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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
02 January 2020 (02.01.2020)

WIPO | PCT

(10) **International Publication Number**
WO 2020/003262 A1

(51) **International Patent Classification:**
H04N 19/463 (2014.01) ***H04N 19/105*** (2014.01) ***H04N 19/513*** (2014.01) ***H04N 19/139*** (2014.01) ***H04N 19/52*** (2014.01) ***H04N 19/176*** (2014.01) ***H04N 19/577*** (2014.01)

(21) **International Application Number:** PCT/IB2019/055550

(22) **International Filing Date:** 01 July 2019 (01.07.2019)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
PCT/CN2018/093897 30 June 2018 (30.06.2018) CN

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

(72) **Inventors: CHUANG, Hsiao Chiang**; 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). **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,

(54) **Title:** SYMMETRIC BI-PREDICTION MODE FOR VIDEO CODING

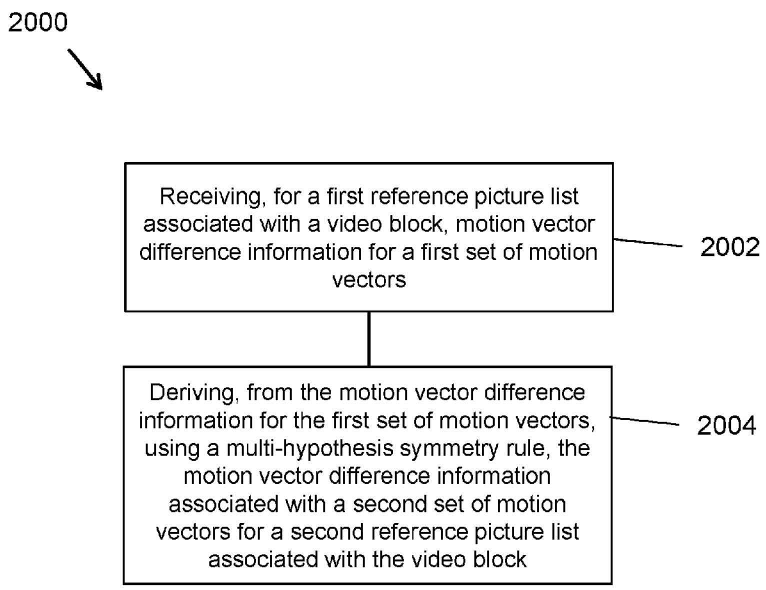


FIG. 20

(57) **Abstract:** A video bitstream processing method comprising generating, in response to a mirror mode flag in the video bitstream, a second motion vector difference information based on a symmetry rule and a first motion vector difference information; and reconstructing a video block using the first motion vector difference and the second motion vector difference information, wherein the reconstruction is performed bi-predictively.

[Continued on next page]

SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:
— *with international search report (Art. 21(3))*
— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

SYMMETRIC BI-PREDICTION MODE FOR VIDEO CODING

CROSS-REFERENCE TO RELATED APPLICATIONS

[001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefit of International Patent Application No. PCT/CN2018/093897, filed on June 30, 2018. For all purposes under the U.S. law, the entire disclosure of International Patent Application No. PCT/CN2018/093897 is incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[002] This document is related to image and video coding technologies.

BACKGROUND

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

SUMMARY

[004] The disclosed techniques may be used by visual media decoder or encoder embodiments in which coding efficiency is improved using symmetricity of motion vectors to reduce bits used for signaling motion information.

[005] In one example aspect, a video bitstream processing method is disclosed. The method includes generating, in response to a mirror mode flag in the video bitstream, a second motion vector difference information based on a symmetry rule and a first motion vector difference information. The method further includes reconstructing a video block in a current picture using the first motion vector difference information and the second motion vector difference, wherein the reconstruction is performed using bi-predictive prediction.

[006] In another example aspect, another method of video bitstream processing is disclosed. The method includes receiving, for a first reference picture list associated with a video block, motion vector difference information for a first set of motion vectors. The method further includes deriving, from the motion vector difference information for the first set of motion vectors, using a multi-hypothesis symmetry rule, motion vector difference information associated with a second set of motion vectors for a second reference picture list associated with the video block, wherein the multi-hypothesis symmetry rule specifies that the second motion vector difference value is (0,0), and a corresponding motion vector predictor is set to a mirrored motion vector value derived from the first motion vector difference information, and performing, using a result of the deriving, a conversion between the video block and a bitstream representation of the video block.

