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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 May 2020 (22.05.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/098752 A1

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

(21) International Application Number:
PCT/CN2019/118528

(22) International Filing Date:
14 November 2019 (14.11.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2018/115354
14 November 2018 (14.11.2018) 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, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **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, 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,

(54) Title: IMPROVEMENTS OF AFFINE PREDICTION MODE

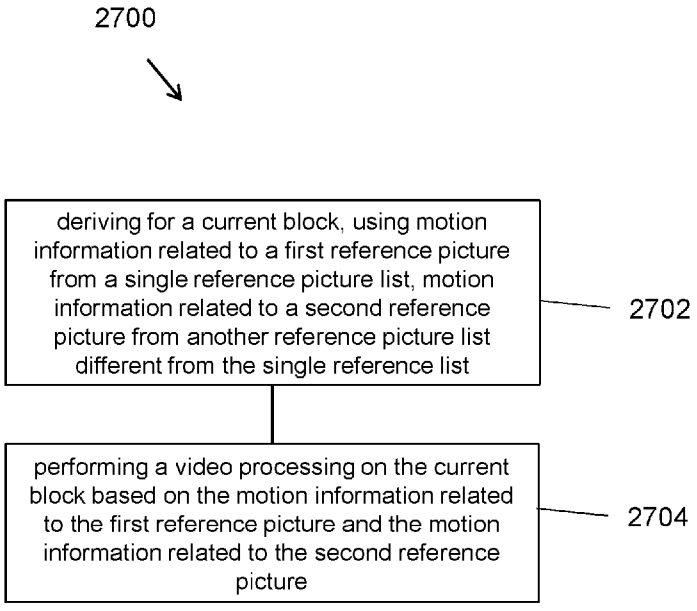


FIG. 27

(57) Abstract: Improvements of affine prediction mode are described. In one example, a method for video processing is disclosed. The method includes: deriving for a current block, using motion information related to a first reference picture from a single reference picture list, motion information related to a second reference picture from another reference picture list different from the single reference list; and performing a video processing on the current block based on the motion information related to the first reference picture and the motion information related to the second reference picture..

WO 2020/098752 A1

[Continued on next page]

MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

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

Improvements of Affine Prediction Mode

[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 benefits of International Patent Application No. PCT/CN2018/115354, filed on November 14, 2018. The entire disclosures thereof are incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[0002] This patent document relates to image and video coding and decoding.

BACKGROUND

[0003] Digital video accounts for the largest bandwidth use on the internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, it is expected that the bandwidth demand for digital video usage will continue to grow.

SUMMARY

[0004] The disclosed techniques may be used by video decoder or encoder embodiments during video decoding or encoding using affine motion prediction or compensation tools.

[0005] In one example aspect, a method for video processing is disclosed. The method comprises: deriving for a current block, using motion information related to a first reference picture from a single reference picture list, motion information related to a second reference picture from another reference picture list different from the single reference list; and performing a video processing on the current block based on the motion information related to the first reference picture and the motion information related to the second reference picture.

[0006] In another example aspect, a method for video processing is disclosed. The method comprises: determining an affine mode type associated with a current block; and performing a

video processing on the current block based on the determined affine mode type; wherein the affine mode type is restricted for use to a portion of a video picture in which the current block is located and the determined affine mode type is prevented from use to other portions of the video picture.

[0007] In another example aspect, a method for video processing is disclosed. The method comprises: generating an affine candidate list for a current block, wherein the affine candidate list comprises an additional affine candidate which is generated by averaging a plurality of affine candidates from the affine candidate list; and performing a video processing on the current block based on the expanded affine candidate list.

[0008] In another example aspect, a video processing apparatus is disclosed. The video processing apparatus comprises a processor configured to implement the methods as described herein.

[0009] In another example aspect, a computer program product stored on a non-transitory computer readable media is disclosed. The computer program product includes program code for carrying out the methods as described herein.

[0010] In yet another example aspect, a video encoder apparatus is disclosed. The video encoder apparatus includes a processor that is configured to implement a method described herein.

[0011] In yet another example aspect, a video decoder apparatus is disclosed. The video decoder apparatus includes a processor that is configured to implement a method described herein.

[0012] In yet another aspect, a computer readable medium having code stored thereupon is disclosed. The code, when executed by a processor, causes the processor to implement a method described in the present document.

[0013] These, and other, aspects are described in the present document.

BRIEF DESCRIPTION OF DRAWINGS

[0014] FIG. 1 is an example of derivation process for merge candidates list construction.

[0015] FIG. 2 shows example positions of spatial merge candidates.

[0016] FIG. 3 shows an example of candidate pairs considered for redundancy check of spatial merge candidates.

[0017] FIG. 4A-4B show example positions for the second PU of N×2N and 2N×N partitions.

[0018] FIG. 5 is an example illustration of motion vector scaling for temporal merge candidate.

[0019] FIG. 6 shows example candidate positions for temporal merge candidate, C0 and C1.

[0020] FIG. 7 shows an example of combined bi-predictive merge candidate.

[0021] FIG. 8 shows an example derivation process for motion vector prediction candidates.

[0022] FIG. 9 is an example illustration of motion vector scaling for spatial motion vector candidate.

[0023] FIG. 10 shows an example of alternative temporal motion vector prediction (ATMVP) motion prediction for a CU.

[0024] FIG. 11 shows an example of one CU with four sub-blocks (A-D) and its neighbouring blocks (a–d).

[0025] FIG. 12 is a flowchart of an example of encoding with different MV precision

[0026] FIG. 13A - 13B show 135 degree partition type (splitting from top-left corner to bottom-right corner), and 45 degree splitting patterns. An illustration of splitting a CU into two triangular prediction units (two splitting patterns).

[0027] FIG. 14 shows an example of position of the neighboring blocks.

[0028] FIG. 15 shows examples of Above and Left blocks.

[0029] FIG. 16A-16B show examples of 2 control point motion vectors (CPMVs) and 3 CPMVs.

[0030] FIG. 17 shows an example of two CPMVs.

[0031] FIG. 18A-18B show examples of 4 and 6 parameter affine models.

[0032] FIG. 19 MVP for AF_INTER for inherited affine candidates.

[0033] FIG. 20 shows an example of constructing affine motion predictors in AF_INTER.

[0034] FIG. 21A-21B show examples of control point motion vectors in affine coding in AF_MERGE.

[0035] FIG. 22 shows examples of candidate positions for affine merge mode.

[0036] FIG. 23 shows an example of intra-picture block copy operation.

[0037] FIG. 24 shows candidates position for affine merge mode

[0038] FIG. 25 shows modified merge list construction process.

[0039] FIG. 26 is a block diagram of an example of a video processing apparatus.

[0040] FIG.27 is a flowchart for an example of a video processing method.

[0041] FIG.28 is a flowchart for another example of a video processing method.

[0042] FIG.29 is a flowchart for yet another example of a video processing method.

DETAILED DESCRIPTION

[0043] The present document provides various techniques that can be used by a decoder of video bitstreams to improve the quality of decompressed or decoded digital video. Furthermore, a video encoder may also implement these techniques during the process of encoding in order to reconstruct decoded frames used for further encoding.

[0044] 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.

[0045] 1. Summary

[0046] The present document is related to video coding technologies. Specifically, it is related to affine prediction mode in video coding. It may be applied to the existing video coding standard like HEVC, or the standard (Versatile Video Coding) to be finalized. It may be also applicable to future video coding standards or video codec.

[0047] In the present document, the term "video processing" may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa.

[0048] 2. Introductory comments

[0049] 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.

[0050] 2.1 Inter prediction in HEVC/H.265

[0051] 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.

[0052] 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.

[0053] 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.

[0054] 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.

[0055] The following text provides the details on the inter prediction modes specified in HEVC. The description will start with the merge mode.

[0056] 2.1.1 Reference picture list

[0057] 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.

[0058] 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.

[0059] 2.1.2 Merge Mode

[0060] 2.1.2.1 Derivation of candidates for merge mode

[0061] 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:

- Step 1: Initial candidates derivation
    - Step 1.1: Spatial candidates derivation
    - Step 1.2: Redundancy check for spatial candidates
    - Step 1.3: Temporal candidates derivation
- Step 2: Additional candidates insertion
    - Step 2.1: Creation of bi-predictive candidates
    - Step 2.2: Insertion of zero motion candidates

[0062] 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×2N prediction unit.

[0063] In the following, the operations associated with the aforementioned steps are detailed.

[0064] 2.1.2.2 Spatial candidates derivation

[0065] 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 A_1, B_1, B_0, A_0 and B_2. Position B_2 is considered only when any PU of position A_1, B_1, B_0, A_0 is not available (e.g. because it belongs to another slice or tile) or is intra coded. After candidate at position A_1 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 2Nx2N. As an example, Fig. 4A-4B depict the second PU for the case of N×2N and 2N×N, respectively. When the current PU is partitioned as N×2N, candidate at position A_1 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 B_1 is not considered when the current PU is partitioned as 2N×N.

[0066] 2.1.2.3 Temporal candidates derivation

[0067] 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, tb and td, where tb is defined to be the POC difference between the reference picture of the current picture and the current picture and td 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.

[0068] In the co-located PU (Y) belonging to the reference frame, the position for the temporal candidate is selected between candidates C_0 and C_1, as depicted in Fig. 6. If PU at position C_0 is not available, is intra coded, or is outside of the current coding tree unit (CTU aka. LCU, largest coding unit) row, position C_1 is used. Otherwise, position C_0 is used in the derivation of the temporal merge candidate.

[0069] 2.1.2.4 Additional candidates insertion

[0070] 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.

[0071] 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. The number of reference frames used by these candidates is one and two for uni and bi-directional prediction, respectively. Finally, no redundancy check is performed on these candidates.

[0072] 2.1.3 AMVP

[0073] 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.

[0074] 2.1.3.1 Derivation of AMVP candidates

[0075] Fig. 8 summarizes derivation process for motion vector prediction candidate.

[0076] 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.

[0077] 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.

[0078] 2.1.3.2 Spatial motion vector candidates

[0079] 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 A_0, A_1,and scaled A_0,scaled A_1. The order of derivation for the above side of the current PU is defined as B_0, B_1, B_2, scaled B_0, scaled B_1, scaled B_2. 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.

- No spatial scaling
  - (1) Same reference picture list, and same reference picture index (same POC)
  - (2) Different reference picture list, but same reference picture (same POC)
- Spatial scaling
  - (3) Same reference picture list, but different reference picture (different POC)
  - (4) Different reference picture list, and different reference picture (different POC)

[0080] 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.

[0081] 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.

[0082] 2.1.3.3 Temporal motion vector candidates

[0083] 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.

[0084] 2.2 Sub-CU based motion vector prediction methods in JEM

[0085] In the JEM with QTBT, each CU can have at most one set of motion parameters for each prediction direction. Two sub-CU level motion vector prediction methods are considered in the encoder by splitting a large CU into sub-CUs and deriving motion information for all the sub-CUs of the large CU. Alternative temporal motion vector prediction (ATMVP) method allows each CU to fetch multiple sets of motion information from multiple blocks smaller than the

current CU in the collocated reference picture. In spatial-temporal motion vector prediction (STMVP) method motion vectors of the sub-CUs are derived recursively by using the temporal motion vector predictor and spatial neighbouring motion vector.

[0086] To preserve more accurate motion field for sub-CU motion prediction, the motion compression for the reference frames is currently disabled.

[0087] 2.2.1 Alternative temporal motion vector prediction

[0088] In the alternative temporal motion vector prediction (ATMVP) method, the motion vectors temporal motion vector prediction (TMVP) is modified by fetching multiple sets of motion information (including motion vectors and reference indices) from blocks smaller than the current CU. As an example, the sub-CUs are square N×N blocks (N is set to 4 by default). Fig. 10 shows an example of ATMVP motion prediction for a CU.

[0089] ATMVP predicts the motion vectors of the sub-CUs within a CU in two steps. The first step is to identify the corresponding block in a reference picture with a so-called temporal vector. The reference picture is called the motion source picture. The second step is to split the current CU into sub-CUs and obtain the motion vectors as well as the reference indices of each sub-CU from the block corresponding to each sub-CU, as an example.

[0090] In the first step, a reference picture and the corresponding block is determined by the motion information of the spatial neighbouring blocks of the current CU. To avoid the repetitive scanning process of neighbouring blocks, the first merge candidate in the merge candidate list of the current CU is used. The first available motion vector as well as its associated reference index are set to be the ***temporal vector*** and the index to the motion source picture. This way, in ATMVP, the corresponding block may be more accurately identified, compared with TMVP, wherein the corresponding block (sometimes called collocated block) is always in a bottom-right or center position relative to the current CU.

[0091] In the second step, a corresponding block of the sub-CU is identified by the temporal vector in the motion source picture, by adding to the coordinate of the current CU the temporal vector. For each sub-CU, the motion information of its corresponding block (the smallest motion grid that covers the center sample) is used to derive the motion information for the sub-CU. After the motion information of a corresponding N×N block is identified, it is converted to the motion vectors and reference indices of the current sub-CU, in the same way as TMVP of HEVC,

wherein motion scaling and other procedures apply. For example, the decoder checks whether the low-delay condition (i.e. the POCs of all reference pictures of the current picture are smaller than the POC of the current picture) is fulfilled and possibly uses motion vector MV_x (the motion vector corresponding to reference picture list X) to predict motion vector MV_y (with X being equal to 0 or 1 and Y being equal to 1−X) for each sub-CU.

[0092] 2.2.2 Spatial-temporal motion vector prediction (STMVP)

[0093] In this method, the motion vectors of the sub-CUs are derived recursively, following raster scan order. Fig. 11 illustrates this concept. Let us consider an 8×8 CU which contains four 4×4 sub-CUs A, B, C, and D. The neighbouring 4×4 blocks in the current frame are labelled as a, b, c, and d.

[0094] The motion derivation for sub-CU A starts by identifying its two spatial neighbours. The first neighbour is the N×N block above sub-CU A (block c). If this block c is not available or is intra coded the other N×N blocks above sub-CU A are checked (from left to right, starting at block c). The second neighbour is a block to the left of the sub-CU A (block b). If block b is not available or is intra coded other blocks to the left of sub-CU A are checked (from top to bottom, staring at block b). The motion information obtained from the neighbouring blocks for each list is scaled to the first reference frame for a given list. Next, temporal motion vector predictor (TMVP) of sub-block A is derived by following the same procedure of TMVP derivation as specified in HEVC. The motion information of the collocated block at location D is fetched and scaled accordingly. Finally, after retrieving and scaling the motion information, all available motion vectors (up to 3) are averaged separately for each reference list. The averaged motion vector is assigned as the motion vector of the current sub-CU.

[0095] 2.2.3 Sub-CU motion prediction mode signalling

[0096] The sub-CU modes are enabled as additional merge candidates and there is no additional syntax element required to signal the modes. Two additional merge candidates are added to merge candidates list of each CU to represent the ATMVP mode and STMVP mode. Up to seven merge candidates are used, if the sequence parameter set indicates that ATMVP and STMVP are enabled. The encoding logic of the additional merge candidates is the same as for the merge candidates in the HM, which means, for each CU in P or B slice, two more RD checks is needed for the two additional merge candidates.

[0097] In the JEM, all bins of merge index is context coded by CABAC. While in HEVC, only the first bin is context coded and the remaining bins are context by-pass coded.

[0098] 2.3 Inter prediction methods in VVC

[0099] 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).

[00100] 2.3.1 Adaptive motion vector difference resolution

[00101] In HEVC, motion vector differences (MVDs) (between the motion vector and predicted motion vector of a PU) are signalled in units of quarter luma samples when use_integer_mv_flag is equal to 0 in the slice header. In the VVC, a locally adaptive motion vector resolution (LAMVR) is introduced. In the VVC, MVD can be coded in units of quarter luma samples, integer luma samples or four luma samples (i.e., ¼-pel, 1-pel, 4-pel). The MVD resolution is controlled at the coding unit (CU) level, and MVD resolution flags are conditionally signalled for each CU that has at least one non-zero MVD components.

