



(51) International Patent Classification:

H04N 19/52 (2014.01) H04N 19/537 (2014.01)
H04N 19/567 (2014.01)

(21) International Application Number:

PCT/CN2020/076927

(22) International Filing Date:

27 February 2020 (27.02.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2019/076300
27 February 2019 (27.02.2019) CN

(71) Applicants: **BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD.** [CN/CN]; Room B-0035, 2/F, No. 3 Building, No. 30, Shixing Road, Shijingshan District, Beijing 100041 (CN). **BYTEDANCE INC.** [US/US];

12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

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

(54) Title: REGRESSION-BASED MOTION VECTOR FIELD BASED SUB-BLOCK MOTION VECTOR DERIVATION

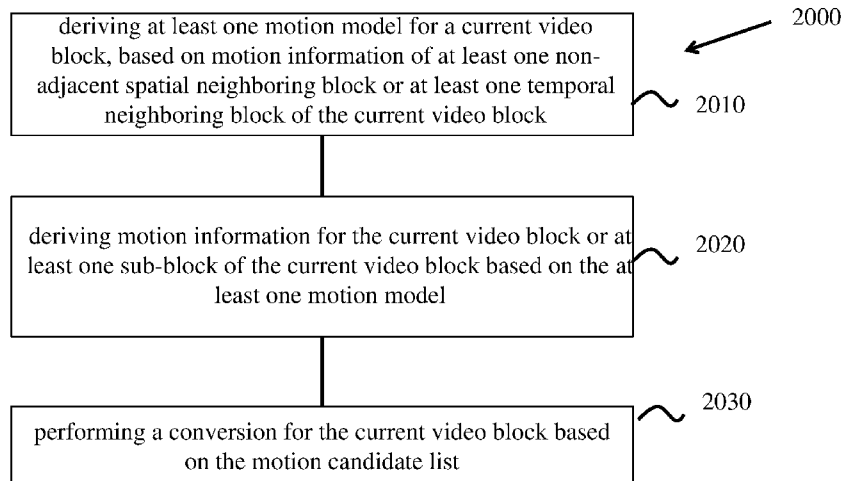


FIG. 20

(57) Abstract: Devices, systems and methods for digital video coding, which includes deriving RMVF-based motion vector information, are described. An exemplary method for video processing includes deriving at least one motion model for a current video block, based on motion information of at least one non-adjacent spatial neighboring block or at least one temporal neighboring block of the current video block; deriving motion information for the current video block or at least one sub-block of the current video block based on the at least one motion model; and performing a conversion for the current video block based on the derived motion information.



CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

REGRESSION-BASED MOTION VECTOR FIELD BASED SUB-BLOCK MOTION VECTOR DERIVATION

TECHNICAL FIELD

[0001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefit of International Patent Applications PCT/CN2019/076300, filed on February 27, 2019. The entire disclosure thereof is incorporated by reference as part of the disclosure of this application.

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

BACKGROUND

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

SUMMARY

[0004] Devices, systems and methods related to digital video coding, and specifically, to deriving motion vectors are described. The described methods may be applied to existing video coding standards (e.g., High Efficiency Video Coding (HEVC) or Versatile Video Coding) and future video coding standards or video codecs.

[0005] In one representative aspect, there is disclosed a method for video processing, comprising: deriving at least one motion model for a current video block, based on motion information of at least one non-adjacent spatial neighboring block or at least one temporal neighboring block of the current video block; deriving motion information for the current video block or at least one sub-block of the current video block based on the at least one motion model; and performing a conversion for the current video block based on the derived motion information.

[0006] In another representative aspect, there is disclosed a method for video processing, comprising: deriving, one or more control point motion vector predictors (CPMVPs) of an affine model for a current video block, from at least one set of neighboring blocks in a regression-based motion vector field (RMVF) scheme; updating a motion candidate list for the current video block based on the one or more CPMVPs, wherein the one or more CPMVPs are associated with a specific affine motion mode; and performing a conversion for the current video block based on the motion candidate list.

[0007] In still another representative aspect, there is disclosed a method for video processing, comprising: deriving at least one set of affine parameters of an affine model for a current video block by utilizing a regression-based motion vector field (RMVF) scheme at least based on at least one set of motion information stored in a lookup table; performing a conversion for the current video block based on the at least one set of affine parameters.

[0008] In yet another representative aspect, there is disclosed a method for video processing, comprising: maintaining for a current video block, an affine motion candidate list which includes a plurality of affine motion candidates, wherein the current video block is coded in affine mode; reordering the plurality of affine motion candidates in the affine motion candidate list based on at least one set of motion information associated with one or more blocks converted previously to update the affine motion candidate list; and performing a conversion for the current video block based on the updated affine motion candidate list..

[0009] Further, in a representative aspect, any one or more of the disclosed methods is an encoder-side implementation.

[00010] Also, in a representative aspect, any one or more of the disclosed methods is a decoder-side implementation.

[00011] One or more of the above-described methods are embodied in the form of processor-executable code and stored in a computer-readable program medium.

[00012] In yet another representative aspect, an apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon is disclosed. The instructions upon

execution by the processor, cause the processor to implement any one or more of the disclosed methods.

[00013] The above and other aspects and features of the disclosed technology are described in greater detail in the drawings, the description and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[00014] FIG. 1 shows an example of constructing a merge candidate list.

[00015] FIG. 2 shows an example of positions of spatial candidates.

[00016] FIG. 3 shows an example of candidate pairs subject to a redundancy check of spatial merge candidates.

[00017] FIGs. 4A and 4B show examples of the position of a second prediction unit (PU) based on the size and shape of the current block.

[00018] FIG. 5 shows an example of motion vector scaling for temporal merge candidates.

[00019] FIG. 6 shows an example of candidate positions for temporal merge candidates.

[00020] FIG. 7 shows an example of generating a combined bi-predictive merge candidate.

[00021] FIG. 8 shows an example of constructing motion vector prediction candidates.

[00022] FIG. 9 shows an example of motion vector scaling for spatial motion vector candidates.

[00023] FIG. 10 shows an example of the coding flow for history-based motion vector prediction (HMVP) candidates.

[00024] FIG. 11 shows an example of updating a table in the HMVP method.

[00025] FIG. 12 shows an example of motion information prediction.

[00026] FIG. 13 shows an example of an affine motion model.

[00027] FIG. 14 shows an example of an affine motion vector field per sub-block.

[00028] FIG. 15 shows example candidate positions for an affine merge mode.

[00029] FIGs. 16A and 16B show a 4-parameter affine model and a 6-parameter affine model respectively.

[00030] FIG. 17 shows an example of neighboring motion vectors used in derivation of RMVF motion information.

[00031] FIG. 18 shows an example of a reduced set of neighboring motion vectors used in derivation of RMVF motion information.

[00032] FIG. 19 is a block diagram of an example of a hardware platform for implementing a visual media decoding or a visual media encoding technique described in the present document.

[00033] FIG. 20 illustrates a flowchart of an example method for video processing.

[00034] FIG. 21 illustrates a flowchart of another example method for video processing.

[00035] FIG. 22 illustrates a flowchart of another example method for video processing.

[00036] FIG. 23 illustrates a flowchart of another example method for video processing.

DETAILED DESCRIPTION

1. Video coding in HEVC/H.265

[00037] Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM). In April 2018, the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC.

2.1. Inter prediction in HEVC/H.265

[00038] 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 signaled using `inter_pred_idc`. Motion vectors may be explicitly coded as deltas relative to predictors.

[00039] 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 neighboring 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 signaled explicitly per each PU. Such a mode is named Advanced motion vector prediction (AMVP) in this patent document.

[00040] When signaling 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.

[00041] When signaling 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.

[00042] The following text provides the details on the inter prediction modes specified in HEVC. The description will start with the merge mode.

2.1.1. Reference picture list

[00043] 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.

[00044] 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. Reference pictures included in List 0/1 can be from past and future pictures in terms of capturing/display order.

2.1.2. Merge Mode

2.1.2.1. Derivation of candidate for merge mode

[00045] 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

[00046] 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 signaled in slice header. Since the number of candidates is constant, index of best merge candidate is encoded using truncated unary binarization (TU). If the size of CU is equal to 8, all the PUs of the current CU share a single merge candidate list, which is identical to the merge candidate list of the $2N \times 2N$ prediction unit.

[00047] In the following, the operations associated with the aforementioned steps are detailed.

2.1.2.2. Spatial candidates derivation

[00048] 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 $2N \times 2N$. As an example, FIGs. 4A and 4B depict examples when the second PU is $N \times 2N$ and $2N \times N$, respectively. When the current PU is partitioned as $N \times 2N$, 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 $2N \times N$.

2.1.2.3. Temporal candidates derivation

[00049] 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 signaled in the slice header. The scaled motion vector for temporal merge candidate is obtained as illustrated by the dotted line in FIG. 5, which is scaled from the motion vector of the co-located PU using the POC distances, t_b and t_d , where t_b is defined to be the POC difference between the reference picture of the current picture and the current picture and t_d is defined to be the POC difference between the reference picture of the co-located picture and the co-located picture. The reference picture index of temporal merge candidate is set equal to zero. A practical realization of the scaling process is described in the HEVC specification. For a B-slice, two motion vectors, one is

for reference picture list 0 and the other is for reference picture list 1, are obtained and combined to make the bi-predictive merge candidate.

[00050] 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 a/k/a LCU, largest coding unit) row, position C_1 is used. Otherwise, position C_0 is used in the derivation of the temporal merge candidate.

2.1.2.4. Additional candidates insertion

[00051] 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.

[00052] Zero motion candidates are inserted to fill the remaining entries in the merge candidates list and therefore hit the MaxNumMergeCand capacity. These candidates have zero spatial displacement and a reference picture index which starts from zero and increases every time a new zero motion candidate is added to the list. Finally, no redundancy check is performed on these candidates.

2.1.3. AMVP

[00053] AMVP exploits spatio-temporal correlation of motion vector with neighboring 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

neighboring 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 signaling, 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.

2.1.3.1. Derivation of AMVP candidates

[00054] FIG. 8 summarizes derivation process for motion vector prediction candidate.

[00055] 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.

[00056] 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.

2.1.3.2. Spatial motion vector candidates

[00057] 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)

[00058] 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 neighboring 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.

[00059] 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.

2.1.3.3. Temporal motion vector candidates

[00060] 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 signaled to the decoder to indicate which reference picture are used.

2.2. Inter prediction methods in VVC

[00061] 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), Advanced TMVP (ATMVP, aka SbTMVP), Generalized Bi-Prediction (GBI), Bi-directional Optical flow (BIO).

[00062] In VVC, a QuadTree/BinaryTree/MultipleTree (QT/BT/TT) structure is adopted to divide a picture into square or rectangle blocks.

[00063] Besides QT/BT/TT, separate tree (a.k.a. Dual coding tree) is also adopted in VVC for I-frames. With separate tree, the coding block structure are signaled separately for the luma and chroma components.

2.2.1. History based motion vector prediction

[00064] A history-based MVP (HMVP) method is proposed wherein a HMVP candidate is defined as the motion information of a previously coded block. A table with multiple HMVP candidates is maintained during the encoding/decoding process. The table is emptied when a new slice is encountered. 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. 10.

[00065] In one example, the table size is set to be L (e.g., $L = 16$ or 6 , or 44), which indicates up to L HMVP candidates may be added to the table.

- 1) In one embodiment, if there are more than L HMVP candidates from the previously coded blocks, a First-In-First-Out (FIFO) rule is applied so that the table always contains the latest previously coded L motion candidates. FIG. 11 depicts an example wherein the FIFO rule is applied to remove a HMVP candidate and add a new one to the table used in the proposed method.
- 2) In another embodiment, whenever adding a new motion candidate (such as the current block is inter-coded and non-affine mode), a redundancy checking process is applied firstly to identify whether there are identical or similar motion candidates in LUTs.

2.2.2. Sub-CU based motion vector prediction

[00066] 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.

[00067] To preserve more accurate motion field for sub-CU motion prediction, the motion compression for the reference frames is currently disabled.

2.2.2.1. Alternative temporal motion vector prediction

[00068] 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 shown in FIG. 12, the sub-CUs are square $N \times N$ blocks (N is set to 4 by default).

[00069] 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 shown in FIG. 12.

[00070] 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.

[00071] 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 $N \times N$ 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.

2.2.3. Affine motion compensation prediction

[00072] 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. A simplified affine transform motion compensation prediction is applied. As shown FIG. 13, the affine motion field of the block is described by two control point motion vectors.

[00073] The motion vector field (MVF) of a block is described by the following equation:

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

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

[00074] To derive motion vector of each $M \times N$ sub-block, the motion vector of the center sample of each sub-block, as shown in FIG. 14, is calculated according to Equation 1, and rounded to 1/16 fraction accuracy.

[00075] There are two affine motion modes: AF_INTER mode and sub-block merge mode.

2.2.3.1. AF_INTER mode

[00076] 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. In AF_INTER mode, control point motion vector predictor (CPMVP) is derived in affine AMVP list. The coordinates of the 3 control points CP1, CP2 and CP3 is (0, 0), (W, 0) and (H, 0) respectively, where W and H are the width and height of current block.

[00077] Affine AMVP list is constructed with following steps:

[00078] 1) Insert inherited affine AMVP candidates

[00079] Inherited affine candidate means that the candidate is derived from the affine motion model of its valid neighbor affine coded block. At most 2 candidates may be derived in this step. As shown in FIG. 15, A0, A1 are checked to generate the first candidate, and B0, B1, B2 are checked to generate the second candidate.

[00080] For each available candidate, its affine parameter is used to derive CPMVP, and the CPMVP is inserted into the affine AMVP list.

[00081] 2) Insert constructed affine AMVP candidates.

[00082] If the number of candidates in affine AMVP list is less than the maximum affine AMVP list size (denoted by MaxAffineAmvpListSize), constructed affine candidate is inserted into the candidate list. Constructed affine candidate means the CPMVP is constructed by combining the neighbor motion information of each control point.

[00083] The motion information for the control points is derived from the specified spatial neighbors shown in FIG. 15. CP_k (k=1, 2, 3) represents the k-th control point. A0, A1, A2, B0, B1, B2 and B3 are spatial positions for predicting CP_k (k=1, 2, 3).

[00084] The MVP of each control point is obtained according to the following priority order:

- 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.
- For CP2, the checking priority is B1->B0.
- For CP3, the checking priority is A1->A0.

[00085] If MVP of CP1 and CP2 are available for 4-parameter affine model, or MVP of CP1, CP2 and CP3 are available for 6-parameter affine model, the CPMVP is inserted into the affine AMVP list.

[00086] If the number of candidates in affine AMVP list is still less than MaxAffineAmvpListSize, same MVP may be assigned to all control points as follows until size of the affine AMVP list is equal to MaxAffineAmvpListSize:

- a) If MVP of CP2 is available, it is set as MVPs of CP1 and CP0, and such CPMVP is inserted into the affine AMVP list.
- b) If MVP of CP1 is available, it is set as MVPs of CP0 and CP2, and such CPMVP is inserted into the affine AMVP list.
- c) If MVP of CP0 is available, it is set as MVPs of CP1 and CP2, and such CPMVP is inserted into the affine AMVP list.

[00087] 3) Padding with zero motion vectors

[00088] If the number of candidates in affine merge candidate list is less than MaxAffineAmvpListSize, zero motion vectors with zero reference indices are used as the MVPs of all control points, until the list is full.

[00089] After CPMVP of the current affine CU are determined, affine motion estimation is applied and the control point motion vector (CPMV) is found. Then the difference of the CPMVs and the CPMVP is signalled in the bitstream.

