



(51) International Patent Classification:

H04N 19/00 (2014.01)

(21) International Application Number:

PCT/CN2020/078749

(22) International Filing Date:

11 March 2020 (11.03.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2019/077639

11 March 2019 (11.03.2019)

CN

(71) Applicants: **BEIJING BYTEDANCE NETWORK**

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

(72) Inventors: **ZHANG, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **LIU, Hongbin**; Jinritoutiao Post Office, Chi-

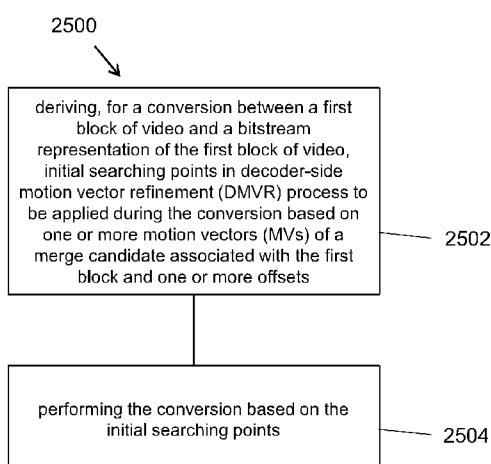
na Satellite Communications Tower, No. 63, Zhichun Road, Haidian District, Beijing 100080 (CN). **XU, Jizheng**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **WANG, Yue**; Jinritoutiao Post Office, China Satellite Communications Tower, No. 63, Zhichun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

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

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, 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,

(54) Title: MOTION VECTOR REFINEMENT IN VIDEO CODING



(57) Abstract: Motion Vector Refinement in Video Coding is disclosed. A method of video processing comprises: deriving, for a conversion between a first block of video and a bitstream representation of the first block of video, initial searching points in decoder-side motion vector refinement (DMVR) process to be applied during the conversion based on one or more motion vectors (MVs) of a merge candidate associated with the first block and one or more offsets; and performing the conversion based on the initial searching points.

FIG. 25

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))*

MOTION VECTOR REFINEMENT IN VIDEO CODING

CROSS-REFERENCE TO RELATED APPLICATION

[0001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefits of International Patent Application No. PCT/CN2019/077639, filed on March 11, 2019. The entire disclosures of International Patent Application No. PCT/CN2019/077639 is incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

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

BACKGROUND

[0001] Currently, efforts are underway to improve the performance of current video codec technologies to provide better compression ratios or provide video coding and decoding schemes that allow for lower complexity or parallelized implementations. Industry experts have recently proposed several new video coding tools and tests are currently underway for determining their effectivity.

SUMMARY

[0003] Devices, systems and methods related to digital video coding, and specifically, to management of 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.

[0004] In one representative aspect, the disclosed technology may be used to perform a method of visual media processing. The method includes performing a conversion between a current video block and a bitstream representation of the current video block, wherein the conversion includes a decoder motion vector refinement (DMVR) step for refining motion information signaled in the bitstream representation; and using, during the DMVR step, at least

one motion vector as a starting value for the refining, wherein the at least one motion vector equals an offset added to a candidate motion vector in a set of candidate motion vectors.

[0005] In another representative aspect, the disclosed technology may be used to perform another method of visual media processing. The method includes performing a conversion between a current video block and a bitstream representation of the current video block, wherein the conversion includes a use of one or more of: a decoder motion vector refinement (DMVR) step, a Bi-directional Optical flow (BDOF) step, or a combined intra-inter prediction step, and wherein co-existence of the DMVR step, the BDOF step, and the combined intra-inter prediction step is based at least on a dimension of the current video block.

[0006] In yet another representative aspect, the disclosed technology may be used to perform another method of visual media processing. The method includes performing a conversion between a current video block and a bitstream representation of the current video block, wherein the conversion includes a decoder motion vector refinement (DMVR) step for refining original motion information signaled in the bitstream representation thereby resulting in a refined motion information usable in a de-blocking step; and computing, for at least a subset block of the current video block, a difference of the refined motion information and the original motion information.

[0007] In another representative aspect, the disclosed technology may be used to perform another method of visual media processing. The method includes deriving, for a conversion between a first block of video and a bitstream representation of the first block of video, initial searching points in decoder-side motion vector refinement (DMVR) process to be applied during the conversion based on one or more motion vectors (MVs) of a merge candidate associated with the first block and one or more offsets; and performing the conversion based on the initial searching points.

[0008] In another representative aspect, the disclosed technology may be used to perform another method of visual media processing. The method includes determining, for a conversion between a first block of video and a bitstream representation of the first block of video, that at least one of decoder motion vector refinement (DMVR) process, Bi-directional Optical flow (BDOF) process and combined intra-inter prediction process is disabled based on a predetermined rule; and performing the conversion based on the determination.

[0009] In another representative aspect, the disclosed technology may be used to perform another method of visual media processing. The method includes deriving, for a conversion between a first block of video and a bitstream representation of the first block of video, motion vectors (MVs) associated with the first block, the MVs being refined by applying decoder-side motion vector refinement (DMVR) process; and performing the conversion by using the refined MVs in de-blocking process.

[0010] In another representative aspect, the disclosed technology may be used to perform another method of visual media processing. The method includes calculating, for a conversion between a first block of video and a bitstream representation of the first block of video, motion vector (MV) difference (dMV) between refined MV (rMV) and non-refined MV (nMV) associated with each basic block of the first block, the rMV being a motion vector refined by applying decoder-side motion vector refinement (DMVR) process, the nMV being a motion vector not refined by the DMVR process; and performing the conversion by using the calculated MV difference.

[0011] Further, in a 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.

[0012] Also, a computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out any one or more of the disclosed methods is disclosed.

[0013] 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

[0014] FIG. 1 shows an example of constructing a merge candidate list.

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

[0016] FIG. 3 shows an example of candidate pairs subject to a redundancy check of spatial merge candidates.

[0017] 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.

5 [0018] FIG. 5 shows an example of motion vector scaling for temporal merge candidates.

[0019] FIG. 6 shows an example of candidate positions for temporal merge candidates.

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

[0021] FIG. 8 shows an example of constructing motion vector prediction candidates.

[0022] FIG. 9 shows an example of motion vector scaling for spatial motion vector candidates.

10 [0023] FIG. 10 show examples of neighboring samples for deriving local illumination compensation parameters.

[0024] FIGs. 11A and 11B show illustrations in connection with a 4-parameter affine model and a 6-parameter affine model respectively.

[0025] FIG. 12 shows an example of an affine motion vector field per sub-block.

15 [0026] FIGs. 13A and 13B show examples of a 4-parameter affine model and a 6-parameter affine model respectively.

[0027] FIG. 14 shows an example of motion vector prediction for affine inter mode for inherited affine candidates.

20 [0028] FIG. 15 shows an example of motion vector prediction for affine inter mode for constructed affine candidates.

[0029] FIGs. 16A and 16B show illustrations in connection with an affine merge mode.

[0030] FIG. 17 shows examples of candidate positions for an affine merge mode

[0031] FIG. 18 shows an example of a merge with motion vector differences (MMVD) mode search process.

25 [0032] FIG. 19 shows an example of a MMVD search point.

[0033] FIG. 20 shows an example of decoder-side motion video refinement (DMVR) in JEM7.

[0034] FIG. 21 show an example of motion vector difference (MVD) in connection with DMVR.

5 [0035] FIG. 22 show an example illustrating checks on motion vectors.

[0036] FIG. 23 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.

[0037] FIG. 24 shows a flowchart of an example method for video coding.

[0038] FIG. 25 shows a flowchart of an example method for video coding.

10 [0039] FIG. 26 shows a flowchart of an example method for video coding.

[0040] FIG. 27 shows a flowchart of an example method for video coding.

[0041] FIG. 28 shows a flowchart of an example method for video coding.

DETAILED DESCRIPTION

15 1. Video coding in HEVC/H.265

[0042] 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
20 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

[0043] Each inter-predicted PU has motion parameters for one or two reference picture lists.

5 Motion parameters include a motion vector and a reference picture index. Usage of one of the two reference picture lists may also be signalled using *inter_pred_idc*. Motion vectors may be explicitly coded as deltas relative to predictors.

[0044] 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
10 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
15 for each reference picture list and reference picture list usage are signalled explicitly per each PU. Such a mode is named Advanced motion vector prediction (AMVP) in this disclosure.

[0045] When signalling indicates that one of the two reference picture lists is to be used, the PU is produced from one block of samples. This is referred to as ‘uni-prediction’. Uni-prediction is available both for P-slices and B-slices.

20 [0046] When signalling indicates that both of the reference picture lists are to be used, the PU is produced from two blocks of samples. This is referred to as ‘bi-prediction’. Bi-prediction is available for B-slices only.

[0047] The following text provides the details on the inter prediction modes specified in HEVC. The description will start with the merge mode.

25

2.1.1. Reference picture list

[0048] 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.

5 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.

[0049] A single reference picture list, List 0, is used for a P slice and two reference picture lists, List 0 and List 1 are used for B slices. It should be noted reference pictures included in List 0/1
10 could be from past and future pictures in terms of capturing/display order.

2.1.2. Merge Mode

2.1.2.1. Derivation of candidates for merge mode

[0050] 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
15 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

[0051] These steps are also schematically depicted in Figure 2-1. For spatial merge candidate
25 derivation, a maximum of four merge candidates are selected among candidates that are located in five different positions. For temporal merge candidate derivation, a maximum of one merge candidate is selected among two candidates. Since constant number of candidates for each PU is assumed at decoder, additional candidates are generated when the number of candidates obtained

from step 1 does not reach the maximum number of merge candidate (MaxNumMergeCand) which is signalled in slice header. Since the number of candidates is constant, index of best merge candidate is encoded using truncated unary binarization (TU). If the size of CU is equal to 8, all the PUs of the current CU share a single merge candidate list, which is identical to the merge candidate list of the $2N \times 2N$ prediction unit.

[0052] In the following, the operations associated with the aforementioned steps are detailed.

2.1.2.2. Spatial candidates derivation

[0053] In the derivation of spatial merge candidates, a maximum of four merge candidates are selected among candidates located in the positions depicted in FIG. 2. The order of derivation is A1, B1, B0, A0 and B2. Position B2 is considered only when any PU of position A1, B1, B0, A0 is not available (e.g. because it belongs to another slice or tile) or is intra coded. After candidate at position A1 is added, the addition of the remaining candidates is subject to a redundancy check which ensures that candidates with same motion information are excluded from the list so that coding efficiency is improved. To reduce computational complexity, not all possible candidate pairs are considered in the mentioned redundancy check. Instead only the pairs linked with an arrow in FIG. 3 are considered and a candidate is only added to the list if the corresponding candidate used for redundancy check has not the same motion information. Another source of duplicate motion information is the “second PU” associated with partitions different from $2N \times 2N$. As an example, FIG. 4 depicts the second PU for the case of $N \times 2N$ and $2N \times N$, respectively. When the current PU is partitioned as $N \times 2N$, candidate at position A1 is not considered for list construction. In fact, by adding this candidate will lead to two prediction units having the same motion information, which is redundant to just have one PU in a coding unit. Similarly, position B1 is not considered when the current PU is partitioned as $2N \times N$.

2.1.2.3. Temporal candidates derivation

[0054] In this step, only one candidate is added to the list. Particularly, in the derivation of this temporal merge candidate, a scaled motion vector is derived based on co-located PU belonging to the picture which has the smallest POC difference with current picture within the given reference picture list. The reference picture list to be used for derivation of the co-located PU is explicitly

signalled in the slice header. The scaled motion vector for temporal merge candidate is obtained as illustrated by the dotted line in FIG. 5, which is scaled from the motion vector of the co-located PU using the POC distances, tb and td , where tb is defined to be the POC difference between the reference picture of the current picture and the current picture and td is defined to be the POC difference between the reference picture of the co-located picture and the co-located picture. The reference picture index of temporal merge candidate is set equal to zero. A practical realization of the scaling process is described in the HEVC specification. For a B-slice, two motion vectors, one is for reference picture list 0 and the other is for reference picture list 1, are obtained and combined to make the bi-predictive merge candidate.

[0055] 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

[0056] 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.

[0057] 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

5 [0058] 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
10 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

15 [0059] FIG. 8 summarizes derivation process for motion vector prediction candidate.

[0060] 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.

20 [0061] 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
25 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

[0062] 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)

[0063] 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.

[0064] In a spatial scaling process, the motion vector of the neighboring 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

[0065] Apart for the reference picture index derivation, all processes for the derivation of temporal merge candidates are the same as for the derivation of spatial motion vector candidates (see FIG. 6). The reference picture index is signalled to the decoder.

5 2.2. Local illumination compensation in JEM

[0066] Local Illumination Compensation (LIC) is based on a linear model for illumination changes, using a scaling factor a and an offset b . And it is enabled or disabled adaptively for each inter-mode coded coding unit (CU).

[0067] When LIC applies for a CU, a least square error method is employed to derive the parameters a and b by using the neighboring samples of the current CU and their corresponding reference samples. More specifically, as illustrated in FIG. 10, the subsampled (2:1 subsampling) neighboring samples of the CU and the corresponding samples (identified by motion information of the current CU or sub-CU) in the reference picture are used.

2.2.1 Derivation of prediction blocks

15 [0068] The LIC parameters are derived and applied for each prediction direction separately. For each prediction direction, a first prediction block is generated with the decoded motion information, then a temporary prediction block is obtained via applying the LIC model. Afterwards, the two temporary prediction blocks are utilized to derive the final prediction block.

[0069] When a CU is coded with merge mode, the LIC flag is copied from neighboring
20 blocks, in a way similar to motion information copy in merge mode; otherwise, an LIC flag is signaled for the CU to indicate whether LIC applies or not.

[0070] When LIC is enabled for a picture, additional CU level RD check is needed to determine whether LIC is applied or not for a CU. When LIC is enabled for a CU, mean-removed sum of absolute difference (MR-SAD) and mean-removed sum of absolute Hadamard-transformed
25 difference (MR-SATD) are used, instead of SAD and SATD, for integer pel motion search and fractional pel motion search, respectively.

[0071] To reduce the encoding complexity, the following encoding scheme is applied in the JEM.

- LIC is disabled for the entire picture when there is no obvious illumination change between a current picture and its reference pictures. To identify this situation, histograms of a current picture and every reference picture of the current picture are calculated at the encoder. If the histogram difference between the current picture and every reference picture of the current picture is smaller than a given threshold, LIC is disabled for the current picture; otherwise, LIC is enabled for the current picture.

2.3. Inter prediction methods in VVC

10 [0072] 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) (a.k.a. Bi-directional optical flow (BDOF)).

2.3.1. Coding block structure in VVC

15 [0073] In VVC, a QuadTree/BinaryTree/MultipleTree (QT/BT/TT) structure is adopted to divide a picture into square or rectangle blocks.

[0074] Besides QT/BT/TT, separate tree (a.k.a. Dual coding tree) is also adopted in VVC for I-slices/tiles. With separate tree, the coding block structure are signaled separately for the luma and chroma components.

20 2.3.2. Adaptive motion vector difference resolution

[0075] In HEVC, motion vector differences (MVDs) (between the motion vector and predicted motion vector of a PU) are signalled in units of quarter luma samples when use_integer_mv_flag is equal to 0 in the slice header. In the VVC, a locally adaptive motion vector resolution (AMVR) is introduced. In the VVC, MVD can be coded in units of quarter luma samples, integer luma samples or four luma samples (i.e., 1/4-pel, 1-pel, 4-pel). The MVD resolution is controlled at the coding unit (CU) level, and MVD resolution flags are conditionally signalled for each CU that has at least one non-zero MVD components.

25

[0076] For a CU that has at least one non-zero MVD components, a first flag is signalled to indicate whether quarter luma sample MV precision is used in the CU. When the first flag (equal to 1) indicates that quarter luma sample MV precision is not used, another flag is signalled to indicate whether integer luma sample MV precision or four luma sample MV precision is used.

5 [0077] When the first MVD resolution flag of a CU is zero, or not coded for a CU (meaning all MVDs in the CU are zero), the quarter luma sample MV resolution is used for the CU. When a CU uses integer-luma sample MV precision or four-luma-sample MV precision, the MVPs in the AMVP candidate list for the CU are rounded to the corresponding precision.

2.3.3. Affine motion compensation prediction

10 [0078] In HEVC, only translation motion model is applied for motion compensation prediction (MCP). While in the real world, there are many kinds of motion, e.g. zoom in/out, rotation, perspective motions and the other irregular motions. In VVC, a simplified affine transform motion compensation prediction is applied with 4-parameter affine model and 6-parameter affine model. As shown FIG. 11, the affine motion field of the block is described by two control point
15 motion vectors (CPMVs) for the 4-parameter affine model and 3 CPMVs for the 6-parameter affine model.

