0 if DI is smaller than  $T$ . Alternatively, the **first resolution bin** is equal to 1 if DI is smaller than  $T$ .
- vii. In one example, if DI is smaller than  $T$ , a code for **short distanceindex** is further signaled after the **first resolution bin** to indicate the value of DI.
  - (i) In one example, DI is signaled. DI can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.
    - a. When DI is binarized as a truncated code such as truncated unary code, the maximum coded value is  $T-1$ .
  - (ii) In one example,  $S=T-1-DI$  is signaled.  $T-1-DI$  can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.
    - a. When  $T-1-DI$  is binarized as a truncated code such as truncated unary code, the maximum coded value is  $T-1$ .
    - b. After  $S$  is parsed, DI is reconstructed as  $DI=T-S-1$ .
- viii. In one example, if DI is not smaller than  $T$ , a code for **long distanceindex** is further signaled after the **first resolution bin** to indicate the value of DI.
  - (i) In one example,  $B=DI-T$  is signaled.  $DI-T$  can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.
    - a. When  $DI-T$  is binarized as a truncated code such as truncated unary code, the maximum coded value is  $D_{Max}$

- T, where DMax is the maximum allowed distance, such as 7 in VTM-3.0.
- b. After B is parsed, DI is reconstructed as  $DI = B + T$ .
- (ii) In one example,  $B' = DMax - DI$  is signaled, where DMax is the maximum allowed distance, such as 7 in VTM-3.0. DMax-DI can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.
  - a. When DMax-DI is binarized as a truncated code such as truncated unary code, the maximum coded value is  $DMax - T$ , where DMax is the maximum allowed distance, such as 7 in VTM-3.0.
  - b. After B' is parsed, DI is reconstructed as  $DI = DMax - B'$ .
- k. In one example, a **first resolution bin** is signaled to indicate whether DI is greater than a predefined number T or not. Alternatively, a **first resolution bin** is signaled to indicate whether the distance is greater than a predefined number or not.
  - i. In one example, the distance index number T corresponds to 1-Pel distance. For example,  $T = 2$  with the Table 2a defined in VTM-3.0.
  - ii. In one example, the distance index number T corresponds to 1/2-Pel distance. For example,  $T = 1$  with the Table 2a defined in VTM-3.0.
  - iii. In one example, the distance index number T corresponds to W-Pel distance. For example,  $T = 3$  corresponding to 2-Pel distance with the Table 2a defined in VTM-3.0.
  - iv. In one example, the **first resolution bin** is equal to 0 if DI is greater than T. Alternatively, the **first resolution bin** is equal to 1 if DI is greater than T.
  - v. In one example, if DI is not greater than T, a code for **short distance index** is further signaled after the **first resolution bin** to indicate the value of DI.
    - (i) In one example, DI is signaled. DI can be binarized with unary code, truncated unary code, fixed-length code, Exponential-

Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.

a. When DI is binarized as a truncated code such as truncated unary code, the maximum coded value is T.

(ii) In one example,  $S=T-DI$  is signaled.  $T-DI$  can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.

a. When  $T-DI$  is binarized as a truncated code such as truncated unary code, the maximum coded value is T.

b. After S is parsed, DI is reconstructed as  $DI=T-S$ .

vi. In one example, if DI is greater than T, a code for **long distanceindex** is further signaled after the **first resolution bin** to indicate the value of DI.

(i) In one example,  $B=DI-1-T$  is signaled.  $DI-1-T$  can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.

a. When  $DI-1-T$  is binarized as a truncated code such as truncated unary code, the maximum coded value is  $DMax - 1 - T$ , where DMax is the maximum allowed distance, such as 7 in VTM-3.0.

b. After B is parsed, DI is reconstructed as  $DI = B+T+1$ .

(ii) In one example,  $B' = DMax-DI$  is signaled, where DMax is the maximum allowed distance, such as 7 in VTM-3.0.  $DMax-DI$  can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.

a. When  $DMax-DI$  is binarized as a truncated code such as truncated unary code, the maximum coded value is  $DMax - 1 - T$ , where DMax is the maximum allowed distance, such as 7 in VTM-3.0.

b. After  $B'$  is parsed, DI is reconstructed as  $DI = DMax-B'$ .

- I. Several possible binarization methods for the distance index are: (It should be noted that two binarization methods should be considered as identical if a process to change all “1” in one method to be “0” and all “0” in one method to be “1”, will produce the same codewords as the other method.)

| Distance<br>IDX | 0    | 1    | 2    | 3   | 4    | 5     | 6      | 7      |
|-----------------|------|------|------|-----|------|-------|--------|--------|
| Bins            | 01   | 00   | 10   | 110 | 1110 | 11110 | 111110 | 111111 |
| Bins            | 01   | 00   | 11   | 101 | 1001 | 10001 | 100001 | 100000 |
| Bins            | 00   | 01   | 10   | 110 | 1110 | 11110 | 111110 | 111111 |
| Bins            | 00   | 01   | 11   | 101 | 1001 | 10001 | 100001 | 100000 |
| Bins            | 11   | 10   | 00   | 010 | 0110 | 01110 | 011110 | 011111 |
| Bins            | 11   | 10   | 01   | 001 | 0001 | 00001 | 000001 | 000000 |
| Bins            | 10   | 11   | 00   | 010 | 0110 | 01110 | 011110 | 011111 |
| Bins            | 10   | 11   | 01   | 001 | 0001 | 00001 | 000001 | 000000 |
| Bins            | 010  | 0110 | 0111 | 10  | 110  | 1110  | 11110  | 11111  |
| Bins            | 0111 | 0110 | 010  | 10  | 110  | 1110  | 11110  | 11111  |

4. It is proposed that one or more probability contexts are used to code the 1<sup>st</sup> syntax
- In one example, the 1<sup>st</sup> syntax is the **first resolution bin** mentioned above.
  - In one example, which probability context is used is derived from the **first resolution bins** of neighbouring blocks.
  - In one example, which probability context is used is derived from the **LAMVR values** of neighbouring blocks (e.g., AMVR\_mode values).
5. It is proposed that one or more probability contexts are used to code the 2<sup>nd</sup> syntax.
- In one example, the 2<sup>nd</sup> syntax is the **short distance index** mentioned above.
    - In one example, the first bin to code the **short distance index** is coded with a probability context, and other bins are by-pass coded.
    - In one example, the first N bins to code the **short distance index** are coded with probability contexts, and other bins are by-pass coded.
    - In one example, all bins to code the **short distance index** are coded with probability contexts.
    - In one example, different bins may have different probability contexts.

- v. In one example, several bins share a single probability context.
  - (i) In one example, these bins are consecutive.
- vi. In one example, which probability context is used is derived from the **short distance indices** of neighbouring blocks.
- b. In one example, the 2<sup>nd</sup> syntax is **long distance index** mentioned above.
  - i. In one example, the first bin to code the **long distance index** is coded with a probability context, and other bins are by-pass coded.
  - ii. In one example, the first N bins to code the **long distance index** are coded with probability contexts, and other bins are by-pass coded.
  - iii. In one example, all bins to code the **long distance index** are coded with probability contexts.
  - iv. In one example, different bins may have different probability contexts.
  - v. In one example, several bins share a single probability context.
    - (i) In one example, these bins are consecutive.
  - vi. In one example, which probability context is used is derived from the **long distance indices** of neighbouring blocks.

#### Interaction with LAMVR

- 6. It is proposed that the 1<sup>st</sup> syntax (e.g., **first resolution bin**) is coded depending on the probability models used to code the LAMVR information.
  - a. In one example, the **first resolution bin** is coded with the same manner (e.g., shared context, or same context index derivation method but with neighboring blocks' LAMVR information replaced by MMVD information) as coding the first MVD resolution flag.
    - i. In one example, which probability context is used to code the **first resolution bin** is derived from the LAMVR information of neighbouring blocks.
      - (i) In one example, which probability context is used to code the **first resolution bin** is derived from the first MVD resolution flags of neighbouring blocks.
  - b. Alternatively, the first MVD resolution flag is coded and serve as the **first resolution bin** when the distance index is coded.

- c. In one example, which probability model is used to code the **first resolution bin** may depend on the coded LAMVR information.
    - i. For example, which probability model is used to code the **first resolution bin** may depend on the MV resolution of neighbouring blocks.
- 7. It is proposed that the first bin to code the **short distance index** is coded with a probability context.
  - a. In one example, the first bin to code the **short distance index** is coded with the same manner (e.g., shared context, or same context index derivation method but with neighboring blocks' LAMVR information replaced by MMVD information) as coding the second MVD resolution flag.
  - b. Alternatively, the second MVD resolution flag is coded and serve as the first bin to code the **short distance index** when the distance index is coded.
  - c. In one example, which probability model is used to code the first bin to code the **short distance index** may depend on the coded LAMVR information.
    - i. For example, which probability model is used to code the first bin to code the **short distance index** may depend on the MV resolution of neighbouring blocks.
- 8. It is proposed that the first bin to code the **long distance index** is coded with a probability context.
  - a. In one example, the first bin to code the **long distance index** is coded with the same manner (e.g., shared context, or same context index derivation method but with neighboring blocks' LAMVR information replaced by MMVD information) as coding the second MVD resolution flag.
  - b. Alternatively, the second MVD resolution flag is coded and serve as the first bin to code the **long distance index** when the distance index is coded.
  - c. In one example, which probability model is used to code the first bin to code the **long distance index** may depend on the coded LAMVR information.
    - i. For example, which probability model is used to code the first bin to code the **long distance index** may depend on the MV resolution of neighbouring blocks.
- 9. For the LAMVR mode, in arithmetic coding, the first MVD resolution flag is coded with one of three probability contexts: C0, C1 or C2; while the second MVD resolution flag is

coded with a forth probability context: C3. Examples to derive the probability context to code the distance index is described as below.

- a. The probability context Cx for the **first resolution bin** is derived as (L represents the left neighbouring block and A represents the above neighbouring block):  
 If L is available, inter-coded, and its first MVD resolution flag is not equal to zero, xL is set equal to 1; otherwise, xL is set equal to 0.  
 If A is available, inter-coded, and its first MVD resolution flag is not equal to zero, xA is set equal to 1; otherwise, xA is set equal to 0.  
 x is set equal to xL+xA.
  - b. The probability context for the first bin to code the **long distance index** is C3.
  - c. The probability context for the first bin to code the **short distance index** is C3.
10. It is proposed that the LAMVR MVD resolution is signaled when MMVD mode is applied.
- a. It is proposed that the syntax used LAMVR MVD resolution signaling is reused when coding the side information of MMVD mode.
  - b. When the signaled LAMVR MVD resolution is ¼-pel, a **short distance index** is signaled to indicate the MMVD distance in a first sub-set. For example, the **short distance index** can be 0 or 1, to represent the MMVD distance to be ¼-pel or 1/2 -pel, respectively.
  - c. When the signaled LAMVR MVD resolution is 1-pel, a **medium distance index** is signaled to indicate the MMVD distance in a second sub-set. For example, the **medium distance index** can be 0 or 1, to represent the MMVD distance to be 1-pel or 2-pel, respectively.
  - d. When the signaled LAMVR MVD resolution is 4-pel in a third sub-set, a **long distance index** is signaled to indicate the MMVD distance. For example, the **medium distance index** can be X, to represent the MMVD distance to be (4<<X)-pel.
  - e. In the following disclosure, a sub-set distance index may refer to a **short distance index**, ora **medium distance index**, or a **long distance index**.
    - i. In one example, the sub-set distance index can be binarized with unary code, truncated unary code, fixed-length code, Exponential-Golomb code, truncated Exponential-Golomb code, Rice code, or any other codes.

- (i) Specifically, a sub-set distance index can be binarized as a flag if there are only two possible distances in the sub-set.
  - (ii) Specifically, a sub-set distance index is not signaled if there is only one possible distance in the sub-set.
  - (iii) Specifically, the maximum value is set to be the number of possible distance in the sub-set minus 1 if a sub-set distance index is binarized as a truncated code.
  - ii. In one example, the first bin to code the sub-set distance index is coded with a probability context, and other bins are by-pass coded.
  - iii. In one example, the first N bins to code the sub-set distance index are coded with probability contexts, and other bins are by-pass coded.
  - iv. In one example, all bins to code the sub-set distance index are coded with probability contexts.
  - v. In one example, different bins may have different probability contexts.
  - vi. In one example, several bins share a single probability context.
    - (i) In one example, these bins are consecutive.
  - vii. It is proposed that one distance cannot appear in two different distance sub-sets.
  - viii. In one example, there may be more distances can be signaled in the short distance sub-set.
    - (i) For example, the distances signaled in the short distance sub-set must be in a sub-pel, but not an integer-pel. For example,  $5/4$ -pel,  $3/2$ -pel  $7/4$ -pel may be in the short distance sub-set, but 3-pel cannot be in the short distance sub-set.
  - ix. In one example, there may be more distances can be signaled in the medium distance sub-set.
    - (i) For example, the distances signaled in the medium distance sub-set must be integer-pel but not in a form of  $4N$ , where N is an integer. For example, 3-pel, 5-pel may be in the medium distance sub-set, but 24-pel cannot be in the medium distance sub-set.
  - x. In one example, there may be more distances can be signaled in the long distance sub-set.
    - (i) For example, the distances signaled in the long distance sub-set must be integer-pel in a form of  $4N$ , where N is an integer. For example, 4-pel, 8-pel, 16-pel or 24-pel may be in the long distance sub-set.
11. It is proposed that a variable to store the MV resolution of the current block may be decided by the UMVE distance.

- a. In one example, the MV resolution of the current block is set to be  $\frac{1}{4}$ -Pel if the UMVE distance  $< T1$  or  $\leq T1$ .
  - b. In one example, the first and second MVD resolution flags of the current block are set to be 0 if the UMVE distance  $< T1$  or  $\leq T1$ .
  - c. In one example, the MV resolution of the current block is set to be 1-Pel if the UMVE distance  $> T1$  or  $\geq T1$ .
  - d. In one example, the first MVD resolution flag of the current block is set to be 1, and the second MVD resolution flag of the current block is set to be 0, if the UMVE distance  $> T1$  or  $\geq T1$ .
  - e. In one example, the MV resolution of the current block is set to be 4-Pel if the UMVE distance  $> T2$  or  $\geq T2$ .
  - f. In one example, the first and second MVD resolution flags of the current block are set to be 1, if the UMVE distance  $> T2$  or  $\geq T2$ .
  - g. In one example, the MV resolution of the current block is set to be 1-Pel if the UMVE distance  $> T1$  or  $\geq T1$  and the UMVE distance  $< T2$  or  $\leq T2$ .
  - h. In one example, the first MVD resolution flag of the current block is set to be 1, and the second MVD resolution flag of the current block is set to be 0, if the UMVE distance  $> T1$  or  $\geq T1$  and the UMVE distance  $< T2$  or  $\leq T2$ .
  - i.  $T1$  and  $T2$  can be any numbers. For example,  $T1 = 1$ -Pel and  $T2 = 4$ -Pel.
12. It is proposed that a variable to store the MV resolution of the current block may be decided by the UMVE distance index.
- a. In one example, the MV resolution of the current block is set to be  $\frac{1}{4}$ -Pel if the UMVE distance index  $< T1$  or  $\leq T1$ .
  - b. In one example, the first and second MVD resolution flags of the current block are set to be 0 if the UMVE distance index  $< T1$  or  $\leq T1$ .
  - c. In one example, the MV resolution of the current block is set to be 1-Pel if the UMVE distance index  $> T1$  or  $\geq T1$ .
  - d. In one example, the first MVD resolution flag of the current block is set to be 1, and the second MVD resolution flag of the current block is set to be 0, if the UMVE distance index  $> T1$  or  $\geq T1$ .
  - e. In one example, the MV resolution of the current block is set to be 4-Pel if the UMVE distance index  $> T2$  or  $\geq T2$ .

