



(51) International Patent Classification:
H04N 19/56 (2014.01)

(21) International Application Number:
PCT/CN2020/138695

(22) International Filing Date:
23 December 2020 (23.12.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2019/127388
23 December 2019 (23.12.2019) CN

(71) Applicants: **BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD.** [CN/CN]; Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District, Beijing 100041 (CN). **BYTEDANCE INC.** [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) Inventors: **ZHANG, Na**, Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). **LIU, Hongbin**,

Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). **ZHANG, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Kai**; 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, 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, IT, 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,

(54) Title: IMPROVEMENTS ON MERGE MODE WITH MOTION VECTOR DIFFERENCE

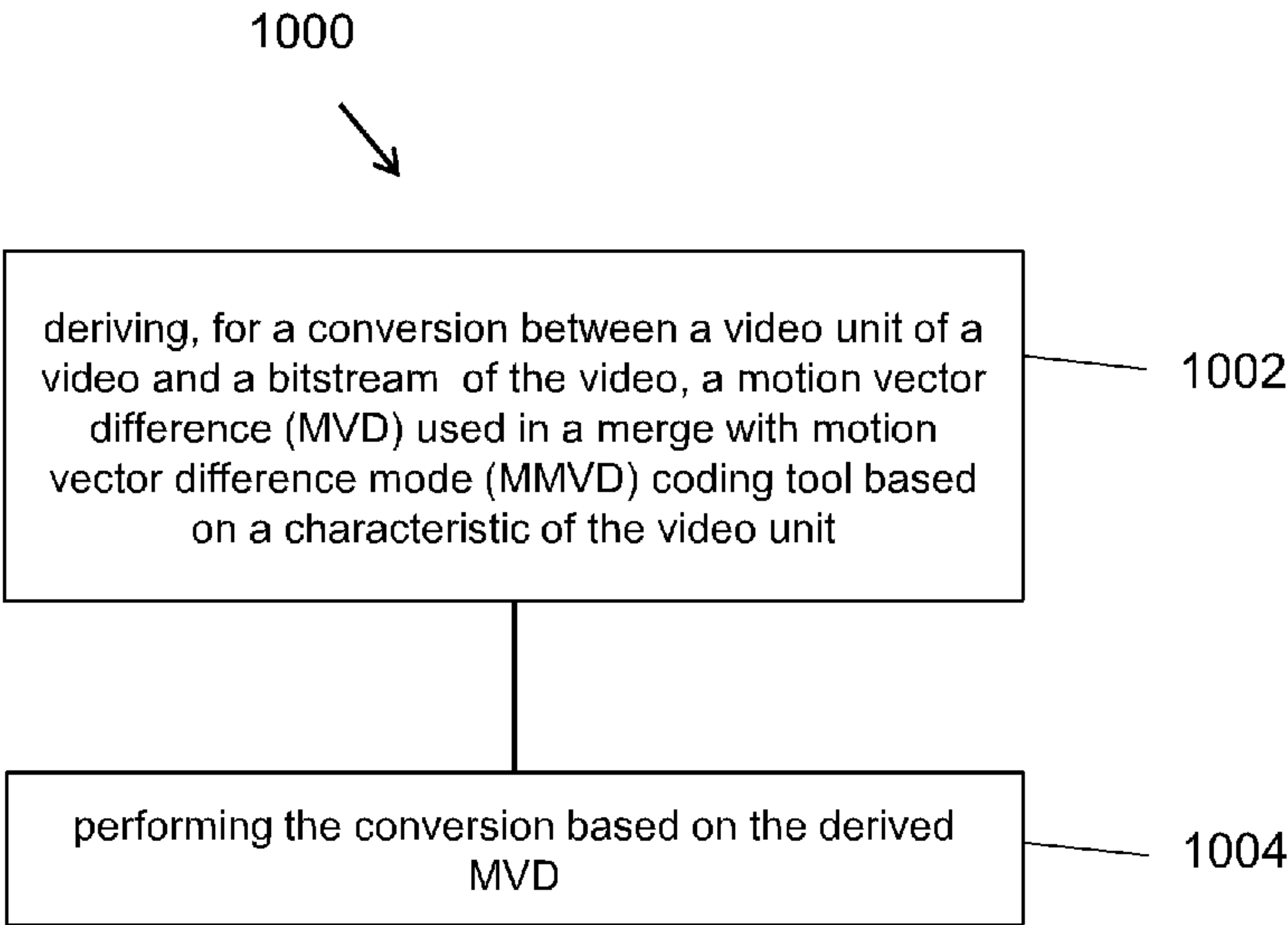


FIG. 10

(57) Abstract: Improvements on merge mode with motion vector difference are described. One example method of video processing includes deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; and performing the conversion based on the derived MVD.

SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, 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))*

IMPROVEMENTS ON MERGE MODE WITH MOTION VECTOR DIFFERENCE

CROSS-REFERENCE TO RELATED APPLICATION

[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/CN2019/127388, filed on December 23, 2019. The entire disclosures of International Patent Application No. PCT/CN2019/127388 are incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[002] This patent document relates to image and video 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 techniques that can be used by video encoders and decoders to perform cross-component adaptive loop filtering during video encoding or decoding.

[005] In one example aspect, a video processing method is disclosed. The method includes determining, for a conversion between a video unit of a video and a coded representation of the video, a calculation operation for a motion vector difference (MVD) for use with a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit and performing the conversion based on the determining.

[006] In another example aspect, a video processing method is disclosed. The method includes performing a conversion between a video unit of a video and a coded representation of the video, wherein the conversion uses a motion vector scaling process during an operation that is dependent of a resolution of the video.

[007] In another example aspect, a video processing method is disclosed. The method includes generating, for a conversion between a video unit of a video and a coded representation of the

video, a merge candidate list in which non-adjacent spatial merge candidates of the video unit are inserted in the merge list, and performing the conversion using the merge candidate list.

[008] In another example aspect, a video processing method is disclosed. The method includes generating, for a conversion between a video unit of a video and a coded representation of the video, a candidate list in which a candidate that is generated by averaging M spatial neighboring candidates and N temporal neighboring candidates, where M and N are positive integers; and performing the conversion using the merge candidate list.

[009] In another example aspect, a video processing method is disclosed. The method includes generating, for a conversion between a video unit of a video and a coded representation of the video, a merge list wherein a construction process used for generating the merge list checks a number of candidates in a defined order and performing the conversion using the merge candidate list.

[0010] In another example aspect, a video processing method is disclosed. The method includes performing a conversion between a video unit of a video and a coded representation of the video using two long term reference pictures and a motion vector scaling process during the conversion.

[0011] In another example aspect, a video processing method is disclosed. The method includes deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; and performing the conversion based on the derived MVD.

[0012] In another example aspect, a video processing method is disclosed. The method includes deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) using a motion vector (MV) scaling process, wherein the MV scaling process is dependent of a resolution of the video; and performing the conversion based on the derived MVD.

[0013] In another example aspect, a video processing method is disclosed. The method includes deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) using a motion vector (MV) scaling process, wherein the MV scaling process uses two long term reference pictures; and performing the conversion based on the derived MVD.

[0014] In another example aspect, a method for storing bitstream of a video is disclosed. The method includes deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; generating the bitstream from the

video unit based on the derived MVD; and storing the bitstream in a non-transitory computer-readable recording medium.

[0015] In another example aspect, a video processing method is disclosed. The method includes constructing, for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein non-adjacent spatial merge candidates associated with the current block are inserted into the merge candidate list; and performing the conversion based on the merge candidate list.

[0016] In another example aspect, a video processing method is disclosed. The method includes constructing, for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein the construction process of the merge candidate list checks a number of different kinds of candidates in a defined order; and performing the conversion based on the merge candidate list.

[0017] In another example aspect, a method for storing bitstream of a video is disclosed. The method includes constructing, for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein non-adjacent spatial merge candidates associated with the current block are inserted into the merge candidate list; and generating the bitstream from the video unit based on the merge candidate list; and storing the bitstream in a non-transitory computer-readable recording medium.

[0018] In another example aspect, a video processing method is disclosed. The method includes constructing, for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein a spatial-temporal motion vector prediction (STMVP) candidate associated with the current block are added to the merge candidate list, and the STMVP candidate is derived as an averaging candidate of M spatial neighboring motion candidates and/or N temporal neighboring motion candidates, M and N being positive integers; and performing the conversion based on the merge candidate list.

[0019] In another example aspect, a method for storing bitstream of a video is disclosed. The method includes constructing, for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein a spatial-temporal motion vector prediction (STMVP) candidate associated with the current block are added to the merge candidate list, and the STMVP candidate is derived as an averaging candidate of M spatial neighboring motion candidates and/or N temporal neighboring motion candidates, M and N being positive integers; generating the bitstream from the video unit based on on the merge candidate list; and storing the bitstream in a non-transitory computer-readable recording medium.

[0020] In yet another example aspect, a video encoder apparatus is disclosed. The video encoder comprises a processor configured to implement above-described methods.

[0021] In yet another example aspect, a video decoder apparatus is disclosed. The video decoder comprises a processor configured to implement above-described methods.

[0022] In yet another example aspect, a computer readable medium having code stored thereon is disclosed. The code embodies one of the methods described herein in the form of processor-executable code.

[0023] In yet another example aspect, a computer readable medium storing a bitstream of a video which is generated by above-described methods performed by a video processing apparatus.

[0024] These, and other, features are described throughout the present document.

BRIEF DESCRIPTION OF DRAWINGS

[0025] FIG. 1 shows an example of offsets added to either horizontal or vertical component of a starting motion vector (MV).

[0026] FIG. 2 shows an HEVC spatial neighboring blocks of the current block.

[0027] FIG. 3 illustrates the relationship between the virtual block and the current block.

[0028] FIG. 4 is a block diagram of an example video processing system in which disclosed techniques may be implemented.

[0029] FIG. 5 is a block diagram of an example hardware platform used for video processing.

[0030] FIG. 6 is a flowchart for an example method of video processing.

[0031] FIG. 7 is a block diagram that illustrates a video coding system in accordance with some embodiments of the present disclosure.

[0032] FIG. 8 is a block diagram that illustrates an encoder in accordance with some embodiments of the present disclosure.

[0033] FIG. 9 is a block diagram that illustrates a decoder in accordance with some embodiments of the present disclosure.

[0034] FIG. 10 is a flowchart for an example method of video processing.

[0035] FIG. 11 is a flowchart for an example method of video processing.

[0036] FIG. 12 is a flowchart for an example method of video processing.

[0037] FIG. 13 is a flowchart for an example method for storing bitstream of a video.

[0038] FIG. 14 is a flowchart for an example method of video processing.

[0039] FIG. 15 is a flowchart for an example method of video processing.

[0040] FIG. 16 is a flowchart for an example method for storing bitstream of a video.

[0041] FIG. 17 is a flowchart for an example method of video processing.

[0042] FIG. 18 is a flowchart for an example method for storing bitstream of a video.

DETAILED DESCRIPTION

[0043] Section headings are used in the present document for ease of understanding and do not limit the applicability of techniques and embodiments disclosed in each section only to that section. Furthermore, H.266 terminology is used in some description only for ease of understanding and not for limiting scope of the disclosed techniques. As such, the techniques described herein are applicable to other video codec protocols and designs also.

1. Summary

[0044] This patent document is related to video coding technologies. Specifically, it is related to merge mode 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.

2. Background

[0045] 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.

2.1. Merge mode with MVD (MMVD)

[0046] In addition to merge mode, where the implicitly derived motion information is directly used for prediction samples generation of the current CU, the merge mode with motion vector differences (MMVD) is introduced in VVC, which is also known as ultimate motion vector

expression. A MMVD flag is signaled right after sending a skip flag and merge flag to specify whether MMVD mode is used for a CU.

[0047] In MMVD, a merge candidate (which is called, base merge candidate) is selected, it is further refined by the signaled MVD information. The related syntax elements include an index to specify MVD distance (denoted by **mmvd_distance_idx**), and an index for indication of motion direction (denoted by **mmvd_direction_idx**). In MMVD mode, one for the first two candidates in the merge list is selected to be used as MV basis (or base merge candidate). The merge candidate flag is signaled to specify which one is used.

[0048] Distance index specifies motion magnitude information and indicate the pre-defined offset from the starting point. As shown in FIG. 1, an offset is added to either horizontal component or vertical component of starting MV. The relation of distance index and pre-defined offset is specified in Table 3.

Table 3: The relation of distance index and pre-defined offset

Distance IDX	0	1	2	3	4	5	6	7
Offset (in unit of luma sample)	1/4	1/2	1	2	4	8	16	32

[0049] Direction index represents the direction of the MVD relative to the starting point. The direction index can represent of the four directions as shown in Table 4. It's noted that the meaning of MVD sign could be variant according to the information of starting MVs. When the starting MVs is an un-prediction MV or bi-prediction MVs with both lists point to the same side of the current picture (i.e. POCs of two references are both larger than the POC of the current picture, or are both smaller than the POC of the current picture), the sign in Table 4 specifies the sign of MV offset added to the starting MV. When the starting MVs is bi-prediction MVs with the two MVs point to the different sides of the current picture (i.e. the POC of one reference is larger than the POC of the current picture, and the POC of the other reference is smaller than the POC of the current picture), the sign in Table 4 specifies the sign of MV offset added to the list0 MV component of starting MV and the sign for the list1 MV has opposite value.

Table 4: Sign of MV offset specified by direction index

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

2.2.1 Derivation of MVD for each reference picture list

[0050] One internal MVD (denoted by **MmvdOffset**) is firstly derived according to the decoded indices of MVD distance (denoted by **mmvd_distance_idx**), and motion direction (denoted by **mmvd_direction_idx**).

[0051] Afterwards, if the internal MVD is determined, the final MVD to be added to the base merge candidate for each reference picture list is further derived according to POC distances of reference pictures relative to the current picture, and reference picture types (long-term or short-term). More specifically, the following steps are performed in order:

- If the base merge candidate is bi-prediction, the POC distance between current picture and reference picture in list 0, and the POC distance between current picture and reference picture in list 1 is calculated, denoted by **POCDiffL0**, and **POCDiffL1**, respectively.
- If **POCDiffL0** is equal to **POCDiffL1**, the final MVD for two reference picture lists are both set to the internal MVD.
- Otherwise, if **Abs(POCDiffL0)** is greater than or equal to **Abs(POCDiffL1)**, the final MVD for reference picture list 0 is set to the internal MVD, and the final MVD for reference picture list 1 is set to the scaled MVD using the internal MVD reference picture types of the two reference pictures (both are not long-term reference pictures) or the internal MVD or (zero MV minus the internal MVD) depending on the POC distances.
- Otherwise, if **Abs(POCDiffL0)** is smaller than **Abs(POCDiffL1)**, the final MVD for reference picture list 1 is set to the internal MVD, and the final MVD for reference picture list 0 is set to the scaled MVD using the internal MVD reference picture types of the two reference pictures (both are not long-term reference pictures) or the internal MVD or (zero MV minus the internal MVD) depending on the POC distances.
- If the base merge candidate is uni-prediction from reference picture list X, the final MVD for reference picture list X is set to the internal MVD, and the final MVD for reference picture list Y ($Y=1-X$) is set to 0.

2.2.2 Spec for MMVD in VVC

[0052] Spec of MMVD (in JVET-P2001-vE) is as follows:

7.3.9.7 Merge data syntax

merge_data(x0, y0, cbWidth, cbHeight, chType) {	Descriptor
if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
if(MaxNumIbcMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
if(MaxNumSubblockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_subblock_flag [x0][y0]	ae(v)
if(merge_subblock_flag[x0][y0] == 1) {	
if(MaxNumSubblockMergeCand > 1)	
merge_subblock_idx [x0][y0]	ae(v)
} else {	
if((cbWidth * cbHeight) >= 64 && (sps_ciip_enabled_flag && cu_skip_flag[x0][y0] == 0 && cbWidth < 128 && cbHeight < 128) (sps_triangle_enabled_flag && MaxNumTriangleMergeCand > 1 && slice_type == B)))	
regular_merge_flag [x0][y0]	ae(v)
if(regular_merge_flag[x0][y0] == 1) {	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag [x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] == 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag [x0][y0]	ae(v)
mmvd_distance_idx [x0][y0]	ae(v)
mmvd_direction_idx [x0][y0]	ae(v)
} else if(MaxNumMergeCand > 1)	
merge_idx [x0][y0]	ae(v)
} else {	
...	
}	
}	

mmvd_merge_flag[x0][y0] equal to 1 specifies that merge mode with motion vector difference is used to generate the inter prediction parameters of the current coding unit. **mmvd_merge_flag**[x0][y0] equal to 0 specifies that merge mode with motion vector difference is not used to generate the inter prediction paramters. 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.

When **mmvd_merge_flag**[x0][y0] is not present, it is inferred to be equal to 0.

mmvd_cand_flag[x0][y0] specifies whether the first (0) or the second (1) candidate in the merging candidate list is used with the motion vector difference derived from **mmvd_distance_idx**[x0][y0] and

$\text{mmvd_direction_idx}[x0][y0]$. 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.

When $\text{mmvd_cand_flag}[x0][y0]$ is not present, it is inferred to be equal to 0.

$\text{mmvd_distance_idx}[x0][y0]$ specifies the index used to derive $\text{MmvdDistance}[x0][y0]$ as specified in Table 17. 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.

Table 17 – Specification of $\text{MmvdDistance}[x0][y0]$ based on $\text{mmvd_distance_idx}[x0][y0]$

$\text{mmvd_distance_idx}[x0][y0]$	$\text{MmvdDistance}[x0][y0]$	
	$\text{pic_fpel_mmvd_enabled_flag} = 0$	$\text{pic_fpel_mmvd_enabled_flag} = 1$
0	1	4
1	2	8
2	4	16
3	8	32
4	16	64
5	32	128
6	64	256
7	128	512

$\text{mmvd_direction_idx}[x0][y0]$ specifies index used to derive $\text{MmvdSign}[x0][y0]$ as specified in Table 18. 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.

Table 18 – Specification of $\text{MmvdSign}[x0][y0]$ based on $\text{mmvd_direction_idx}[x0][y0]$

$\text{mmvd_direction_idx}[x0][y0]$	$\text{MmvdSign}[x0][y0][0]$	$\text{MmvdSign}[x0][y0][1]$
0	+1	0
1	-1	0
2	0	+1
3	0	-1

Both components of the merge plus MVD offset $\text{MmvdOffset}[x0][y0]$ are derived as follows:

$$\text{MmvdOffset}[x0][y0][0] = (\text{MmvdDistance}[x0][y0] \ll 2) * \text{MmvdSign}[x0][y0][0] \quad (181)$$

$$\text{MmvdOffset}[x0][y0][1] = (\text{MmvdDistance}[x0][y0] \ll 2) * \text{MmvdSign}[x0][y0][1] \quad (182)$$

8.5.2.7 Derivation process for merge motion vector difference

Inputs to this process are:

- a luma location (x_{Cb}, y_{Cb}) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- reference indices refIdxL0 and refIdxL1 ,

- prediction list utilization flags predFlagL0 and predFlagL1.

Outputs of this process are the luma merge motion vector differences in 1/16 fractional-sample accuracy mMvdL0 and mMvdL1.

The variable currPic specifies the current picture.