[007] In another example aspect, another method of video bitstream processing is disclosed. The method includes receiving, for a video block, a first motion vector difference information associated with a first reference picture list. The method further includes receiving, for the video block, a second motion vector difference information associated with a second reference picture list, and deriving, from the first motion vector difference information and the second motion vector difference information, using a multi-hypothesis symmetry rule, a third motion vector difference information associated with the first reference picture list and a fourth motion vector difference information associated with the second reference picture list, wherein the multi-hypothesis symmetry rule specifies that the second motion vector difference value is (0,0), and a corresponding motion vector predictor is set to a mirrored motion vector value derived from the first motion vector difference information..

[008] In another example aspect, another method of video bitstream processing is disclosed. The method includes receiving a future frame of video relative to a reference frame of video, receiving a motion vector related to the future frame of video and a past frame of video, applying a predetermined relationship between the future frame of video and the past frame of video, and reconstructing the past frame of video based on the future frame of video, the motion vector, and the predetermined relationship between the past frame of video and the future frame of video.

[009] In another example aspect, the above-described method may be implemented by a video decoder apparatus that comprises a processor.

[0010] In another example aspect, the above-described method may be implemented by a video encoder apparatus comprising a processor for decoding encoded video during video encoding process.

[0011] In yet another example aspect, these methods may be embodied in the form of processor-executable instructions and stored on a computer-readable program medium.

[0012] These, and other, aspects are further described in the present document.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] FIG. 1 shows an example of a derivation process for merge candidates list construction.

[0014] FIG. 2 shows example positions of spatial merge candidates.

[0015] FIG. 3 is an illustration of motion vector scaling for spatial motion vector candidate.

[0016] FIG. 4 shows an example derivation process for motion vector prediction candidates.

[0017] FIG. 5 shows example of candidate pairs considered for redundancy check of spatial merge candidates.

[0018] FIG. 6 shows example positions for the second PU of Nx2N and 2NxN partitions.

[0019] FIG. 7 is an example of motion vector scaling for temporal merge candidates.

[0020] FIG. 8 shows an example of candidate positions for temporal merge candidates labeled as C0 and C1.

[0021] FIG. 9 shows an example of combined bi-predictive merge candidates.

[0022] FIG. 10 shows an example of a bilateral matching process.

[0023] FIG. 11 shows an example of a template matching process.

[0024] FIG. 12 illustrates an example of unilateral motion estimation (ME) in frame rate up-conversion (FRUC).

[0025] FIG. 13 illustrates an example of a bilateral template matching process.

[0026] FIG. 14 illustrates an example of alternative temporal motion vector prediction (ATMVP) method.

[0027] FIG. 15 shows an example identifying source block and source picture.

[0028] FIG. 16 is an example of one coding unit (CU) with four sub-blocks (A-D) and its neighboring sub-blocks (a-d).

[0029] FIG. 17 shows a block diagram example of a video encoding apparatus.

[0030] FIG. 18 is a block diagram of an example of a video processing apparatus.

[0031] FIG. 19 is a flowchart for an example of a video bitstream processing method.

[0032] FIG. 20 is a flowchart for another example of a video bitstream processing method.

DETAILED DESCRIPTION

[0033] Section headings are used in the present document to facilitate ease of understanding and do not limit the embodiments disclosed in a section to only that section. As such, embodiments from one section can be combined with embodiments from other sections. Furthermore, while certain embodiments are described with reference specific video codecs, the disclosed techniques are applicable to other video coding technologies also. Furthermore, while some embodiments describe video coding steps in detail, it will be understood that corresponding steps decoding that undo the coding will be implemented by a decoder. Furthermore, the term video processing encompasses video coding or compression, video decoding or decompression and video transcoding in which video pixels are represented from one compressed format into another compressed format or at a different compressed bitrate.

[0034] The present document provides various techniques that can be used by a decoder of video bitstreams to improve the quality of decompressed or decoded digital video. Furthermore, a video encoder may also implement these techniques during the process of encoding in order to reconstruct decoded frames used for further encoding.