[00090] 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. 16A. In an example, it is proposed to derive the MV as follows, i.e., mvd_1 and mvd_2 are predicted from mvd_0 .

$$mv_0 = \overline{mv}_0 + mvd_0$$

$$mv_1 = \overline{mv}_1 + mvd_1 + mvd_0$$

$$mv_2 = \overline{mv}_2 + mvd_2 + mvd_0$$

[00091] Wherein \overline{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. 16B. 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.

2.2.3.2. Fast affine ME algorithm in AF_INTER mode

[00092] In affine mode, MV of 2 or 3 control points needs to be determined jointly. Directly searching the multiple MVs jointly is computationally complex. A fast affine ME algorithm is proposed and is adopted into VTM/BMS.

[00093] The fast affine ME algorithm is described for the 4-parameter affine model, and the idea can be extended to 6-parameter affine model.

$$\begin{cases} x' = ax + by + c \\ y' = -bx + ay + d \end{cases} \quad (2)$$

$$\begin{cases} mv_{(x,y)}^h = x' - x = (a - 1)x + by + c \\ mv_{(x,y)}^v = y' - y = -bx + (a - 1)y + d \end{cases} \quad (3)$$

[00094] Replace $(a - 1)$ with a' , then the motion vector can be rewritten as:

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

[00095] Suppose motion vectors of the two controls points $(0, 0)$ and $(0, w)$ are known, from Equation (5), the affine parameters can be derived as follows:

$$\begin{cases} c = mv_{(0,0)}^h \\ d = mv_{(0,0)}^v \end{cases} \quad (5)$$

[00096] The motion vectors can be rewritten in vector form as:

$$MV(p) = A(P) * MV_C^T \quad (6)$$

Wherein

$$A(P) = \begin{bmatrix} 1 & x & 0 & y \\ 0 & y & 1 & -x \end{bmatrix} \quad (7)$$

$$MV_C = [mv_{(0,0)}^h \quad a \quad mv_{(0,0)}^v \quad b] \quad (8)$$

$P = (x, y)$ is the pixel position.

[00097] At encoder, MVD of AF_INTER are derived iteratively. Denote $MV^i(P)$ as the MV derived in the i th iteration for position P and denote dMV_C^i as the delta updated for MV_C in the i th iteration. Then in the $(i+1)$ th iteration,

$$\begin{aligned} MV^{i+1}(P) &= A(P) * \left((MV_C^i)^T + (dMV_C^i)^T \right) \\ &= A(P) * (MV_C^i)^T + A(P) * (dMV_C^i)^T \\ &= MV^i(P) + A(P) * (dMV_C^i)^T \end{aligned} \quad (9)$$

[00098] Denote Pic_{ref} as the reference picture and denote Pic_{cur} as the current picture and denote $Q = P + MV^i(P)$. Suppose MSE is used as the matching criterion, then the following expression can be written:

$$\begin{aligned} &\min \sum_P \left(Pic_{cur}(P) - Pic_{ref}(P + MV^{i+1}(P)) \right)^2 \\ &= \min \sum_P \left(Pic_{cur}(P) - Pic_{ref}(Q + A(P) * (dMV_C^i)^T) \right)^2 \end{aligned} \quad (10)$$

Suppose $(dMV_C^i)^T$ is small enough, $Pic_{ref}(Q + A(P) * (dMV_C^i)^T)$ can be approximately rewritten as follows with 1th order Taylor expansion:

$$Pic_{ref}(Q + A(P) * (dMV_C^i)^T) \approx Pic_{ref}(Q) + Pic_{ref}'(Q) * A(P) * (dMV_C^i)^T \quad (11)$$

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

$$\begin{aligned} &\min \sum_P \left(Pic_{cur}(P) - Pic_{ref}(Q) - Pic_{ref}'(Q) * A(P) * (dMV_C^i)^T \right)^2 \\ &\min \sum_P \left(E^{i+1}(P) - Pic_{ref}'(Q) * A(P) * (dMV_C^i)^T \right)^2. \end{aligned} \quad (12)$$

dMV_C^i can be derived by setting the derivative of the error function to zero. Then, delta MV of the control points $(0, 0)$ and $(0, w)$ can be calculated according to $A(P) * (dMV_C^i)^T$,

$$dMV_{(0,0)}^h = dMV_C^i[0] \quad (13)$$

$$dMV_{(0,w)}^h = dMV_C^i[1] * w + dMV_C^i[2] \quad (14)$$

$$dMV_{(0,0)}^v = dMV_C^i[2] \quad (15)$$

$$dMV_{(0,w)}^v = -dMV_C^i[3] * w + dMV_C^i[2] \quad (16)$$

[00099] Suppose such MVD derivation process is iterated by n times, then the final MVD is calculated as follows,

$$fdMV_{(0,0)}^h = \sum_{i=0}^{n-1} dMV_C^i[0] \quad (17)$$

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

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

$$fdMV_{(0,w)}^v = \sum_{i=0}^{n-1} -dMV_C^i[3] * w + \sum_{i=0}^{n-1} dMV_C^i[2] \quad (20)$$

[000100] In an example, i.e., predicting delta MV of control point (0, w), denoted by mvd_1 from delta MV of control point (0, 0), denoted by mvd_0 , now actually only $(\sum_{i=0}^{n-1} dMV_C^i[1] * w, -\sum_{i=0}^{n-1} -dMV_C^i[3] * w)$ is encoded for mvd_1 .

2.2.3.3. Sub-block merge mode

[000101] Sub-block merge candidate list is constructed with following steps:

- 1) Insert ATMVP candidate.
- 2) Insert inherited affine merge candidates

[000102] Inherited affine merge candidate means that the candidate is derived from the affine motion model of its valid neighbor affine coded block. At most 2 candidates may be derived in this step. As shown in FIG. 15, A0, A1 are checked to generate the first candidate, and B0, B1, B2 are checked to generate the second candidate.

[000103] For each available candidate, its affine parameter is used to derive CPMVP, and the CPMVP is inserted into the sub-block merge list.

- 3) Insert constructed affine merge candidates

[000104] If the number of candidates in sub-block merge candidate list is less than MaxNumAffineCand (set to 5 in this contribution), constructed affine merge candidates are inserted

into the candidate list. Constructed affine merge candidate means the candidate is constructed by combining the neighbor motion information of each control point.

[000105] The motion information for the control points is derived firstly from the specified spatial neighbors and temporal neighbor shown in FIG. 15. CP_k (k=1, 2, 3, 4) represents the k-th control point. A₀, A₁, A₂, B₀, B₁, B₂ and B₃ are spatial positions for predicting CP_k (k=1, 2, 3); T is temporal position for predicting CP₄.

[000106] 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.

[000107] The motion information of each control point is obtained according to the following priority order:

- 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.
- For CP₂, the checking priority is B₁->B₀.
- For CP₃, the checking priority is A₁->A₀.
- For CP₄, T is used.

[000108] Secondly, the combinations of controls points are used to construct an affine merge candidate.

[000109] 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}).

[000110] Motion information of two control points are needed to construct a 4-parameter affine candidate. The two control points can be selected from one of the following two combinations ({CP₁, CP₂}, {CP₁, CP₃}).

[000111] The combinations of constructed affine candidates are inserted into to candidate list as following order:

[000112] {CP1, CP2, CP3}, {CP1, CP2, CP4}, {CP1, CP3, CP4}, {CP2, CP3, CP4}, {CP1, CP2}, {CP1, CP3}.

[000113] For reference list X (X being 0 or 1) of a combination, if different control points use same reference picture, such combination is considered as “available”, otherwise, it is considered as “unavailable”.

4) Padding with zero motion vectors

[000114] If the number of candidates in sub-block 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.

2.3. Regression-based Motion Vector Field

[000115] Regression-based Motion Vector Field (RMVF) based sub-block motion vector derivation is proposed in an example. This tool attempts to model the motion vectors of each block on a sub-block level based on the spatially neighboring motion vectors.

[000116] FIG. 17 illustrates the neighboring 4×4 motion blocks that are used for motion parameter derivation of the proposed method. As seen, one line and row of immediate neighboring motion vectors in 4×4 sub-block basis (and their center locations) from each side of the block are used in the regression process.

[000117] In order to reduce the number of neighboring motion information for RMVF parameter derivation, the method of FIG. 18 is used in which almost half of the neighboring 4×4 motion blocks are used for motion parameter derivation.

[000118] Five conventional regions (bottom-left, left, top-left, top, top-right) as shown in the figures are used when collecting the motion information for motion parameter derivation. The above-right (with W/2 length) and bottom-left (with H/2 length) reference motion regions are limited to only half of the corresponding width or height of the current block.

[000119] In RMVF mode, motion of the block is defined by a 6-parameter motion model. These parameters a_{xx} , a_{xy} , a_{yx} , a_{yy} , b_x and b_y are calculated by solving a linear regression model in mean square error (MSE) sense. The input to the regression model consists of the center locations (x, y) and motion vectors (mv_x and mv_y) of the available neighboring 4×4 sub-blocks as defined above.

[000120] Motion vector (MV_{X_subPU} , MV_{Y_subPU}) for an 8×8 sub-block with center location at (X_{subPU}, Y_{subPU}) is then calculated as:

$$\begin{bmatrix} MV_{X_subPU} \\ MV_{Y_subPU} \end{bmatrix} = \begin{bmatrix} a_{xx} & a_{xy} \\ a_{yx} & a_{yy} \end{bmatrix} \begin{bmatrix} X_{subPU} \\ Y_{subPU} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \end{bmatrix} \quad (21)$$

[000121] The motion vectors are calculated for 8×8 sub-blocks relative to the center location of each sub-block. Consequently, motion compensation is also applied in 8×8 sub-block accuracy in the RMVF mode.

[000122] For efficient modelling of the motion information (e.g., the motion vector field), RMVF can be applied only in the cases that at least one motion vector from at least three of the candidate regions are available.

[000123] Further details:

- The minimum coding unit size for RMVF mode is 8×8
- Decision for uni- or bi-directional prediction is done based on the picture type (P/B picture) and availability of neighboring motion information for that prediction direction.
- Reference index in both prediction directions is set to 0.
- When calculating the motion parameters, the conventional motion vector scaling used in VTM-3.0 is applied to the motion vectors that have different reference pictures than the current block.
- The sub-block motion compensation is done as in VTM-3.0 (i.e., 8×8 sub-block size).

3. Drawbacks of existing RMVF implementations

[000124] The current RMVF method may have the following problems:

1. It cannot be applied to affine inter mode.
2. It uses neighboring motion vectors for derivation of the affine parameters. For small resolution sequences like WVGA, WQVGA or 1080P sequences or small block sizes, few available MVs are available for a block and RMVF may be inefficient.
3. Only 6-parameter affine parameters are derived.

4. Example techniques and embodiments

[000125] The detailed embodiments described below should be considered as examples to explain general concepts. These embodiments should not be interpreted narrowly way. Furthermore, these embodiments can be combined in any manner.

[000126] The RMVF embodiments described below can represent the technology described in section 2.3, or any variances of the RMVF or any coding tools that rely on a neighboring block's motion information to derive a motion model for coding a current block.

1. RMVF may utilize motion information from non-adjacent blocks.
 - a. In one example, RMVF may utilize coded information from temporal blocks located in reference pictures.
 - b. Alternatively, furthermore, non-adjacent blocks shall be within the same CTU.
 - c. Alternatively, furthermore, non-adjacent blocks shall be within the same CTU row.
 - d. In one example, which neighboring/non-adjacent/temporal blocks to be utilized may be changed from block to block.
 - e. In one example, whether to utilize non-adjacent/temporal blocks in RMVF may depend on the availability of motion information from neighboring blocks.
2. RMVF may be utilized for the inter affine mode wherein some control point predictors may be derived from RMVF.
 - a. In one example, RMVF method may be used to derive CPMVPs and the derived CPMVPs (i.e., RMVF affine AMVP candidates) may be inserted into the exiting affine AMVP list.
 - i. Alternatively, a new affine AMVP list may be constructed according to the derived CPMVPs from RMVF. Alternatively, furthermore, whether to use the exiting affine AMVP list with candidates derived from RMVF or use separate affine AMVP lists may be explicitly or implicitly signaled in block/CU/CTU/tile level.
 - b. In one example, 4-parameter affine parameter may be derived by RMVF and may be used to generate CPMVP for all control points of the block.

- c. In one example, 6-parameter affine parameter may be derived by RMVF and may be used to generate CPMVP for all control points of the block.
 - d. In one example, more than one set of 4-parameter affine parameter or/and 6-parameter affine parameter may be derived by RMVF by using different sets of neighboring adjacent or/and non-adjacent MVs.
 - e. In one example, CPMVPs derived by RMVF may be inserted before the inherited affine AMVP candidates.
 - f. In one example, CPMVPs derived by RMVF may be inserted after certain inherited affine AMVP candidates or all inherited affine candidates.
 - i. Alternatively, furthermore, before the constructed affine AMVP candidates.
 - ii. CPMVPs derived by RMVF may be interleaved with the inherited affine AMVP candidates.
 - g. In one example, CPMVP derived by RMVF may be inserted after certain constructed affine AMVP candidates or all constructed affine AMVP candidates.
 - i. Alternatively, furthermore, before the zero MVs.
 - ii. CPMVPs derived by RMVF may be interleaved with the constructed affine AMVP candidates.
 - h. CPMVPs derived by RMVF may be interleaved with the zero MVs.
3. RMVF method may be used to derive CPMVP and the derived CPMVP (i.e., RMVF affine merge candidates) may be inserted into the sub-block merge list/affine merge list.
- a. In one example, 4-parameter affine parameter may be derived by RMVF. Alternatively, furthermore, those parameters may be used to generate CPMVP for all control points of the block.
 - b. In one example, 6-parameter affine parameter may be derived by RMVF. Alternatively, furthermore, those parameters may be used to generate CPMVP for all control points of the block.
 - c. In one example, more than one set of 4-parameter affine parameter or/and 6-parameter affine parameter may be derived by RMVF by using different sets of neighboring adjacent or/and non-adjacent MVs.

- d. In one example, the prediction direction of the RMVF affine merge candidates may depend on the slice type.
 - i. For example, the prediction direction may be the list 0 for P slice.
 - ii. For example, the prediction direction may be list 0 and list 1 for B slice.
 - e. In one example, reference picture of the RMVF affine merge candidates may be always the first reference picture in each reference list.
 - i. Alternatively, reference picture of the RMVF affine merge candidates may be signaled in VPS/SPS/PPS/tile group header/tile header/slice header etc.
 - f. In one example, CPMVPs derived by RMVF may be inserted before the inherited affine merge candidates.
 - g. In one example, CPMVPs derived by RMVF may be inserted after certain inherited affine merge candidates or all inherited affine merge candidates.
 - i. Alternatively, furthermore, before the constructed affine merge candidates.
 - ii. CPMVPs derived by RMVF may be interleaved with the inherited affine merge candidates.
 - h. In one example, CPMVP derived by RMVF may be inserted after certain constructed affine merge candidates or all constructed affine merge candidates.
 - i. Alternatively, furthermore, before the zero MVs.
 - ii. CPMVPs derived by RMVF may be interleaved with the constructed affine merge candidates.
 - i. CPMVPs derived by RMVF may be interleaved with the zero MVs.
4. Motion information (such as CPMVs from previously coded blocks) and/or affine model parameters stored in HMVP lookup table may be used to derive affine parameters using RMVF method.
- a. Alternatively, an independent MV lookup table (called RMVF lookup table), similar with HMVP lookup table, may be maintained for RMVF.
 - b. In one example, neighboring adjacent or/and non-adjacent MVs or/and MVs stored in HMVP lookup table or/and MVs stored in RMVF lookup table may be used to derive affine parameters using RMVF method.