[00102] For a CU that has at least one non-zero MVD components, a first flag is signalled to indicate whether quarter luma sample MV precision is used in the CU. When the first flag (equal to 1) indicates that quarter luma sample MV precision is not used, another flag is signalled to indicate whether integer luma sample MV precision or four luma sample MV precision is used.

[00103] When the first MVD resolution flag of a CU is zero, or not coded for a CU (meaning all MVDs in the CU are zero), the quarter luma sample MV resolution is used for the CU. When a CU uses integer-luma sample MV precision or four-luma-sample MV precision, the MVPs in the AMVP candidate list for the CU are rounded to the corresponding precision.

[00104] In the encoder, CU-level RD checks are used to determine which MVD resolution is to be used for a CU. That is, the CU-level RD check is performed three times for each MVD resolution. To accelerate encoder speed, the following encoding schemes are applied in the JEM.

- During RD check of a CU with normal quarter luma sample MVD resolution, the motion information of the current CU (integer luma sample accuracy) is stored. The stored motion information (after rounding) is used as the starting point for further small range motion vector refinement during the RD check for the same CU with integer luma sample

and 4 luma sample MVD resolution so that the time-consuming motion estimation process is not duplicated three times.

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

[00105] The encoding process is shown in Fig. 12. First, 1/4 pel MV is tested and the RD cost is calculated and denoted as *RDCost0*, then integer MV is tested and the RD cost is denoted as *RDCost1*. If $RDCost1 < th * RDCost0$ (wherein *th* is a positive value), then 4-pel MV is tested; otherwise, 4-pel MV is skipped. Basically, motion information and RD cost etc. are already known for 1/4 pel MV when checking integer or 4-pel MV, which can be reused to speed up the encoding process of integer or 4-pel MV.

[00106] 2.3.2 Triangular prediction mode

[00107] The concept of the triangular prediction mode (TPM) is to introduce a new triangular partition for motion compensated prediction. As shown in Fig. 13A-13B, it splits a CU into two triangular prediction units, in either diagonal or inverse diagonal direction. Each triangular prediction unit in the CU is inter-predicted using its own uni-prediction motion vector and reference frame index which are derived from a single uni-prediction candidate list. An adaptive weighting process is performed to the diagonal edge after predicting the triangular prediction units. Then, the transform and quantization process are applied to the whole CU. It is noted that this mode is only applied to merge mode (note: skip mode is treated as a special merge mode).

[00108] Fig. 13A-13B is an illustration of splitting a CU into two triangular prediction units (two splitting patterns); Fig. 13A：135 degree parttion type (splitting from top-left corner to bottom-right corner), and Fig. 13B: 45 degree splitting patterns

[00109] 2.3.2.1 Uni-prediction candidate list for TPM

[00110] The uni-prediction candidate list, named TPM motion candidate list, consists of five uni-prediction motion vector candidates. It is derived from seven neighboring blocks including five spatial neighboring blocks (1 to 5) and two temporal co-located blocks (6 to 7), as shown in Fig. 14. The motion vectors of the seven neighboring blocks are collected and put into the uni-

prediction candidate list according in the order of uni-prediction motion vectors, L0 motion vector of bi-prediction motion vectors, L1 motion vector of bi-prediction motion vectors, and averaged motion vector of the L0 and L1 motion vectors of bi-prediction motion vectors. If the number of candidates is less than five, zero motion vector is added to the list. Motion candidates added in this list for TPM are called TPM candidates, motion information derived from spatial/temporal blocks are called regular motion candidates.

[00111] More specifically, the following steps are involved:

1) Obtain regular motion candidates from A_1, B_1, B_0, A_0, B_2, Col and Col2 (corresponding to block 1-7 in Fig. 14) without any pruning operations.
2) Set variable numCurrMergeCand = 0
3) For each regular motion candidates derived from A_1, B_1, B_0, A_0, B_2, Col and Col2 and numCurrMergeCand is less than 5, if the regular motion candidate is **uni-prediction** (either from List 0 or List 1), it is directly added to the merge list as an TPM candidate with numCurrMergeCand increased by 1. Such a TPM candidate is named '**originally uni-predicted candidate**'.
   Full pruning is applied.
4) For each motion candidates derived from A_1, B_1, B_0, A_0, B_2, Col and Col2 and numCurrMergeCand is less than 5, if the regular motion candidate is **bi-prediction**, the motion information from **List 0** is added to the TPM merge list (that is, modified to be uni-prediction from List 0) as a new TPM candidate and numCurrMergeCand increased by 1. Such a TPM candidate is named '**Truncated List0-predicted candidate**'.
   Full pruning is applied.
5) For each motion candidates derived from A_1, B_1, B_0, A_0, B_2, Col and Col2 and numCurrMergeCand is less than 5, if the regular motion candidate is **bi-prediction**, the motion information from **List 1** is added to the TPM merge list (that is, modified to be uni-prediction from List 1) and numCurrMergeCand increased by 1. Such a TPM candidate is named '**Truncated List1-predicted candidate**'.
   Full pruning is applied.
6) For each motion candidates derived from A_1, B_1, B_0, A_0, B_2, Col and Col2 and numCurrMergeCand is less than 5, if the regular motion candidate is bi-prediction,

- **If List 0 reference picture's slice QP is smaller than List 1 reference picture's slice QP,** the motion information of List 1 is firstly scaled to List 0 reference picture, and the average of the two MVs (one is from original List 0, and the other is the scaled MV from List 1) is added to the TPM merge list, such a candidate is called **averaged uni-prediction from List 0 motion candidate** and numCurrMergeCand increased by 1.
- Otherwise, the motion information of List 0 is firstly scaled to List 1 reference picture, and the average of the two MVs (one is from original List 1, and the other is the scaled MV from List 0) is added to the TPM merge list , such a TPM candidate is called **averaged uni-prediction from List 1 motion candidate** and numCurrMergeCand increased by 1.

Full pruning is applied.

7) If numCurrMergeCand is less than 5, zero motion vector candidates are added.

[00112] When inserting a candidate to the list, if it has to be compared to all previously added candidates to see whether it is identical to one of them, such a process is called full pruning.

[00113] 2.3.2.2 Adaptive weighting process

[00114] After predicting each triangular prediction unit, an adaptive weighting process is applied to the diagonal edge between the two triangular prediction units to derive the final prediction for the whole CU. Two weighting factor groups are defined as follows:

- 1st weighting factor group: {7/8, 6/8, 4/8, 2/8, 1/8} and {7/8, 4/8, 1/8} are used for the luminance and the chrominance samples, respectively;
- 2nd weighting factor group: {7/8, 6/8, 5/8, 4/8, 3/8, 2/8, 1/8} and {6/8, 4/8, 2/8} are used for the luminance and the chrominance samples, respectively.

[00115] Weighting factor group is selected based on the comparison of the motion vectors of two triangular prediction units. The 2nd weighting factor group is used when the reference pictures of the two triangular prediction units are different from each other or their motion vector difference is larger than 16 pixels. Otherwise, the 1st weighting factor group is used. An example is shown in Fig. 15.

[00116] 2.3.2.3 Signaling of triangular prediction mode (TPM)

[00117] One bit flag to indicate whether TPM is used may be firstly signaled. Afterwards, the indications of two splitting patterns (as depicted in Fig. 13), and selected merge indices for each of the two partitions are further signaled.

[00118] 2.3.2.3.1 Signaling of TPM flag

[00119] Let's denote one luma block's width and height by W and H, respectively. If W*H < 64, triangular prediction mode is disabled.

[00120] When one block is coded with affine mode, triangular prediction mode is also disabled.

[00121] When one block is coded with merge mode, one bit flag may be signaled to indicate whether the triangular prediction mode is enabled or disabled for the block.

[00122] The flag is coded with 3 contexts, based on the following equation:

[00123] Ctx index = ((left block **L** available && **L** is coded with TPM?) 1: 0)

[00124] + ((Above block **A** available && A is coded with TPM?) 1: 0);

[00125] 2.3.2.3.2 Signaling of an indication of two splitting patterns (as depicted in Fig. 13), and selected merge indices for each of the two partitions

[00126] It is noted that splitting patterns, merge indices of two partitions are jointly coded. As an example, it is restricted that the two partitions couldn't use the same reference index. Therefore, there are 2 (splitting patterns) * N (maximum number of merge candidates) *(N-1) possibilities wherein N is set to 5. One indication is coded and the mapping between the splitting patterns, two merge indices and coded indication are derived from the array defined below:

```
const uint8_t g_TriangleCombination[TRIANGLE_MAX_NUM_CANDS][3] = {
 { 0, 1, 0 }, { 1, 0, 1 }, { 1, 0, 2 }, { 0, 0, 1 }, { 0, 2, 0 },
 { 1, 0, 3 }, { 1, 0, 4 }, { 1, 1, 0 }, { 0, 3, 0 }, { 0, 4, 0 },
 { 0, 0, 2 }, { 0, 1, 2 }, { 1, 1, 2 }, { 0, 0, 4 }, { 0, 0, 3 },
 { 0, 1, 3 }, { 0, 1, 4 }, { 1, 1, 4 }, { 1, 1, 3 }, { 1, 2, 1 },
 { 1, 2, 0 }, { 0, 2, 1 }, { 0, 4, 3 }, { 1, 3, 0 }, { 1, 3, 2 },
 { 1, 3, 4 }, { 1, 4, 0 }, { 1, 3, 1 }, { 1, 2, 3 }, { 1, 4, 1 },
 { 0, 4, 1 }, { 0, 2, 3 }, { 1, 4, 2 }, { 0, 3, 2 }, { 1, 4, 3 },
 { 0, 3, 1 }, { 0, 2, 4 }, { 1, 2, 4 }, { 0, 4, 2 }, { 0, 3, 4 }};
```

[00127] splitting patterns (45 degree or 135 degree) = g_TriangleCombination[signaled indication][0];

[00128] Merge index of candidate A = g_TriangleCombination[signaled indication][1];

[00129] Merge index of candidate B = g_TriangleCombination[signaled indication][2];

[00130] Once the two motion candidates A and B are derived, the two partitions' (PU1 and PU2) motion information could be set either from A or B. Whether PU1 uses the motion information of merge candidate A or B is dependent on the prediction directions of the two motion candidates. Table 1 shows the relationship between two derived motion candidates A and B, with the two partitions.

Table 1: Derivation of partitions' motion information from derived two merge candidates (A, B)

Prediction direction of A	Prediction direction of B	PU1's motion information	PU2's motion information
L0	L0	A (L0)	B (L0)
L1	L1	B (L1)	A (L1)
L0	L1	A (L0)	B (L1)
L1	L0	B (L0)	A (L1)

[00131] 2.3.2.3.3 Entropy coding of the indication (denoted by merge_triangle_idx)

[00132] merge_triangle_idx is within the range [0, 39], inclusively. K-th order Exponential Golomb (EG) code is used for binarization of merge_triangle_idx wherein K is set to 1.

K-th order EG

[00133] To encode larger numbers in fewer bits (at the expense of using more bits to encode smaller numbers), this can be generalized using a nonnegative integer parameter k. To encode a nonnegative integer x in an order-k exp-Golomb code:

1. Encode $\lfloor x/2^k \rfloor$ using order-0 exp-Golomb code described above, then

2. Encode x mod 2^k in binary

Table 2: Exp-Golomb-k coding examples

x	*k*=0	*k*=1	*k*=2	*x*	*k*=0	*k*=1	*k*=2
0	1	10	100	10	0001011	001100	01110
1	010	11	101	11	0001100	001101	01111
2	011	0100	110	12	0001101	001110	0010000
3	00100	0101	111	13	0001110	001111	0010001
4	00101	0110	01000	14	0001111	00010000	0010010
5	00110	0111	01001	15	000010000	00010001	0010011
6	00111	001000	01010	16	000010001	00010010	0010100
7	0001000	001001	01011	17	000010010	00010011	0010101
8	0001001	001010	01100	18	000010011	00010100	0010110
9	0001010	001011	01101	19	000010100	00010101	0010111

[00134] 2.3.3 Affine motion compensation prediction

[00135] In HEVC, only translation motion model is applied for motion compensation prediction (MCP). While in the real world, there are many kinds of motion, e.g. zoom in/out, rotation, perspective motions and the other irregular motions. In VVC, a simplified affine transform motion compensation prediction is applied with 4-parameter affine model and 6-parameter affine model. As shown Fig. 16A-16B, the affine motion field of the block is described by two control point motion vectors (CPMVs) for the 4-parameter affine model (Fig. 16A) and 3 CPMVs for the 6-parameter affine model (Fig. 16B).

[00136] The motion vector field (MVF) of a block is described by the following equations with the 4-parameter affine model (wherein the 4-parameter are defined as the variables *a, b, e* and *f*) in equation (1) and 6-parameter affine model (wherein the 4-parameter are defined as the variables *a, b, c, d, e* and *f*) in equation (2) respectively:

$$\begin{cases} mv^h(x,y) = ax - by + e = \dfrac{(mv_1^h - mv_0^h)}{w}x - \dfrac{(mv_1^v - mv_0^v)}{w}y + mv_0^h \\ mv^v(x,y) = bx + ay + f = \dfrac{(mv_1^v - mv_0^v)}{w}x + \dfrac{(mv_1^h - mv_0^h)}{w}y + mv_0^v \end{cases} \quad (1)$$

$$\begin{cases} mv^h(x,y) = ax + cy + e = \dfrac{(mv_1^h - mv_0^h)}{w}x + \dfrac{(mv_2^h - mv_0^h)}{h}y + mv_0^h \\ mv^v(x,y) = bx + dy + f = \dfrac{(mv_1^v - mv_0^v)}{w}x + \dfrac{(mv_2^v - mv_0^v)}{h}y + mv_0^v \end{cases} \quad (2)$$

[00137] where (mv^h_0, mv^h_0) is motion vector of the top-left corner control point, and (mv^h_1, mv^h_1) is motion vector of the top-right corner control point and (mv^h_2, mv^h_2) is motion vector of the bottom-left corner control point, all of the three motion vectors are called **control point motion vectors (CPMV)**, (x, y) represents the coordinate of a **representative point** relative to the top-left sample within current block and ($mv^h(x,y)$, $mv^v(x,y)$) is the motion vector derived for a sample located at (x, y). The CP motion vectors may be signaled (like in the affine AMVP mode) or derived on-the-fly (like in the affine merge mode). *w* and *h* are the width and height of the current block. In practice, the division is implemented by right-shift with a rounding operation. In VTM, the **representative point** is defined to be the center position of a sub-block, e.g., when the coordinate of the left-top corner of a sub-block relative to the top-left sample within current block is (xs, ys), the coordinate of the representative point is defined to be (xs+2, ys+2). For each sub-block (i.e., 4x4 in VTM), the representative point is utilized to derive the motion vector for the whole sub-block.

[00138] In order to further simplify the motion compensation prediction, sub-block based affine transform prediction is applied. To derive motion vector of each M×N (both M and N are set to 4 in current VVC) sub-block, the motion vector of the center sample of each sub-block, as shown in Fig. 17, is calculated according to Equation (1) and (2), and rounded to 1/16 fraction accuracy. Then the motion compensation interpolation filters for 1/16-pel are applied to generate the prediction of each sub-block with derived motion vector. The interpolation filters for 1/16-pel are introduced by the affine mode.

[00139] After MCP, the high accuracy motion vector of each sub-block is rounded and saved as the same accuracy as the normal motion vector.

[00140] 2.3.3.1 Signaling of affine prediction

[00141] Similar to the translational motion model, there are also two modes for signaling the side information due affine prediction. They are AFFINE_INTER and AFFINE_MERGE modes.

[00142] 2.3.3.2 AF_INTER mode

[00143] For CUs with both width and height larger than 8, AF_INTER mode can be applied. An affine flag in CU level is signalled in the bitstream to indicate whether AF_INTER mode is used.

