



(51) International Patent Classification:

H04N 19/503 (2014.01) H04N 19/136 (2014.01)

H04N 19/176 (2014.01)

(21) International Application Number:

PCT/CN2020/078793

(22) International Filing Date:

11 March 2020 (11.03.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2019/077637

11 March 2019 (11.03.2019)

CN

PCT/CN2019/078147

14 March 2019 (14.03.2019)

CN

PCT/CN2019/078505

18 March 2019 (18.03.2019)

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

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

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

(54) Title: IMPROVEMENT ON MOTION CANDIDATE LIST CONSTRUCTION

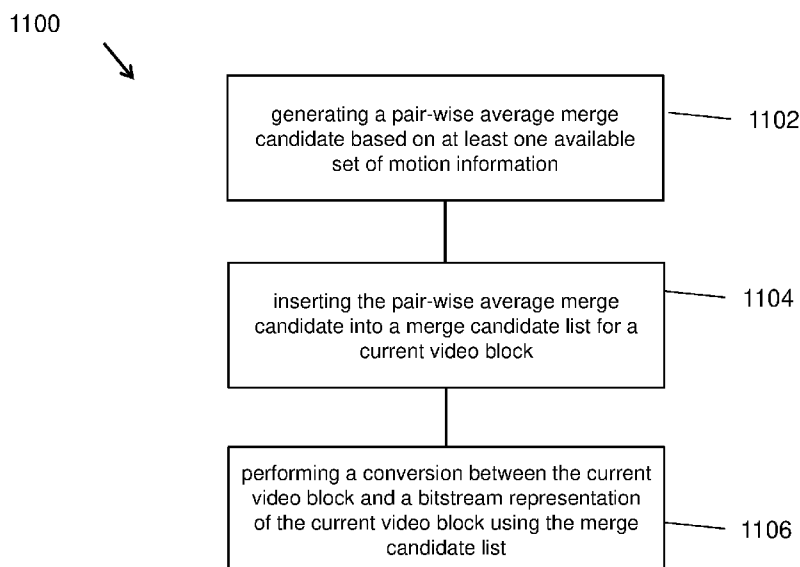


FIG. 11

(57) Abstract: A method for video processing, includes generating a pair-wise average merge candidate based on at least one available set of motion information; inserting the pair-wise average merge candidate into a merge candidate list for a current video block; and performing a conversion between the current video block and a bitstream representation of the current video block using the merge candidate list.

HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, 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))*

IMPROVEMENT ON MOTION CANDIDATE LIST CONSTRUCTION

TECHNICAL FIELD

[0001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefit of International Patent Applications PCT/CN2019/077637, filed on March 11, 2019, PCT/CN2019/078147, filed on March 14, 2019, PCT/CN2019/078505, filed on March 18, 2019. The entire disclosure thereof is incorporated by reference as part of the disclosure of this application.

[0002] This patent document relates to video coding techniques, devices and systems.

BACKGROUND

[0003] In spite of the advances in video compression, digital video still 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

[0004] In one example aspect, a method of video processing is disclosed. The method includes generating a pair-wise average merge candidate based on at least one available set of motion information; inserting the pair-wise average merge candidate into a merge candidate list for a current video block; and performing a conversion between the current video block and a bitstream representation of the current video block using the merge candidate list.

[0005] In another example aspect, another method of video processing is disclosed. The method includes determining whether a pair-wise average merge candidate can be derived based on at least one set of motion information; inserting at least one default merge candidate into a merge candidate list for a current video block in response to determining that no pair-wise average merge candidate can be derived; and performing a conversion between the current video block and a bitstream representation of the current video block using the merge candidate list.

[0006] In still another example aspect, an apparatus in a video system is disclosed. The apparatus includes a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method as described above.

[0007] In yet another example aspect, a non-transitory computer readable media is disclosed. The non-transitory computer readable media has program code stored thereupon, the program code, when executed, causing a processor to implement the method as described above.

[0008] In yet another example aspect, a video encoder apparatus configured to implement one of the above-described methods is disclosed.

[0009] In yet another example aspect, a video decoder apparatus configured to implement one of the above-described methods is disclosed.

[0010] In yet another aspect, a computer-readable medium is disclosed. A processor-executable code for implementing one of the above-described methods is stored on the computer-readable medium.

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

BRIEF DESCRIPTION OF DRAWINGS

[0012] FIG. 1 shows an example derivation process for merge candidates list construction.

[0013] FIG. 2 shows examples of positions of spatial merge candidates.

[0014] FIG. 3 shows examples of candidate pairs considered for redundancy check of spatial merge candidates.

[0015] FIG. 4 shows examples of positions for the second PU of $N \times 2N$ and $2N \times N$ partitions.

[0016] FIG. 5 is an example illustration of motion vector scaling for temporal merge candidate.

[0017] FIG. 6 shows examples of candidate positions for temporal merge candidate, C0 and C1.

[0018] FIG. 7 shows an example of combined bi-predictive merge candidate.

[0019] FIG. 8 summarizes derivation process for motion vector prediction candidate.

[0020] FIG. 9 shows an illustration of motion vector scaling for spatial motion vector candidate.

[0021] FIG. 10 shows an example apparatus for implementing a technique described in the present document.

[0022] FIG. 11 is a flowchart for an example method of video processing.

[0023] FIG. 12 is a flowchart for another example method of video processing.

DETAILED DESCRIPTION

[0024] The present document provides various techniques that can be used by a decoder of image or video bitstreams to improve the quality of decompressed or decoded digital video or images. For brevity, the term “video” is used herein to include both a sequence of pictures (traditionally called video) and individual images. Furthermore, a video encoder may also implement these techniques during the process of encoding in order to reconstruct decoded frames used for further encoding.

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

[0026] 1. Brief Summary

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

[0028] 2. Initial Discussion

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

the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC.

[0030] 2.1 Inter prediction in HEVC/H.265

[0031] Each inter-predicted PU has motion parameters for one or two reference picture lists. Motion parameters include **a motion vector and a reference picture index**. Usage of one of the two reference picture lists may also be signalled using `inter_pred_idc`. Motion vectors may be explicitly coded as deltas relative to predictors.

[0032] When a CU is coded with skip mode, one PU is associated with the CU, and there are no significant residual coefficients, no coded motion vector delta or reference picture index. A merge mode is specified whereby the motion parameters for the current PU are obtained from neighbouring PUs, including spatial and temporal candidates. The merge mode can be applied to any inter-predicted PU, not only for skip mode. The alternative to merge mode is the explicit transmission of motion parameters, where motion vector (to be more precise, motion vector differences (MVD) compared to a motion vector predictor), corresponding reference picture index for each reference picture list and reference picture list usage are signalled explicitly per each PU. Such a mode is named Advanced motion vector prediction (AMVP) in this disclosure.

[0033] When signalling indicates that one of the two reference picture lists is to be used, the PU is produced from one block of samples. This is referred to as 'uni-prediction'. Uni-prediction is available both for P-slices and B-slices.

[0034] When signalling indicates that both of the reference picture lists are to be used, the PU is produced from two blocks of samples. This is referred to as 'bi-prediction'. Bi-prediction is available for B-slices only.

[0035] The following text provides the details on the inter prediction modes specified in HEVC. The description will start with the merge mode.

[0036] 2.1.1 Reference picture list

[0037] In HEVC, the term inter prediction is used to denote prediction derived from data elements (e.g., sample values or motion vectors) of reference pictures other than the current decoded picture. Like in H.264/AVC, a picture can be predicted from multiple reference pictures. The reference pictures that are used for inter prediction are organized in one or more reference

picture lists. The reference index identifies which of the reference pictures in the list should be used for creating the prediction signal.

[0038] A single reference picture list, List 0, is used for a P slice and two reference picture lists, List 0 and List 1 are used for B slices. It should be noted reference pictures included in List 0/1 could be from past and future pictures in terms of capturing/display order.

[0039] **2.1.2 Merge Mode in HEVC**

[0040] **2.1.2.1 Derivation of candidates for merge mode**

[0041] When a PU is predicted using merge mode, an index pointing to an entry in the merge candidates list is parsed from the bitstream and used to retrieve the motion information. The construction of this list is specified in the HEVC standard and can be summarized according to the following sequence of steps:

[0042] Step 1: Initial candidates derivation

[0043] Step 1.1: Spatial candidates derivation

[0044] Step 1.2: Redundancy check for spatial candidates

[0045] Step 1.3: Temporal candidates derivation

[0046] Step 2: Additional candidates insertion

[0047] Step 2.1: Creation of bi-predictive candidates

[0048] Step 2.2: Insertion of zero motion candidates

[0049] These steps are also schematically depicted in FIG. 1. For spatial merge candidate derivation, a maximum of four merge candidates are selected among candidates that are located in five different positions. For temporal merge candidate derivation, a maximum of one merge candidate is selected among two candidates. Since constant number of candidates for each PU is assumed at decoder, additional candidates are generated when the number of candidates obtained from step 1 does not reach the maximum number of merge candidate (MaxNumMergeCand) which is signalled in slice header. Since the number of candidates is constant, index of best merge candidate is encoded using truncated unary binarization (TU). If the size of CU is equal to 8, all the PUs of the current CU share a single merge candidate list, which is identical to the merge candidate list of the $2N \times 2N$ prediction unit.

[0050] In the following, the operations associated with the aforementioned steps are detailed.

[0051] **2.1.2.2 Spatial candidates derivation**

[0052] In the derivation of spatial merge candidates, a maximum of four merge candidates are selected among candidates located in the positions depicted in FIG. 2. The order of derivation is A1, B1, B0, A0 and B2. Position B2 is considered only when any PU of position A1, B1, B0, A0 is not available (e.g. because it belongs to another slice or tile) or is intra coded. After candidate at position A1 is added, the addition of the remaining candidates is subject to a redundancy check which ensures that candidates with same motion information are excluded from the list so that coding efficiency is improved. To reduce computational complexity, not all possible candidate pairs are considered in the mentioned redundancy check. Instead only the pairs linked with an arrow in FIG. 3 are considered and a candidate is only added to the list if the corresponding candidate used for redundancy check has not the same motion information. Another source of duplicate motion information is the “second PU” associated with partitions different from $2N \times 2N$. As an example, FIG. 4 depicts the second PU for the case of $N \times 2N$ and $2N \times N$, respectively. When the current PU is partitioned as $N \times 2N$, candidate at position A1 is not considered for list construction. In fact, by adding this candidate will lead to two prediction units having the same motion information, which is redundant to just have one PU in a coding unit. Similarly, position B1 is not considered when the current PU is partitioned as $2N \times N$.

[0053] 2.1.2.3 Temporal candidates derivation

[0054] In this step, only one candidate is added to the list. Particularly, in the derivation of this temporal merge candidate, a scaled motion vector is derived based on co-located PU belonging to the picture which has the smallest POC difference with current picture within the given reference picture list. The reference picture list to be used for derivation of the co-located PU is explicitly signalled in the slice header. The scaled motion vector for temporal merge candidate is obtained as illustrated by the dotted line in FIG. 5, which is scaled from the motion vector of the co-located PU using the POC distances, t_b and t_d , where t_b is defined to be the POC difference between the reference picture of the current picture and the current picture and t_d is defined to be the POC difference between the reference picture of the co-located picture and the co-located picture. The reference picture index of temporal merge candidate is set equal to zero. A practical realization of the scaling process is described in the HEVC specification. For a B-slice, two motion vectors, one is for reference picture list 0 and the other is for reference picture list 1, are obtained and combined to make the bi-predictive merge candidate.

[0055] In the co-located PU (Y) belonging to the reference frame, the position for the temporal candidate is selected between candidates C0 and C1, as depicted in FIG. 6. If PU at position C0 is not available, is intra coded, or is outside of the current coding tree unit (CTU aka. LCU, largest coding unit) row, position C1 is used. Otherwise, position C0 is used in the derivation of the temporal merge candidate.

[0056] Temporal Motion Vector Prediction is also known as “TMVP”.