[0079] The motion vector field (MVF) of a block is described by the following equations with the 4-parameter affine model (wherein the 4-parameter are defined as the variables a , b , e and f) in equation (1) and 6-parameter affine model (wherein the 4-parameter are defined as the variables a , b ,
20 c , d , e and f) in equation (2) respectively:

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

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

where (mv_0^h, mv_0^v) is motion vector of the top-left corner control point, and (mv_1^h, mv_1^v) is motion vector of the top-right corner control point and (mv_2^h, mv_2^v) is motion vector of the bottom-left

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

[0080] In order to further simplify the motion compensation prediction, sub-block based affine transform prediction is applied. To derive motion vector of each $M \times N$ (both M and N are set to 4 in current VVC) sub-block, the motion vector of the center sample of each sub-block, as shown in FIG. 12, is calculated according to Equation (1) and (2), and rounded to $1/16$ fraction accuracy. Then the motion compensation interpolation filters for $1/16$ -pel are applied to generate the prediction of each sub-block with derived motion vector. The interpolation filters for $1/16$ -pel are introduced by the affine mode.

[0081] After MCP, the high accuracy motion vector of each sub-block is rounded and saved as the same accuracy as the normal motion vector.

2.3.3.1. Signaling of affine prediction

[0082] Similar to the translational motion model, there are also two modes for signaling the side information due affine prediction. They are AFFINE_INTER and AFFINE_MERGE modes.

2.3.3.2. AF_INTER mode

[0083] For CUs with both width and height larger than 8, AF_INTER mode can be applied. An affine flag in CU level is signaled in the bitstream to indicate whether AF_INTER mode is used.

[0084] In this mode, for each reference picture list (List 0 or List 1), an affine AMVP candidate list is constructed with three types of affine motion predictors in the following order,

wherein each candidate includes the estimated CPMVs of the current block. The differences of the best CPMVs found at the encoder side (such as $mv_0\ mv_1\ mv_2$ in FIG. 15) and the estimated CPMVs are signaled. In addition, the index of affine AMVP candidate from which the estimated CPMVs are derived is further signaled.

5 1) Inherited affine motion predictors

[0085] The checking order is similar to that of spatial MVPs in HEVC AMVP list construction. First, a left inherited affine motion predictor is derived from the first block in $\{A1, A0\}$ that is affine coded and has the same reference picture as in current block. Second, an above inherited affine motion predictor is derived from the first block in $\{B1, B0, B2\}$ that is affine coded and has the same reference picture as in current block. The five blocks A1, A0, B1, B0, B2 are depicted in FIG. 14.

[0086] Once a neighboring block is found to be coded with affine mode, the CPMVs of the coding unit covering the neighboring block are used to derive predictors of CPMVs of current block. For example, if A1 is coded with non-affine mode and A0 is coded with 4-parameter affine mode, the left inherited affine MV predictor will be derived from A0. In this case, the CPMVs of a CU covering A0, as denoted by MV_0^N for the top-left CPMV and MV_1^N for the top-right CPMV in FIG. 16B are utilized to derive the estimated CPMVs of current block, denoted by MV_0^C, MV_1^C, MV_2^C for the top-left (with coordinate (x0, y0)), top-right (with coordinate (x1, y1)) and bottom-right positions (with coordinate (x2, y2)) of current block.

20 2) Constructed affine motion predictors

[0087] A constructed affine motion predictor consists of control-point motion vectors (CPMVs) that are derived from neighboring inter coded blocks, as shown in FIG. 15, that have the same reference picture. If the current affine motion model is 4-parameter affine, the number of CPMVs is 2, otherwise if the current affine motion model is 6-parameter affine, the number of CPMVs is 3. The top-left CPMV $\overline{mv}_0$ is derived by the MV at the first block in the group $\{A, B, C\}$ that is inter coded and has the same reference picture as in current block. The top-right CPMV $\overline{mv}_1$ is derived by the MV at the first block in the group $\{D, E\}$ that is inter coded and has the same reference picture as in current block. The bottom-left CPMV $\overline{mv}_2$ is derived by the MV at the first block in the group $\{F, G\}$ that is inter coded and has the same reference picture as in current block.

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

10 [0088] No pruning process is applied when inserting a constructed affine motion predictor into the candidate list.

3) Normal AMVP motion predictors

[0089] The following applies until the number of affine motion predictors reaches the maximum.

- 15 1) Derive an affine motion predictor by setting all CPMVs equal to $\overline{mv}_2$ if available.
- 2) Derive an affine motion predictor by setting all CPMVs equal to $\overline{mv}_1$ if available.
- 3) Derive an affine motion predictor by setting all CPMVs equal to $\overline{mv}_0$ if available.
- 4) Derive an affine motion predictor by setting all CPMVs equal to HEVC TMVP if available.
- 5) Derive an affine motion predictor by setting all CPMVs to zero MV.

20 Note that $\overline{mv}_i$ is already derived in constructed affine motion predictor.

[0090] In AF_INTER mode, when 4/6-parameter affine mode is used, 2/3 control points are used, and therefore 2/3 MVD needs to be coded for these control points, as shown in FIG. 13. It is proposed to derive the MV as follows, i.e., mvd_1 and mvd_2 are predicted from mvd_0 .

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

$$25 \quad mv_1 = \overline{mv}_1 + mvd_1 + mvd_0$$

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

wherein $\overline{mv}_i$, mvd_i and mv_i are the predicted motion vector, motion vector difference and motion vector of the top-left pixel ($i = 0$), top-right pixel ($i = 1$) or left-bottom pixel ($i = 2$) respectively, as

shown in FIG. 13B. Please note that the addition of two motion vectors (e.g., $mvA(xA, yA)$ and $mvB(xB, yB)$) is equal to summation of two components separately, that is, $newMV = mvA + mvB$ and the two components of $newMV$ is set to $(xA + xB)$ and $(yA + yB)$, respectively.

2.3.3.3. AF_MERGE mode

5 [0091] When a CU is applied in AF_MERGE mode, it gets the first block coded with affine mode from the valid neighbor reconstructed blocks. And the selection order for the candidate block is from left, above, above right, left bottom to above left as shown in FIG. 16A (denoted by A, B, C, D, E in order). For example, if the neighbor left bottom block is coded in affine mode as denoted by A0 in FIG. 16B, the Control Point (CP) motion vectors mv_0^N , mv_1^N and mv_2^N of the top left corner, above right corner and left bottom corner of the neighboring CU/PU which contains the block A are
 10 fetched. And the motion vector mv_0^C , mv_1^C and mv_2^C (which is only used for the 6-parameter affine model) of the top left corner/top right/bottom left on the current CU/PU is calculated based on mv_0^N , mv_1^N and mv_2^N . It should be noted that in the current VTM, sub-block (e.g. 4×4 block in VTM) located at the top-left corner stores mv_0 , the sub-block located at the top-right corner stores mv_1 if
 15 the current block is affine coded. If the current block is coded with the 6-parameter affine model, the sub-block located at the bottom-left corner stores mv_2 ; otherwise (with the 4-parameter affine model), LB stores mv_2' . Other sub-blocks store the MVs used for MC.

[0092] After the CPMV of the current CU mv_0^C , mv_1^C and mv_2^C are derived, according to the simplified affine motion model Equation (1) and (2), the MVF of the current CU is generated. In
 20 order to identify whether the current CU is coded with AF_MERGE mode, an affine flag is signaled in the bitstream when there is at least one neighbor block is coded in affine mode.

[0093] An affine merge candidate list is constructed with following steps:

1) Insert inherited affine candidates

[0094] Inherited affine candidate means that the candidate is derived from the affine motion
 25 model of its valid neighbor affine coded block. The maximum two inherited affine candidates are derived from affine motion model of the neighboring blocks and inserted into the candidate list. For the left predictor, the scan order is {A0, A1}; for the above predictor, the scan order is {B0, B1, B2}.

2) Insert constructed affine candidates

[0095] If the number of candidates in affine merge candidate list is less than MaxNumAffineCand (e.g., 5), constructed affine candidates are inserted into the candidate list. Constructed affine candidate means the candidate is constructed by combining the neighbor motion information of each control point.

- a) The motion information for the control points is derived firstly from the specified spatial neighbors and temporal neighbor shown in Figure 2-19. 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.

[0096] 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.

[0097] 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.

b) Secondly, the combinations of controls points are used to construct an affine merge candidate.

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

II. Motion information of two control points are needed to construct a 4-parameter affine candidate. The two control points can be selected from one of the two combinations

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

III. The combinations of constructed affine candidates are inserted into to candidate list as following order:

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

i. For each combination, the reference indices of list X for each CP are checked, if they are all the same, then this combination has valid CPMVs for list X. If the combination does not have valid CPMVs for both list 0 and list 1, then this
10 combination is marked as invalid. Otherwise, it is valid, and the CPMVs are put into the sub-block merge list.

3) Padding with zero motion vectors

[0098] If the number of candidates in affine merge candidate list is less than 5, zero motion vectors with zero reference indices are insert into the candidate list, until the list is full.

15 [0099] More specifically, for the sub-block merge candidate list, a 4-parameter merge candidate with MVs set to (0, 0) and prediction direction set to uni-prediction from list 0 (for P slice) and bi-prediction (for B slice).

2.3.4. Merge with motion vector differences (MMVD)

[0100] Ultimate motion vector expression (UMVE, also known as MMVD) is presented.
20 UMVE is used for either skip or merge modes with a proposed motion vector expression method.

[0101] FIG. 18 shows an example of an ultimate vector expression (UMVE) search process. FIG. 19 shows an example of a UMVE search point. UMVE re-uses merge candidate as same as those included in the regular merge candidate list in VVC. Among the merge candidates, a base candidate can be selected, and is further expanded by the proposed motion vector expression
25 method.

[0102] UMVE provides a new motion vector difference (MVD) representation method, in which a starting point, a motion magnitude and a motion direction are used to represent a MVD.

[0103] This proposed technique uses a merge candidate list as it is. But only candidates which are default merge type (MRG_TYPE_DEFAULT_N) are considered for UMVE's expansion.

[0104] Base candidate index defines the starting point. Base candidate index indicates the best candidate among candidates in the list as follows.

5

Table 1. Base candidate IDX

Base candidate IDX	0	1	2	3
N th MVP	1 st MVP	2 nd MVP	3 rd MVP	4 th MVP

[0105] If the number of base candidates is equal to 1, Base candidate IDX is not signaled.

[0106] Distance index is motion magnitude information. Distance index indicates the pre-defined distance from the starting point information. Pre-defined distance is as follows:

10

Table 2. Distance IDX

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

[0107] Direction index represents the direction of the MVD relative to the starting point. The direction index can represent of the four directions as shown below.

15

Table 3. Direction IDX

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

[0108] UMVE flag is signaled right after sending a skip flag or merge flag. If skip or merge flag is true, UMVE flag is parsed. If UMVE flag is equal to 1, UMVE syntaxes are parsed. But, if not 1, AFFINE flag is parsed. If AFFINE flag is equal to 1, that is AFFINE mode, But, if not 1, skip/merge index is parsed for VTM's skip/merge mode.

5 [0109] Additional line buffer due to UMVE candidates is not needed. Because a skip/merge candidate of software is directly used as a base candidate. Using input UMVE index, the supplement of MV is decided right before motion compensation. There is no need to hold long line buffer for this.

[0110] In current common test condition, either the first or the second merge candidate in the
10 merge candidate list could be selected as the base candidate.

[0111] UMVE is also known as Merge with MV Differences (MMVD) in VVC.

2.3.5. Decoder-side Motion Vector Refinement (DMVR)

[0112] In bi-prediction operation, for the prediction of one block region, two prediction blocks, formed using a motion vector (MV) of list0 and a MV of list1, respectively, are combined to
15 form a single prediction signal. In the decoder-side motion vector refinement (DMVR) method, the two motion vectors of the bi-prediction are further refined.

2.3.5.1. DMVR in JEM

[0113] In JEM design, the motion vectors are refined by a bilateral template matching process. The bilateral template matching applied in the decoder to perform a distortion-based search between
20 a bilateral template and the reconstruction samples in the reference pictures in order to obtain a refined MV without transmission of additional motion information. An example is depicted in FIG. 20. The bilateral template is generated as the weighted combination (i.e. average) of the two prediction blocks, from the initial MV0 of list0 and MV1 of list1, respectively, as shown in FIG. 20. The template matching operation consists of calculating cost measures between the generated
25 template and the sample region (around the initial prediction block) in the reference picture. For each of the two reference pictures, the MV that yields the minimum template cost is considered as the updated MV of that list to replace the original one. In the JEM, nine MV candidates are

searched for each list. The nine MV candidates include the original MV and 8 surrounding MVs with one luma sample offset to the original MV in either the horizontal or vertical direction, or both. Finally, the two new MVs, i.e., MV0' and MV1' as shown in FIG. 20, are used for generating the final bi-prediction results. A sum of absolute differences (SAD) is used as the cost measure. Please
 5 note that when calculating the cost of a prediction block generated by one surrounding MV, the rounded MV (to integer pel) is actually used to obtain the prediction block instead of the real MV.

2.3.5.2. DMVR in VVC

[0114] For DMVR in VVC, MVD mirroring between list 0 and list 1 is assumed as shown in FIG. 21, and bilateral matching is performed to refine the MVs, i.e., to find the best MVD among
 10 several MVD candidates. Denote the MVs for two reference picture lists by MVL0(L0X, L0Y), and MVL1(L1X, L1Y). The MVD denoted by (MvdX, MvdY) for list 0 that could minimize the cost function (e.g., SAD) is defined as the best MVD. For the SAD function, it is defined as the SAD between the reference block of list 0 derived with a motion vector (L0X+MvdX, L0Y+MvdY) in the list 0 reference picture and the reference block of list 1 derived with a motion vector (L1X-
 15 MvdX, L1Y-MvdY) in the list 1 reference picture.

[0115] The motion vector refinement process may iterate twice. In each iteration, at most 6 MVDs (with integer-pel precision) may be checked in two steps, as shown in Figure 2-22. In the first step, MVD (0, 0), (-1, 0), (1, 0), (0, -1), (0, 1) are checked. In the second step, one of the MVD (-1, -1), (-1, 1), (1, -1) or (1, 1) may be selected and further checked. Suppose function Sad(x, y)
 20 returns SAD value of the MVD (x, y). The MVD, denoted by (MvdX, MvdY), checked in the second step is decided as follows:

MvdX = -1;

MvdY = -1;

If (Sad(1, 0) < Sad(-1, 0))

MvdX = 1;

If (Sad(0, 1) < Sad(0, -1))

MvdY = 1;

[0116] In the first iteration, the initial searching point is the MVs of a regular merge candidate, and in the second iteration, the initial searching point is the MVs of a regular merge candidate, plus the selected best MVDs in the first iteration. DMVR applies only when one reference picture is a preceding picture and the other reference picture is a following picture, and the two reference pictures are with same picture order count distance from the current picture.

[0117] To further simplify the process of DMVR, it proposed several changes to the design in JEM. More specifically, the adopted DMVR design to VTM-4.0 (to be released soon) has the following main features:

- Early termination when (0,0) position SAD between list0 and list1 is smaller than a threshold.
- Early termination when SAD between list0 and list1 is zero for some position.
- Block sizes for DMVR: $W \times H \geq 64$ & $H \geq 8$, wherein W and H are the width and height of the block.
- Split the CU into multiple of 16x16 sub-blocks for DMVR of CU size $> 16 \times 16$. If only width or height of the CU is larger than 16, it is only split in vertical or horizontal direction.
- Reference block size $(W+7) \times (H+7)$ (for luma).
- 25 points SAD-based integer-pel search (i.e. $(\pm) 2$ refinement search range, single stage)
- Bilinear-interpolation based DMVR.
- “Parametric error surface equation” based sub-pel refinement. This procedure is performed only when the minimum SAD cost is not equal to zero and the best MVD is (0, 0) in the last MV refinement iteration.
- Luma/chroma MC w/ reference block padding (if needed).
- Refined MVs used for MC and TMVPs only.

2.3.5.2.1 Usage of DMVR

[0118] When the following conditions are all true, DMVR may be enabled:

- DMVR enabling flag in the SPS (i.e., `sps_dmvr_enabled_flag`) is equal to 1
- TPM flag, inter-affine flag and subblock merge flag (either ATMVP or affine merge), MMVD flag are all equal to 0
- Merge flag is equal to 1

- Current block is bi-predicted, and POC distance between current picture and reference picture in list 1 is equal to the POC distance between reference picture in list 0 and current picture
- The current CU height is greater than or equal to 8
- 5 – Number of luma samples (CU width*height) is greater than or equal to 64

2.3.5.2.2. “Parametric error surface equation” based sub-pel refinement

The method is summarized below:

1. The parametric error surface fit is computed only if the center position is the best cost position in a given iteration.
- 10 2. The center position cost and the costs at (-1,0), (0,-1), (1,0) and (0,1) positions from the center are used to fit a 2-D parabolic error surface equation of the form

$$E(x, y) = A(x - x_0)^2 + B(y - y_0)^2 + C$$

where (x_0, y_0) corresponds to the position with the least cost and C corresponds to the minimum cost value. By solving the 5 equations in 5 unknowns, (x_0, y_0) is computed as:

$$x_0 = (E(-1,0) - E(1,0)) / (2(E(-1,0) + E(1,0) - 2E(0,0)))$$

$$y_0 = (E(0,-1) - E(0,1)) / (2(E(0,-1) + E(0,1) - 2E(0,0)))$$

(x_0, y_0) can be computed to any desired sub-pixel precision by adjusting the precision at which the division is performed (i.e. how many bits of quotient are computed). For 1/16th-pel accuracy, just 4-bits in the absolute value of the quotient needs to be computed, which lends itself to a fast shifted subtraction based implementation of the 2 divisions desired per CU.