- f. In one example, the first and second MVD resolution flags of the current block are set to be 1, if the UMVE distance index  $> T2$  or  $\geq T2$ .
  - g. In one example, the MV resolution of the current block is set to be 1-Pel if the UMVE distance  $> T1$  or  $\geq T1$  and the UMVE distance index  $< T2$  or  $\leq T2$ .
  - h. In one example, the first MVD resolution flag of the current block is set to be 1, and the second MVD resolution flag of the current block is set to be 0, if the UMVE distance index  $> T1$  or  $\geq T1$  and the UMVE distance index  $< T2$  or  $\leq T2$ .
  - i.  $T1$  and  $T2$  can be any numbers. For example,  $T1 = 2$  and  $T2 = 3$ , or  $T1 = 2$  and  $T2 = 4$ ;
13. The variable to store MV resolution of UMVE coded blocks may be utilized for coding following blocks which are coded with LAMVR mode.
- a. Alternatively, the variable to store MV resolution of UMVE coded blocks may be utilized for coding following blocks which are coded with UMVE mode.
  - b. Alternatively, the MV precisions of LAMVR-coded blocks may be utilized for coding following UMVE-coded blocks.
14. The above bullets may be also applicable to coding the direction index.

#### Mapping between distance index and distance

15. It is proposed that the relationship between the distance index (DI) and the distance may not be the exponential relationship as in VTM-3.0. (Distance =  $\frac{1}{4}$ -Pel  $\times 2^{DI}$ )
- a. In one example, the mapping may be piece-wised.
    - i. For example, Distance =  $f1(DI)$  when  $T0 \leq DI < T1$ , Distance =  $f2(DI)$  when  $T1 \leq DI < T2$ , ... Distance =  $fn(DI)$  when  $T_{n-1} \leq DI < T_n$ ....
      - (i) For example, Distance =  $\frac{1}{4}$ -Pel  $\times 2^{DI}$  when  $DI < T1$ , Distance =  $a \times DI + b$  when  $T1 \leq DI < T2$ , and Distance =  $c \times 2^{DI}$  when  $DI \geq T2$ . In one example,  $T1 = 4$ ,  $a = 1$ ,  $b = -1$ ,  $T2 = 6$ ,  $c = 1/8$ .
16. It is proposed that distance table size may be larger than 8, such as 9, 10, 12, 16.
17. It is proposed that distance shorter than 1/4-pel may be included in the distance table, such as 1/8-pel, 1/16-pel or 3/8-pel.
18. It is proposed than distances not in the form of  $2^X$ -pel may be included in the distance table, such as 3-pel, 5-pel, 6-pel, etc.
19. It is proposed that the distance table may be different for different directions.

- a. Correspondingly, the parsing process for the distance index may be different for different directions.
  - b. In one example, four directions with the direction index 0, 1, 2 and 3 have different distance tables.
  - c. In one example, two x-directions with the direction index 0 and 1 have the same distance table.
  - d. In one example, two y-directions with the direction index 2 and 3 have the same distance table.
  - e. In one example, x-directions and y-directions may have two different distance tables.
    - i. Correspondingly, the parsing process for the distance index may be different for x-directions and y-directions.
    - ii. In one example, the distance table for y-directions may have less possible distances than the distance table for x-directions.
    - iii. In one example, the shortest distance in the distance table for y-directions may be shorter than the shortest distance in the distance table for x-directions.
    - iv. In one example, the longest distance in the distance table for y-directions may be shorter than the longest distance in the distance table for x-directions.
20. It is proposed that different distance tables may be used for different block width and/or height.
- a. In one example, different distance tables may be used for different block width when the direction is along the x-axis.
  - b. In one example, different distance tables may be used for different block height when the direction is along the y-axis.
21. It is proposed that different distance tables may be used when the POC distance is different. POC difference is calculated as  $|\text{POC of the current picture} - \text{POC of the reference picture}|$ .
22. It is proposed that different distance tables may be used for different base candidates.
23. It is proposed that for the ratio of two distances (MVD precisions) with consecutive indices is not fixed to be 2.

- a. In one example, the ratio of two distances (MVD precisions) with consecutive indices is fixed to M (e.g.,  $M = 4$ ).
- b. In one example, the delta of two distances (MVD precisions) with consecutive indices (instead of ratios) may be fixed for all indices. Alternatively, the delta of two distances (MVD precisions) with consecutive indices may be different for different indices.
- c. In one example, the ratio of two distances (MVD precisions) with consecutive indices may be different for different indices.
  - i. In one example, the set of distances such as {1-pel, 2-pel, 4-pel, 8-pel, 16-pel, 32-pel, 48-pel, 64-pel} may be used.
  - ii. In one example, the set of distances such as {1-pel, 2-pel, 4-pel, 8-pel, 16-pel, 32-pel, 64-pel, 96-pel} may be used.
  - iii. In one example, the set of distances such as {1-pel, 2-pel, 3-pel, 4-pel, 5-pel, 16-pel, 32-pel} may be used.

24. The signaling of MMVD side information may be done in the following way:

- a. When the current block is inter and non-merge (which may include e.g., non-skip, non-sub-block, non-triangular, non-MHIntra) mode, a MMVD flag may be firstly signaled, followed by a sub-set index of distance, a distance index within the sub-set, a direction index. Here, MMVD is treated as a different mode from merge mode.
- b. Alternatively, when the current block is merge mode, a MMVD flag may be further signaled, followed by a sub-set index of distance, a distance index within the sub-set, a direction index. Here, MMVD is treated as a special merge mode.

25. The direction of MMVD and the distance of MMVD may be signaled jointly.

- a. In one example, whether and how to signal MMVD distance may depend on MMVD direction.
- b. In one example, whether and how to signal MMVD direction may depend on MMVD distance.
- c. In one example, a joint codeword is signaled with one or more syntax elements. The MMVD distance and MMVD direction can be derived from the code word. For example, the codeword is equal to MMVD distance index + MMVD direction

index \* 7. In another example, the MMVD a codeword table is designed. Each codeword corresponds to a unique combination of MMVD distance and MMVD direction.

26. Some exemplary UMVE distances tables are listed below:

a. The distance tables size is 9:

| Distance<br>IDX   | 0           | 1           | 2           | 3     | 4     | 5     | 6      | 7      | 8      |
|-------------------|-------------|-------------|-------------|-------|-------|-------|--------|--------|--------|
| Pixel<br>distance | 1/8-<br>pel | 1/4-<br>pel | 1/2-<br>pel | 1-pel | 2-pel | 4-pel | 8-pel  | 16-pel | 32-pel |
| Pixel<br>distance | 1/4-<br>pel | 1/2-<br>pel | 1-pel       | 2-pel | 3-pel | 4-pel | 8-pel  | 16-pel | 32-pel |
| Pixel<br>distance | 1/4-<br>pel | 1/2-<br>pel | 1-pel       | 2-pel | 4-pel | 6-pel | 8-pel  | 16-pel | 32-pel |
| Pixel<br>distance | 1/4-<br>pel | 1/2-<br>pel | 1-pel       | 2-pel | 4-pel | 8-pel | 12-pel | 16-pel | 32-pel |
| Pixel<br>distance | 1/4-<br>pel | 1/2-<br>pel | 1-pel       | 2-pel | 4-pel | 8-pel | 16-pel | 24-pel | 32-pel |

b. The distance tables size is 10:

| Distance<br>IDX   | 0           | 1           | 2           | 3     | 4     | 5     | 6     | 7          | 8          | 9          |
|-------------------|-------------|-------------|-------------|-------|-------|-------|-------|------------|------------|------------|
| Pixel<br>distance | 1/8-<br>pel | 1/4-<br>pel | 1/2-<br>pel | 1-pel | 2-pel | 4-pel | 8-pel | 16-<br>pel | 32-<br>pel | 64-<br>pel |
| Pixel<br>distance | 1/8-<br>pel | 1/4-<br>pel | 1/2-<br>pel | 1-pel | 2-pel | 3-pel | 4-pel | 8-pel      | 16-<br>pel | 32-<br>pel |
| Pixel<br>distance | 1/8-<br>pel | 1/4-<br>pel | 1/2-<br>pel | 1-pel | 2-pel | 4-pel | 6-pel | 8-pel      | 16-<br>pel | 32-<br>pel |
| Pixel<br>distance | 1/8-<br>pel | 1/4-<br>pel | 1/2-<br>pel | 1-pel | 2-pel | 4-pel | 8-pel | 12-<br>pel | 16-<br>pel | 32-<br>pel |
| Pixel<br>distance | 1/8-<br>pel | 1/4-<br>pel | 1/2-<br>pel | 1-pel | 2-pel | 4-pel | 8-pel | 16-<br>pel | 24-<br>pel | 32-<br>pel |
| Pixel             | 1/4-        | 1/2-        | 1-pel       | 2-pel | 3-pel | 4-pel | 6-pel | 8-pel      | 16-        | 32-        |

|                       |         |         |       |       |       |       |       |        |        |        |
|-----------------------|---------|---------|-------|-------|-------|-------|-------|--------|--------|--------|
| <b>distance</b>       | pel     | pel     |       |       |       |       |       |        | pel    | pel    |
| <b>Pixel distance</b> | 1/4-pel | 1/2-pel | 1-pel | 2-pel | 4-pel | 6-pel | 8-pel | 12-pel | 16-pel | 32-pel |
| <b>Pixel distance</b> | 1/4-pel | 1/2-pel | 1-pel | 2-pel | 3-pel | 4-pel | 8-pel | 12-pel | 16-pel | 32-pel |
| <b>Pixel distance</b> | 1/4-pel | 1/2-pel | 1-pel | 2-pel | 3-pel | 4-pel | 8-pel | 16-pel | 24-pel | 32-pel |

a. The distance tables size is 12:

|                       |         |         |         |       |       |       |       |       |        |        |        |        |
|-----------------------|---------|---------|---------|-------|-------|-------|-------|-------|--------|--------|--------|--------|
| <b>Distance IDX</b>   | 0       | 1       | 2       | 3     | 4     | 5     | 6     | 7     | 8      | 9      | 10     | 11     |
| <b>Pixel distance</b> | 1/8-pel | 1/4-pel | 1/2-pel | 1-pel | 2-pel | 3-pel | 4-pel | 6-pel | 8-pel  | 16-pel | 32-pel | 64-pel |
| <b>Pixel distance</b> | 1/8-pel | 1/4-pel | 1/2-pel | 1-pel | 2-pel | 3-pel | 4-pel | 8-pel | 12-pel | 16-pel | 24-pel | 32-pel |
| <b>Pixel distance</b> | 1/4-pel | 1/2-pel | 1-pel   | 2-pel | 3-pel | 4-pel | 6-pel | 8-pel | 12-pel | 16-pel | 24-pel | 32-pel |
| <b>Pixel distance</b> | 1/4-pel | 1/2-pel | 1-pel   | 2-pel | 3-pel | 4-pel | 6-pel | 8-pel | 12-pel | 16-pel | 32-pel | 64-pel |

27. It is proposed that MMVD distance may be signaled in a granularity-signaling method.

The distance is firstly signaled by index with a rough granularity, followed by one or more indices with finer granularities.

- For example, a first index  $F_1$  represents distances in an ordered set  $M_1$ ; a second index  $F_2$  represents distances in an ordered set  $M_2$ . The final distance is calculated as, such as  $M_1[F_1] + M_2[F_2]$ .
- For example, a first index  $F_1$  represents distances in an ordered set  $M_1$ ; a second index  $F_2$  represents distances in an ordered set  $M_2$ ; and so on, until a  $n$ th index  $F_n$  represents distances in an ordered set  $M_n$ . The final distance is calculated as  $M_1[F_1] + M_2[F_2] + \dots + M_n[F_n]$ .
- For example, the signaling or binarization of  $F_k$  may depend on the signaled  $F_{k-1}$ .

- i. In one example, an entry in  $M_k[F_k]$  must be smaller than  $M_{k-1}[F_{k-1}+1]-M_{k-1}[F_{k-1}]$  when  $F_{k-1}$  is not the largest index referring to  $M_{k-1}$ .  $1 < k \leq n$ .
- d. For example, the signaling or binarization of  $F_k$  may depend on the signaled  $F_s$  for all  $1 \leq s < k$ .
  - i. In one example, an entry in  $M_k[F_k]$  must be smaller than  $M_s[F_s+1]-M_s[F_s]$  for all  $1 \leq s < k$ , when  $1 < k \leq n$ .
- e. In one example, if  $F_k$  is the largest index referring to  $M_k$ , then no further  $F_{k+1}$  is signaled, and the final distance is calculated as  $M_1[F_1]+M_2[F_2]+\dots+M_k[F_k]$ , where  $1 \leq k \leq n$ .
- f. In one example, the entries in  $M_k[F_k]$  may depend on the signaled  $F_{k-1}$ .
- g. In one example, the entries in  $M_k[F_k]$  may depend on the signaled  $F_s$  for all  $1 \leq s < k$ .
- h. For example,  $n=2$ .  $M_1 = \{1/4\text{-pel}, 1\text{-pel}, 4\text{-pel}, 8\text{-pel}, 16\text{-pel}, 32\text{-pel}, 64\text{-pel}, 128\text{-pel}\}$ ,
  - i.  $M_2 = \{0\text{-pel}, 1/4\text{-pel}\}$  when  $F_1 = 0$  ( $M_1[F_1]=1/4\text{-pel}$ );
  - ii.  $M_2 = \{0\text{-pel}, 1\text{-pel}, 2\text{-pel}\}$  when  $F_1 = 1$  ( $M_1[F_1]=1\text{-pel}$ );
  - iii.  $M_2 = \{0\text{-pel}, 1\text{-pel}, 2\text{-pel}, 3\text{-pel}\}$  when  $F_1 = 2$  ( $M_1[F_1]=4\text{-pel}$ );
  - iv.  $M_2 = \{0\text{-pel}, 2\text{-pel}, 4\text{-pel}, 6\text{-pel}\}$  when  $F_1 = 3$  ( $M_1[F_1]=8\text{-pel}$ );
  - v.  $M_2 = \{0\text{-pel}, 4\text{-pel}, 8\text{-pel}, 12\text{-pel}\}$  when  $F_1 = 4$  ( $M_1[F_1]=16\text{-pel}$ );
  - vi.  $M_2 = \{0\text{-pel}, 8\text{-pel}, 16\text{-pel}, 24\text{-pel}\}$  when  $F_1 = 5$  ( $M_1[F_1]=32\text{-pel}$ );
  - vii.  $M_2 = \{0\text{-pel}, 16\text{-pel}, 32\text{-pel}, 48\text{-pel}\}$  when  $F_1 = 6$  ( $M_1[F_1]=32\text{-pel}$ );

#### Slice/Picture Level control

28. It is proposed that how to signal the MMVD side information (e.g., MMVD distance) and/or how to interpret the signaled MMVD side information (e.g., distance index to the distance), may depend on information signaled or inferred at a level higher than CU level. (e.g. sequence level, or picture level or slice level, or tile group level, such as in VPS/SPS/PPS/slice header/picture header/tile group header).
- a. In one example, a code table index is signaled or inferred at a higher level. A specific code table is determined by the table index. The distance index may be signaled with the methods disclosed in item 1-26. Then the distance is derived by querying the entry with the signaled distance index in the specific code table.