[0057] **2.1.2.4 Additional candidates insertion**

[0058] Besides spatial and temporal merge candidates, there are two additional types of merge candidates: combined bi-predictive merge candidate and zero merge candidate. Combined bi-predictive merge candidates are generated by utilizing spatial and temporal merge candidates. Combined bi-predictive merge candidate is used for B-Slice only. The combined bi-predictive candidates are generated by combining the first reference picture list motion parameters of an initial candidate with the second reference picture list motion parameters of another. If these two tuples provide different motion hypotheses, they will form a new bi-predictive candidate. As an example, FIG. 7 depicts the case when two candidates in the original list (on the left), which have mvL0 and refIdxL0 or mvL1 and refIdxL1, are used to create a combined bi-predictive merge candidate added to the final list (on the right). There are numerous rules regarding the combinations which are considered to generate these additional merge candidates.

[0059] Zero motion candidates are inserted to fill the remaining entries in the merge candidates list and therefore hit the MaxNumMergeCand capacity. These candidates have zero spatial displacement and a reference picture index which starts from zero and increases every time a new zero motion candidate is added to the list. Finally, no redundancy check is performed on these candidates.

[0060] **2.1.3 AMVP**

[0061] AMVP exploits spatio-temporal correlation of motion vector with neighbouring PUs, which is used for explicit transmission of motion parameters. For each reference picture list, a motion vector candidate list is constructed by firstly checking availability of left, above temporally neighbouring PU positions, removing redundant candidates and adding zero vector to make the candidate list to be constant length. Then, the encoder can select the best predictor from the candidate list and transmit the corresponding index indicating the chosen candidate. Similarly

with merge index signalling, the index of the best motion vector candidate is encoded using truncated unary. The maximum value to be encoded in this case is 2 (see FIG. 8). In the following sections, details about derivation process of motion vector prediction candidate are provided.

[0062] 2.1.3.1 Derivation of AMVP candidates

[0063] FIG. 8 shows an example derivation process for motion vector prediction candidates.

[0064] In motion vector prediction, two types of motion vector candidates are considered: spatial motion vector candidate and temporal motion vector candidate. For spatial motion vector candidate derivation, two motion vector candidates are eventually derived based on motion vectors of each PU located in five different positions as depicted in FIG. 2.

[0065] For temporal motion vector candidate derivation, one motion vector candidate is selected from two candidates, which are derived based on two different co-located positions. After the first list of spatio-temporal candidates is made, duplicated motion vector candidates in the list are removed. If the number of potential candidates is larger than two, motion vector candidates whose reference picture index within the associated reference picture list is larger than 1 are removed from the list. If the number of spatio-temporal motion vector candidates is smaller than two, additional zero motion vector candidates is added to the list.

[0066] 2.1.3.2 Spatial motion vector candidates

[0067] In the derivation of spatial motion vector candidates, a maximum of two candidates are considered among five potential candidates, which are derived from PUs located in positions as depicted in FIG. 2, those positions being the same as those of motion merge. The order of derivation for the left side of the current PU is defined as A0, A1, and scaled A0, scaled A1. The order of derivation for the above side of the current PU is defined as B0, B1, B2, scaled B0, scaled B1, scaled B2. For each side there are therefore four cases that can be used as motion vector candidate, with two cases not required to use spatial scaling, and two cases where spatial scaling is used. The four different cases are summarized as follows.

[0068] No spatial scaling

[0069] (1) Same reference picture list, and same reference picture index (same POC)

[0070] (2) Different reference picture list, but same reference picture (same POC)

[0071] Spatial scaling

[0072] (3) Same reference picture list, but different reference picture (different POC)

[0073] (4) Different reference picture list, and different reference picture (different POC)

[0074] The no-spatial-scaling cases are checked first followed by the spatial scaling. Spatial scaling is considered when the POC is different between the reference picture of the neighbouring PU and that of the current PU regardless of reference picture list. If all PUs of left candidates are not available or are intra coded, scaling for the above motion vector is allowed to help parallel derivation of left and above MV candidates. Otherwise, spatial scaling is not allowed for the above motion vector.

[0075] In a spatial scaling process, the motion vector of the neighbouring PU is scaled in a similar manner as for temporal scaling, as depicted as FIG. 9. The main difference is that the reference picture list and index of current PU is given as input; the actual scaling process is the same as that of temporal scaling.

[0076] **2.1.3.3 Temporal motion vector candidates**

[0077] Apart for the reference picture index derivation, all processes for the derivation of temporal merge candidates are the same as for the derivation of spatial motion vector candidates (see FIG. 6). The reference picture index is signalled to the decoder.

[0078] **2.2 Inter prediction methods in VVC**

[0079] There are several new coding tools for inter prediction improvement, such as Adaptive motion vector difference resolution (AMVR) for signaling MVD, affine prediction mode, Triangular prediction mode (TPM), ATMVP, Generalized Bi-Prediction (GBI), Bi-directional Optical flow (BIO).

[0080] **2.2.1 Coding block structure in VVC**

[0081] In VVC, a QuadTree/BinaryTree/MultipleTree (QT/BT/TT) structure is adopted to divide a picture into square or rectangle blocks.

[0082] Besides QT/BT/TT, separate tree (a.k.a. Dual coding tree) is also adopted in VVC for I-frames. With separate tree, the coding block structure are signaled separately for the luma and chroma components.

[0083] **2.2.2 Extended merge prediction in VVC**

[0084] In VTM4, the merge candidate list is constructed by including the following five types of candidates in order:

- [0085] Spatial MVP from spatial neighbour CUs
- [0086] Temporal MVP from collocated CUs
- [0087] History-based MVP from an FIFO table
- [0088] Pairwise average MVP
- [0089] Zero MVs.
- [0090] The size of merge list is signalled in slice header and the maximum allowed size of merge list is 6 in VTM4. For each CU code in merge mode, an index of best merge candidate is encoded using truncated unary binarization (TU). The first bin of the merge index is coded with context and bypass coding is used for other bins.
- [0091] The generation process of each category of merge candidates is provided in this session.
- [0092] **2.2.1.1 Spatial candidates derivation**
- [0093] The derivation of spatial merge candidates in VVC is same to that in HEVC.
- [0094] **2.2.2.2 Temporal candidates derivation**
- [0095] The derivation of temporal merge candidates in VVC is same to that in HEVC.
- [0096] **2.2.2.3 History-based merge candidates derivation**
- [0097] The history-based MVP (HMVP) merge candidates are added to merge list after the spatial MVP and TMVP. In this method, the motion information of a previously coded block is stored in a table and used as MVP for the current CU. The table with multiple HMVP candidates is maintained during the encoding/decoding process. The table is reset (emptied) when a new CTU row is encountered. Whenever there is a non-subblock inter-coded CU, the associated motion information is added to the last entry of the table as a new HMVP candidate.
- [0098] In VTM4 the HMVP table size S is set to be 6, which indicates up to 6 History-based MVP (HMVP) candidates may be added to the table. When inserting a new motion candidate to the table, a constrained first-in-first-out (FIFO) rule is utilized wherein redundancy check is firstly applied to find whether there is an identical HMVP in the table. If found, the identical HMVP is removed from the table and all the HMVP candidates afterwards are moved forward,
- [0099] HMVP candidates could be used in the merge candidate list construction process. The latest several HMVP candidates in the table are checked in order and inserted to the candidate list after the TMVP candidate. Redundancy check is applied on the HMVP candidates to the

spatial or temporal merge candidate.

[00100] To reduce the number of redundancy check operations, the following simplifications are introduced:

[00101] Number of HMPV candidates is used for merge list generation is set as $(N \leq 4) ? M : (8 - N)$, wherein N indicates number of existing candidates in the merge list and M indicates number of available HMVP candidates in the table.

[00102] Once the total number of available merge candidates reaches the maximally allowed merge candidates minus 1, the merge candidate list construction process from HMVP is terminated.

[00103] 2.2.2.4 Pair-wise average merge candidate derivation

[00104] Pairwise average candidates are generated by averaging predefined pairs of candidates in the existing merge candidate list, and the predefined pairs are defined as $\{(0, 1)\}$, where the number denote the merge indices to the merge candidate list. The averaged motion vectors are calculated separately for each reference list. If both motion vectors are available in one list, these two motion vectors are averaged even when they point to different reference pictures; if only one motion vector is available, use the one directly; if no motion vector is available, keep this list invalid.

[00105] When the merge list is not full after pair-wise average merge candidates are added, the zero MVPs are inserted in the end until the maximum merge candidate number is encountered.

[00106] 2.2.3 Shared merge list in VVC

[00107] To reduce the decoder complexity and support parallel encoding, JVET-M0147 proposed to share the same merging candidate list for all leaf coding units (CUs) of one ancestor node in the CU split tree for enabling parallel processing of small skip/merge-coded CUs. The ancestor node is named merge sharing node. The shared merging candidate list is generated at the merge sharing node pretending the merge sharing node is a leaf CU.

[00108] More specifically, the following may apply:

[00109] If the block has luma samples no larger than 32, and split to 2 4x4 child blocks, sharing merge lists between very small blocks (e.g. two adjacent 4x4 blocks) is used.

[00110] If the block has luma samples larger than 32, however, after a split, at least one child

block is smaller than the threshold (32), all child blocks of that split share the same merge list (e.g. 16x4 split ternary or 8x8 with quad split).

[00111] Such a restriction is only applied to regular merge mode.

[00112] 2.2.4 IBC merge list

[00113] When IBC is added as proposed in JVET-M0483, HMVP candidates are also applicable for the IBC merge list.

[00114] More specifically, another 5 IBC candidates may be stored. In current implementation, the regular and IBC candidates are stored in the same HMVP table. However, they are utilized and updated independently. The first M ($M \leq 5$) candidates are for the usage of regular merge/AMVP list; and the remaining N candidates ($N \leq 5$) are for the usage of IBC mode. Two counters are maintained to indicate how many regular motion candidates and how many IBC motion candidates in the HMVP table. Therefore, it is equal to use 2 HMVP tables, one is for the regular merge modes, and the other for the IBC mode.

[00115] Share same process as in regular MV merge, but disallow TMVP, zero vector means unavailable as it is invalid. It is noted that for a spatial neighboring block, only if it is coded with IBC mode, the associated motion information may be added to the IBC merge list. Meanwhile, for the HMVP part, only the last few HMVP candidates (which are stored IBC motion candidates) may be considered in the IBC merge list.

[00116] 3. Examples of Problems Solved by Embodiments

[00117] The current design of merge candidate list construction, the pair-wise averaging merge candidate can be built only when the first two merge candidates has already been in the candidate list. In the worst case, all the potential merge candidates must be traversed before building the pair-wise average candidate. Therefore, the pair-wise average candidate cannot be built in parallel with other candidates.

[00118] 4. Example techniques and embodiments

[00119] The listing of techniques below should be considered as examples to explain general concepts. These techniques should not be interpreted in a narrow way. Furthermore, these techniques can be combined in any manner.

[00120] In the following discussion, $SatShift(x, n)$ is defined as

$$SatShift(x, n) = \begin{cases} (x + offset0) >> n & \text{if } x \geq 0 \\ -((-x + offset1) >> n) & \text{if } x < 0 \end{cases}$$

$Shift(x, n)$ is defined as $Shift(x, n) = (x + offset0) >> n$.

[00121] In one example, $offset0$ and/or $offset1$ are set to $(1 << n) >> 1$ or $(1 << (n-1))$. In another example, $offset0$ and/or $offset1$ are set to 0.

[00122] In another example, $offset0 = offset1 = ((1 << n) >> 1) - 1$ or $((1 << (n-1))) - 1$.

$Clip3(min, max, x)$ is defined as

$$Clip3(Min, Max, x) = \begin{cases} Min & \text{if } x < Min \\ Max & \text{if } x > Max \\ x & \text{Otherwise} \end{cases}$$

[00123] In the present document, an operation between two motion vectors means the operation will be applied to both the two components of the motion vector. For example, $MV3 = MV1 + MV2$ is equivalent to $MV3_x = MV1_x + MV2_x$ and $MV3_y = MV1_y + MV2_y$.

[00124] In the present document, the left neighbouring block, left-bottom neighbouring block, above neighbouring block, right-above neighbouring block and left-above neighbouring block are denoted as block A_1 , A_0 , B_1 , B_0 and B_2 as shown in FIG. 2.