3. The computed (x_0, y_0) are added to the integer distance refinement MV to get the sub-pixel accurate refinement delta MV.

25 2.3.6. Combined intra and inter prediction

[0119] Multi-hypothesis prediction is proposed, wherein combined intra and inter prediction is one way to generate multiple hypotheses.

[0120] When the multi-hypothesis prediction is applied to improve intra mode, multi-hypothesis prediction combines one intra prediction and one merge indexed prediction. In a merge CU, one flag is signaled for merge mode to select an intra mode from an intra candidate list when the flag is true. For luma component, the intra candidate list is derived from 4 intra prediction modes including DC, planar, horizontal, and vertical modes, and the size of the intra candidate list can be 3 or 4 depending on the block shape. When the CU width is larger than the double of CU height, horizontal mode is exclusive of the intra mode list and when the CU height is larger than the double of CU width, vertical mode is removed from the intra mode list. One intra prediction mode selected by the intra mode index and one merge indexed prediction selected by the merge index are combined using weighted average. For chroma component, DM is always applied without extra signaling. The weights for combining predictions are described as follow. When DC or planar mode is selected, or the CB width or height is smaller than 4, equal weights are applied. For those CBs with CB width and height larger than or equal to 4, when horizontal/vertical mode is selected, one CB is first vertically/horizontally split into four equal-area regions. Each weight set, denoted as (w_intra_i, w_inter_i) , where i is from 1 to 4 and $(w_intra_1, w_inter_1) = (6, 2)$, $(w_intra_2, w_inter_2) = (5, 3)$, $(w_intra_3, w_inter_3) = (3, 5)$, and $(w_intra_4, w_inter_4) = (2, 6)$, will be applied to a corresponding region. (w_intra_1, w_inter_1) is for the region closest to the reference samples and (w_intra_4, w_inter_4) is for the region farthest away from the reference samples. Then, the combined prediction can be calculated by summing up the two weighted predictions and right-shifting 3 bits. Moreover, the intra prediction mode for the intra hypothesis of predictors can be saved for reference of the following neighboring CUs.

3. Drawbacks of existing implementations

[0121] The current DMVR may have the following problems:

1. The initial searching point in DMVR can only be the MVs of a merge candidate.
2. In the worst case, the decoder can conduct DMVR, BDOF and combined inter-intra prediction sequentially for bi-prediction.
3. The non-refined MVs are used for spatial motion vector prediction and de-blocking filter but refined MVs are used as TMVP. Additional memory is required to store the refined MVs.

4. Example techniques and embodiments

[0122] 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.

5 In the following discussion, $SatShift(x, n)$ is defined as

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

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

In one example, $offset0$ and/or $offset1$ are set to $(1 \ll n) \gg 1$ or $(1 \ll (n-1))$. In another example, $offset0$ and/or $offset1$ are set to 0.

10 $Clip3(min, max, x)$ is defined as

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

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

15 1. It is proposed that the initial searching points in DMVR can be the MVs of a regular merge candidate adding offsets.

a. In one example, suppose the MVs of a merge candidate referring to reference list 0 and reference list 1 are denoted as $MV0$ and $MV1$, respectively, then the initial searching points in DMVR can be $MV0 + offset0$ and $MV1 + offset1$.

20 i. In one example, $offset0$ and/or $offset1$ can be predefined. For example, $offset0 = (4, 0)$ and $offset1 = (-4, 0)$;
ii. Alternatively, $offset0$ and $offset1$ may be signaled from the encoder to the decoder.

b. In one example, the offsets can be signaled from the encoder to the decoder with the MMVD mode.

i. For example, DMVR can be applied when the current block is coded as MMVD and/or MMVD skip mode.

ii. When the current block is coded with MMVD and/or MMVD skip mode, the initial searching points in DMVR is set to be the MVs with the MMVD mode, which is derived as the MVs of a merge candidate adding signaled distances.

2. It is proposed that DMVR, BDOF and combined inter-intra prediction cannot be all applied.

a. In one example, DMVR, BDOF and combined inter-intra prediction cannot be all applied when the dimensions of the current block satisfying some conditions. Suppose the width and height of the current block are W and H, respectively.

i. For example, DMVR, BDOF and combined inter-intra prediction cannot be all applied when $W \geq T1$ and $H \geq T2$. For example, $T1=T2=16$;

1) Alternatively, when $W > T1$ and $H > T2$.

ii. For example, DMVR, BDOF and combined inter-intra prediction cannot be all applied when $W \leq T1$ and $H \leq T2$. For example, $T1=T2=16$;

1) Alternatively, when $W < T1$ and $H < T2$.

iii. For example, DMVR, BDOF and combined inter-intra prediction cannot be all applied when $W \geq T1$ or $H \geq T2$. For example, $T1=T2=16$;

1) Alternatively, when $W > T1$ or $H > T2$.

iv. For example, DMVR, BDOF and combined inter-intra prediction cannot be all applied when $W \leq T1$ or $H \leq T2$. For example, $T1=T2=16$;

1) Alternatively, when $W < T1$ or $H < T2$.

v. For example, DMVR, BDOF and combined inter-intra prediction cannot be all applied when $W \times H \geq T1$. For example, $T1=64$;

1) Alternatively, when $W \times H > T1$.

vi. For example, DMVR, BDOF and combined inter-intra prediction cannot be all applied when $W \times H \leq T1$. For example, $T1=64$;

1) Alternatively, when $W \times H < T1$.

- b. In one example, DMVR cannot be used when both BDOF and inter-intra prediction are applied.
 - c. In one example, BDOF cannot be used when both DMVR and inter-intra prediction are applied.
 - 5 d. In one example, inter-intra prediction cannot be used when both DMVR and BDOF are applied.
3. It is proposed that the DMVR refined MVs are used in the de-blocking process.
4. It is proposed that the difference (denoted as dMV) of refined MV (denoted as rMV) and non-refined MV (denoted as nMV) for each basic-block with dimension $w \times h$ in a block is calculated and stored after decoding the block. For example, $w=h=4$, or $w=h=8$, or $w=h=16$.
- 10 a. dMV is derived as $dMV=rMV-nMV$.
- i. Alternatively, $dMV=nMV-rMV$.
- b. In one example, before the deblocking process, rMV' is calculated as $rMV'=dMV+nMV$ and to be used as the MV for the basic block in the following deblocking process and temporal prediction process.
- 15 i. Alternatively, rMV' is calculated as $rMV'=- (dMV+nMV)$.
- c. In one example, dMV_x and dMV_y can be defined in a range in a conforming bit-stream. For example, dMV_x and dMV_y can satisfy $T1_x \leq dMV_x \leq T2_x$ and $T1_y \leq dMV_y \leq T2_y$. For example, $T1_x=T1_y=-2^K$ and $T2_x=T2_y=2^K-1$, where K is an integer such as 3 or 4.
- 20 i. In one example, the searching range in DMVR can guarantee that dMV can satisfy the constrain.
- ii. In one example, $T1_x/T2_x/T1_y/T2_y/ K$ may be signaled from the encoder to the decoder.
- 25 iii. In one example, $T1_x/T2_x/T1_y/T2_y/ K$ may depend on the searching range of DMVR.
- iv. In one example, $T1_x/T2_x/T1_y/T2_y/ K$ may depend on the standard profile and/or level and/or tier.
- d. In one example, dMV may be clipped.

- i. For example, dMV_x is set to be $\text{Clip3}(T1_x, T2_x, dMV_x)$ and dMV_y is set to be $\text{Clip3}(T1_y, T2_y, dMV_y)$. For example, $T1_x=T1_y=-2^K$ and $T2_x=T2_y=2^K-1$, where K is an integer such as 3 or 4.
- e. In one example, dMV may be quantized to be dMV' and dMV' will be stored.
 - i. For example, dMV_x' is set to be $\text{Shift}(dMV_x, N_x)$ and dMV_y' is set to be $\text{Shift}(dMV_y, N_y)$, For example, $N_x=N_y=1$.
 - 1) Alternatively, dMV_x' is set to be $\text{SatShift}(dMV_x, N_x)$ and dMV_y' is set to be $\text{SatShift}(dMV_y, N_y)$, For example, $N_x=N_y=1$.
 - 2) In one example, N_x/N_y may be signaled from the encoder to the decoder.
 - 3) In one example, N_x/N_y may depend on the searching range of DMVR.
 - 4) In one example, N_x/N_y may depend on the standard profile and/or level and/or tier.
 - ii. In one example, dMV may be dequantized from dMV' before deriving rMV' .
 - 1) For example, $dMV_x=dMV_x' \ll N_x$ and $dMV_y=dMV_y' \ll N_y$.
 - iii. In one example, dMV may be clipped before being quantized.
 - 1) Alternatively, dMV' may be clipped after the quantization.
- f. In one example, rMV' instead of rMV may be used in the motion compensation procedure.

5. Example implementations of the disclosed technology

[0123] FIG. 23 is a block diagram of a video processing apparatus 2300. The apparatus 2300 may be used to implement one or more of the methods described herein. The apparatus 2300 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 2300 may include one or more processors 2302, one or more memories 2304 and video processing hardware 2306. The processor(s) 2302 may be configured to implement one or more methods described in the present document. The memory (memories) 2304 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 2306 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 2302 (e.g., graphics processor core GPU or other signal processing circuitry).

[0124] 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.

[0125] 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.

[0126] FIG. 24 is a flowchart for an example method 2400 of video processing. The method 2400 includes, at 2410, performing a conversion between a current video block and a bitstream representation of the current video block, wherein the conversion includes a decoder motion vector refinement (DMVR) step for refining motion information signaled in the bitstream representation. The method include, at step 2420, using, during the DMVR step, at least one motion vector as a starting value for the refining, wherein the at least one motion vector equals an offset added to a candidate motion vector in a set of candidate motion vectors.

[0127] Some embodiments may be described using the following clause-based format.

[0128] 1. A method of visual media processing, comprising:

[0129] performing a conversion between a current video block and a bitstream representation of the current video block, wherein the conversion includes a decoder motion vector refinement (DMVR) step for refining motion information signaled in the bitstream representation; and

[0130] using, during the DMVR step, at least one motion vector as a starting value for the refining, wherein the at least one motion vector equals an offset added to a candidate motion vector in a set of candidate motion vectors.

[0131] 2. The method of clause 1, wherein the candidate motion vector is included in a merge list.

[0132] 3. The method of clause 1, wherein the offset added to the candidate motion vector is predefined.

[0133] 4. The method of clause 1, wherein the offset added to the candidate motion vector is signaled as a parameter in the bitstream representation.

[0134] 5. The method of clause 1, wherein the current video block is coded in a merge with motion vector differences (MMVD) mode.

[0135] 6. The method of clause 1, wherein the current video block is coded in a merge with motion vector differences (MMVD) skip mode.

[0136] 7. The method of any one or more of clauses 5-6, wherein the candidate motion vector is included in a merge list and the offset added to the candidate motion vector is signaled as a parameter in the bitstream representation.

[0137] 8. A method of visual media processing, comprising:

[0138] performing a conversion between a current video block and a bitstream representation of the current video block, wherein the conversion includes a use of one or more of: a decoder motion vector refinement (DMVR) step, a Bi-directional Optical flow (BDOF) step, or a combined intra-inter prediction step, and wherein co-existence of the DMVR step, the BDOF step, and the combined intra-inter prediction step is based at least on a dimension of the current video block.

[0139] 9. The method of clause 8, further comprising:

[0140] in response to determining that a width of the current video block is greater than or equal to a first threshold value and/or a height of the current video block is greater than or equal to a

second threshold value, disabling co-existent use of the DMVR step, the BDOF step, and the combined intra-inter prediction step.

[0141] 10. The method of clause 8, further comprising:

5 [0142] in response to determining that a width of the current video block is less than or equal to a first threshold value and/or a height of the current video block is less than or equal to a second threshold value, disabling co-existent use of the DMVR step, the BDOF step, and the combined intra-inter prediction step.

[0143] 11. The method of any one or more of clauses 9-10, wherein the first threshold value and the second threshold value are both sixteen (16).

10 [0144] 12. The method of clause 8, further comprising:

[0145] in response to determining that a product of a width of the current video block and a height of the current video block is greater than or equal to a first threshold value, disabling co-existent use of the DMVR step, the BDOF step, and the combined intra-inter prediction step.

[0146] 13. The method of clause 8, further comprising:

15 [0147] in response to determining that a product of a width of the current video block and a height of the current video block is less than or equal to a first threshold value, disabling co-existent use of the DMVR step, the BDOF step, and the combined intra-inter prediction step.

[0148] 14. The method of any one or more of clauses 12-13, wherein the first threshold value is sixty four (64).

20 [0149] 15. The method of any one or more of clause 8-14, further comprising:

[0150] enabling co-existent use of any two of: the DMVR step, the BDOF step, or the combined intra-inter prediction step.

[0151] 16. A method of visual media processing, comprising:

25 [0152] performing a conversion between a current video block and a bitstream representation of the current video block, wherein the conversion includes a decoder motion vector

refinement (DMVR) step for refining original motion information signaled in the bitstream representation thereby resulting in a refined motion information usable in a de-blocking step; and

[0153] computing, for at least a subset block of the current video block, a difference of the refined motion information and the original motion information.

5 **[0154]** 17. The method of clause 8, wherein dimensions of the subset block are 4x4, 8x8, or 16x16.

[0155] 18. The method of any one or more of clause 16-17, wherein computing the difference includes subtracting the refined motion information from the original motion information.

10 **[0156]** 19. The method of any one or more of clause 16-17, wherein computing the difference includes subtracting the original motion information from the refined motion information.

[0157] 20. The method of clause 16, wherein prior to the de-blocking step, a first motion information is computed as a sum of (i) the difference of the refined motion information and (ii) the original motion information, wherein the first motion information is usable in the de-blocking step.

15 **[0158]** 21. The method of clause 20, wherein the first motion information is multiplied with a negative one.

[0159] 22. The method of clause 16, wherein the difference of the refined motion information and the original motion information is a difference vector, wherein a X-component of the difference vector is greater than a x-lower bound and/or lesser than a x-upper bound.

20 **[0160]** 23. The method of clause 16, wherein the difference of the refined motion information and the original motion information is a difference vector, wherein a Y-component of the difference vector is greater than a y-lower bound and/or lesser than a y-upper bound.

[0161] 24. The method of any one or more of clauses 22-23, wherein the x-upper bound, the x-lower bound, the y-upper bound, and the y-lower bound is signaled in the bitstream representation.

25 **[0162]** 25. The method of any one or more of clauses 22-23, wherein a searching range of the DMVR step is defined in a manner such that (i) the X-component of the difference vector is

greater than the x-lower bound and/or lesser than the x-upper bound and (ii) the Y-component of the difference vector is greater than a y-lower bound and/or lesser than a y-upper bound.

[0163] 26. The method of any one or more of clauses 22-23, wherein the difference vector is clipped in a manner such that the X-component of the difference vector is clipped according to a function Clip3(x-lower bound, x-upper bound, the X-component of the difference vector) and the Y-component of the difference vector is clipped according to a function Clip3(y-lower bound, y-upper bound, the y-component of the difference vector), wherein Clip3(min, max, x) is defined as

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

[0164] 27. The method of any one or more of clauses 22-23, wherein the difference vector is quantized.

[0165] 28. The method of clause 27, wherein the X-component of the difference vector is quantized according to a function Shift(the X-component of the difference vector, a first value) and the Y-component of the difference vector is quantized according to a function Shift(the Y-component of the difference vector, a second value), wherein

$$\text{Shift}(x, n) = (x + \text{offset0}) \gg n.$$

wherein $\text{offset0} = (1 \ll n) \gg 1$, the first value and the second value are scalar quantities.

[0166] 29. The method of clause 27, wherein the X-component of the difference vector is quantized according to a function Shift(the X-component of the difference vector, a first value) and the Y-component of the difference vector is quantized according to a function Shift(the Y-component of the difference vector, a second value), wherein

$$\text{Shift}(x, n) = (x + \text{offset0}) \gg n.$$

wherein $\text{offset0} = (1 \ll (n-1))$, the first value and the second value are scalar quantities.

[0167] 30. The method of clause 27, wherein the X-component of the difference vector is quantized according to a function Shift(the X-component of the difference vector, a first value)

and the Y-component of the difference vector is quantized according to a function Shift(the Y-component of the difference vector, a second value), wherein

$$\text{Shift}(x, n) = (x + \text{offset0}) \gg n.$$

wherein $\text{offset0} = 0$, the first value and the second value are scalar quantities.