The luma merge motion vector differences mMvdL0 and mMvdL1 are derived as follows:

- If both predFlagL0 and predFlagL1 are equal to 1, the following applies:

$$\text{currPocDiffL0} = \text{DiffPicOrderCnt}(\text{currPic}, \text{RefPicList}[0][\text{refIdxL0}]) \quad (564)$$

$$\text{currPocDiffL1} = \text{DiffPicOrderCnt}(\text{currPic}, \text{RefPicList}[1][\text{refIdxL1}]) \quad (565)$$

- If currPocDiffL0 is equal to currPocDiffL1, the following applies:

$$\text{mMvdL0}[0] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][0] \quad (566)$$

$$\text{mMvdL0}[1] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][1] \quad (567)$$

$$\text{mMvdL1}[0] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][0] \quad (568)$$

$$\text{mMvdL1}[1] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][1] \quad (569)$$

- Otherwise, if Abs(currPocDiffL0) is greater than or equal to Abs(currPocDiffL1), the following applies:

$$\text{mMvdL0}[0] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][0] \quad (570)$$

$$\text{mMvdL0}[1] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][1] \quad (571)$$

- If RefPicList[0][refIdxL0] is not a long-term reference picture and RefPicList[1][refIdxL1] is not a long-term reference picture, the following applies:

$$\text{td} = \text{Clip3}(-128, 127, \text{currPocDiffL0}) \quad (572)$$

$$\text{tb} = \text{Clip3}(-128, 127, \text{currPocDiffL1}) \quad (573)$$

$$\text{tx} = (16384 + (\text{Abs}(\text{td}) \gg 1)) / \text{td} \quad (574)$$

$$\text{distScaleFactor} = \text{Clip3}(-4096, 4095, (\text{tb} * \text{tx} + 32) \gg 6) \quad (575)$$

$$\text{mMvdL1}[0] = \text{Clip3}(-2^{17}, 2^{17} - 1, (\text{distScaleFactor} * \text{mMvdL0}[0] + 128 - (\text{distScaleFactor} * \text{mMvdL0}[0] \geq 0)) \gg 8) \quad (576)$$

$$\text{mMvdL1}[1] = \text{Clip3}(-2^{17}, 2^{17} - 1, (\text{distScaleFactor} * \text{mMvdL0}[1] + 128 - (\text{distScaleFactor} * \text{mMvdL0}[1] \geq 0)) \gg 8) \quad (577)$$

- Otherwise, the following applies:

$$\begin{aligned} \text{mMvdL1}[0] &= \text{Sign}(\text{currPocDiffL0}) == \text{Sign}(\text{currPocDiffL1}) ? \\ &\quad \text{mMvdL0}[0] : -\text{mMvdL0}[0] \end{aligned} \quad (578)$$

$$\begin{aligned} \text{mMvdL1}[1] &= \text{Sign}(\text{currPocDiffL0}) == \text{Sign}(\text{currPocDiffL1}) ? \\ &\quad \text{mMvdL0}[1] : -\text{mMvdL0}[1] \end{aligned} \quad (579)$$

- Otherwise (Abs(currPocDiffL0) is less than Abs(currPocDiffL1)), the following applies:

$$mMvdL1[0] = MmvdOffset[xCb][yCb][0] \quad (580)$$

$$mMvdL1[1] = MmvdOffset[xCb][yCb][1] \quad (581)$$

- If RefPicList[0][refIdxL0] is not a long-term reference picture and RefPicList[1][refIdxL1] is not a long-term reference picture, the following applies:

$$td = Clip3(-128, 127, currPocDiffL1) \quad (582)$$

$$tb = Clip3(-128, 127, currPocDiffL0) \quad (583)$$

$$tx = (16384 + (Abs(td) >> 1)) / td \quad (584)$$

$$distScaleFactor = Clip3(-4096, 4095, (tb * tx + 32) >> 6) \quad (585)$$

$$mMvdL0[0] = Clip3(-2^{17}, 2^{17} - 1, (distScaleFactor * mMvdL1[0] + 128 - (distScaleFactor * mMvdL1[0] >= 0) >> 8)) \quad (586)$$

$$mMvdL0[1] = Clip3(-2^{17}, 2^{17} - 1, (distScaleFactor * mMvdL1[1] + 128 - (distScaleFactor * mMvdL1[1] >= 0) >> 8)) \quad (587)$$

- Otherwise, the following applies:

$$mMvdL0[0] = Sign(currPocDiffL0) == Sign(currPocDiffL1) ? mMvdL1[0] : -mMvdL1[0] \quad (588)$$

$$mMvdL0[1] = Sign(currPocDiffL0) == Sign(currPocDiffL1) ? mMvdL1[1] : -mMvdL1[1] \quad (589)$$

- Otherwise (predFlagL0 or predFlagL1 are equal to 1), the following applies for X being 0 and 1:

$$mMvdLX[0] = (predFlagLX == 1) ? MmvdOffset[xCb][yCb][0] : 0 \quad (590)$$

$$mMvdLX[1] = (predFlagLX == 1) ? MmvdOffset[xCb][yCb][1] : 0 \quad (591)$$

2.2. JVET-L0323: Long distance merge candidates

[0053] In HEVC, five spatially neighboring blocks shown in FIG. 2 as well as one temporal neighbor are used to derive merge candidates.

[0054] FIG. 2 shows an HEVC spatial neighboring blocks of the current block.

[0055] This contribution proposes to derive the additional merge candidates from the positions non-adjacent to the current block using the same pattern as that in HEVC. To achieve this, for each search round i , a virtual block is generated based on the current block as follows:

[0056] First, the relative position of the virtual block to the current is calculated by:

[0057] $Offsetx = -i * gridX$, $Offsety = -i * gridY$

[0058] where the $Offsetx$ and $Offsety$ denote the offset of the top-left corner of the virtual block relative to the top-left corner of the current block, $gridX$ and $gridY$ are the width and height of the search grid.

[0059] Second, the width and height of the virtual block are calculated by:

[0060] $\text{newWidth} = i * 2 * \text{gridX} + \text{currWidth}$ $\text{newHeight} = i * 2 * \text{gridY} + \text{currHeight}$.

[0061] where the *currWidth* and *currHeight* are the current block width and height. The *newWidth* and *newHeight* are the new block width and height.

[0062] *gridX* and *gridY* are currently set to *currWidth* and *currHeight*, respectively.

[0063] FIG. 3 illustrates the relationship between the virtual block and the current block.

[0064] After generating the virtual block, the blocks A_i , B_i , C_i , D_i and E_i can be regarded as the HEVC spatial neighboring blocks of the virtual block and their positions are obtained with the same pattern as that in HEVC. Obviously, the virtual block is the current block if the search round i is 0. In this case, the blocks A_i , B_i , C_i , D_i and E_i are the spatially neighboring blocks that are used in HEVC merge mode.

[0065] When constructing the merge candidate list, the pruning is performed to guarantee each element in merge candidate list to be unique. As more and more blocks will be checked to derive additional merge candidates, the pruning number is correspondingly increased. To limit the pruning number in the worst case, the maximum number of pruning that is allowed in merge list construction is constrained to a pre-defined value *MaxPruningNum*.

[0066] In the simulations, the maximum search round is set to 2 and *MaxPruningNum* is set to 30.

[0067] Long distance merge candidates is also known as non-adjacent merge candidates.

[0068] FIG. 3 is an illustration of virtual block in the i th search round.

2.3. JVET-M0059: Non-scaling STMVP

[0069] The proposed method derives an averaging candidate as STMVP candidate using two spatial merge candidate and one collocated merge candidate.

[0070] STMVP is inserted before the above-left spatial merge candidate.

[0071] For the spatial candidates, the first and second candidates in the current merge candidate list is used.

[0072] For the temporal candidate, the same position as VTM / HEVC collocated position is used.

[0073] If three candidates, of which the reference are equal to zero, are available, the following apply.

$$\text{mvLX}[0] = (\text{mvLX_A}[0] * 3 + \text{mvLX_B}[0] * 3 + \text{mvLX_C}[0] * 2) / 8$$

$$\text{mvLX}[1] = (\text{mvLX_A}[1] * 3 + \text{mvLX_B}[1] * 3 + \text{mvLX_C}[1] * 2) / 8$$

If two motion information, of which the reference are equal to zero, is available, the following apply

$$\text{mvLX}[0] = (\text{mvLX_A}[0] + \text{mvLX_C}[0]) / 2$$

$$mvLX[1] = (mvLX_A[1] + mvLX_C[1]) / 2$$

or

$$mvLX[0] = (mvLX_B[0] + mvLX_C[0]) / 2$$

$$mvLX[1] = (mvLX_B[1] + mvLX_C[1]) / 2$$

[0074] Note: If the temporal candidate is unavailable, the STMVP mode is off.

[0075] MMVD is also known as Ultimate Motion Vector Expression (UMVE).

3. Technical problems solved by technical solutions and embodiments herein

[0076] The current design of merge mode can be further improved.

1. In MMVD mode, for small blocks (e.g., 4x8/8x4), even only uni-prediction is allowed, two MVDs may be still derived if the base merge candidate is bi-prediction. More specifically, if the selected MV basis (or base merge candidate) is a bi-directional MV, MVD of prediction direction from one reference list X (X = 0 or 1) is set equal to the signaled MVD directly and MVD of the other reference list Y (Y = 1 – X) is derived according to MVD of prediction direction X and the POC (Picture Order Count) distance so that in some cases scaling is required. However, in VTM-7.0, for 4x8/8x4 blocks, the bi-prediction is forbidden. Therefore, there's no need to derive the MVD of L1.
2. In addition, non-adjacent merge candidates and/or STMVP can be used to improve the effectiveness of merge mode. Furthermore, coding efficiency can be improved.

4. Example embodiments and techniques

[0077] The items below should be considered as examples to explain general concepts. These items should not be interpreted in a narrow way. Furthermore, these items can be combined in any manner.

[0078] In MMVD, the internal MVD is the one derived from the signaled syntax elements in the bitstream, such as the MVD distance and direction information. And the final MVD is the one used to refine the base merge candidate, i.e., the one utilized to derive final MVs of a block.

[0079] Hereinafter, *currWidth* and *currHeight* are the width and height of current block (e.g., luma block). *maxNumMergeCand* denotes the merge list size.

[0080] As illustrated in section 2.3, after generating the virtual block, the blocks A_i , B_i , C_i , D_i and E_i can be regarded as the HEVC spatial neighboring blocks of the virtual block and their positions

are obtained with the same pattern as that in HEVC. Obviously, the virtual block is the current block if the search round i is 0. In this case, the blocks A_i , B_i , C_i , D_i and E_i are the spatially neighboring blocks that are used in HEVC merge mode.

[0081] For the spatial candidates, the first, second, and third candidates inserted in the current merge candidate list before STMVP are denoted as F , S , and T .

For the temporal candidate with the same position as VTM / HEVC collocated position used in STMVP is denoted as Col .

MVD derivation for MMVD

1. How to derive the MVD used in MMVD method may depend on the block dimension and/or the allowed prediction directions (e.g., whether only uni-prediction is allowed for a video unit (e.g., CU/PU).
 - a. In one example, if only uni-prediction is allowed for a video unit, only one MVD is derived instead of two from the internal MVD, regardless of the prediction direction associated with the base merge candidate in MMVD.
 - i. In one example, if only prediction from reference picture list X is the prediction direction is denoted by $ListX$ (e.g., X being 0), then the final MVD for the $ListX$ is derived from the internal MVD.
 - (i) Alternatively, furthermore, the final MVD for the $ListX$ is set equal to the internal MVD.
 - (ii) Alternatively, furthermore, the final MVD for the $ListX$ is set equal to the opposite of the internal MVD.
 - (iii) Alternatively, furthermore, the final MVD for the $ListY$ is set to default values, e.g., zero MVD.
 - b. In one example, if certain condition(s) depending on block dimension is satisfied, only one MVD is derived instead of two from the internal MVD, regardless of the prediction direction associated with the base merge candidate in MMVD.
 - i. In one example, the condition is that $currWidth + currHeight$ is less than or equal to N (N is a positive integer). For example, $N = 12$.
 - ii. In one example, the condition is that $currWidth * currHeight$ is less than or equal to N (N is a positive integer). For example, $N = 32$.
 - iii. In one example, the condition is that $currWidth < N1$ or/and $currHeight < N2$ ($N1$, $N2$ are positive integers). For example, $N1 = N2 = 8$.

- iv. In one example, the condition is that $\text{currWidth} < N3 * \text{currHeight}$ and/or $\text{currHeight} < N4 * \text{currWidth}$ ($N3, N4$ are positive integers). For example, $N3 = N4 = 8$.
- 2. It is proposed that in MMVD, when the base merge candidate is a bi-directional MV, the internal MVD may be always directly used (e.g., without scaling) for prediction direction X ($X = 0, 1$) if the block dimension or block shape satisfies one or more conditions.
 - a. In one example, the internal MVD is always directly used for prediction direction 0.
 - b. In one example, the condition is that $\text{currWidth} + \text{currHeight}$ is less than or equal to N (N is a positive integer). For example, $N = 12$.
 - c. In one example, the condition is that $\text{currWidth} * \text{currHeight}$ is less than or equal to N (N is a positive integer). For example, $N = 32$.
 - d. In one example, the condition is that $\text{currWidth} < N1$ or/and $\text{currHeight} < N2$ ($N1, N2$ are positive integers). For example, $N1 = N2 = 8$.
 - e. In one example, the condition is that $\text{currWidth} < N3 * \text{currHeight}$ and/or $\text{currHeight} < N4 * \text{currWidth}$ ($N3, N4$ are positive integers). For example, $N3 = N4 = 8$.
 - f. Opposite of the internal MVD (-MVD) may be used instead of MVD prediction direction X ($X = 0, 1$) if the block dimension or block shape satisfies one or more conditions.
- 3. The MV scaling process (e.g., those used in MMVD, TMVP, etc. al) may take the picture resolution into consideration.
- 4. For two reference pictures which are both long-term reference pictures, the MV scaling process may still be applied.
 - a. In one example, the MV scaling process may be similar to the case wherein two reference pictures are short-term reference pictures, i.e., depending on the POC distances.

Non-adjacent merge candidates

- 5. Non-adjacent spatial merge candidate may be inserted into the merge list.
 - a. In one example, non-adjacent spatial merge candidates are inserted into the merge list after history-based merge candidates.
 - b. In one example, non-adjacent spatial merge candidates are inserted into the merge list after pairwise average merge candidates.

- c. In one example, if the number of the available merge candidates in merge list reaches a pre-defined value after inserting the temporal merge candidate, the non-adjacent spatial merge candidates may not be inserted.
- d. In one example, if the number of the available merge candidates in merge list reaches a pre-defined value when inserting the non-adjacent spatial merge candidates, the inserting process will be terminated.
- e. In one example, the pre-defined value is equal to $\text{maxNumMergeCand} - N$.
 - i. In one example, N is set equal to 1, 2, 3, or 4.
- f. In one example, the maximum search round is set equal to 1 or 2, i.e., five or ten non-adjacent spatial merge candidates can be used to construct the merge list.
- g. In one example, for each search round, the inserting order is A_i , B_i , C_i , D_i , and E_i .
 - i. Alternatively, for each search round, the inserting order is B_i , A_i , C_i , D_i , and E_i .
 - ii. Alternatively, for each search round, the inserting order is B_i , C_i , A_i , D_i , and E_i .
 - iii. Alternatively, for each search round, the inserting order is A_i , D_i , B_i , C_i , and E_i .
- h. In one example, all spatial and temporal merge candidates perform full pruning with all the previous merge candidates in the merge list. The pruning process of history-based merge candidates and pairwise average candidates is not changed.
- i. Alternatively, all spatial, temporal, history-based, and pairwise average merge candidates perform full pruning with all the previous merge candidates in the merge list.
- j. Alternatively, for the non-adjacent spatial merge candidates, A_i performs pruning with A_{i-1} , B_i performs pruning with A_i , C_i performs pruning with B_i , D_i performs pruning with A_i , E_i performs pruning with A_i and B_i . The pruning process of temporal, history-based, and pairwise average candidates is not changed.
- k. In one example, the maximum number of pruning that is allowed in the merge list construction MaxPruningNum may depend on the merge list size maxNumMergeCand .
 - i. For example, MaxPruningNum may be set equal to $\text{maxNumMergeCand} - M$ (M is an integer). For example, $M = 2$.
 - ii. For example, MaxPruningNum may be set equal to $\text{maxNumMergeCand} * M$ (M is an integer). For example, $M = 2$.

- iii. Alternatively, MaxPruningNum is independent of the merge list size maxNumMergeCand. For example, MaxPruningNum may be set equal to 30 or 35.
- l. In one example, the non-adjacent spatial merge candidate position is constrained to be within a predefined area.
 - i. In one example, the area may contain the current CTU row and four sample rows above the current CTU row.
 - ii. In one example, the area may contain the current CTU column and four left sample columns of the current CTU.
 - iii. In one example, the area may contain the current CTU column and the left CTU column of the current CTU.
- m. In one example, the non-adjacent spatial merge candidate position has no constrain in horizontal direction.
- n. In one example, a non-adjacent spatial merge candidate can be used as the base merge candidate for MMVD.
 - i. Alternatively, a non-adjacent spatial merge candidate is not allowed to be used as the base merge candidate for MMVD.
- o. In one example, a non-adjacent spatial merge candidate can be used to generate the inter-intra prediction.
 - i. Alternatively, a non-adjacent spatial merge candidate is not allowed to generate the inter-intra prediction.
- p. In one example, a non-adjacent spatial merge candidate can be used to generate a geometry (GEO) partition and/or triangular partition merge candidate.
 - i. Alternatively, a non-adjacent spatial merge candidate is not allowed to generate geometry (GEO) partition and/or triangular partition merge candidate.
- q. In one example, a non-adjacent spatial merge candidate can be used to generate an affine merge candidate.
- r. In one example, a non-adjacent spatial merge candidate can be used to generate an Advanced Motion Vector Prediction (AMVP) candidate.

Spatial-temporal motion vector prediction (STMVP)

6. STMVP candidate may be derived as an averaging candidate of M spatial neighboring motion candidates and/or N temporal neighboring motion candidates.
 - a. In one example, $M > 2$.
 - b. In one example, the spatial neighboring motion candidates may be derived from other neighboring blocks different from or same as those utilized for merge list construction process.
 - c. In one example, the spatial neighboring motion candidates may be selected from the spatial merge candidates included in the merge list.
 - d. In one example, the spatial neighboring motion candidates may be selected from the first or last M spatial merge candidates included in the merge list before adding a STMVP.
 - e. In one example, the spatial neighboring motion candidates may be selected from the first or last M merge candidates included in the merge list before adding a STMVP.
 - f. In one example, the temporal neighboring motion candidates may be selected from the temporal merge candidates.
 - i. In one example, if the temporal merge candidate is unavailable, the STMVP candidate is considered as unavailable.
 - g. In one example, whether a spatial neighboring motion candidate and/or temporal neighboring motion candidate is treated as valid or not is based on the reference picture information.
 - i. In one example, only when its reference index in at least one reference picture list is equal to or no greater than K (e.g., $K = 0$).
 - ii. In one example, only when its reference indices in both reference picture list are equal to or no greater than K (e.g., $K = 0$).
 - iii. Alternatively, furthermore, when it is treated as invalid, it is not used to derive the STMVP candidate.
 - iv. Alternatively, furthermore, the STMVP candidate is valid if at least one candidate of the first M spatial merge candidates and one collocated merge candidate are valid.
 - h. In one example, M is set equal to 3 and N is set equal to 1.
 - i. In one example, if the reference indices of the four merge candidates are all valid and are all equal to zero in prediction direction X ($X = 0$ or 1), the

motion vector of the STMVP candidate in prediction direction X (denoted as mvLX) is derived as follows:

$$\text{mvLX} = (\text{mvLX_F} * a + \text{mvLX_S} * b + \text{mvLX_T} * c + \text{mvLX_Col} * d) >> e$$

- (i) In one example, a, b, c, d, and e are set equal to 1, 1, 1, 1, and 2.
- ii. In one example, if reference indices of three of the four merge candidates are valid and are equal to zero in prediction direction X (X = 0 or 1), the motion vector of the STMVP candidate in prediction direction X (denoted as mvLX) is derived as follows:

$$\text{mvLX} = (\text{mvLX_F} * a + \text{mvLX_S} * b + \text{mvLX_Col} * c) >> d \quad \text{or}$$

$$\text{mvLX} = (\text{mvLX_F} * a + \text{mvLX_T} * b + \text{mvLX_Col} * c) >> d \quad \text{or}$$

$$\text{mvLX} = (\text{mvLX_S} * a + \text{mvLX_T} * b + \text{mvLX_Col} * c) >> d$$

- (i) In one example, a, b, c, and d are set equal to 3, 3, 2, and 3.
- (ii) In one example, a, b, c, and d are set equal to 2, 2, 4, and 3.
- (iii) In one example, a, b, c, and d are set equal to 1, 1, 6, and 3.
- iii. In one example, if reference indices of two of the four merge candidates are valid and are equal to zero in prediction direction X (X = 0 or 1), the motion vector of the STMVP candidate in prediction direction X (denoted as mvLX) is derived as follows:

$$\text{mvLX} = (\text{mvLX_F} * a + \text{mvLX_Col} * b) >> c \quad \text{or}$$

$$\text{mvLX} = (\text{mvLX_S} * a + \text{mvLX_Col} * b) >> c \quad \text{or}$$

$$\text{mvLX} = (\text{mvLX_T} * a + \text{mvLX_Col} * b) >> c$$

- (i) In one example, a, b, and c are set equal to 1, 1, and 1
- i. In one example, the STMVP candidate may be pruned with all the previous merge candidates in the merge list.
- j. In one example, the STMVP candidate may not be pruned with other merge candidates.
- k. In one example, the STMVP candidate may be only pruned with the above and the left merge candidates.
- l. In one example, a STMVP candidate refers to one or two specific reference pictures.
 - i. For example, the specific reference picture is the reference picture with reference index equal to 0 in a reference list.
 - ii. For example, the specific reference picture is the reference picture of the M spatial neighboring motion candidates and/or the N temporal

neighboring motion candidates with smallest reference index in a reference list.