1. In one example, if only one set of motion information (it may be fetched from a spatial neighbouring block, or from a temporal neighbouring block, or from an HMVP entry, etc.) is available to build the pair-wise average merge candidate, then no pair-wise average merge candidate can be built.
 - a. Alternatively, a pair-wise average merge candidate can be built with the one set of motion information. Suppose the motion information includes: inter-prediction direction denoted as $inter_dir$, reference indices for the two lists denoted as $ref_idx[0]$ and $ref_idx[1]$ ($ref_idx[L]$ $L = 0$ or 1 is marked as “-1” if no motion vector refers to reference list x), GBI index denoted as gbi_idx , the motion vector referring reference list 0 denoted as (mv_x^0, mv_y^0) , and the motion vector referring reference list 1 denoted as (mv_x^1, mv_y^1) . Then the motion information for the pair-wise average merge candidate may be built following one or some of the exemplary rules below:

- 1) In one example, `inter_dir` is copied to the pair-wise average merge candidate.
- 2) In one example, `ref_idx[0]`, `ref_idx[1]` are copied to the pair-wise average merge candidate.
- 3) In one example, `gbi_idx` is copied to the pair-wise average merge candidate.
 - a. Alternatively, GBi index of the pair-wise average merge candidate is set to be zero or other default values meaning GBi off.
- 4) In one example, if (mv_x^L, mv_y^L) where $L = 0$ or 1 is available, $(f(mv_x^L), g(mv_y^L))$ is set to be the motion vector of the pair-wise average merge candidate referring to reference list L . f and g are two functions.
 - a. In one example, $f(m)=g(m)=m + offset$, where *offset* is an integer number such as 3.
 - b. In one example, $f(m)=g(m)=m + (offset \ll Precise)$, where *offset* is an integer number such as 3; *Precise* defines the MV precision, for example, *Precise*=4.
 - c. In one example, $f(m)=g(m)=SatShift(m, k)$, where k is an integer such as 1.
 - d. In one example, $f(m)=g(m)= Shift(m, k)$, where k is an integer such as 1.
 - e. In one example, $f(m)=g(m)= m \ll k$, where k is an integer such as 1.
- b. Alternatively, a pair-wise average merge candidate can be built with the one available set of motion information and one constructed set of motion information.
 - i. In one example, the inter-prediction direction of the constructed set of motion information is set equal to that of the available set of motion information.
 - ii. In one example, the reference index of reference list X of the constructed set of motion information is set equal to that of the available set of motion information. X may be 0 or 1.

- iii. In one example, MV for reference list X of the constructed set of motion information is set equal to zero. X may be 0 or 1.
- 2. In one example, Bullet 1 can be applied when there is only one available merge candidate in the merge candidate list before adding the pair-wise average merge candidate.
- 3. In one example, the first set of motion information and the second set of motion information may be checked whether they are identical or similar before they are used to build the pair-wise average merge candidate.
 - a. In one example, if the two sets of motion information are identical or similar, then no pair-wise average merge candidate can be built with them.
 - b. In an alternative example, if the two sets of motion information are identical or similar, then the pair-wise average merge candidate can be built with them.
 - i. In one example, the second (or the first) set of motion information can be treated as unavailable.
 - ii. Bullet 1 can be applied.
- 4. It is proposed that the pair-wise average merge candidate is always built with motion information fetched from two predefined spatial neighbouring blocks.
 - a. In one example, the pair-wise average merge candidate is always built with motion information fetched from {block A₁, block B₁}.
 - i. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block A₁, block A₀}.
 - ii. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block B₁, block B₀}.
 - iii. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block A₁, block B₀}.
 - iv. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block B₁, block A₀}.
 - v. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block A₁, block B₂}.
 - vi. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block B₁, block B₂}.

- vii. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block A₀, block B₂}.
 - viii. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block B₀, block B₂}.
 - b. In one example, if no motion information can be fetched from the two predefined spatial neighbouring blocks (e.g. the neighbouring blocks are unavailable or intra-coded), then no pair-wise average merge candidate can be built.
 - c. In one example, if motion information can only be fetched from one of the two predefined spatial neighbouring blocks (e.g. the other neighbouring block is unavailable or intra-coded), then Bullet 1 can be applied.
 - d. In one example, Bullet 3 can be applied on the two sets of motion information fetched from the two predefined spatial neighbouring blocks.
 - e. In one example, the two spatial neighboring blocks may be replaced by two blocks within current CTU/CTU row/slice/tile group/tile which may be not necessarily adjacent to the current block.
5. It is proposed that one or more pair-wise average merge candidates are always built with motion information fetched from N predefined spatial neighbouring blocks, where $N > 2$. For example, $N = 3$.
- a. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₀};
 - b. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block A₀};
 - c. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₂};
 - d. In one example, the predefined spatial neighbouring blocks are {block A₁, block A₀, block B₂};
 - e. In one example, the predefined spatial neighbouring blocks are {block B₁, block B₀, block B₂};
 - f. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₀};

- g. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₀, block B₁};
- h. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₀, block B₁, block B₂};
- i. In one example, only one pair-wise average merge candidate can be built with motion information fetched from N predefined spatial neighbouring blocks.
 - i. For example, each pair of the predefined spatial neighbouring blocks are checked in order, until a pair-wise average merge candidate can be built with motion information from one pair of the predefined spatial neighbouring blocks.
 - ii. In one example, if no motion information can be fetched from the N predefined spatial neighbouring blocks, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N predefined spatial neighbouring blocks, then Bullet 1 can be applied.
 - iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of the predefined spatial neighbouring blocks or not.
- j. In one example, S (S>1) pair-wise average merge candidates can be built with motion information fetched from N predefined spatial neighbouring blocks.
 - i. For example, each pair of the predefined spatial neighbouring blocks are checked in order, until the S pair-wise average merge candidates can be built with motion information from S pairs of the predefined spatial neighbouring blocks.
 - ii. In one example, if no motion information can be fetched from the N predefined spatial neighbouring blocks, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N predefined spatial neighbouring blocks, then Bullet 1 can be applied.

- iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of the predefined spatial neighbouring blocks or not.
 - k. In one example, the N spatial neighboring blocks may be replaced by two blocks within current CTU/CTU row/slice/tile group/tile which may be not necessarily adjacent to the current block.
6. It is proposed that one or more pair-wise average merge candidates are always built with motion information fetched from N predefined spatial neighbouring blocks and M predefined TMVPs. For example, N = 2 and M = 1.
- a. In one example, the pair-wise average merge candidate is always built with motion information fetched from {block A₁, block B₁}.
 - i. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block A₁, block A₀}.
 - ii. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block B₁, block B₀}.
 - iii. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block A₁, block B₀}.
 - iv. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block B₁, block A₀}.
 - v. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block A₁, block B₂}.
 - vi. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block B₁, block B₂}.
 - vii. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block A₀, block B₂}.
 - viii. Alternatively, the pair-wise average merge candidate is always built with motion information fetched from {block B₀, block B₂}.
 - b. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₀};

- c. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block A₀};
- d. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₂};
- e. In one example, the predefined spatial neighbouring blocks are {block A₁, block A₀, block B₂};
- f. In one example, the predefined spatial neighbouring blocks are {block B₁, block B₀, block B₂};
- g. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₀};
- h. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₀, block B₁};
- i. In one example, the predefined spatial neighbouring blocks are {block A₁, block B₁, block B₀, block B₁, block B₂};
- j. In one example, TMVP can be derived in the same way as HEVC or VVC.
- k. In one example, only one pair-wise average merge candidate can be built with motion information fetched from N predefined spatial neighbouring blocks and M predefined TMVPs.
 - i. For example, each pair of the predefined spatial neighbouring blocks and/or TMVPs are checked in order, until a pair-wise average merge candidate can be built with motion information from one pair of the predefined spatial neighbouring blocks or TMVPs.
 - 1) For example, the checking order is {block A₁, block B₁}, { block A₁, TMVP}, { block A₂, TMVP}.
 - ii. In one example, if no motion information can be fetched from the N predefined spatial neighbouring blocks and M TMVPs, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N predefined spatial neighbouring blocks and/or M TMVPs, then Bullet 1 can be applied.

- iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of the predefined spatial neighbouring blocks and/or TMVP or not.
- 1. In one example, S ($S > 1$) pair-wise average merge candidates can be built with motion information fetched from N predefined spatial neighbouring blocks and/or M TMVPs.
 - i. For example, each pair of the predefined spatial neighbouring blocks and/or TMVPs are checked in order, until the S pair-wise average merge candidates can be built with motion information from S pairs of the predefined spatial neighbouring blocks and/or TMVPs.
 - ii. In one example, if no motion information can be fetched from the N predefined spatial neighbouring blocks and/or M TMVPs, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N predefined spatial neighbouring blocks and/or M TMVPs, then Bullet 1 can be applied.
 - iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of the predefined spatial neighbouring blocks and/or TMVPs or not.
- m. In one example, the N spatial neighboring blocks may be replaced by two blocks within current CTU/CTU row/slice/tile group/tile which may be not necessarily adjacent to the current block.
- 7. It is proposed that one or more pair-wise average merge candidates are always built with motion information fetched from N entries in the HMVP table. For example, $N = 2$.
 - a. The N entries can be predefined. For example, they may be the first N entries in the table, or they may be the last N entries in the table;
 - i. N should be no larger than the total number of available entries in the table.
 - ii. In one example, N is set to be the total number of available entries in the table.

- iii. Alternatively, the selected N entries may be not associated with consecutive entry indices, e.g., the first and the last entry may be utilized.
- b. In one example, only one pair-wise average merge candidate can be built with motion information fetched from N entries in the HMVP table.
 - i. For example, each pair of the entries in the HMVP table are checked in order, until a pair-wise average merge candidate can be built with motion information from one pair of the entries in the HMVP table.
 - ii. In one example, if no motion information can be fetched from the N entries in the HMVP table, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N entries in the HMVP table, then Bullet 1 can be applied.
 - iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of the entries in the HMVP table or not.
- c. In one example, S ($S > 1$) pair-wise average merge candidates can be built with motion information fetched from N entries in the HMVP table.
 - i. For example, each pair of the N entries in the HMVP table are checked in order, until the S pair-wise average merge candidates can be built with motion information from S pairs of the N entries in the HMVP table.
 - ii. In one example, if no motion information can be fetched from the N entries in the HMVP table, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N entries in the HMVP table, then Bullet 1 can be applied.
 - iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of N entries in the HMVP table or not.