[00144] In this mode, for each reference picture list (List 0 or List 1), an affine AMVP candidate list is constructed with three types of affine motion predictors in the following order, wherein each candidate includes the estimated CPMVs of the current block. The differences of the best CPMVs found at the encoder side (such as mv_0 mv_1 mv_2 in Fig. 20) and the estimated CPMVs are signalled. In addition, the index of affine AMVP candidate from which the estimated CPMVs are derived is further signalled.

1) Inherited affine motion predictors

[00145] The checking order is similar to that of spatial MVPs in HEVC AMVP list construction. First, a left inherited affine motion predictor is derived from the first block in {A1, A0} that is affine coded and has the same reference picture as in current block. Second, an above inherited affine motion predictor is derived from the first block in {B1, B0, B2} that is affine coded and has the same reference picture as in current block. The five blocks A1, A0, B1, B0, B2 are depicted in Fig. 19.

[00146] Once a neighboring block is found to be coded with affine mode, the CPMVs of the coding unit covering the neighboring block are used to derive predictors of CPMVs of current block. For example, if A1 is coded with non-affine mode and A0 is coded with 4-parameter affine mode, the left inherited affine MV predictor will be derived from A0. In this case, the CPMVs of a CU covering A0, as denoted by MV_0^N for the top-left CPMV and MV_1^N for the top-right CPMV in Fig. 21B are utilized to derive the estimated CPMVs of current block, denoted by MV_0^C, MV_1^C, MV_2^C for the top-left (with coordinate (x0, y0)), top-right (with coordinate (x1, y1)) and bottom-right positions (with coordinate (x2, y2)) of current block.

2) Constructed affine motion predictors

[00147] A constructed affine motion predictor consists of control-point motion vectors (CPMVs) that are derived from neighboring inter coded blocks, as shown in Fig. 20, that have the same reference picture. If the current affine motion model is 4-paramter affine, the number of CPMVs is 2, otherwise if the current affine motion model is 6-parameter affine, the number of CPMVs is 3. The top-left CPMV $\overline{mv}_0$ is derived by the MV at the first block in the group {A, B, C} that is inter coded and has the same reference picture as in current block. The top-right CPMV $\overline{mv}_1$ is derived by the MV at the first block in the group {D, E} that is inter coded and has the same reference picture as in current block. The bottom-left CPMV $\overline{mv}_2$ is derived by the MV at the first block in the group {F, G} that is inter coded and has the same reference picture as in current block.

- If the current affine motion model is 4-parameter affine, then a constructed affine motion predictor is inserted into the candidate list only if both $\overline{mv}_0$ and $\overline{mv}_1$ are founded, that is, $\overline{mv}_0$ and $\overline{mv}_1$ are used as the estimated CPMVs for top-left (with coordinate (x0, y0)), top-right (with coordinate (x1, y1)) positions of current block.
- If the current affine motion model is 6-parameter affine, then a constructed affine motion predictor is inserted into the candidate list only if $\overline{mv}_0$, $\overline{mv}_1$ and $\overline{mv}_2$ are all founded, that is, $\overline{mv}_0$, $\overline{mv}_1$ and $\overline{mv}_2$ are used as the estimated CPMVs for top-left (with coordinate (x0, y0)), top-right (with coordinate (x1, y1)) and bottom-right (with coordinate (x2, y2)) positions of current block.

[00148] No pruning process is applied when inserting a constructed affine motion predictor into the candidate list.

3) Normal AMVP motion predictors

[00149] The following applies until the number of affine motion predictors reaches the maximum.

1) Derive an affine motion predictor by setting all CPMVs equal to $\overline{mv}_2$ if available.
2) Derive an affine motion predictor by setting all CPMVs equal to $\overline{mv}_1$ if available.
3) Derive an affine motion predictor by setting all CPMVs equal to $\overline{mv}_0$ if available.

4) Derive an affine motion predictor by setting all CPMVs equal to HEVC TMVP if available.

5) Derive an affine motion predictor by setting all CPMVs to zero MV.

[00150] Note that $\overline{\boldsymbol{mv}}_i$ is already derived in constructed affine motion predictor.

[00151] Fig. 18A-18B show a 4-paramenter affine model and 6-parameter affine model, respectively.

[00152] Fig. 19 shows an example of an MVP for AF_INTER for inherited affine candidates

[00153] Fig. 20 shows an example of an MVP for AF_INTER for constructed affine candidates.

[00154] In AF_INTER mode, when 4/6-parameter affine mode is used, 2/3 control points are required, and therefore 2/3 MVD needs to be coded for these control points, as shown in Fig. 18A-18B. In an example, it is proposed to derive the MV as follows, i.e., mvd_1 and mvd_2 are predicted from mvd_0.

$$\boldsymbol{mv}_0 = \overline{\boldsymbol{mv}}_0 + \boldsymbol{mvd}_0$$

$$\boldsymbol{mv}_1 = \overline{\boldsymbol{mv}}_1 + \boldsymbol{mvd}_1 + \boldsymbol{mvd}_0$$

$$\boldsymbol{mv}_2 = \overline{\boldsymbol{mv}}_2 + \boldsymbol{mvd}_2 + \boldsymbol{mvd}_0$$

[00155] Wherein $\overline{\boldsymbol{mv}}_i$, mvd_i and mv_1 are the predicted motion vector, motion vector difference and motion vector of the top-left pixel (i = 0), top-right pixel (i = 1) or left-bottom pixel (i = 2) respectively, as shown in Fig. 18B. Please note that the addition of two motion vectors (e.g., mvA(xA, yA) and mvB(xB, yB)) is equal to summation of two components separately, that is, newMV = mvA + mvB and the two components of newMV is set to (xA + xB) and (yA + yB), respectively.

[00156] 2.3.3.3 AF_MERGE mode

[00157] When a CU is applied in AF_MERGE mode, it gets the first block coded with affine mode from the valid neighbour reconstructed blocks. And the selection order for the candidate block is from left, above, above right, left bottom to above left as shown in Fig. 21A (denoted by A, B, C, D, E in order). For example, if the neighbour left bottom block is coded in affine mode as denoted by A0 in Fig. 21B, the Control Point (CP) motion vectors mv_0^N, mv_1^N and mv_2^N of the top left corner, above right corner and left bottom corner of the neighbouring CU/PU which contains the block A are fetched. And the motion vector mv_0^C, mv_1^C and mv_2^C (which is only used for the 6-parameter affine model) of the top left corner/top right/bottom left on the current

CU/PU is calculated based on mv_0^N, mv_1^N and mv_2^N. It should be noted that in VTM-2.0, sub-block (e.g. 4×4 block in VTM) located at the top-left corner stores mv0, the sub-block located at the top-right corner stores mv1 if the current block is affine coded. If the current block is coded with the 6-parameter affine model, the sub-block located at the bottom-left corner stores mv2; otherwise (with the 4-parameter affine model), LB stores mv2'. Other sub-blocks stores the MVs used for MC.

[00158] After the CPMV of the current CU mv_0^C, mv_1^C and mv_2^C are derived, according to the simplified affine motion model Equation (1) and (2), the MVF of the current CU is generated. In order to identify whether the current CU is coded with AF_MERGE mode, an affine flag is signalled in the bitstream when there is at least one neighbour block is coded in affine mode.

[00159] Fig. 21A-21B show candidates for AF_MERGE with five neighboring blocks and CPMV predictor derivation, respectively.

[00160] In examples, an affine merge candidate list is constructed with following steps:

1) Insert inherited affine candidates

[00161] Inherited affine candidate means that the candidate is derived from the affine motion model of its valid neighbor affine coded block. The maximum two inherited affine candidates are derived from affine motion model of the neighboring blocks and inserted into the candidate list. For the left predictor, the scan order is {A0, A1}; for the above predictor, the scan order is {B0, B1, B2}.

2) Insert constructed affine candidates

[00162] If the number of candidates in affine merge candidate list is less than MaxNumAffineCand (set to 5), constructed affine candidates are inserted into the candidate list. Constructed affine candidate means the candidate is constructed by combining the neighbor motion information of each control point.

[00163] The motion information for the control points is derived firstly from the specified spatial neighbors and temporal neighbor shown in Fig. 22. CPk (k=1, 2, 3, 4) represents the k-th control point. A0, A1, A2, B0, B1, B2 and B3 are spatial positions for predicting CPk (k=1, 2, 3); T is temporal position for predicting CP4.

[00164] The coordinates of CP1, CP2, CP3 and CP4 is (0, 0), (W, 0), (H, 0) and (W, H), respectively, where W and H are the width and height of current block.

[00165] The motion information of each control point is obtained according to the following priority order:

[00166] For CP1, the checking priority is B2->B3->A2. B2 is used if it is available. Otherwise, if B2 is available, B3 is used. If both B2 and B3 are unavailable, A2 is used. If all the three candidates are unavailable, the motion information of CP1 cannot be obtained.

[00167] For CP2, the checking priority is B1->B0.

[00168] For CP3, the checking priority is A1->A0.

[00169] For CP4, T is used.

[00170] Secondly, the combinations of controls points are used to construct an affine merge candidate.

[00171] Motion information of three control points are needed to construct a 6-parameter affine candidate. The three control points can be selected from one of the following four combinations ({CP1, CP2, CP4}, {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4}). Combinations {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4} will be converted to a 6-parameter motion model represented by top-left, top-right and bottom-left control points.

[00172] Motion information of two control points are needed to construct a 4-parameter affine candidate. The two control points can be selected from one of the two combinations ({CP1, CP2}, {CP1, CP3}). The two combinations will be converted to a 4-parameter motion model represented by top-left and top-right control points.

[00173] The combinations of constructed affine candidates are inserted into to candidate list as following order:

[00174] {CP1, CP2, CP3}, {CP1, CP2, CP4}, {CP1, CP3, CP4}, {CP2, CP3, CP4}, {CP1, CP2}, {CP1, CP3}

[00175] The available combination of motion information of CPs is only added to the affine merge list when the CPs have the same reference index.

3) Padding with zero motion vectors

[00176] If the number of candidates in affine merge candidate list is less than 5, zero motion vectors with zero reference indices are insert into the candidate list, until the list is full.

[00177] 2.3.4 Current Picture Referencing

[00178] Intra block copy (IBC, or intra picture block compensation), also named current picture referencing (CPR) was adopted in HEVC screen content coding extensions (SCC). This tool is very efficient for coding of screen content video in that repeated patterns in text and graphics rich content occur frequently **within the same picture**. Having a previously reconstructed block with equal or similar pattern as a predictor can effectively reduce the prediction error and therefore improve coding efficiency. An example of the intra block compensation is illustrated in FIG. 23.

[00179] Similar to the design of CRP in HEVC SCC, In VVC, The use of the IBC mode is signaled at both sequence and picture level. When the IBC mode is enabled at sequence parameter set (SPS), it can be enabled at picture level. When the IBC mode is enabled at picture level, the current reconstructed picture is treated as a reference picture. Therefore, no syntax change on block level is needed on top of the existing VVC inter mode to signal the use of the IBC mode.

[00180] Main features:

- It is treated as a normal inter mode. Therefore, merge and skip modes are also available for the IBC mode. The merge candidate list construction is unified, containing merge candidates from the neighboring positions that are either coded in the IBC mode or the HEVC inter mode. Depending on the selected merge index, the current block under merge or skip mode can merge into either an IBC mode coded neighbor or otherwise an normal inter mode coded one with different pictures as reference pictures.
- Block vector prediction and coding schemes for the IBC mode reuse the schemes used for motion vector prediction and coding in the HEVC inter mode (AMVP and MVD coding).
- The motion vector for the IBC mode, also referred as block vector, is coded with **integer-pel precision**, but stored in memory in 1/16-pel precision after decoding as quarter-pel precision is required in interpolation and deblocking stages. When used in motion vector prediction for the IBC mode, the stored vector predictor will be right shifted by 4.
- Search range: it is restricted to be within the current CTU.
- CPR is disallowed when affine mode/triangular mode/GBI/weighted prediction is enabled.

[00181] 2.3.5 Merge list design in VVC

[00182] There are three different merge list construction processes supported in VVC:

1) **Sub-block merge candidate list**: it includes ATMVP and affine merge candidates. One merge list construction process is shared for both affine modes and ATMVP mode. Here, the ATMVP and affine merge candidates may be added in order. Sub-block merge list size is signaled in slice header, and maximum value is 5.
2) **Uni-Prediction TPM merge list**: For triangular prediction mode, one merge list construction process for the two partitions is shared even two partitions could select their own merge candidate index. When constructing this merge list, the spatial neighbouring blocks and two temporal blocks of the block are checked. The motion information derived from spatial neighbours and temporal blocks are called regular motion candidates in our IDF. These regular motion candidates are further utilized to derive multiple TPM candidates. Please note the transform is performed in the whole block level, even two partitions may use different motion vectors for generating their own prediction blocks. Uni-Prediction TPM merge list size is fixed to be 5.
3) **Regular merge list**: For remaining coding blocks, one merge list construction process is shared. Here, the spatial/temporal/HMVP, pairwise combined bi-prediction merge candidates and zero motion candidates may be inserted in order. Regular merge list size is signaled in slice header, and maximum value is 6.

[00183] 2.3.5.1 Sub-block merge candidate list

[00184] It is suggested that all the sub-block related motion candidates are put in a separate merge list in addition to the regular merge list for non-sub block merge candidates.

[00185] The sub-block related motion candidates are put in a separate merge list is named as 'sub-block merge candidate list'.

[00186] In one example, the sub-block merge candidate list includes affine merge candidates, and ATMVP candidate, and/or sub-block based STMVP candidate.

[00187] In an example, the ATMVP merge candidate in the normal merge list is moved to the first position of the affine merge list. Such that all the merge candidates in the new list (i.e., sub-block based merge candidate list) are based on sub-block coding tools.

[00188] 2.3.5.2 Regular merge list

[00189] Different from the merge list design, in VVC, the history-based motion vector prediction (HMVP) method is employed.

[00190] In HMVP, the previously coded motion information is stored. The motion information of a previously coded block is defined as an HMVP candidate. Multiple HMVP candidates are stored in a table, named as the HMVP table, and this table is maintained during the encoding/decoding process on-the-fly. The HMVP table is emptied when starting coding/decoding a new slice. Whenever there is an inter-coded block, the associated motion information is added to the last entry of the table as a new HMVP candidate. The overall coding flow is depicted in Fig. 24.

[00191] HMVP candidates could be used in both AMVP and merge candidate list construction processes. Fig. 25 depicts the modified merge candidate list construction process (highlighted in gray). When the merge candidate list is not full after the TMVP candidate insertion, HMVP candidates stored in the HMVP table could be utilized to fill in the merge candidate list. Considering that one block usually has a higher correlation with the nearest neighbouring block in terms of motion information, the HMVP candidates in the table are inserted in a descending order of indices. The last entry in the table is firstly added to the list, while the first entry is added in the end. Similarly, redundancy removal is applied on the HMVP candidates. Once the total number of available merge candidates reaches the maximal number of merge candidates allowed to be signaled, the merge candidate list construction process is terminated.

[00192] 3. Examples of technical solved by disclosed embodiments

[00193] In the current design of VVC, the affine prediction mode could achieve significant coding gains for sequences with affine motion. However, it may have the following problems:

1) For bi-prediction affine mode, the correlation of affine motion information among the two reference picture list is not taken into consideration.
2) For affine merge candidates derivation process, the affine model (4-parameter or 6-paramater) type is directly inherited from neighboring blocks which requires additional line buffer size to store the affine model type.

[00194] 4. Description of various techniques

[00195] The detailed inventions below should be considered as examples to explain general concepts. These inventions should not be interpreted in a narrow way. Furthermore, these inventions can be combined in any manner.