- b. In one example, a parameter  $X$  is signaled or inferred at a higher level. The distance index may be signaled with the methods disclosed in item 1-26. Then a distance  $D'$  is derived by querying the entry with the signaled distance index in a code table. Then the final distance  $D$  is calculated as  $D=f(D', X)$ .  $f$  can be any function. For example,  $f(D', X) = D' \ll X$  or  $f(D', X) = D' * X$ , or  $f(D', X) = D' + X$ , or  $f(D', X) = D'$  right shifted by  $X$  (with rounding or not).
- c. In one example, valid MV resolution is signaled or inferred at a higher level. Only the MMVD distances with valid MV resolution can be signaled.
  - i. For example, the signaling method of MMVD information at CU level may depend on the valid MV resolutions signaled at a higher level.
    - (i) For example, the signaling method of MMVD distance resolution information at CU level may depend on the valid MV resolutions signaled at a higher level.
    - (ii) For example, the number of distance sub-sets may depend on the valid MV resolutions signaled at a higher level.
    - (iii) For example, the meaning of each sub-set may depend on the valid MV resolutions signaled at a higher level.
  - ii. For example, a minimum MV resolution (such as  $\frac{1}{4}$ -pel, or 1-pel, or 4-pel) is signaled.
    - (i) For example, when the minimum MV resolution is  $\frac{1}{4}$ -pel, the distance index is signaled as disclosed in item 1-26.
    - (ii) For example, when the minimum MV resolution is 1-pel, the flag to signal whether the distance resolution is  $\frac{1}{4}$ -pel (such as the first resolution bin in LAMVR) is not signaled. Only medium distance index and long distance index disclosed in Item 10 can be signaled following the LAMVR information.
    - (iii) For example, when the minimum MV resolution is 4-pel, the flag to signal whether the distance resolution is  $\frac{1}{4}$ -pel (such as the first resolution bin in LAMVR) is not signaled; and the flag to signal whether the distance resolution is 1-pel (such as the second resolution bin in LAMVR) is not signaled. Only long distance index

disclosed in Item 10 can be signaled following the LAMVR information.

(iv) For example, when the minimum MV resolution is 1-pel, the distance resolution is signaled in the same manner as when the minimum MV resolution is 1/4-pel. But the meanings of distance sub-sets may be different.

a. For example, the short distance sub-set represented by the short distance index is redefined as the very-long distance sub-set. For example, two distances that can be signaled within this very-long sub-set are 64-pel and 128-pel.

29A. It is proposed the encoder may decide whether the slice/tile/picture/sequence/group of CTUs/group of blocks is screen content or not by checking the ratio of the block that has one or more similar or identical blocks within the same slice/tile/picture/sequence/group of CTUs/group of blocks.

- a. In one example, if the ratio is larger than a threshold, it is considered as screen content.
- b. In one example, if the ratio is larger than a first threshold and smaller than a second threshold, it is considered as screen content.
- c. In one example, the slice/tile/picture/sequence/group of CTUs/group of blocks may be split into MxN non-overlapped blocks. For each MxN block, the encoder checks whether there is another (or more) MxN block is similar with or identical to it. For example, MxN is equal to 4x4.
- d. In one example, only partial of the blocks are checked when calculating the ratio. For example, only blocks in even row and even columns are checked.
- e. In one example, a key value, e.g., cyclic redundancy check (CRC) code, may be generated for each MxN block, key values of two blocks are compared to check whether the two blocks are identical or not.
  - i. In one example, the key value may be generated using only some color components of the block. For example, the key value is generated by using luma component only.
  - ii. In one example, the key value may be generated using only some pixels of the block. For example, only even rows of the block are used.

- f. In one example, SAD/SATD/SSE or mean-removed SAD/SATD/SSE may be used to measure the similarity of two blocks.
  - i. In one example, SAD/SATD/SSE or mean-removed SAD/SATD/SSE may be only calculated for some pixels. For example, SAD/SATD/SSE or mean-removed SAD/SATD/SSE is only calculated for the even rows.

#### Affine MMVD

29B. It is proposed that the indications of usage of Affine MMVD may be signaled only when the merge index of the sub-block merge list is larger than K (where K = 0 or 1).

- a. Alternatively, when there are separate lists for affine merge list and other merge lists (such as ATMVP list), the indication of usage of affine MMVD may be signaled only when the affine mode is enabled. Alternatively, furthermore, the indication of usage of affine MMVD may be signaled only when the affine mode is enabled and there are more than 1 base affine candidate.

30. It is proposed that the MMVD method can also be applied to other sub-block based coding tools in addition to affine mode, such as ATMVP mode. In one example, if the current CU applies ATMVP, and MMVD on/off flag is set to be 1, MMVD is applied to ATMVP.

- a. In one example, one set of MMVD side information may be applied to all sub-blocks, in this case, one set of MMVD side information signaled. Alternatively, different sub-blocks may choose different sets, in this case, multiple sets of MMVD side information may be signaled.
- b. In one embodiment, the MV of each sub-block is added with the signaled MVD (a.k.a. offset or distance)
- c. In one embodiment, the method to signal the MMVD information when the sub-block merge candidate is an ATMVP merge candidate is the same to the method when the sub-block merge candidate is an affine merge candidate.
- d. In one embodiment, a POC distance based offset mirroring method is used for Bi-prediction to add the MVD on the MV of each sub-block when the sub-block merge candidate is an ATMVP merge candidate.

31. It is proposed that the MV of each sub-block is added with the signaled MVD (a.k.a. offset or distance) when the sub-block merge candidate is an affine merge candidate.

32. It is proposed that the MMVD signaling methods disclosed in bullet 1- 28 can also be applied to signal the MVD used by the affine MMVD mode.
- a. In one embodiment, the LAMVR information used to signal the MMVD information of affine MMVD mode may be different from the LAMVR information used to signal the MMVD information of non-affine MMVD mode.
    - i. For example, the LAMVR information used to signal the MMVD information of affine MMVD mode are also used to signal the MV precision used in affine inter-mode; but the LAMVR information used to signal the MMVD information of non-affine MMVD mode is used to signal the MV precision used in non-affine inter-mode.
33. It is proposed that the MVD information in MMVD mode for sub-block merge candidates should be signaled in the same way as the MVD information in MMVD mode for regular merge candidates.
- a. For example, they share the same distance table;
  - b. For example, they share the same mapping between the a distance index and a distance.
  - c. For example, they share the same directions.
  - d. For example, they share the same binarization methods.
  - e. For example, they share the same arithmetic coding contexts.
34. It is proposed that the MMVD side information signalling may be dependent on the coded mode, such as affine or normal merge or triangular merge mode or ATMVP mode.
35. It is proposed that the pre-defined MMVD side information may be dependent on the coded mode, such as affine or normal merge or triangular merge mode or ATMVP mode.
36. It is proposed that the pre-defined MMVD side information may be dependent on the color subsampling method (e.g., 4:2:0, 4:2:2, 4:4:4), and/or color component.

#### Triangle MMVD

37. It is proposed that MMVD can be applied on triangular prediction mode.
- a. After a TPM merge candidate is signaled, the MMVD information is signaled. The signaled TPM merge candidate is treated as the base merge candidate.
    - i. For example, the MMVD information is signaled with the same signaling method as the MMVD for regular merge;

- ii. For example, the MMVD information is signaled with the same signaling method as the MMVD for affine merge or other kinds of sub-block merge;
    - iii. For example, the MMVD information is signaled with a signaling method different to that of the MMVD for regular merge, affine merge or other kinds of sub-block merge;
  - b. In one example, the MV of each triangle partition is added with the signaled MVD;
  - c. In one example, the MV of one triangle partition is added with the signaled MVD, and the MV of the other triangle partition is added with  $f(\text{signaled MVD})$ ,  $f$  is any function.
    - i. In one example,  $f$  depends on the reference picture POCs or reference indices of the two triangle partitions.
    - ii. In one example,  $f(\text{MVD}) = -\text{MVD}$  if the reference picture of one triangle partition is before the current picture in displaying order and the reference picture of the other triangle partition is after the current picture in displaying order.
38. It is proposed that the MMVD signaling methods disclosed in bullet 1- 28 can also be applied to signal the MVD used by the triangular MMVD mode.
- a. In one embodiment, the LAMVR information used to signal the MMVD information of affine MMVD mode may be different from the LAMVR information used to signal the MMVD information of non-affine MMVD mode.
    - i. For example, the LAMVR information used to signal the MMVD information of affine MMVD mode are also used to signal the MV precision used in affine inter-mode; but the LAMVR information used to signal the MMVD information of non-affine MMVD mode is used to signal the MV precision used in non-affine inter-mode.
39. For all above bullets, the MMVD side information may include e.g., the offset table (distance), and direction information.

## **[00178] 5. Example embodiments**

**[00179]** This section shows some embodiments for the improved MMVD design.

### **[00180] 5.1 Embodiment #1 (MMVD distance index coding)**

In one embodiment, to code MMVD distance, a first resolution bin is coded. For example, it may be coded with the same probability context as the first flag of MV resolution.

- If the resolution bin is 0, a following flag is coded. For example, it may be coded with another probability context to indicate the short distance index. If the flag is 0, the index is 0; if the flag is 1, the index is 1.
- Otherwise (the resolution bin is 0), the long distance index L is coded as a truncated unary code, with the maximum value MaxDI -2, where MaxDI is the maximum possible distance index, equal to 7 in the embodiment. After paring out L, the distance index is reconstructed as L+2. In an exemplary C-style embodiment:

```

DI = 2;
while( DI<7 ){
    oneBin = parseOneBin();
    if(oneBin == 0){
        break;
    }
    DI++;
}

```

The first bin of the long distance index is coded with a probability context, and other bins are by-pass coded. In a C-style embodiment:

```

DI = 2;
while( DI<7 ){
    if( DI==2){
        oneBin = parseOneBinWithContext();
    }
    else{
        oneBin = parseOneBinBypass();
    }
    if(oneBin == 0){
        break;
    }
    DI++;
}

```

An example of proposed syntax changes are **highlighted**, and the deleted parts are marked with ~~strikethrough~~.

|                                                |       |
|------------------------------------------------|-------|
| ...                                            | ae(v) |
| <b>mmvd_flag[ x0 ][ y0 ]</b>                   |       |
| if( <del>mmvd_flag[ x0 ][ y0 ]</del> == 1 ) {  |       |
| <b>mmvd_merge_flag[ x0 ][ y0 ]</b>             | ae(v) |
| <b>mmvd_distance_idx[ x0 ][ y0 ]</b>           | ae(v) |
| <b>mmvd_distance_subset_idx[ x0 ][ y0 ]</b>    | ae(v) |
| <b>mmvd_distance_idx_in_subset[ x0 ][ y0 ]</b> | ae(v) |

|                                       |              |
|---------------------------------------|--------------|
| <b>mmvd_direction_idx[ x0 ][ y0 ]</b> | <b>ae(v)</b> |
| ...                                   |              |

In one example, the **mmvd\_distance\_subset\_idx** represents the resolution index as mentioned above, **mmvd\_distance\_idx\_in\_subset** represents the short or long distance index according to the resolution index. Truncated unary may be used to code **mmvd\_distance\_idx\_in\_subset**. In a Random Access test under the common test condition, the embodiment can achieve 0.15% coding gain in average and 0.34% gain on UHD sequences (class A1).

|                | Y      | U      | V      |
|----------------|--------|--------|--------|
| Class A1       | -0.34% | -0.86% | -0.50% |
| Class A2       | -0.15% | 0.30%  | 0.07%  |
| Class B        | -0.09% | 0.13%  | -0.44% |
| Class C        | -0.08% | 0.05%  | 0.09%  |
| Class E        |        |        |        |
| <b>Overall</b> | -0.15% | -0.06% | -0.21% |
| Class D        | -0.05% | 0.23%  | -0.10% |
| Class F        | -0.18% | -0.08% | -0.26% |

## [00181] 5.2 Embodiment #2 (MMVD side information coding)

MMVD is treated as a separate mode which is not treated as a merge mode. Therefore, MMVD flag may be further coded only when merge flag is 0.