8. It is proposed that one or more pair-wise average merge candidates are always built with motion information fetched from N predefined spatial neighbouring blocks and motion information fetched from M entries in the HMVP table. For example, $N = 1$, $M = 1$.
- a. The N predefined spatial neighbouring blocks can be predefined as claimed in Bullet 6.
 - b. The M entries in the HMVP table can be predefined as claimed in Bullet 7.
 - c. In one example, only one pair-wise average merge candidate can be built with motion information fetched from N predefined spatial neighbouring blocks and/or M entries in the HMVP table.
 - i. For example, each pair of the N predefined spatial neighbouring blocks and/or the entries in the HMVP table are checked in order, until a pair-wise average merge candidate can be built with motion information from one pair of N predefined spatial neighbouring blocks and/or M entries in the HMVP table.
 - ii. In one example, if no motion information can be fetched from the N predefined spatial neighbouring blocks and/or M entries in the HMVP table, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table, then Bullet 1 can be applied.
 - iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of N predefined spatial neighbouring blocks and/or M entries in the HMVP table or not.
 - d. In one example, S ($S > 1$) pair-wise average merge candidates can be built with motion information fetched from N predefined spatial neighbouring blocks and/or M entries in the HMVP table.
 - i. For example, each pair of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table are checked in order, until the S pair-wise average merge candidates can be built with motion information from

- S pairs of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table.
- ii. In one example, if no motion information can be fetched from the N predefined spatial neighbouring blocks and/or M entries in the HMVP table, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table, then Bullet 1 can be applied.
 - iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of N predefined spatial neighbouring blocks and/or M entries in the HMVP table or not.
9. It is proposed that one or more pair-wise average merge candidates are always built with motion information fetched from N predefined spatial neighbouring blocks motion information fetched from M entries in the HMVP table and P TMVPs. For example, $N = 1$, $M = 1$, $P = 1$.
- a. The N predefined spatial neighbouring blocks can be predefined as claimed in Bullet 6.
 - b. The M entries in the HMVP table can be predefined as claimed in Bullet 7.
 - c. TMVP can be derived in the same way as in HEVC or VVC.
 - d. In one example, only one pair-wise average merge candidate can be built with motion information fetched from the N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs.
 - i. For example, each pair of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs are checked in order, until a pair-wise average merge candidate can be built with motion information from one pair of N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs.
 - ii. In one example, if no motion information can be fetched from the N predefined spatial neighbouring blocks and/or M entries in the HMVP

- table, and/or P TMVPs then no pair-wise average merge candidate can be built.
- iii. In one example, if motion information can only be fetched from one of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs, then Bullet 1 can be applied.
 - iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs or not.
- e. In one example, S ($S > 1$) pair-wise average merge candidates can be built with motion information fetched from N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs.
- i. For example, each pair of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs are checked in order, until the S pair-wise average merge candidates can be built with motion information from S pairs of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs.
 - ii. In one example, if no motion information can be fetched from the N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs, then no pair-wise average merge candidate can be built.
 - iii. In one example, if motion information can only be fetched from one of the N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs, then Bullet 1 can be applied.
 - iv. Bullet 3 can be applied to determine whether a pair-wise average merge candidate can be built with motion information from one pair of N predefined spatial neighbouring blocks and/or M entries in the HMVP table and/or P TMVPs or not.
10. In one example, if no pair-wise average merge candidate can be built (e.g., all blocks used for pair-wise average candidate derivation are intra-coded), then one or multiple default merge candidates are put into the merge candidate list.

- a. In one example, default merge candidate(s) are put into the merge candidate list before the zero candidate.
 - i. Alternatively, default merge candidate(s) are put into the merge candidate list after the zero candidate.
 - b. In one example, default merge candidate(s) may be predefined.
 - i. For example, default merge candidate(s) may be (S, 0) or (0, S);
 - 1) Alternatively, default merge candidate(s) may be (S, S) or (S, -S);
 - ii. In the above examples, S is a non-zero integer such as N or -N, where N may be 1, 2, 3, 4... In another example, S is a non-zero integer such as 2N or -2N, where N may be 1, 2, 3, 4... In still another example, S is a non-zero integer such as (2N-1) or -(2N-1), where N may be 1, 2, 3, 4... In still another example, S is a non-zero integer such as 2^N or -2^N , where N may be 1, 2, 3, 4...
 - c. In one example, default merge candidate(s) may be signaled from encoder to decoder.
 - d. In one example, MV of a default merge candidate must refer to an integer position.
 - i. For example, MV of a default merge candidate must refer to an integer position for an intra-block copy (IBC) merge candidate list.
 - e. In one example, if pair-wise average candidate couldn't found from predefined blocks, it may be derived from HMVP table.
 - f. Alternatively, the insertion of pair-wise average candidate may be skipped.
11. The methods disclosed in this document can be applied to build the pair-wise average merge candidate(s) in any kind of merge candidate list.
- a. For example, they can be used to build the pair-wise average merge candidate(s) in regular merge candidate list.
 - b. For example, they can be used to build the pair-wise average merge candidate(s) in triangular merge candidate list.
 - c. For example, they can be used to build the pair-wise average merge candidate(s) in IBC merge candidate list.
 - d. For example, they can be used to build the pair-wise average merge candidate(s) in sub-block-based merge candidate list.
 - e. For example, they can be used to build the pair-wise average merge candidate(s) in affine merge candidate list.
12. In one example, a pair-wise merge candidate derived from above methods may be further modified before being added to the candidate list.
- a. In one example, for an IBC merge candidate list, the MV of a pair-wise merge candidate derived with proposed methods should be rounded to integer precision. Suppose the storage MV precision is P, MV_pair is the MV of a pair-wise merge

candidate derived with proposed methods, then MV_pair' which is rounded from MV_pair is put into the IBC merge candidate list.

- b. In one example, $MV_pari' = \text{Shift}(MV_pair, P) \ll P$;
 - c. In one example, $MV_pari' = \text{SatShift}(MV_pair, P) \ll P$;
 - d. In one example, $MV_pari' = MV_pair \& (-1 \ll P)$.
13. In one example, the pair-wise merge candidate derivation methods disclosed in this document can be applied when the pair-wise merge candidate(s) are in the regular merge candidate list.
- a. Alternatively, the pair-wise merge candidate derivation methods disclosed in this document can be applied when the pair-wise merge candidate(s) are in the intra block copy (IBC) merge candidate list.
 - b. Alternatively, the pair-wise merge candidate derivation methods disclosed in this document can be applied when the pair-wise merge candidate(s) are in the sub-block based merge candidate list.

[00125] 6. Examples of syntax structures for certain embodiments

An exemplary syntax design in an embodiment is disclosed as below. The section numbers refer to the current release standard of the VVC document. Examples of changes to the syntax to implement some embodiments are highlighted using bold faced italicized text.

8.5.2.2 Derivation process for luma motion vectors for merge mode

This process is only invoked when $\text{merge_flag}[x_{Cb}][y_{Pb}]$ is equal to 1, where (x_{Cb}, y_{Cb}) specify the top-left sample of the current luma coding block relative to the top-left luma sample of the current picture.

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,
- a variable $cbWidth$ specifying the width of the current coding block in luma samples,
- a variable $cbHeight$ specifying the height of the current coding block in luma samples.

Outputs of this process are:

- the luma motion vectors in 1/16 fractional-sample accuracy $mvL0[0][0]$ and $mvL1[0][0]$,
- the reference indices $refIdxL0$ and $refIdxL1$,
- the prediction list utilization flags $predFlagL0[0][0]$ and $predFlagL1[0][0]$,
- the bi-prediction weight index $gbIdx$.

The bi-prediction weight index $gbIdx$ is set equal to 0.

The variables x_{Smr} , y_{Smr} , $smrWidth$, $smrHeight$, and $smrNumHmvpCand$ are derived as follows:

$$x_{Smr} = \text{IsInSmr}[x_{Cb}][y_{Cb}] ? \text{SmrX}[x_{Cb}][y_{Cb}] : x_{Cb} \quad (8-276)$$

$$ySmr = IsInSmr[xCb][yCb] ? SmrY[xCb][yCb] : yCb \quad (8-277)$$

$$smrWidth = IsInSmr[xCb][yCb] ? SmrW[xCb][yCb] : cbWidth \quad (8-278)$$

$$smrHeight = IsInSmr[xCb][yCb] ? SmrH[xCb][yCb] : cbHeight \quad (8-279)$$

$$\begin{aligned} & smrNumHmvpCand \\ & IsInSmr[xCb][yCb] ? NumHmvpSmrCand : NumHmvpCand \end{aligned} \quad (8-280) =$$

The motion vectors $mvL0[0][0]$ and $mvL1[0][0]$, the reference indices $refIdxL0$ and $refIdxL1$ and the prediction utilization flags $predFlagL0[0][0]$ and $predFlagL1[0][0]$ are derived by the following ordered steps:

1. The derivation process for merging candidates from neighbouring coding units as specified in clause 8.5.2.3 is invoked with the luma coding block location (xCb, yCb) set equal to $(xSmr, ySmr)$, the luma coding block width $cbWidth$, and the luma coding block height $cbHeight$ set equal to $smrWidth$ and $smrHeight$ as inputs, and the output being the availability flags $availableFlagA_0$, $availableFlagA_1$, $availableFlagB_0$, $availableFlagB_1$ and $availableFlagB_2$, the reference indices $refIdxLXA_0$, $refIdxLXA_1$, $refIdxLXB_0$, $refIdxLXB_1$ and $refIdxLXB_2$, the prediction list utilization flags $predFlagLXA_0$, $predFlagLXA_1$, $predFlagLXB_0$, $predFlagLXB_1$ and $predFlagLXB_2$, and the motion vectors $mvLXA_0$, $mvLXA_1$, $mvLXB_0$, $mvLXB_1$ and $mvLXB_2$, with X being 0 or 1, and the bi-prediction weight indices $gbIdxA_0$, $gbIdxA_1$, $gbIdxB_0$, $gbIdxB_1$, $gbIdxB_2$.
2. The reference indices, $refIdxLXCol$, with X being 0 or 1, and the bi-prediction weight index $gbIdxCol$ for the temporal merging candidate Col are set equal to 0.
3. The derivation process for temporal luma motion vector prediction as specified in in clause 8.5.2.11 is invoked with the luma location (xCb, yCb) set equal to $(xSmr, ySmr)$, the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ set equal to $smrWidth$ and $smrHeight$ and the variable $refIdxL0Col$ as inputs, and the output being the availability flag $availableFlagL0Col$ and the temporal motion vector $mvL0Col$. The variables $availableFlagCol$, $predFlagL0Col$ and $predFlagL1Col$ are derived as follows:

$$availableFlagCol = availableFlagL0Col \quad (8-281)$$

$$predFlagL0Col = availableFlagL0Col \quad (8-282)$$

$$predFlagL1Col = 0 \quad (8-283)$$

4. When $tile_group_type$ is equal to B, the derivation process for temporal luma motion vector prediction as specified in clause 8.5.2.11 is invoked with the luma location (xCb, yCb) set equal to $(xSmr, ySmr)$, the luma coding block width $cbWidth$, the luma coding block height $cbHeight$ set equal to $smrWidth$ and $smrHeight$ and the variable

refIdxL1Col as inputs, and the output being the availability flag availableFlagL1Col and the temporal motion vector mvL1Col. The variables availableFlagCol and predFlagL1Col are derived as follows:

$$\text{availableFlagCol} = \text{availableFlagL0Col} \mid \mid \text{availableFlagL1Col} \quad (8-284)$$

$$\text{predFlagL1Col} = \text{availableFlagL1Col} \quad (8-285)$$

5. The merging candidate list, mergeCandList, is constructed as follows:

$$\begin{aligned} & i = 0 \\ & \text{if}(\text{availableFlagA}_1) \\ & \quad \text{mergeCandList}[i++] = \text{A}_1 \\ & \text{if}(\text{availableFlagB}_1) \\ & \quad \text{mergeCandList}[i++] = \text{B}_1 \\ & \text{if}(\text{availableFlagB}_0) \\ & \quad \text{mergeCandList}[i++] = \text{B}_0 \quad (8-286) \\ & \text{if}(\text{availableFlagA}_0) \\ & \quad \text{mergeCandList}[i++] = \text{A}_0 \\ & \text{if}(\text{availableFlagB}_2) \\ & \quad \text{mergeCandList}[i++] = \text{B}_2 \\ & \text{if}(\text{availableFlagCol}) \\ & \quad \text{mergeCandList}[i++] = \text{Col} \end{aligned}$$

6. The variable numCurrMergeCand and numOrigMergeCand are set equal to the number of merging candidates in the mergeCandList.
7. When numCurrMergeCand is less than (MaxNumMergeCand – 1) and smrNumHmvpcand is greater than 0, the following applies:
- The derivation process of history-based merging candidates as specified in 8.5.2.6 is invoked with mergeCandList, isInSmr set equal to IsInSmr[xCb][yCb], and numCurrMergeCand as inputs, and modified mergeCandList and numCurrMergeCand as outputs.
 - numOrigMergeCand is set equal to numCurrMergeCand.
8. *When numCurrMergeCand is less than MaxNumMergeCand and availableFlagA₁ || availableFlagB₁ is not equal to FALSE, the following applies:*
- *The derivation process for pairwise average merging candidate specified in clause 8.5.2.4 is invoked with mergeCandList, the reference indices refIdxL0N and refIdxL1N, the prediction list utilization flags predFlagL0N and predFlagL1N, the motion vectors mvL0N and mvL1N with N being A₁ and B₁, availability flags availableFlagA₁ and availableFlagB₁, numCurrMergeCand and numOrigMergeCand as inputs, and the output is assigned to mergeCandList, numCurrMergeCand, the reference indices refIdxL0avgCand and*

refIdxL1avgCand, the prediction list utilization flags predFlagL0avgCand and predFlagL1avgCand and the motion vectors mvL0avgCand and mvL1avgCand of candidate avgCand being added into mergeCandList. The bi-prediction weight index gbiIdx of candidate avgCand being added into mergeCandList is set equal to 0.