- 5 **[0168]** 31. The method of clause 27, wherein the X-component of the difference vector is quantized according to a function SatShift(the X-component of the difference vector, a first value) and the Y-component of the difference vector is quantized according to a function SatShift(the Y-component of the difference vector, a second value), wherein

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

- 10 wherein offset0 , offset1 are set to $(1 \ll n) \gg 1$ or $(1 \ll (n-1))$, and the first value and the second value are scalar quantities.

- [0169]** 32. The method of clause 27, wherein the X-component of the difference vector is quantized according to a function SatShift(the X-component of the difference vector, a first value) and the Y-component of the difference vector is quantized according to a function SatShift(the Y-component of the difference vector, a second value), wherein
- 15

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

wherein offset0 , offset1 are both set to 0, and the first value and the second value are scalar quantities.

- [0170]** 33. The method of any one or more of clauses 28-33, wherein the first value and the second value are signaled in the bitstream representation.
- 20

- [0171]** 34. The method of any one or more of clauses 28-33, wherein the first value and the second value are associated with a searching range of the DMVR step.

[0172] 35. The method of any one or more of clauses 28-33, wherein the first value and the second value are associated with a profile information or a tier information of the current video block.

[0173] 36. The method of any one or more of clauses 1 through 35, wherein the visual media processing is an encoder-side implementation.

[0174] 37. The method of any one or more of clauses 1 through 35, wherein the visual media processing is a decoder-side implementation.

[0175] 38. 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 or more of clauses 1 to 37.

[0176] 39. 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 or more of clauses 1 to 37.

[0177] FIG. 25 is a flowchart for an example method 2500 of video processing. The method 2500 includes, at 2502, deriving, for a conversion between a first block of video and a bitstream representation of the first block of video, initial searching points in decoder-side motion vector refinement (DMVR) process to be applied during the conversion based on one or more motion vectors (MVs) of a merge candidate associated with the first block and one or more offsets; and at 2504, performing the conversion based on the initial searching points.

[0178] In some examples, the initial searching points are derived as the one or more MVs of the merge candidate adding the offsets.

[0179] In some examples, when the one or more MVs of the merge candidate include a first MV (MV0) referring to reference list 0 and a second MV (MV1) referring to reference list 1, the initial searching points are derived as $MV0 + offset0$ and $MV1 + offset1$, $offset0$ being an offset corresponding to the first MV (MV0), and $offset1$ being an offset corresponding to the second MV (MV1).

[0180] In some examples, $offset0$ and/or $offset1$ are predefined.

[0181] In some examples, $\text{offset0} = (4, 0)$ and $\text{offset1} = (-4, 0)$.

[0182] In some examples, offset0 and/or offset1 are signaled from encoder to decoder.

[0183] In some examples, the offsets are signaled from the encoder to the decoder with Merge with motion vector differences (MMVD) mode.

5 [0184] In some examples, the DMVR process is applied when the first block is coded as MMVD and/or MMVD skip mode.

[0185] In some examples, when the first block is coded with MMVD and/or MMVD skip mode, the initial searching points in the DMVR process are set to be the MVs with the MMVD mode, which are derived as the MVs of the merge candidate adding the signaled offsets.

10 [0186] FIG. 26 is a flowchart for an example method 2600 of video processing. The method 2600 includes, at 2602, determining, for a conversion between a first block of video and a bitstream representation of the first block of video, that at least one of decoder motion vector refinement (DMVR) process, Bi-directional Optical flow (BDOF) process and combined intra-inter prediction process is disabled based on a predetermined rule; and at 2604, performing the conversion based on
15 the determination.

[0187] In some examples, when dimensions of the first block satisfy one or more conditions, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, dimensions of the first block including at least one of a width W , a height H or $W \times H$.

20 [0188] In some examples, when $W \geq T1$ and $H \geq T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.

[0189] In some examples, when $W > T1$ and $H > T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.

[0190] In some examples, when $W \leq T1$ and $H \leq T2$, at least one of the DMVR process, the
25 BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.

[0191] In some examples, when $W < T1$ and $H < T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.

[0192] In some examples, when $W \geq T1$ or $H \geq T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 and T2 are integers.

[0193] In some examples, when $W > T1$ or $H > T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 and T2 are integers.

5 [0194] In some examples, when $W \leq T1$ or $H \leq T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 and T2 are integers.

[0195] In some examples, when $W < T1$ or $H < T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 and T2 are integers.

[0196] In some examples, $T1=T2=16$.

10 [0197] In some examples, $T1=T2=8$.

[0198] In some examples, $T1=T2=128$.

[0199] In some examples, when $W \times H \geq T1$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 is an integer.

15 [0200] In some examples, when $W \times H > T1$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 is an integer.

[0201] In some examples, when $W \times H \leq T1$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 is an integer.

[0202] In some examples, when $W \times H < T1$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 is an integer.

20 [0203] In some examples, $T1=64$.

[0204] In some examples, $T1=128$.

[0205] In some examples, when both the BDOF process and the inter-intra prediction process are applied, the DMVR process is disabled.

25 [0206] In some examples, when the inter-intra prediction process is applied, the DMVR process is disabled.

[0207] In some examples, when both the DMVR process and the inter-intra prediction process are applied, the BDOF process is disabled.

[0208] In some examples, when the inter-intra prediction process is applied, the BDOF process is disabled.

5 [0209] In some examples, when both the DMVR process and the BDOF process are applied, the inter-intra prediction process is disabled.

[0210] FIG. 27 is a flowchart for an example method 2700 of video processing. The method 2700 includes, at 2702, deriving, for a conversion between a first block of video and a bitstream representation of the first block of video, motion vectors (MVs) associated with the first block, the
10 MVs being refined by applying decoder-side motion vector refinement (DMVR) process; and at 2704, performing the conversion by using the refined MVs in de-blocking process.

[0211] FIG. 28 is a flowchart for an example method 2800 of video processing. The method 2800 includes, at 2802, calculating, for a conversion between a first block of video and a bitstream representation of the first block of video, motion vector (MV) difference (dMV) between refined
15 MV (rMV) and non-refined MV (nMV) associated with each basic block of the first block, the rMV being a motion vector refined by applying decoder-side motion vector refinement (DMVR) process, the nMV being a motion vector not refined by the DMVR process; and at 2804 performing the conversion by using the calculated MV difference.

[0212] In some examples, the basic block has a width w and a height h , where $w=h=4$, or
20 $w=h=8$, or $w=h=16$.

[0213] In some examples, the MV difference dMV is derived as $dMV=rMV-nMV$ or $dMV=nMV-rMV$.

[0214] In some examples, refined MV before deblocking process (rMV') is calculated as $rMV'=dMV+nMV$, and is to be used as the MV for the basic block in the following deblocking
25 process and temporal prediction process.

[0215] In some examples, refined MV before deblocking process (rMV') is calculated as $rMV' = -(dMV + nMV)$, and is to be used as the MV for the basic block in the following deblocking process and temporal prediction process.

[0216] In some examples, the MV difference dMV has a horizontal component (dMV_x) and a vertical component (dMV_y), dMV_x and dMV_y being in a range in a conforming bitstream.

[0217] In some examples, dMV_x and dMV_y satisfy $T1_x \leq dMV_x \leq T2_x$ and $T1_y \leq dMV_y \leq T2_y$, $T1_x$, $T2_x$, $T1_y$ and $T2_y$ being integers.

[0218] In some examples, $T1_x = T1_y = -2^K$ and $T2_x = T2_y = 2^K - 1$, where K is an integer.

[0219] In some examples, K is 3 or 4.

[0220] In some examples, the searching range in the DMVR process guarantees that the MV difference dMV can satisfy the constrain.

[0221] In some examples, one or more of $T1_x$, $T2_x$, $T1_y$, $T2_y$ and K are signaled from the encoder to the decoder.

[0222] In some examples, one or more of $T1_x$, $T2_x$, $T1_y$, $T2_y$ and K depend on the searching range of the DMVR process.

[0223] In some examples, one or more of $T1_x$, $T2_x$, $T1_y$, $T2_y$ and K depend on the standard profile and/or level and/or tier.

[0224] In some examples, the motion vector difference dMV is clipped with a function $Clip3(\text{Min}, \text{Max}, x)$, the function $Clip3(\text{Min}, \text{Max}, x)$ being defined as

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

[0225] In some examples, a horizontal component dMV_x of dMV is set to be $Clip3(T1_x, T2_x, dMV_x)$ and a vertical component dMV_y of dMV is set to be $Clip3(T1_y, T2_y, dMV_y)$.

[0226] In some examples, $T1_x = T1_y = -2^K$ and $T2_x = T2_y = 2^K - 1$, where K is an integer.

[0227]

[0228] In some examples, K is 3 or 4.

[0229] In some examples, the MV difference dMV is stored after the conversion.

[0230] In some examples, the MV difference dMV is quantized to be dMV', and dMV' is stored.

- 5 [0231] In some examples, a horizontal component dMV_x' of dMV' is set to be Shift(dMV_x, N_x) and a vertical component dMV_y' of dMV' is set to be Shift(dMV_y, N_y), N_x and N_y being integers, where Shift(x, n) is defined as:

$$\text{Shift}(x, n) = (x + \text{offset0}) \gg n,$$

where offset0 is set to $(1 \ll n) \gg 1$ or $(1 \ll (n-1))$, or offset0 is set to 0.

- 10 [0232] In some examples, a horizontal component dMV_x' of dMV' is set to be SatShift(dMV_x, N_x) and a vertical component dMV_y' of dMV' is set to be SatShift(dMV_y, N_y), N_x and N_y being integers, where SatShift(x, n) is defined as:

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

where offset0 and/or offset1 is set to $(1 \ll n) \gg 1$ or $(1 \ll (n-1))$, or offset0 and/or offset1 is set to 0.

- 15 [0233] In some examples, N_x = N_y = 1.

[0234] In some examples, N_x and/or N_y are signaled from the encoder to the decoder.

[0235] In some examples, N_x and/or N_y depend on the searching range of the DMVR process.

[0236] In some examples, N_x and/or N_y depend on the standard profile and/or level and/or tier.

- 20 [0237] In some examples, the MV difference dMV is dequantized from dMV' before deriving rMV'.

[0238] In some examples, dMV_x = dMV_x' << N_x and dMV_y = dMV_y' << N_y.

[0239] In some examples, the MV difference dMV is clipped before being quantized.

[0240]

[0241] In some examples, the MV difference dMV is clipped after being quantized.

[0242] In some examples, the refined MV before deblocking process rMV' is used in the motion compensation procedure.

[0243] In some examples, the conversion generates the first block of video from the bitstream representation.

[0244] In some examples, the conversion generates the bitstream representation from the first block of video.

[0245] 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.

[0246] 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
5 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.

[0247] 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
10 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).

[0248] 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
15 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
20 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.

[0249] While this patent document contains many specifics, these should not be construed as
25 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
5 combination may be directed to a subcombination or variation of a subcombination.

[0250] 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
10 this patent document should not be understood as requiring such separation in all embodiments.

[0251] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this patent document.

CLAIMS

1. A method of video processing, comprising:
deriving, for a conversion between a first block of video and a bitstream representation of
5 the first block of video, initial searching points in decoder-side motion vector refinement (DMVR)
process to be applied during the conversion based on one or more motion vectors (MVs) of a merge
candidate associated with the first block and one or more offsets; and
performing the conversion based on the initial searching points.
- 10 2. The method of claim 1, wherein the initial searching points are derived as the one or more
MVs of the merge candidate adding the offsets.
3. The method of any of claims 1-2, wherein when the one or more MVs of the merge
candidate include a first MV (MV0) referring to reference list 0 and a second MV (MV1) referring to
15 reference list 1, the initial searching points are derived as $MV0 + offset0$ and $MV1 + offset1$, $offset0$
being an offset corresponding to the first MV (MV0), and $offset1$ being an offset corresponding to
the second MV (MV1).
4. The method of claim 3, wherein $offset0$ and/or $offset1$ are predefined.
- 20 5. The method of claim 4, wherein $offset0 = (4, 0)$ and $offset1 = (-4, 0)$.
6. The method of claim 3, wherein $offset0$ and/or $offset1$ are signaled from encoder to decoder.
- 25 7. The method of any of claims 1-2, wherein the offsets are signaled from the encoder to the
decoder with Merge with motion vector differences (MMVD) mode.
8. The method of claim 7, wherein the DMVR process is applied when the first block is coded
as MMVD and/or MMVD skip mode.

30

9. The method of claim 8, wherein when the first block is coded with MMVD and/or MMVD skip mode, the initial searching points in the DMVR process are set to be the MVs with the MMVD mode, which are derived as the MVs of the merge candidate adding the signaled offsets.

5 10. A method of video processing, comprising:

determining, for a conversion between a first block of video and a bitstream representation of the first block of video, that at least one of decoder motion vector refinement (DMVR) process, Bi-directional Optical flow (BDOF) process and combined intra-inter prediction process is disabled based on a predetermined rule; and

10 performing the conversion based on the determination.

11. The method of claim 10, wherein when dimensions of the first block satisfy one or more conditions, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, dimensions of the first block including at least one of a width W , a
15 height H or $W \times H$.

12. The method of claim 11, wherein when $W \geq T1$ and $H \geq T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.

20 13. The method of claim 11, wherein when $W > T1$ and $H > T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.

25 14. The method of claim 11, wherein when $W \leq T1$ and $H \leq T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.

15. The method of claim 11, wherein when $W < T1$ and $H < T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.
- 5 16. The method of claim 11, wherein when $W \geq T1$ or $H \geq T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.
- 10 17. The method of claim 11, wherein when $W > T1$ or $H > T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.
- 15 18. The method of claim 11, wherein when $W \leq T1$ or $H \leq T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.
- 20 19. The method of claim 11, wherein when $W < T1$ or $H < T2$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ and $T2$ are integers.
- 20 20. The method of any of claims 12-19, wherein $T1=T2=16$.
21. The method of any of claims 12-19, wherein $T1=T2=8$.
- 25 22. The method of any of claims 12-19, wherein $T1=T2=128$.
23. The method of claim 11, wherein when $W \times H \geq T1$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, $T1$ is an integer.

24. The method of claim 11, wherein when $W \times H > T1$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 is an integer.
25. The method of claim 11, wherein when $W \times H \leq T1$, at least one of the DMVR process, the
5 BDOF process and the combined intra-inter prediction process is disabled, T1 is an integer.
26. The method of claim 11, wherein when $W \times H < T1$, at least one of the DMVR process, the BDOF process and the combined intra-inter prediction process is disabled, T1 is an integer.
- 10 27. The method of any of claims 21-24, wherein $T1=64$.
28. The method of any of claims 21-24, wherein $T1=128$.
29. The method of claim 10, wherein when both the BDOF process and the inter-intra prediction
15 process are applied, the DMVR process is disabled.
30. The method of claim 10, wherein when the inter-intra prediction process is applied, the DMVR process is disabled.
- 20 31. The method of claim 10, wherein when both the DMVR process and the inter-intra prediction process are applied, the BDOF process is disabled.
32. The method of claim 10, wherein when the inter-intra prediction process is applied, the BDOF process is disabled.
- 25 33. The method of claim 10, wherein when both the DMVR process and the BDOF process are applied, the inter-intra prediction process is disabled.
34. A method of video processing, comprising:

deriving, for a conversion between a first block of video and a bitstream representation of the first block of video, motion vectors (MVs) associated with the first block, the MVs being refined by applying decoder-side motion vector refinement (DMVR) process; and performing the conversion by using the refined MVs in de-blocking process.

5

35. A method of video processing, comprising:

calculating, for a conversion between a first block of video and a bitstream representation of the first block of video, motion vector (MV) difference (dMV) between refined MV (rMV) and non-refined MV (nMV) associated with each basic block of the first block, the rMV being a motion vector refined by applying decoder-side motion vector refinement (DMVR) process, the nMV being a motion vector not refined by the DMVR process; and performing the conversion by using the calculated MV difference.

10

15 36. The method of claim 35, wherein the basic block has a width w and a height h , where $w=h=4$, or $w=h=8$, or $w=h=16$.

37. The method of claim 35 or 36, wherein the MV difference dMV is derived as $dMV=rMV-nMV$ or $dMV=nMV-rMV$.

20

38. The method of any of claims 35-37, wherein refined MV before deblocking process (rMV') is calculated as $rMV'=dMV+nMV$, and is to be used as the MV for the basic block in the following deblocking process and temporal prediction process.

25 39. The method of any of claims 35-37, wherein refined MV before deblocking process (rMV') is calculated as $rMV'=(dMV+nMV)$, and is to be used as the MV for the basic block in the following deblocking process and temporal prediction process.

40. The method of any of claims 35-39, wherein the MV difference dMV has a horizontal component (dMV_x) and a vertical component (dMV_y), dMV_x and dMV_y being in a range in a conforming bitstream.

5 41. The method of claim 40, wherein dMV_x and dMV_y satisfy $T1x \leq dMVx \leq T2x$ and $T1y \leq dMvy \leq T2y$, T1x, T2x, T1y and T2y being integers.

42. The method of claim 41, wherein $T1x=T1y=-2^K$ and $T2x=T2y=2^K-1$, where K is an integer.

10 43. The method of claim 42, wherein K is 3 or 4.