1. It is proposed that the candidates added to one reference picture list could be used to predict the CPMVs of the other reference picture.
    a. In one example, the CPMVs of one reference picture could be used to predict the CPMVs of the other reference picture.
    b. In one example, the coded MVDs of one reference picture could be (scaled if necessary) used to predict the MVDs of another reference picture.
2. A symmetric affine coding mode is proposed wherein the motion information of one reference picture list (list X) is signalled while the motion information of another reference picture list (list Y wherein Y is unequal to X) is always skipped.
    a. In one example, the motion information (such as CPMVs) of the reference picture list (list Y) without signalling could be derived from that of the reference picture list (list X).
    b. In one example, the prediction direction of this mode is also set to bi-prediction.
    c. In one example, it is added as a new coding mode. Alternatively, it may be used to replace the uni-affine coded mode.
3. It is proposed that the affine model types (e.g., 4-parameter or 6-parameter) may be utilized to decide the insertion order of affine candidates in constructing the affine candidate list (e.g., affine AMVP/merge candidate list, sub-block merge candidate list).
    a. For the affine AMVP candidate list, the neighboring blocks with the same affine model type may be given a higher priority. For example, the motion information of a neighboring block with the same affine model type may be added to the AMVP list before that of a second neighboring block with different affine model type.
    b. In one example, affine type may be further signaled for affine merge mode.
    c. For the affine merge candidate list and/or sub-block merge candidate list, the neighboring blocks with the same affine model type may be given a higher priority.

i. In one example, the motion information of a neighboring block with the same affine model type as the first affine candidate may be added to the merge list before that of a second neighboring block with different affine model type.

ii. In one example, combinations of constructed affine candidates may be re-ordered with 4-parameter affine candidates (2 CPMVs) added before 6-parameter affine candidates.

d. For the affine merge candidate list and/or sub-block merge candidate list, more constructed affine candidates with the same affine model type as the affine model type of a selected merge candidate may be constructed.

i. In one example, the selected merge candidate is the first available affine merge candidate.

ii. In one example, the selected merge candidate is the affine merge candidate associated with certain position of a spatial neighboring block.

e. For the affine merge candidate list and/or sub-block merge candidate list, order of constructed affine candidates may be dependent on the affine model type of a selected affine merge candidate.

i. In one example, the selected merge candidate is the first available affine merge candidate.

ii. In one example, the selected merge candidate is the affine merge candidate associated with certain position of a spatial neighboring block.

4. It is proposed that the affine model type associated with a block is not stored and utilized for coding following blocks.

a. Alternatively, such information is could be stored but only used for coding following blocks within the current CTU or within the same MxN region, or current CTU row. In one example, one picture/tile/slice may be split to non-overlapped regions with sizes equal to MxN, e.g., 64x64.

b. In one example, after decoding an AMVP-affine-coded block with 4-parameter affine model, instead of storing 2 CPMVs (from top-left and top-right positions), 3 CPMVs (from top-left, top-right and bottom-left positions) may be stored.

i. In one example, the top-left and top-right CPMVs may be utilized to derive the bottom-left CPMV.

c. In one example, for each affine merge candidate, the 6-parameter affine model is utilized. Alternatively, for each affine merge candidate, the 4-parameter affine model is utilized.

5. It is proposed that the affine candidates may be reordered instead of using fixed insertion order.
   a. In one example, the reordering is dependent on derived MVs of representative neighboring positions relative to the current block. Each affine candidate is used to derive motion vectors of several representative neighboring positions, and then differences of the derived MVs and the decoded MVs associated with those representative neighboring positions are calculated. Finally, affine candidates are reordered in ascending order of the differences.
   b. In one example, the difference metric is the MSE (mean squared error).
   c. Alternatively, furthermore, before calculating the difference, the derived MVs may be further scaled if the affine candidates have different reference pictures from the representative neighboring blocks.
   d. Alternatively, furthermore, before calculating the difference, both derived MVs and representative neighboring MVs may be scaled to some selected reference pictures.
   e. In one example, only some of the affine candidates are reordered. For example, only neighboring affine candidates are reordered. They may always be inserted before the constructed affine candidates.
   f. In one example, only constructed affine candidates are reordered. They may always be inserted after the neighboring affine candidates.
   g. In one example, only the first N affine candidates are reordered.
   h. In one example, only the first N of the reordered affine merge candidate are inserted into the sub-block merge list.
   i. In one example, if such reordering is performed, maximum length of the sub-block merge list is reduced by K. For example, K=2.

6. It is proposed the reordering method described in bullet 4 may be applied to affine AMVP list construction.
    a. In one example, no affine AMVP index is signaled and only the first of the reordered affine AMVP candidates is used as the predictor.
7. It is proposed that multiple (for example, 2) affine candidates may be averaged to generate new affine candidates.
    a. In one example, only affine candidates with same reference pictures are used for generating the average affine candidates.
    b. In one example, affine candidates with different reference pictures may be used to generate the average affine candidates, and all affine candidates are scaled to the same reference pictures.
        i. In one example, reference pictures of anyone of these affine candidates may be used as reference pictures of the average affine candidates.
        ii. In one example, reference pictures of the average affine candidates may be defined for each CU/tile/slice/picture/video/tile and may be signaled in tile head/slice header/PPS/VPS/SPS.
        iii. In one example, reference pictures are predefined implicitly at both encoder and decoder.
        iv. In one example, scaling is not performed.

[00196] FIG. 26 is a block diagram of a video processing apparatus 2600. The apparatus 2600 may be used to implement one or more of the methods described herein. The apparatus 2600 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 2600 may include one or more processors 2602, one or more memories 2604 and video processing hardware 2606. The processor(s) 2602 may be configured to implement one or more methods described in the present document. The memory (memories) 2604 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 2606 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00197] FIG. 27 is a flowchart for an example method 2700 of video processing. The method may be performed by a video encoder, in its decode loop, or by a video decoder. The method

2700 includes deriving(2702) for a current block, using motion information related to a first reference picture from a single reference picture list, motion information related to a second reference picture from another reference picture list different from the single reference list; and performing(2704) a video processing on the current block based on the motion information related to the first reference picture and the motion information related to the second reference picture.

[00198] FIG. 28 is a flowchart for an example method 2800 of video processing. The method may be performed by a video encoder, in its decode loop, or by a video decoder. The method 2800 includes determining(2802) an affine mode type associated with a current block; and performing(2804)a video processing on the current block based on the determined affine mode type; wherein the affine mode type is restricted for use to a portion of a video picture in which the current block is located and the determined affine mode type is prevented from use to other portions of the video picture.

[00199] FIG. 29 is a flowchart for an example method 2900 of video processing. The method may be performed by a video encoder, in its decode loop, or by a video decoder. The method 2900 includes generating(2902) an affine candidate list for a current block, wherein the affine candidate list comprises an additional affine candidate which is generated by averaging a plurality of affine candidates from the affine candidate list; and performing(2904) a video processing on the current block based on the expanded affine candidate list.

[00200] It will be appreciated that several techniques have been disclosed that will benefit video encoder and decoder embodiments incorporated within video processing devices such as smartphones, laptops, desktops, and similar devices by allowing the use of affine models in video compression and decompression as described in numerous techniques and embodiments in the present document.

[00201] Some embodiments may be described using the following examples.

[00202] 1. A method for video processing, comprising:

deriving for a current block, using motion information related to a first reference picture from a single reference picture list, motion information related to a second reference picture from another reference picture list different from the single reference list; and

performing a video processing on the current block based on the motion information related to the first reference picture and the motion information related to the second reference picture.

[00203] 2. The method of example 1, wherein the motion information related to the first reference picture is signalled, and the deriving the motion information related to the second reference picture comprises:

deriving the motion information related to the second reference picture from the signalled motion information related to first reference picture.

[00204] 3. The method of any one of examples 1-2, wherein the motion information comprises control point motion vectors(CPMVs), and the CPMVs related to the second reference picture are derived from the CPMVs related to the first reference picture.

[00205] 4. The method of any one of examples 1-2, wherein the motion information comprises motion vector differences(MVDs), and the MVDs related to the second reference picture are derived from the MVDs related to the first reference picture.

[00206] 5. The method of any one of examples 1-4, wherein performing the video processing comprise performing a prediction for the current block based on a bi-prediction affine mode.

[00207] 6. The method of any one of examples 1-4, wherein performing the video processing comprise performing a prediction for the current block based on a symmetric affine mode.

[00208] 7. The method of example 6, wherein the symmetric affine mode is set to a bi-directional affine mode.

[00209] 8.The method of example 6, wherein the symmetric affine mode is a mode independent from a bi-directional affine mode and a uni-directional affine mode; or selectable affine modes for the current block comprises the symmetric affine mode and does not comprises a uni-directional affine mode.

[00210] 9. A method for video processing, comprising:

determining an affine mode type associated with a current block; and

performing a video processing on the current block based on the determined affine mode type;

wherein the affine mode type is restricted for use to a portion of a video picture in which the current block is located and the determined affine mode type is prevented from use to other portions of the video picture.

[00211] 10. The method of example 9, wherein the portion of the video picture is the current block, and the affine mode type is not stored.

[00212] 11. The method of example 9 or 10, wherein the portion of the video picture corresponds to at least one of:

a coding tree unit(CTU) in which the current block is located;

a M×N region in which the current block is located; and

a CTU row in which the current block is located.

[00213] 12. The method of example 11, wherein the M×N region is one of a plurality of non-overlapped M×N regions with a same size split from at least one of a picture, tile, and slice.

[00214] 13. The method of example 11 or 12, wherein M=N=64.

[00215] 14. The method of any one of examples 9-13, wherein the motion information associated with the current block comprise control point motion vectors(CPMVs) of the current block, the method comprising:

storing three control point motion vectors(CPMVs) for the current block regardless whether 6-parameter affine model or 4-parameter affine model being applied to the current block.

[00216] 15. The method of example 14, wherein the three CPMVs comprises CPMVs located respectively at a top-left corner, a top-right corner and a bottom-left corner of the current block.

[00217] 16. The method of example 15, further comprising:

[00218] deriving the CPMV located at the bottom-left corner of the current block from the CPMVs located respectively at the top-left and top-right corners of the current block.

[00219] 17. The method of any one of examples 14-16, wherein an advanced motion vector prediction (AMVP) mode is applied to the current block.

[00220] 18. The method of any one of examples 14-16, wherein an affine merge mode is applied to the current block, the method comprising:

deriving the motion information of the current block from a plurality of affine merge candidates each utilizes 6-parameter affine model or 4-parameter affine model.

[00221] 19. A method for video processing, comprising:

generating an affine candidate list for a current block, wherein the affine candidate list comprises an additional affine candidate which is generated by averaging a plurality of affine candidates from the affine candidate list; and

performing a video processing on the current block based on the expanded affine candidate list.

[00222] 20. The method of example 19, wherein an amount of the plurality of affine candidates is 2.

[00223] 21. The method of example 19 or 20, wherein the plurality of affine candidates refers to a same reference picture.

[00224] 22. The method of example 19 or 20, wherein at least two of the plurality of affine candidates refer to different reference pictures.

[00225] 23. The method of example 22, wherein all of the plurality of affine candidates are scaled to a same reference picture for the additional candidate.

[00226] 24. The method of example 22 or 23, wherein a reference picture to which any one of the plurality of affine candidates refer is used as a reference picture for the additional candidate.

[00227] 25. The method of any one of examples 21-24, wherein the reference picture for the additional candidate is defined for at least one of a coding unit level, a tile level, a slice level, a picture level, a video level.

[00228] 26. The method of any one of examples 21-25, wherein the reference picture for the additional candidate is signalled in at least one of a tile header, a slice header, a picture parameter set(PPS), a video parameter set(VPS), a sequence parameter set(SPS).

[00229] 27. The method of any one of examples 21-25, wherein the reference picture for the additional candidate is predefined implicitly at both encoding and decoding sides.

[00230] 28. The method of any one of examples 1-27, wherein the video processing comprises at least one of encoding the video block into the bitstream representation of the video block and decoding the video block from the bitstream representation of the video block.

[00231] 29. A video processing apparatus comprising a processor configured to implement the method of any one of examples 1 to 28.

[00232] 30. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of examples 1 to 28.

[00233] 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.

[00234] 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.

[00235] 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).

[00236] 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.

[00237] 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.

[00238] 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.

[00239] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this patent document.

CLAIMS

1. A method for video processing, comprising:

deriving for a current block, using motion information related to a first reference picture from a single reference picture list, motion information related to a second reference picture from another reference picture list different from the single reference list; and

performing a video processing on the current block based on the motion information related to the first reference picture and the motion information related to the second reference picture.

2. The method of claim 1, wherein the motion information related to the first reference picture is signalled, and the deriving the motion information related to the second reference picture comprises:

deriving the motion information related to the second reference picture from the signalled motion information related to first reference picture.

3. The method of any one of claims 1-2, wherein the motion information comprises control point motion vectors(CPMVs), and the CPMVs related to the second reference picture are derived from the CPMVs related to the first reference picture.

4. The method of any one of claims 1-2, wherein the motion information comprises motion vector differences(MVDs), and the MVDs related to the second reference picture are derived from the MVDs related to the first reference picture.

5. The method of any one of claims 1-4, wherein performing the video processing comprise performing a prediction for the current block based on a bi-prediction affine mode.

6. The method of any one of claims 1-4, wherein performing the video processing comprise performing a prediction for the current block based on a symmetric affine mode.

7. The method of claim 6, wherein the symmetric affine mode is set to a bi-directional affine mode.

8. The method of claim 6, wherein the symmetric affine mode is a mode independent from a bi-directional affine mode and a uni-directional affine mode; or selectable affine modes for the current block comprises the symmetric affine mode and does not comprises a uni-directional affine mode.

9. A method for video processing, comprising:

determining an affine mode type associated with a current block; and

performing a video processing on the current block based on the determined affine mode type;

wherein the affine mode type is restricted for use to a portion of a video picture in which the current block is located and the determined affine mode type is prevented from use to other portions of the video picture.

10. The method of claim 9, wherein the portion of the video picture is the current block, and the affine mode type is not stored.

11. The method of claim 9 or 10, wherein the portion of the video picture corresponds to at least one of:

a coding tree unit(CTU) in which the current block is located;

a M×N region in which the current block is located; and

a CTU row in which the current block is located.

12. The method of claim 11, wherein the M×N region is one of a plurality of non-overlapped M×N regions with a same size split from at least one of a picture, tile, and slice.

13. The method of claim 11 or 12, wherein M=N=64.

14. The method of any one of claims 9-13, wherein the motion information associated with the current block comprise control point motion vectors(CPMVs) of the current block, the method comprising:

storing three control point motion vectors(CPMVs) for the current block regardless whether 6-parameter affine model or 4-parameter affine model being applied to the current block.

15. The method of claim 14, wherein the three CPMVs comprises CPMVs located respectively at a top-left corner, a top-right corner and a bottom-left corner of the current block.

16. The method of claim 15, further comprising:

deriving the CPMV located at the bottom-left corner of the current block from the CPMVs located respectively at the top-left and top-right corners of the current block.

17. The method of any one of claims 14-16, wherein an advanced motion vector prediction (AMVP) mode is applied to the current block.

18. The method of any one of claims 14-16, wherein an affine merge mode is applied to the current block, the method comprising:

deriving the motion information of the current block from a plurality of affine merge candidates each utilizes 6-parameter affine model or 4-parameter affine model.

19. A method for video processing, comprising:

generating an affine candidate list for a current block, wherein the affine candidate list comprises an additional affine candidate which is generated by averaging a plurality of affine candidates from the affine candidate list; and

performing a video processing on the current block based on the expanded affine candidate list.

20. The method of claim 19, wherein an amount of the plurality of affine candidates is 2.

21. The method of claim 19 or 20, wherein the plurality of affine candidates refers to a same reference picture.

22. The method of claim 19 or 20, wherein at least two of the plurality of affine candidates refer to different reference pictures.

23. The method of claim 22, wherein all of the plurality of affine candidates are scaled to a same reference picture for the additional candidate.

24. The method of claim 22 or 23, wherein a reference picture to which any one of the plurality of affine candidates refer is used as a reference picture for the additional candidate.

25. The method of any one of claims 21-24, wherein the reference picture for the additional candidate is defined for at least one of a coding unit level, a tile level, a slice level, a picture level, a video level.

26. The method of any one of claims 21-25, wherein the reference picture for the additional candidate is signalled in at least one of a tile header, a slice header, a picture parameter set(PPS), a video parameter set(VPS), a sequence parameter set(SPS).