- c. In one example, only MVs stored in RMVF lookup table may be used to derive affine parameters using RMVF method.
 - d. In one example, only MVs stored in HMVP lookup table may be used to derive affine parameters using RMVF method.
5. Motion information of neighboring or/and non-adjacent blocks, or/and motion information stored in HMVP lookup tables or/and motion information stored in RMVF lookup tables may be used to reorder affine motion candidates in a candidate list, such as the affine merge candidate list.
- a. In one example, for each affine merge candidate, affine parameters of it is used to derive MVs of K ($K \geq 1$) neighboring adjacent or/and non-adjacent positions. The distance between the derived MVs (DMV) and the signaled MVs (SMV) is calculated for these positions, denoted as $dist_i$ for position i ($i \leq K$). A cost value is calculated based on $dist_i$, e.g., $cost = f(dist_1, \dots, dist_K)$. The affine merge candidates may be arranged in ascending order of the cost value.
 - b. In one example, distance between the DMV and the SMV may be defined as sum of the square difference of DMV and SMV as follows:

$$dist(DMV, SMV) = (DMV_x - SMV_x)^2 + (DMV_y - SMV_y)^2.$$
 Wherein Mv_x and Mv_y are the horizontal and vertical component of Mv .
 - c. In one example, distance between the DMV and the SMV may be defined as sum of the absolute difference of DMV and SMV as follows:

$$dist(DMV, SMV) = |DMV_x - SMV_x| + |DMV_y - SMV_y|.$$
 - d. In one example, the cost value is defined as the sum of $dist_i$.
 - e. In one example, fixed neighboring adjacent or/and non-adjacent positions are predefined and used.
 - i. If block in a position is not coded with inter mode, it may be considered as “unavailable” and is not used.
 - ii. In one example, non-adjacent positions may be the positions of the MVs stored in HMVP lookup table or/and RMVF lookup table.
 - f. If the prediction direction or/and reference picture of the affine merge candidate is different from a neighboring block, MV scaling may be performed to generate the

derived MV. The MV scaling process may depend on the POC (picture order count) of reference pictures of the affine merge candidate, and POC of reference pictures of the neighboring block.

6. If a neighboring adjacent or/and non-adjacent MV refer to a different reference picture from the target reference picture, it may be considered as “unavailable” and may be not used in RMVF derivation.
7. RMVF may be enabled/disabled according to the rule on block dimension.
 - a. In one example, when a block size contains less than $M \times H$ samples, e.g., 16 or 32 or 64 luma samples, RMVF is not allowed.
 - b. In one example, when a block size contains more than $M \times H$ samples, e.g., 16 or 32 or 64 luma samples, RMVF is not allowed.
 - c. Alternatively, when minimum size of a block’s width or/and height is smaller than or no larger than X , RMVF is not allowed. In one example, X is set to 8.
 - d. Alternatively, when a block’s width $> th1$ or $\geq th1$ and/or a block’s height $> th2$ or $\geq th2$, RMVF is not allowed. In one example, X is set to 64.
 - i. For example, RMVF is disabled for $M \times M$ (e.g., 128×128) block.
 - ii. For example, RMVF is disabled for $N \times M / M \times N$ block, e.g., wherein $N \geq 64$, $M = 128$.
 - iii. For example, RMVF is disabled for $N \times M / M \times N$ block, e.g., wherein $N \geq 4$, $M = 128$.
 - e. Alternatively, when a block’s width $< th1$ or $\leq th1$ and/or a block’s height $< th2$ or $\leq th2$, RMVF is not allowed. In one example, $th1$ and/or $th2$ is set to 8.
8. Indications to tell whether proposed method is enabled or not may be signaled in PPS/VPS/picture header/slice header/tile group header/CTU groups.
 - a. In one example, if separate tree partition structure (e.g., dual tree) is applied, the indications may be signaled more than once for one picture/slice/tile group/CTU.
 - b. In one example, if separate tree partition structure (e.g., dual tree) is applied, the indications may be signaled separately for different color components.
 - i. Alternatively, the indications may be signaled only for one color component and may be applied for all other color components.

5. Example implementations of the disclosed technology

[000127] FIG. 19 is a block diagram of a video processing apparatus 1900. The apparatus 1900 may be used to implement one or more of the methods described herein. The apparatus 1900 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1900 may include one or more processors 1902, one or more memories 1904 and video processing hardware 1906. The processor(s) 1902 may be configured to implement one or more methods described in the present document. The memory (memories) 1904 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1906 may be used to implement, in hardware circuitry, some techniques described in the present document, and may be partly or completely be a part of the processors 1902 (e.g., graphics processor core GPU or other signal processing circuitry).

[000128] In the present document, the term “video processing” may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream.

[000129] It will be appreciated that the disclosed methods and techniques will benefit video encoder and/or decoder embodiments incorporated within video processing devices such as smartphones, laptops, desktops, and similar devices by allowing the use of the techniques disclosed in the present document.

[000130] FIG. 20 is a flowchart for an example method 2000 of video processing. The method 2000 includes, at 2010, deriving at least one motion model for a current video block, based on motion information of at least one non-adjacent spatial neighboring block or at least one temporal neighboring block of the current video block. The method further includes, at 2020, deriving motion information for the current video block or at least one sub-block of the current video block based on

the at least one motion model; and at 2030, performing a conversion for the current video block based on the derived motion information.

[000131] FIG. 21 is a flowchart for an example method 2100 of video processing. The method 2100 includes, at 2110, deriving, one or more control point motion vector predictors (CPMVPs) of an affine model for a current video block, from at least one set of neighboring blocks in a regression-based motion vector field (RMVF) scheme; at 2120, updating a motion candidate list for the current video block based on the one or more CPMVPs, wherein the one or more CPMVPs are associated with a specific affine motion mode; and at 2130, performing a conversion for the current video block based on the motion candidate list.

[000132] FIG. 22 is a flowchart for an example method 2200 of video processing. The method 2200 includes, at 2210, deriving at least one set of affine parameters of an affine model for a current video block by utilizing a regression-based motion vector field (RMVF) scheme at least based on at least one set of motion information stored in a lookup table; and at 2220, performing a conversion for the current video block based on the at least one set of affine parameters.

[000133] FIG. 23 is a flowchart for an example method 2300 of video processing. The method 2300 includes, at 2310, maintaining for a current video block, an affine motion candidate list which includes a plurality of affine motion candidates, wherein the current video block is coded in affine mode; at 2320, reordering the plurality of affine motion candidates in the affine motion candidate list based on at least one set of motion information associated with one or more blocks converted previously to update the affine motion candidate list; and at 2330, performing a conversion for the current video block based on the updated affine motion candidate list.

[000134] Some embodiments may be described using the following examples.

[000135] In one aspect, there is disclosed a method for video processing, comprising: deriving at least one motion model for a current video block, based on motion information of at least one non-adjacent spatial neighboring block or at least one temporal neighboring block of the current video block; deriving motion information for the current video block or at least one sub-block of the current video block based on the at least one motion model; and performing a conversion for the current video block based on the derived motion information.

- [000136] In an example, the motion model comprises a 4-parameter affine model.
- [000137] In an example, the motion model comprises a 6-parameter affine model.
- [000138] In an example, the motion model is derived using a regression-based vector field (RMVF) scheme.
- [000139] In an example, the at least one temporal neighboring block is located in at least one of reference pictures.
- [000140] In an example, the at least one non-adjacent spatial neighboring block is located within a same coding tree unit(CTU) as the current video block.
- [000141] In an example, the at least one non-adjacent spatial neighboring block is located within a same coding tree unit (CTU) row as the current video block.
- [000142] In an example, the at least one non-adjacent spatial neighboring and the at least one temporal neighboring block are determined based on a characteristic of the current video block.
- [000143] In an example, whether the at least one non-adjacent spatial neighboring block or the at least one temporal neighboring block is used for deriving the at least one motion model depends on the availability of motion information of one or more adjacent spatial neighboring blocks of the current video block.
- [000144] In an example, whether a regression-based vector field (RMVF) scheme is enabled to derive the at least one motion model depends on a block dimension of the current video block.
- [000145] In an example, the RMVF scheme is disabled if the block dimension of the current video block satisfies at least one of the following:
- an amount of samples contained in the current video block is less than a first threshold;
 - a minimum size of a height and width of the current video block is not more than a second threshold;
 - the height of the current video block is not more than a third threshold; and
 - the width of the current video block is not more than a fourth threshold.
- [000146] In an example, the first threshold is equal to one of 16, 32 and 64.

- [000147] In an example, the second threshold is equal to 8.
- [000148] In an example, at least one of the third and fourth thresholds is equal to 8.
- [000149] In an example, the RMVF scheme is disabled if the block dimension of the current video block satisfies at least one of the following:
- an amount of samples contained in the current video block is more than a fifth threshold;
 - a height of the current video block is not less than a sixth threshold; and
 - a width of the current video block is not less than a seventh threshold.
- [000150] In an example, the fifth threshold is equal to one of 16, 32 and 64.
- [000151] In an example, at least one of the sixth and seventh thresholds is equal to 64.
- [000152] In an example, the RMVF scheme is disabled if the current video block has a size of $M \times M$, $M \times N$ or $N \times M$, wherein $M=128$, and $N= 64$ or 4 .
- [000153] In an example, whether a regression-based vector field (RMVF) scheme is enabled to derive the at least one motion model depends on an indication signaled in at least one of a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header and CTU groups.
- [000154] In an example, the indication is signaled more than once for at least one of a picture, slice, tile group and CTU which covers the current video block if separate tree partition structures are applied to the current video block.
- [000155] In an example, the indication is signaled separately for different color components of the current video block if the separate tree partition structures are applied to the different color components.
- [000156] In an example, the at least one non-adjacent spatial neighboring block or the at least one temporal neighboring block exclude any neighboring block whose motion vector refers to a different reference picture from a target reference picture.
- [000157] In another aspect, there is disclosed a method for video processing, comprising: deriving, one or more control point motion vector predictors (CPMVPs) of an affine model for a

current video block, from at least one set of neighboring blocks in a regression-based motion vector field (RMVF) scheme; updating a motion candidate list for the current video block based on the one or more CPMVPs, wherein the one or more CPMVPs are associated with a specific affine motion mode; and performing a conversion for the current video block based on the motion candidate list.

[000158] In an example, the at least one set of neighboring blocks comprise at least one of adjacent blocks and non-adjacent blocks of the current video block.

[000159] In an example, the specific affine motion mode is an affine advanced motion vector prediction (AMVP) mode and the motion candidate list is an affine AMVP list, the one or more CPMVPs being inserted as RMVF-based affine candidates.

[000160] In an example, the affine AMVP list is selected from one of an existing affine AMVP list and a separate affine AMVP list based on an indication signaled in a level of at least one of a block, a coding unit (CU), a coding tree unit (CTU) and a tile, wherein the existing affine AMVP list comprises at least one affine AMVP candidate which is not derived in the RMVF scheme, and the separate affine AMVP list only comprises RMVF-based affine candidates.

[000161] In an example, the specific affine motion mode is an affine merge mode and the motion candidate list is an affine merge list, the one or more CPMVPs being inserted as RMVF-based affine candidates.

[000162] In an example, the affine merger list is a sub-block based affine merge list

[000163] In an example, the motion candidate list comprises one or more inherited affine candidates, and the RMVF-based affine candidates are arranged in the motion candidate list in one of following orders:

before all of the inherited affine candidates;

after at least one of one or more inherited affine candidates; or

interleaved with the one or more inherited affine candidates.

[000164] In an example, the motion candidate list further comprises one or more constructed affine candidates, and the RMVF-based affine candidates are arranged in the motion candidate list in one of following orders:

before all of the constructed affine candidates;

after at least one of one or more constructed affine candidates; or

interleaved with the one or more constructed affine candidates.

[000165] In an example, the motion candidate list further comprises one or more zero motion vectors(MVs), and the RMVF-based affine candidates are arranged before all zero MVs or interleaved with the zero MVs in the motion candidate list.

[000166] In an example, at least one set of affine parameters of the affine model for the current video block is determined from at least one set of neighboring blocks in the RMVF scheme to derive the one or more CPMVPs of the current video block.

[000167] In an example, more than one set of affine parameters of the affine model for the current video block are determined from different sets of neighboring blocks in the RMVF scheme to derive the one or more CPMVPs of the current video block.

[000168] In an example, the affine model is a 6-parameter affine model or a 4-parameter affine model.

[000169] In an example, each of the RMVF-based affine candidates is associated with a prediction direction which depends on a slice type.

[000170] In an example, the prediction direction corresponds to list 0 for a P-type slice.

[000171] In an example, the prediction direction corresponds to list 0 and list 1 for a B-type slice.

[000172] In an example, each of the RMVF-based affine candidates is associated with a reference picture included in a reference list.

[000173] In an example, the reference picture corresponds to a first reference picture in the reference list.

[000174] In an example, the reference picture is determined from an indication signaled in one of a video parameter set (VPS), a sequence parameter set (SPS), a picture parameter set (PPS), a tile group header, a tile header, or a slice header.

[000175] In another aspect, there is disclosed a method for video processing, comprising: deriving at least one set of affine parameters of an affine model for a current video block by utilizing a regression-based motion vector field (RMVF) scheme at least based on at least one set of motion information stored in a lookup table; performing a conversion for the current video block based on the at least one set of affine parameters.

[000176] In an example, the at least one set of affine parameters of the affine model is derived based on the at least one set of motion information stored in the lookup table and at least one set of motion information associated with at least one adjacent or non-adjacent neighboring block.

[000177] In an example, the lookup table comprises a history-based motion vector prediction (HMVP) lookup table or a RMVF lookup table which stores motion information of at least one block converted previously.

[000178] In an example, the HMVP lookup table is different from the RMVF lookup table.

[000179] In an example, information on a position and dimension of the at least block is stored together with the motion information of the at least block in the lookup table.

[000180] In an example, the set of motion information comprises at least one of control point motion vector predictors and affine model parameters.

[000181] In another aspect, there is disclosed a method for video processing, comprising: maintaining for a current video block, an affine motion candidate list which includes a plurality of affine motion candidates, wherein the current video block is coded in affine mode; reordering the plurality of affine motion candidates in the affine motion candidate list based on at least one set of motion information associated with one or more blocks converted previously to update the affine motion candidate list; and performing a conversion for the current video block based on the updated affine motion candidate list.

[000182] In an example, the plurality of affine motion candidates is associated with a 4-parameter affine motion model or a 6-parameter affine motion model.

[000183] In an example, the one or more blocks converted previously are one or more adjacent or non-adjacent neighboring blocks.

[000184] In an example, the at least one set of motion information associated with one or more blocks converted previously is stored in a lookup table.

[000185] In an example, the lookup table comprises a history-based motion vector prediction (HMVP) lookup table or a RMVF lookup table which stores at least one set of motion information of one or more blocks converted previously.

[000186] In an example, the HMVP lookup table is different from the RMVF lookup table.

[000187] In an example, information on position and dimension of one or more blocks converted previously is stored together with the at least one set of motion information in the lookup table.

[000188] In an example, the at least one set of motion information comprises at least one set of motion vectors (MVs), and the plurality of affine motion candidates is reordered based on an ascending order of a function value associated with the at least one set of MVs.

[000189] In an example, the function value is obtained as follows:

deriving, from affine parameters of each of the plurality of affine motion candidates, motion vectors associated with K neighboring blocks, the derived motion vectors being represented as DMV_i , for $i=1$ to K;

determining, a difference $dist_i$ between each of the derived motion vectors DMV_i and a signaled motion vector SMV_i of an associated neighboring block; and

obtaining the function value based on a function $f(dist_1, dist_2, \dots, dist_K)$.

[000190] In an example, the function $f(dist_1, dist_2, \dots, dist_K) = dist_1 + dist_2 + \dots + dist_K$.