Signaling of Bi-prediction in HEVC

[0035] In HEVC, the inter PU-level signaling can be divided into three different modes. Table 1 and Table 2 show the related syntax elements for inter PU signaling in HEVC. The first mode is the skip mode, where only a single merge index needs to be signalled (merge_idx). The second mode is the merge mode, where only the merge flag (merge_flag) and the merge index (merge_idx) are signalled. The third mode is the AMVP mode, where a direction index (inter_pred_idc), a reference index (ref_idx_l0/ref_idx_l1), mvp index (mvp_l0_flag/mvp_l1_flag), and the MVD (mvd_coding) are signaled.

[0036] Among all the three modes, bi-predictive AMVP mode poses the more rate-consuming case while it provides the freedom to capture a wide variety of motion models including acceleration and other non-linear motion models. The motion vectors of two lists are signaled separately to provide such freedom.

AMVP derivation in HEVC

Motion vector prediction in AMVP mode

[0037] Motion vector prediction exploits spatio-temporal correlation of motion vector with neighboring PUs, which is used for explicit transmission of motion parameters. It constructs a motion vector candidate list by firstly checking availability of left, above temporally neighboring PU positions, removing redundant candidates and adding zero vector to make the candidate list to be constant length. Then, the encoder can select the best predictor from the candidate list and transmit the corresponding index indicating the chosen candidate. Similarly, with merge index signaling, the index of the best motion vector candidate is encoded using truncated unary. The maximum value to be encoded in this case is 2. In the following sections, details about derivation process of motion vector prediction candidate are provided.

Table 1. inter PU-level syntax element in HEVC

prediction_unit(x0, y0, nPbW, nPbH) {	**Descriptor**
if(cu_skip_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else { /* MODE_INTER */	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(slice_type = = B)	
inter_pred_idc[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(num_ref_idx_l0_active_minus1 > 0)	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0)	
mvp_l0_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(num_ref_idx_l1_active_minus1 > 0)	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] = = PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
} else	
mvd_coding(x0, y0, 1)	
mvp_l1_flag[x0][y0]	ae(v)
}	
}	
}	
}	

Table 2. Syntax elements of MVD coding in HEVC

mvd_coding(x0, y0, refList) {	**Descriptor**
abs_mvd_greater0_flag[0]	ae(v)
abs_mvd_greater0_flag[1]	ae(v)
if(abs_mvd_greater0_flag[0])	
abs_mvd_greater1_flag[0]	ae(v)
if(abs_mvd_greater0_flag[1])	
abs_mvd_greater1_flag[1]	ae(v)
if(abs_mvd_greater0_flag[0]) {	
if(abs_mvd_greater1_flag[0])	
abs_mvd_minus2[0]	ae(v)
mvd_sign_flag[0]	ae(v)
}	
if(abs_mvd_greater0_flag[1]) {	
if(abs_mvd_greater1_flag[1])	
abs_mvd_minus2[1]	ae(v)
mvd_sign_flag[1]	ae(v)
}	
}	

Motion vector prediction candidates

[0038] FIG. 1 summarizes derivation process for motion vector prediction candidate.

[0039] 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 Figure 2.

[0040] For temporal motion vector candidate derivation, one motion vector candidate is selected from two candidates, which are derived based on two different co-located positions. After the first list of spatio-temporal candidates is made, duplicated motion vector candidates in the list are removed. If the number of potential candidates is larger than two, motion vector candidates whose reference picture index within the associated reference picture list is larger than 1 are removed from the list. If the number of spatio-temporal motion vector candidates is smaller than two, additional zero motion vector candidates is added to the list.

Spatial motion vector candidates

[0041] In the derivation of spatial motion vector candidates, a maximum of two candidates are considered among five potential candidates, which are derived from PUs located in positions as depicted in FIG. 2, those positions being the same as those of motion merge. The order of derivation for the left side of the current PU is defined as A0, A1, scaled A0, and scaled A1. The order of derivation for the above side of the current PU is defined as B0, B1, B2, scaled B0, scaled B1, and scaled B2. For each side there are therefore four cases that can be used as motion vector candidate, with two cases not required to use spatial scaling, and two cases where spatial scaling is used. The four different cases are summarized as follows.

[0042] No spatial scaling

[0043] (1) Same reference picture list, and same reference picture index (same picture order count (POC))

[0044] (2) Different reference picture list, but same reference picture index (same POC)

[0045] Spatial scaling

[0046] (3) Same reference picture list, but different reference picture index (different POC)

[0047] (4) Different reference picture list, and different reference picture index (different POC)

[0048] The non-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 is not available or are intra coded, scaling for the above motion vector can help parallel derivation of left and above MV candidates. Otherwise, spatial scaling is not allowed for the above motion vector.