27. The method of any one of claims 21-25, wherein the reference picture for the additional candidate is predefined implicitly at both encoding and decoding sides.

28. The method of any one of claims 1-27, wherein the video processing comprises at least one of encoding the video block into the bitstream representation of the video block and decoding the video block from the bitstream representation of the video block.

29. A video processing apparatus comprising a processor configured to implement the method of any one of claims 1 to 28.

30. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of claims 1 to 28.

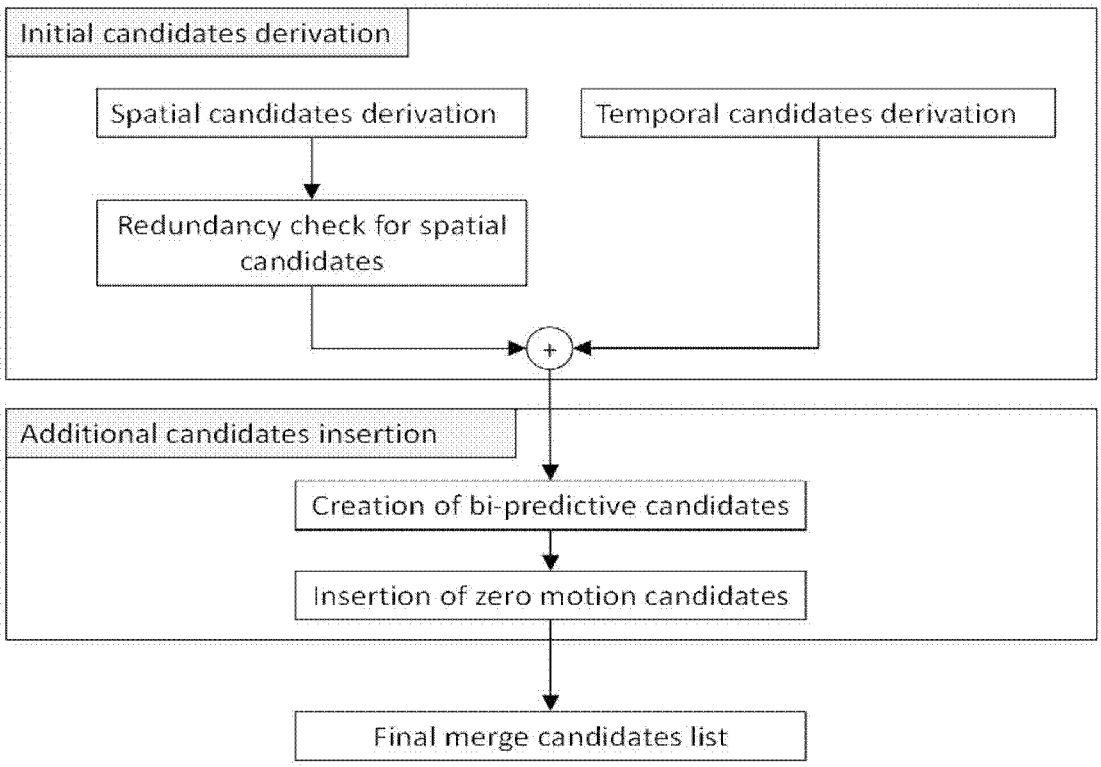

Initial candidates derivation
Spatial candidates derivation
Temporal candidates derivation
Redundancy check for spatial candidates
+
Additional candidates insertion
Creation of bi-predictive candidates
Insertion of zero motion candidates
Final merge candidates list

FIG. 1

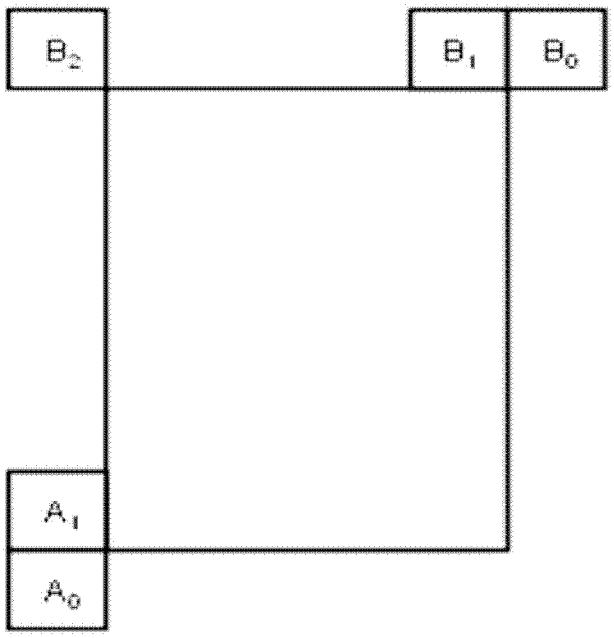


FIG. 2

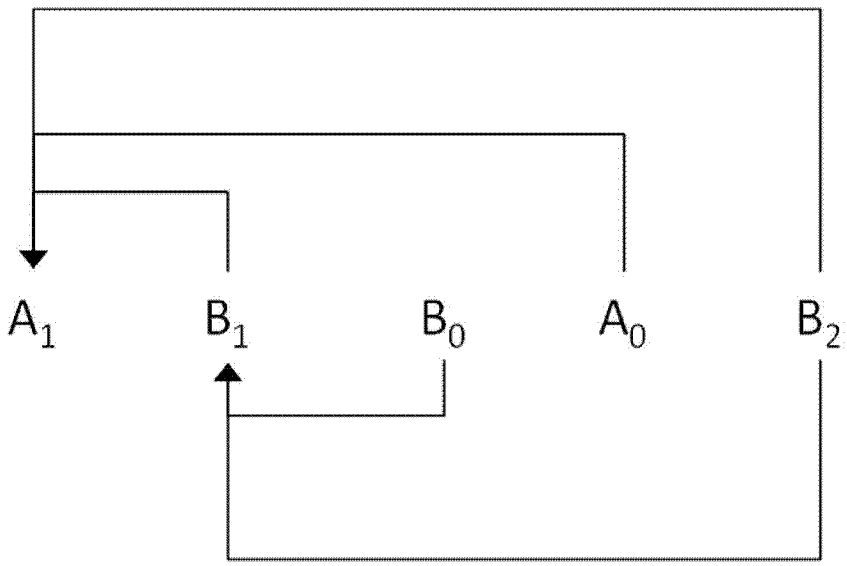
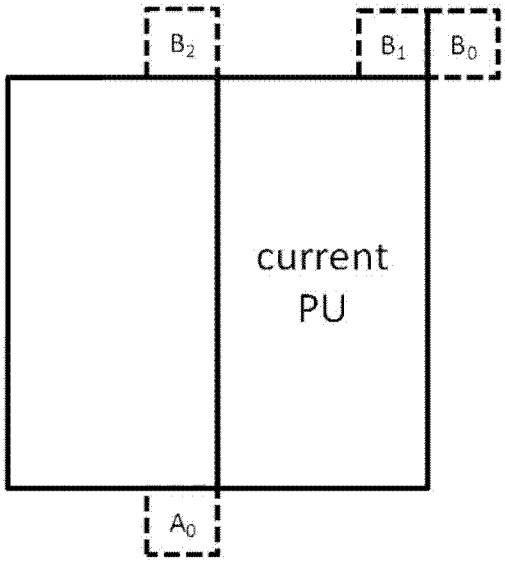
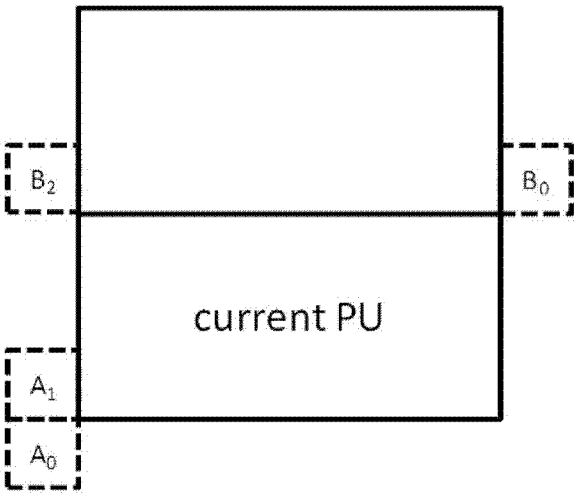


FIG. 3



(a) second PU of Nx2N

FIG. 4A



(b)second PU of 2NxN

FIG. 4B

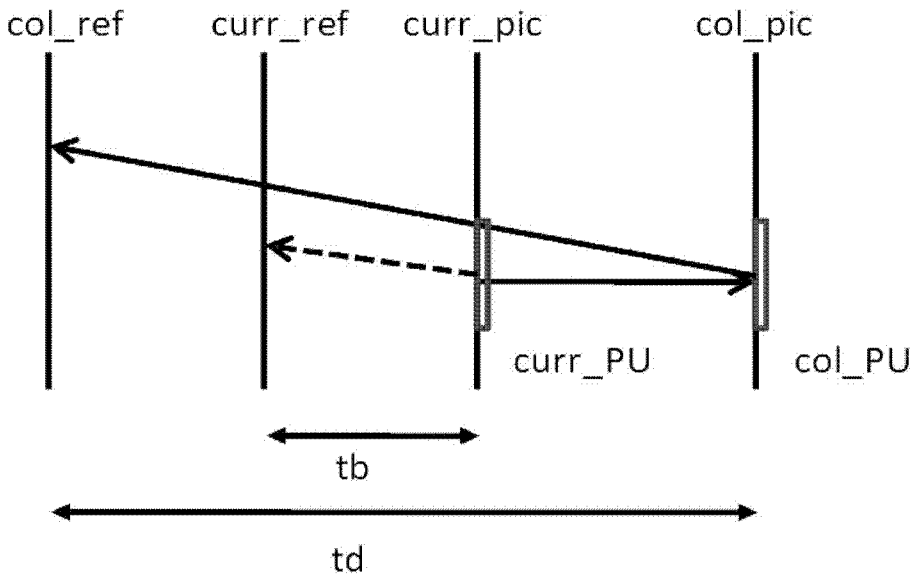


FIG. 5

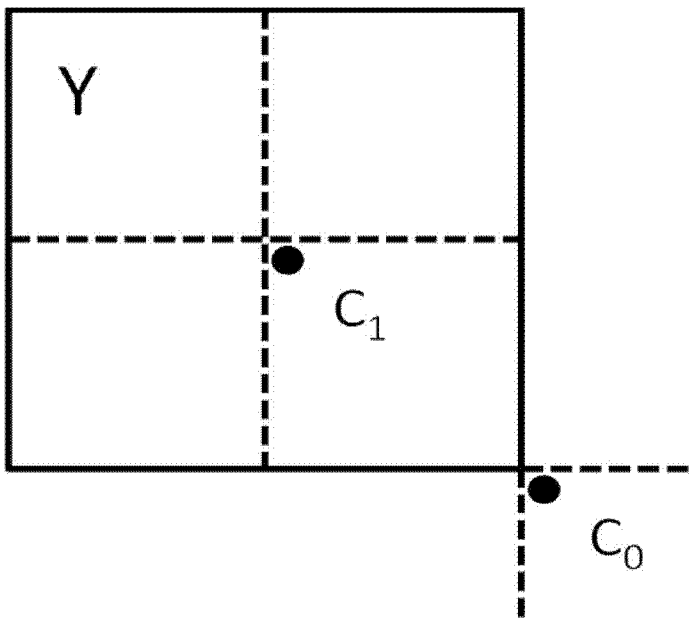


FIG. 6

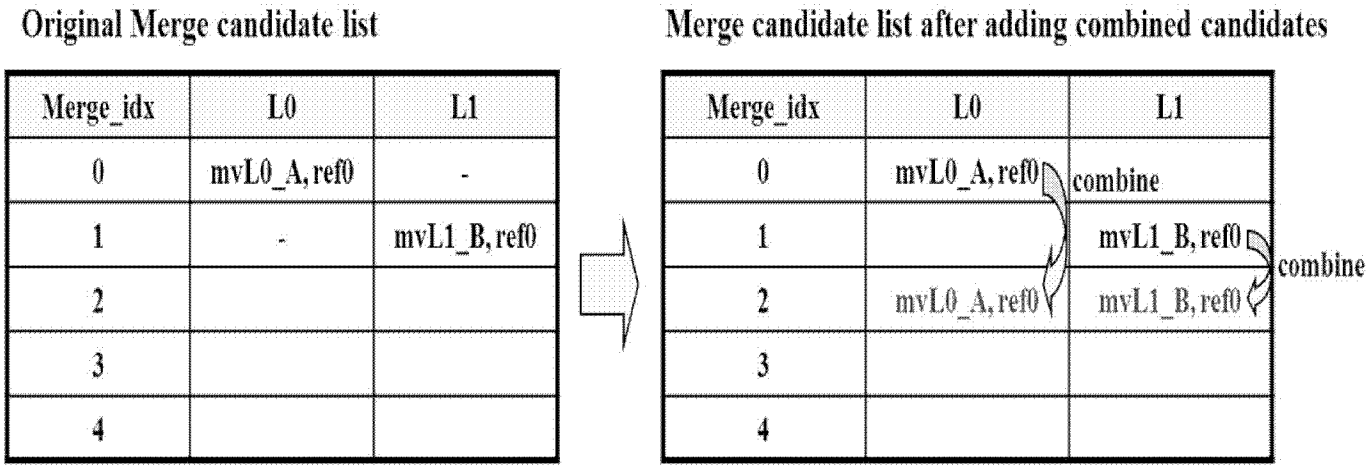

Original Merge candidate list
Merge_idx
L0
L1
0
mvL0_A, ref0
-
1
-
mvL1_B, ref0
2
3
4
Merge candidate list after adding combined candidates
Merge_idx
L0
L1
0
mvL0_A, ref0
combine
1
mvL1_B, ref0
combine
2
mvL0_A, ref0
mvL1_B, ref0
3
4

FIG. 7

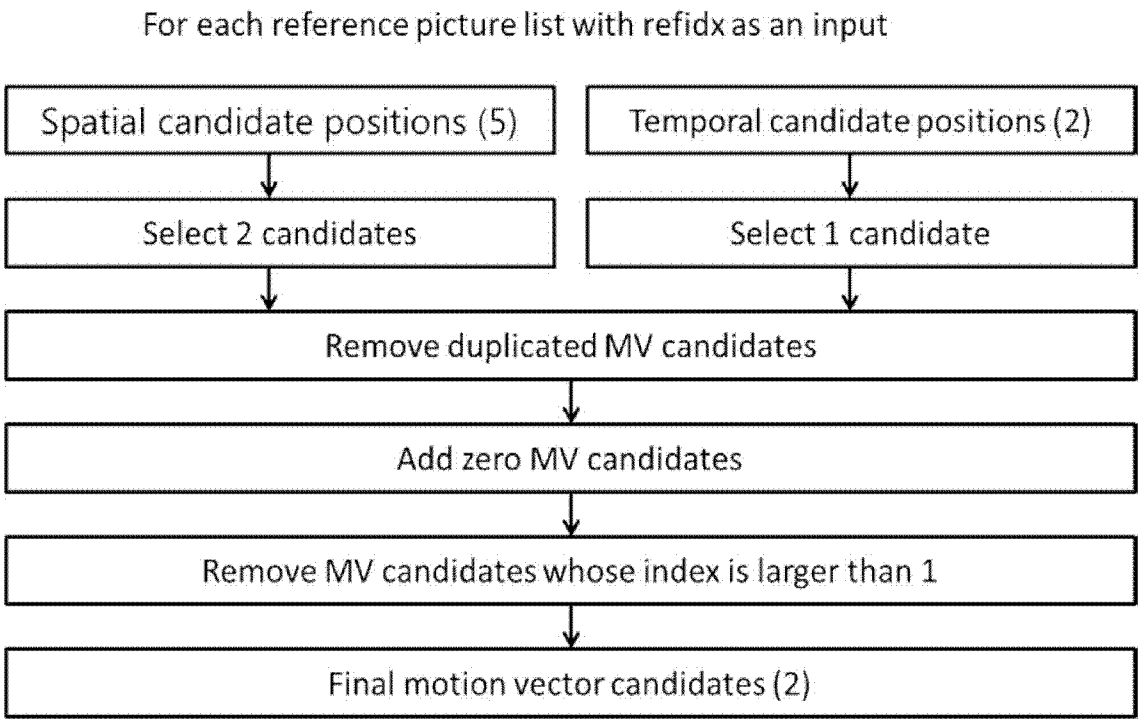

For each reference picture list with refidx as an input
Spatial candidate positions (5)
Temporal candidate positions (2)
Select 2 candidates
Select 1 candidate
Remove duplicated MV candidates
Add zero MV candidates
Remove MV candidates whose index is larger than 1
Final motion vector candidates (2)