[000191] In an example, $dist_i = (DMV_{i_x} - SMV_{i_x})^2 + (DMV_{i_y} - SMV_{i_y})^2$, wherein DMV_{i_x} and DMV_{i_y} represent horizontal and vertical components of DMV_i respectively, and SMV_{i_x} and SMV_{i_y} represent horizontal and vertical components of SMV_i respectively

[000192] In an example, $dist_i(DMV_i, SMV_i) = |DMV_{i_x} - SMV_{i_x}| + |DMV_{i_y} - SMV_{i_y}|$,

wherein DMV_{i_x} and DMV_{i_y} represent horizontal and vertical components of DMV_i respectively, and SMV_{i_x} and SMV_{i_y} represent horizontal and vertical components of SMV_i respectively.

[000193] In an example, the K neighboring blocks are pre-defined and comprise at least one of adjacent neighboring blocks and non-adjacent neighboring blocks.

[000194] In an example, the set of MVs associated with the blocks stored in the lookup table.

[000195] In an example, the method further comprises: performing a scaling on the derived motion vector if the affine merge candidate has a different prediction direction and/or a different reference picture from a neighboring block.

[000196] In an example, the scaling is performed based on a picture order count (POC) of the reference picture of the affine merge candidate and POC of the reference picture of the neighboring block.

[000197] In an example, the conversion includes encoding the current video block into the bitstream representation of a video and decoding the current video block from the bitstream representation of the video.

[000198] In another aspect, there is disclosed an apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method described above.

[000199] In another aspect, there is disclosed a computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method described above.

[000200] 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.

[000201] 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.

[000202] 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).

[000203] 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.

[000204] 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.

[000205] 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.

[000206] 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 at least one motion model for a current video block, based on motion information of at least one non-adjacent spatial neighboring block or at least one temporal neighboring block of the current video block;
deriving motion information for the current video block or at least one sub-block of the current video block based on the at least one motion model; and
performing a conversion for the current video block based on the derived motion information.
2. The method of claim 1, wherein the motion model comprises a 4-parameter affine model.
3. The method of claim 1, wherein the motion model comprises a 6-parameter affine model.
4. The method of any one of claims 1-3, wherein the motion model is derived using a regression-based vector field (RMVF) scheme.
5. The method of any one of claims 1-4, wherein the at least one temporal neighboring block is located in at least one of reference pictures.
6. The method of any one of claims 1-4, wherein the at least one non-adjacent spatial neighboring block is located within a same coding tree unit(CTU) as the current video block.
7. The method of any one of claims 1-4, wherein the at least one non-adjacent spatial neighboring block is located within a same coding tree unit(CTU) row as the current video block.

8. The method of any one of claims 1-7, wherein the at least one non-adjacent spatial neighboring and the at least one temporal neighboring block are determined based on a characteristic of the current video block.

9. The method of any one of claims 1-8, wherein whether the at least one non-adjacent spatial neighboring block or the at least one temporal neighboring block is used for deriving the at least one motion model depends on the availability of motion information of one or more adjacent spatial neighboring blocks of the current video block.

10. The method of any one of claims 1-9, wherein whether a regression-based vector field (RMVF) scheme is enabled to derive the at least one motion model depends on a block dimension of the current video block.

11. The method of claim 10, wherein the RMVF scheme is disabled if the block dimension of the current video block satisfies at least one of the following:

- an amount of samples contained in the current video block is less than a first threshold;
- a minimum size of a height and width of the current video block is not more than a second threshold;
- the height of the current video block is not more than a third threshold; and
- the width of the current video block is not more than a fourth threshold.

12. The method of claim 11, wherein the first threshold is equal to one of 16, 32 and 64.

13. The method of claim 11, wherein the second threshold is equal to 8.

14. The method of claim 11, wherein at least one of the third and fourth thresholds is equal to 8.

15. The method of claim 10, wherein the RMVF scheme is disabled if the block dimension of the current video block satisfies at least one of the following:
an amount of samples contained in the current video block is more than a fifth threshold;
a height of the current video block is not less than a sixth threshold; and
a width of the current video block is not less than a seventh threshold.
16. The method of claim 15, wherein the fifth threshold is equal to one of 16, 32 and 64.
17. The method of claim 16, wherein at least one of the sixth and seventh thresholds is equal to 64.
18. The method of claim 17, wherein the RMVF scheme is disabled if the current video block has a size of $M \times M$, $M \times N$ or $N \times M$, wherein $M=128$, and $N=64$ or 4.
19. The method of any one of claims 1-18, wherein whether a regression-based vector field (RMVF) scheme is enabled to derive the at least one motion model depends on an indication signaled in at least one of a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header and CTU groups.
20. The method of claim 19, wherein the indication is signaled more than once for at least one of a picture, slice, tile group and CTU which covers the current video block if separate tree partition structures are applied to the current video block.
21. The method of claim 20, wherein the indication is signaled separately for different color components of the current video block if the separate tree partition structures are applied to the different color components.
22. The method of any one of claims 1-21, wherein the at least one non-adjacent spatial neighboring block or the at least one temporal neighboring block exclude any neighboring block whose motion vector refers to a different reference picture from a target reference picture.

23. A method for video processing, comprising:
deriving, one or more control point motion vector predictors (CPMVPs) of an affine model for a current video block, from at least one set of neighboring blocks in a regression-based motion vector field (RMVF) scheme;
updating a motion candidate list for the current video block based on the one or more CPMVPs, wherein the one or more CPMVPs are associated with a specific affine motion mode; and
performing a conversion for the current video block based on the motion candidate list.

24. The method of claim 23, wherein the at least one set of neighboring blocks comprise at least one of adjacent blocks and non-adjacent blocks of the current video block.

25. The method of claim 23 or 24, wherein the specific affine motion mode is an affine advanced motion vector prediction (AMVP) mode and the motion candidate list is an affine AMVP list, the one or more CPMVPs being inserted as RMVF-based affine candidates.

26. The method of claim 25, wherein the affine AMVP list is selected from one of an existing affine AMVP list and a separate affine AMVP list based on an indication signaled in a level of at least one of a block, a coding unit (CU), a coding tree unit (CTU) and a tile, wherein the existing affine AMVP list comprises at least one affine AMVP candidate which is not derived in the RMVF scheme, and the separate affine AMVP list only comprises RMVF-based affine candidates.

27. The method of claim 23 or 24, wherein the specific affine motion mode is an affine merge mode and the motion candidate list is an affine merge list, the one or more CPMVPs being inserted as RMVF-based affine candidates.

28. The method of claim 27, wherein the affine merger list is a sub-block based affine merge list.

29. The method of any one of claims 25-28, wherein the motion candidate list comprises one or more inherited affine candidates, and the RMVF-based affine candidates are arranged in the motion candidate list in one of following orders:

- before all of the inherited affine candidates;
- after at least one of one or more inherited affine candidates; or
- interleaved with the one or more inherited affine candidates.

30. The method of claim 29, wherein the motion candidate list further comprises one or more constructed affine candidates, and the RMVF-based affine candidates are arranged in the motion candidate list in one of following orders:

- before all of the constructed affine candidates;
- after at least one of one or more constructed affine candidates; or
- interleaved with the one or more constructed affine candidates.

31. The method of claim 29 or 30, wherein the motion candidate list further comprises one or more zero motion vectors(MVs), and the RMVF-based affine candidates are arranged before all zero MVs or interleaved with the zero MVs in the motion candidate list.

32. The method of any one of claims 23-31, wherein at least one set of affine parameters of the affine model for the current video block is determined from at least one set of neighboring blocks in the RMVF scheme to derive the one or more CPMVPs of the current video block.

33. The method of claim 32, wherein more than one set of affine parameters of the affine model for the current video block are determined from different sets of neighboring blocks in the RMVF scheme to derive the one or more CPMVPs of the current video block.

34. The method of any one of claims 23-33, wherein the affine model is a 6-parameter affine model or a 4-parameter affine model.

35. The method of claim 27 or 28, wherein each of the RMVF-based affine candidates is associated with a prediction direction which depends on a slice type.

36. The method of claim 35, wherein the prediction direction corresponds to list 0 for a P-type slice.

37. The method of claim 35, wherein the prediction direction corresponds to list 0 and list 1 for a B-type slice.

38. The method of any one of claims 27-28 and 35-37, wherein each of the RMVF-based affine candidates is associated with a reference picture included in a reference list.

39. The method of claim 38, wherein the reference picture corresponds to a first reference picture in the reference list.

40. The method of claim 38 or 39, wherein the reference picture is determined from an indication signaled in one of a video parameter set (VPS), a sequence parameter set (SPS), a picture parameter set (PPS), a tile group header, a tile header, or a slice header.

41. A method for video processing, comprising:
deriving at least one set of affine parameters of an affine model for a current video block by utilizing a regression-based motion vector field (RMVF) scheme at least based on at least one set of motion information stored in a lookup table;
performing a conversion for the current video block based on the at least one set of affine parameters.

42. The method of claim 41, wherein the at least one set of affine parameters of the affine model is derived based on the at least one set of motion information stored in the lookup table and at least one set of motion information associated with at least one adjacent or non-adjacent neighboring block.

43. The method of any one of claims 41-42, wherein the lookup table comprises a history-based motion vector prediction (HMVP) lookup table or a RMVF lookup table which stores motion information of at least one block converted previously.

44. The method of claim 43, wherein the HMVP lookup table is different from the RMVF lookup table.

45. The method of claim 43 or 44, wherein information on a position and dimension of the at least block is stored together with the motion information of the at least block in the lookup table.

46. The method of any one of claims 41-45, wherein the set of motion information comprises at least one of control point motion vector predictors and affine model parameters.

47. A method for video processing, comprising:
maintaining for a current video block, an affine motion candidate list which includes a plurality of affine motion candidates, wherein the current video block is coded in affine mode;
reordering the plurality of affine motion candidates in the affine motion candidate list based on at least one set of motion information associated with one or more blocks converted previously to update the affine motion candidate list; and
performing a conversion for the current video block based on the updated affine motion candidate list.

48. The method claim 47, wherein the plurality of affine motion candidates is associated with a 4-parameter affine motion model or a 6-parameter affine motion model.

49. The method of claim 47 or 48, wherein the one or more blocks converted previously are one or more adjacent or non-adjacent neighboring blocks.

50. The method of claim 47 or 48, wherein the at least one set of motion information associated with one or more blocks converted previously is stored in a lookup table.

51. The method of claim 50, wherein the lookup table comprises a history-based motion vector prediction (HMVP) lookup table or a RMVF lookup table which stores at least one set of motion information of one or more blocks converted previously.

52. The method of claim 51, wherein the HMVP lookup table is different from the RMVF lookup table.

53. The method of any one of claims 50-52, wherein information on position and dimension of one or more blocks converted previously is stored together with the at least one set of motion information in the lookup table.

54. The method of any one of claims 47-53, wherein the at least one set of motion information comprises at least one set of motion vectors (MVs), and the plurality of affine motion candidates is reordered based on an ascending order of a function value associated with the at least one set of MVs.

55. The method of claim 54, wherein the function value is obtained as follows:

deriving, from affine parameters of each of the plurality of affine motion candidates, motion vectors associated with K neighboring blocks, the derived motion vectors being represented as DMV_i , for $i=1$ to K;

determining, a difference $dist_i$ between each of the derived motion vectors DMV_i and a signaled motion vector SMV_i of an associated neighboring block; and

obtaining the function value based on a function $f(dist_1, dist_2, \dots, dist_K)$.

56. The method claim 55, wherein the function $f(dist_1, dist_2, \dots, dist_K) = dist_1 + dist_2 + \dots + dist_K$.

57. The method of claim 55 or 56, wherein

$dist_i = (DMV_{i_x} - SMV_{i_x})^2 + (DMV_{i_y} - SMV_{i_y})^2$, wherein DMV_{i_x} and DMV_{i_y} represent horizontal and vertical components of DMV_i respectively, and SMV_{i_x} and SMV_{i_y} represent horizontal and vertical components of SMV_i respectively.

58. The method of claim 55 or 56, wherein

$$dist_i(DMV_i, SMV_i) = |DMV_{i_x} - SMV_{i_x}| + |DMV_{i_y} - SMV_{i_y}|,$$

wherein DMV_{i_x} and DMV_{i_y} represent horizontal and vertical components of DMV_i respectively, and SMV_{i_x} and SMV_{i_y} represent horizontal and vertical components of SMV_i respectively.

59. The method of any one of claims 55-58, wherein the K neighboring blocks are pre-defined and comprise at least one of adjacent neighboring blocks and non-adjacent neighboring blocks.

60. The method of claim 59, wherein the set of MVs associated with the blocks stored in the lookup table.

61. The method of any one of claims 55-60, further comprising:

performing a scaling on the derived motion vector if the affine merge candidate has a different prediction direction and/or a different reference picture from a neighboring block.

62. The method of claim 61, wherein the scaling is performed based on a picture order count (POC) of the reference picture of the affine merge candidate and POC of the reference picture of the neighboring block.

63. The method of any of claims 1 to 62, wherein the conversion includes encoding the current video block into the bitstream representation of a video and decoding the current video block from the bitstream representation of the video.

64. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of claims 1 to 63.

65. 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 63.