44. The method of any of claims 39-43, wherein the searching range in the DMVR process guarantees that the MV difference dMV can satisfy the constrain.

15 45. The method of any of claims 40-44, wherein one or more of T1x, T2x, T1y, T2y and K are signaled from the encoder to the decoder.

46. The method of any of claims 40-44, wherein one or more of T1x, T2x, T1y, T2y and K depend on the searching range of the DMVR process.

20 47. The method of any of claims 40-44, wherein one or more of T1x, T2x, T1y, T2y and K depend on the standard profile and/or level and/or tier.

25 48. The method of any of claims 40-47, wherein the motion vector difference dMV is clipped with a function $Clip3(\text{Min}, \text{Max}, x)$, the function $Clip3(\text{Min}, \text{Max}, x)$ being defined as

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

49. The method of claim 48, wherein a horizontal component dMV_x of dMV is set to be $\text{Clip3}(T1_x, T2_x, dMV_x)$ and a vertical component dMV_y of dMV is set to be $\text{Clip3}(T1_y, T2_y, dMV_y)$.

50. The method of claim 49, wherein $T1_x=T1_y=-2^K$ and $T2_x=T2_y=2^K-1$, where K is an integer.

51. The method of claim 50, wherein K is 3 or 4.

52. The method of any of claim 35-51, wherein the MV difference dMV is stored after the conversion.

53. The method of any of claims 35 - 52, wherein the MV difference dMV is quantized to be dMV' , and dMV' is stored.

54. The method of claim 53, wherein a horizontal component dMV_x' of dMV' is set to be $\text{Shift}(dMV_x, N_x)$ and a vertical component dMV_y' of dMV' is set to be $\text{Shift}(dMV_y, N_y)$, N_x and N_y being integers, where $\text{Shift}(x, n)$ is defined as:

$$\text{Shift}(x, n) = (x + \text{offset0}) \gg n,$$

where offset0 is set to $(1 \ll n) \gg 1$ or $(1 \ll (n-1))$, or offset0 is set to 0.

55. The method of claim 53, wherein a horizontal component dMV_x' of dMV' is set to be $\text{SatShift}(dMV_x, N_x)$ and a vertical component dMV_y' of dMV' is set to be $\text{SatShift}(dMV_y, N_y)$, N_x and N_y being integers, where $\text{SatShift}(x, n)$ is defined as:

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

where offset0 and/or offset1 is set to $(1 \ll n) \gg 1$ or $(1 \ll (n-1))$, or offset0 and/or offset1 is set to 0.

56. The method of any of claims 54 - 55, wherein $N_x = N_y = 1$.

57. The method of any of claims 54 - 55, wherein N_x and/or N_y are signaled from the encoder to the decoder.

58. The method of any of claims 54 - 55, wherein N_x and/or N_y depend on the searching range of the DMVR process.

59. The method of any of claims 54 - 55, wherein N_x and/or N_y depend on the standard profile and/or level and/or tier.

60. The method of any of claims 53 - 59, wherein the MV difference dMV is dequantized from dMV' before deriving rMV' .

61. The method of claim 60, wherein $dMV_x = dMV'_x \ll N_x$ and $dMV_y = dMV'_y \ll N_y$.

62. The method of any of claims 53 - 61, wherein the MV difference dMV is clipped before being quantized.

63. The method of any of claims 53 - 61, wherein the MV difference dMV is clipped after being quantized.

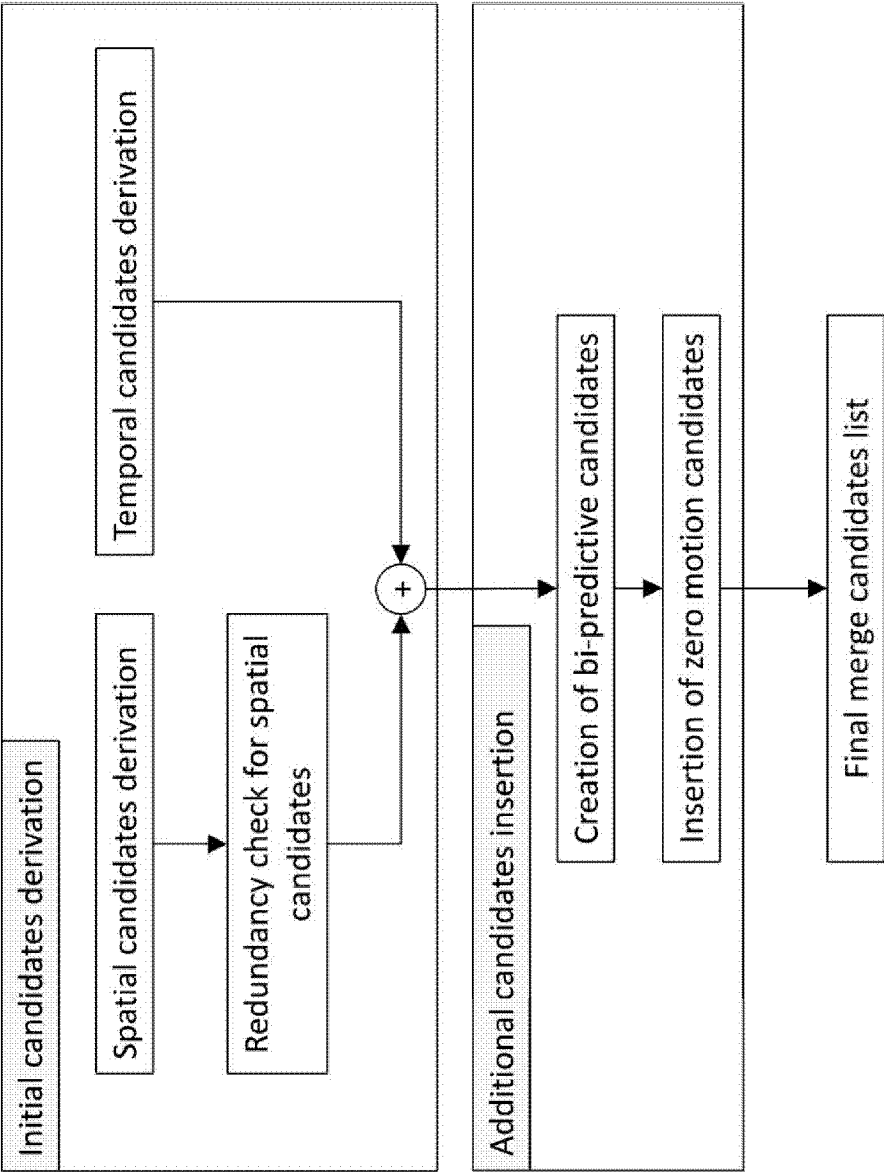
64. The method of claim 39, wherein the refined MV before deblocking process rMV' is used in the motion compensation procedure.

65. The method of anyone of claims 1-64, wherein the conversion generates the first block of video from the bitstream representation.

66. The method of anyone of claims 1-64, wherein the conversion generates the bitstream representation from the first block of video.

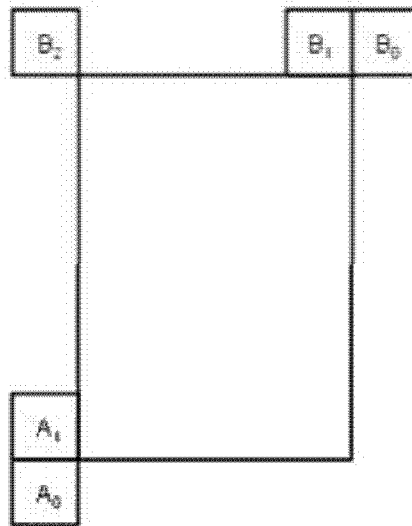
67. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of claims 1 to 66.

- 5 68. 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 66.



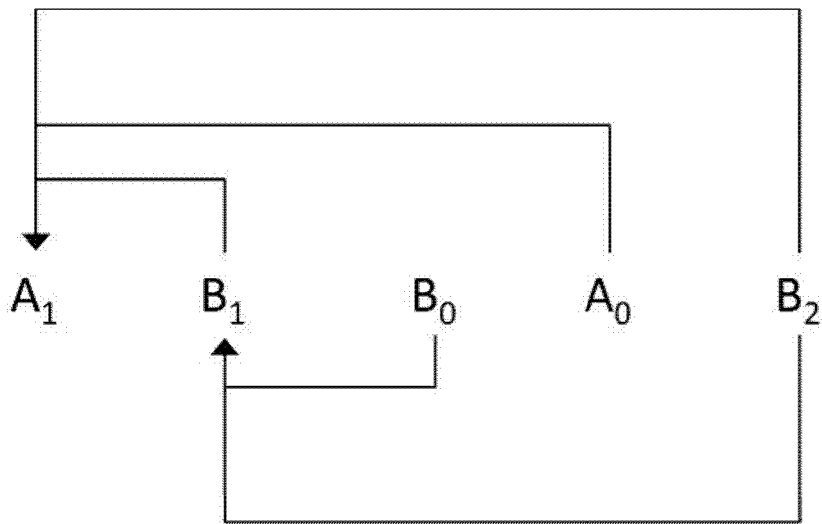
Derivation process for merge candidates list construction

FIG. 1



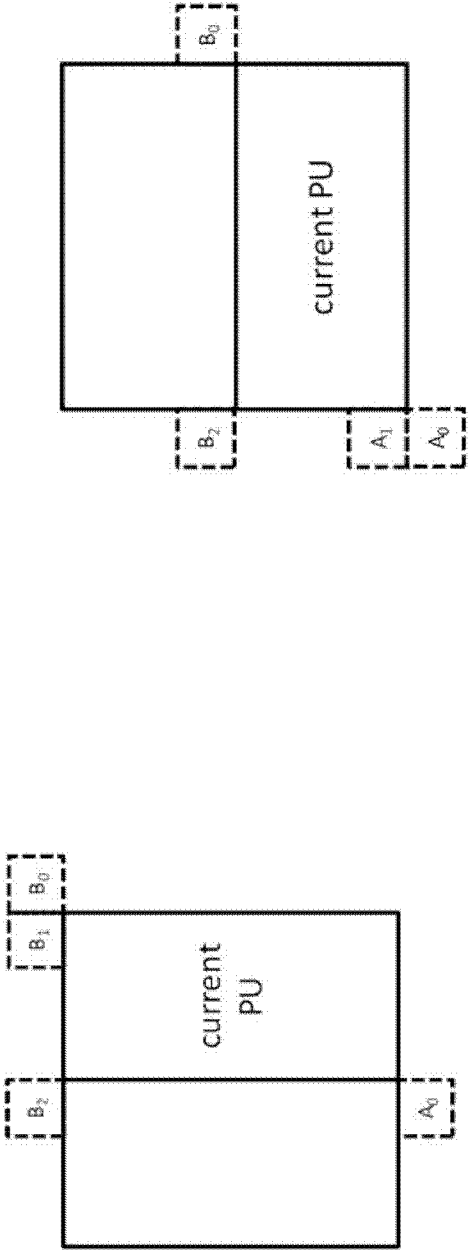
Positions of spatial merge candidates

FIG. 2



Candidate pairs considered for redundancy check of spatial merge candidates

FIG. 3



Positions for the second PU of $N \times 2N$ and $2N \times N$ partitions
second PU of $N \times 2N$ second PU of $2N \times N$

FIG. 4A

FIG. 4B

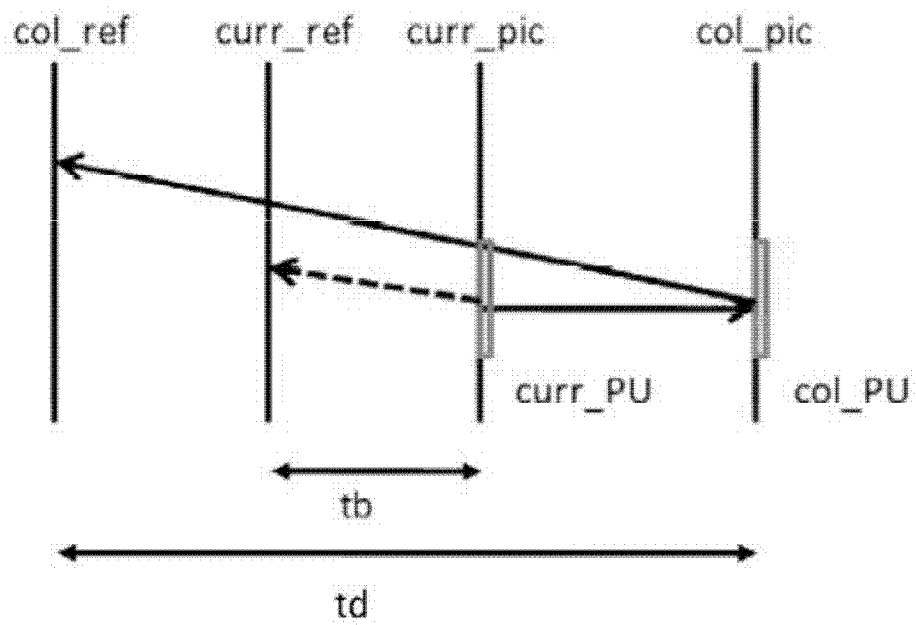
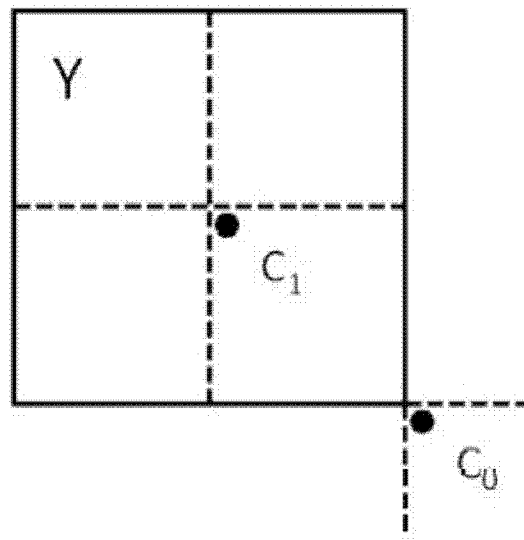


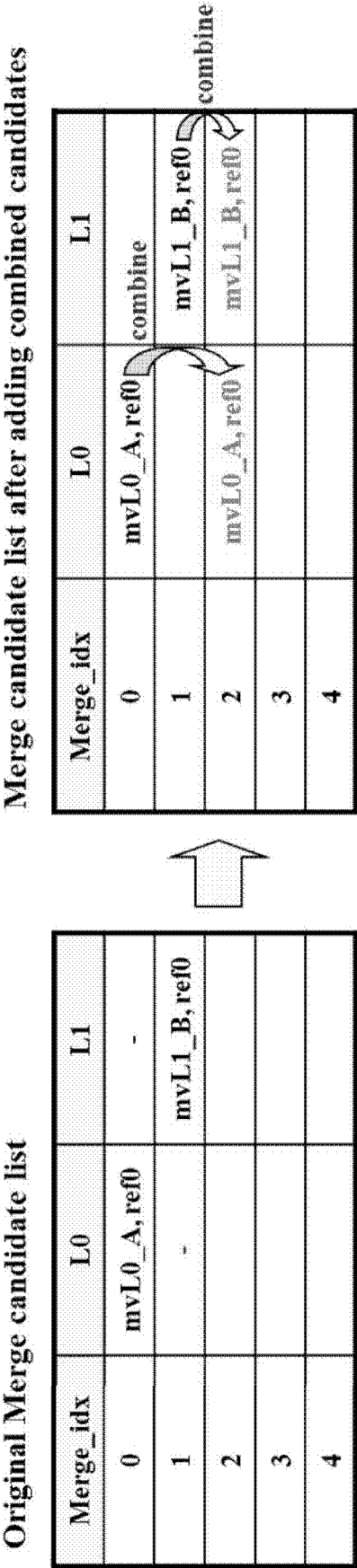
Illustration of motion vector scaling for temporal merge candidate

FIG. 5



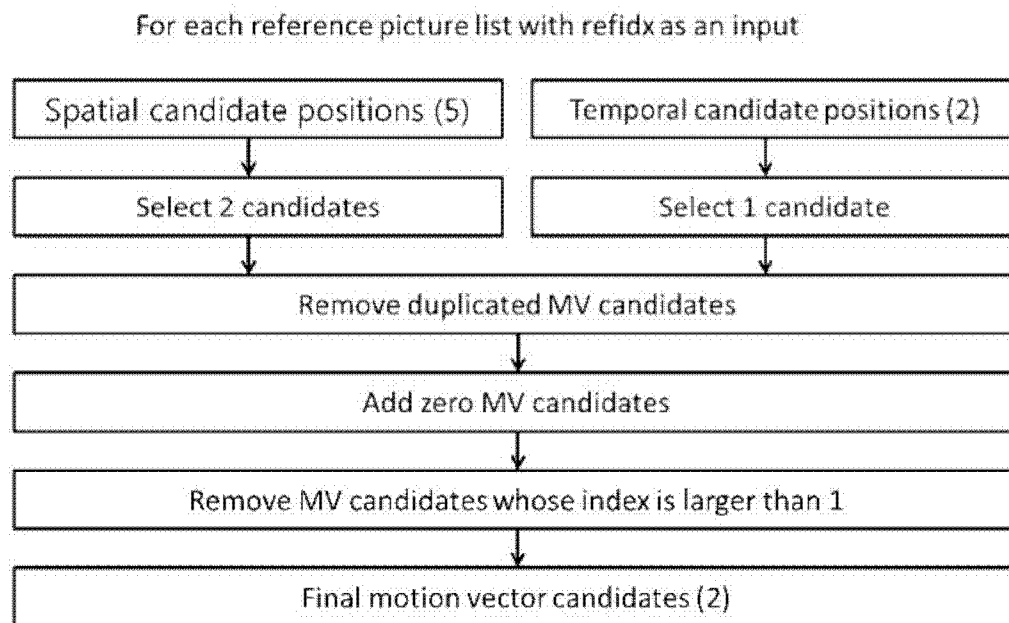
Candidate positions for temporal merge candidate, C0 and C1