FIG. 8

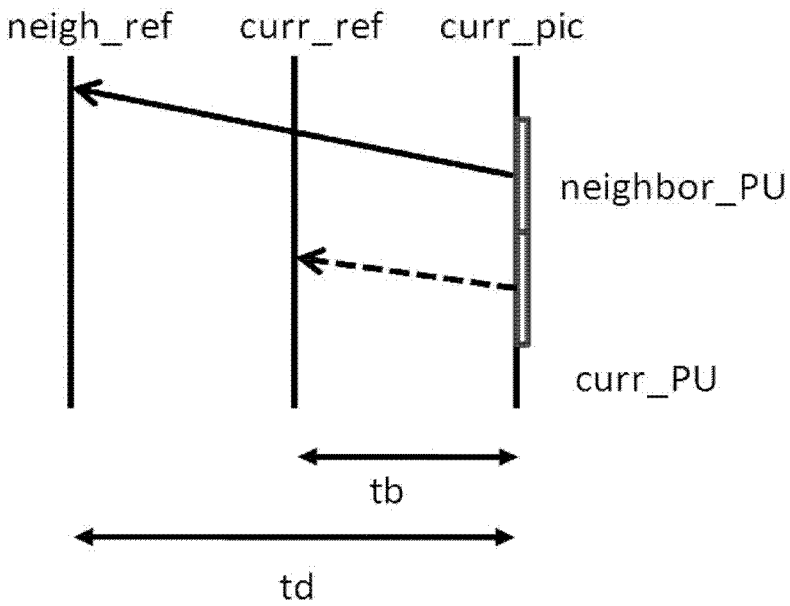


FIG. 9

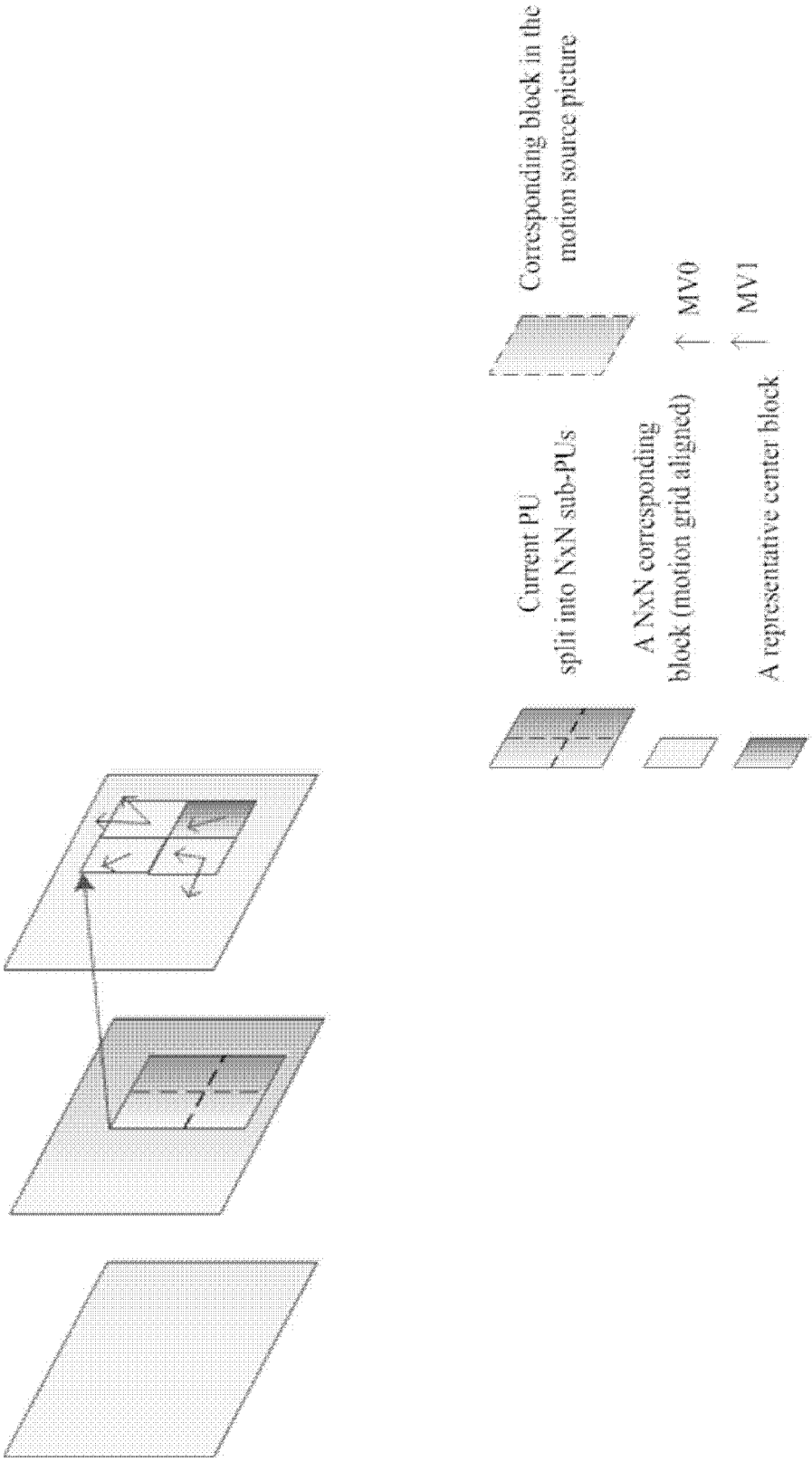

Corresponding block in the motion source picture
MV0
MV1
Current PU split into NxN sub-PUs
A NxN corresponding block (motion grid aligned)
A representative center block

FIG. 10

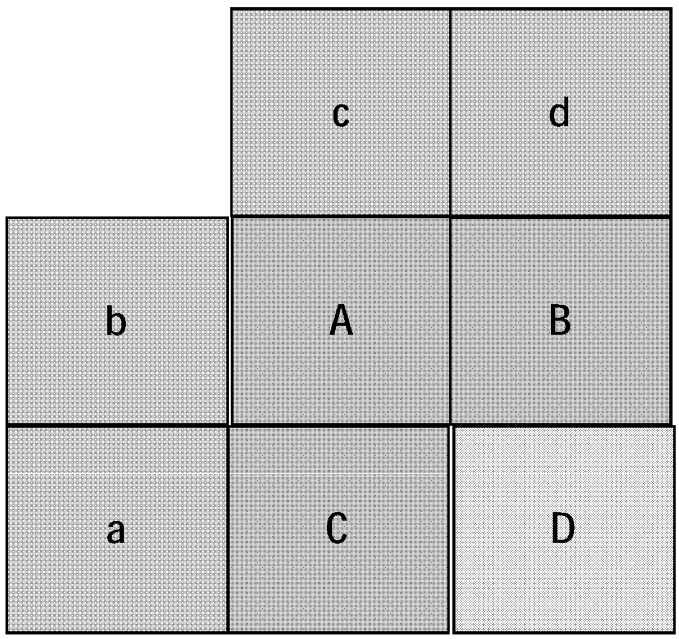


FIG. 11

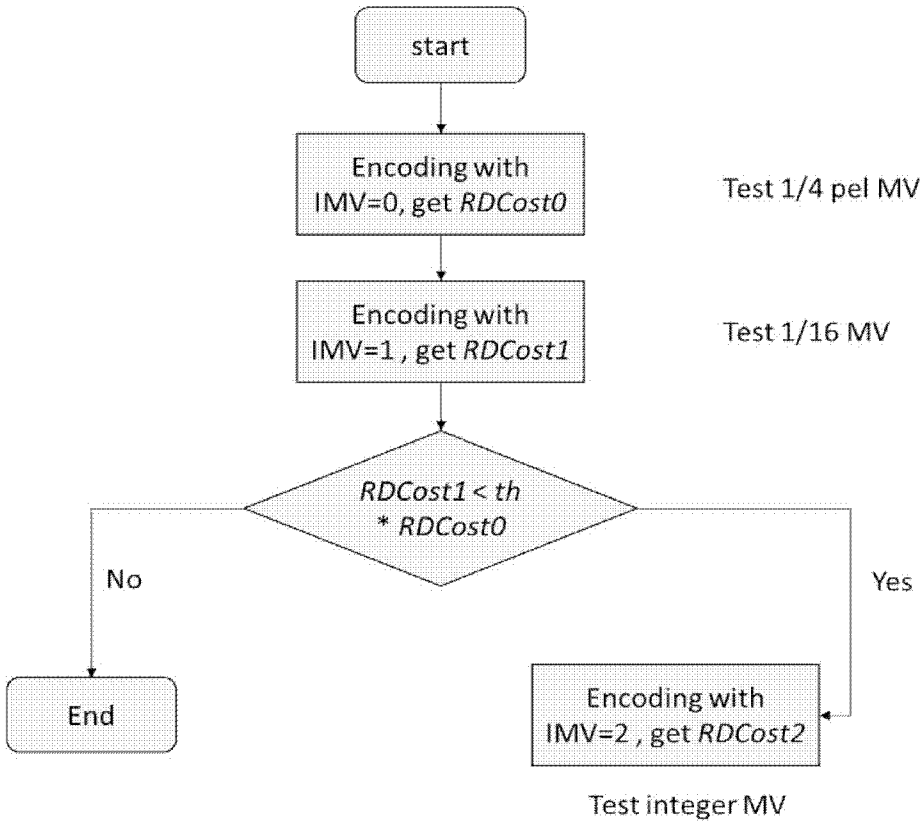

start
Encoding with IMV=0, get RDCost0
Test 1/4 pel MV
Encoding with IMV=1 , get RDCost1
Test 1/16 MV
RDCost1 < th * RDCost0
No
Yes
End
Encoding with IMV=2 , get RDCost2
Test integer MV