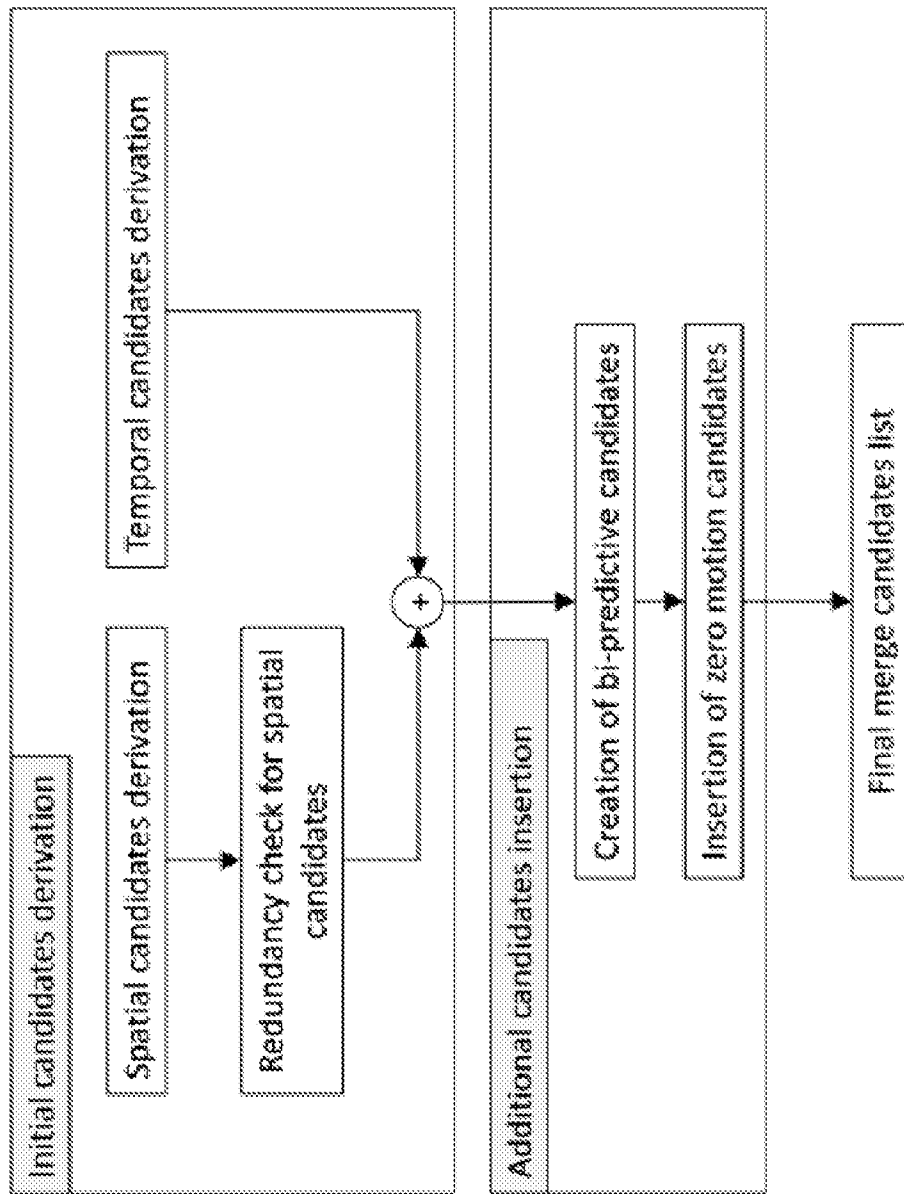


FIG. 1

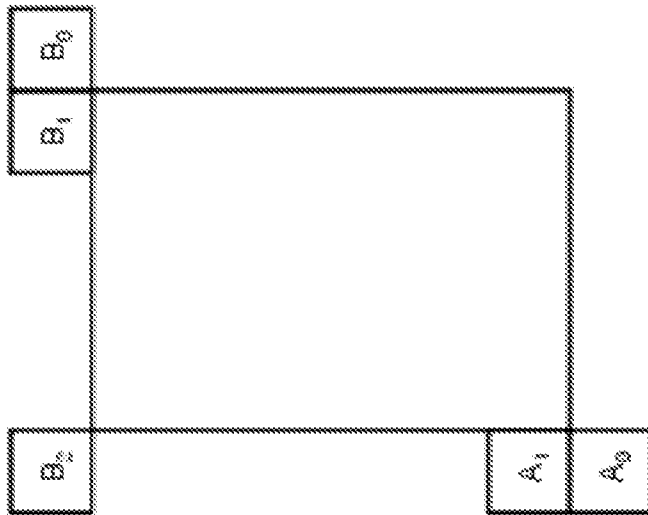


FIG. 2

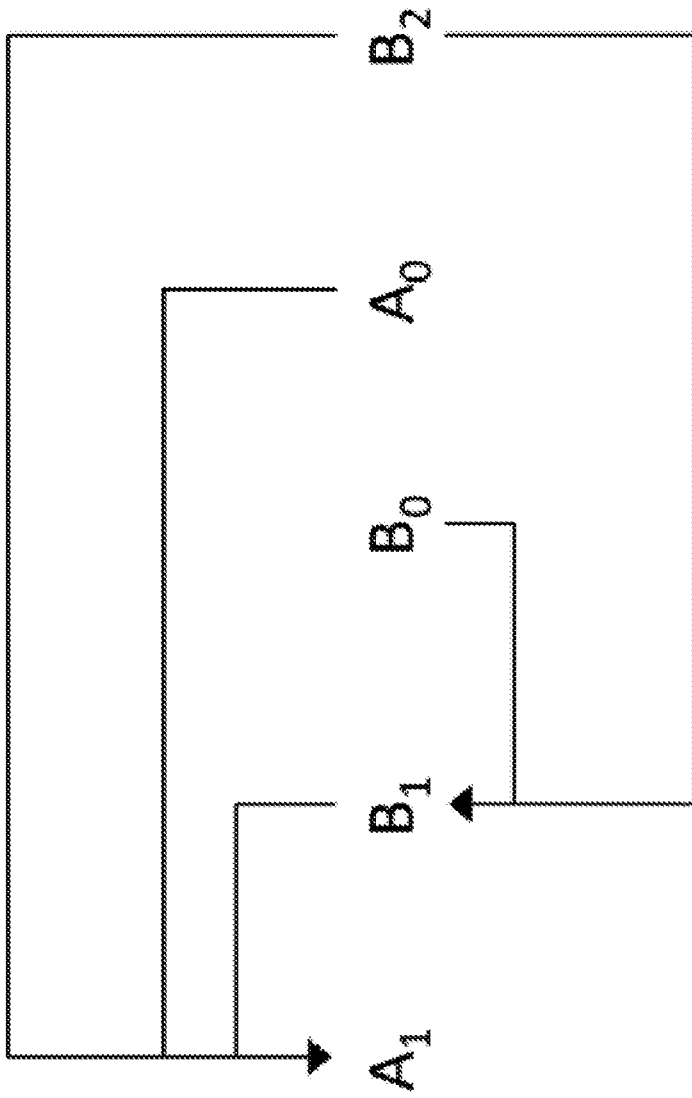
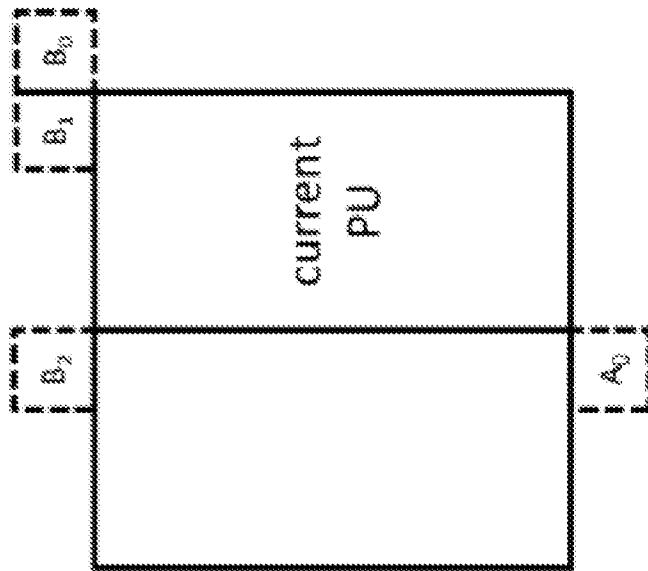
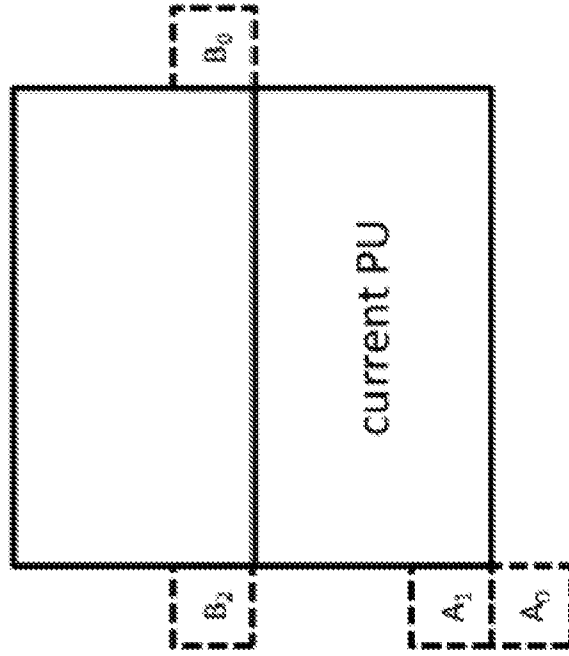