FIG. 6



Example of combined bi-predictive merge candidate

FIG. 7



Derivation process for motion vector prediction candidates

FIG. 8

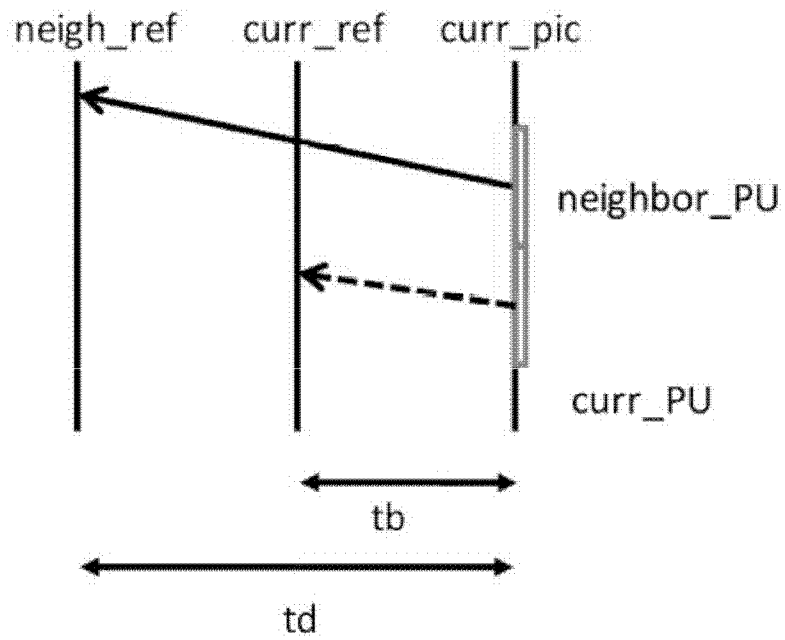
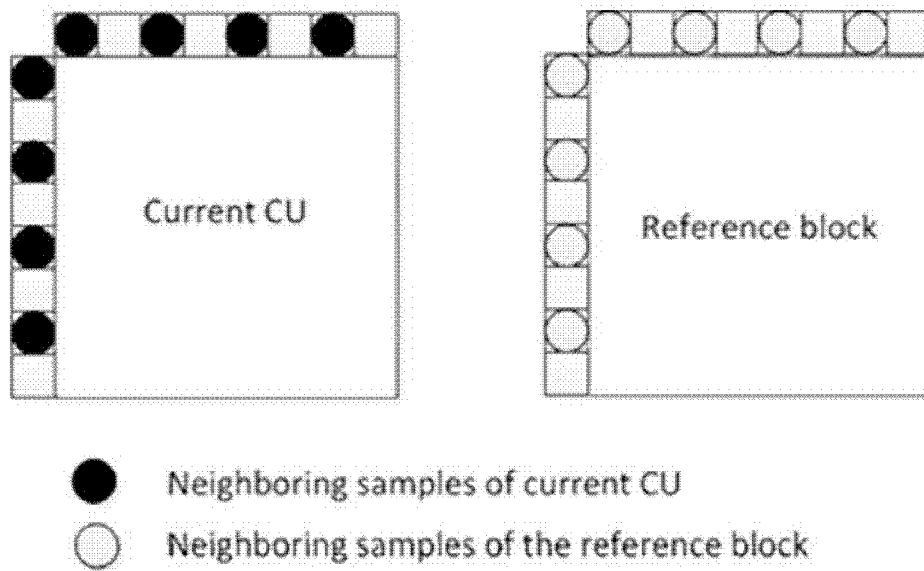


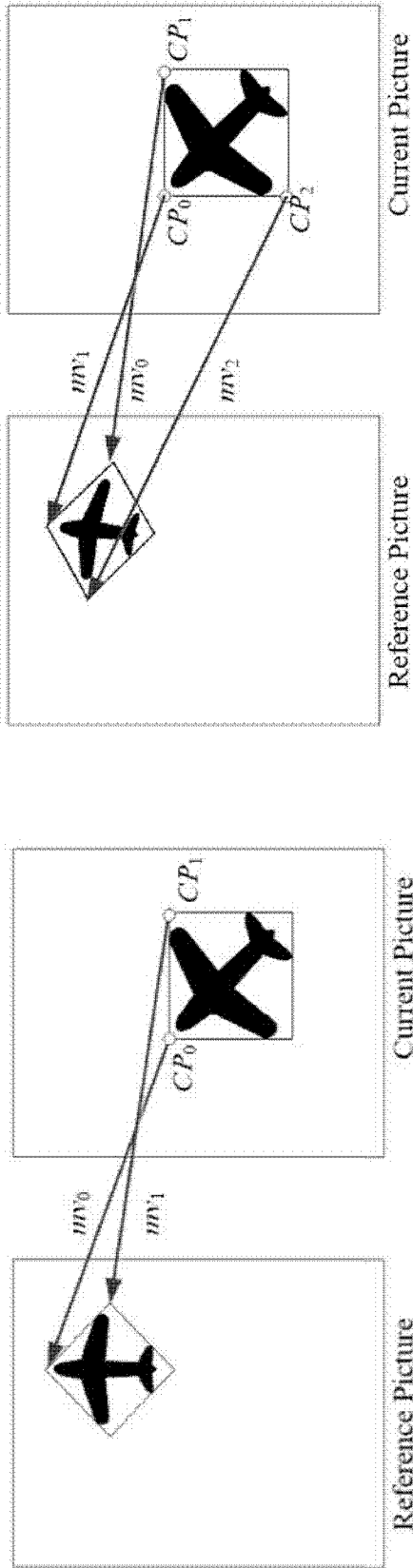
Illustration of motion vector scaling for spatial motion vector candidate

FIG. 9



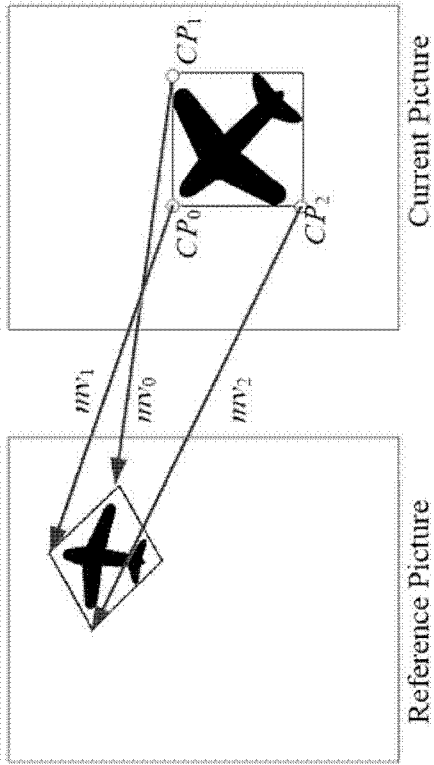
Neighbouring samples used for deriving IC parameters