Merge list construction process

7. The merge list construction process may include the following candidates which are checked in order.
 - a. 1st set of spatial merge candidates (e.g., derived from B, A, C, D), STMVP, 2nd set of spatial merge candidates (e.g., derived from E), TMVP, HMVP, Pairwise-average merge candidates, zero motion vector merge candidates.
 - b. Spatial merge candidates derived from adjacent blocks (e.g., derived from B, A, C, D, E), TMVP, 1st set of spatial merge candidates derived from non-adjacent blocks (e.g., derived from B₁, A₁, C₁, D₁, E₁), HMVP, Pairwise-average merge candidates, zero motion vector merge candidates.
 - c. Spatial merge candidates derived from adjacent blocks (e.g., derived from B, A, C, D, E), TMVP, spatial merge candidates derived from non-adjacent blocks (e.g., derived from B₁, A₁, C₁, D₁, E₁, B₂, A₂, C₂, D₂, E₂), HMVP, Pairwise-average merge candidates, zero motion vector merge candidates.
 - d. 1st set of spatial merge candidates (e.g., derived from B, A, C, D), STMVP, 2nd set of spatial merge candidates (e.g., derived from E), TMVP, 1st set of spatial merge candidates derived from non-adjacent blocks (e.g., derived from B₁, A₁, C₁, D₁, E₁), HMVP, Pairwise-average merge candidates, zero motion vector merge candidates.
 - e. In above examples, if the corresponding candidate is not available, or invalid, or being identical or similar to existing (added before the corresponding candidate) candidates, the corresponding candidate is not included in the motion candidate list.
8. The above methods may be applied to other kinds of motion candidate list in addition to the merge candidate list.
 - a. Alternatively, the above methods may be applied to the block vector candidate list construction process for IBC-coded blocks. In this case, the check of reference picture index equal to K may be replaced by checking whether the block is coded with IBC mode.

5. Embodiment

Deleted parts are highlighted in grey ~~strikethrough~~, and newly added parts are highlighted in grey.

5.1. Embodiment #1 on MMVD

If the selected MV basis (or base merge candidate) in MMVD mode is a bi-directional MV and the sum of the width and height of the block is smaller than or equal to 12, MVD of prediction direction 0(L0) is set equal to the signaled MVD directly and the MMVD merge candidate is converted to a L0 uni-prediction candidate.

8.5.2.7 Derivation process for merge motion vector difference

Inputs to this process are:

- a luma location (xCb, yCb) of the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture,
- reference indices refIdxL0 and refIdxL1,
- prediction list utilization flags predFlagL0 and predFlagL1.

Outputs of this process are the luma merge motion vector differences in 1/16 fractional-sample accuracy mMvdL0 and mMvdL1.

The variable currPic specifies the current picture.

The luma merge motion vector differences mMvdL0 and mMvdL1 are derived as follows:

- If both predFlagL0 and predFlagL1 are equal to 1 and the sum of the width and height of the current luma coding block is larger than 12, the following applies:

$$\text{currPocDiffL0} = \text{DiffPicOrderCnt}(\text{currPic}, \text{RefPicList}[0][\text{refIdxL0}]) \quad (564)$$

$$\text{currPocDiffL1} = \text{DiffPicOrderCnt}(\text{currPic}, \text{RefPicList}[1][\text{refIdxL1}]) \quad (565)$$

- If currPocDiffL0 is equal to currPocDiffL1, the following applies:

$$\text{mMvdL0}[0] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][0] \quad (566)$$

$$\text{mMvdL0}[1] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][1] \quad (567)$$

$$\text{mMvdL1}[0] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][0] \quad (568)$$

$$\text{mMvdL1}[1] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][1] \quad (569)$$

- Otherwise, if Abs(currPocDiffL0) is greater than or equal to Abs(currPocDiffL1), the following applies:

$$\text{mMvdL0}[0] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][0] \quad (570)$$

$$\text{mMvdL0}[1] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][1] \quad (571)$$

- If RefPicList[0][refIdxL0] is not a long-term reference picture and RefPicList[1][refIdxL1] is not a long-term reference picture, the following applies:

$$\text{td} = \text{Clip3}(-128, 127, \text{currPocDiffL0}) \quad (572)$$

$$tb = \text{Clip3}(-128, 127, \text{currPocDiffL1}) \quad (573)$$

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (574)$$

$$\text{distScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (575)$$

$$\begin{aligned} \text{mMvdL1}[0] = \text{Clip3}(-2^{17}, 2^{17} - 1, (\text{distScaleFactor} * \text{mMvdL0}[0] + \\ 128 - (\text{distScaleFactor} * \text{mMvdL0}[0] \geq 0)) \gg 8) \end{aligned} \quad (576)$$

$$\begin{aligned} \text{mMvdL1}[1] = \text{Clip3}(-2^{17}, 2^{17} - 1, (\text{distScaleFactor} * \text{mMvdL0}[1] + \\ 128 - (\text{distScaleFactor} * \text{mMvdL0}[1] \geq 0)) \gg 8) \end{aligned} \quad (577)$$

– Otherwise, the following applies:

$$\begin{aligned} \text{mMvdL1}[0] = \text{Sign}(\text{currPocDiffL0}) == \text{Sign}(\text{currPocDiffL1}) ? \\ \text{mMvdL0}[0] : -\text{mMvdL0}[0] \end{aligned} \quad (578)$$

$$\begin{aligned} \text{mMvdL1}[1] = \text{Sign}(\text{currPocDiffL0}) == \text{Sign}(\text{currPocDiffL1}) ? \\ \text{mMvdL0}[1] : -\text{mMvdL0}[1] \end{aligned} \quad (579)$$

– Otherwise ($\text{Abs}(\text{currPocDiffL0})$ is less than $\text{Abs}(\text{currPocDiffL1})$), the following applies:

$$\text{mMvdL1}[0] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][0] \quad (580)$$

$$\text{mMvdL1}[1] = \text{MmvdOffset}[\text{xCb}][\text{yCb}][1] \quad (581)$$

– If $\text{RefPicList}[0][\text{refIdxL0}]$ is not a long-term reference picture and $\text{RefPicList}[1][\text{refIdxL1}]$ is not a long-term reference picture, the following applies:

$$td = \text{Clip3}(-128, 127, \text{currPocDiffL1}) \quad (582)$$

$$tb = \text{Clip3}(-128, 127, \text{currPocDiffL0}) \quad (583)$$

$$tx = (16384 + (\text{Abs}(td) \gg 1)) / td \quad (584)$$

$$\text{distScaleFactor} = \text{Clip3}(-4096, 4095, (tb * tx + 32) \gg 6) \quad (585)$$

$$\begin{aligned} \text{mMvdL0}[0] = \text{Clip3}(-2^{17}, 2^{17} - 1, (\text{distScaleFactor} * \text{mMvdL1}[0] + \\ 128 - (\text{distScaleFactor} * \text{mMvdL1}[0] \geq 0)) \gg 8) \end{aligned} \quad (586)$$

$$\begin{aligned} \text{mMvdL0}[1] = \text{Clip3}(-2^{17}, 2^{17} - 1, (\text{distScaleFactor} * \text{mMvdL1}[1] + \\ 128 - (\text{distScaleFactor} * \text{mMvdL1}[1] \geq 0)) \gg 8) \end{aligned} \quad (587)$$

– Otherwise, the following applies:

$$\begin{aligned} \text{mMvdL0}[0] = \text{Sign}(\text{currPocDiffL0}) == \text{Sign}(\text{currPocDiffL1}) ? \\ \text{mMvdL1}[0] : -\text{mMvdL1}[0] \end{aligned} \quad (588)$$

$$\begin{aligned} \text{mMvdL0}[1] = \text{Sign}(\text{currPocDiffL0}) == \text{Sign}(\text{currPocDiffL1}) ? \\ \text{mMvdL1}[1] : -\text{mMvdL1}[1] \end{aligned} \quad (589)$$

- Otherwise (predFlagL0 or predFlagL1 are equal to 1 or the sum of the width and height of the current luma coding block is smaller than or equal to 12), the following applies for X being 0 and 1:

$$mMvdLX[0] = (\text{predFlagLX} = 1) ? \text{MmvdOffset}[xCb][yCb][0] : 0 \quad (590)$$

$$mMvdLX[1] = (\text{predFlagLX} = 1) ? \text{MmvdOffset}[xCb][yCb][1] : 0 \quad (591)$$

[0082] FIG. 4. is a block diagram showing an example video processing system 1900 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 1900. The system 1900 may include input 1902 for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input 1902 may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as Wi-Fi or cellular interfaces.

[0083] The system 1900 may include a coding component 1904 that may implement the various coding or encoding methods described in the present document. The coding component 1904 may reduce the average bitrate of video from the input 1902 to the output of the coding component 1904 to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component 1904 may be either stored, or transmitted via a communication connected, as represented by the component 1906. The stored or communicated bitstream (or coded) representation of the video received at the input 1902 may be used by the component 1908 for generating pixel values or displayable video that is sent to a display interface 1910. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as “coding” operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

[0084] Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on. Examples of storage interfaces include SATA (serial advanced technology attachment), PCI, IDE interface, and the like. The techniques described in the present document may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

[0085] FIG. 5 is a block diagram of a video processing apparatus 3600. The apparatus 3600 may be used to implement one or more of the methods described herein. The apparatus 3600 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 3600 may include one or more processors 3602, one or more memories 3604 and video processing hardware 3606. The processor(s) 3602 may be configured to implement one or more methods described in the present document. The memory (memories) 3604 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 3606 may be used to implement, in hardware circuitry, some techniques described in the present document.

[0086] FIG. 7 is a block diagram that illustrates an example video coding system 100 that may utilize the techniques of this disclosure.

[0087] As shown in FIG. 7, video coding system 100 may include a source device 110 and a destination device 120. Source device 110 generates encoded video data which may be referred to as a video encoding device. Destination device 120 may decode the encoded video data generated by source device 110 which may be referred to as a video decoding device.

[0088] Source device 110 may include a video source 112, a video encoder 114, and an input/output (I/O) interface 116.

[0089] Video source 112 may include a source such as a video capture device, an interface to receive video data from a video content provider, and/or a computer graphics system for generating video data, or a combination of such sources. The video data may comprise one or more pictures. Video encoder 114 encodes the video data from video source 112 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. I/O interface 116 may include a modulator/demodulator (modem) and/or a transmitter. The encoded video data may be transmitted directly to destination device 120 via I/O interface 116 through network 130a. The encoded video data may also be stored onto a storage medium/server 130b for access by destination device 120.

[0090] Destination device 120 may include an I/O interface 126, a video decoder 124, and a display device 122.

[0091] I/O interface 126 may include a receiver and/or a modem. I/O interface 126 may acquire encoded video data from the source device 110 or the storage medium/ server 130b. Video decoder 124 may decode the encoded video data. Display device 122 may display the decoded

video data to a user. Display device 122 may be integrated with the destination device 120, or may be external to destination device 120 which be configured to interface with an external display device.

[0092] Video encoder 114 and video decoder 124 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding(VVM) standard and other current and/or further standards.

[0093] FIG. 8 is a block diagram illustrating an example of video encoder 200, which may be video encoder 114 in the system 100 illustrated in FIG.7.

[0094] Video encoder 200 may be configured to perform any or all of the techniques of this disclosure. In the example of FIG. 8, video encoder 200 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of video encoder 200. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0095] The functional components of video encoder 200 may include a partition unit 201, a predication unit 202 which may include a mode select unit 203, a motion estimation unit 204, a motion compensation unit 205 and an intra prediction unit 206, a residual generation unit 207, a transform unit 208, a quantization unit 209, an inverse quantization unit 210, an inverse transform unit 211, a reconstruction unit 212, a buffer 213, and an entropy encoding unit 214.

[0096] In other examples, video encoder 200 may include more, fewer, or different functional components. In an example, predication unit 202 may include an intra block copy(IBC) unit. The IBC unit may perform predication in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0097] Furthermore, some components, such as motion estimation unit 204 and motion compensation unit 205 may be highly integrated, but are represented in the example of FIG. 8 separately for purposes of explanation.

[0098] Partition unit 201 may partition a picture into one or more video blocks. Video encoder 200 and video decoder 300 may support various video block sizes.

[0099] Mode select unit 203 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra- or inter-coded block to a residual generation unit 207 to generate residual block data and to a reconstruction unit 212 to reconstruct the encoded block for use as a reference picture. In some example, Mode select unit 203 may select a combination of intra and inter predication (CIIP) mode in which the predication is based on an inter predication signal and an intra predication signal. Mode select unit 203 may also select a resolution for a

motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter-prediction.

[00100] To perform inter prediction on a current video block, motion estimation unit 204 may generate motion information for the current video block by comparing one or more reference frames from buffer 213 to the current video block. Motion compensation unit 205 may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from buffer 213 other than the picture associated with the current video block.

[00101] Motion estimation unit 204 and motion compensation unit 205 may perform different operations for a current video block, for example, depending on whether the current video block is in an I slice, a P slice, or a B slice.

[00102] In some examples, motion estimation unit 204 may perform uni-directional prediction for the current video block, and motion estimation unit 204 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. Motion estimation unit 204 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. Motion estimation unit 204 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. Motion compensation unit 205 may generate the predicted video block of the current block based on the reference video block indicated by the motion information of the current video block.

[00103] In other examples, motion estimation unit 204 may perform bi-directional prediction for the current video block, motion estimation unit 204 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. Motion estimation unit 204 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. Motion estimation unit 204 may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. Motion compensation unit 205 may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.

[00104] In some examples, motion estimation unit 204 may output a full set of motion information for decoding processing of a decoder.

[00105] In some examples, motion estimation unit 204 may do not output a full set of motion information for the current video. Rather, motion estimation unit 204 may signal the motion information of the current video block with reference to the motion information of another video block. For example, motion estimation unit 204 may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

[00106] In one example, motion estimation unit 204 may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder 300 that the current video block has the same motion information as the another video block.

[00107] In another example, motion estimation unit 204 may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder 300 may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

[00108] As discussed above, video encoder 200 may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder 200 include advanced motion vector predication (AMVP) and merge mode signaling.

[00109] Intra prediction unit 206 may perform intra prediction on the current video block. When intra prediction unit 206 performs intra prediction on the current video block, intra prediction unit 206 may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

[00110] Residual generation unit 207 may generate residual data for the current video block by subtracting (e.g., indicated by the minus sign) the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

[00111] In other examples, there may be no residual data for the current video block for the current video block, for example in a skip mode, and residual generation unit 207 may not perform the subtracting operation.

[00112] Transform processing unit 208 may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

[00113] After transform processing unit 208 generates a transform coefficient video block associated with the current video block, quantization unit 209 may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[00114] Inverse quantization unit 210 and inverse transform unit 211 may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. Reconstruction unit 212 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the predication unit 202 to produce a reconstructed video block associated with the current block for storage in the buffer 213.

[00115] After reconstruction unit 212 reconstructs the video block, loop filtering operation may be performed reduce video blocking artifacts in the video block.

[00116] Entropy encoding unit 214 may receive data from other functional components of the video encoder 200. When entropy encoding unit 214 receives the data, entropy encoding unit 214 may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[00117] FIG. 9 is a block diagram illustrating an example of video decoder 300 which may be video decoder 114 in the system 100 illustrated in FIG.7.

[00118] The video decoder 300 may be configured to perform any or all of the techniques of this disclosure. In the example of FIG. 8, the video decoder 300 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 300. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[00119] In the example of FIG. 9, video decoder 300 includes an entropy decoding unit 301, a motion compensation unit 302, an intra prediction unit 303, an inverse quantization unit 304, an inverse transformation unit 305, and a reconstruction unit 306 and a buffer 307. Video decoder 300 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 200 (FIG. 8).

[00120] Entropy decoding unit 301 may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). Entropy decoding unit 301 may decode the entropy coded video data, and from the entropy decoded video data, motion compensation unit 302 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. Motion compensation unit 302 may, for example, determine such information by performing the AMVP and merge mode.

[00121] Motion compensation unit 302 may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

[00122] Motion compensation unit 302 may use interpolation filters as used by video encoder 20 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. Motion compensation unit 302 may determine the interpolation filters used by video encoder 200 according to received syntax information and use the interpolation filters to produce predictive blocks.

[00123] Motion compensation unit 302 may use some of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter-encoded block, and other information to decode the encoded video sequence.

[00124] Intra prediction unit 303 may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. Inverse quantization unit 303 inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by entropy decoding unit 301. Inverse transform unit 303 applies an inverse transform.

[00125] Reconstruction unit 306 may sum the residual blocks with the corresponding prediction blocks generated by motion compensation unit 202 or intra-prediction unit 303 to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in buffer 307, which provides reference blocks for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

[00126] A listing of solutions preferred by some embodiments is provided next.

[00127] The following solutions show example embodiments of techniques discussed in the previous section (e.g., item 1).

[00128] 1. A method of video processing (e.g., method 600 depicted in FIG. 6), comprising: determining (602), for a conversion between a video unit of a video and a coded representation of the video, a calculation operation for a motion vector difference (MVD) for use with a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; and performing (604) the conversion based on the determining.

[00129] 2. The method of solution 1, wherein the characteristic of the video unit comprises a dimension or allowed prediction directions for the video unit.

[00130] 3. The method of solutions 1-2, wherein the characteristic of the video unit is that only uni-prediction is allowed, and based on the characteristic, the calculation operation determines a single MVD regardless of a prediction direction of a base merge candidate associated with the MMVD coding tool.

[00131] 4. The method of solution 2, wherein due to the dimension of the video unit satisfying a condition, the calculation operation determines a single MVD regardless of a prediction direction of a base merge candidate associated with the MMVD coding tool.

[00132] The following solutions show example embodiments of techniques discussed in the previous section (e.g., item 2).

[00133] 5. The method of solution 1, wherein the characteristic of the video unit is that a base merge candidate associated with the MMVD coding tool is a bi-directional motion vector, and the conversion comprises using an internal motion vector difference directly for prediction direction X, in case that the video unit satisfies a block size condition, where $X = 0$ or 1 .

[00134] 6. The method of solution 5, wherein the internal motion vector difference is directly used for prediction direction 0.