FIG. 3



second PU of $N \times 2N$

FIG. 4A



second PU of $2N \times N$

FIG. 4B

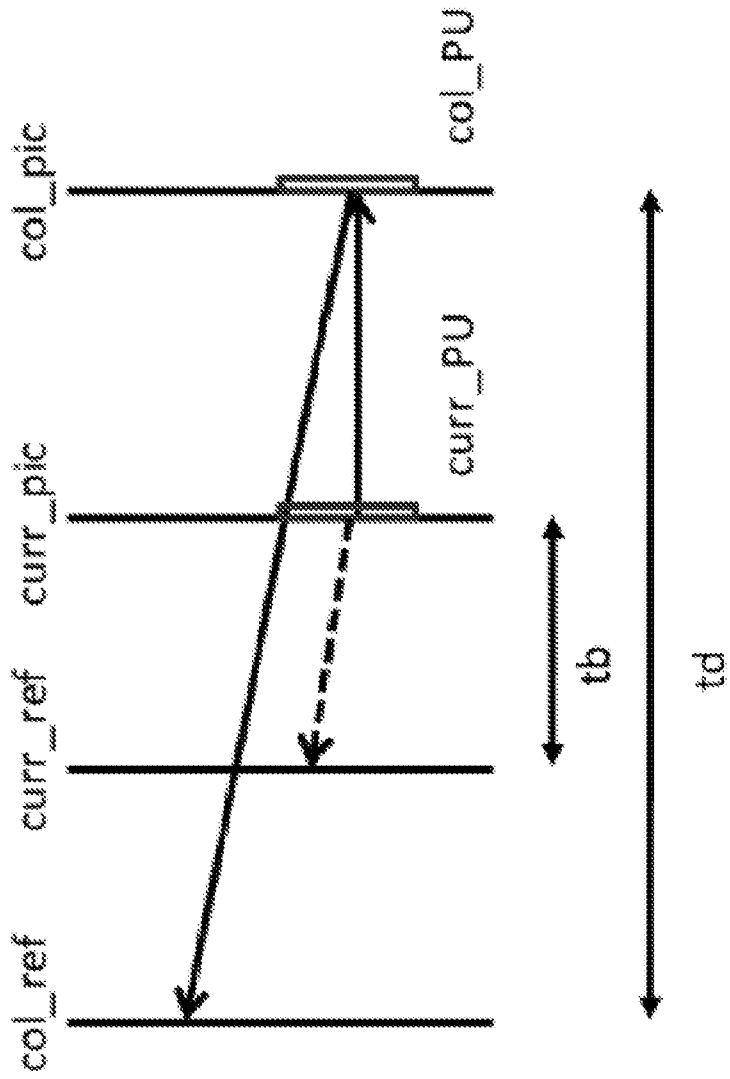


FIG. 5

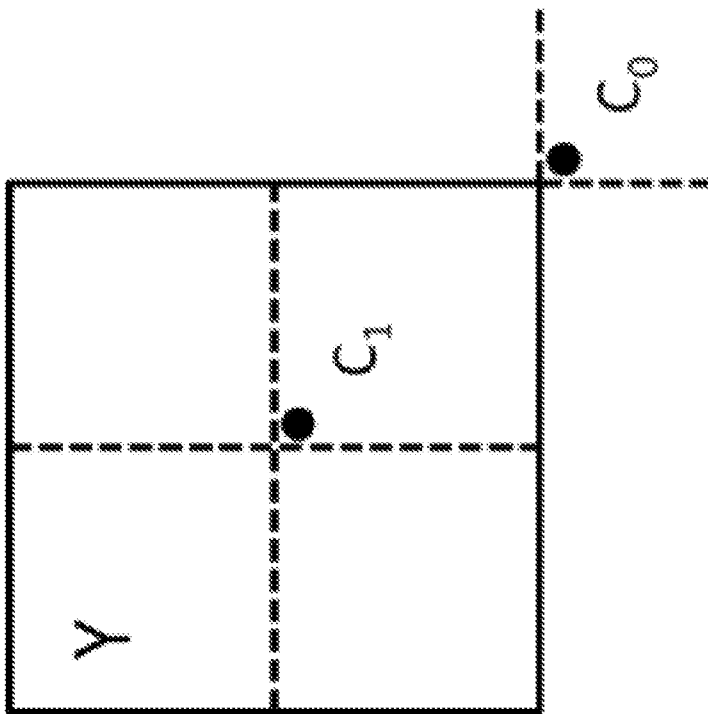


FIG. 6

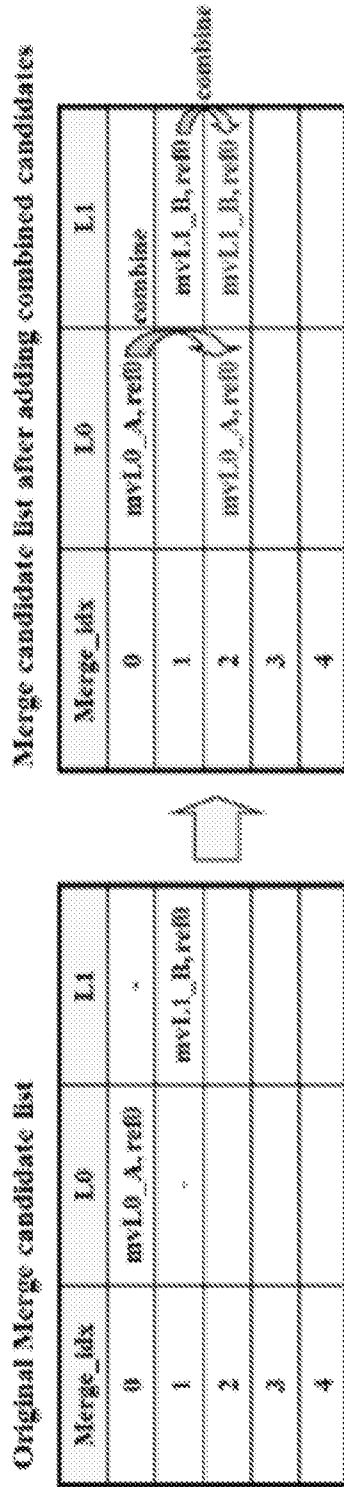


FIG. 7

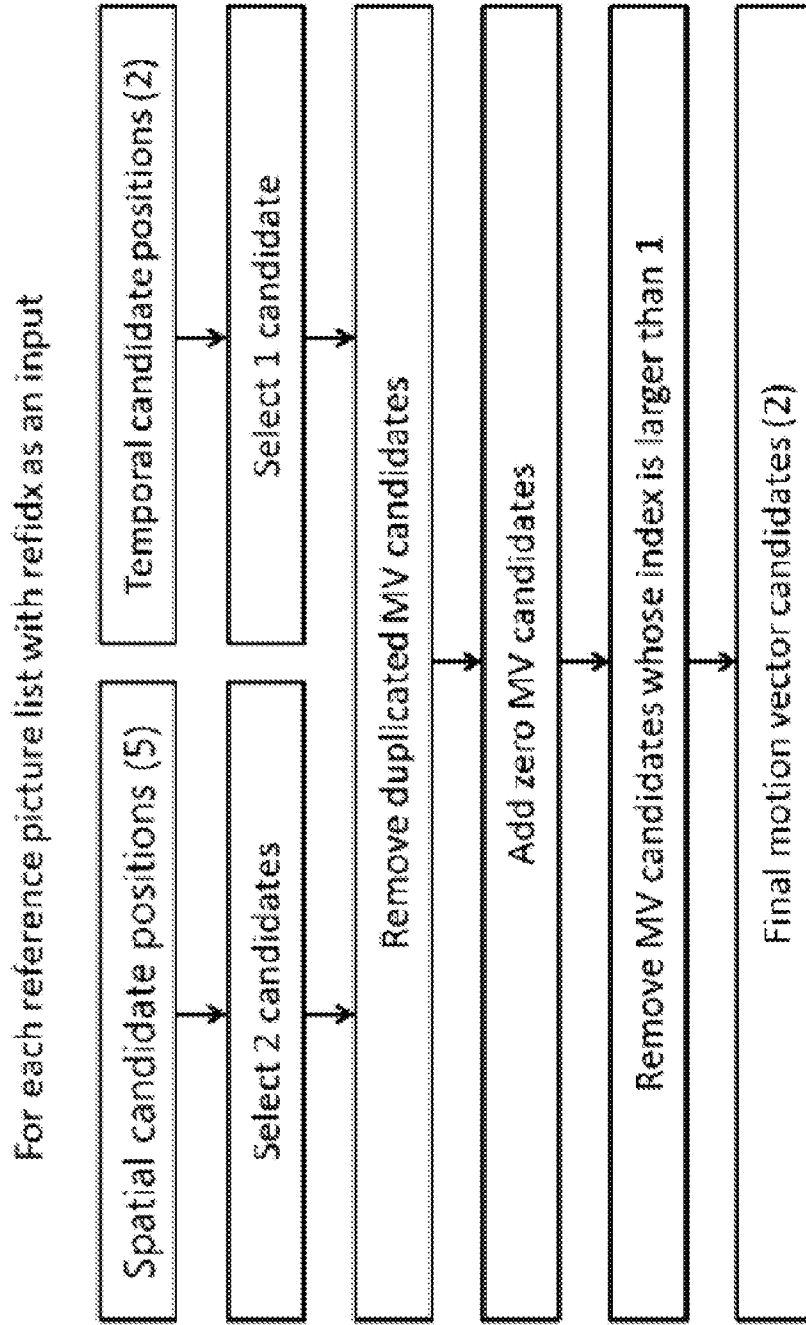


FIG. 8

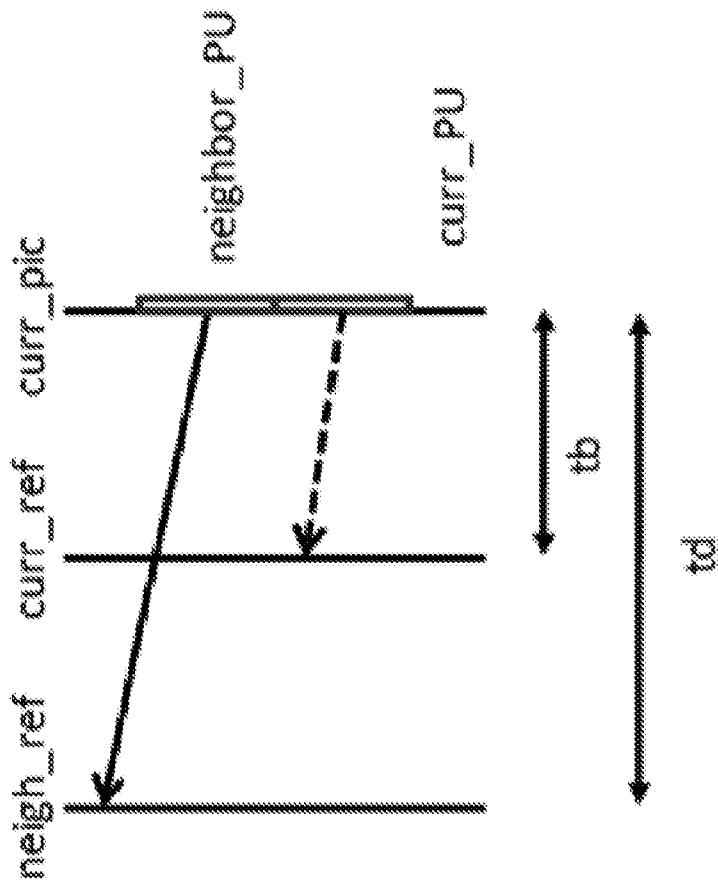


FIG. 9

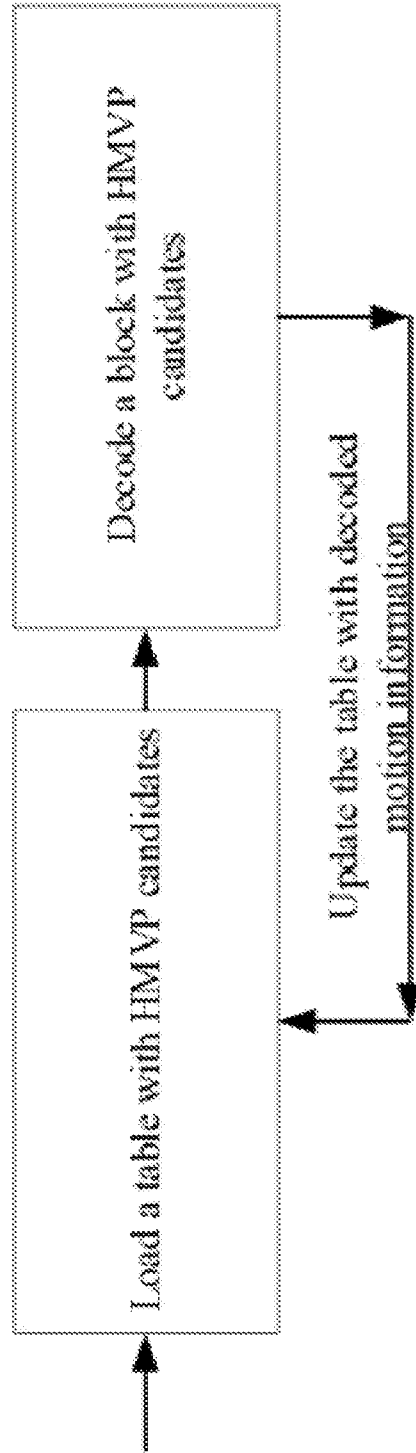


FIG. 10

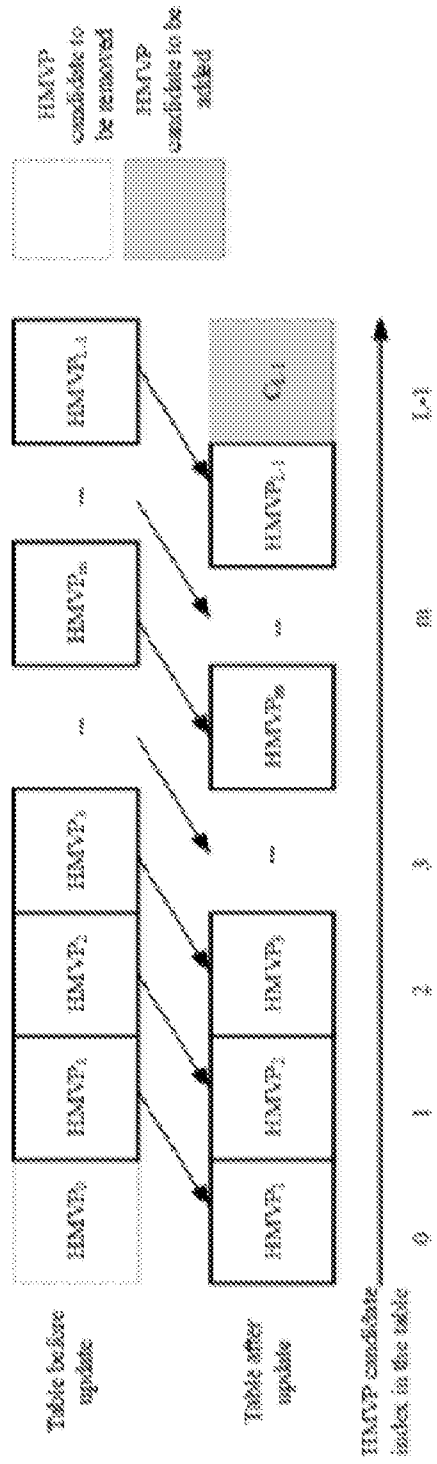


FIG. 11

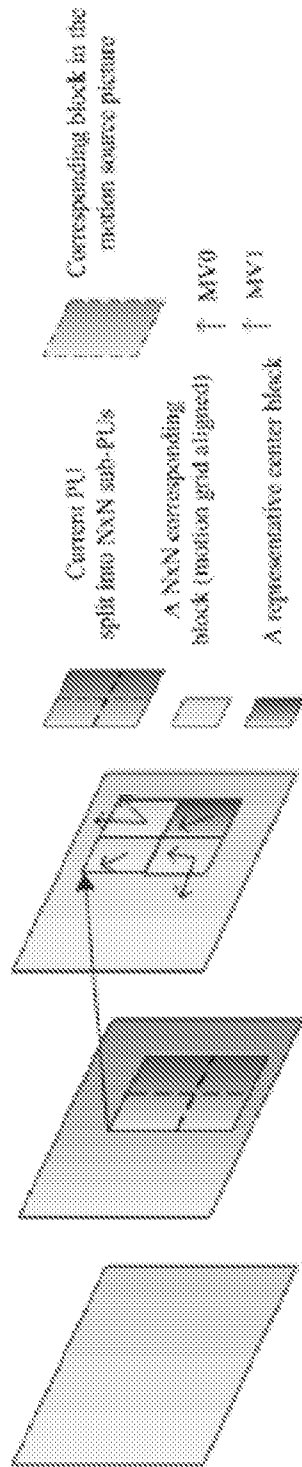


FIG. 12

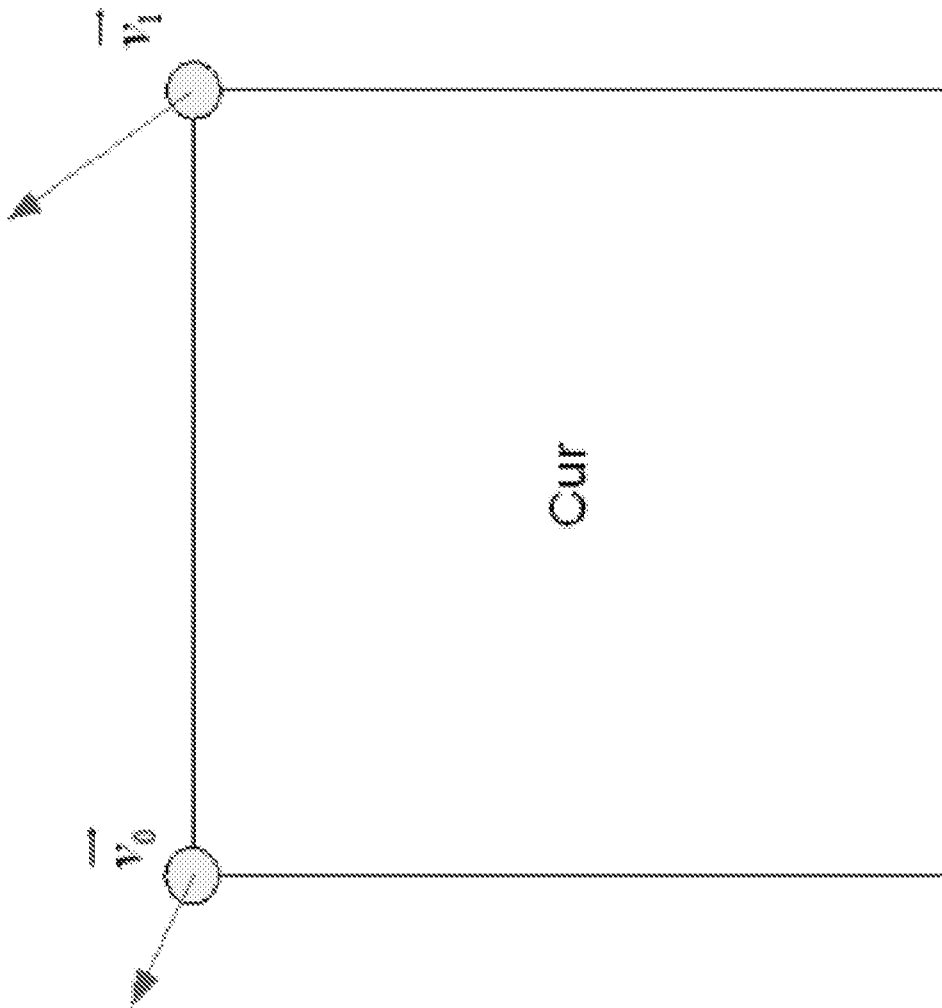


FIG. 13

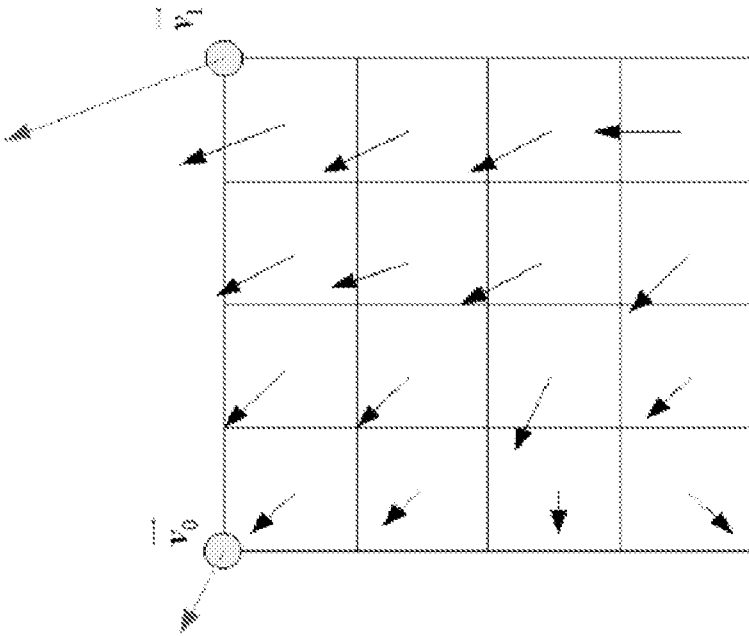


FIG. 14

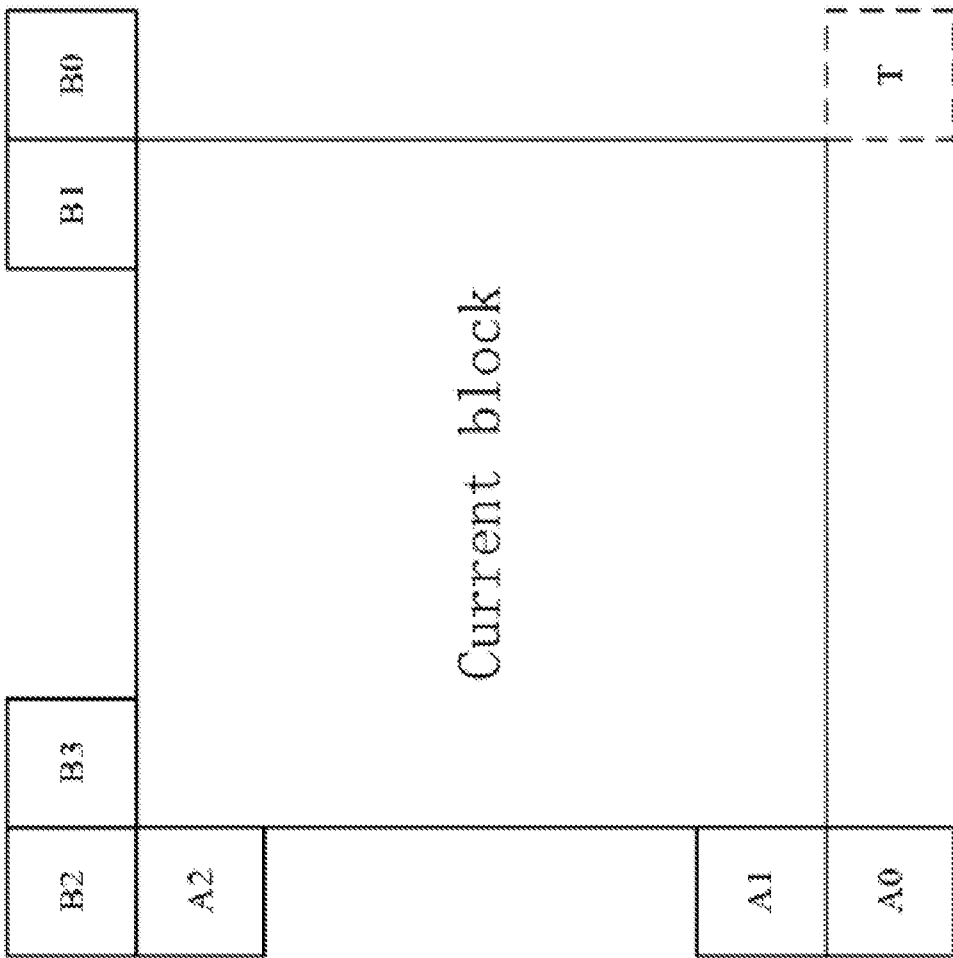
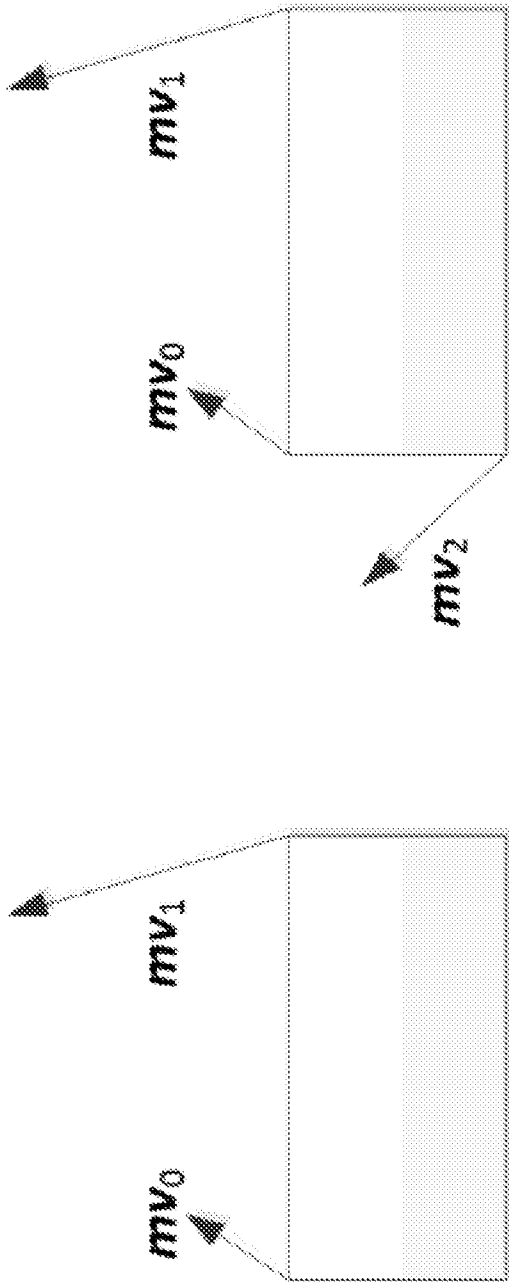