| <b>coding_unit( x0, y0, cbWidth, cbHeight, treeType ) {</b> | <b>Descriptor</b> |
|-------------------------------------------------------------|-------------------|
| <b>if( slice_type != 1 ) {</b>                              |                   |
| <b>cu_skip_flag[ x0 ][ y0 ]</b>                             | <b>ae(v)</b>      |
| <b>if( cu_skip_flag[ x0 ][ y0 ] == 0 )</b>                  |                   |
| <b>pred_mode_flag</b>                                       | <b>ae(v)</b>      |
| <b>}</b>                                                    |                   |
| <b>if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {</b>         |                   |
| ...                                                         |                   |
| <b>}</b>                                                    |                   |
| <b>} else { /* MODE_INTER */</b>                            |                   |
| <b>if( cu_skip_flag[ x0 ][ y0 ] == 0 ) {</b>                |                   |
| <b>merge_flag[ x0 ][ y0 ]</b>                               | <b>ae(v)</b>      |
| <b>if( merge_flag[ x0 ][ y0 ] ) {</b>                       |                   |
| <b>merge_data( x0, y0, cbWidth, cbHeight )</b>              |                   |
| <b>} else {</b>                                             |                   |
| <b>mmvd_flag[ x0 ][ y0 ]</b>                                | <b>ae(v)</b>      |
| <b>if( mmvd_flag[ x0 ][ y0 ] == 1 ) {</b>                   |                   |
| <b>mmvd_merge_flag[ x0 ][ y0 ]</b>                          | <b>ae(v)</b>      |
| <b>mmvd_distance_subset_idx[ x0 ][ y0 ]</b>                 | <b>ae(v)</b>      |
| <b>mmvd_distance_idx_in_subset[ x0 ][ y0 ]</b>              | <b>ae(v)</b>      |
| <b>mmvd_direction_idx[ x0 ][ y0 ]</b>                       | <b>ae(v)</b>      |
| <b>} else {</b>                                             |                   |

|                                                                    |       |
|--------------------------------------------------------------------|-------|
| if( slice_type == B )                                              |       |
| <b>inter_pred_idc</b> [ x0 ][ y0 ]                                 | ae(v) |
| if( sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16 ) { |       |
| <b>inter_affine_flag</b> [ x0 ][ y0 ]                              | ae(v) |
| if( sps_affine_type_flag && inter_affine_flag[ x0 ][ y0 ] )        |       |
| <b>cu_affine_type_flag</b> [ x0 ][ y0 ]                            | ae(v) |
| }                                                                  |       |
| ...                                                                |       |
| }                                                                  |       |
| }                                                                  |       |

|                                                                                                                                          |                   |
|------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| merge_data( x0, y0, cbWidth, cbHeight ) {                                                                                                | <b>Descriptor</b> |
| <del>mmvd_flag</del> [ x0 ][ y0 ]                                                                                                        | <del>ae(v)</del>  |
| <del>if( mmvd_flag[ x0 ][ y0 ] == 1 ) {</del>                                                                                            |                   |
| <del>mmvd_merge_flag</del> [ x0 ][ y0 ]                                                                                                  | <del>ae(v)</del>  |
| <del>mmvd_distance_idx</del> [ x0 ][ y0 ]                                                                                                | <del>ae(v)</del>  |
| <del>mmvd_direction_idx</del> [ x0 ][ y0 ]                                                                                               | <del>ae(v)</del>  |
| <del>} else {</del>                                                                                                                      |                   |
| if( MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8 )                                                                       |                   |
| <b>merge_subblock_flag</b> [ x0 ][ y0 ]                                                                                                  | ae(v)             |
| if( merge_subblock_flag[ x0 ][ y0 ] == 1 ) {                                                                                             |                   |
| if( MaxNumSubblockMergeCand > 1 )                                                                                                        |                   |
| <b>merge_subblock_idx</b> [ x0 ][ y0 ]                                                                                                   | ae(v)             |
| } else {                                                                                                                                 |                   |
| if( sps_mh_intra_enabled_flag && cu_skip_flag[ x0 ][ y0 ] == 0 &&<br>( cbWidth * cbHeight ) >= 64 && cbWidth < 128 && cbHeight < 128 ) { |                   |
| <b>mh_intra_flag</b> [ x0 ][ y0 ]                                                                                                        | ae(v)             |
| if( mh_intra_flag[ x0 ][ y0 ] ) {                                                                                                        |                   |
| if ( cbWidth <= 2 * cbHeight    cbHeight <= 2 * cbWidth )                                                                                |                   |
| <b>mh_intra_luma_mpm_flag</b> [ x0 ][ y0 ]                                                                                               | ae(v)             |
| if( mh_intra_luma_mpm_flag[ x0 ][ y0 ] )                                                                                                 |                   |
| <b>mh_intra_luma_mpm_idx</b> [ x0 ][ y0 ]                                                                                                | ae(v)             |
| }                                                                                                                                        |                   |
| }                                                                                                                                        |                   |
| if( sps_triangle_enabled_flag && slice_type == B && cbWidth * cbHeight >= 16 )                                                           |                   |
| <b>merge_triangle_flag</b> [ x0 ][ y0 ]                                                                                                  | ae(v)             |
| if( merge_triangle_flag[ x0 ][ y0 ] )                                                                                                    |                   |
| <b>merge_triangle_idx</b> [ x0 ][ y0 ]                                                                                                   | ae(v)             |
| else if( MaxNumMergeCand > 1 )                                                                                                           |                   |
| <b>merge_idx</b> [ x0 ][ y0 ]                                                                                                            | ae(v)             |
| }                                                                                                                                        |                   |
| }                                                                                                                                        |                   |

In one embodiment, the MMVD information is signaled as:

|                                                 |       |
|-------------------------------------------------|-------|
| if( mmvd_flag[ x0 ][ y0 ] == 1 ) {              |       |
| <b>mmvd_merge_flag</b> [ x0 ][ y0 ]             | ae(v) |
| <b>amvr_mode</b> [ x0 ][ y0 ]                   | ae(v) |
| <b>mmvd_distance_idx_in_subset</b> [ x0 ][ y0 ] | ae(v) |
| <b>mmvd_direction_idx</b> [ x0 ][ y0 ]          | ae(v) |
| }                                               |       |

**mmvd\_distance\_idx\_in\_subset**[ x0 ][ y0 ] is binarized as a truncated unary code. The maximum value of the truncated unary code is 1 if **amvr\_mode**[ x0 ][ y0 ] < 2; Otherwise ( **amvr\_mode**[ x0 ][ y0 ] is equal to 2), The maximum value is set to be 3.

**mmvd\_distance\_idx**[x0][y0] is set equal to **mmvd\_distance\_idx\_in\_subset**[ x0 ][ y0 ]+2\***amvr\_mode**[ x0 ][ y0 ].

Which probability contexts are used by **mmvd\_distance\_idx\_in\_subset**[ x0 ][ y0 ] depends on **amvr\_mode**[ x0 ][ y0 ].

### 5.3 Embodiment #3 (MMVD slice level control)

In slice header, a syntax element **mmvd\_integer\_flag** is signaled.

The syntax change is described as follows, and the newly added parts are highlighted in *italics*

#### 7.3.2.1 Sequence parameter set RBSP syntax

| seq_parameter_set_rbsp( ) {                                       | Descriptor |
|-------------------------------------------------------------------|------------|
| ...                                                               | u(1)       |
| <b>sps_gbi_enabled_flag</b>                                       | u(1)       |
| <b>sps_mh_intra_enabled_flag</b>                                  | u(1)       |
| <b>sps_fracmmvd_enabled_flag</b>                                  | u(1)       |
| <b>sps_triangle_enabled_flag</b>                                  | u(1)       |
| <b>sps_ladf_enabled_flag</b>                                      | u(1)       |
| if ( <b>sps_ladf_enabled_flag</b> ) {                             |            |
| <b>sps_num_ladf_intervals_minus2</b>                              | u(2)       |
| <b>sps_ladf_lowest_interval_qp_offset</b>                         | se(v)      |
| for( i = 0; i < <b>sps_num_ladf_intervals_minus2</b> + 1; i++ ) { |            |
| <b>sps_ladf_qp_offset</b> [ i ]                                   | se(v)      |
| <b>sps_ladf_delta_threshold_minus1</b> [ i ]                      | ue(v)      |
| }                                                                 |            |
| }                                                                 |            |
| <b>rbsp_trailing_bits</b> ( )                                     |            |
| }                                                                 |            |

#### 7.3.3.1 General slice header syntax

| slice_header( ) {                          | Descriptor |
|--------------------------------------------|------------|
| ...                                        | ue(v)      |
| if ( <b>slice_type</b> != I ) {            |            |
| if( <b>sps_temporal_mvp_enabled_flag</b> ) |            |
| <b>slice_temporal_mvp_enabled_flag</b>     | u(1)       |
| if( <b>slice_type</b> == B )               |            |

|                                               |       |
|-----------------------------------------------|-------|
| <b>mvd_l1_zero_flag</b>                       | u(1)  |
| if( slice_temporal_mvp_enabled_flag ) {       |       |
| if( slice_type == B )                         |       |
| <b>collocated_from_l0_flag</b>                | u(1)  |
| }                                             |       |
| <b>six_minus_max_num_merge_cand</b>           | ue(v) |
| if( sps_affine_enable_flag )                  |       |
| <b>five_minus_max_num_subblock_merge_cand</b> | ue(v) |
| if( sps_fracmmvd_enable_flag )                |       |
| <b>slice_fracmmvd_flag</b>                    | u(1)  |
| }                                             |       |
| <b>slice_qp_delta</b>                         | se(v) |
| ...                                           |       |
| }                                             |       |

#### 7.4.3.1 Sequence parameter set RBSP semantics

**sps\_fracmmvd\_enabled\_flag** equal to 1 specifies that **slice\_fracmmvd\_flag** is present in the slice header syntax for B slices and P slices. **sps\_fracmmvd\_enabled\_flag** equal to 0 specifies that **slice\_fracmmvd\_flag** is not present in the slice header syntax for B slices and P slices.

#### 7.4.4.1 General slice header semantics

**slice\_fracmmvd\_flag** specifies the distance table used to derive  $MmvdDistance[x0][y0]$ .

When not present, the value of **slice\_fracmmvd\_flag** is inferred to be 1.

|                                                             |                   |
|-------------------------------------------------------------|-------------------|
| <b>coding_unit( x0, y0, cbWidth, cbHeight, treeType ) {</b> | <b>Descriptor</b> |
| if( slice_type != I ) {                                     |                   |
| <b>cu_skip_flag[ x0 ][ y0 ]</b>                             | ae(v)             |
| if( cu_skip_flag[ x0 ][ y0 ] == 0 )                         |                   |
| <b>pred_mode_flag</b>                                       | ae(v)             |
| }                                                           |                   |
| if( CuPredMode[ x0 ][ y0 ] == MODE_INTRA ) {                |                   |
| ...                                                         |                   |
| }                                                           |                   |
| } else { /* MODE_INTER */                                   |                   |
| if( cu_skip_flag[ x0 ][ y0 ] == 0 ) {                       |                   |
| <b>merge_flag[ x0 ][ y0 ]</b>                               | ae(v)             |
| if( merge_flag[ x0 ][ y0 ] ) {                              |                   |
| merge_data( x0, y0, cbWidth, cbHeight )                     |                   |
| } else {                                                    |                   |
| <b>mmvd_flag[ x0 ][ y0 ]</b>                                | ae(v)             |
| if( mmvd_flag[ x0 ][ y0 ] == 1 ) {                          |                   |
| <b>mmvd_merge_flag[ x0 ][ y0 ]</b>                          | ae(v)             |
| <b>mmvd_distance_subset_idx[ x0 ][ y0 ]</b>                 | ae(v)             |
| <b>mmvd_distance_idx_in_subset[ x0 ][ y0 ]</b>              | ae(v)             |

|                                                                    |              |
|--------------------------------------------------------------------|--------------|
| <b>mmvd_direction_idx[ x0 ][ y0 ]</b>                              | <b>ae(v)</b> |
| } else {                                                           |              |
| if( slice_type == B )                                              |              |
| <b>inter_pred_idc[ x0 ][ y0 ]</b>                                  | <b>ae(v)</b> |
| if( sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16 ) { |              |
| <b>inter_affine_flag[ x0 ][ y0 ]</b>                               | <b>ae(v)</b> |
| if( sps_affine_type_flag && inter_affine_flag[ x0 ][ y0 ] )        |              |
| <b>cu_affine_type_flag[ x0 ][ y0 ]</b>                             | <b>ae(v)</b> |
| }                                                                  |              |
| ...                                                                |              |
| }                                                                  |              |
| }                                                                  |              |

|                                                                                                                                          |                         |
|------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| merge_data( x0, y0, cbWidth, cbHeight ) {                                                                                                | <b>Descript<br/>or</b>  |
| <del>    <b>mmvd_flag[ x0 ][ y0 ]</b></del>                                                                                              | <del><b>ae(v)</b></del> |
| <del>    if( mmvd_flag[ x0 ][ y0 ] == 1 ) {</del>                                                                                        |                         |
| <del>        <b>mmvd_merge_flag[ x0 ][ y0 ]</b></del>                                                                                    | <del><b>ae(v)</b></del> |
| <del>        <b>mmvd_distance_idx[ x0 ][ y0 ]</b></del>                                                                                  | <del><b>ae(v)</b></del> |
| <del>        <b>mmvd_direction_idx[ x0 ][ y0 ]</b></del>                                                                                 | <del><b>ae(v)</b></del> |
| <del>    } else {</del>                                                                                                                  |                         |
| if( MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8 )                                                                       |                         |
| <b>merge_subblock_flag[ x0 ][ y0 ]</b>                                                                                                   | <b>ae(v)</b>            |
| if( merge_subblock_flag[ x0 ][ y0 ] == 1 ) {                                                                                             |                         |
| if( MaxNumSubblockMergeCand > 1 )                                                                                                        |                         |
| <b>merge_subblock_idx[ x0 ][ y0 ]</b>                                                                                                    | <b>ae(v)</b>            |
| } else {                                                                                                                                 |                         |
| if( sps_mh_intra_enabled_flag && cu_skip_flag[ x0 ][ y0 ] == 0 &&<br>( cbWidth * cbHeight ) >= 64 && cbWidth < 128 && cbHeight < 128 ) { |                         |
| <b>mh_intra_flag[ x0 ][ y0 ]</b>                                                                                                         | <b>ae(v)</b>            |
| if( mh_intra_flag[ x0 ][ y0 ] ) {                                                                                                        |                         |
| if ( cbWidth <= 2 * cbHeight    cbHeight <= 2 * cbWidth )                                                                                |                         |
| <b>mh_intra_luma_mpm_flag[ x0 ][ y0 ]</b>                                                                                                | <b>ae(v)</b>            |
| if( mh_intra_luma_mpm_flag[ x0 ][ y0 ] )                                                                                                 |                         |
| <b>mh_intra_luma_mpm_idx[ x0 ][ y0 ]</b>                                                                                                 | <b>ae(v)</b>            |
| }                                                                                                                                        |                         |
| }                                                                                                                                        |                         |
| }                                                                                                                                        |                         |
| if( sps_triangle_enabled_flag && slice_type == B && cbWidth * cbHeight >= 16 )                                                           |                         |
| <b>merge_triangle_flag[ x0 ][ y0 ]</b>                                                                                                   | <b>ae(v)</b>            |
| if( merge_triangle_flag[ x0 ][ y0 ] )                                                                                                    |                         |
| <b>merge_triangle_idx[ x0 ][ y0 ]</b>                                                                                                    | <b>ae(v)</b>            |
| else if( MaxNumMergeCand > 1 )                                                                                                           |                         |

|                              |              |
|------------------------------|--------------|
| <b>merge_idx[ x0 ][ y0 ]</b> | <b>ae(v)</b> |
| }                            |              |
| }                            |              |

In one embodiment, the MMVD information is signaled as:

|                                                |              |
|------------------------------------------------|--------------|
| if( mmvd_flag[ x0 ][ y0 ] = 1 ) {              |              |
| <b>mmvd_merge_flag[ x0 ][ y0 ]</b>             | <b>ae(v)</b> |
| <b>amvr_mode[ x0 ][ y0 ]</b>                   | <b>ae(v)</b> |
| <b>mmvd_distance_idx_in_subset[ x0 ][ y0 ]</b> | <b>ae(v)</b> |
| <b>mmvd_direction_idx[ x0 ][ y0 ]</b>          | <b>ae(v)</b> |
| }                                              |              |

**mmvd\_distance\_idx\_in\_subset[ x0 ][ y0 ]** is binarized as a truncated unary code. The maximum value of the truncated unary code is 1 if **amvr\_mode[ x0 ][ y0 ]** < 2; Otherwise ( **amvr\_mode[ x0 ][ y0 ]** is equal to 2), The maximum value is set to be 3.