[00135] 7. The method of solution 5, wherein the block size condition is that a height plus a width of the video unit is less than or equal to N, where N is a positive integer less than total picture pixel width.

[00136] The following solutions show example embodiments of techniques discussed in the previous section (e.g., item 3).

[00137] 8. A method of video processing, comprising:

[00138] performing a conversion between a video unit of a video and a coded representation of the video, wherein the conversion uses a motion vector scaling process during an operation that is dependent of a resolution of the video.

[00139] 9. The method of solution 8, wherein the operation comprises encoding or decoding using a merge with motion vector difference tool.

[00140] 10. The method of solution 8, wherein the operation comprises encoding or decoding using a temporal motion vector prediction process.

[00141] The following solutions show example embodiments of techniques discussed in the previous section (e.g., item 4).

[00142] 11. A method of video processing, comprising:

[00143] performing a conversion between a video unit of a video and a coded representation of the video using two long term reference pictures and a motion vector scaling process during the conversion.

[00144] 12. The method of solution 11, wherein the motion vector scaling process is based on a picture order count of the two long term reference pictures.

[00145] The following solutions show example embodiments of techniques discussed in the previous section (e.g., item 5).

[00146] 13. A method of video processing, comprising:

[00147] generating, for a conversion between a video unit of a video and a coded representation of the video, a merge candidate list in which non-adjacent spatial merge candidates of the video unit are inserted in the merge list; and

[00148] performing the conversion using the merge candidate list.

[00149] 14. The method of solution 13, wherein the non-adjacent spatial merge candidates are inserted into the merge list after history-based merge candidates.

[00150] 15. The method of solution 13, wherein non-adjacent spatial merge candidates are inserted into the merge list after pairwise average merge candidates.

[00151] 16. The method of solution 13, wherein, after the number of the available merge candidates in merge list reaches a pre-defined value after inserting the temporal merge candidate, the non-adjacent spatial merge candidates are not inserted.

[00152] 17. The method of solution 13, wherein the non-adjacent spatial merge candidates are inserted in a defined ordered.

[00153] The following solutions show example embodiments of techniques discussed in the previous section (e.g., item 6).

[00154] 18. A method of video processing, comprising:

[00155] generating, for a conversion between a video unit of a video and a coded representation of the video, a candidate list in which a candidate that is generated by averaging M spatial neighboring candidates and N temporal neighboring candidates, where M and N are positive integers; and

[00156] performing the conversion using the merge candidate list.

[00157] 19. The method of solution 18, wherein $M > 2$.

[00158] 20. The method of solutions 18-19, wherein the M spatial neighboring candidates may be derived from spatial merge candidates includes in a merge list.

[00159] 21. The method of solutions 18-19, wherein the N temporal neighboring candidates may be derived from temporal merge candidates includes in a merge list.

[00160] 22. The method of solution 18, wherein $M = 3$ and $N=1$.

[00161] The following solutions show example embodiments of techniques discussed in the previous section (e.g., item 7).

[00162] 23. A method of video processing, comprising:

[00163] generating, for a conversion between a video unit of a video and a coded representation of the video, a merge list wherein a construction process used for generating the merge list checks a number of candidates in a defined order; and

[00164] performing the conversion using the merge candidate list.

[00165] 24. The method of solution 23, wherein the defined order comprising: a first set of spatial merge candidates, spatio-temporal merge candidates, a second set of merge candidates, temporal motion vector predictor, history based motion vector predictor, a pairwise average merge candidate vector and a zero motion vector merge candidate.

[00166] 25. The method of solution 23, wherein the defined order comprises: spatial merge candidates derived from adjacent blocks (e.g., derived from B, A, C, D, E), TMVP, 1st set of spatial merge candidates derived from non-adjacent blocks (e.g., derived from B1, A1, C1, D1, E1), HMVP, Pairwise-average merge candidates, zero motion vector merge candidates.

[00167] 26. The method of any of solutions 1-25, wherein the video unit comprises a video block or a video coding tree unit or a video transform unit or a video coding unit.

[00168] 27. The method of any of solutions 1-26, wherein the performing the conversion comprising encoding the video to generate the coded representation.

[00169] 28. The method of any of solutions 1-26, wherein the performing the conversion comprises parsing and decoding the coded representation to generate the video.

[00170] 29. A video decoding apparatus comprising a processor configured to implement a method recited in one or more of solutions 1 to 28.

[00171] 30. A video encoding apparatus comprising a processor configured to implement a method recited in one or more of solutions 1 to 28.

[00172] 31. A computer program product having computer code stored thereon, the code, when executed by a processor, causes the processor to implement a method recited in any of solutions 1 to 28.

[00173] 32. A method, apparatus or system described in the present document.

[00174] FIG. 10 shows a flowchart of an example method for video processing. The method includes deriving(1002), for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode

(MMVD) coding tool based on a characteristic of the video unit; and performing (1004) the conversion based on the derived MVD.

[00175] In some examples, the characteristic of the video unit comprises a dimension or shape of the video unit and/or allowed prediction directions for the video unit, and the dimension of the video unit includes parameters of currWidth and currHeight indicating the width and height of the video unit, respectfully.

[00176] In some examples, when the characteristic of the video unit indicates that only uni-prediction is allowed for the video unit, a single MVD is derived from a internal MVD associated with the MMVD coding tool, regardless of a prediction direction associated with a base merge candidate in the MMVD coding tool, wherein the internal MVD is the one derived from signaled syntax elements in the bitstream.

[00177] In some examples, if only prediction from reference picture list X is the allowed prediction direction, which is denoted by ListX, the MVD for the ListX is derived from the internal MVD, X being 0 or 1.

[00178] In some examples, the MVD for the ListX is set equal to the internal MVD.

[00179] In some examples, the MVD for the ListX is set equal to opposite of the internal MVD.

[00180] In some examples, the MVD for prediction from reference picture list Y, which is denoted by ListY, is set to default values, Y being 1 or 0.

[00181] In some examples, when the characteristic of the video unit indicates that the dimension of the video unit satisfying a condition, a single MVD is derived from a internal MVD associated with the MMVD coding tool, regardless of a prediction direction associated with a base merge candidate in the MMVD coding tool, wherein the internal MVD is the one derived from signaled syntax elements in the bitstream.

[00182] In some examples, the condition is that currWidth + currHeight is less than or equal to N, where N is a positive integer.

[00183] In some examples, N=12, or N=32.

[00184] In some examples, the condition is that currWidth < N1 or/and currHeight < N2, wherer N1, N2 are positive integers.

[00185] In some examples, N1 = N2 = 8.

[00186] In some examples, the condition is that currWidth < N3 * currHeight and/or currHeight < N4 * currWidth, where N3, N4 are positive integers.

[00187] In some examples, N3 = N4 = 8.

[00188] In some examples, if the characteristic of the video unit indicates that the dimension or the shape of the video unit satisfying one or more conditions, when a base merge candidate is a

bi-directional MV, an internal MVD is always directly used for prediction direction X without scaling, where $X = 0$ or 1 .

[00189] In some examples, the internal MVD is always directly used for prediction direction 0.

[00190] In some examples, the condition is that $\text{currWidth} + \text{currHeight}$ is less than or equal to N , where N is a positive integer.

[00191] In some examples, $N=12$ or $N=32$.

[00192] In some examples, the condition is that $\text{currWidth} < N3 * \text{currHeight}$ and/or $\text{currHeight} < N4 * \text{currWidth}$, where $N3$, $N4$ are positive integers.

[00193] In some examples, $N3 = N4 = 8$.

[00194] In some examples, opposite of the internal MVD is used for prediction direction X without scaling, where $X = 0$ or 1 .

[00195] FIG. 11 shows a flowchart of an example method for video processing. The method includes deriving (1102), for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) using a motion vector (MV) scaling process, wherein the MV scaling process is dependent of a resolution of the video; and performing (1104) the conversion based on the derived MVD.

[00196] In some examples, the MVD is used in a merge with motion vector difference mode (MMVD) coding tool or a temporal motion vector prediction (TMVP) coding tool.

[00197] FIG. 12 shows a flowchart of an example method for video processing. The method includes deriving (1202), for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) using a motion vector (MV) scaling process, wherein the MV scaling process uses two long term reference pictures; and performing (1204) the conversion based on the derived MVD.

[00198] In some examples, the video unit includes at least one of a coding unit (CU), a prediction unit (PU) or a block of the video.

[00199] In some examples, the conversion includes encoding the video unit of video into the bitstream.

[00200] In some examples, the conversion includes decoding the video unit of video from the bitstream.

[00201] FIG. 13 shows a flowchart of an example method for storing bitstream of a video. The method includes deriving (1302), for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; generating (1304) the

bitstream from the video unit based on the derived MVD; and storing (1306) the bitstream in a non-transitory computer-readable recording medium.

[00202] FIG. 14 shows a flowchart of an example method for video processing. The method includes constructing (1402), for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein non-adjacent spatial merge candidates associated with the current block are inserted into the merge candidate list; and performing (1404) the conversion based on the merge candidate list.

[00203] In some examples, the non-adjacent spatial merge candidates are inserted into the merge list after history-based merge candidates.

[00204] In some examples, the non-adjacent spatial merge candidates are inserted into the merge list after pairwise average merge candidates.

[00205] In some examples, if the number of available merge candidates in the merge candidate list reaches a pre-defined value after inserting temporal merge candidate, the non-adjacent spatial merge candidates are not inserted.

[00206] In some examples, when inserting the non-adjacent spatial merge candidates, if the number of available merge candidates in the merge candidate list reaches a pre-defined value the inserting, process is terminated.

[00207] In some examples, the pre-defined value is equal to $\text{maxNumMergeCand} - N$, wherein maxNumMergeCand denotes size of the merge candidate list, N is a positive integer.

[00208] In some examples, N is set equal to 2, 3 or 4.

[00209] In some examples, a maximum search round during construction of the merge candidate list is set equal to 1 or 2.

[00210] In some examples, for each search round, the non-adjacent spatial merge candidates are inserted in a predefined inserting order, and the non-adjacent spatial merge candidates include a candidate A_i derived from a non-adjacent left block of the current block, a candidate B_i derived from a non-adjacent above block of the current block, a candidate C_i derived from a non-adjacent above right block of the current block, a candidate D_i derived from a non-adjacent left bottom block n of the current block, and a candidate E_i derived from a non-adjacent above left block of the current block left, where i is the search round.

[00211] In some examples, the inserting order is A_i, B_i, C_i, D_i , and E_i , the inserting order is B_i, A_i, C_i, D_i , and E_i , the inserting order is B_i, C_i, A_i, D_i , and E_i , or the inserting order is A_i, D_i, B_i, C_i , and E_i .

[00212] In some examples, all spatial and temporal merge candidates perform full pruning process with all previous merge candidates in the merge candidate list, and pruning process of history-based merge candidates and pairwise average candidates is not changed.

[00213] In some examples, all spatial, temporal, history-based, and pairwise average merge candidates perform full pruning process with all previous merge candidates in the merge candidate list.

[00214] In some examples, for the non-adjacent spatial merge candidates, A_i performs pruning with A_{i-1} , B_i performs pruning process with A_i , C_i performs pruning with B_i , D_i performs pruning with A_i , E_i performs pruning with A_i and B_i , and the pruning process of temporal, history-based, and pairwise average candidates is not changed.

[00215] In some examples, a maximum number of pruning that is allowed during construction of the merge candidate list, which is denoted as $MaxPruningNum$, depends on size of the merge candidate list, which is denoted as $maxNumMergeCand$.

[00216] In some examples, $MaxPruningNum$ is set equal to $maxNumMergeCand - M$, where M is an integer.

[00217] In some examples, $M=2$.

[00218] In some examples, $MaxPruningNum$ is set equal to $maxNumMergeCand * M$, where M is an integer.

[00219] In some examples, $M=2$.

[00220] In some examples, a maximum number of pruning that is allowed during construction of the merge candidate list, which is denoted as $MaxPruningNum$, is independent of size of the merge candidate list, which is denoted as $maxNumMergeCand$.

[00221] In some examples, $MaxPruningNum$ is set equal to 30 or 35.

[00222] In some examples, positions of the non-adjacent spatial merge candidates are constrained to be within a predefined area.

[00223] In some examples, the predefined area contains current coding tree unit (CTU) row and four sample rows above the current CTU row.

[00224] In some examples, the predefined area contains current CTU column and four left sample columns of the current CTU column.

[00225] In some examples, the predefined area contains current CTU column and left CTU column of the current CTU column.

[00226] In some examples, positions of the non-adjacent spatial merge candidates have no constrain in horizontal direction.

[00227] In some examples, the non-adjacent spatial merge candidate is allowed to be used as base merge candidate for a merge with motion vector difference mode (MMVD) coding tool.

[00228] In some examples, the non-adjacent spatial merge candidate is not allowed to be used as base merge candidate for a merge with motion vector difference mode (MMVD) coding tool.

[00229] In some examples, the non-adjacent spatial merge candidate is allowed to be used to generate inter-intra prediction.

[00230] In some examples, the non-adjacent spatial merge candidate is not allowed to be used to generate inter-intra prediction.

[00231] In some examples, the non-adjacent spatial merge candidate is allowed to be used to generate geometry (GEO) partition and/or triangular partition merge candidate.

[00232] In some examples, the non-adjacent spatial merge candidate is not allowed to be used to generate geometry (GEO) partition and/or triangular partition merge candidate.

[00233] In some examples, the non-adjacent spatial merge candidate is allowed to be used to generate an affine merge candidate.

[00234] In some examples, the non-adjacent spatial merge candidate is allowed to be used to generate an Advanced Motion Vector Prediction (AMVP) candidate.

[00235] FIG. 15 shows a flowchart of an example method for video processing. The method includes constructing (1502), for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein the construction process of the merge candidate list checks a number of different kinds of candidates in a defined order; and performing (1504) the conversion based on the merge candidate list.

[00236] In some examples, the defined order comprising: a first set of spatial merge candidates, spatial temporal motion vector predictor (STMVP) candidates, a second set of spatial merge candidates, temporal motion vector predictor (TMVP) candidates, history-based motion vector predictor (HMVP) candidates, pairwise average merge candidates and zero motion vector merge candidates.

[00237] In some examples, the first set of spatial merge candidates include a candidate B derived from an above block of the current block, a candidate A derived from a left block of the current block, a candidate C derived from an above right block of the current block and a fourth candidate D derived from a left bottom block of the current block, and the second set of spatial merge candidates include a candidate E derived from an above left block of the current block.

[00238] In some examples, the defined order comprising: spatial merge candidates derived from adjacent blocks of the current block, TMVP candidates, a first set of spatial merge candidates

derived from non-adjacent blocks of the current block, HMVP candidates, pairwise average merge candidates and zero motion vector merge candidates.

[00239] In some examples, the spatial merge candidates derived from adjacent blocks of the current block include a candidate B derived from an above block of the current block, a candidate A derived from a left block of the current block, a candidate C derived from an above right block of the current block and a fourth candidate D derived from a left bottom block of the current block, and a candidate E derived from an above left block of the current block, and the first set of spatial merge candidates derived from non-adjacent blocks of the current block include a candidate B1 derived from a non-adjacent above block of the current block, a candidate A1 derived from a non-adjacent left block of the current block, a candidate C1 derived from a non-adjacent above right block of the current block, a candidate D1 derived from a non-adjacent left bottom block n of the current block, and a candidate E1 derived from a non-adjacent above left block of the current block left, which are derived in a first search round.

[00240] In some examples, the defined order comprising: spatial merge candidates derived from adjacent blocks of the current block, TMVP candidates, spatial merge candidates derived from non-adjacent blocks of the current block, HMVP candidates, pairwise average merge candidates and zero motion vector merge candidates.

[00241] In some examples, the spatial merge candidates derived from adjacent blocks of the current block include a candidate B derived from an above block of the current block, a candidate A derived from a left block of the current block, a candidate C derived from an above right block of the current block and a fourth candidate D derived from a left bottom block of the current block, and a candidate E derived from an above left block of the current block, and the spatial merge candidates derived from non-adjacent blocks of the current block include: a candidate B1 derived from a non-adjacent above block of the current block, a candidate A1 derived from a non-adjacent left block of the current block, a candidate C1 derived from a non-adjacent above right block of the current block, a candidate D1 derived from a non-adjacent left bottom block n of the current block, and a candidate E1 derived from a non-adjacent above left block of the current block left, which are derived in a first search round, and a candidate B2 derived from a non-adjacent above block of the current block, a candidate A2 derived from a non-adjacent left block of the current block, a candidate C2 derived from a non-adjacent above right block of the current block, a candidate D2 derived from a non-adjacent left bottom block n of the current block, and a candidate E2 derived from a non-adjacent above left block of the current block left, which are derived in a second search round.

[00242] In some examples, the defined order comprising: a first set of spatial merge candidates, STMVP candidates, a second set of spatial merge candidates, temporal motion vector predictor (TMVP) candidates, a first set of spatial merge candidates derived from non-adjacent blocks of the current block, HMVP candidates, pairwise average merge candidates and zero motion vector merge candidates.

[00243] In some examples, the first set of spatial merge candidates include a candidate B derived from an above block of the current block, a candidate A derived from a left block of the current block, a candidate C derived from an above right block of the current block and a fourth candidate D derived from a left bottom block of the current block, and the first set of spatial merge candidates derived from non-adjacent blocks of the current block include a candidate B1 derived from a non-adjacent above block of the current block, a candidate A1 derived from a non-adjacent left block of the current block, a candidate C1 derived from a non-adjacent above right block of the current block, a candidate D1 derived from a non-adjacent left bottom block n of the current block, and a candidate E1 derived from a non-adjacent above left block of the current block left, which are derived in a first search round.

[00244] In some examples, if a corresponding candidate is not available, or invalid, or being identical or similar to existing candidates in the merge candidate list, the corresponding candidate is not included in the merge candidate list.

[00245] In some examples, the conversion includes encoding the video unit of video into the bitstream.

[00246] In some examples, the conversion includes decoding the video unit of video from the bitstream.

[00247] FIG. 16 shows a flowchart of an example method for storing bitstream of a video. The method includes constructing (1602), for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein non-adjacent spatial merge candidates associated with the current block are inserted into the merge candidate list; and generating (1604) the bitstream from the video unit based on the merge candidate list; and storing (1606) the bitstream in a non-transitory computer-readable recording medium.

[00248] FIG. 17 shows a flowchart of an example method for video processing. The method includes constructing (1702), for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein a spatial-temporal motion vector prediction (STMVP) candidate associated with the current block are added to the merge candidate list, and the STMVP candidate is derived as an averaging candidate of M

spatial neighboring motion candidates and/or N temporal neighboring motion candidates, M and N being positive integers; and performing (1704) the conversion based on the merge candidate list.

[00249] In some examples, $M > 2$.

[00250] In some examples, the spatial neighboring motion candidates are derived from other neighboring blocks different from or same as those utilized during the construction process of the merge candidate list.

[00251] In some examples, the spatial neighboring motion candidates are selected from spatial merge candidates included in the merge candidate list.

[00252] In some examples, the spatial neighboring motion candidates are selected from the first or last M spatial merge candidates included in the merge candidate list before adding a STMVP candidate.

[00253] In some examples, the spatial neighboring motion candidates are selected from the first or last M merge candidates included in the merge candidate list before adding a STMVP candidate.

[00254] In some examples, the temporal neighboring motion candidates are selected from temporal merge candidates included in the merge candidate list.

[00255] In some examples, if the temporal merge candidate is unavailable, the STMVP candidate is considered as unavailable.

[00256] In some examples, whether a spatial neighboring motion candidate and/or temporal neighboring motion candidate is treated as valid or not is based on reference picture information associated with the current block.

[00257] In some examples, only when its reference index in at least one reference picture list is equal to or no greater than K, it is treated as valid, where K is an integer.

[00258] In some examples, only when its reference indices in both reference picture lists are equal to or no greater than K, it is treated as valid, where K is an integer.