- *numOrigMergeCand is set equal to numCurrMergeCand.*
- ...

8.5.2.4 Derivation process for pairwise average merging candidate

Inputs to this process are:

- *a merging candidate list mergeCandList,*
- *the reference indices refIdxL0N and refIdxL1N with N being A₁ and B₁,*
- *the prediction list utilization flags predFlagL0N and predFlagL1N with N being A₁ and B₁,*
- *the motion vectors in 1/16 fractional-sample accuracy mvL0N and mvL1N with N being A₁ and B₁,*
- *availability flags availableFlagA₁ and availableFlagB*
- *the number of elements numCurrMergeCand within mergeCandList.*

Outputs of this process are:

- *the merging candidate list mergeCandList,*
- *the number of elements numCurrMergeCand within mergeCandList,*
- *the reference indices refIdxL0avgCand and refIdxL1avgCand of candidate avgCand added into mergeCandList during the invocation of this process,*
- *the prediction list utilization flags predFlagL0avgCand and predFlagL1avgCand of candidate avgCand added into mergeCandList during the invocation of this process,*
- *the motion vectors in 1/16 fractional-sample accuracy mvL0avgCand and mvL1avgCand of candidate avgCand added into mergeCandList during the invocation of this process.*

The variable numRefLists is derived as follows:

$$\text{numRefLists} = (\text{tile_group_type} == \text{B}) ? 2 : 1 \quad (8-314)$$

The following assignments are made, with p0Cand being the candidate at position 0 and p1Cand being the candidate at position 1 in the merging candidate list mergeCandList:

$$p0Cand = A_1 \quad (8-315)$$

$$p1Cand = B_1 \quad (8-316)$$

The candidate avgCand is added at the end of mergeCandList, i.e., mergeCandList[numCurrMergeCand] is set equal to avgCand, and the reference indices, the

prediction list utilization flags and the motion vectors of avgCand are derived as follows and numCurrMergeCand is incremented by 1:

If availableFlagA₁ is equal to TRUE and availableFlagB₁ is equal to TRUE, the following applies:

- For each reference picture list LX with X ranging from 0 to (numRefLists – 1), the following applies:
- If predFlagLXp0Cand is equal to 1 and predFlagLXp1Cand is equal to 1, the variables refIdxLXavgCand, predFlagLXavgCand, mvLXavgCand[0], and mvLXavgCand[1] are derived as follows:

$$\text{refIdxLXavgCand} = \text{refIdxLXp0Cand} \quad (8-317)$$

$$\text{predFlagLXavgCand} = 1 \quad (8-318)$$

$$\text{mvLXavgCand}[0] = (\text{mvLXp0Cand}[0] + \text{mvLXp1Cand}[0]) / 2 \quad (8-319)$$

$$\text{mvLXavgCand}[1] = (\text{mvLXp0Cand}[1] + \text{mvLXp1Cand}[1]) / 2 \quad (8-320)$$

- Otherwise, if predFlagLXp0Cand is equal to 1 and predFlagLXp1Cand is equal to 0, the variables refIdxLXavgCand, predFlagLXavgCand, mvLXavgCand[0], mvLXavgCand[1] are derived as follows:

$$\text{refIdxLXavgCand} = \text{refIdxLXp0Cand} \quad (8-321)$$

$$\text{predFlagLXavgCand} = 1 \quad (8-322)$$

$$\text{mvLXavgCand}[0] = \text{mvLXp0Cand}[0] \quad (8-323)$$

$$\text{mvLXavgCand}[1] = \text{mvLXp0Cand}[1] \quad (8-324)$$

- Otherwise, if predFlagLXp0Cand is equal to 0 and predFlagLXp1Cand is equal to 1, the variables refIdxLXavgCand, predFlagLXavgCand, mvLXavgCand[0], mvLXavgCand[1] are derived as follows:

$$\text{refIdxLXavgCand} = \text{refIdxLXp1Cand} \quad (8-325)$$

$$\text{predFlagLXavgCand} = 1 \quad (8-326)$$

$$\text{mvLXavgCand}[0] = \text{mvLXp1Cand}[0] \quad (8-327)$$

$$\text{mvLXavgCand}[1] = \text{mvLXp1Cand}[1] \quad (8-328)$$

- Otherwise, if predFlagLXp0Cand is equal to 0 and predFlagLXp1Cand is equal to 0, the variables refIdxLXavgCand, predFlagLXavgCand, mvLXavgCand[0], mvLXavgCand[1] are derived as follows:

$$\text{refIdxLXavgCand} = -1(8-329)$$

$$\text{predFlagLXavgCand} = 0(8-330)$$

$$\text{mvLXavgCand}[0] = 0(8-331)$$

$$\text{mvLXavgCand}[1] = 0(8-332)$$

- When numRefLists is equal to 1, the following applies:

$$\text{refIdxL1avgCand} = -1(8-333)$$

$$\text{predFlagL1avgCand} = 0(8-334)$$

Otherwise, the following applies:

- *If availableFlagB₁ is equal to TRUE, p0Cand is set equal B₁.*
- *For each reference picture list LX with X ranging from 0 to (numRefLists – 1), the following applies::*
 - *If predFlagLXp0Cand is equal to 1 and predFlagLXp1Cand is equal to 1, the variables refIdxLXavgCand, predFlagLXavgCand, mvLXavgCand[0], and mvLXavgCand[1] are derived as follows:*

$$\text{refIdxLXavgCand} = \text{refIdxLXp0Cand} \quad (8-317)$$

$$\text{predFlagLXavgCand} = 1(8-318)$$

$$\text{mvLXavgCand}[0] = \text{mvLXp0Cand}[0] \geq 0 ? ((\text{mvLXp0Cand}[0] + 1) \gg 1) : -(\text{mvLXp0Cand}[0] + 1) \gg 1 \quad (8-319)$$

$$\text{mvLXavgCand}[1] = \text{mvLXp0Cand}[1] \geq 0 ? ((\text{mvLXp0Cand}[1] + 1) \gg 1) : -(\text{mvLXp0Cand}[1] + 1) \gg 1 \quad (8-320)$$

[00126] FIG. 1000 is a block diagram of a video processing apparatus 1000. The apparatus 1000 may be used to implement one or more of the methods described herein. The apparatus 1000 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1000 may include one or more processors 1002, one or more memories 1004 and video processing hardware 1006. The processor(s) 1002 may be configured to implement one or more methods described in the present document. The memory (memories) 1004 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1006 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00127] FIG. 11 is a flowchart for an example method 1100 of video processing. The method

1100 comprises: generating(1102) a pair-wise average merge candidate based on at least one available set of motion information; inserting(1104) the pair-wise average merge candidate into a merge candidate list for a current video block; and performing a conversion(1106) between the current video block and a bitstream representation of the current video block using the merge candidate list..

[00128] FIG. 12 is a flowchart for an example method 1200 of video processing. The method 1200 includes determining(1202) whether a pair-wise average merge candidate can be derived based on at least one set of motion information; inserting(1204) at least one default merge candidate into a merge candidate list for a current video block in response to determining that no pair-wise average merge candidate can be derived; and performing a conversion(1206) between the current video block and a bitstream representation of the current video block using the merge candidate list.

[00129] Additional embodiments and techniques can as described in the following examples.

[00130] In one aspect, a method of video processing is disclosed, the method comprising: generating a pair-wise average merge candidate based on at least one available set of motion information; inserting the pair-wise average merge candidate into a merge candidate list for a current video block; and performing a conversion between the current video block and a bitstream representation of the current video block using the merge candidate list.

[00131] In one example, said at least one available set of motion information is fetched from a neighboring block or an entry in a history-based motion vector prediction (HMVP) table.

[00132] In one example, the neighboring block comprises one of a spatial neighboring block and a temporal neighboring block.

[00133] In one example, the at least one available set of motion information includes at least one of an inter-prediction direction, a reference index for a reference picture list X, a motion vector (MV), and a Bi-prediction with CU-level Weights (BCW) index, wherein $X = 0, 1$.

[00134] In one example, at least one of the inter-prediction direction, the reference index for the reference picture list X, and the Bi-prediction with CU-level Weights (BCW) index is copied to the pair-wise average merge candidate to generate corresponding parameters of the pair-wise average merge candidate.

[00135] In one example, a BCW index of the pair-wise average merge candidate is set to be a

default value if the BCW of the pair-wise average merge candidate is disabled.

[00136] In one example, the default value is equal to zero.

[00137] In one example, the MV of the at least one available set of motion information comprises a first component and a second component, and a first function f_1 and a second function f_2 are applied to the first and second components of the MV respectively to generate an MV of the pair-wise average merge candidate.

[00138] In one example, the first function $f_1 = m + offset$, and $offset$ is an integer number, m representing a corresponding component.

[00139] In one example, the first function $f_1 = m + (offset \ll Precise)$, and $offset$ is an integer number, $Precise$ defines a MV precision, m representing a corresponding component.

[00140] In one example, $offset = 3$.

[00141] In one example, $Precise = 4$.

[00142] In one example, the first function $f_1 = Shift(m, k)$, and $Shift(m, k) = (m + offset0) \gg k$, and k is an integer, m representing a corresponding component.

[00143] In one example, the first function $f_1 = SatShift(m, k)$, and

$$Satshift(m, k) = \begin{cases} (m + offset0) \gg k & \text{if } m \geq 0 \\ -((-m + offset1) \gg k) & \text{if } m < 0 \end{cases}$$

wherein k is an integer, m representing a corresponding component.

[00144] In one example,

$$offset0 = 0;$$

$$offset0 = (1 \ll n) \gg 1; \text{ or}$$

$$offset0 = 1 \ll (n-1).$$

[00145] In one example,

$$offset1 = 0;$$

$$offset1 = (1 \ll n) \gg 1; \text{ or}$$

$$offset1 = 1 \ll (n-1).$$

[00146] In one example, the first function $f1 = m \ll k$, and k is an integer, m representing a

corresponding component.

[00147] In one example, where $k=1$.

[00148] In one example, the second function f_2 is identical to the first function f_1 .

[00149] In one example, the MV of the at least one available set of motion information refers to the reference picture list X , wherein $X=0$ or 1 .

[00150] In one example, the method further comprises

constructing a set of motion information from said at least one available set of motion information, and

generating the pair-wise average merge candidate from said at least one available set of motion information and the constructed set of motion information.

[00151] In one example, an inter-prediction direction of the constructed set of motion information is set equal to that of the at least one available set of motion information.

[00152] In one example, a reference index for reference picture list X of the constructed set of motion information is set equal to that of the at least one available set of motion information, wherein $X=0$ or 1 .

[00153] In one example, a motion vector for the reference picture list X of the constructed set of motion information is set to zero.

[00154] In one example, the merge candidate list comprises only one available merge candidate before the pair-wise average merge candidate is inserted.

[00155] In one example, the at least one available set of motion information is selected from one of a first set of motion information and a second set of motion information which are identical or similar to each other.

[00156] In one example, the other of the first set of motion information and the second set of motion information, which is not selected as the at least one available set of motion information, is treated as unavailable.

[00157] In one example, the pair-wise average merge candidate is modified before being inserted into the merge candidate list.

[00158] In one example, the merge candidate list is an intra block copy (IBC) merge candidate list, and the pair-wise average merge candidate is modified by rounding an MV of the pair-wise average merge candidate to an integer precision.

[00159] In one example, the MV of the pair-wise average merge candidate is rounded as one of:

$$\text{MV_pari}' = \text{Shift}(\text{MV_pair}, P) \ll P;$$

$$\text{MV_pari}' = \text{SatShift}(\text{MV_pair}, P) \ll P; \text{ or}$$

$$\text{MV_pari}' = \text{MV_pair} \& (-1 \ll P),$$

wherein MV_pair represents the MV of the pair-wise average merge candidate, P represents a storage precision, and MV_pair' represents the rounded pair-wise average merge candidate in the IBC merge candidate list.