FIG. 10



4-paramter affine

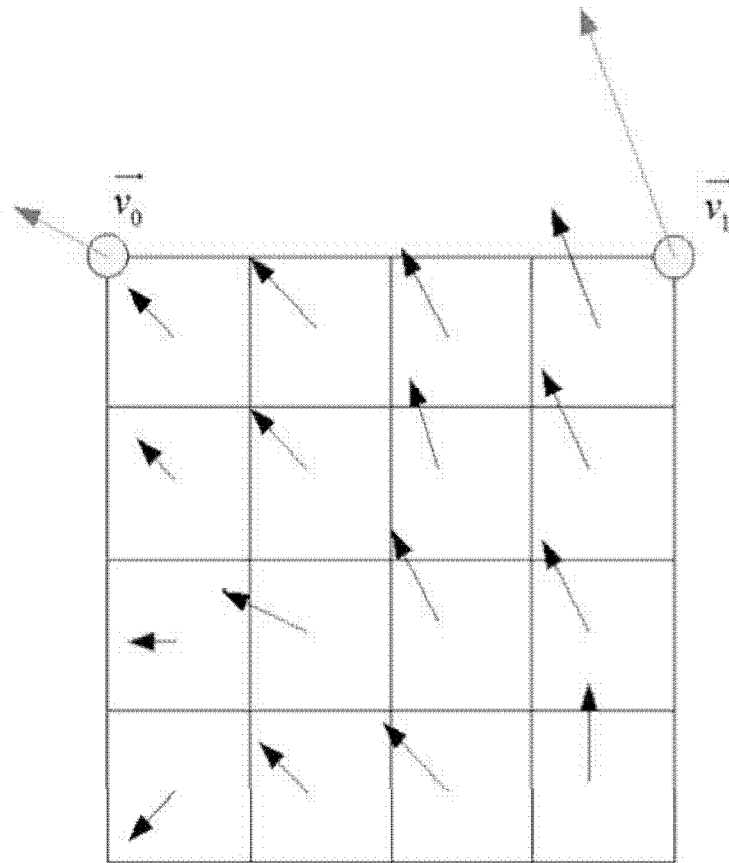
FIG. 11A



6-paramaeter affine

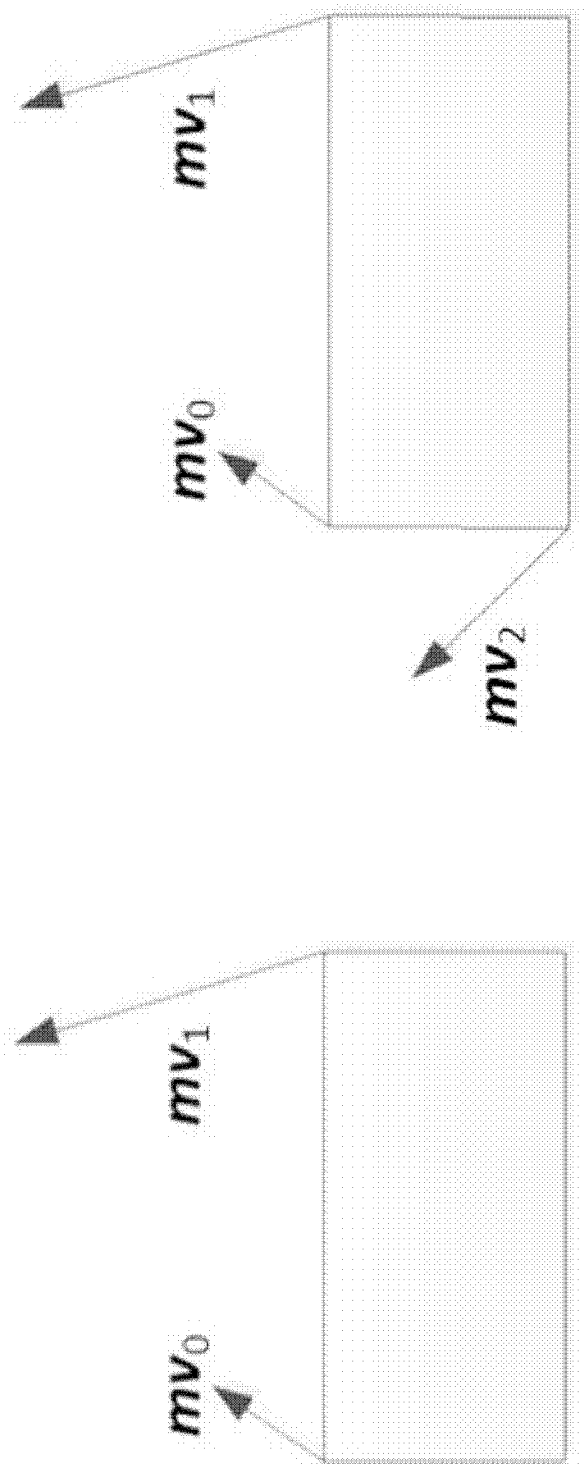
FIG. 11B

Simplified affine motion model



Example of Affine MVF per sub-block

FIG. 12

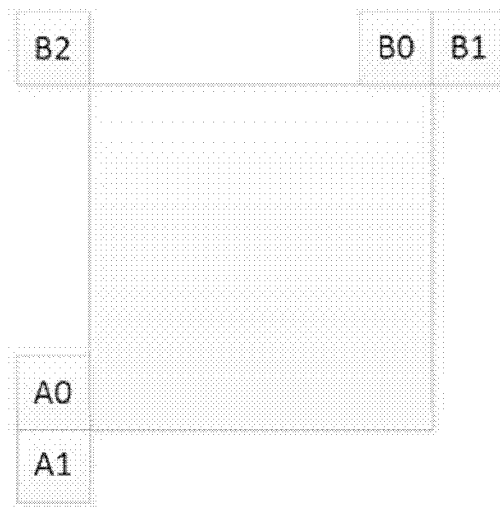


6-parameter affine model

FIG. 13B

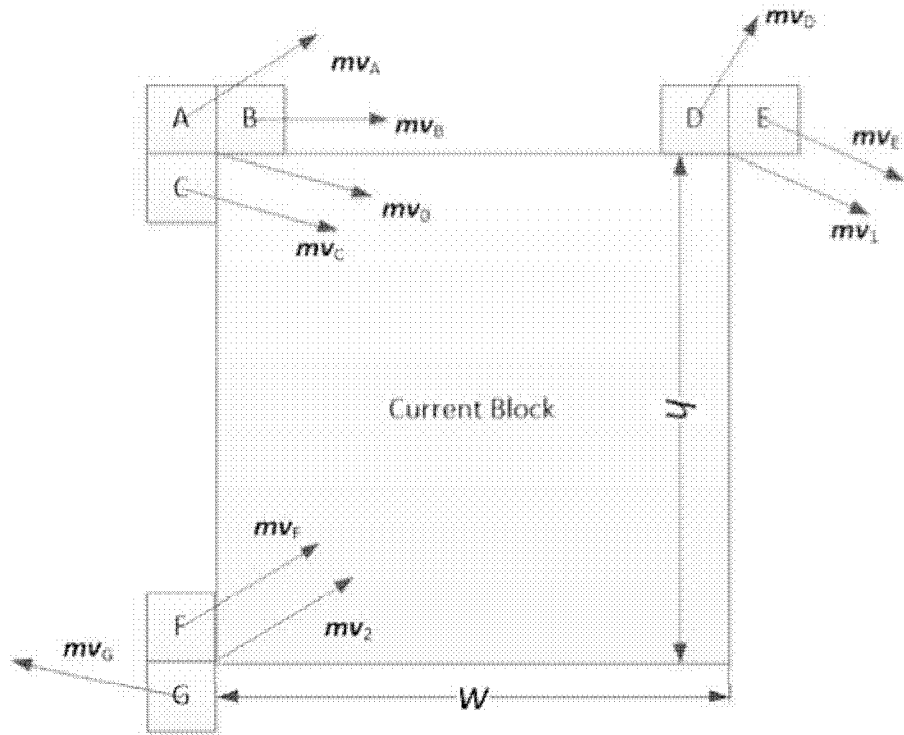
4-parameter affine model

FIG. 13A



MVP for AF_INTER for inherited affine candidates

FIG. 14



MVP for AF_INTER for constructed affine candidates

FIG. 15

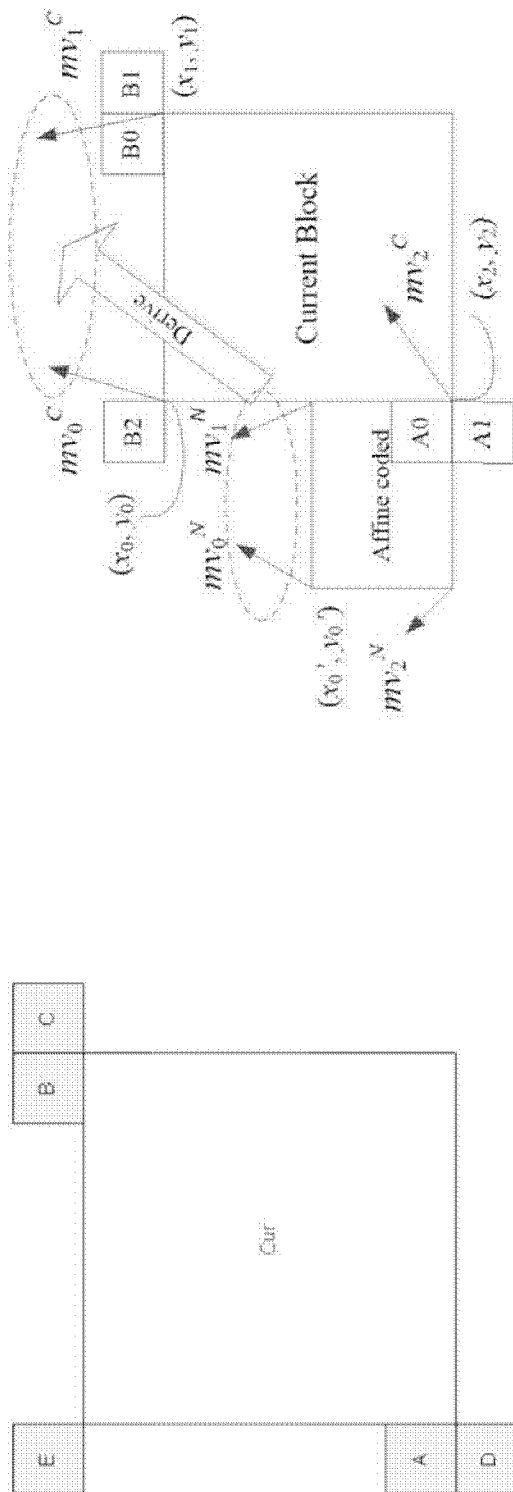
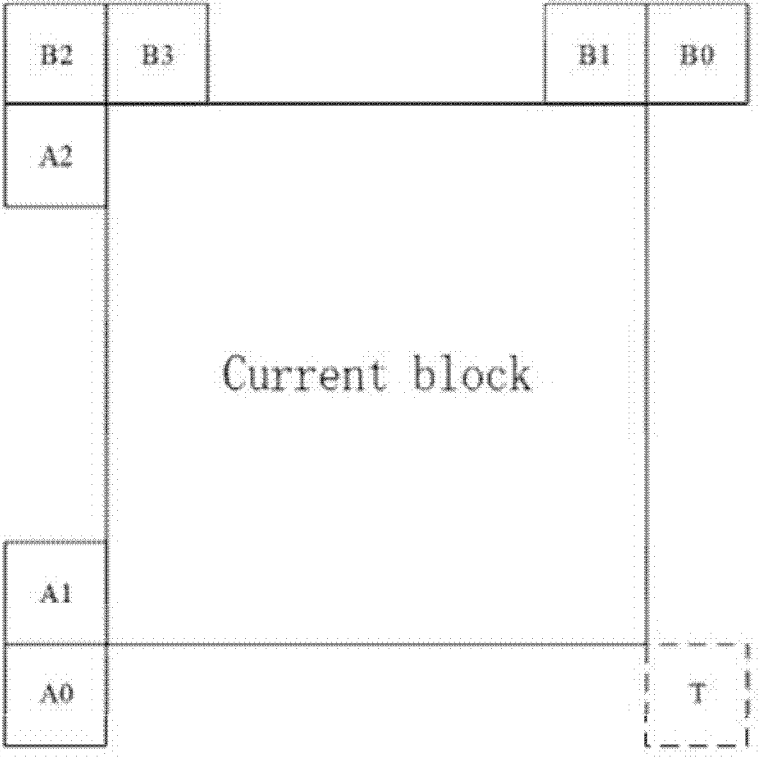


FIG. 16B

FIG. 16A



Candidates position for affine merge mode

FIG. 17

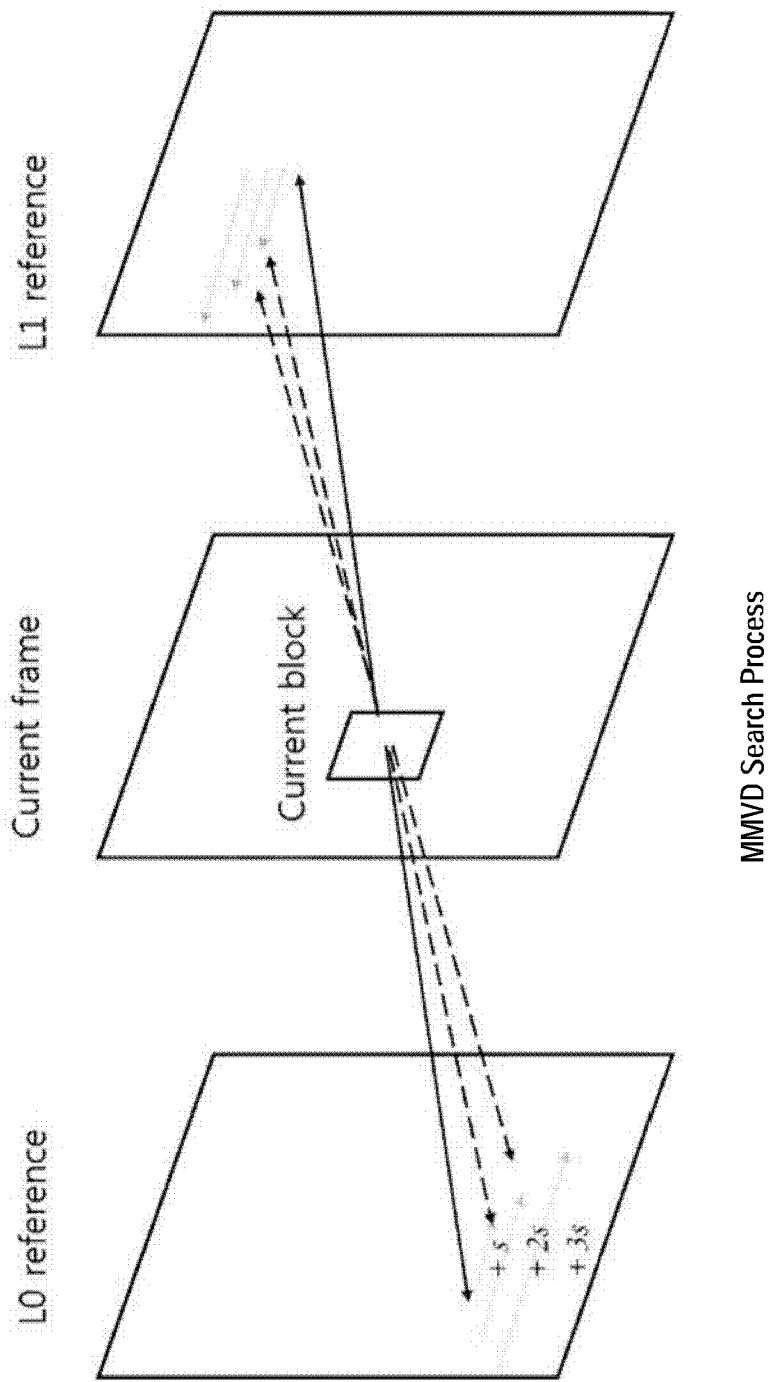


FIG. 18

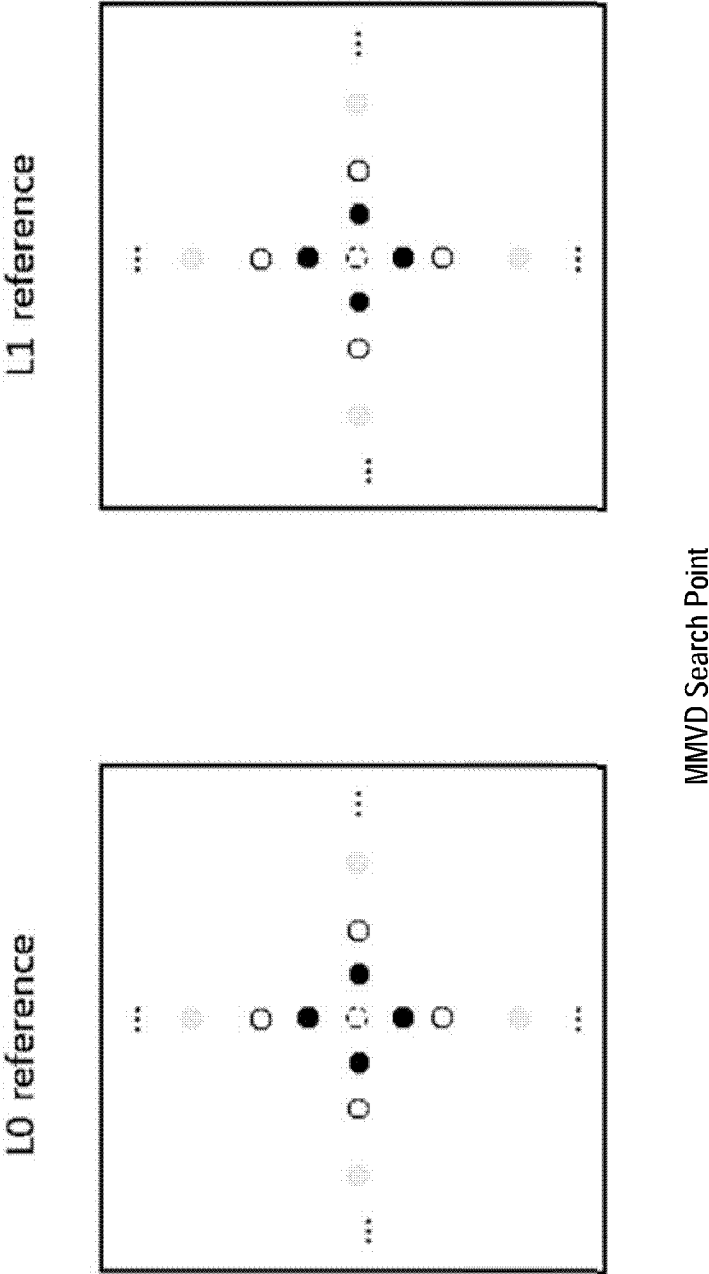
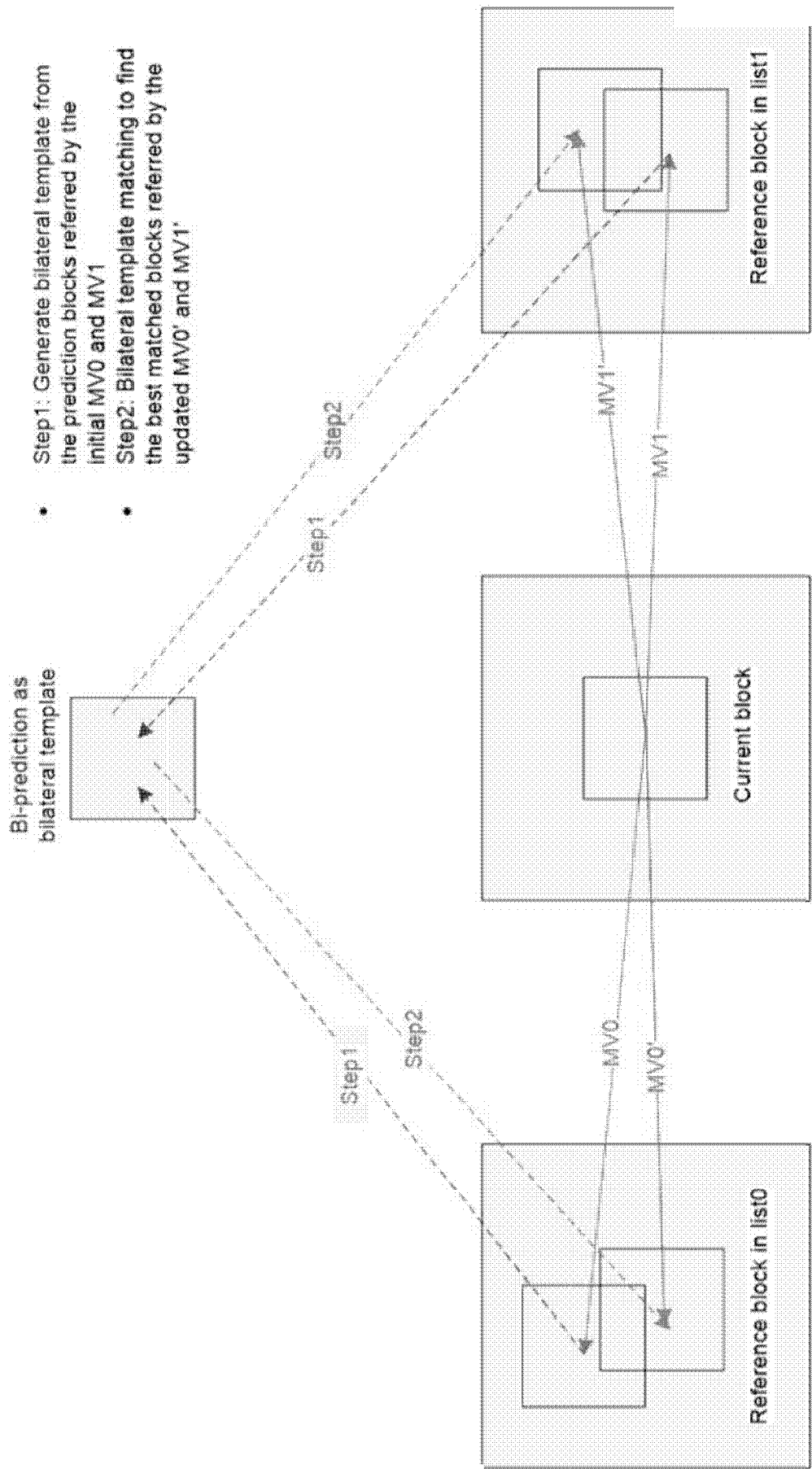
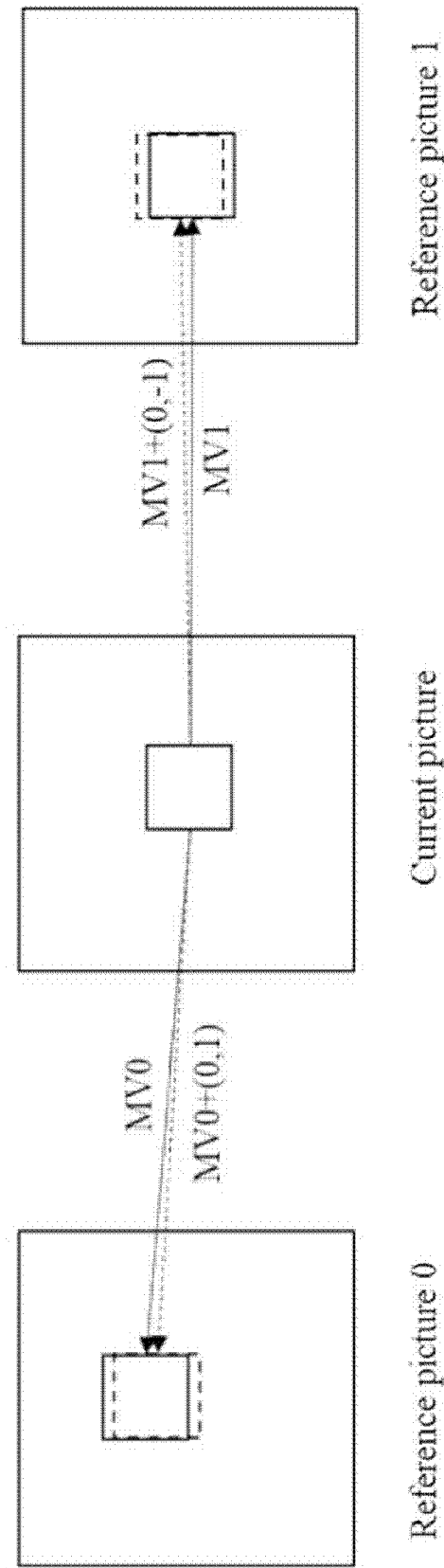


FIG. 19



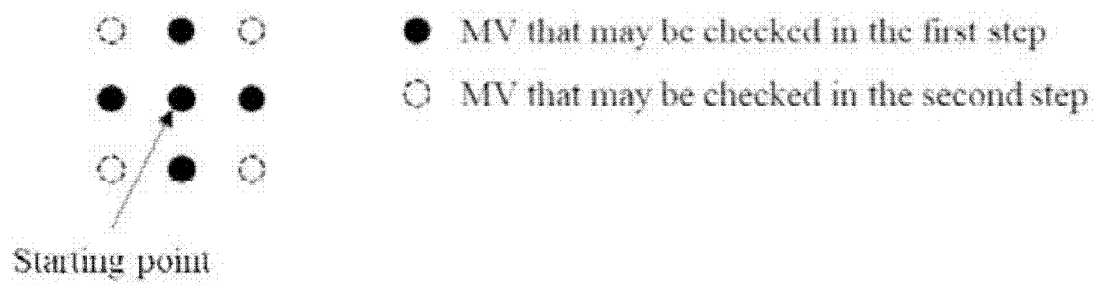
DMVR based on bilateral template matching