[0049] 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 Figure 3. 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.

Temporal motion vector candidates

[0050] 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. The reference picture index is signaled to the decoder.

Merge mode in HEVC

Candidates for merge mode

[0051] When a PU is predicted using merge mode, an index pointing to an entry in the merge candidates list is parsed from the bitstream and used to retrieve the motion information. The construction of this list is specified in the HEVC standard and can be summarized according to the following sequence of steps:

[0052] Step 1: Initial candidates derivation

[0053] - Step 1.1: Spatial candidates derivation

[0054] - Step 1.2: Redundancy check for spatial candidates

[0055] - Step 1.3: Temporal candidates derivation

[0056] Step 2: Additional candidates insertion

[0057] - Step 2.1: Creation of bi-predictive candidates

[0058] - Step 2.2: Insertion of zero motion candidates

[0059] These steps are also schematically depicted in Figure 4. For spatial merge candidate derivation, a maximum of four merge candidates are selected among candidates 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 does not reach to maximum number of merge candidate (MaxNumMergeCand) which is signaled in slice header. Since the number of candidates is constant, index of best merge candidate is encoded using truncated unary binarization (TU). If the size of CU is equal to 8, all the PUs of the current CU shares a single merge candidate list, which is identical to the merge candidate list of the 2N×2N prediction unit.

[0060] In the following sub-sections, the detailed operations of each of the above steps are described.

Spatial candidates

[0061] In the derivation of spatial merge candidates, a maximum of four merge candidates are selected among candidates located in the positions depicted in Figure 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 Figure 5 are considered and a candidate is only added to the list if the corresponding candidate used for redundancy check has not the same motion information. Another source of duplicate motion information is the “second PU” associated with partitions different from 2Nx2N. As an example, Figure 6 depicts the second PU for the case of N×2N and 2N×N, respectively. When the current PU is partitioned as N×2N, candidate at position 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×N.

Temporal candidates

[0062] In this step, only one candidate is added to the list. Particularly, in the derivation of this temporal merge candidate, a scaled motion vector is derived based on co-located PU belonging to the picture which has the smallest POC difference with current picture within the given reference picture list. The reference picture list to be used for derivation of the co-located PU is explicitly signaled in the slice header. The scaled motion vector for temporal merge candidate is obtained as illustrated by the dotted line in Figure 7, 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.

[0063] In the co-located PU (Y) belonging to the reference frame, the position for the temporal candidate is selected between candidates C0 and C1, as depicted in Figure 8. If PU at position C0 is not available, is intra coded, or is outside of the current coding tree unit (CTU), position C1 is used. Otherwise, position C0 is used in the derivation of the temporal merge candidate.

Additional candidate insertion

[0064] Besides spatio-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 spatio-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, Figure 9 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.

[0065] Zero motion candidates are inserted to fill the remaining entries in the merge candidates list and therefore hit the MaxNumMergeCand capacity. These candidates have zero spatial displacement and a reference picture index which starts from zero and increases every time a new zero motion candidate is added to the list. The number of reference frames used by these candidates is one and two for uni- and bi-directional prediction, respectively. Finally, no redundancy check is performed on these candidates.

Pattern matched motion vector

[0066] Pattern matched motion vector derivation (PMMVD) mode is a special merge mode based on Frame-Rate Up Conversion (FRUC) techniques. With this mode, motion information of a block is not signaled but derived at decoder side.

[0067] A FRUC flag is signaled for a CU when its merge flag is true. When the FRUC flag is false, a merge index is signaled and the regular merge mode is used. When the FRUC flag is true, an additional FRUC mode flag is signaled to indicate which method (bilateral matching or template matching) is to be used to derive motion information for the block.

[0068] At encoder side, the decision on whether using FRUC merge mode for a CU is based on RD cost selection as done for normal merge candidate. That is the two matching modes (bilateral matching and template matching) are both checked for a CU by using RD cost selection. The one leading to the minimal cost is further compared to other CU modes. If a FRUC matching mode is the most efficient one, FRUC flag is set to true for the CU and the related matching mode is used.