FIG. 15



6-parameter affine model

4-parameter affine model

FIG. 16B

FIG. 16A

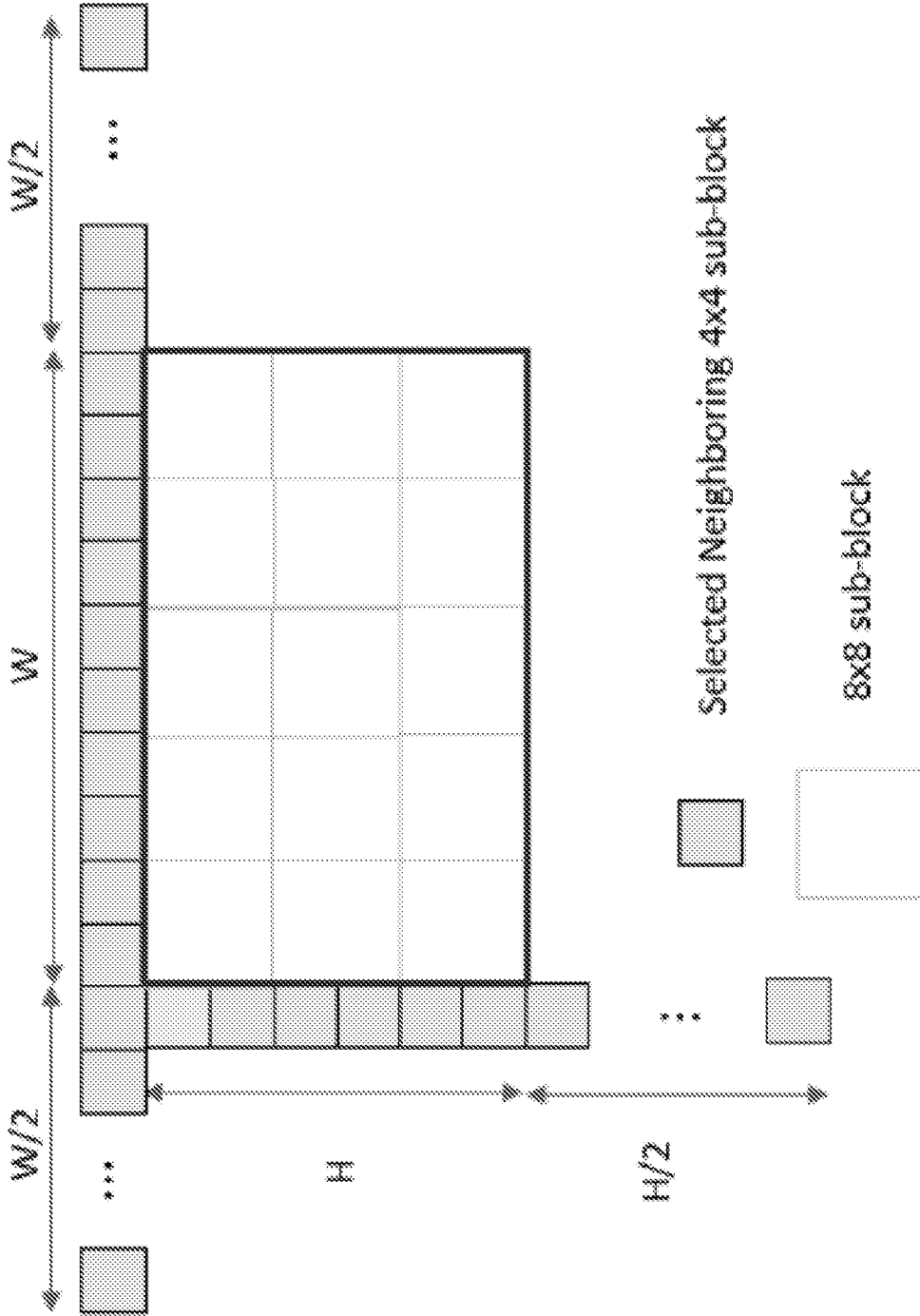


FIG. 17

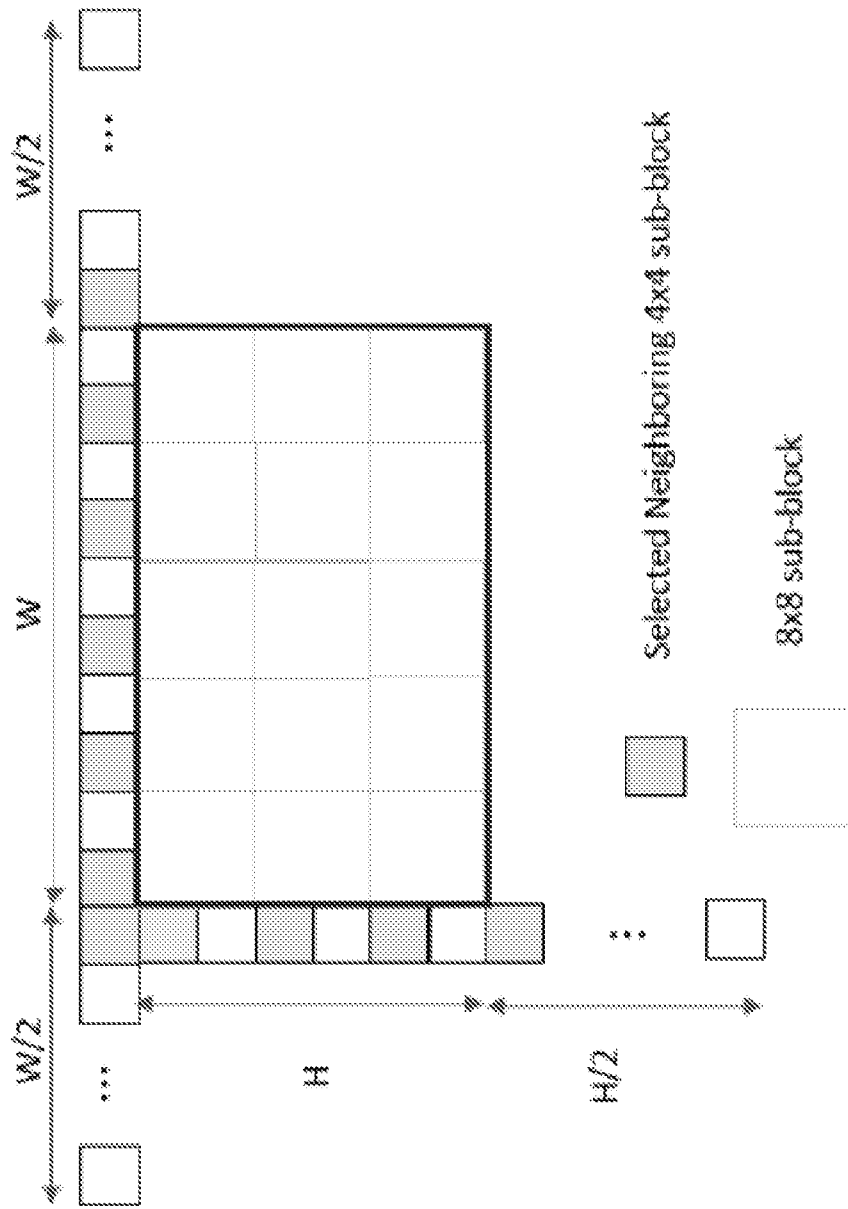


FIG. 18

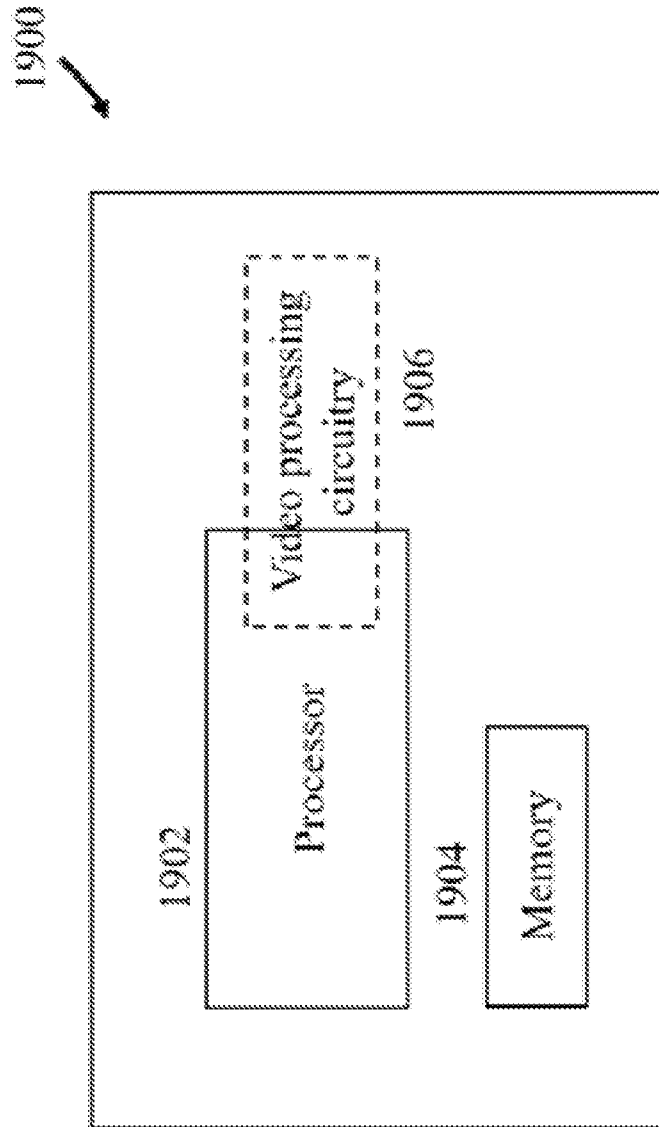


FIG. 19

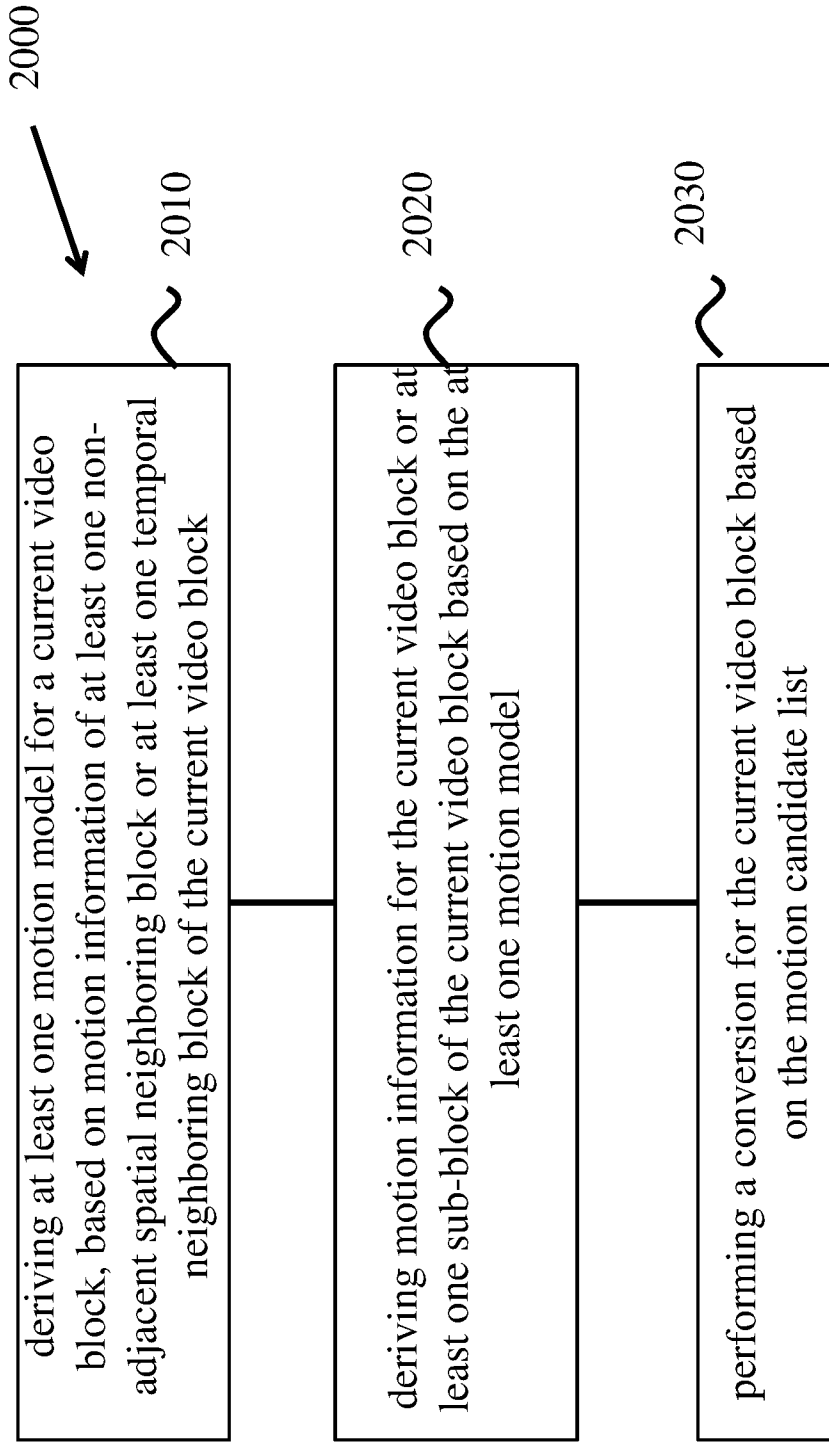


FIG. 20

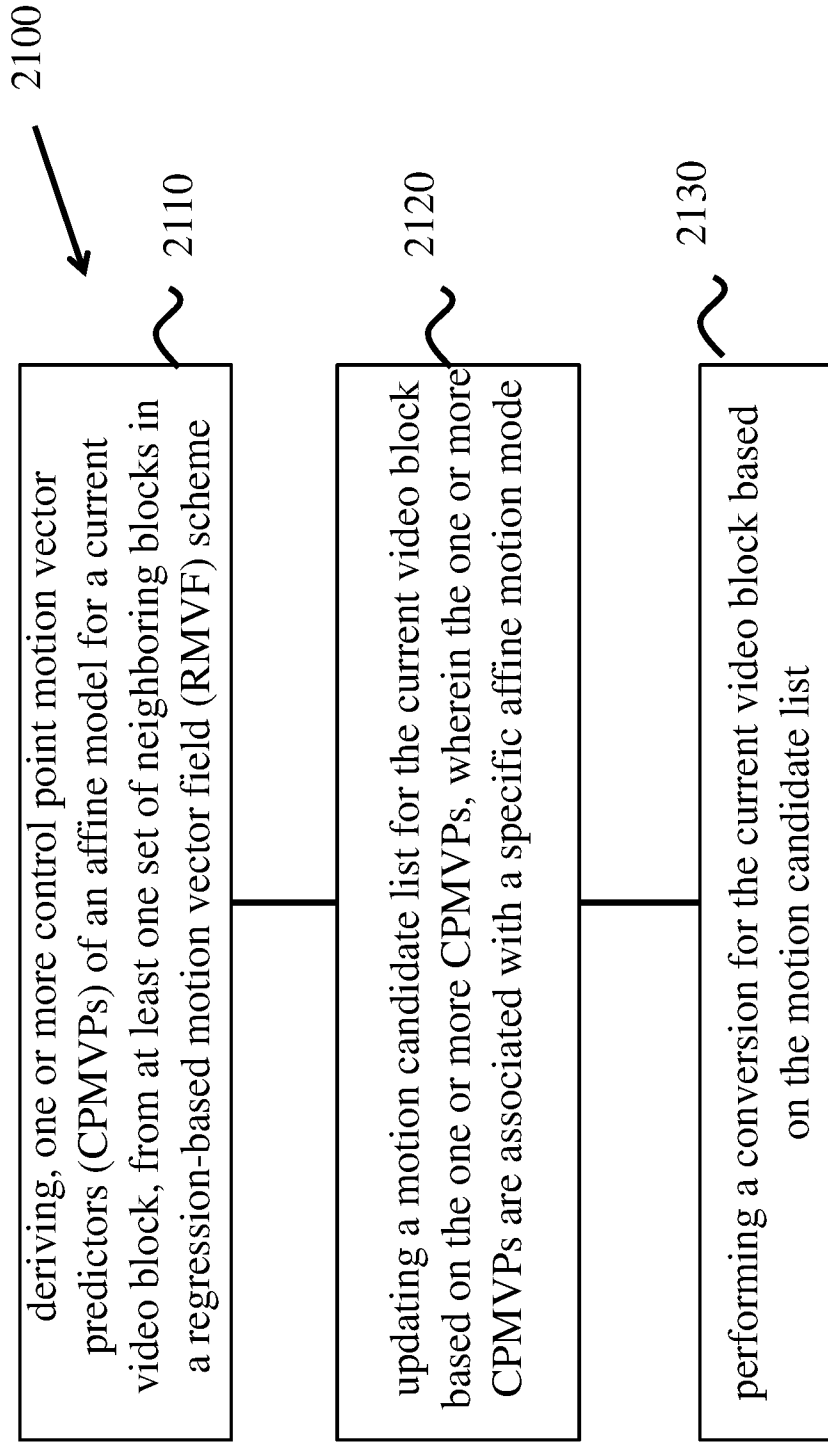


FIG. 21

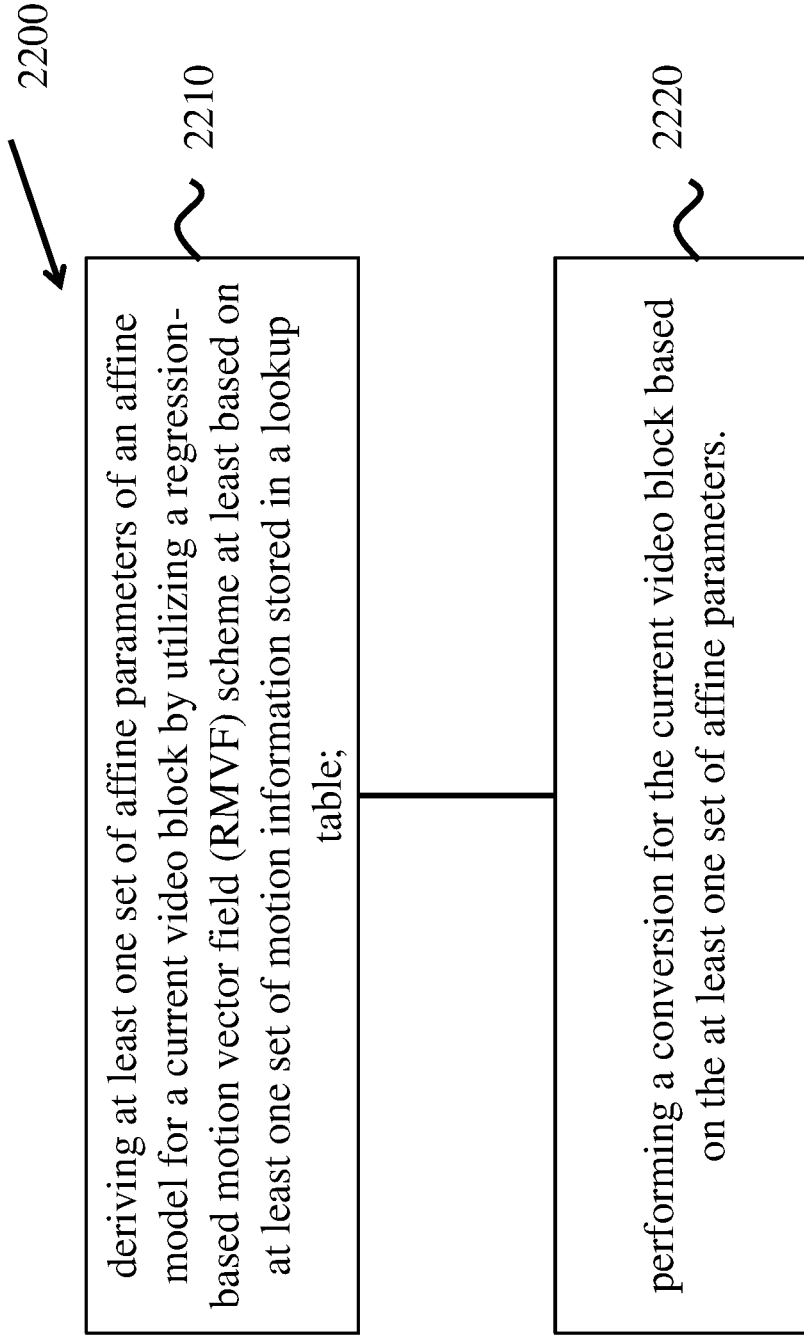


FIG. 22

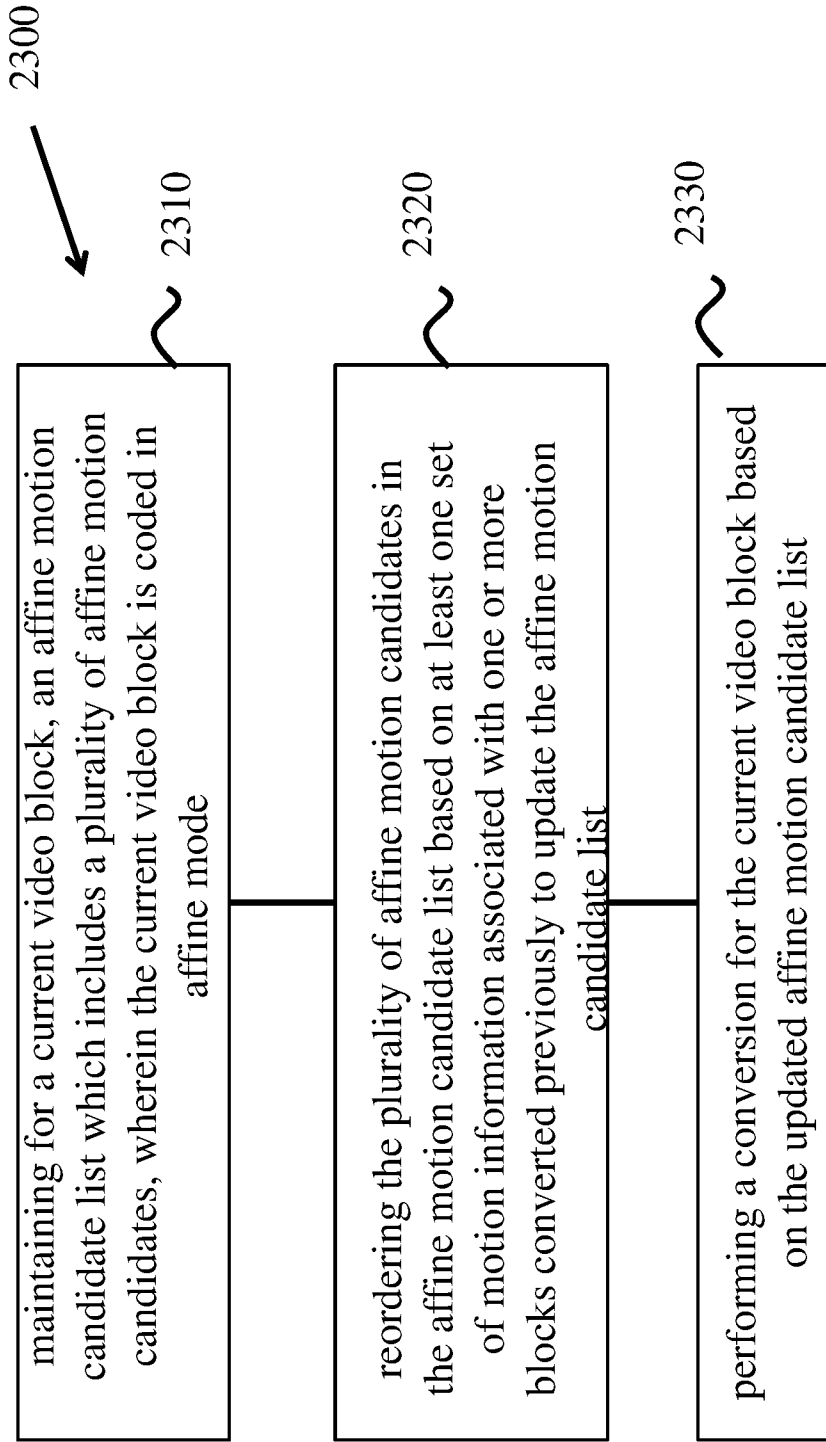


FIG. 23

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/076927

A. CLASSIFICATION OF SUBJECT MATTER		
H04N 19/52(2014.01)i; H04N 19/567(2014.01)i; H04N 19/537(2014.01)i		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
H04N		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
CNPAT, CNKI, WPI, EPODOC, IEEE: video, image, +cod+, +compress+, motion, mov+, model, affine, non, exclud+, adjacent, spatial, temporal, neighboring, block?, sub+, regress+, base?, vector, field, RMVF		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2018126163 A1 (QUALCOMM INCORPORATED) 05 July 2018 (2018-07-05) description, paragraphs 0088-0109	1-22, 64-65
Y	WO 2019002215 A1 (INTERDIGITAL VC HOLDINGS, INC.) 03 January 2019 (2019-01-03) description, paragraphs 7-21	1-22, 64-65
Y	EP 3422720 A1 (THOMSON LICENSING) 02 January 2019 (2019-01-02) description, paragraphs 0004-0024	1-22, 64-65
Y	US 2018359483 A1 (QUALCOMM INCORPORATED) 13 December 2018 (2018-12-13) description, paragraphs 0102, 0109	1-22, 64-65
Y	GHAZNAVI-YOUVALARI, Ramin et al. "CE4-related: Merge mode with Regression based Motion Vector Field (RMVF) (JVET-L0171)" <i>Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 12th Meeting: Macao, CN, 06 October 2018 (2018-10-06),</i> abstract-2 proposed method	4, 10-18
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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
13 May 2020		27 May 2020
Name and mailing address of the ISA/CN		Authorized officer
National Intellectual Property Administration, PRC 6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088 China		XIE, Jiani
Facsimile No. (86-10)62019451		Telephone No. 86-(10)-53961703

C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	GHAZNAVI-YOUVALARI, Ramin et al. "CE2: Merge Mode with Regression-based Motion Vector Field (Test 2.3.3) (JVET-M0302)" <i>Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 13th Meeting: Marrakech, MA, 02 January 2019 (2019-01-02),</i> abstract-2 proposed method	4, 10-18