FIG. 12

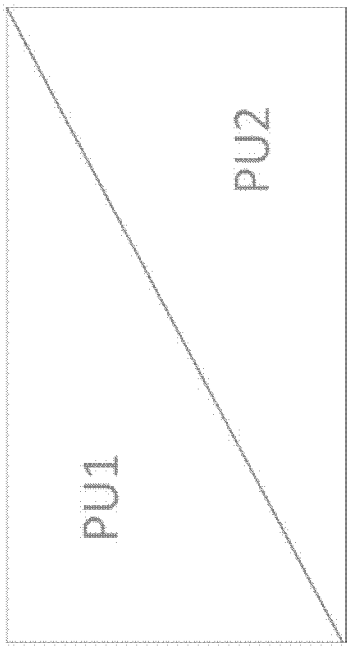


FIG. 13B

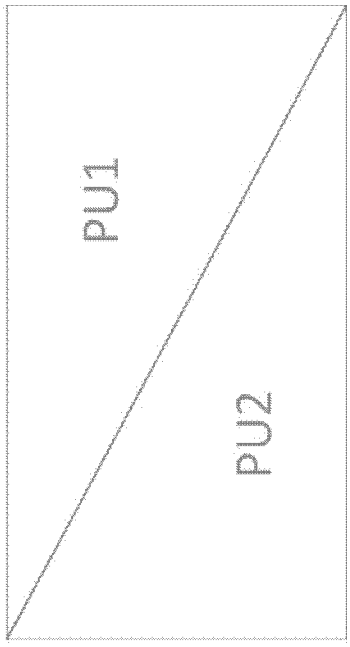


FIG. 13A

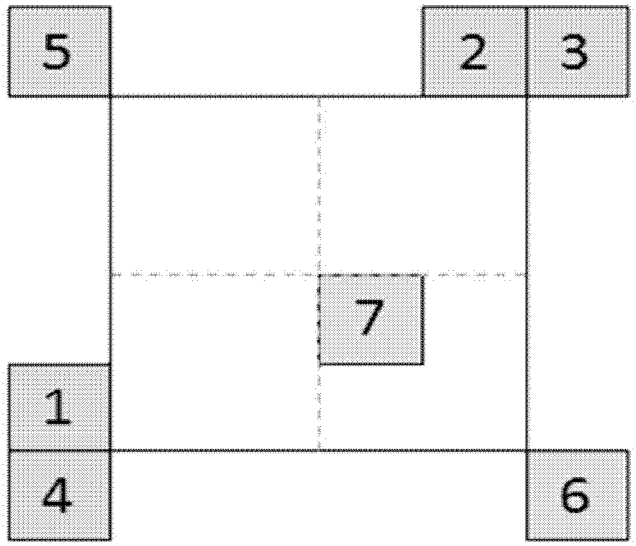


FIG. 14

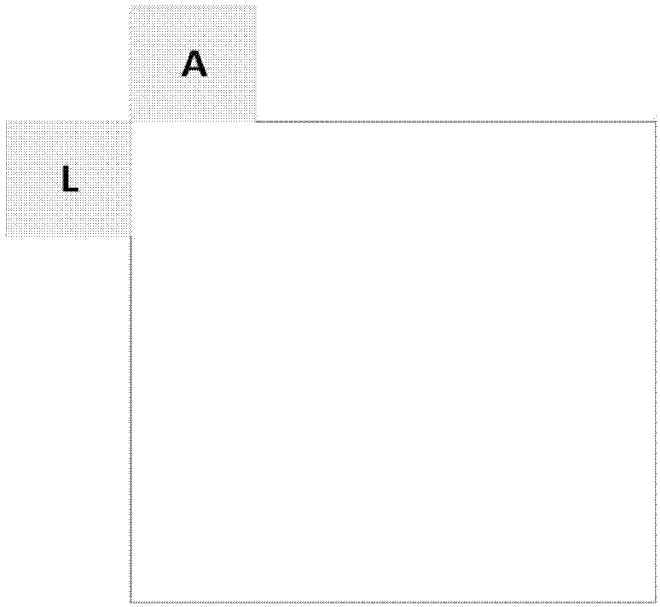


FIG. 15

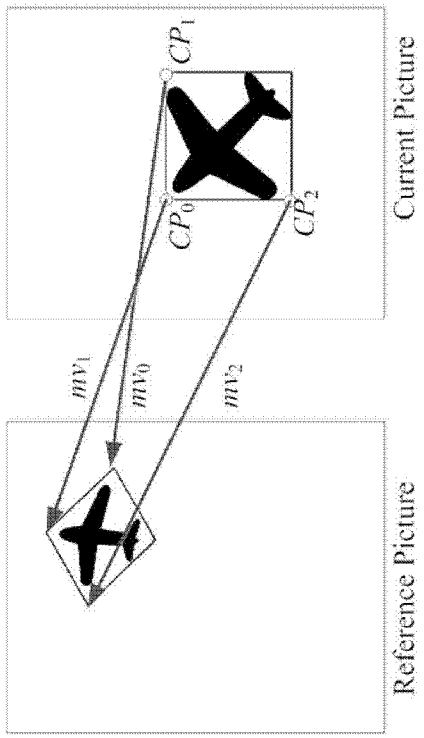


FIG. 16B

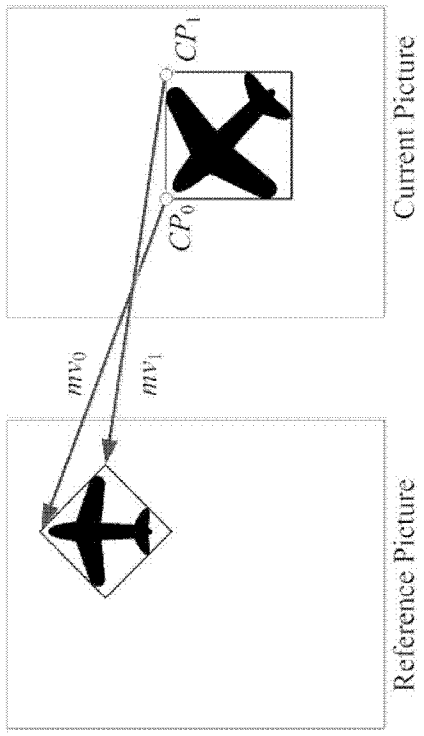


FIG. 16A

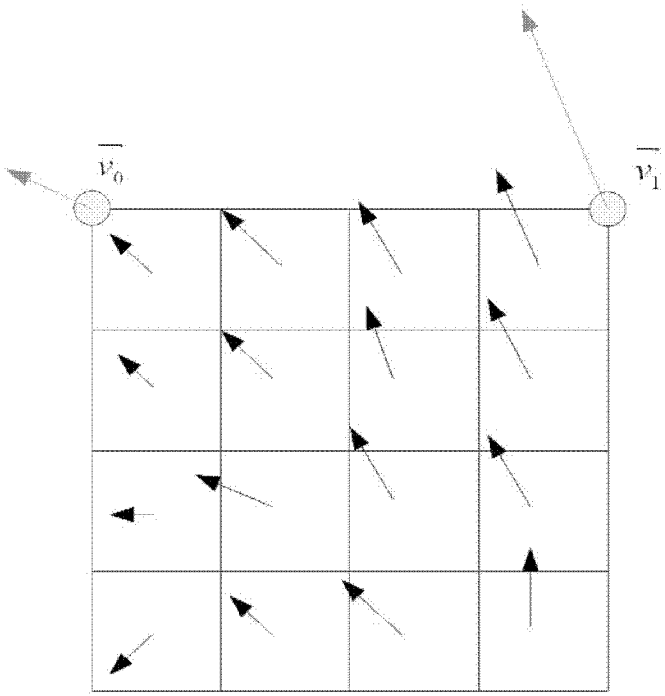


FIG. 17

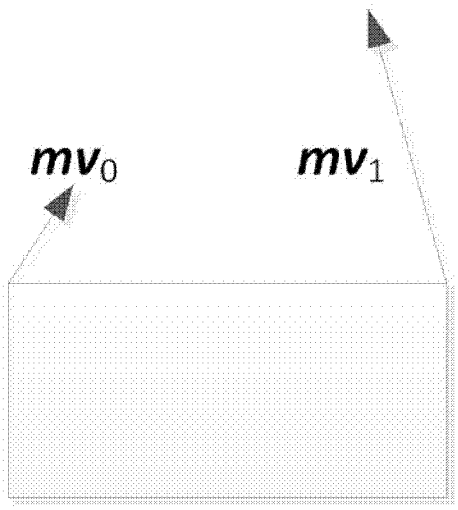


FIG. 18A

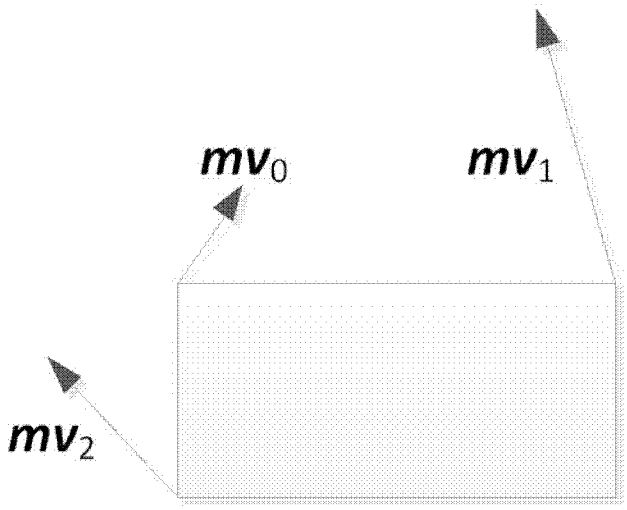


FIG. 18B

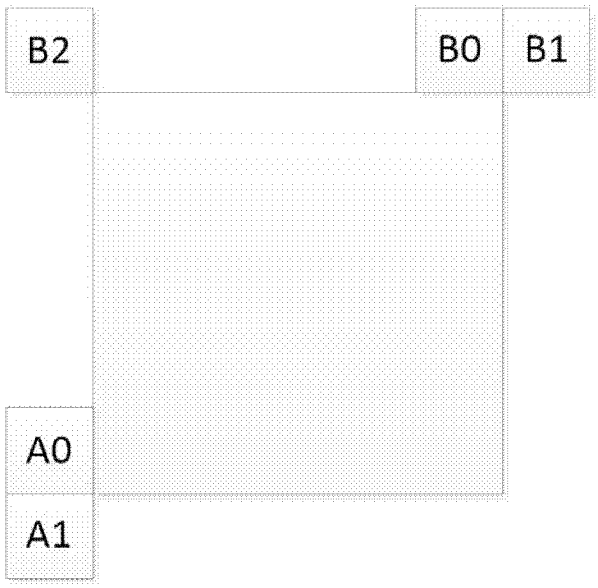


FIG. 19

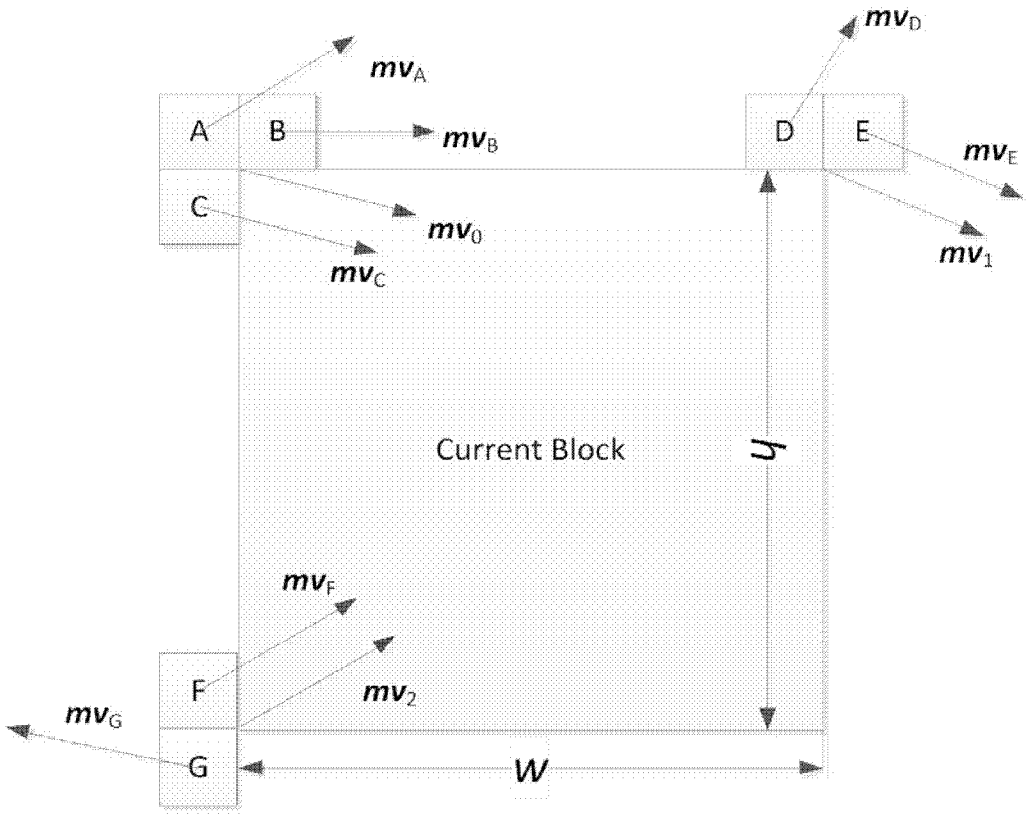

mv_A
mv_D
A
B
D
E
mv_B
mv_E
C
mv_0
mv_1
mv_C
Current Block
H
mv_F
F
mv_2
mv_G
G
W

FIG. 20

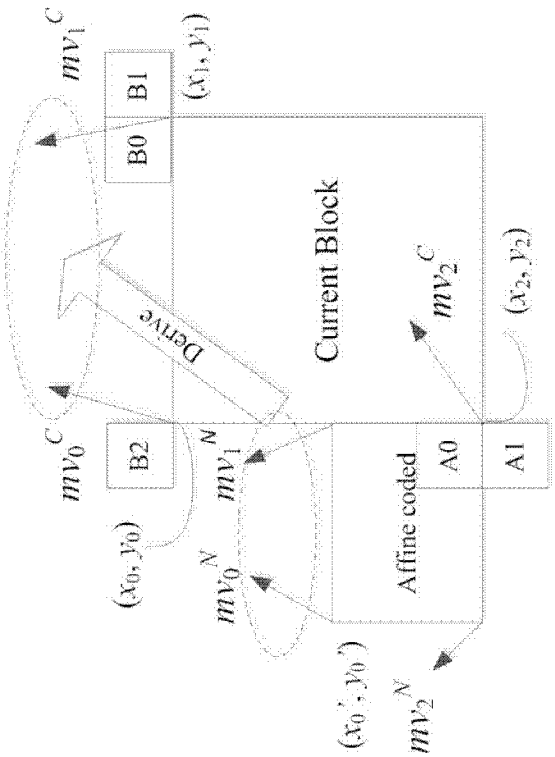


FIG. 21B

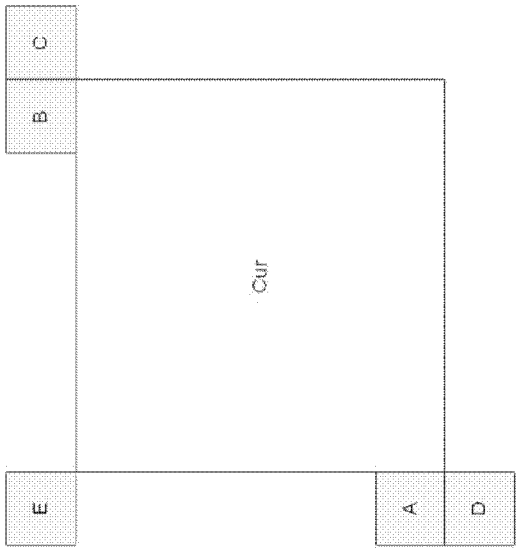


FIG. 21A

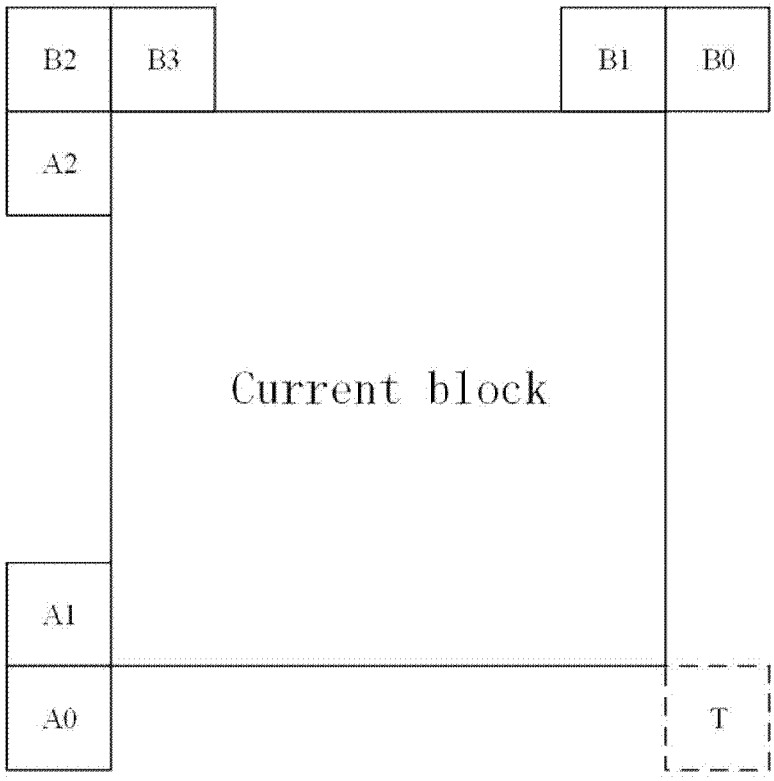


FIG. 22

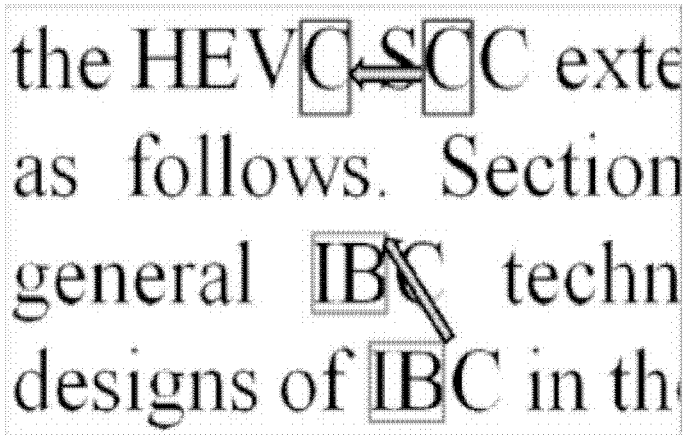

the HEVC SCC exte
as follows. Section
general IBC techn
designs of IBC in th

FIG. 23

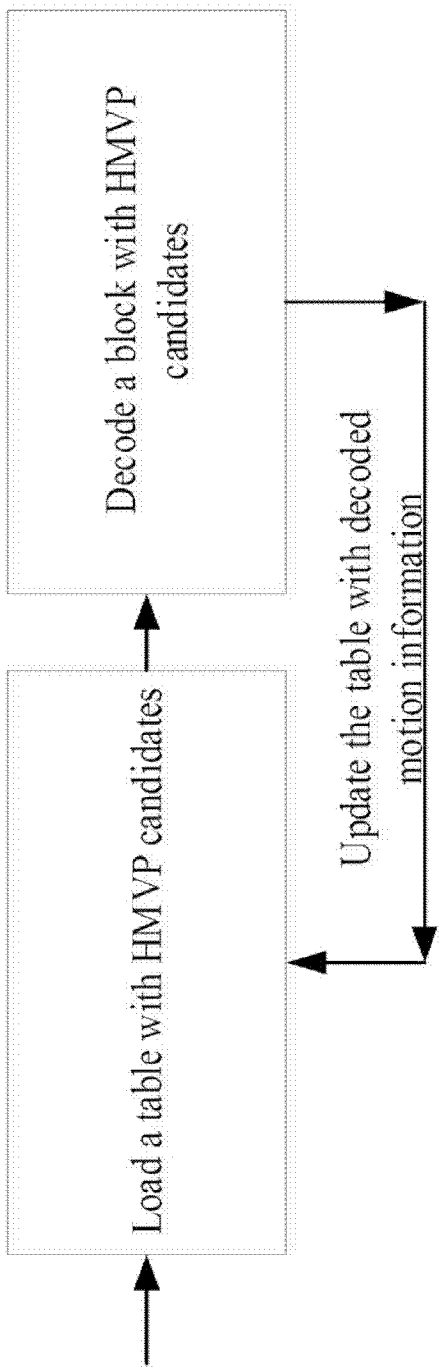

Load a table with HMVP candidates
Decode a block with HMVP candidates
Update the table with decoded motion information