[00160] In another aspect, a method of video processing is disclosed, the method comprising:

determining whether a pair-wise average merge candidate can be derived based on at least one set of motion information;

inserting at least one default merge candidate into a merge candidate list for a current video block in response to determining that no pair-wise average merge candidate can be derived; and

performing a conversion between the current video block and a bitstream representation of the current video block using the merge candidate list.

[00161] In one example, no pair-wise average merge candidate can be derived if at least one set of motion information indicates that all blocks to be used for deriving the pair-wise average merge candidate are intra coded.

[00162] In one example, no pair-wise average merge candidate can be derived if only one set of motion information is available in the merging candidate list before inserting the pair-wise average candidate.

[00163] In one example, the at least one default merge candidate is inserted into the merge candidate list before candidates with zero MVs or after the candidates with zero MVs.

[00164] In one example, the at least one default merge candidate is predefined.

[00165] In one example, the at least one default merge candidate is one of:

(S, 0);

(0, S);

(S, S);

(S, -S), wherein S is a non-zero integer.

[00166] In one example,

$S = N$;

$S = -N$;

$S = 2N$;

$S = -2N$;

$S = (2N-1)$;

$S = -(2N-1)$;

$S = 2^N$;

$S = -2^N$, where N is a natural number.

[00167] In one example, the at least one default merge candidate is signaled from an encoding side to a decoding side.

[00168] In one example, an MV of the at least one default merge candidate refers to an integer position for a specific merge candidate list.

[00169] In one example, the specific merge candidate list comprises an intra block copy (IBC) merge candidate list.

[00170] In one example, the pair-wise average merge candidate can be derived from entries of HMVP table if no pair-wise average merge candidate can be derived from one or more predefined block.

[00171] In one example, the merge candidate list belongs to one of:

a regular merge candidate list;

a triangular merge candidate list;

an intra block copy (IBC) merge candidate list;

a sub-block based merge candidate list; and
an affine merge candidate list.

[00172] In one example, the conversion includes encoding the current video block into the bitstream representation of a video and decoding the current video block from the bitstream representation of the video.

[00173] In an aspect, an apparatus in a video system is disclosed, the apparatus comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method as described above.

[00174] In an aspect, a non-transitory computer readable media is disclosed, the non-transitory computer readable media having program code stored thereupon, the program code, when executed, causing a processor to implement the method as described above.

[00175] It will be appreciated that the disclosed techniques may be embodied in video encoders or decoders to improve compression efficiency using techniques that include the use of pair-wise averaging to generate merge candidates during video coding or decoding operation.

[00176] 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.

[00177] 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.

[00178] 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).

[00179] 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.

[00180] 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.

[00181] 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.

[00182] 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 of video processing, comprising:
 - generating a pair-wise average merge candidate based on at least one available set of motion information;
 - inserting the pair-wise average merge candidate into a merge candidate list for a current video block; and
 - performing a conversion between the current video block and a bitstream representation of the current video block using the merge candidate list.
2. The method of claim 1, wherein said at least one available set of motion information is fetched from a neighboring block or an entry in a history-based motion vector prediction (HMVP) table.
3. The method of claim 2, wherein the neighboring block comprises one of a spatial neighboring block and a temporal neighboring block.
4. The method of any one of claims 1-3, wherein the at least one available set of motion information includes at least one of an inter-prediction direction, a reference index for a reference picture list X, a motion vector (MV), and a Bi-prediction with CU-level Weights (BCW) index, wherein $X = 0, 1$.
5. The method of claim 4, wherein at least one of the inter-prediction direction, the reference index for the reference picture list X, and the Bi-prediction with CU-level Weights (BCW) index is copied to the pair-wise average merge candidate to generate corresponding parameters of the pair-wise average merge candidate.
6. The method of claim 5, wherein a BCW index of the pair-wise average merge candidate is set to be a default value if the BCW of the pair-wise average merge candidate is disabled.

7. The method of claim 6, wherein the default value is equal to zero.
8. The method of claim 3, wherein the MV of the at least one available set of motion information comprises a first component and a second component, and a first function f_1 and a second function f_2 are applied to the first and second components of the MV respectively to generate an MV of the pair-wise average merge candidate.
9. The method of claim 8, wherein the first function $f_1 = m + offset$, and $offset$ is an integer number, m representing a corresponding component.
10. The method of claim 8, wherein the first function $f_1 = m + (offset \ll Precise)$, and $offset$ is an integer number, $Precise$ defines a MV precision, m representing a corresponding component.
11. The method of claim 9 or 10, wherein $offset = 3$.
12. The method of claim 10, wherein $Precise = 4$.
13. The method of claim 8, wherein the first function $f_1 = Shift(m, k)$, and $Shift(m, k) = (m + offset0) \gg k$, and k is an integer, m representing a corresponding component.

14. The method of claim 8, wherein the first function $f_1 = SatShift(m, k)$, and

$$Satshift(m, k) = \begin{cases} (m + offset0) \gg k & \text{if } m \geq 0 \\ -((-m + offset1) \gg k) & \text{if } m < 0 \end{cases}$$

wherein k is an integer, m representing a corresponding component.

15. The method of claim 13 or 14, wherein

$$offset0 = 0;$$

$offset0 = (1 \ll n) \gg 1$; or

$offset0 = 1 \ll (n-1)$.

16. The method of claim 14, wherein

$offset1 = 0$;

$offset1 = (1 \ll n) \gg 1$; or

$offset1 = 1 \ll (n-1)$.

17. The method of claim 8, wherein the first function $f_1 = m \ll k$, and k is an integer, m representing a corresponding component.

18. The method of any one of claims 13-17, where $k=1$.

19. The method of any one of claims 9-18, wherein the second function f_2 is identical to the first function f_1 .

20. The method of any one of claims 8-19, wherein the MV of the at least one available set of motion information refers to the reference picture list X , wherein $X=0$ or 1 .

21. The method of any one of claim 1-20, further comprising:

constructing a set of motion information from said at least one available set of motion information, and

generating the pair-wise average merge candidate from said at least one available set of motion information and the constructed set of motion information.

22. The method of claim 21, wherein an inter-prediction direction of the constructed set of motion information is set equal to that of the at least one available set of motion information.

23. The method of claim 21 or 22, wherein a reference index for reference picture list X of the constructed set of motion information is set equal to that of the at least one available set of motion information, wherein $X=0$ or 1.
24. The method of claim 23, wherein a motion vector for the reference picture list X of the constructed set of motion information is set to zero.
25. The method of any one of claims 1-24, wherein the merge candidate list comprises only one available merge candidate before the pair-wise average merge candidate is inserted.
26. The method of any one of claims 1-25, wherein the at least one available set of motion information is selected from one of a first set of motion information and a second set of motion information which are identical or similar to each other.
27. The method of claim 26, wherein the other of the first set of motion information and the second set of motion information, which is not selected as the at least one available set of motion information, is treated as unavailable.
28. The method of any one of claims 1-27, wherein the pair-wise average merge candidate is modified before being inserted into the merge candidate list.
29. The method of claim 28, wherein the merge candidate list is an intra block copy (IBC) merge candidate list, and the pair-wise average merge candidate is modified by rounding an MV of the pair-wise average merge candidate to an integer precision.
30. The method of claim 29, wherein the MV of the pair-wise average merge candidate is rounded as one of:

$$MV_pari' = \text{Shift}(MV_pair, P) \ll P;$$

$$MV_pari' = \text{SatShift}(MV_pair, P) \ll P; \text{ or}$$

$$MV_pari' = MV_pair \& (-1 \ll P),$$

wherein MV_{pair} represents the MV of the pair-wise average merge candidate, P represents a storage precision, and MV_{pair}' represents the rounded pair-wise average merge candidate in the IBC merge candidate list.

31. A method of video processing, comprising:

determining whether a pair-wise average merge candidate can be derived based on at least one set of motion information;

inserting at least one default merge candidate into a merge candidate list for a current video block in response to determining that no pair-wise average merge candidate can be derived; and

performing a conversion between the current video block and a bitstream representation of the current video block using the merge candidate list.

32. The method of claim 31, wherein no pair-wise average merge candidate can be derived if at least one set of motion information indicates that all blocks to be used for deriving the pair-wise average merge candidate are intra coded.

33. The method of claim 31, wherein no pair-wise average merge candidate can be derived if only one set of motion information is available in the merging candidate list before inserting the pair-wise average candidate.

34. The method of any one of claims 31-33, wherein the at least one default merge candidate is inserted into the merge candidate list before candidates with zero MVs or after the candidates with zero MVs.

35. The method of any one of claims 31-34, wherein the at least one default merge candidate is predefined.

36. The method of claim 35, wherein the at least one default merge candidate is one of:

(S, 0);
(0, S);
(S, S);
(S, -S), wherein S is a non-zero integer.

37. The method of claim 36, wherein

S= N;
S= -N;
S= 2N;
S= -2N;
S= (2N-1);
S= -(2N-1);
S= 2^N ;
S= -2^N , where N is a natural number.

38. The method of any one of claims of 31-37, wherein the at least one default merge candidate is signaled from an encoding side to a decoding side.

39. The method of any one of claims of 31-38, wherein an MV of the at least one default merge candidate refers to an integer position for a specific merge candidate list.

40. The method of claim 39, wherein the specific merge candidate list comprises an intra block copy (IBC) merge candidate list.

41. The method of any one of claims 31-40, wherein the pair-wise average merge candidate can be derived from entries of HMVP table if no pair-wise average merge candidate can be derived from one or more predefined block.

42. The method of any one of claims 1-41, wherein the merge candidate list belongs to one of:
a regular merge candidate list;

a triangular merge candidate list;
an intra block copy (IBC) merge candidate list;
a sub-block based merge candidate list; and
an affine merge candidate list.

43. The method of any of claims 1 to 42, wherein the conversion includes encoding the current video block into the bitstream representation of a video and decoding the current video block from the bitstream representation of the video.

44. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of claims 1 to 43.

45. A non-transitory computer readable media, having program code stored thereupon, the program code, when executed, causing a processor to implement the method in any one of claims 1 to 43.