**mmvd\_distance\_idx[x0][y0]** is set equal to **mmvd\_distance\_idx\_in\_subset[ x0 ][ y0 ]+2\*amvr\_mode[ x0 ][ y0 ]**. In one example, the probability contexts are used by **mmvd\_distance\_idx\_in\_subset[ x0 ][ y0 ]** depends on **amvr\_mode[ x0 ][ y0 ]**.

The array indices x0, y0 specify the location ( x0, y0 ) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture. The **mmvd\_distance\_idx[x0][y0]** and the **MmvdDistance[ x0 ][ y0 ]** is as follows:

**Table 7-9 – Specification of MmvdDistance[ x0 ][ y0 ] based on mmvd\_distance\_idx[ x0 ][ y0 ] when slice\_fracmmvd\_flag is equal to 1.**

| <b>mmvd_distance_idx[ x0 ][ y0 ]</b> | <b>MmvdDistance[ x0 ][ y0 ]</b> |
|--------------------------------------|---------------------------------|
| 0                                    | 1                               |
| 1                                    | 2                               |
| 2                                    | 4                               |
| 3                                    | 8                               |
| 4                                    | 16                              |
| 5                                    | 32                              |
| 6                                    | 64                              |
| 7                                    | 128                             |

**Table 7-9 – Specification of MmvdDistance[ x0 ][ y0 ] based on mmvd\_distance\_idx[ x0 ][ y0 ] when slice\_fracmmvd\_flag is equal to 0.**

| <i>mmvd_distance_idx[ x0 ][ y0 ]</i> | <i>MmvdDistance[ x0 ][ y0 ]</i> |
|--------------------------------------|---------------------------------|
| <b>0</b>                             | <b>4</b>                        |
| <b>1</b>                             | <b>8</b>                        |
| <b>2</b>                             | <b>16</b>                       |
| <b>3</b>                             | <b>32</b>                       |
| <b>4</b>                             | <b>64</b>                       |
| <b>5</b>                             | <b>128</b>                      |
| <b>6</b>                             | <b>256</b>                      |
| <b>7</b>                             | <b>512</b>                      |

When *mmvd\_integer\_flag* is equal to 1, *mmvd\_distance* = *mmvd\_distance*<<2.

**[00182]** FIG. 10 is a flowchart for an example method 1000 for video processing. The method 1000 includes determining(1002) at least one syntax element indicating a distance index(DI) for a current video block, wherein the at least one syntax element is not binarized with a truncated unary code; and performing(1004) a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element.

**[00183]** FIG. 11 is a flowchart for an example method 1100 for video processing. The method 1100 includes determining(1102) at least one syntax element indicating a motion vector difference(MVD)precision for a current video block, wherein the MVD precision indicates a distance between a motion vector of the current video block and a base candidate selected from a merge candidate list; and performing(1104) a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element.

**[00184]** With reference to the methods 1000, 1100, some examples of motion vector signaling are described in Section 4 of this document, and the aforementioned methods may include the features and steps described below.

**[00185]** In one aspect, there is disclosed a method for video processing, comprising:

determining at least one syntax element indicating a motion vector difference(MVD)precision for a current video block, wherein the MVD precision indicates a distance between a motion vector of the current video block and a base candidate selected from a merge candidate list;

performing a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element;

wherein MVD precisions allowed for the current video block are classified as entries into a plurality of subsets each is assigned with a subset index, and the at least one syntax element

comprises a first syntax element indicating the subset index and a second syntax element indicating a distance index(DI) which indicates an entry in a subset with the subset index.

**[00186]** In one example, the first syntax element comprises mmvd\_distance\_subset\_idx and the second syntax element comprises mmvd\_distance\_idx\_in\_subset.

**[00187]** In one example, the first syntax element is signaled before the second syntax element.

**[00188]** In one example, at least one of the first and second syntax elements is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

**[00189]** In one example, the second syntax element is binarized with the truncated unary code with a maximum value equal to an amount of entries in the subset minus 1.

**[00190]** In one example, the second syntax element is binarized as a flag if there are only two entries in the subset.

**[00191]** In one example, the second syntax element is not signaled if there is only one entry in the subset.

**[00192]** In one example, an amount of the plurality of subsets is two, and a first of two subsets comprises all entries with fractional precisions, and a second of the two subsets comprises all entries with integer precisions.

**[00193]** In one example, an amount of the plurality of subsets is two; and a first of two subsets comprises one entry, and a second of the two subsets comprises remaining entries.

**[00194]** In one example, an amount of the plurality of subsets is three, and a first of three subsets comprises all entries with fractional precisions, a second of the three subsets comprises all entries with integer precisions less than 4-pel, and a third of the three subsets comprises all entries with integer precisions not less than 4-pel.

**[00195]** In one example, the current video block is coded with a locally adaptive motion vector resolution (LAMVR) mode, and an amount of the plurality of subsets is equal to an amount of allowed MVD precisions in the LAMVR mode.

**[00196]** In one example, the first syntax element is signaled by reusing a same way as that is used for signaling the LAMVR mode.

**[00197]** In one example, the way for signaling the LAMVR mode comprises signaling at least one of context offset index .

**[00198]** In one example, the second syntax element depends on at least one associated LAMVR syntax element.

**[00199]** In one example, the at least one associated LAMVR syntax element comprises a syntax element `amvr_mode`.

**[00200]** In one example, at least one of an amount of the plurality of subsets and/or a definition of each of the plurality of subsets is predefined or adaptively changed.

**[00201]** In one example, at least one of the first syntax element and the second syntax element is not signaled explicitly.

**[00202]** In one example, at least one of the first syntax element and the second syntax element is determined based on one or more characteristics of at least one of the current video block and previously coded blocks.

**[00203]** In one example, the one or more characteristics comprises a coding mode, a height or a width of at least one of the current video block and previously coded blocks.

**[00204]** In one example, the second syntax element comprises a first resolution bin, and the first resolution bin indicates whether the distance index (DI) is smaller than T1 or indicates whether the DI is greater than T2, T1 and T2 being a predetermined integer.

**[00205]** In one example, the second syntax element comprises a first resolution bin, wherein the first resolution bin indicates whether an entry in the subset is smaller than a first threshold or greater than a second threshold.

**[00206]** In one example, the second syntax element further comprises a first code after the first resolution bin to indicate a value of the DI.

**[00207]** In one example, the first syntax element comprises `mmvd_distance_subset_idx`, the second syntax element comprises one of `mmvd_short_distance_idx_in_subset` and `mmvd_long_distance_idx_in_subset` which is the first code and comprises `mmvd_resolution_flag` which is the first resolution bin, wherein `mmvd_long_distance_idx_in_subset` or `mmvd_short_distance_idx_in_subset` is signaled after the `mmvd_resolution_flag`.

**[00208]** In one example, the first syntax element comprises `mmvd_distance_subset_idx`, the second syntax element comprises `mmvd_distance_idx_in_subset` which is the first code and `mmvd_resolution_flag` which is the first resolution bin, wherein the `mmvd_distance_idx_in_subset` is signaled after the `mmvd_resolution_flag`.

**[00209]** In one example, whether `mmvd_short_distance_idx_in_subset` or `mmvd_long_distance_idx_in_subset` is signaled after the first syntax element `mmvd_distance_subset_idx` depends on a value of the `mmvd_resolution_flag`.

**[00210]** In one example,  $T1$  is equal to 2 and corresponds to a 1-pel precision distance; or  
 $T1$  is equal to 1 and corresponds to a 1/2-pel precision distance; or  
 $T1$  is equal to 3 and corresponds to a 2-pel precision distance.

**[00211]** In one example,  $T2$  is equal to 2 and corresponds to a 1-pel precision distance; or  
 $T2$  is equal to 1 and corresponds to a 1/2-pel precision distance; or  
 $T2$  is equal to 3 and corresponds to a 2-pel precision distance.

**[00212]** In one example, if the distance index(DI) is smaller than  $T1$ , the first code is a code for short distance index signaled after the first resolution bin to indicates a value of the DI.

**[00213]** In one example, the code for short distance index corresponds to the distance index or a difference between the distance index DI and a third threshold.

**[00214]** In one example, the distance index or difference is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

**[00215]** In one example, the distance index is binarized with a truncated unary code with a maximum value equal to  $T1-1$ .

**[00216]** In one example, the third threshold is  $T1-1$ , and the difference is equal to  $T1-1-DI$  and the difference is binarized with the truncated unary code with a maximum value equal to  $T1-1$ .

**[00217]** In one example, if the distance index(DI) is not smaller than  $T1$ , the first code is a code for long distance index signaled after the first resolution bin to indicate a value of the DI.

**[00218]** In one example, the code for long distance index corresponds to a difference between the distance index DI and a fourth threshold, and the difference is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

**[00219]** In one example, the fourth threshold is  $T1$ , and the difference is equal to  $DI-T1$ ; or  
the fourth threshold is  $D_{max}$ , and the difference is equal to  $D_{max}-DI$ , wherein  $D_{max}$  represents a maximum allowed distance index.

**[00220]** In one example, the difference is binarized with the truncated unary code with a maximum value equal to  $D_{max}-T1$ .

**[00221]** In one example, if the distance index(DI) is not greater than  $T2$ , the first code is a code for short distance index is signaled after the first resolution bin to indicate a value of the DI.

**[00222]** In one example, the code for short distance index corresponds to the distance index or a difference between the distance index DI and a fifth threshold.

**[00223]** In one example, the distance index or difference is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

**[00224]** In one example, the distance index is binarized with a truncated unary code with a maximum value equal to  $T_2$ .

**[00225]** In one example, the fifth threshold is  $T_2$ , and the difference is equal to  $T_2 - DI$  and the difference is binarized with the truncated unary code with a maximum value equal to  $T_2$ .

**[00226]** In one example, if the distance index ( $DI$ ) is greater than  $T_2$ , the first code is a code for long distance index signaled after the first resolution bin to indicate a value of  $DI$ .

**[00227]** In one example, the code for long distance index corresponds to a difference between the distance index  $DI$  and a sixth threshold, and the difference is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

**[00228]** In one example, the sixth threshold is  $T_2 + 1$ , and the difference is equal to  $DI - T_2 - 1$ ; or

the sixth threshold is  $D_{max}$ , and the difference is equal to  $D_{max} - DI$ , wherein  $D_{max}$  represents a maximum allowed distance index.

**[00229]** In one example, the difference is binarized with the truncated unary code with a maximum value equal to  $D_{max} - 1 - T_2$ .

**[00230]** In one example, the distance index is binarized as follows:

| Distance<br>IDX | 0   | 1    | 2    | 3   | 4    | 5     | 6      | 7      |
|-----------------|-----|------|------|-----|------|-------|--------|--------|
| Bins            | 01  | 00   | 10   | 110 | 1110 | 11110 | 111110 | 111111 |
| Bins            | 01  | 00   | 11   | 101 | 1001 | 10001 | 100001 | 100000 |
| Bins            | 00  | 01   | 10   | 110 | 1110 | 11110 | 111110 | 111111 |
| Bins            | 00  | 01   | 11   | 101 | 1001 | 10001 | 100001 | 100000 |
| Bins            | 11  | 10   | 00   | 010 | 0110 | 01110 | 011110 | 011111 |
| Bins            | 11  | 10   | 01   | 001 | 0001 | 00001 | 000001 | 000000 |
| Bins            | 10  | 11   | 00   | 010 | 0110 | 01110 | 011110 | 011111 |
| Bins            | 10  | 11   | 01   | 001 | 0001 | 00001 | 000001 | 000000 |
| Bins            | 010 | 0110 | 0111 | 10  | 110  | 1110  | 11110  | 11111  |

| Bins | 0111 | 0110 | 010 | 10 | 110 | 1110 | 11110 | 11111 |
|------|------|------|-----|----|-----|------|-------|-------|
|------|------|------|-----|----|-----|------|-------|-------|

**[00231]** In one example, at least one of the first syntax element and the second syntax element is coded using one or more probability contexts.

**[00232]** In one example, the first resolution bin of the current video block is coded using one or more probability contexts, wherein the one or more probability contexts are determined based on one or more first resolution bins of neighboring blocks.

**[00233]** In one example, the one or more probability contexts are determined based on one or more locally adaptive motion vector resolution (LAMVR) values of neighboring blocks.

**[00234]** In one example, the one or more LAMVR values are a value of AMVR\_mode.

**[00235]** In one example, a first bin to code the first code is coded with one or more probability contexts, and remaining bins are bypass coded.

**[00236]** In one example, first N bins to code the first code are coded with one or more probability contexts, and remaining bins are bypass coded.

**[00237]** In one example all bins to code the first code are coded with one or more probability contexts.

**[00238]** In one example, different bins to code the first code are coded with different probability contexts.

**[00239]** In one example, a part of bins to code the first code are coded with a single probability context.

**[00240]** In one example, the part of bins are consecutive bins.

**[00241]** In one example, the one or more probability contexts are determined based on one or more first codes of neighboring blocks of the current video block.

**[00242]** In one aspect, there is disclosed a method for video processing , comprising:

determining at least one syntax element indicating a distance index(DI) for a current video block, wherein the at least one syntax element is not binarized with a truncated unary code; and

performing a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element.

**[00243]** In one example, the at least one syntax element is binarized with one of a fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

**[00244]** In one example, the current video block is coded with a Merge with Motion Vector Difference (MMVD) mode.

**[00245]** In an example, the conversion includes encoding the current video block into the bitstream representation of the current video block and decoding the current video block from the bitstream representation of the current video block.