A	CN 106537915 A (MEDIATEK SINGAPORE PTE. LTD.) 22 March 2017 (2017-03-22) the whole document	1-22, 64-65
A	WO 2017200771 A1 (QUALCOMM INCORPORATED) 23 November 2017 (2017-11-23) the whole document	1-22, 64-65

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] claim 1, which refers to a method comprising deriving at least one motion model for a current video block;
- [2] claim 23, which refers to a method comprising deriving one or more CPMVPs of an affine model for a current video block;
- [3] claim 41, which refers to a method comprising deriving at least one set of affine parameters of an affine model for a current video block by utilizing a RMVF scheme at least based on at least one set of motion information stored in a lookup table;
- [4] claim 47, which refers to a method comprising maintaining for a current video block, an affine motion candidate list which includes a plurality of affine motion candidates, wherein the current video block is coded in affine mode.
- [5] The four inventions do not have the same or corresponding technical feature. Further, they do not have the same or corresponding special technical feature. Therefore the inventions are not so linked as to form a single general inventive concept, as required by PCT Rule 13.1, 13.2 and 13.3.

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.: **1-22, 64, 65**

- 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/CN2020/076927

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
WO	2018126163	A1	05 July 2018	EP	3563572	A1	06 November 2019
				CN	110024403	A	16 July 2019
				US	2018192069	A1	05 July 2018
WO	2019002215	A1	03 January 2019	CN	110870314	A	06 March 2020
EP	3422720	A1	02 January 2019	None			
US	2018359483	A1	13 December 2018	WO	2018231700	A1	20 December 2018
				US	10602180	B2	24 March 2020
				CA	3063173	A1	20 December 2018
				TW	201904299	A	16 January 2019
				CO	2019013930	A2	17 January 2020
				AU	2018283967	A1	05 December 2019
				CN	110754086	A	04 February 2020
				KR	20200017406	A	18 February 2020
CN	106537915	A	22 March 2017	EP	3167613	A4	27 December 2017
				US	10582210	B2	03 March 2020
				WO	2016008408	A1	21 January 2016
				US	2017214932	A1	27 July 2017
				CN	106537915	B	05 July 2019
				BR	112017001100	A2	10 July 2018
				EP	3167613	A1	17 May 2017
				WO	2016008157	A1	21 January 2016
WO	2017200771	A1	23 November 2017	US	10560712	B2	11 February 2020
				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
				JP	2019519980	A	11 July 2019
				CN	109155855	A	04 January 2019