FIG. 24

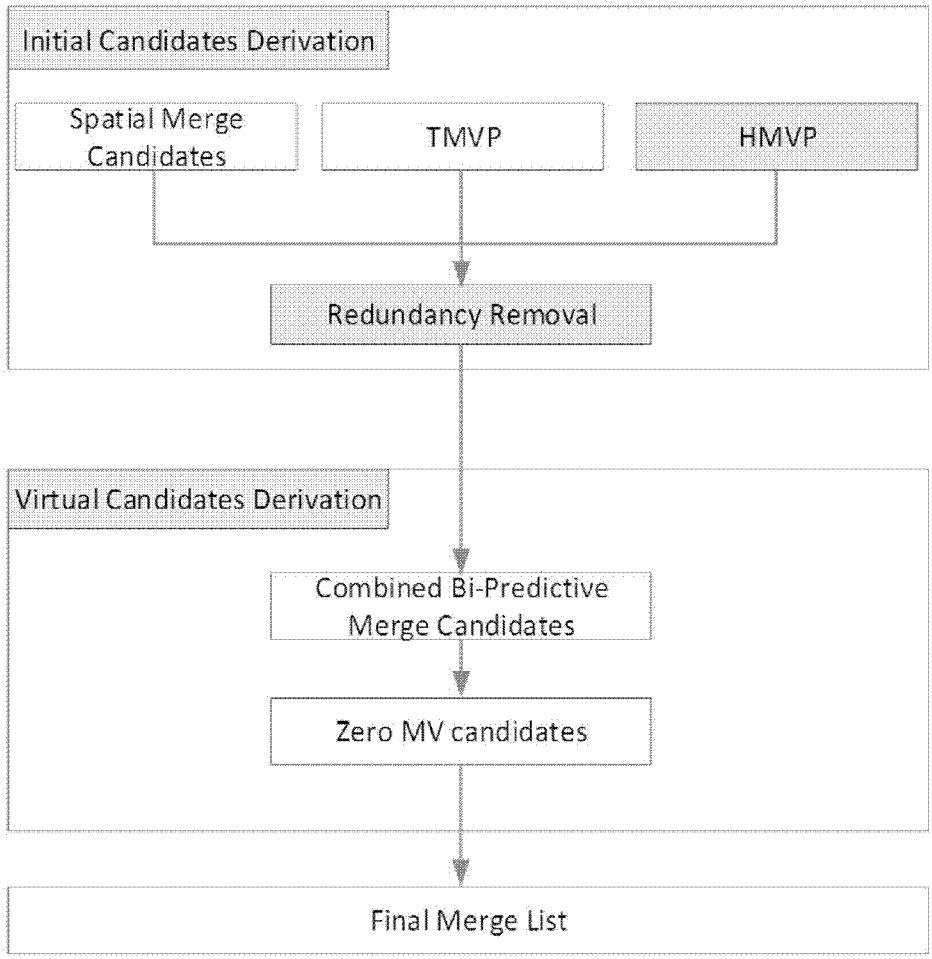

Initial Candidates Derivation
Spatial Merge Candidates
TMVP
HMVP
Redundancy Removal
Virtual Candidates Derivation
Combined Bi-Predictive Merge Candidates
Zero MV candidates
Final Merge List

FIG. 25

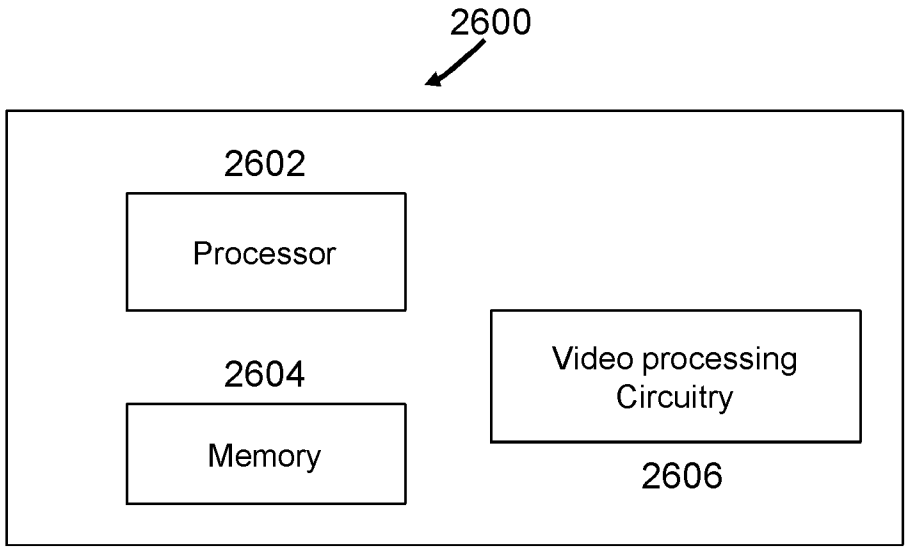

2600
2602
Processor
2604
Memory
Video processing
Circuitry
2606

FIG. 26

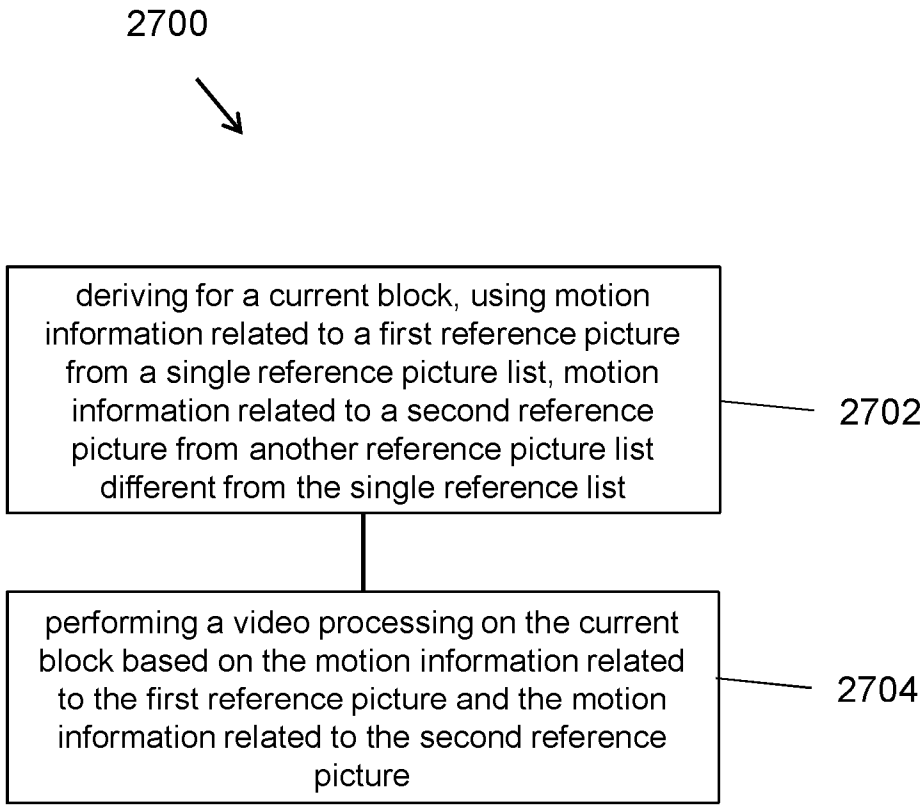

2700
deriving for a current block, using motion information related to a first reference picture from a single reference picture list, motion information related to a second reference picture from another reference picture list different from the single reference list
2702
performing a video processing on the current block based on the motion information related to the first reference picture and the motion information related to the second reference picture
2704

FIG. 27

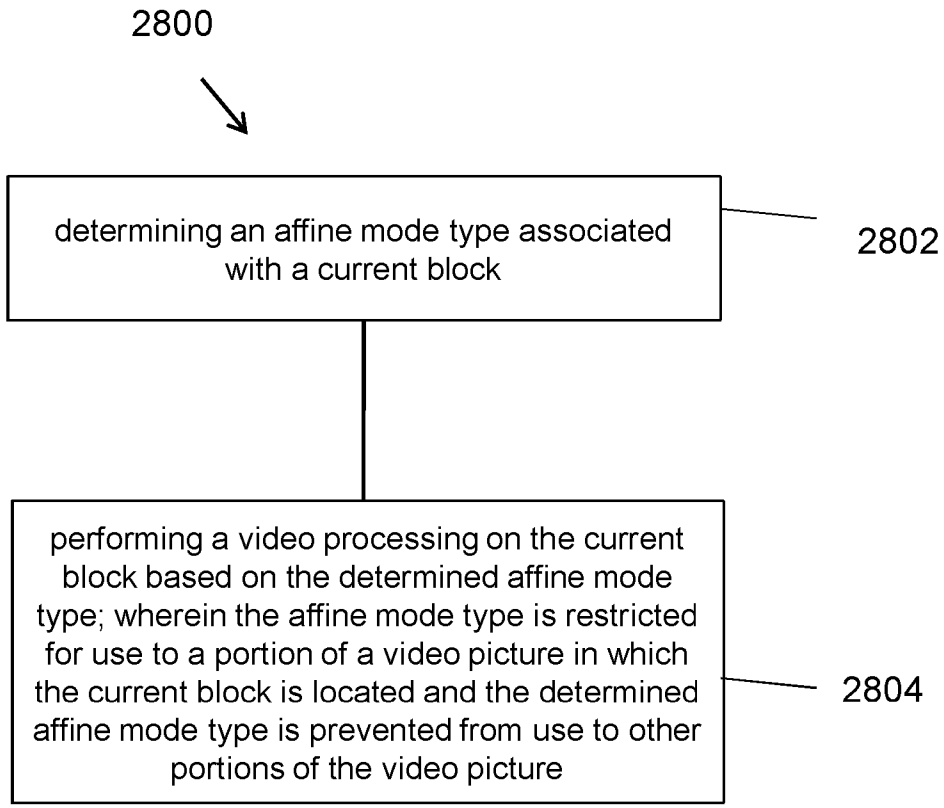

2800
determining an affine mode type associated with a current block
2802
performing a video processing on the current block based on the determined affine mode type; wherein the affine mode type is restricted for use to a portion of a video picture in which the current block is located and the determined affine mode type is prevented from use to other portions of the video picture
2804

FIG. 28

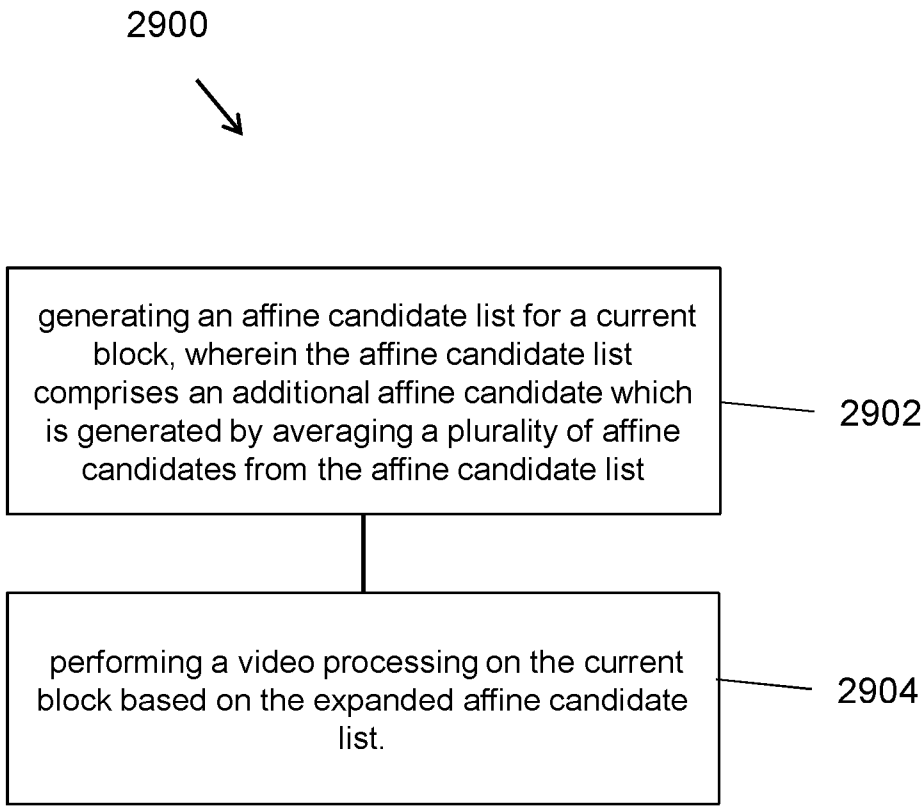

2900
generating an affine candidate list for a current block, wherein the affine candidate list comprises an additional affine candidate which is generated by averaging a plurality of affine candidates from the affine candidate list
2902
performing a video processing on the current block based on the expanded affine candidate list.
2904

FIG. 29

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2019/118528

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/52(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

CNKI, CNABS, CNTXT, IEEE, JCTVC, JVET, VEN, WOTXT, USTXT: code, encode, decode, reference, picture, list, multiple, two, bi, second, block, affine, mode, model, type, CTU, region, portion, candidate, average, calculate

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2018067823 A1 (QUALCOMM INC) 12 April 2018 (2018-04-12) description, paragraphs 0048 to 0247 and figures 1-21A	1-30
X	WO 2017200771 A1 (QUALCOMM INC) 23 November 2017 (2017-11-23) description, paragraphs 0028 to 0193 and figures 1-14	1-30
X	WO 2018064495 A1 (QUALCOMM INC) 05 April 2018 (2018-04-05) description, paragraphs 0071 to 0278 and figures 1-20	1-30
X	WO 2018200960 A1 (QUALCOMM INC) 01 November 2018 (2018-11-01) description, paragraphs 0029 to 0201 and figures 1-13	1-30
X	US 2018270500 A1 (QUALCOMM INC) 20 September 2018 (2018-09-20) description, paragraphs 0060 to 0229 and figures 1-17	1-30

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

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

Date of the actual completion of the international search	Date of mailing of the international search report
19 January 2020	**06 February 2020**
Name and mailing address of the ISA/CN **National Intellectual Property Administration, PRC** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing** **100088** **China**	Authorized officer **SUN,Yifan**
Facsimile No. **(86-10)62019451**	Telephone No. **86-010-62089961**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2019/118528

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

[1] I: Claims 1-8 directs to a method for video processing..

[2] II: Claims 9-18 directs to a method for video processing..

[3] III: Claims 19-27 directs to a method for video processing..

[4] In these groups of claims, the common or corresponding technical feature is: "performing a video processing on the current block". However, this feature has already been disclosed by the prior art WO 2018067823 A1; and other technical features included in the these groups of claims are not common or corresponding each other. Therefore, the common or corresponding special technical features are not included in these groups of claims. The inventions are not so linked as to form a single general inventive concept, as required by PCT Rule 13.1.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☑ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2019/118528

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
WO	2018067823	A1	12 April 2018	PH	12019500417	A1	03 June 2019
				TW	201817237	A	01 May 2018
				JP	2019535192	A	05 December 2019
				US	2018098063	A1	05 April 2018
				CN	109792533	A	21 May 2019
				MX	2019003762	A	12 August 2019
				US	10448010	B2	15 October 2019
				AU	2017340631	A1	14 March 2019
				CA	3035587	A1	12 April 2018
				SG	11201901632Q	A	29 April 2019
				EP	3523972	A1	14 August 2019
				BR	112019006838	A2	25 June 2019
				KR	20190058503	A	29 May 2019
WO	2017200771	A1	23 November 2017	EP	3459249	A1	27 March 2019
				US	2017332095	A1	16 November 2017
				BR	112018073397	A2	19 March 2019
				KR	20190006967	A	21 January 2019
				CA	3020244	A1	23 November 2017
				TW	201742465	A	01 December 2017
				CN	109155855	A	04 January 2019
				JP	2019519980	A	11 July 2019
WO	2018064495	A1	05 April 2018	WO	2018064492	A1	05 April 2018
				EP	3520407	A1	07 August 2019
				US	2018098062	A1	05 April 2018
				CN	109792527	A	21 May 2019
				JP	2019534622	A	28 November 2019
				US	2018098087	A1	05 April 2018
				BR	112019006580	A2	02 July 2019
				EP	3520406	A1	07 August 2019
				KR	20190055104	A	22 May 2019
				CN	109792526	A	21 May 2019
WO	2018200960	A1	01 November 2018	US	2018316929	A1	01 November 2018
US	2018270500	A1	20 September 2018	SG	11201907090W	A	27 September 2019
				CN	110383839	A	25 October 2019
				KR	20190120389	A	23 October 2019
				AU	2018234607	A1	22 August 2019
				WO	2018169923	A1	20 September 2018