FIG. 20



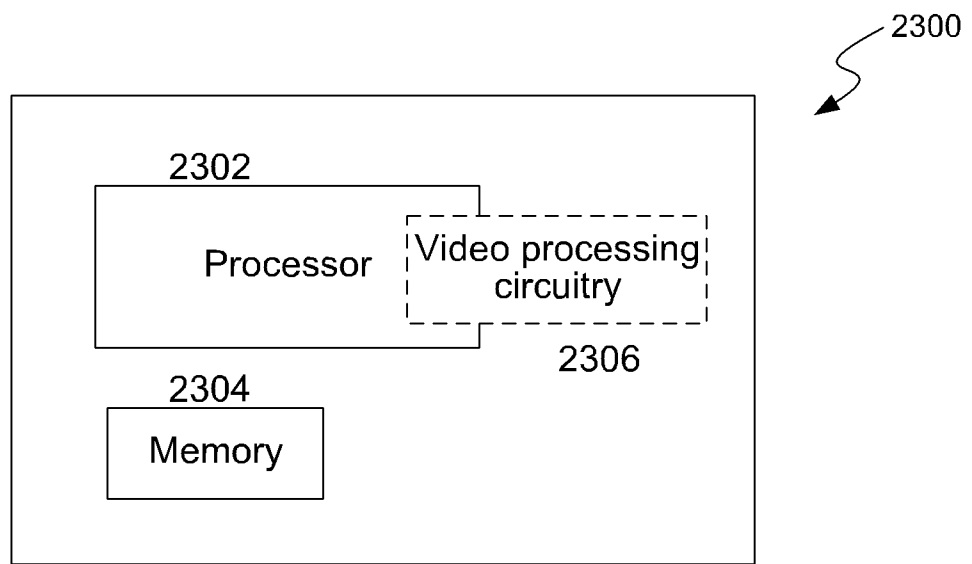
an example of MVD (0, 1) mirrored between list 0 and list 1 in DMVR

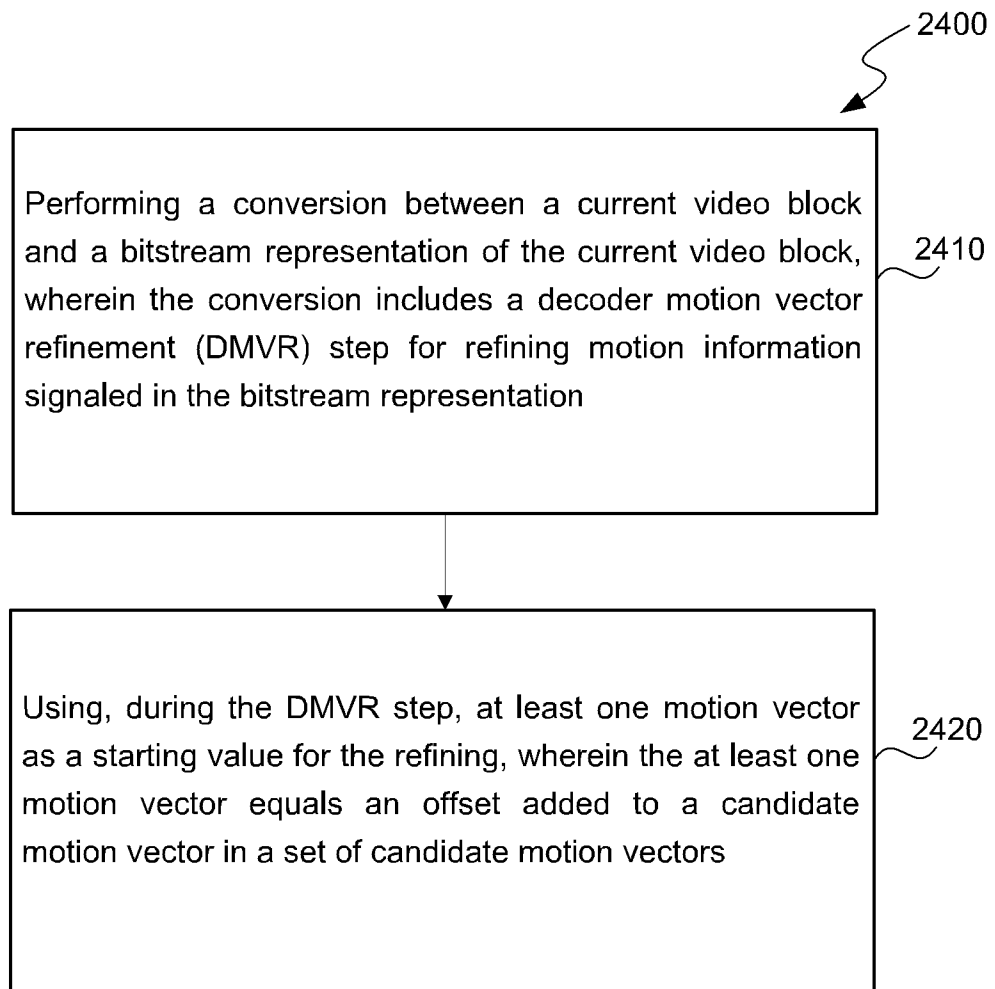
FIG. 21



MVs that may be checked in one iteration

FIG. 22

**FIG. 23**

**FIG. 24**

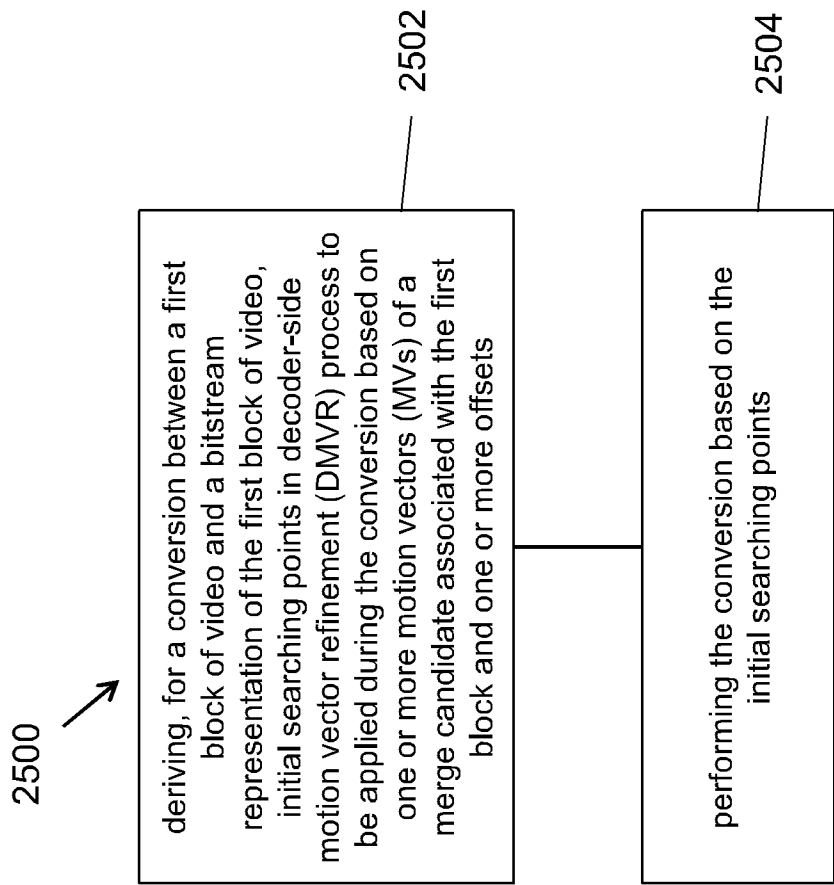


FIG. 25

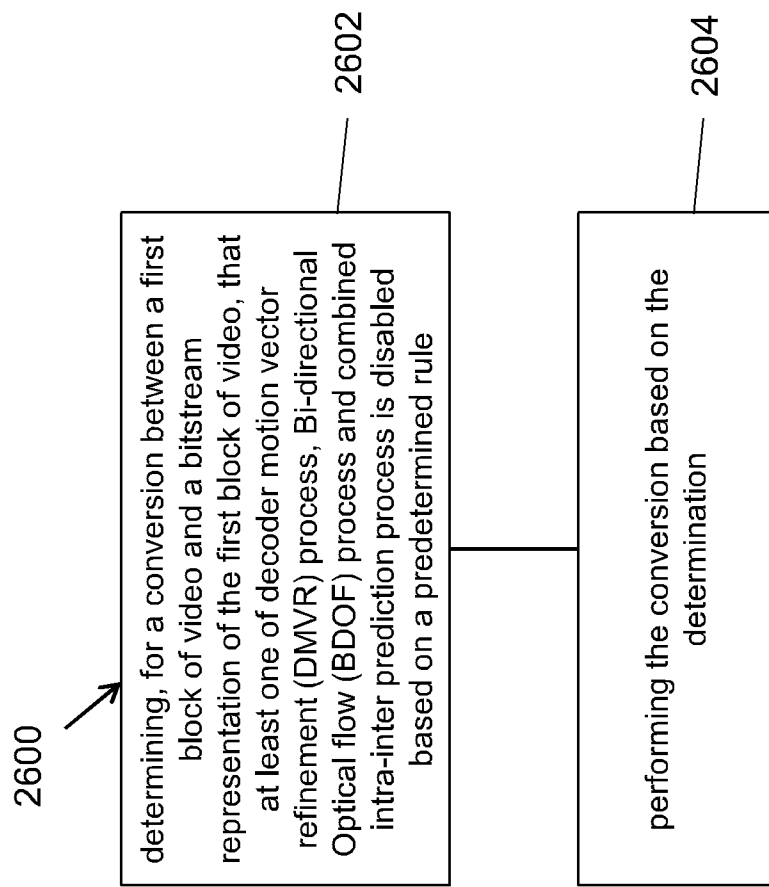
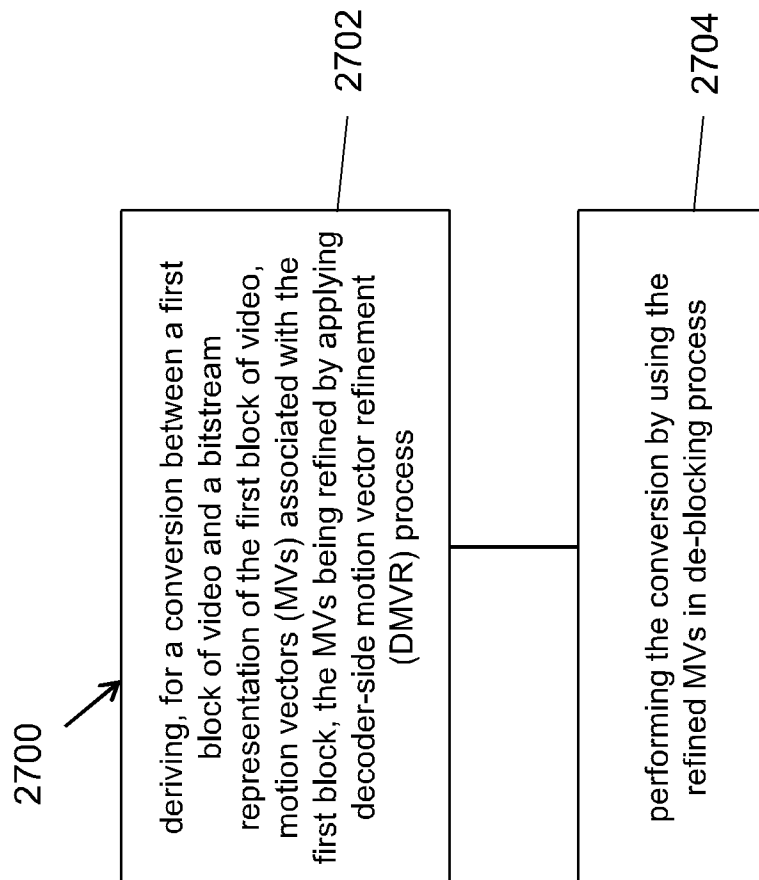
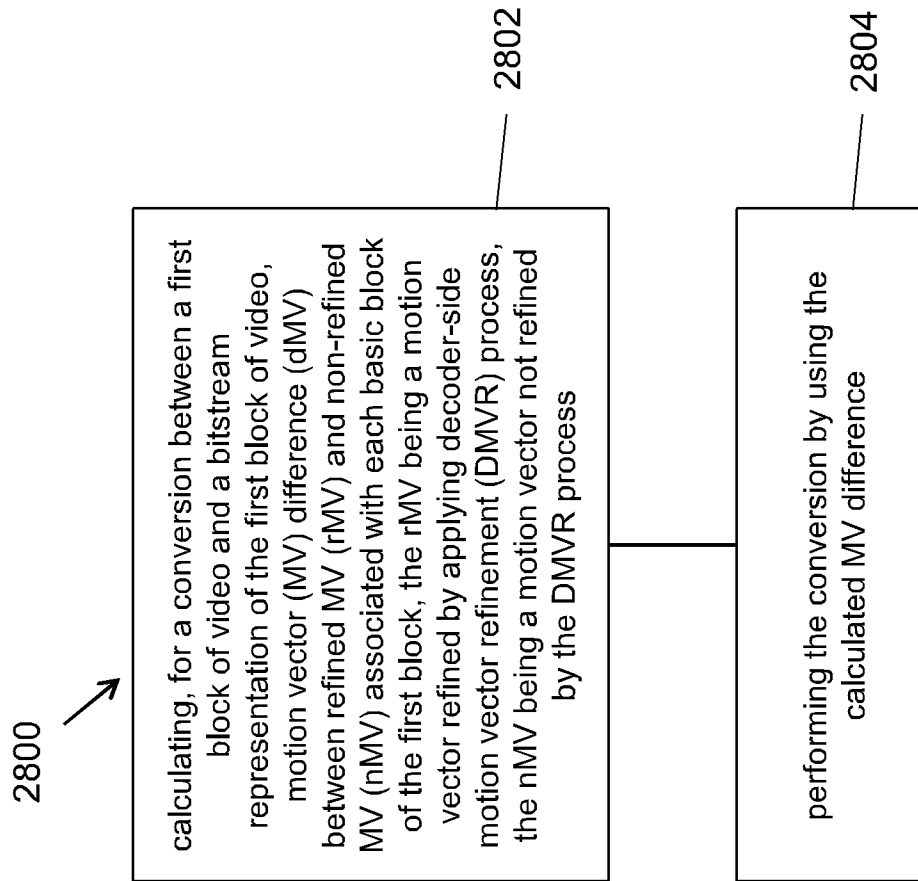


FIG. 26

**FIG. 27**

**FIG. 28**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/078749

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/00(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

CNKI, CNPAT, WPI, EPODOC, IEEE: encode,decode,decoder-side motion vector refinement,DMVR,refine,MV, motion vector,offset, initial searching point, Bi-directional Optical flow, BDOF,combined intra-inter prediction, CIIP, disable, not able, deblock,de-block,difference,clip,shift,height,width

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018192071 A1 (MEDIATEK INC.) 05 July 2018 (2018-07-05) claims 1-18, 24, description paragraphs 0060, 0071, 0114, 0115, 0123	34-68
A	US 2018192071 A1 (MEDIATEK INC.) 05 July 2018 (2018-07-05) claims 1-18, 24, description paragraphs 0060, 0071, 0114, 0115, 0123	1-33
A	US 2019020895 A1 (FUTUREWEI TECHNOLOGIES, INC.) 17 January 2019 (2019-01-17) the whole document	1-68
A	WO 2019001259 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 03 January 2019 (2019-01-03) the whole document	1-68
A	CN 101198064 A (WUHAN UNIVERSITY) 11 June 2008 (2008-06-11) the whole document	1-68
A	US 2018041769 A1 (MEDIATEK INC.) 08 February 2018 (2018-02-08) the whole document	1-68



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

“&” document member of the same patent family

Date of the actual completion of the international search

21 May 2020

Date of mailing of the international search report

10 June 2020

Name and mailing address of the ISA/CN

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

Facsimile No. (86-10)62019451

Authorized officer

LI,Jing

Telephone No. 86-10-53961705

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/078749

C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant	
A	LIU, Hongbin, et al. "CE9-related: Simplification of Decoder Side Motion Vector Derivation" <i>Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 11th Meeting: Ljubljana, 18 July 2018 (2018-07-18),</i> the whole document	1-68

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2020/078749

Patent document cited in search report			Publication date (day/month/year)	Patent family member(			
US	2018192071	A1	05 July 2018	BR	112019013832		
				WO	2018127119	A1	12 July 2018
				CN	110169073	A	23 August 2019
				TW	201841505	A	16 November 2018
				TW	I677238	B	11 November 2019
				EP	3545682	A1	02 October 2019
US	2019020895	A1	17 January 2019	WO	2019001259	A1	03 January 2019
				US	10477237	B2	12 November 2019
WO	2019001259	A1	03 January 2019	US	2019020895	A1	17 January 2019
				US	10477237	B2	12 November 2019
CN	101198064	A	11 June 2008	None			
US	2018041769	A1	08 February 2018	TW	201813396	A	01 April 2018
				WO	2018028559	A1	15 February 2018
				CN	109565590	A	02 April 2019
				TW	I669953	B	21 August 2019