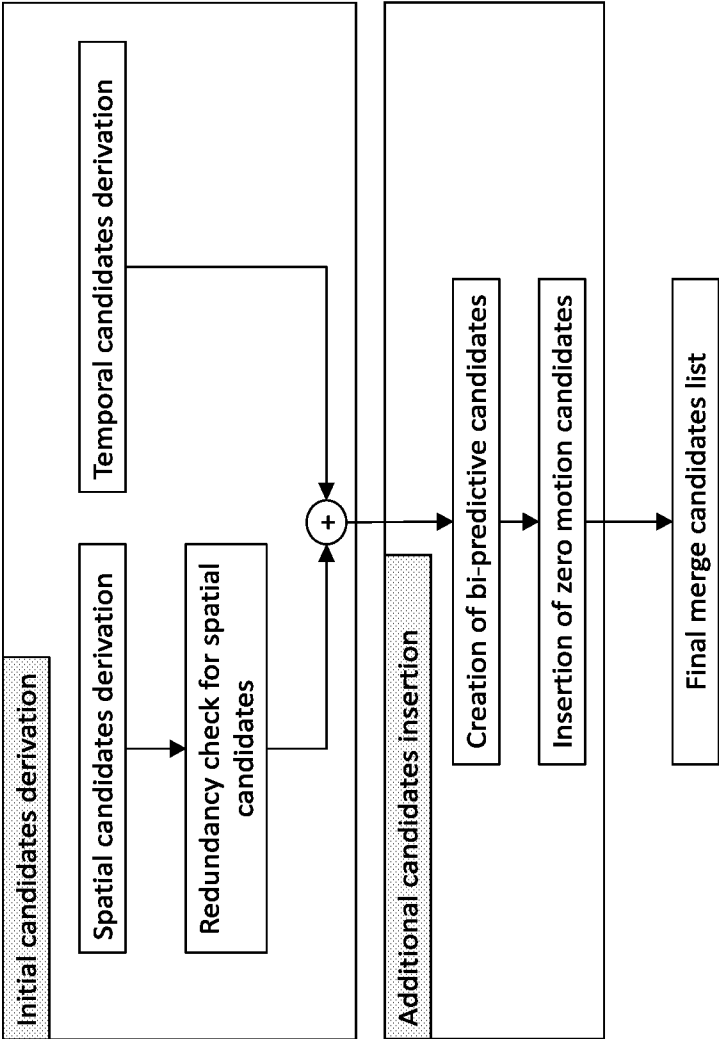


FIG. 1

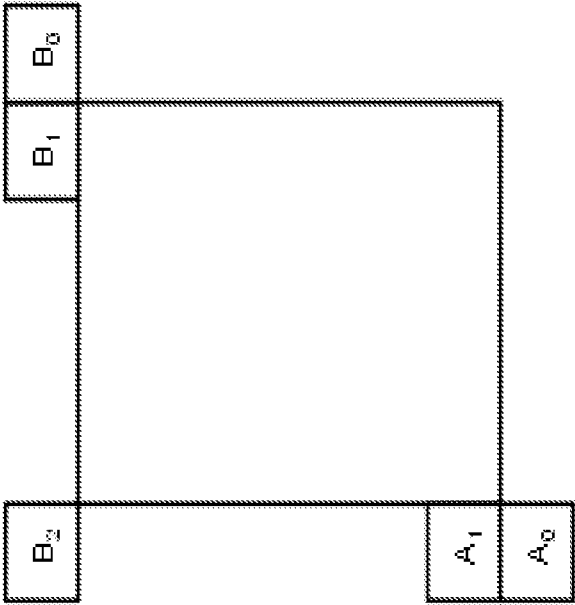


FIG. 2

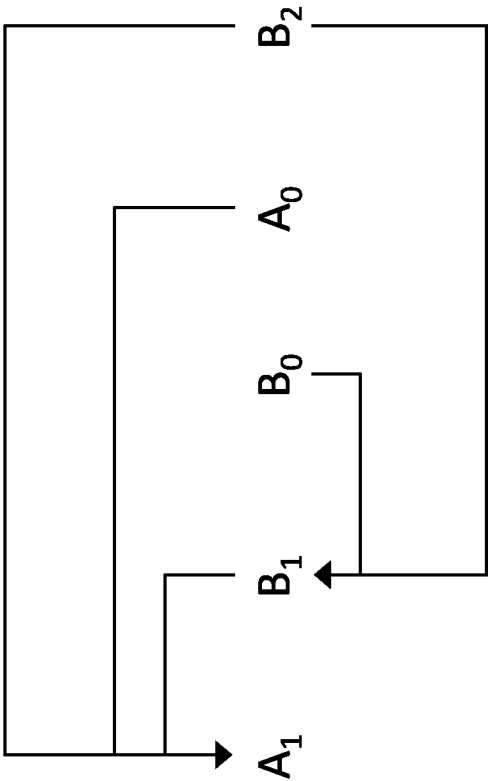


FIG. 3

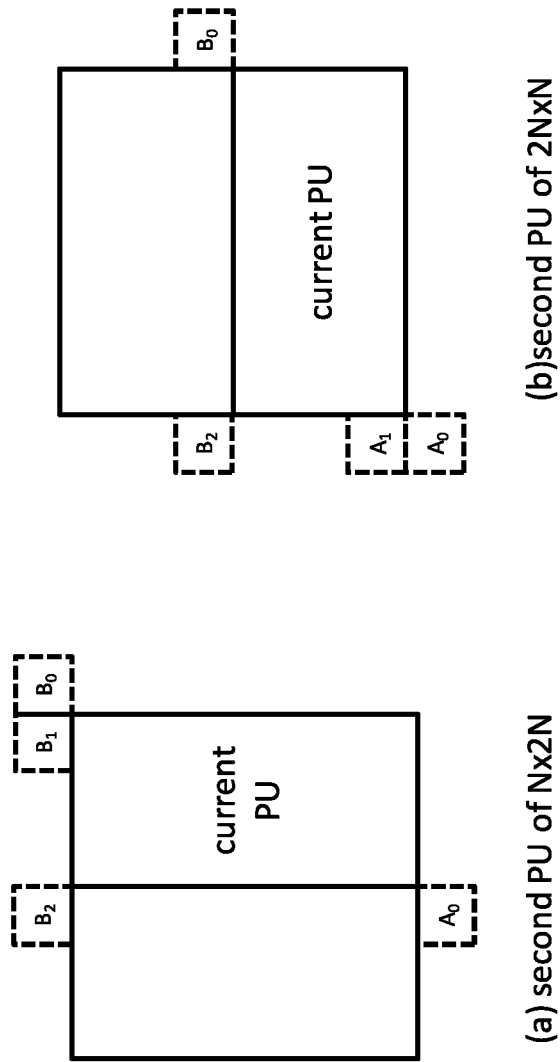


FIG. 4

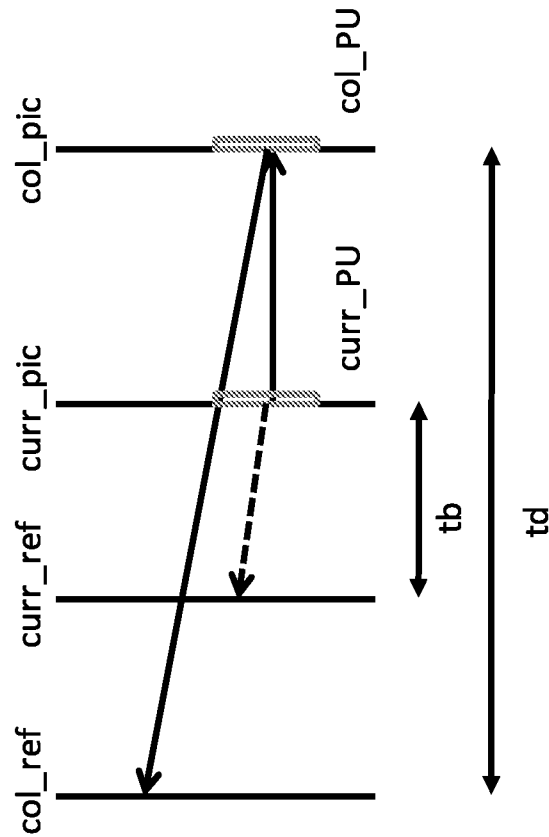


FIG. 5

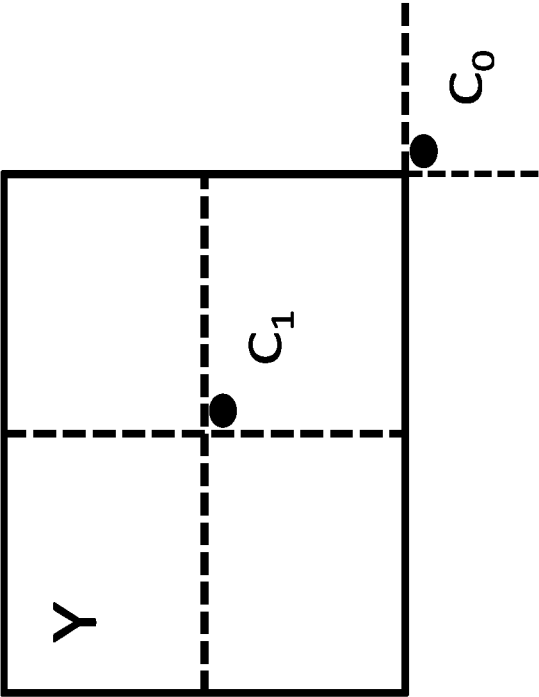


FIG. 6

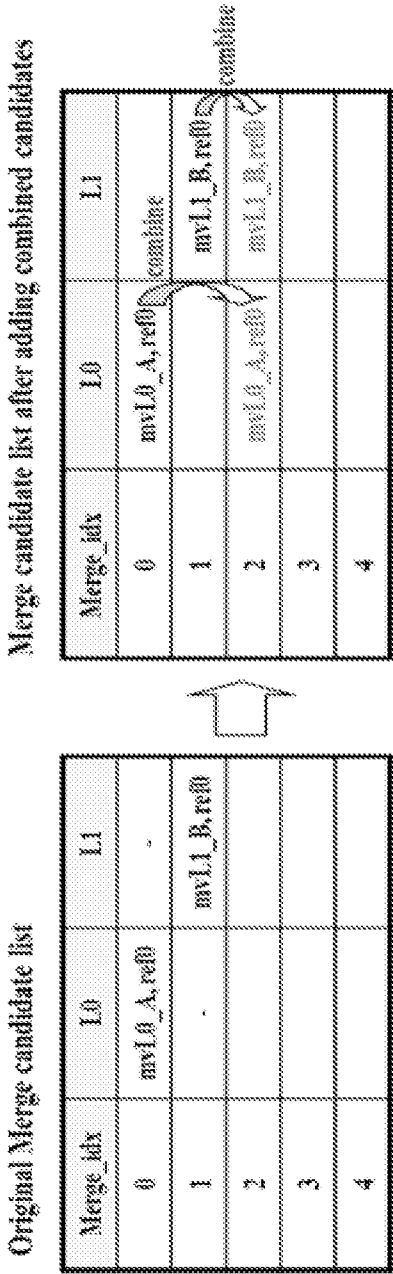


FIG. 7

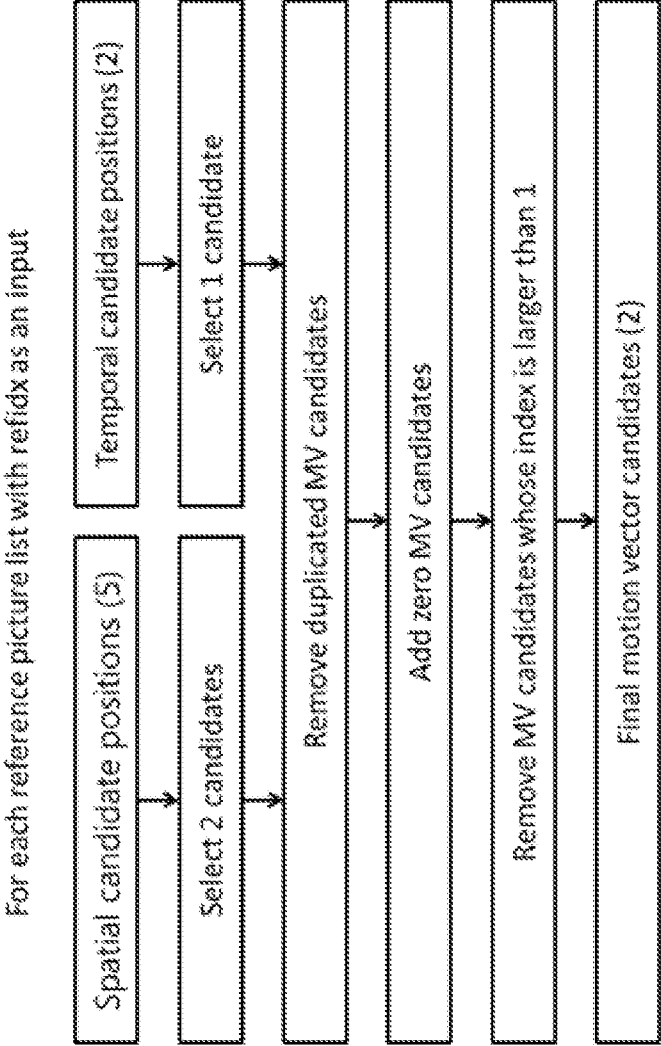


FIG. 8

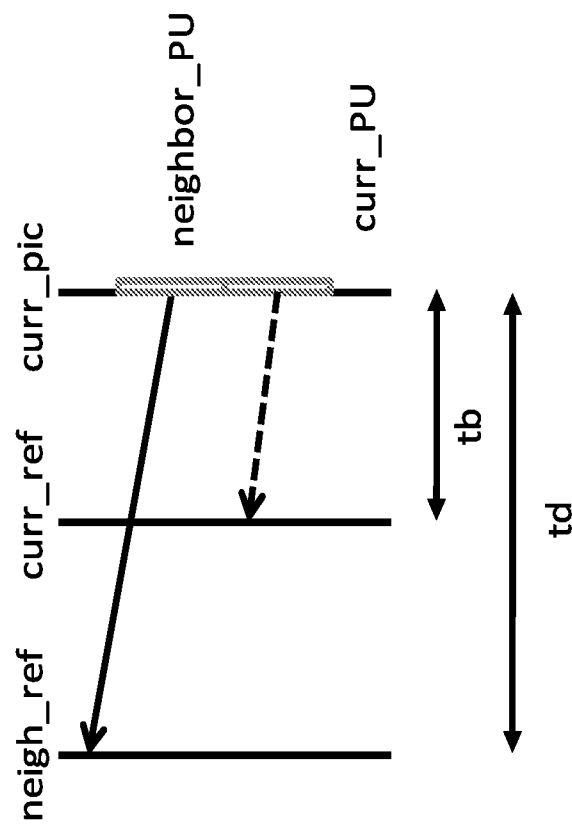


FIG. 9

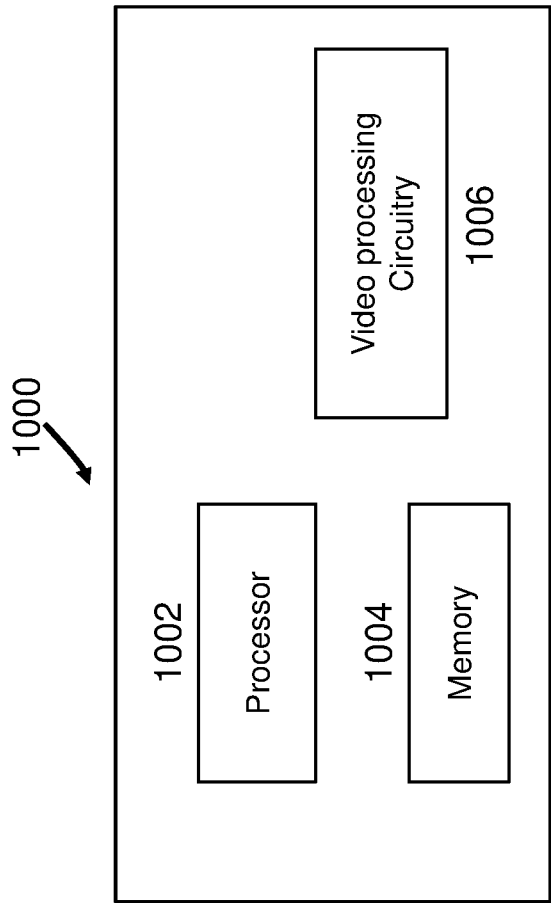


FIG. 10

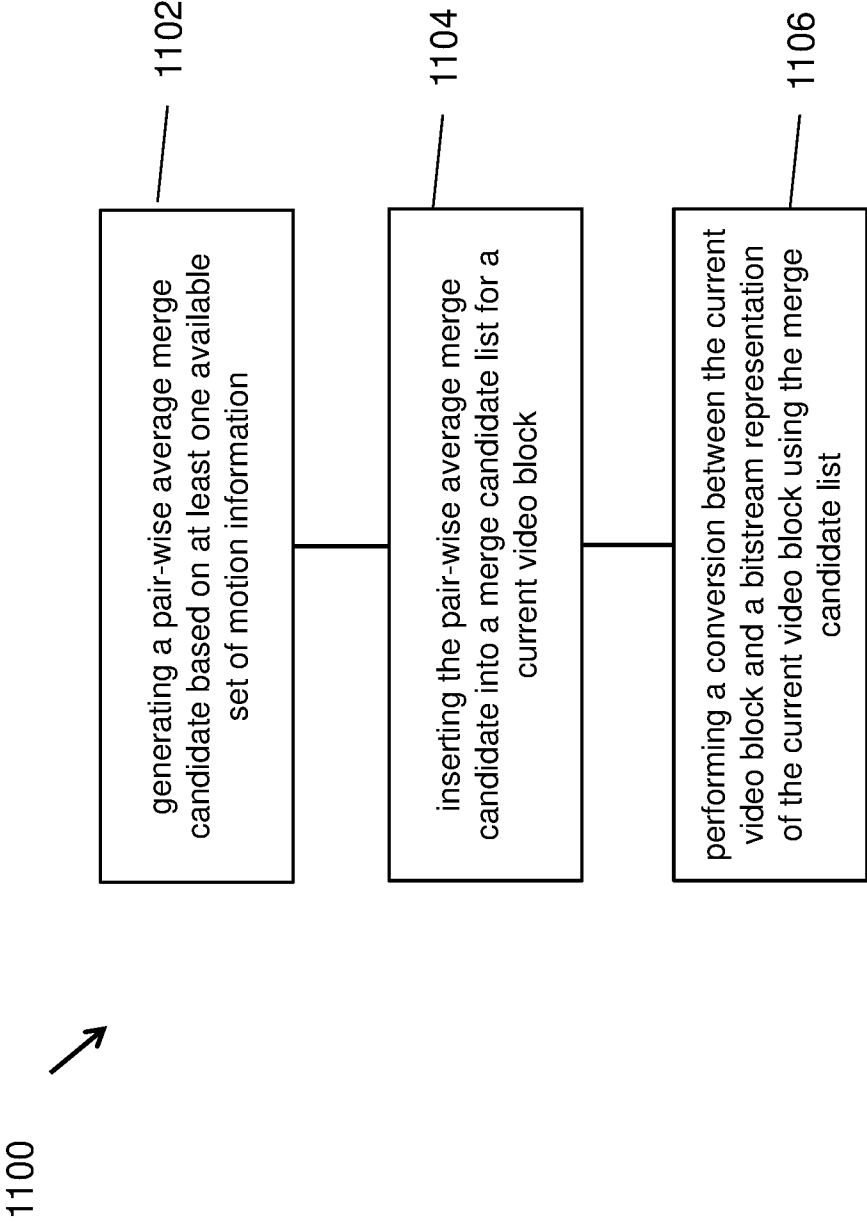
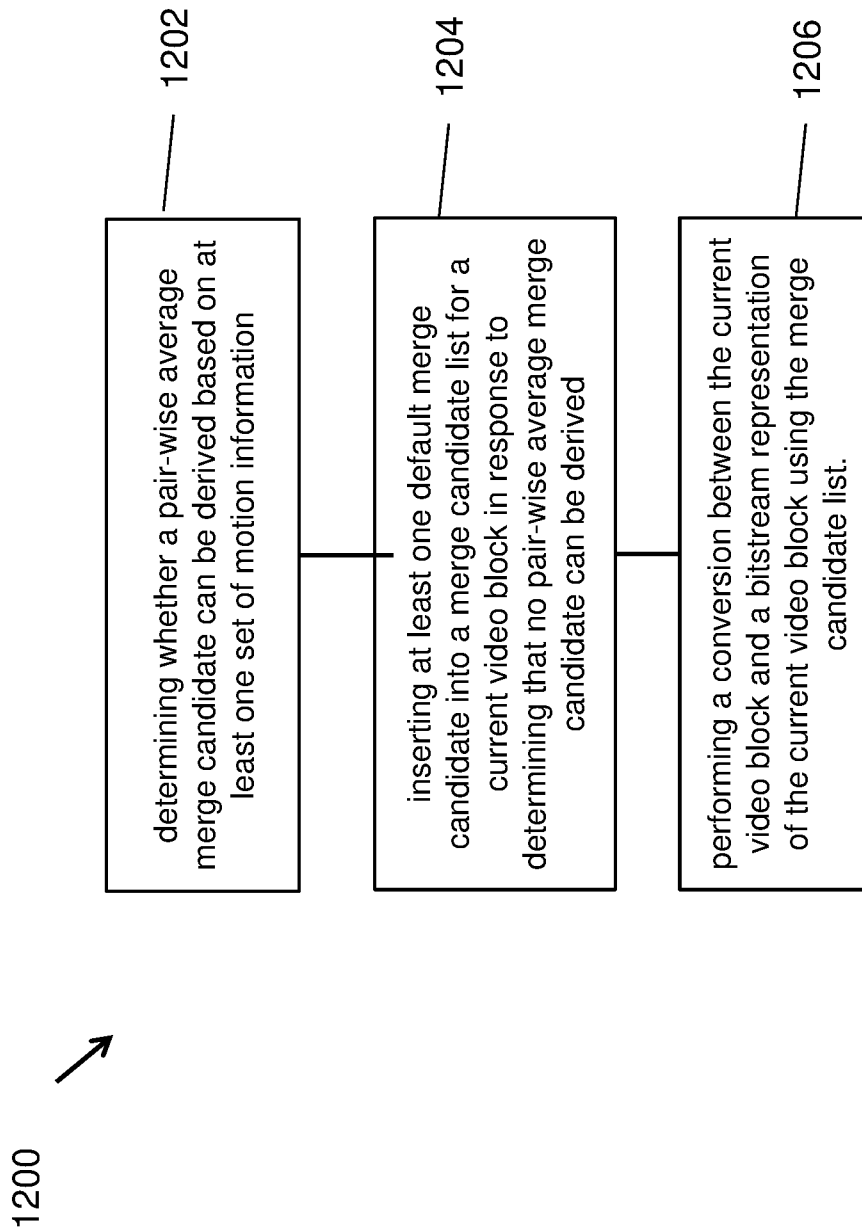


FIG. 11

**FIG. 12**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/078793

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/503(2014.01)i; H04N 19/176(2014.01)i; H04N 19/136(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

CNPAT,CNKI,EPODOC,WPI,IEEE:average, merge, candidate, motion, list, bitstream, block, video, direction, reference,index, CU, BCW, MV

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 110710210 A (KT CORP.) 17 January 2020 (2020-01-17) description, paragraphs 48-57, 183, 193, 201	1, 44, 45
X	CN 108353184 A (MEDIATEK INC.) 31 July 2018 (2018-07-31) description, paragraphs 8, 15, 31, 34, 47, 49, 72, 75	1-30, 44, 45
X	US 2018352223 A1 (MEDIATEK INC.) 06 December 2018 (2018-12-06) description, paragraphs 4, 8, 35, 40, 52	1-30, 44, 45
Y	CN 108353184 A (MEDIATEK INC.) 31 July 2018 (2018-07-31) description, paragraphs 8, 15, 31, 34, 47, 49, 72, 75	31-43