**[00246]** In an aspect, there is disclosed an apparatus in a video system, the apparatus comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of examples described as above.

**[00247]** In an aspect, there is disclosed a computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of examples described as above.

**[00248]** FIG. 12 is a block diagram of a video processing apparatus 1200. The apparatus 1200 may be used to implement one or more of the methods described herein. The apparatus 1200 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1200 may include one or more processors 1202, one or more memories 1204 and video processing hardware 1206. The processor(s) 1202 may be configured to implement one or more methods described in the present document. The memory (memories) 1204 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1206 may be used to implement, in hardware circuitry, some techniques described in the present document, and may be partly or completely be a part of the processors 1202 (e.g., graphics processor core GPU or other signal processing circuitry).

**[00249]** In the present document, the term “video processing” may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream.

**[00250]** It will be appreciated that several techniques have been disclosed that will benefit video encoder and decoder embodiments incorporated within video processing devices such as

smartphones, laptops, desktops, and similar devices by allowing the use of virtual motion candidates that are constructed based on various rules disclosed in the present document.

**[00251]** The disclosed and other solutions, examples, embodiments, modules and the functional operations described in this document can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this document and their structural equivalents, or in combinations of one or more of them. The disclosed and other embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them. The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

**[00252]** A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

**[00253]** The processes and logic flows described in this document can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

**[00254]** Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

**[00255]** While this patent document contains many specifics, these should not be construed as limitations on the scope of any subject matter or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular techniques. Certain features that are described in this patent document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

**[00256]** Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown

or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described in this patent document should not be understood as requiring such separation in all embodiments.

**[00257]** Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this patent document.

## CLAIMS

1. A method for video processing, comprising:

determining at least one syntax element indicating a motion vector difference (MVD) precision for a current video block, wherein the MVD precision indicates a distance between a motion vector of the current video block and a base candidate selected from a merge candidate list;

performing a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element;

wherein MVD precisions allowed for the current video block are classified as entries into a plurality of subsets each is assigned with a subset index, and the at least one syntax element comprises a first syntax element indicating the subset index and a second syntax element indicating a distance index(DI) which indicates an entry in a subset with the subset index.

2. The method of claim 1, wherein the first syntax element comprises mmvd\_distance\_subset\_idx and the second syntax element comprises mmvd\_distance\_idx\_in\_subset.

3. The method of claim 1 or 2, wherein the first syntax element is signaled before the second syntax element.

4. The method of any one of claims 1-3, wherein at least one of the first and second syntax elements is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

5. The method of claim 4, wherein the second syntax element is binarized with the truncated unary code with a maximum value equal to an amount of entries in the subset minus 1.

6. The method of any one of claims 1-3, wherein the second syntax element is binarized as a flag if there are only two entries in the subset.

7. The method of any one of claims 1-2, wherein the second syntax element is not signaled if there is only one entry in the subset.
8. The method of any one of claims 1-7, wherein an amount of the plurality of subsets is two, and a first of two subsets comprises all entries with fractional precisions, and a second of the two subsets comprises all entries with integer precisions.
9. The method of any one of claims 1-7, wherein an amount of the plurality of subsets is two; and a first of two subsets comprises one entry, and a second of the two subsets comprises remaining entries.
10. The method of any one of claims 1-7, wherein an amount of the plurality of subsets is three, and a first of three subsets comprises all entries with fractional precisions, a second of the three subsets comprises all entries with integer precisions less than 4-pel, and a third of the three subsets comprises all entries with integer precisions not less than 4-pel.
11. The method of any one of claims 1-7, wherein the current video block is coded with a locally adaptive motion vector resolution (LAMVR) mode, and an amount of the plurality of subsets is equal to an amount of allowed MVD precisions in the LAMVR mode.
12. The method of claim 10, wherein the first syntax element is signaled by reusing a same way as that is used for signaling the LAMVR mode.
13. The method of claim 12, wherein the way for signaling the LAMVR mode comprises signaling at least one context offset index.
14. The method of any one of claims 11-13, wherein the second syntax element depends on at least one associated LAMVR syntax element.
15. The method of claim 14, wherein the at least one associated LAMVR syntax element comprises a syntax element `amvr_mode`.

16. The method of any one of claims 1-15, wherein at least one of an amount of the plurality of subsets and/or a definition of each of the plurality of subsets is predefined or adaptively changed.

17. The method of any one of claims 1-2, wherein at least one of the first syntax element and the second syntax element is not signaled explicitly.

18. The method of claim 17, wherein at least one of the first syntax element and the second syntax element is determined based on one or more characteristics of at least one of the current video block and previously coded blocks.

19. The method of claim 18, wherein the one or more characteristics comprises a coding mode, a height or a width of at least one of the current video block and previously coded blocks.

20. The method of claim 1, wherein the second syntax element comprises a first resolution bin, and the first resolution bin indicates whether the distance index (DI) is smaller than T1 or indicates whether the DI is greater than T2, T1 and T2 being a predetermined integer.

21. The method of claim 1, wherein the second syntax element comprises a first resolution bin, wherein the first resolution bin indicates whether an entry in the subset is smaller than a first threshold or greater than a second threshold.

22. The method of claim 20 or 21, wherein the second syntax element further comprises a first code after the first resolution bin to indicate a value of the DI.

23. The method of claim 22, wherein the first syntax element comprises `mmvd_distance_subset_idx`, the second syntax element comprises one of `mmvd_short_distance_idx_in_subset` and `mmvd_long_distance_idx_in_subset` which is the first code and comprises `mmvd_resolution_flag` which is the first resolution bin, wherein `mmvd_long_distance_idx_in_subset` or `mmvd_short_distance_idx_in_subset` is signaled after the `mmvd_resolution_flag`.

24. The method of claim 22, wherein the first syntax element comprises `mmvd_distance_subset_idx`, the second syntax element comprises `mmvd_distance_idx_in_subset` which is the first code and `mmvd_resolution_flag` which is the first resolution bin, wherein the `mmvd_distance_idx_in_subset` is signaled after the `mmvd_resolution_flag`.
25. The method of claim 23, wherein whether `mmvd_short_distance_idx_in_subset` or `mmvd_long_distance_idx_in_subset` is signaled after the first syntax element `mmvd_distance_subset_idx` depends on a value of the `mmvd_resolution_flag`.
26. The method of claim 20, wherein
- T1 is equal to 2 and corresponds to a 1-pel precision distance; or
  - T1 is equal to 1 and corresponds to a 1/2-pel precision distance; or
  - T1 is equal to 3 and corresponds to a 2-pel precision distance.
27. The method of claim 20, wherein
- T2 is equal to 2 and corresponds to a 1-pel precision distance; or
  - T2 is equal to 1 and corresponds to a 1/2-pel precision distance; or
  - T2 is equal to 3 and corresponds to a 2-pel precision distance.
28. The method of claim 22, wherein if the distance index (DI) is smaller than T1, the first code is a code for short distance index signaled after the first resolution bin to indicate a value of the DI.
29. The method of claim 28, wherein the code for short distance index corresponds to the distance index or a difference between the distance index DI and a third threshold.
30. The method of claim 29, wherein the distance index or difference is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

31. The method of claim 30, wherein the distance index is binarized with a truncated unary code with a maximum value equal to  $T1-1$ .

32. The method of claim 30, wherein the third threshold is  $T1-1$ , and the difference is equal to  $T1-1-DI$  and the difference is binarized with the truncated unary code with a maximum value equal to  $T1-1$ .

33. The method of claim 22 wherein if the distance index (DI) is not smaller than  $T1$ , the first code is a code for long distance index signaled after the first resolution bin to indicate a value of the DI.

34. The method of claim 33, wherein the code for long distance index corresponds to a difference between the distance index DI and a fourth threshold, and the difference is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

35. The method of claim 34, wherein

the fourth threshold is  $T1$ , and the difference is equal to  $DI-T1$ ; or

the fourth threshold is  $Dmax$ , and the difference is equal to  $Dmax-DI$ , wherein  $Dmax$  represents a maximum allowed distance index.

36. The method of claim 35, wherein the difference is binarized with the truncated unary code with a maximum value equal to  $Dmax-T1$ .

37. The method of claim 22, wherein if the distance index (DI) is not greater than  $T2$ , the first code is a code for short distance index is signaled after the first resolution bin to indicate a value of the DI.

38. The method of claim 37, wherein the code for short distance index corresponds to the distance index or a difference between the distance index DI and a fifth threshold.

39. The method of claim 38, wherein the distance index or difference is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

40. The method of claim 39, wherein the distance index is binarized with a truncated unary code with a maximum value equal to  $T_2$ .

41. The method of claim 39, wherein the fifth threshold is  $T_2$ , and the difference is equal to  $T_2 - DI$  and the difference is binarized with the truncated unary code with a maximum value equal to  $T_2$ .

42. The method of claim 22, wherein if the distance index( $DI$ ) is greater than  $T_2$ , the first code is a code for long distance index signaled after the first resolution bin to indicate a value of  $DI$ .

43. The method of claim 42, wherein the code for long distance index corresponds to a difference between the distance index  $DI$  and a sixth threshold, and the difference is binarized with one of a unary code, a truncated unary code, fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

44. The method of claim 43, wherein

the sixth threshold is  $T_2 + 1$ , and the difference is equal to  $DI - T_2 - 1$ ; or

the sixth threshold is  $D_{max}$ , and the difference is equal to  $D_{max} - DI$ , wherein  $D_{max}$  represents a maximum allowed distance index.

45. The method of claim 44, wherein the difference is binarized with the truncated unary code with a maximum value equal to  $D_{max} - 1 - T_2$ .

46. The method of claim 1, wherein the distance index is binarized as follows:

| Distance<br>IDX | 0  | 1  | 2  | 3   | 4    | 5     | 6      | 7      |
|-----------------|----|----|----|-----|------|-------|--------|--------|
| Bins            | 01 | 00 | 10 | 110 | 1110 | 11110 | 111110 | 111111 |
| Bins            | 01 | 00 | 11 | 101 | 1001 | 10001 | 100001 | 100000 |

|             |      |      |      |     |      |       |        |        |
|-------------|------|------|------|-----|------|-------|--------|--------|
| <b>Bins</b> | 00   | 01   | 10   | 110 | 1110 | 11110 | 111110 | 111111 |
| <b>Bins</b> | 00   | 01   | 11   | 101 | 1001 | 10001 | 100001 | 100000 |
| <b>Bins</b> | 11   | 10   | 00   | 010 | 0110 | 01110 | 011110 | 011111 |
| <b>Bins</b> | 11   | 10   | 01   | 001 | 0001 | 00001 | 000001 | 000000 |
| <b>Bins</b> | 10   | 11   | 00   | 010 | 0110 | 01110 | 011110 | 011111 |
| <b>Bins</b> | 10   | 11   | 01   | 001 | 0001 | 00001 | 000001 | 000000 |
| <b>Bins</b> | 010  | 0110 | 0111 | 10  | 110  | 1110  | 11110  | 11111  |
| <b>Bins</b> | 0111 | 0110 | 010  | 10  | 110  | 1110  | 11110  | 11111  |

47. The method of claims 1-46, wherein at least one of the first syntax element and the second syntax element is coded using one or more probability contexts.

48. The method of claim 47, wherein the first resolution bin of the current video block is coded using one or more probability contexts, wherein the one or more probability contexts are determined based on one or more first resolution bins of neighboring blocks.

49. The method of claim 47 or 48, wherein the one or more probability contexts are determined based on one or more locally adaptive motion vector resolution (LAMVR) values of neighboring blocks.

50. The method of claim 49, wherein the one or more LAMVR values are a value of AMVR\_mode.

51. The method of claim 47, wherein a first bin to code the first code is coded with one or more probability contexts, and remaining bins are bypass coded.

52. The method of claim 47, wherein first N bins to code the first code are coded with one or more probability contexts, and remaining bins are bypass coded.

53. The method of claim 47, wherein all bins to code the first code are coded with one or more probability contexts.

54. The method of claim 47, wherein different bins to code the first code are coded with different probability contexts.

55. The method of claim 47, wherein a part of bins to code the first code are coded with a single probability context.

56. The method of claim 55, wherein the part of bins are consecutive bins.

57. The method of any one of claims 52-56, wherein the one or more probability contexts are determined based on one or more first codes of neighboring blocks of the current video block.

58. A method for video processing, comprising:

determining at least one syntax element indicating a distance index(DI) for a current video block, wherein the at least one syntax element is not binarized with a truncated unary code; and

performing a conversion between a current video block and a bitstream representation of the current video block based on the at least one syntax element.

59. The method of claim 58, wherein the at least one syntax element is binarized with one of a fixed-length code, an Exponential-Golomb code, a truncated Exponential-Golomb code and a Rice code.

60. The method of claim 58 or 59, wherein the current video block is coded with a Merge with Motion Vector Difference (MMVD) mode.

61. The method of any of claims 1 to 60, wherein the conversion includes encoding the current video block into the bitstream representation of the current video block and decoding the current video block from the bitstream representation of the current video block.

62. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of claims 1 to 61.

63. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of claims 1 to 61.