[00259] In some examples, $K = 0$.

[00260] In some examples, when it is treated as invalid, it is not used to derive the STMVP candidate.

[00261] In some examples, if at least one candidate of the first M spatial merge candidates and one collocated merge candidate are valid, the STMVP candidate is valid.

[00262] In some examples, $M = 3$ and $N = 1$, and the STMVP candidate is derived as an averaging candidate of four merge candidates.

[00263] In some examples, if reference indices of the four merge candidates are all valid and are all equal to zero in prediction direction X, X being 0 or 1, motion vector of the STMVP candidate in prediction direction X, which is denoted as mvLX is derived as follows:

$$\text{mvLX} = (\text{mvLX_F} * a + \text{mvLX_S} * b + \text{mvLX_T} * c + \text{mvLX_Col} * d) >> e,$$

where a, b, c, d and e are integers.

[00264] In some examples, a, b, c, d, and e are set equal to 1, 1, 1, 1, and 2.

[00265] In some examples, if reference indices of three of the four merge candidates are valid and are equal to zero in prediction direction X, X being= 0 or 1, motion vector of the STMVP candidate in prediction direction X, which is denoted as mvLX, is derived as follows:

$$\text{mvLX} = (\text{mvLX_F} * a + \text{mvLX_S} * b + \text{mvLX_Col} * c) >> d; \text{ or}$$

$$\text{mvLX} = (\text{mvLX_F} * a + \text{mvLX_T} * b + \text{mvLX_Col} * c) >> d; \text{ or}$$

$$\text{mvLX} = (\text{mvLX_S} * a + \text{mvLX_T} * b + \text{mvLX_Col} * c) >> d,$$

where a, b, c and d are integers.

[00266] In some examples, a, b, c, and d are set equal to 3, 3, 2, and 3, or a, b, c, and d are set equal to 2, 2, 4, and 3, or a, b, c, and d are set equal to 1, 1, 6, and 3.

[00267] In some examples, if reference indices of two of the four merge candidates are valid and are equal to zero in prediction direction X, X being= 0 or 1, motion vector of the STMVP candidate in prediction direction X, which is denoted as mvLX, is derived as follows:

$$\text{mvLX} = (\text{mvLX_F} * a + \text{mvLX_Col} * b) >> c; \text{ or}$$

$$\text{mvLX} = (\text{mvLX_S} * a + \text{mvLX_Col} * b) >> c; \text{ or}$$

$$\text{mvLX} = (\text{mvLX_T} * a + \text{mvLX_Col} * b) >> c,$$

where a, b and c are integers.

[00268] In some examples, a, b, and c are set equal to 1, 1, and 1.

[00269] In some examples, the STMVP candidate is pruned with all the previous merge candidates in the merge candidate list.

[00270] In some examples, the STMVP candidate is not pruned with other merge candidates.

[00271] In some examples, the STMVP candidate is only pruned with above and left merge candidates.

[00272] In some examples, the STMVP candidate refers to one or two specific reference pictures.

[00273] In some examples, the specific reference picture is the reference picture with reference index equal to 0 in a reference list.

[00274] In some examples, the specific reference picture is the reference picture of the M spatial neighboring motion candidates and/or the N temporal neighboring motion candidates with smallest reference index in a reference list.

[00275] In some examples, the conversion includes encoding the video unit of video into the bitstream.

[00276] In some examples, the conversion includes decoding the video unit of video from the bitstream.

[00277] FIG. 18 shows a flowchart of an example method for storing bitstream of a video. The method includes constructing (1802), for a conversion between a current block of a video and a bitstream representation of the video, a merge candidate list for the current block, wherein a spatial-temporal motion vector prediction (STMVP) candidate associated with the current block are added to the merge candidate list, and the STMVP candidate is derived as an averaging candidate of M spatial neighboring motion candidates and/or N temporal neighboring motion candidates, M and N being positive integers; generating (1804) the bitstream from the video unit based on on the merge candidate list; and storing (1806) the bitstream in a non-transitory computer-readable recording medium.

[00278] 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.

[00279] 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.

[00280] 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.

[00281] 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).

[00282] 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.

[00283] 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.

[00284] 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.

[00285] 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:
 - deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; and
 - performing the conversion based on the derived MVD.
2. The method of claim 1, wherein the characteristic of the video unit comprises a dimension or shape of the video unit and/or allowed prediction directions for the video unit, and the dimension of the video unit includes parameters of currWidth and currHeight indicating the width and height of the video unit, respectively.
3. The method of claim 2, wherein when the characteristic of the video unit indicates that only uni-prediction is allowed for the video unit, a single MVD is derived from a internal MVD associated with the MMVD coding tool, regardless of a prediction direction associated with a base merge candidate in the MMVD coding tool, wherein the internal MVD is the one derived from signaled syntax elements in the bitstream.
4. The method of claim 3, wherein if only prediction from reference picture list X is the allowed prediction direction, which is denoted by ListX, the MVD for the ListX is derived from the internal MVD, X being 0 or 1.
5. The method of claim 4, wherein the MVD for the ListX is set equal to the internal MVD.
6. The method of claim 4, wherein the MVD for the ListX is set equal to opposite of the internal MVD.
7. The method of claim 4, wherein the MVD for prediction from reference picture list Y, which is denoted by ListY, is set to default values, Y being 1 or 0.
8. The method of claim 2, wherein when the characteristic of the video unit indicates that the dimension of the video unit satisfying a condition, a single MVD is derived from a internal MVD

associated with the MMVD coding tool, regardless of a prediction direction associated with a base merge candidate in the MMVD coding tool, wherein the internal MVD is the one derived from signaled syntax elements in the bitstream.

9. The method of claim 8, wherein the condition is that $\text{currWidth} + \text{currHeight}$ is less than or equal to N , where N is a positive integer.

10. The method of claim 9, wherein $N=12$, or $N=32$.

11. The method of claim 8, wherein the condition is that $\text{currWidth} < N1$ or/and $\text{currHeight} < N2$, where $N1$, $N2$ are positive integers.

12. The method of claim 11, wherein $N1 = N2 = 8$.

13. The method of claim 8, wherein the condition is that $\text{currWidth} < N3 * \text{currHeight}$ and/or $\text{currHeight} < N4 * \text{currWidth}$, where $N3$, $N4$ are positive integers.

14. The method of claim 13, wherein $N3 = N4 = 8$.

15. The method of claim 2, wherein if the characteristic of the video unit indicates that the dimension or the shape of the video unit satisfying one or more conditions, when a base merge candidate is a bi-directional MV, an internal MVD is always directly used for prediction direction X without scaling, where $X = 0$ or 1 .

16. The method of claim 15, wherein the internal MVD is always directly used for prediction direction 0 .

17. The method of claim 15, wherein the condition is that $\text{currWidth} + \text{currHeight}$ is less than or equal to N , where N is a positive integer.

18. The method of claim 17, wherein $N=12$ or $N=32$.

19. The method of claim 15, wherein the condition is that $\text{currWidth} < N3 * \text{currHeight}$ and/or $\text{currHeight} < N4 * \text{currWidth}$, where $N3$, $N4$ are positive integers.

20. The method of claim 19, wherein $N3 = N4 = 8$.
21. The method of claim 15, wherein opposite of the internal MVD is used for prediction direction X without scaling, where $X = 0$ or 1 .
22. A method for video processing, comprising:
deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) using a motion vector (MV) scaling process, wherein the MV scaling process is dependent of a resolution of the video; and
performing the conversion based on the derived MVD.
23. The method of claim 22, wherein the MVD is used in a merge with motion vector difference mode (MMVD) coding tool or a temporal motion vector prediction (TMVP) coding tool.
24. A method for video processing, comprising:
deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) using a motion vector (MV) scaling process, wherein the MV scaling process uses two long term reference pictures; and
performing the conversion based on the derived MVD.
25. The method of any of claims 1-24, wherein the video unit includes at least one of a coding unit (CU), a prediction unit (PU) or a block of the video.
26. The method of any of claims 1-25, wherein the conversion includes encoding the video unit of video into the bitstream .
27. The method of anyone of claims 1-25, wherein the conversion includes decoding the video unit of video from the bitstream.
28. An apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to:

deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; and
performing the conversion based on the derived MVD.

29. A non-transitory computer readable media storing instructions that cause a processor to:
deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; and
performing the conversion based on the derived MVD.

30. A non-transitory computer readable media storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises:

deriving, for a conversion between a video unit of the video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit; and
generating the bitstream from the video unit based on the derived MVD.

31. A method for storing bitstream of a video, comprising:

deriving, for a conversion between a video unit of a video and a bitstream of the video, a motion vector difference (MVD) used in a merge with motion vector difference mode (MMVD) coding tool based on a characteristic of the video unit;
generating the bitstream from the video unit based on the derived MVD; and
storing the bitstream in a non-transitory computer-readable recording medium.