Y	US 2015085930 A1 (QUALCOMM INC.) 26 March 2015 (2015-03-26) description, paragraphs 35, 42	31-43
A	E-G. Mora ET AL. "MODIFICATION OF THE MERGE CANDIDATE LIST FOR DEPENDENT VIEWS IN" IEEE, 31 December 2013 (2013-12-31), the whole document	1-45



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

13 May 2020

Date of mailing of the international search report

29 May 2020

Name and mailing address of the ISA/CN

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

Facsimile No. (86-10)62019451

Authorized officer

XI,Huining

Telephone No. 86-10-53961813

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2020/078793

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	110710210	A	17 January 2020	CA	3067528	A1	25 April 2019
				WO	2019078664	A1	25 April 2019
				KR	20190044533	A	30 April 2019
				AU	2018351405	A1	16 January 2020
CN	108353184	A	31 July 2018	EP	3357245	A1	08 August 2018
				WO	2017076221	A1	11 May 2017
				US	2018324454	A1	08 November 2018
				US	10412407	B2	10 September 2019
US	2018352223	A1	06 December 2018	TW	201904291	A	16 January 2019
				WO	2018219310	A1	06 December 2018
				US	10523934	B2	31 December 2019
US	2015085930	A1	26 March 2015	KR	101835240	B1	06 March 2018
				JP	6151442	B2	21 June 2017
				US	9554150	B2	24 January 2017
				KR	20160058809	A	25 May 2016
				WO	2015042399	A1	26 March 2015
				CN	105580372	B	28 May 2019
				CN	105580372	A	11 May 2016
				EP	3047649	A1	27 July 2016
				JP	2016536817	A	24 November 2016