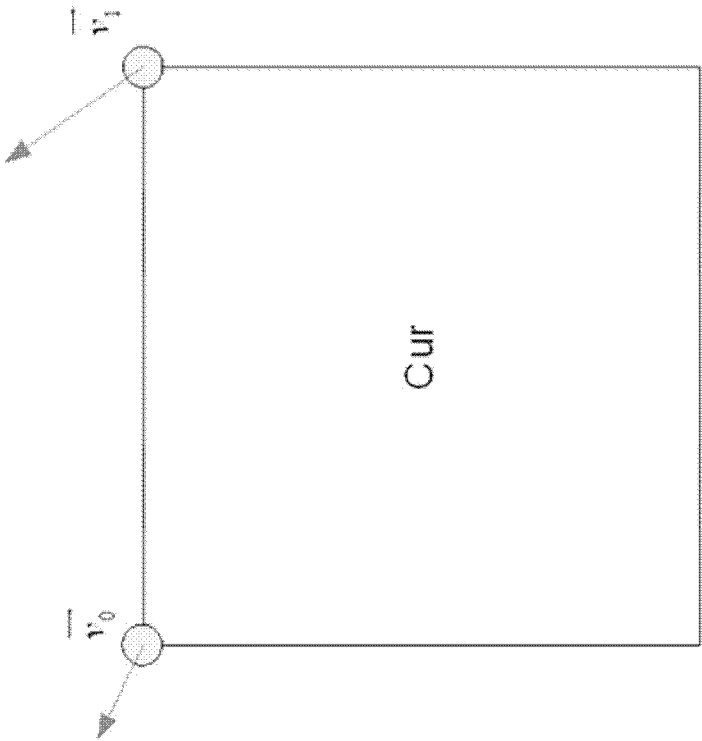


FIG. 1

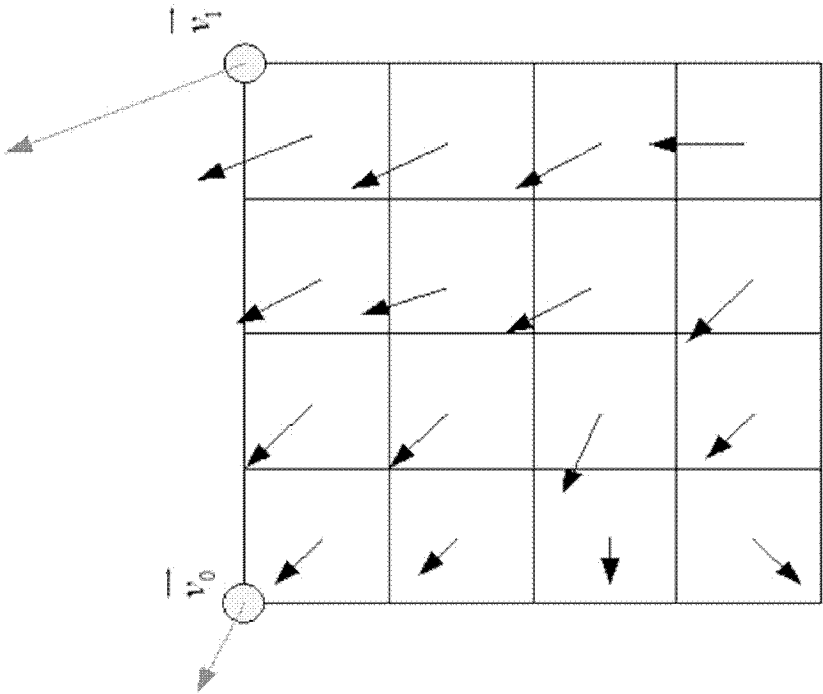


FIG. 2

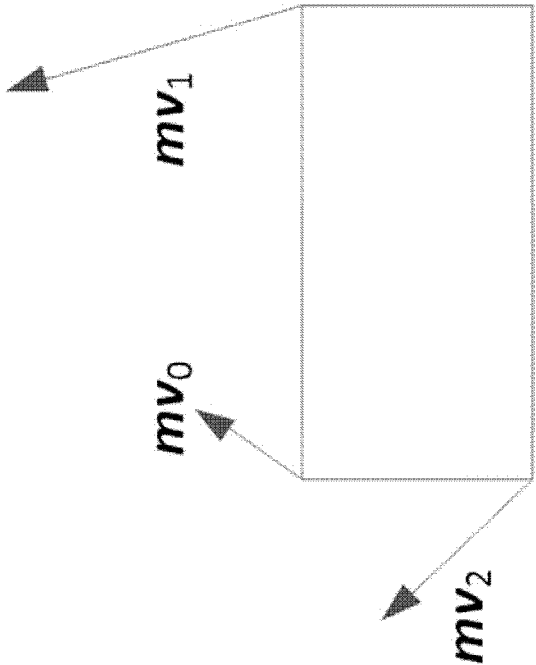


FIG. 3B

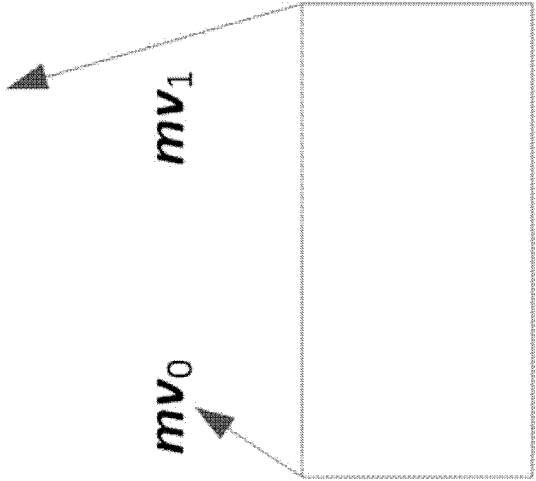


FIG. 3A

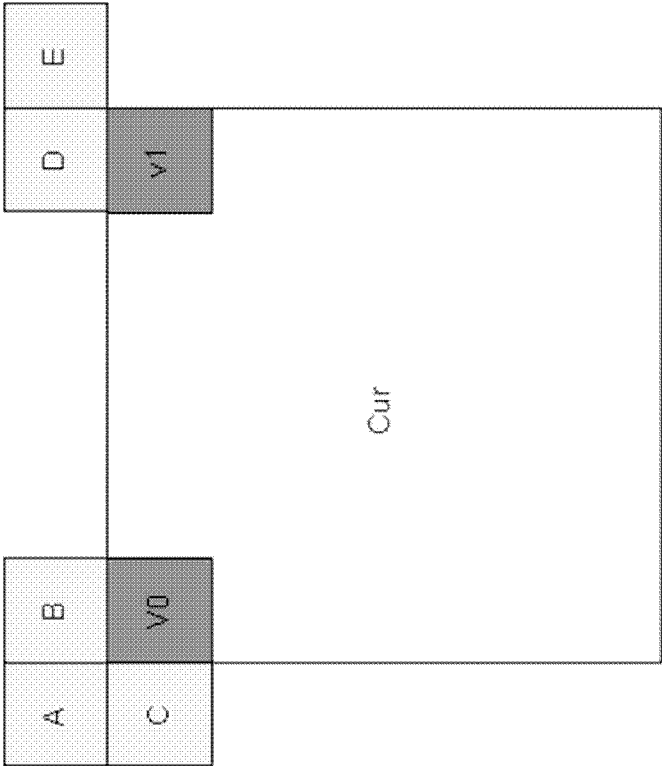


FIG. 4

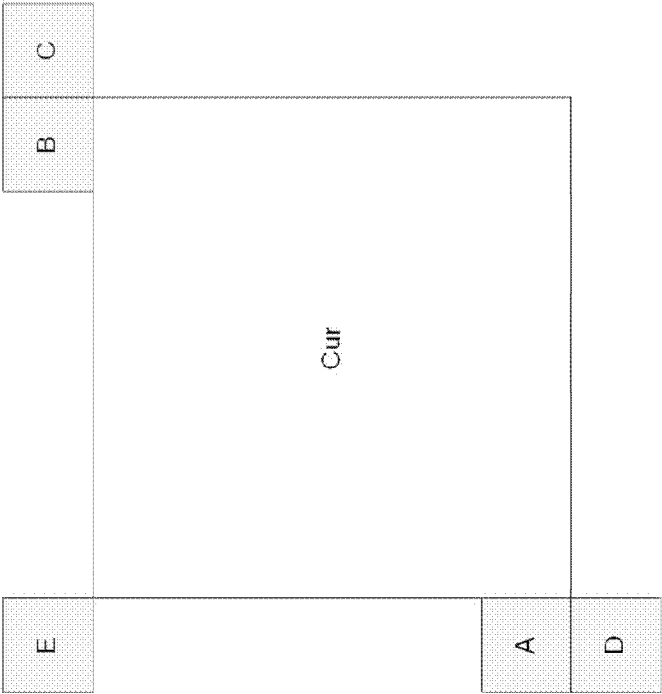


FIG. 5A

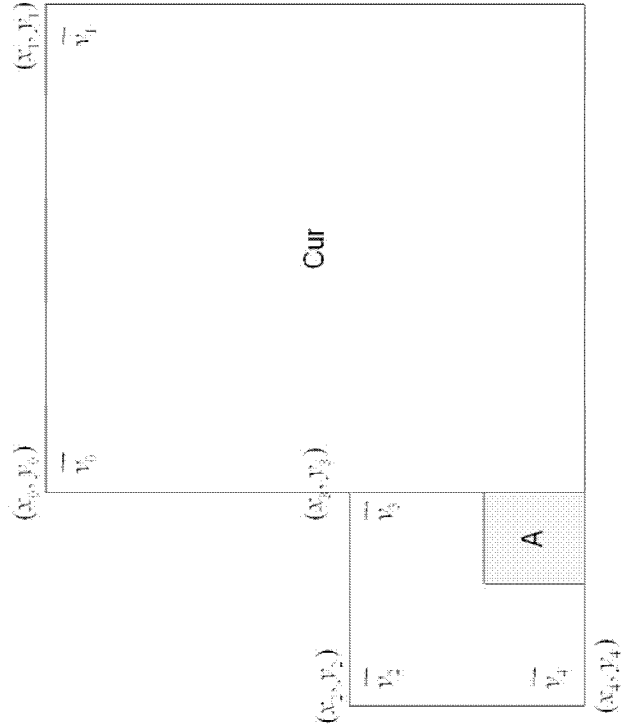


FIG. 5B

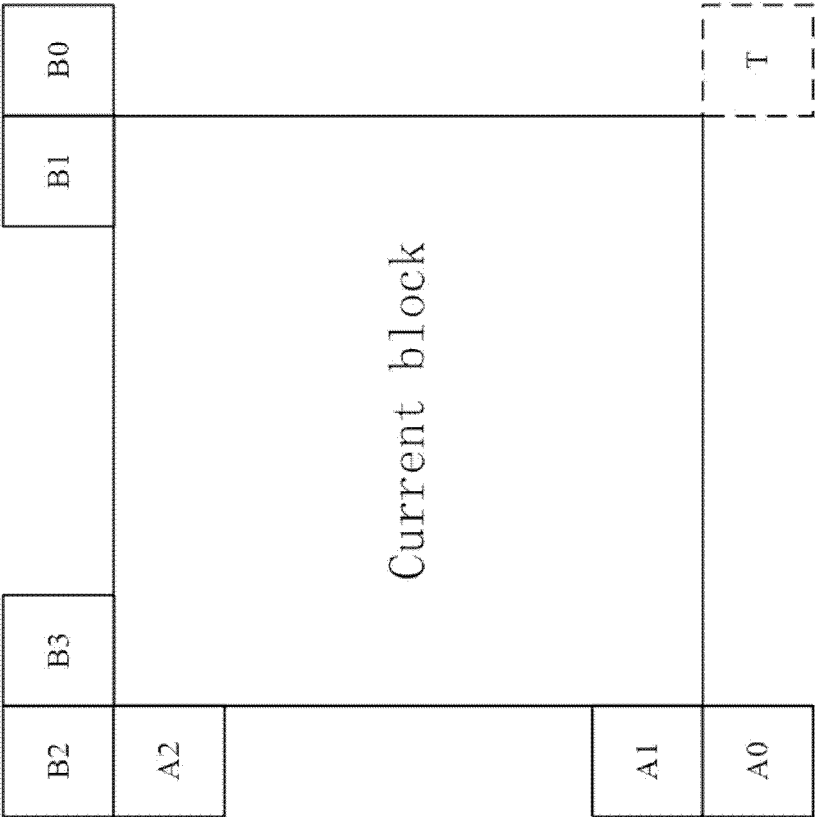


FIG. 6

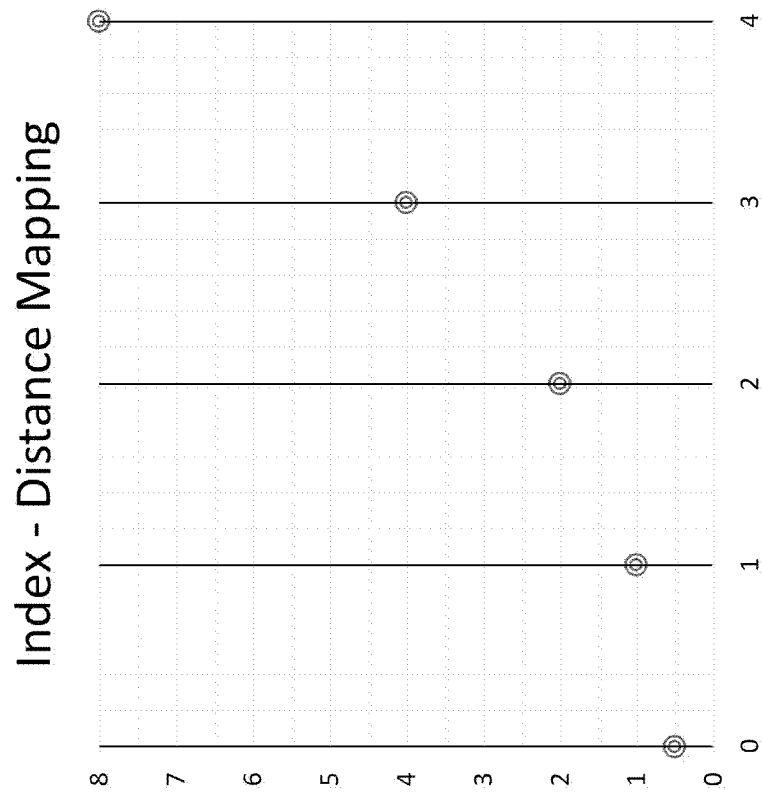


FIG. 7

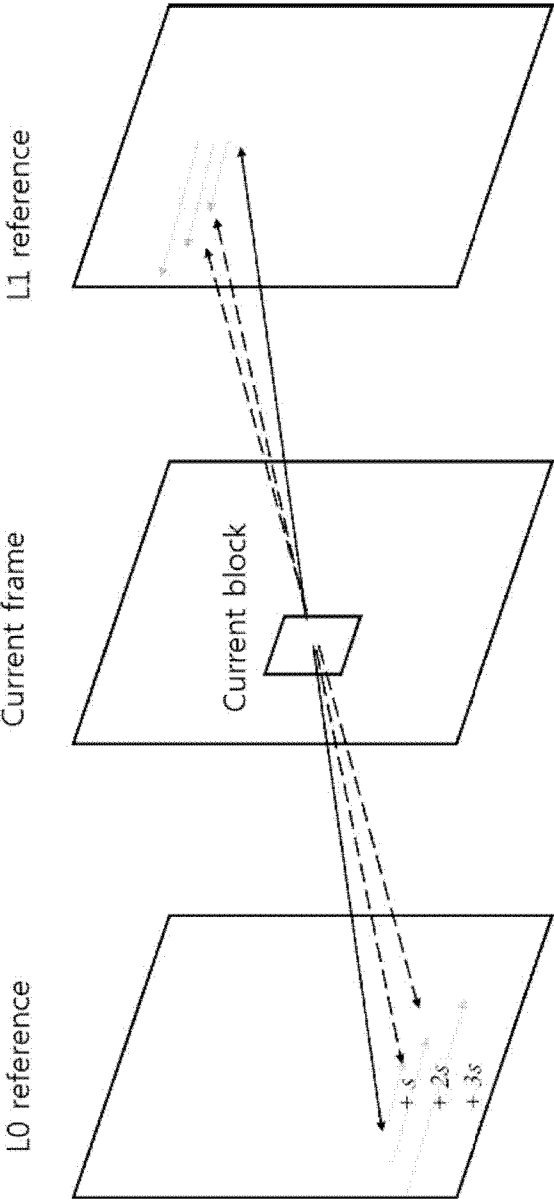


FIG. 8

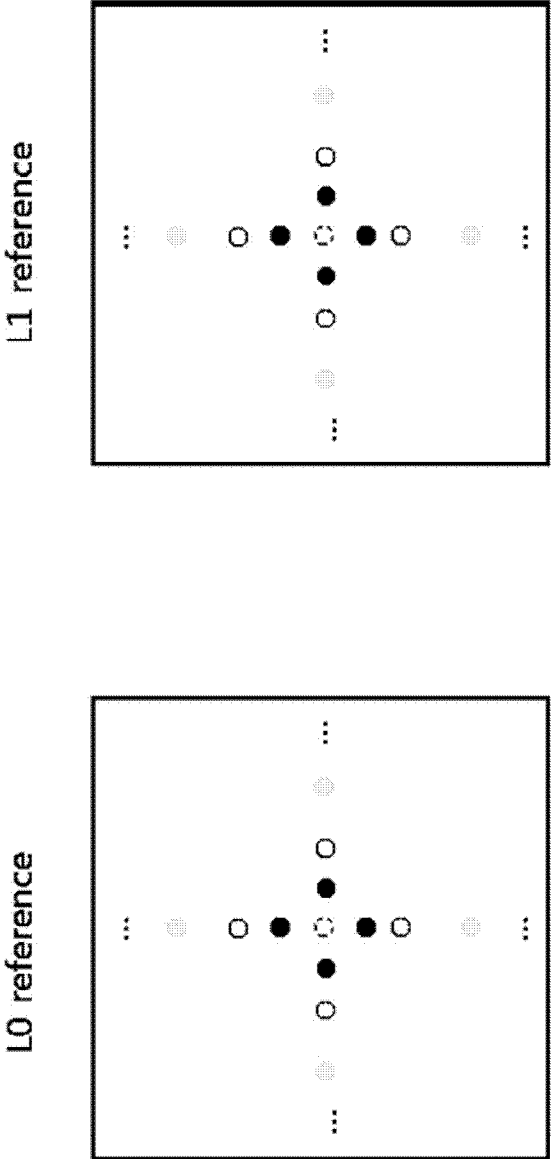


FIG. 9

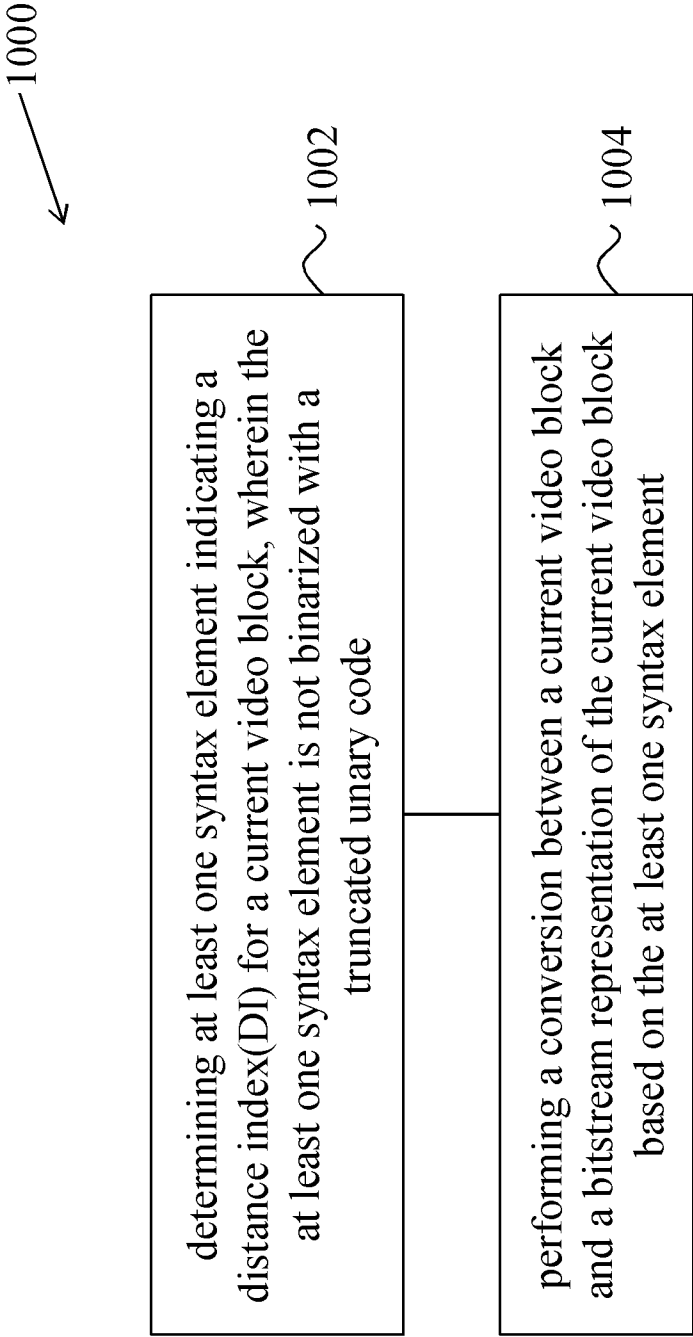


FIG. 10

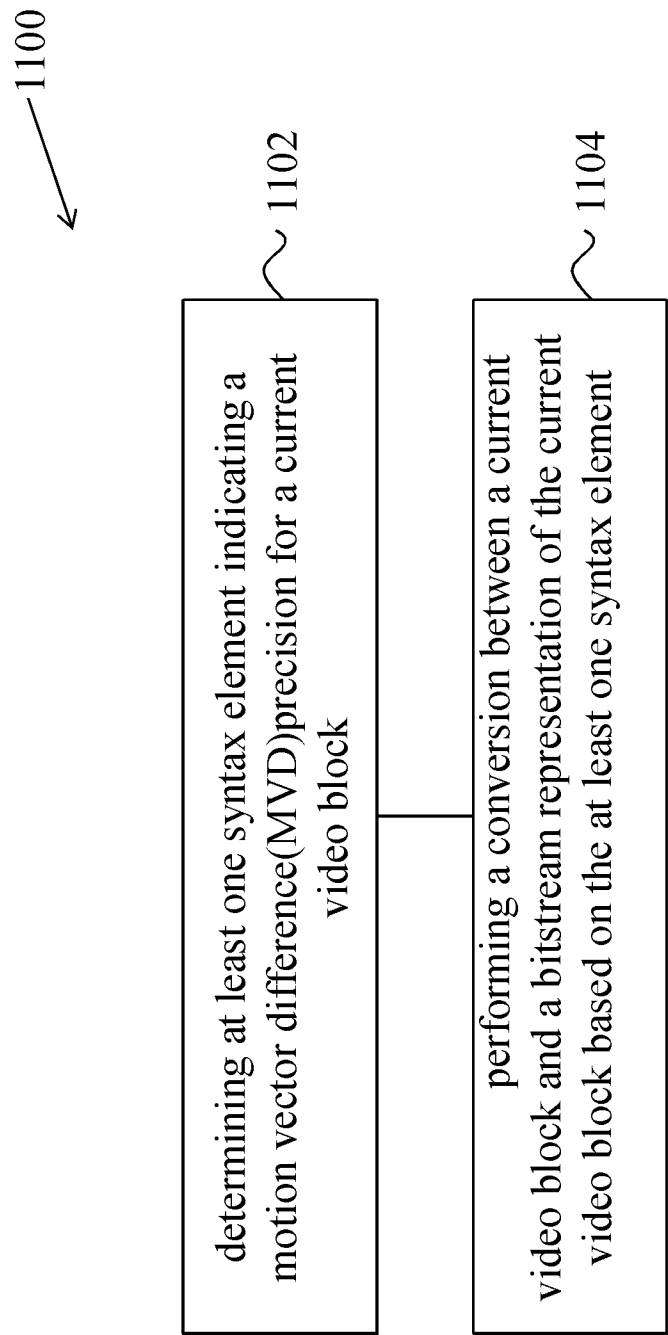


FIG. 11

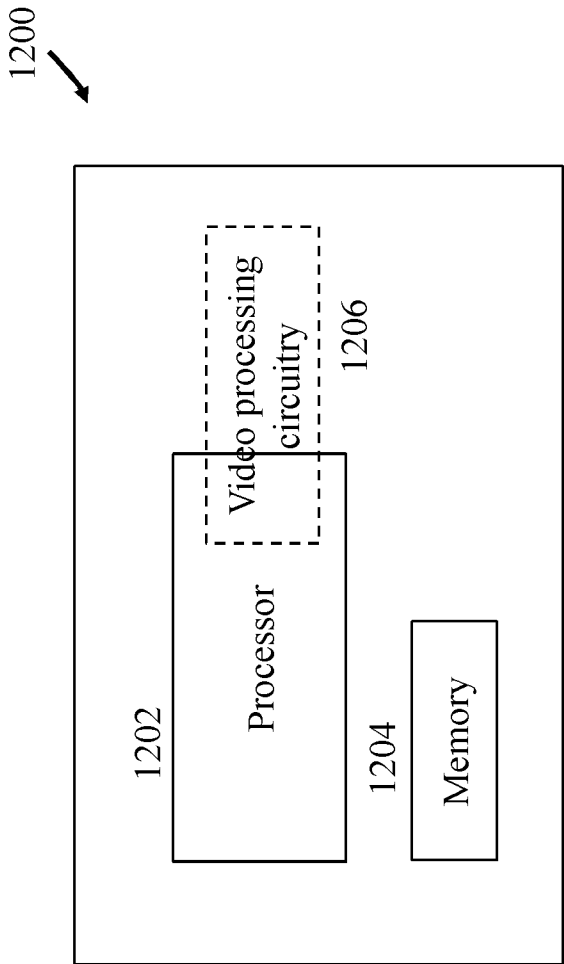


FIG. 12

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/130712

**A. CLASSIFICATION OF SUBJECT MATTER**

H04N 19/52(2014.01)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)

CNPAT,CNKI,EPODOC,WPI,IEEE,JVET: MVD, motion vector difference, precision, index, distance, conversion,syntax

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                            | Relevant to claim No. |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| A         | US 2018270485 A1 (LG ELECTRONICS INC.) 20 September 2018 (2018-09-20)<br>description, paragraphs 0176-0187 and figure 15                                                                                                      | 1-63                  |
| A         | WO 2018064524 A1 (QUALCOMM INCORPORATED) 05 April 2018 (2018-04-05)<br>the whole document                                                                                                                                     | 1-63                  |
| A         | WO 2018065296 A1 (THOMSON LICENSING) 12 April 2018 (2018-04-12)<br>the whole document                                                                                                                                         | 1-63                  |
| A         | WO 2018117546 A1 (KT CORPORATION) 28 June 2018 (2018-06-28)<br>the whole document                                                                                                                                             | 1-63                  |
| A         | CN 108141604 A (INDUSTRY-UNIVERSITY COOPERATION FOUNDATION HANYANG UNIVERSITY) 08 June 2018 (2018-06-08)<br>the whole document                                                                                                | 1-63                  |
| A         | HE, Yuwen et al. "CE4-related: Adaptive precision for affine MVD coding"<br><i>Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11</i> ,<br>12 October 2018 (2018-10-12),<br>the whole document | 1-63                  |



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

“&amp;” document member of the same patent family

Date of the actual completion of the international search

14 March 2020

Date of mailing of the international search report

01 April 2020

Name and mailing address of the ISA/CN

National Intellectual Property Administration, PRC  
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing  
100088  
China

Facsimile No. (86-10)62019451

Authorized officer

ZHANG,Suqing

Telephone No. 86-(10)-53961816

**INTERNATIONAL SEARCH REPORT**  
**Information on patent family members**

International application No.

**PCT/CN2019/130712**

| Patent document<br>cited in search report |            |    | Publication date<br>(day/month/year) | Patent family member(s) |              |    | Publication date<br>(day/month/year) |
|-------------------------------------------|------------|----|--------------------------------------|-------------------------|--------------|----|--------------------------------------|
| US                                        | 2018270485 | A1 | 20 September 2018                    | WO                      | 2017052009   | A1 | 30 March 2017                        |
|                                           |            |    |                                      | KR                      | 20180059444  | A  | 04 June 2018                         |