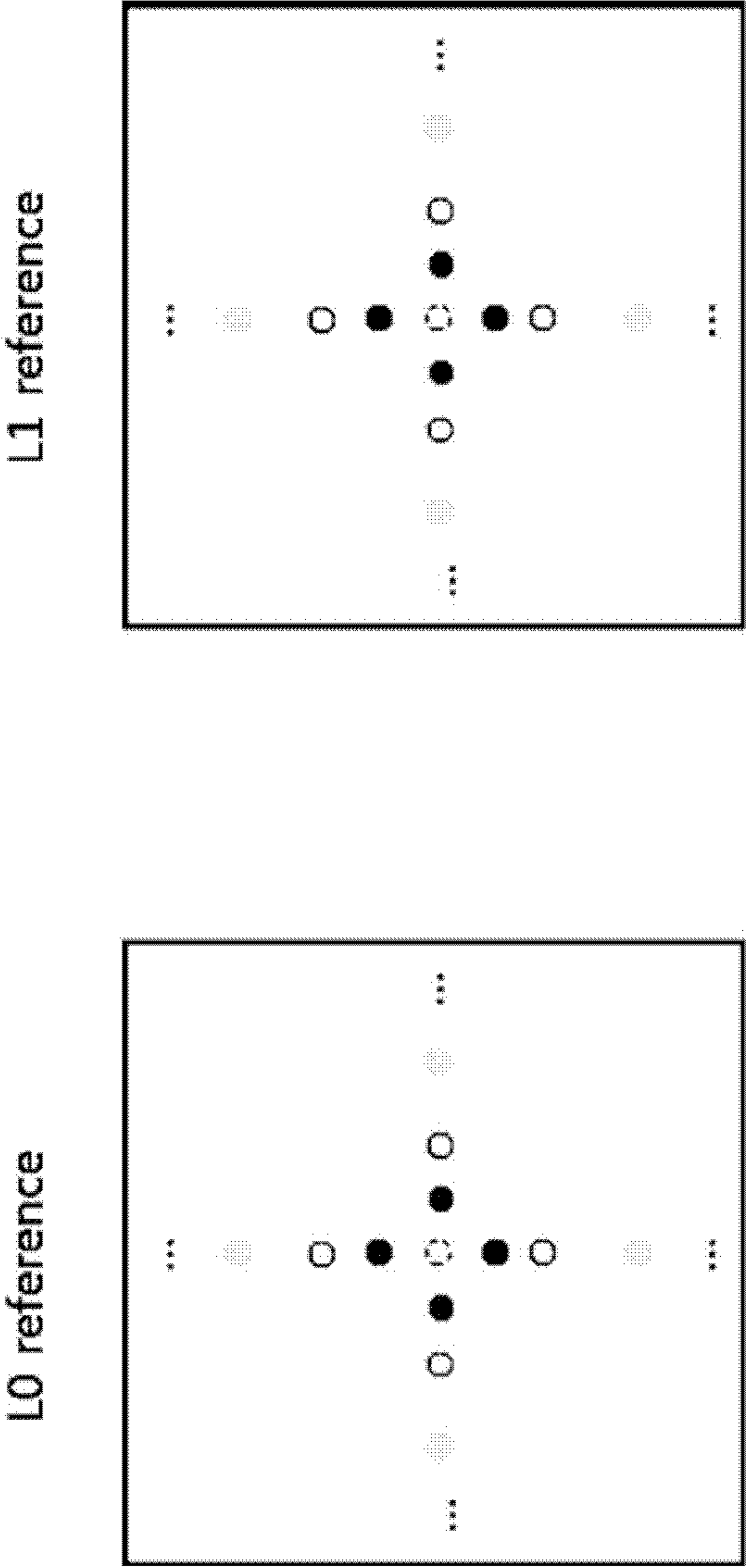


FIG. 1

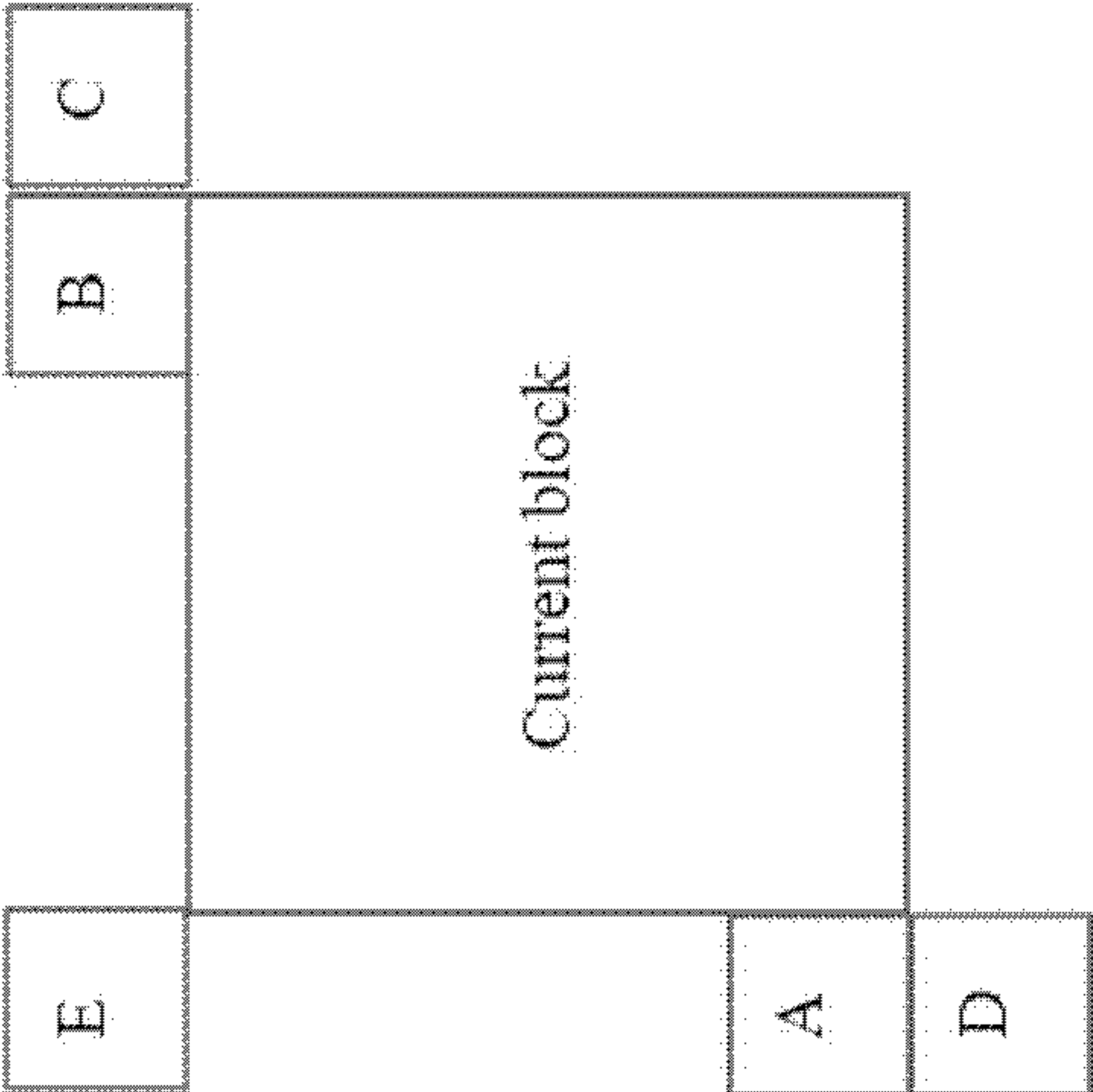


FIG. 2

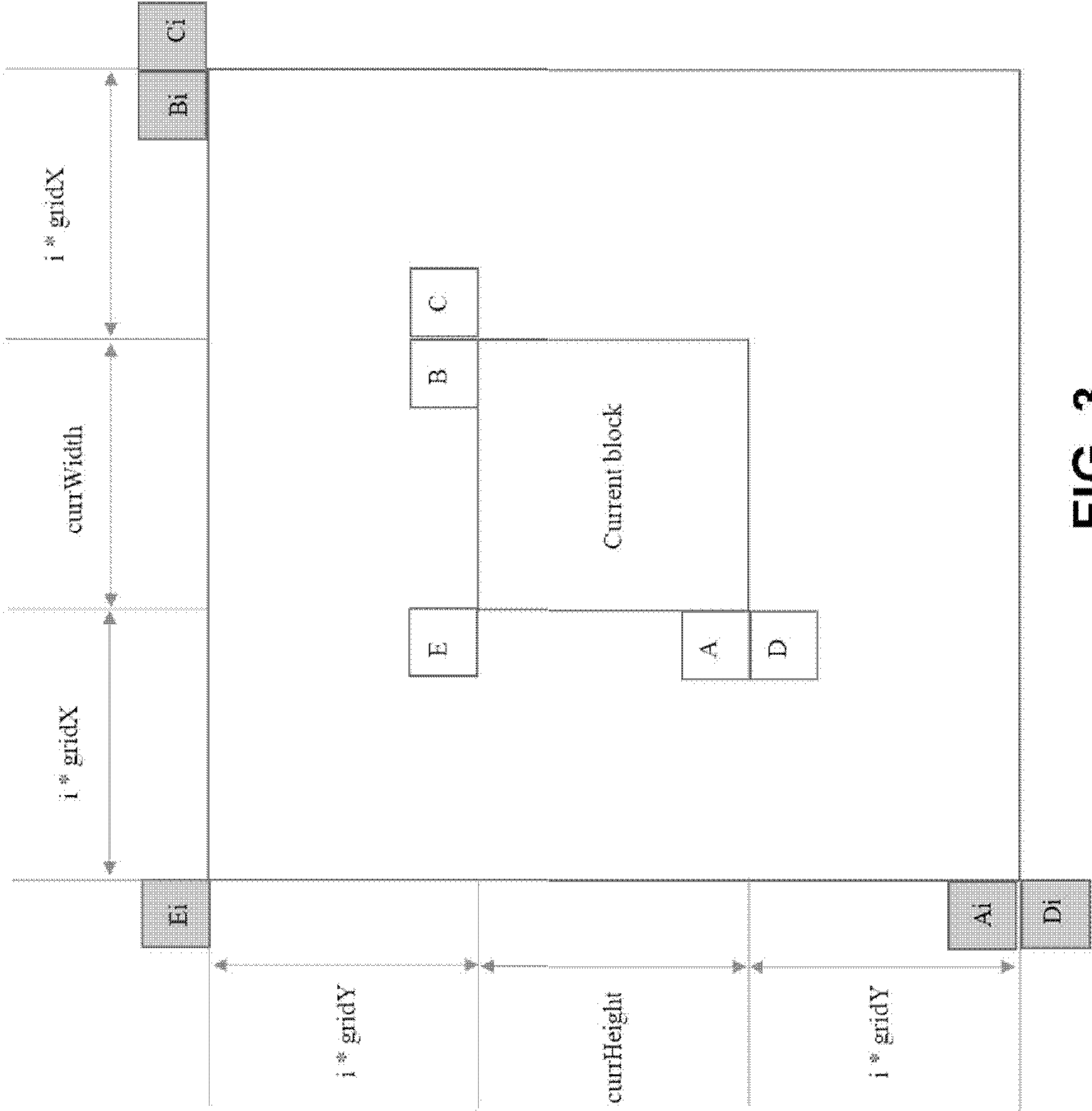


FIG. 3

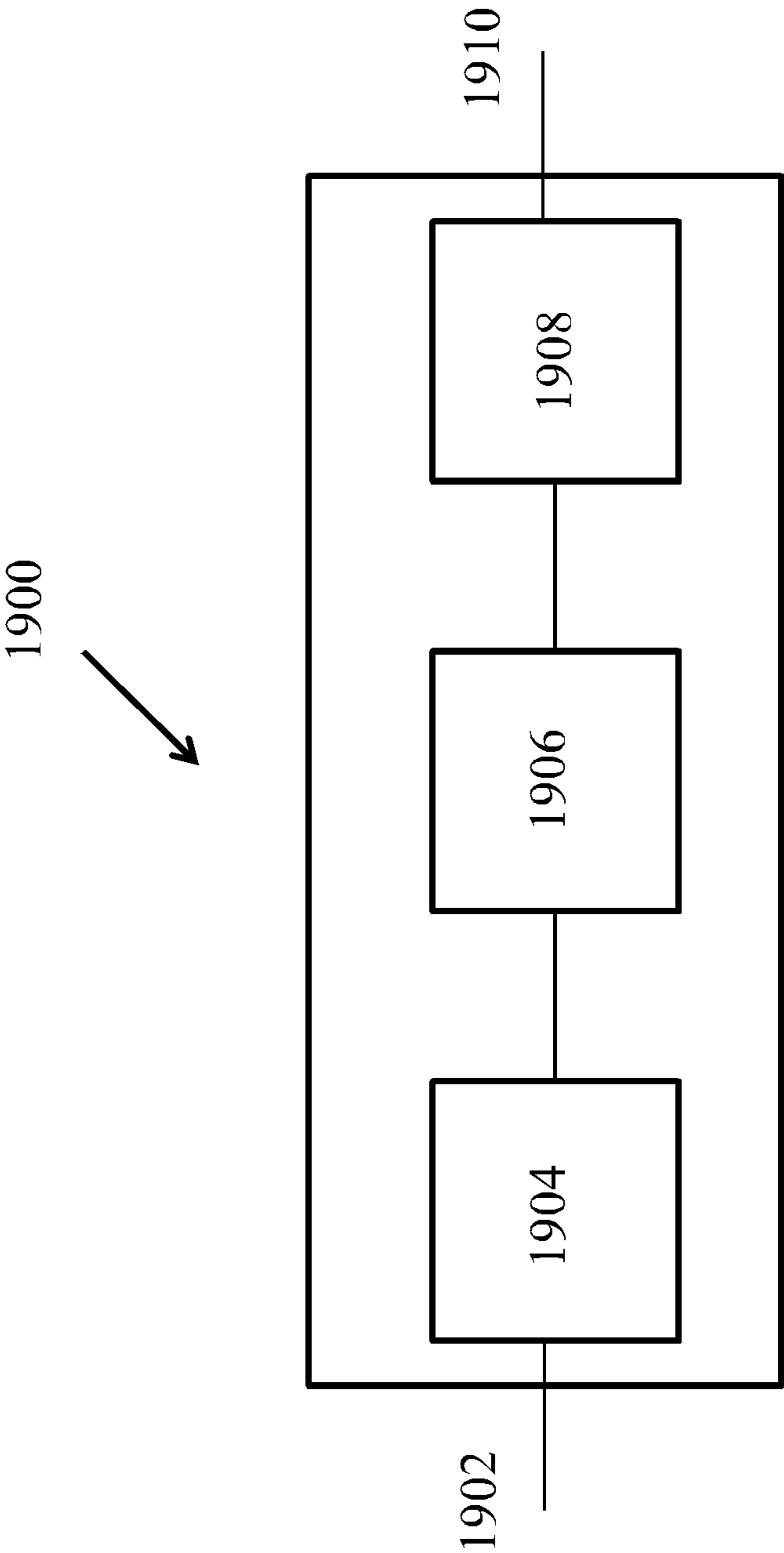


FIG. 4

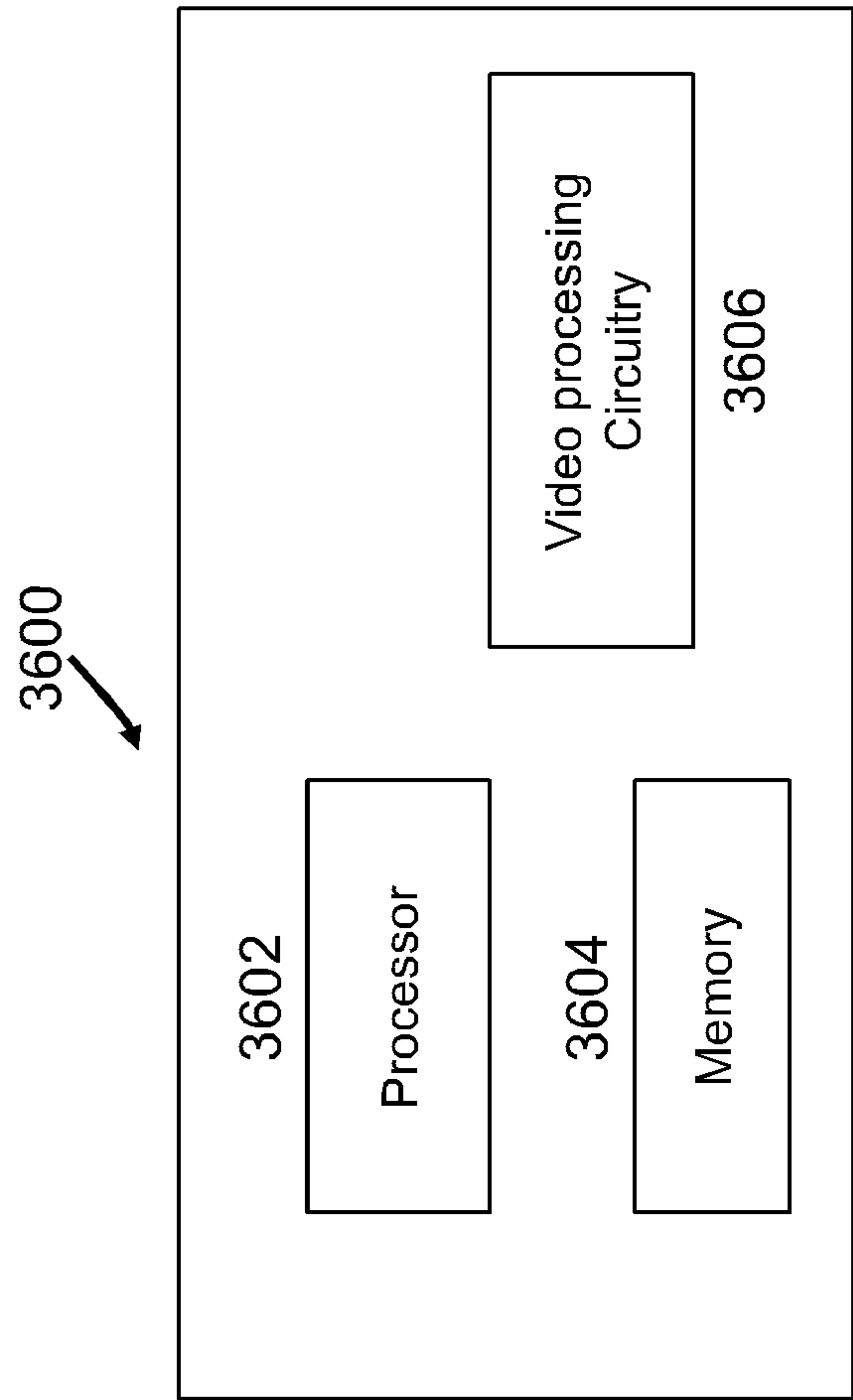


FIG. 5

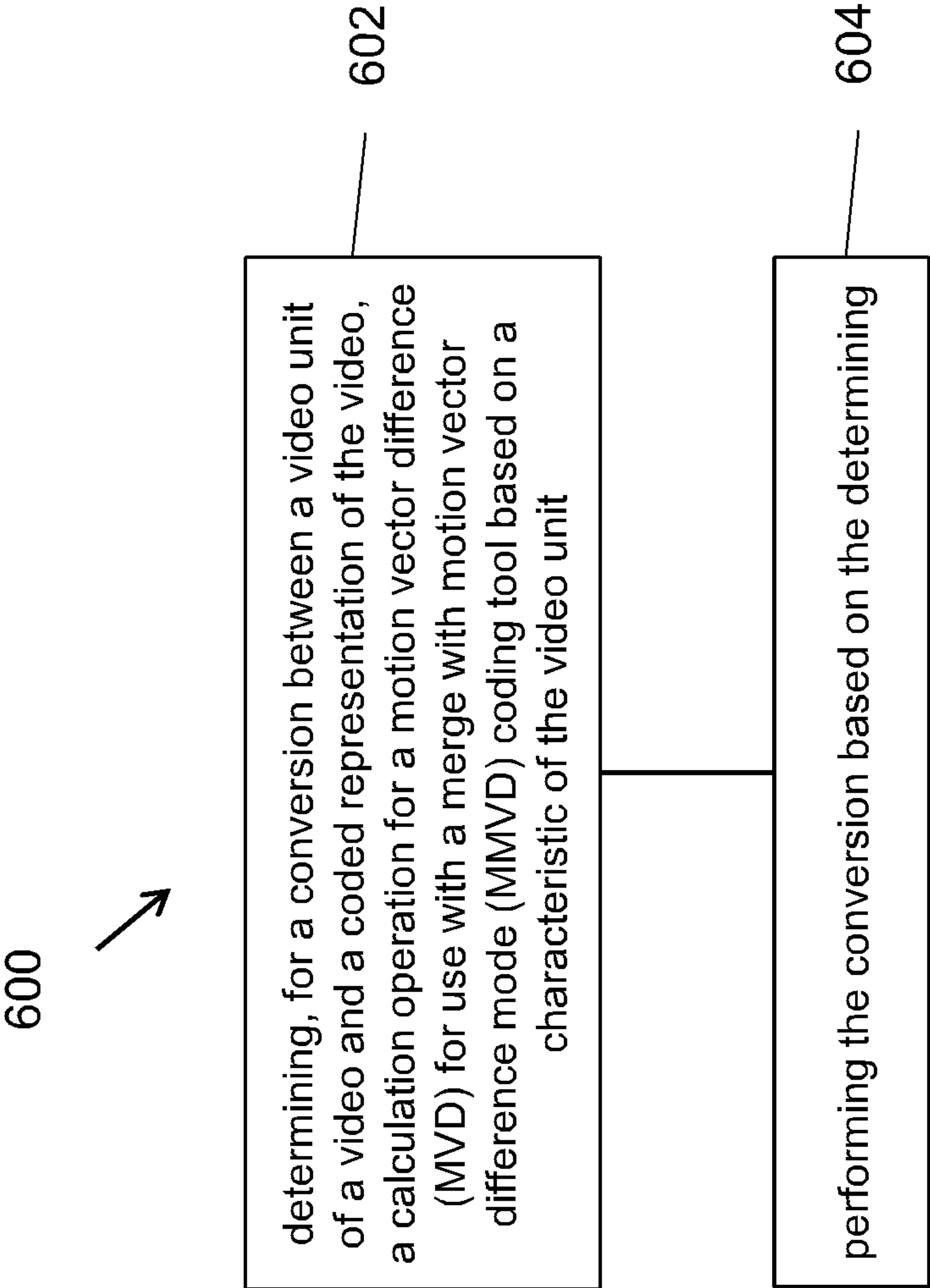


FIG. 6

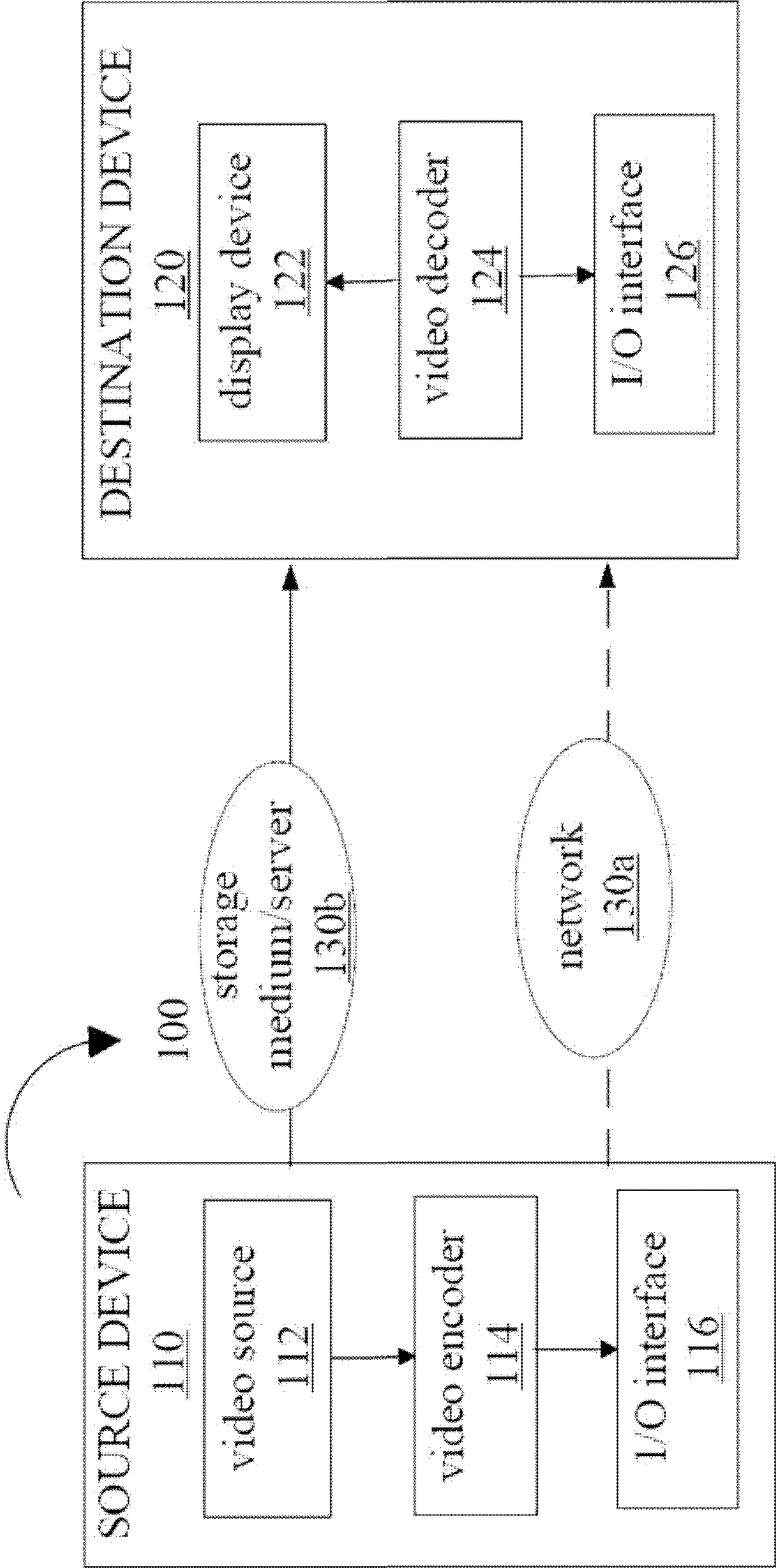


FIG. 7

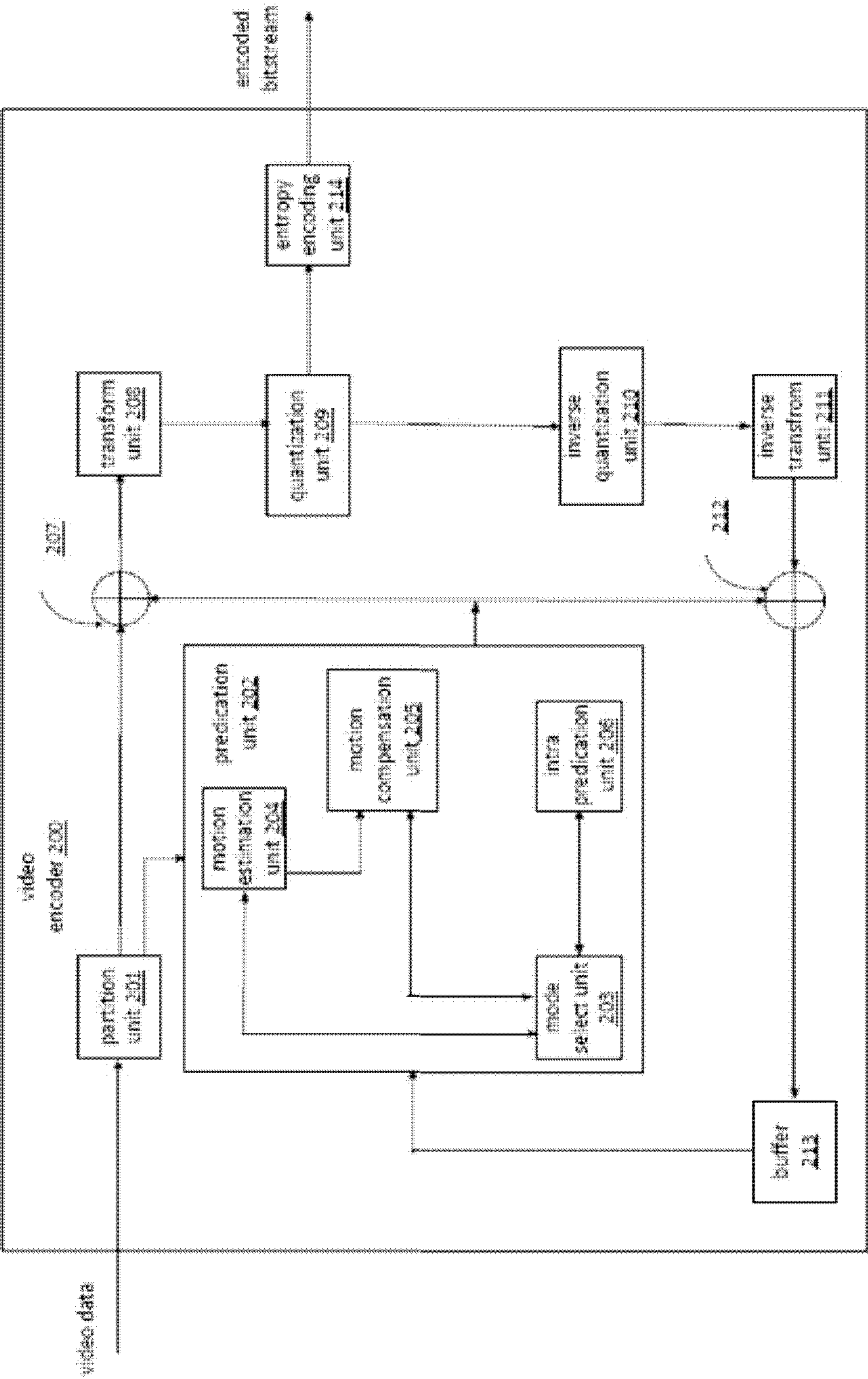


FIG. 8

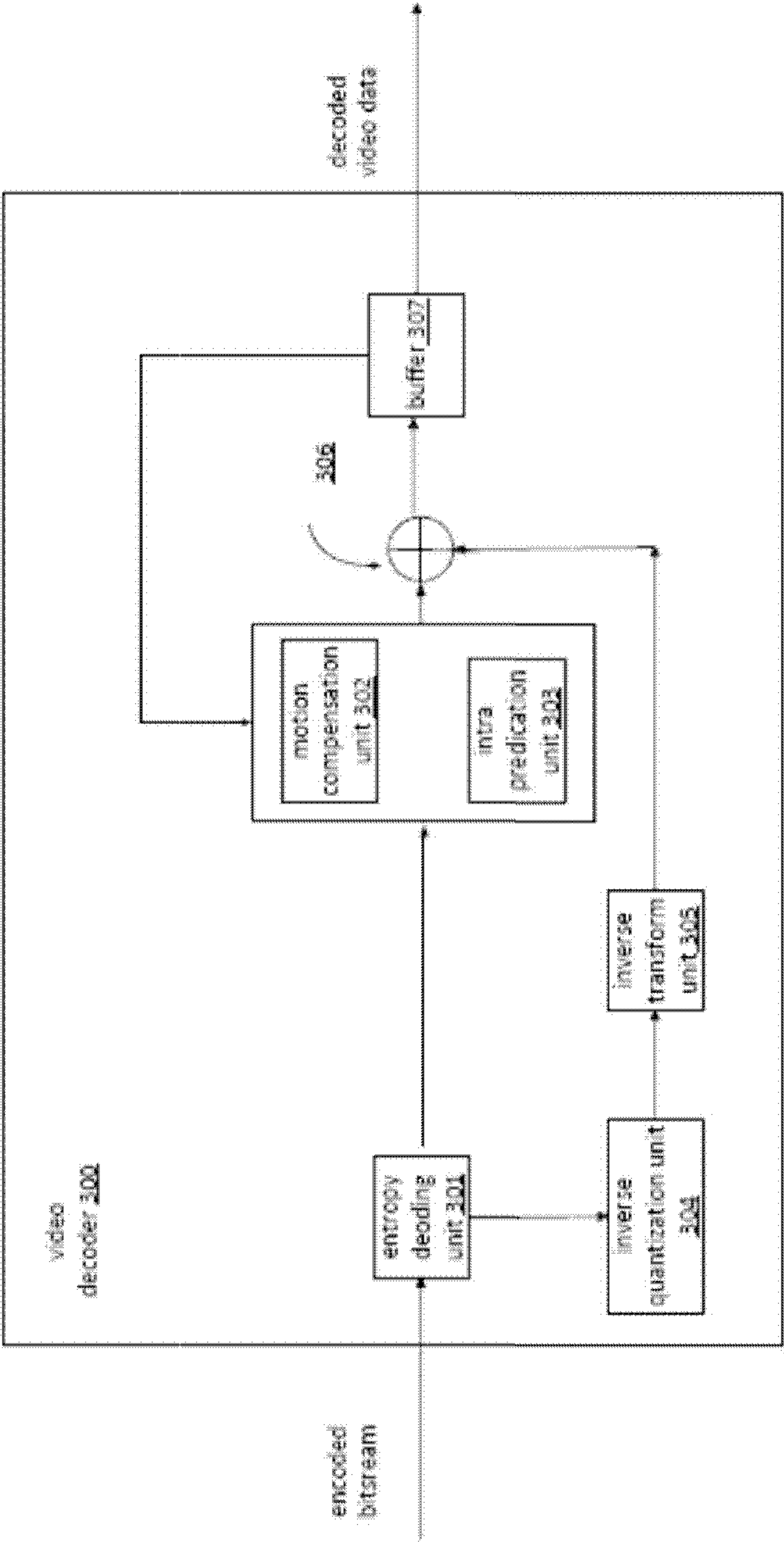


FIG. 9

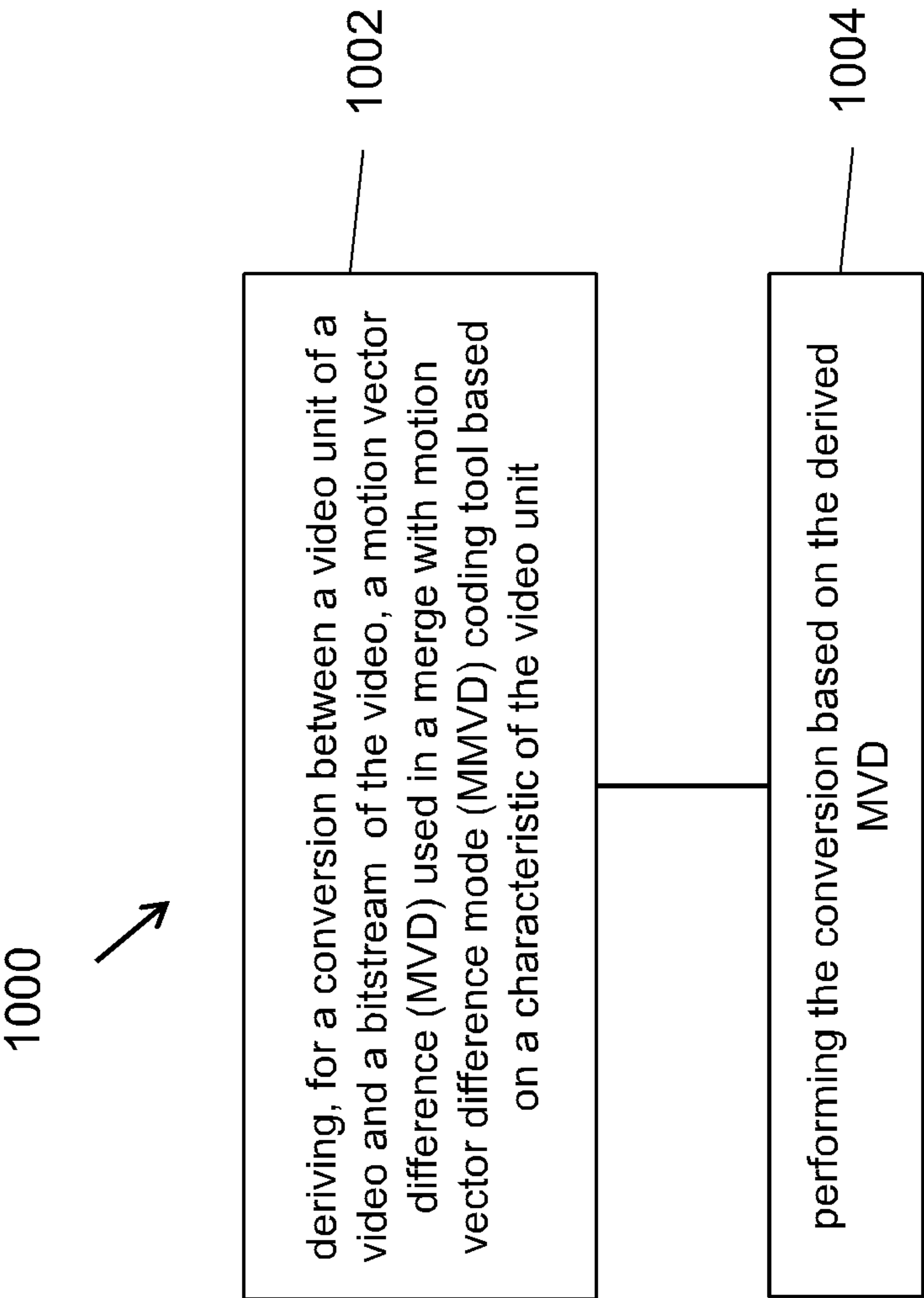


FIG. 10

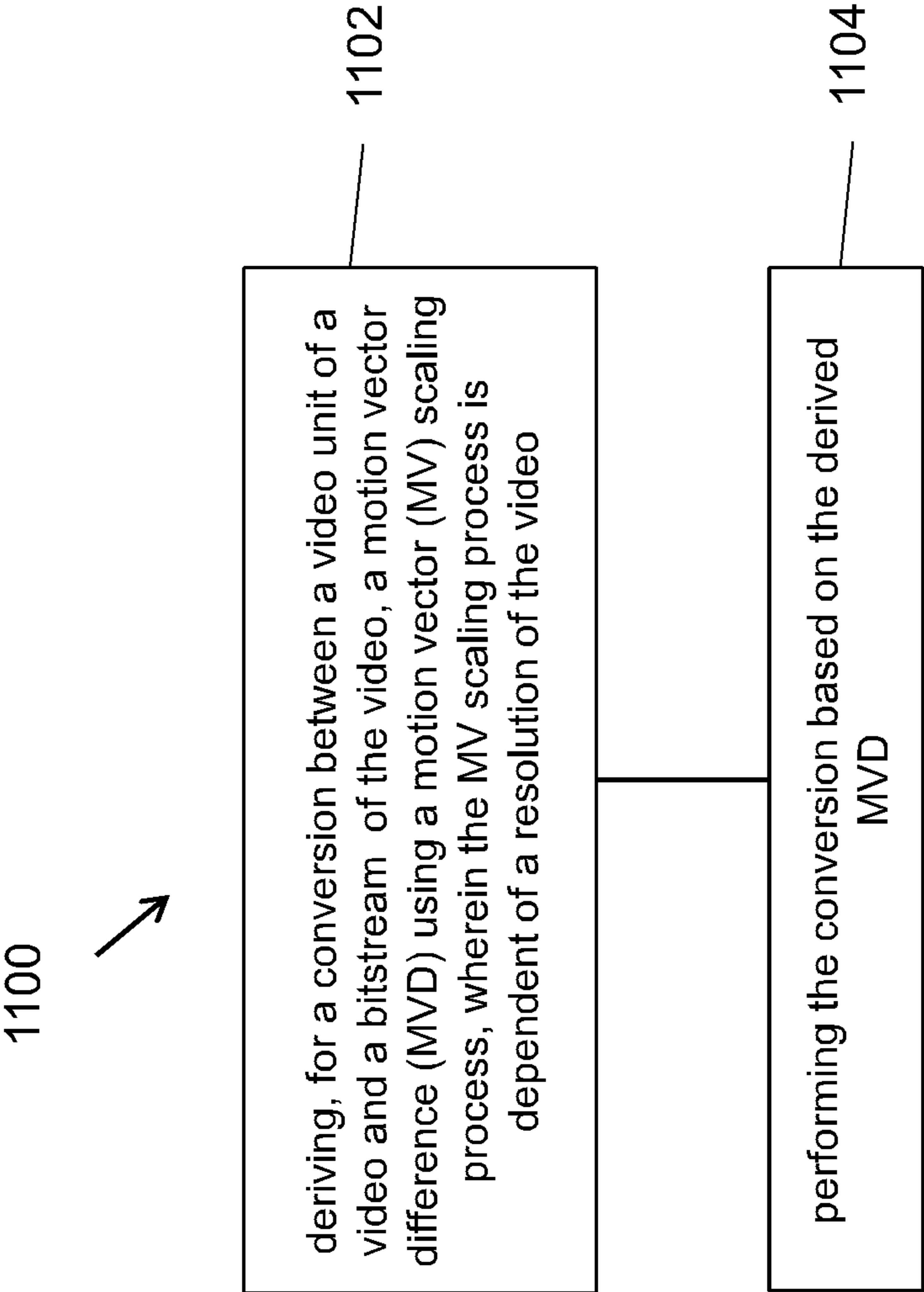


FIG. 11

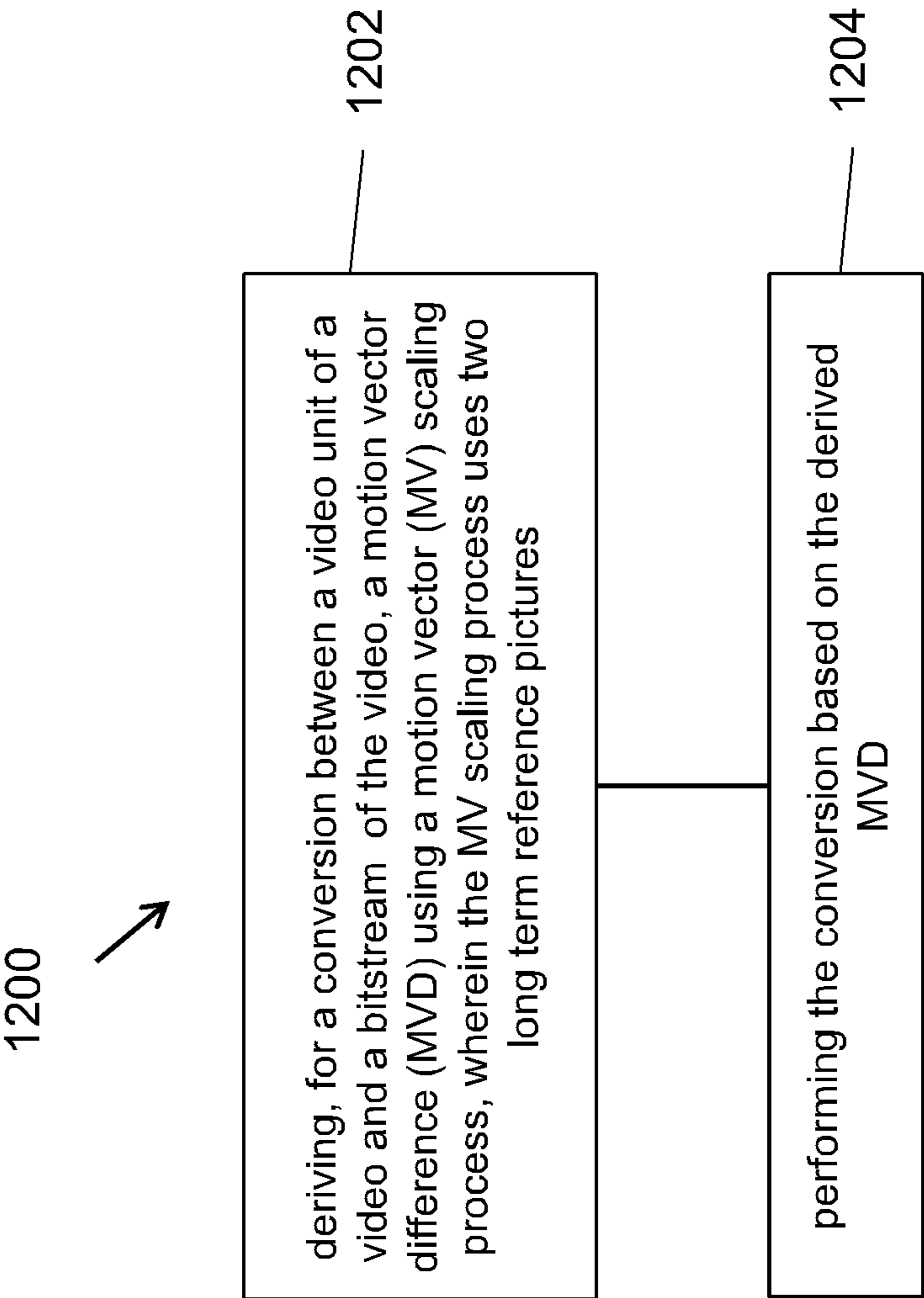


FIG. 12

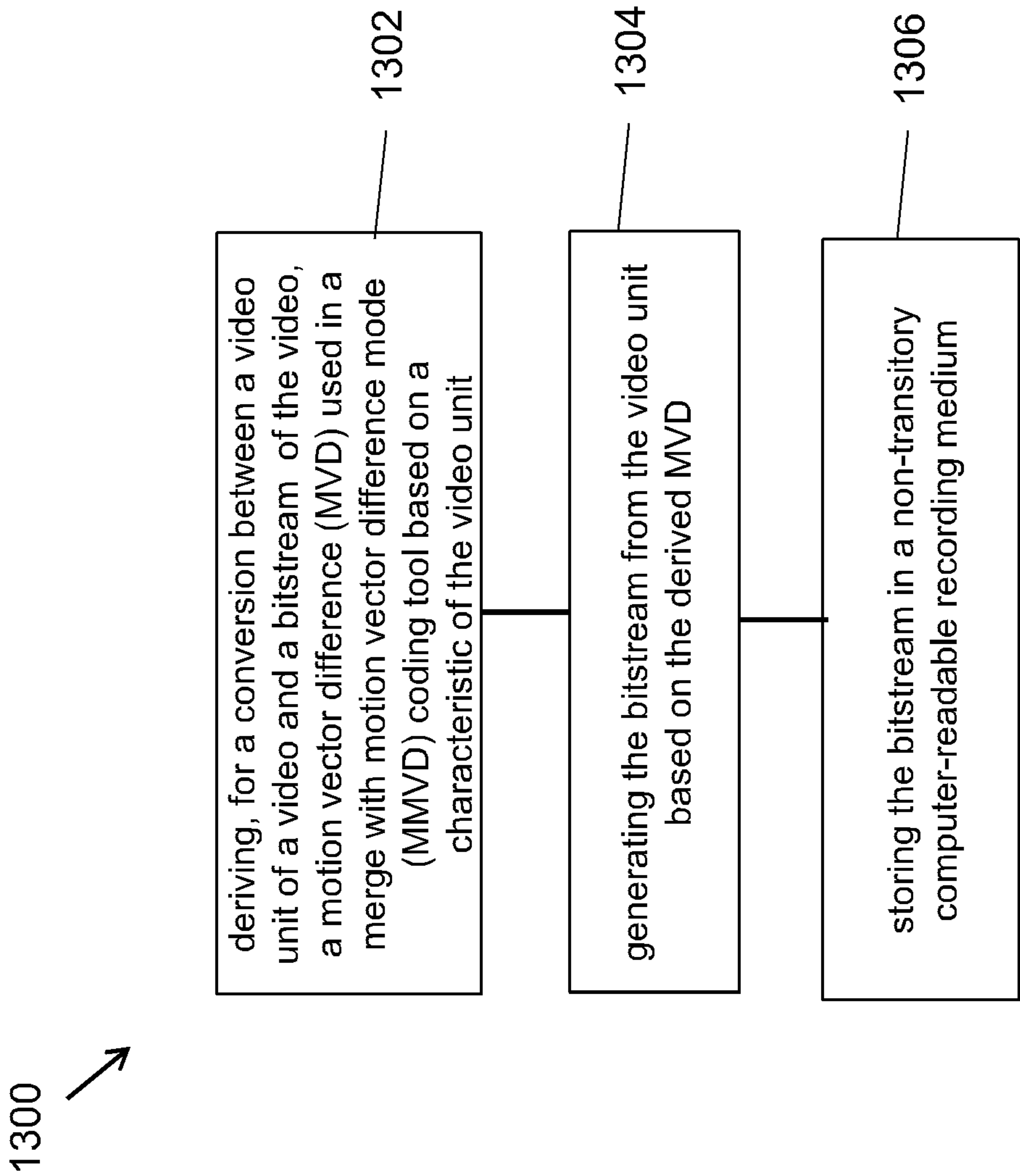


FIG. 13

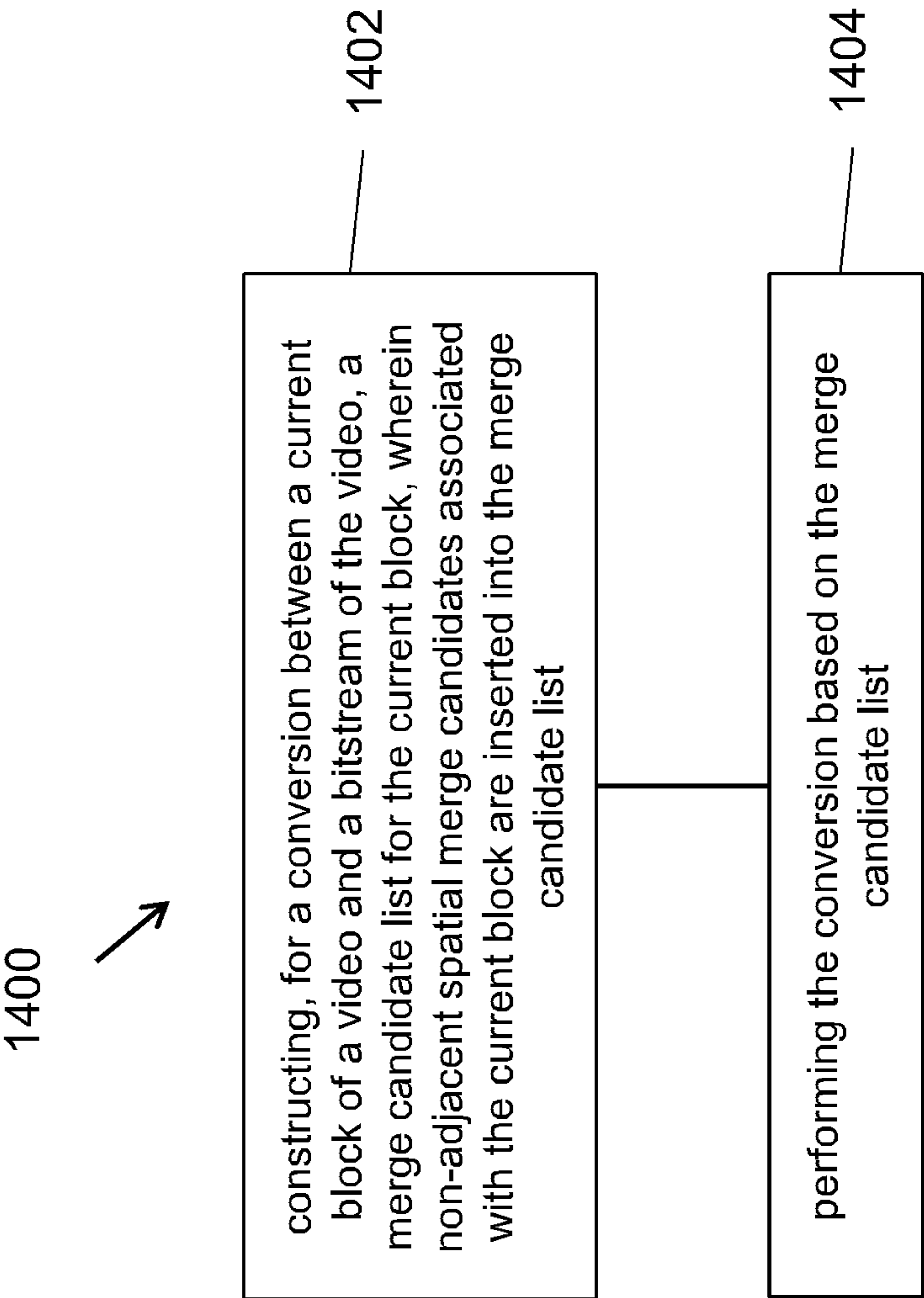


FIG. 14

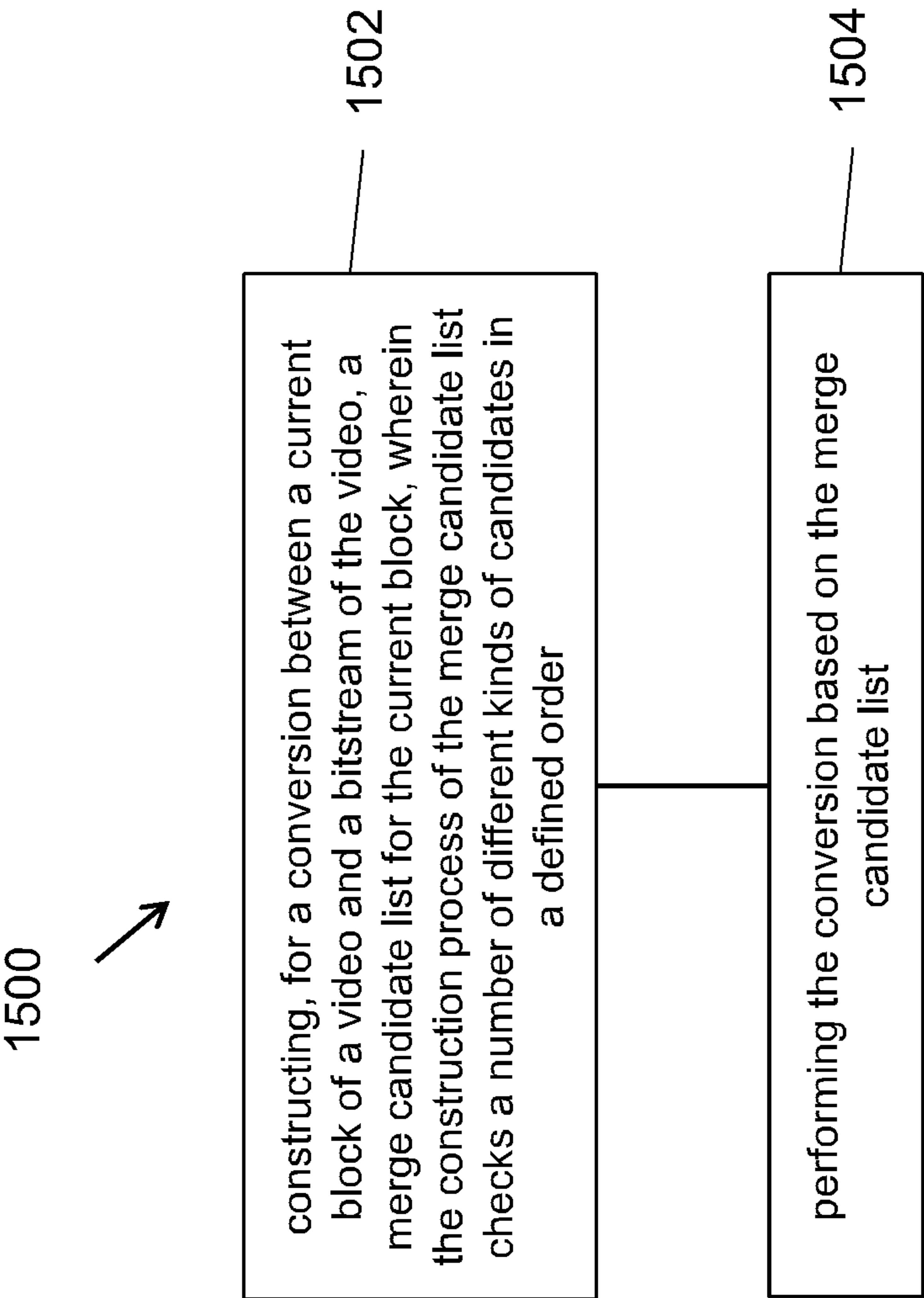


FIG. 15

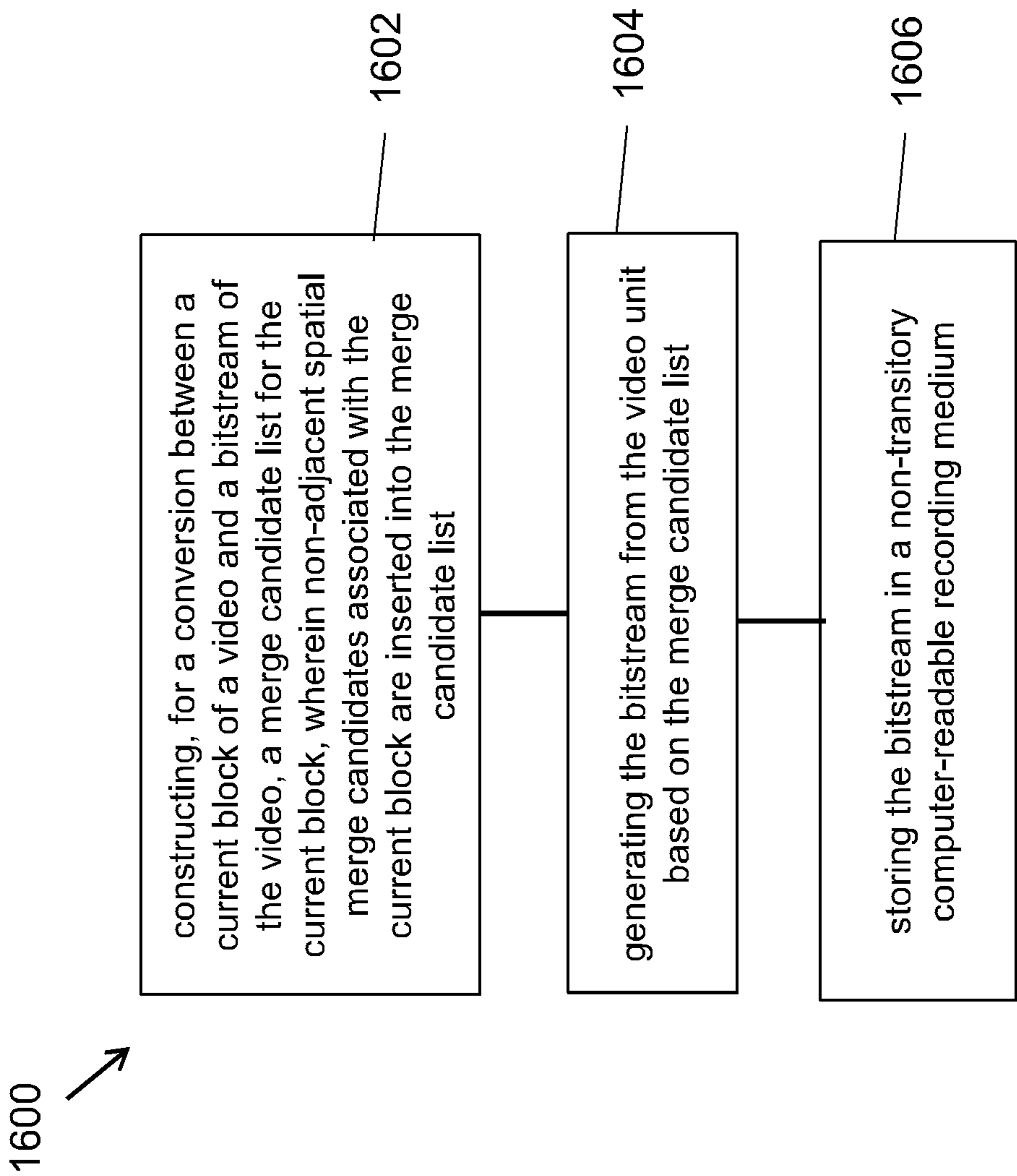


FIG. 16

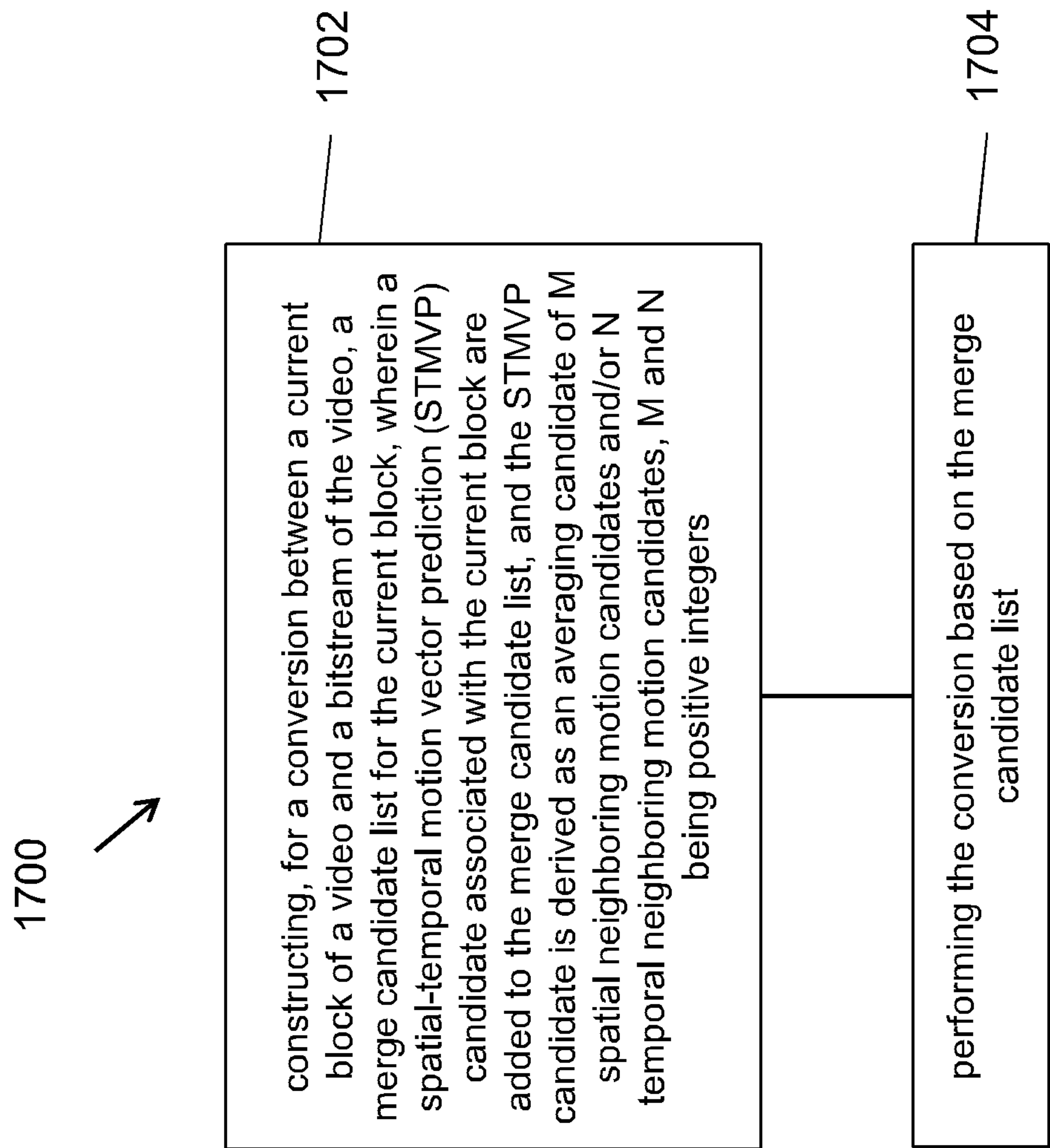


FIG. 17

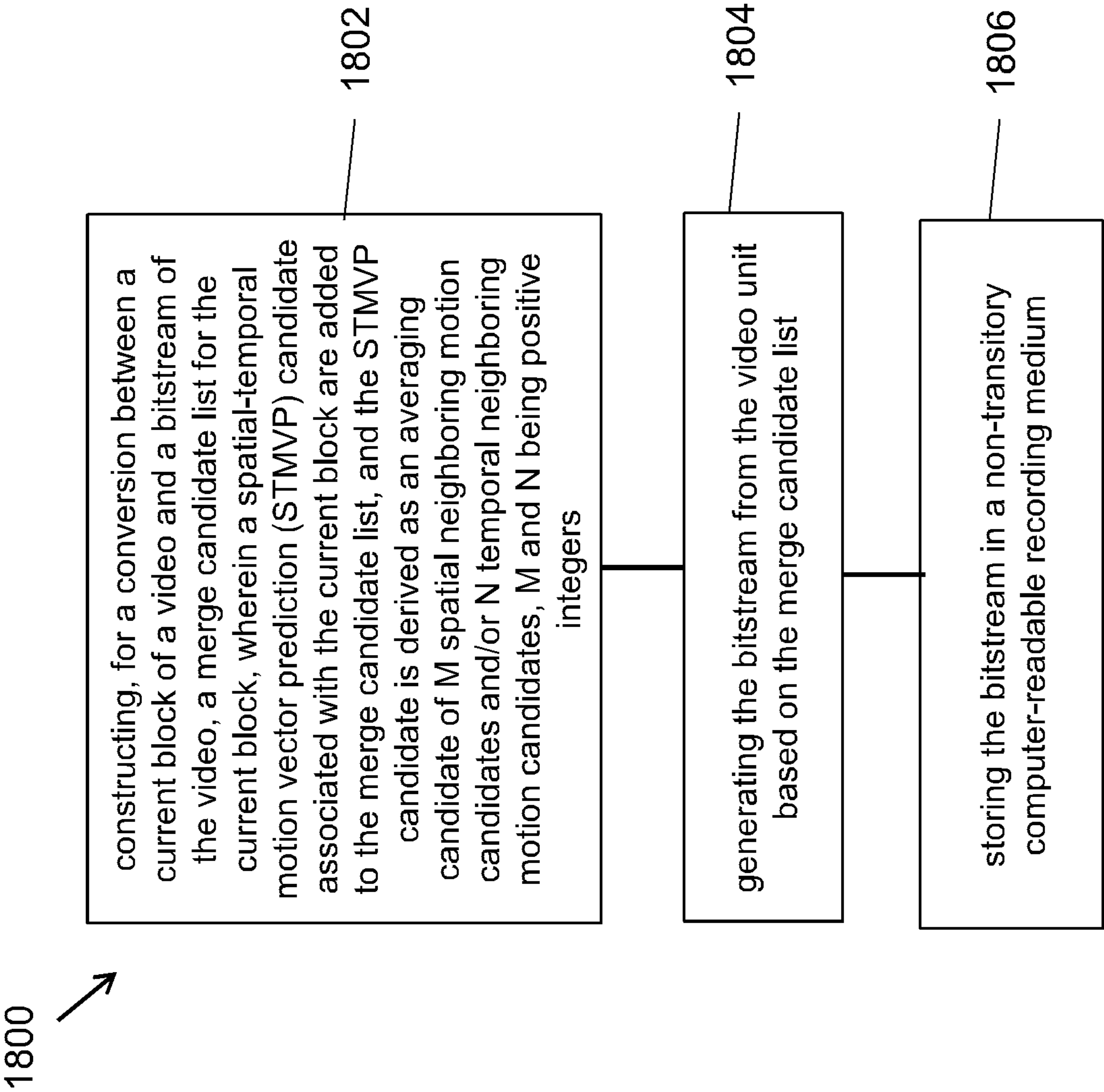


FIG. 18

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/138695

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/56(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;WPI;EPODOC;IEEE,JVET:video,bitstream,motion,vector,difference,MV,MVD,MMVD,cod+,tool,characteristic,predict+,scal+,resolution,reference,picture,direction