|                                           |            |    |                                      | US                      | 10547847     | B2 | 28 January 2020                      |
|                                           |            |    |                                      | EP                      | 3355582      | A1 | 01 August 2018                       |
|                                           |            |    |                                      | JP                      | 2018533298   | A  | 08 November 2018                     |
|                                           |            |    |                                      | EP                      | 3355582      | A4 | 17 April 2019                        |
|                                           |            |    |                                      | CN                      | 108353176    | A  | 31 July 2018                         |
| WO                                        | 2018064524 | A1 | 05 April 2018                        | EP                      | 3520410      | A1 | 07 August 2019                       |
|                                           |            |    |                                      | KR                      | 20190054082  | A  | 21 May 2019                          |
|                                           |            |    |                                      | BR                      | 112019006304 | A2 | 02 July 2019                         |
|                                           |            |    |                                      | US                      | 2018091816   | A1 | 29 March 2018                        |
|                                           |            |    |                                      | JP                      | 2019533363   | A  | 14 November 2019                     |
|                                           |            |    |                                      | US                      | 10462462     | B2 | 29 October 2019                      |
|                                           |            |    |                                      | CA                      | 3034925      | A1 | 05 April 2018                        |
|                                           |            |    |                                      | CN                      | 109792531    | A  | 21 May 2019                          |
|                                           |            |    |                                      | TW                      | 201824863    | A  | 01 July 2018                         |
| WO                                        | 2018065296 | A1 | 12 April 2018                        | EP                      | 3301918      | A1 | 04 April 2018                        |
| WO                                        | 2018117546 | A1 | 28 June 2018                         | KR                      | 20180073464  | A  | 02 July 2018                         |
|                                           |            |    |                                      | CN                      | 110100440    | A  | 06 August 2019                       |
|                                           |            |    |                                      | MX                      | 2019007219   | A  | 05 September 2019                    |
| CN                                        | 108141604  | A  | 08 June 2018                         | WO                      | 2016195453   | A1 | 08 December 2016                     |
|                                           |            |    |                                      | EP                      | 3306935      | A4 | 02 January 2019                      |
|                                           |            |    |                                      | EP                      | 3306935      | A1 | 11 April 2018                        |
|                                           |            |    |                                      | US                      | 2018176596   | A1 | 21 June 2018                         |