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KO, Geonjung et al. "Non-CE4: Simplification of merge MVD derivation" <i>JVET of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, 19–27 March 2019 Document: JVET-N0240</i> , 27 March 2019 (2019-03-27), pages 1-2	1-31
A	CHEN, Huanbang et al. "CE4: Symmetrical MVD mode (Test 4.5.1)" <i>JVET of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, 3–12 Oct. 2018 Document: JVET-M0370</i> , 12 October 2018 (2018-10-12), the whole document	1-31
A	LUO, Jiancong et al. "CE4-related: Simplified symmetric MVD based on CE4.4.3" <i>JVET of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA, 9–18 Jan. 2019 Document: JVET-M0444</i> , 18 January 2019 (2019-01-18), the whole document	1-31
A	US 2015264355 A1 (MEDIATEK INC.) 17 September 2015 (2015-09-17) the whole document	1-31
A	CN 110572672 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD,) 13 December 2019 (2019-12-13) the whole document	1-31



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	“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
“A” document defining the general state of the art which is not considered to be of particular relevance	“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
“E” earlier application or patent but published on or after the international filing date	“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
“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)	“&” document member of the same patent family
“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	

Date of the actual completion of the international search

24 February 2021

Date of mailing of the international search report

24 March 2021

Name and mailing address of the ISA/CN

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

Authorized officer

WU,Qian

Facsimile No. (86-10)62019451

Telephone No. 86- (010) -53961822

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/138695

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 110447228 A (QUALCOMM INC.) 12 November 2019 (2019-11-12) the whole document	1-31

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/138695

Patent document cited in search report				Publication date (day/month/year)		Patent family member(s)		Publication date (day/month/year)	
US	2015264355	A1	17 September 2015	US	9900625	B2	20 February 2018		
CN	110572672	A	13 December 2019	HK	40018863	A0	09 October 2020		
CN	110447228	A	12 November 2019	US	2018278951	A1	27 September 2018		
				US	10701391	B2	30 June 2020		
				WO	2018175911	A1	27 September 2018		