[0069] Motion derivation process in FRUC merge mode has two steps. A CU-level motion search is first performed, then followed by a Sub-CU level motion refinement. At CU level, an initial motion vector is derived for the whole CU based on bilateral matching or template matching. First, a list of MV candidates is generated and the candidate which leads to the minimum matching cost is selected as the starting point for further CU level refinement. Then a local search based on bilateral matching or template matching around the starting point is performed and the MV results in the minimum matching cost is taken as the MV for the whole CU. Subsequently, the motion information is further refined at sub-CU level with the derived CU motion vectors as the starting points.

[0070] For example, the following derivation process is performed for a W×H CU motion information derivation. At the first stage, MV for the whole W×H CU is derived. At the second stage, the CU is further split into M×M sub-CUs. The value of M is calculated as in (1), D is a predefined splitting depth which is set to 3 by default in the JEM. Then the MV for each sub-CU is derived.

$$\boldsymbol{M} = \max\{4, \min\{\frac{M}{2^D}, \frac{N}{2^D}\}\} \qquad \textbf{(Equation 1)}$$

[0071] As shown in the FIG. 10, the bilateral matching is used to derive motion information of the current CU by finding the closest match between two blocks along the motion trajectory of the current CU in two different reference pictures. Under the assumption of continuous motion trajectory, the motion vectors MV0 and MV1 pointing to the two reference blocks shall be proportional to the temporal distances, i.e., TD0 and TD1, between the current picture and the two reference pictures. As a special case, when the current picture is temporally between the two reference pictures and the temporal distance from the current picture to the two reference pictures is the same, the bilateral matching becomes mirror based bi-directional MV.

[0072] As shown in Figure 11, template matching is used to derive motion information of the current CU by finding the closest match between a template (top and/or left neighboring blocks of the current CU) in the current picture and a block (same size to the template) in a reference picture. In addition to the FRUC merge mode, the template matching is also applied to AMVP mode. In JEM, there are two AMVP candidates. With template matching method, a new candidate is derived.

If the newly derived candidate by template matching is different to the first existing AMVP candidate, it is inserted at the very beginning of the AMVP candidate list and then the list size is set to two (meaning remove the second existing AMVP candidate). When applied to AMVP mode, only CU level search is applied.

CU level MV candidate set

[0073] The MV candidate set at CU level consists of:

[0074] - Original AMVP candidates if the current CU is in AMVP mode

[0075] - all merge candidates,

[0076] - several MVs in the interpolated MV field.

[0077] - top and left neighboring motion vectors

[0078] When using bilateral matching, each valid MV of a merge candidate is used as an input to generate a MV pair with the assumption of bilateral matching. For example, one valid MV of a merge candidate is (MVa, refa) at reference list A. Then the reference picture refb of its paired bilateral MV is found in the other reference list B so that refa and refb are temporally at different sides of the current picture. If such a refb is not available in reference list B, refb is determined as a reference which is different from refa and its temporal distance to the current picture is the minimal one in list B. After refb is determined, MVb is derived by scaling MVa based on the temporal distance between the current picture and refa, refb.

[0079] Four MVs from the interpolated MV field are also added to the CU level candidate list. More specifically, the interpolated MVs at the position (0, 0), (W/2, 0), (0, H/2) and (W/2, H/2) of the current CU are added.

[0080] When FRUC is applied in AMVP mode, the original AMVP candidates are also added to CU level MV candidate set.

[0081] At the CU level, up to 15 MVs for AMVP CUs and up to 13 MVs for merge CUs are added to the candidate list.

Sub-CU level MV candidate set

[0082] The MV candidate set at sub-CU level consists of:

[0083] - an MV determined from a CU-level search,

[0084] - top, left, top-left and top-right neighboring MVs,

[0085] - scaled versions of collocated MVs from reference pictures,

[0086] - up to 4 ATMVP candidates,

[0087] - up to 4 STMVP candidates

[0088] The scaled MVs from reference pictures are derived as follows. All the reference pictures in both lists are traversed. The MVs at a collocated position of the sub-CU in a reference picture are scaled to the reference of the starting CU-level MV.

[0089] ATMVP and STMVP candidates are limited to the four first ones.

[0090] At the sub-CU level, up to 17 MVs are added to the candidate list.

Generation of interpolated MV field

[0091] Before coding a frame, interpolated motion field is generated for the whole picture based on unilateral ME. Then the motion field may be used later as CU level or sub-CU level MV candidates.

[0092] First, the motion field of each reference pictures in both reference lists is traversed at 4×4 block level. For each 4×4 block, if the motion associated to the block passing through a 4×4 block in the current picture (as shown in Figure 12) and the block has not been assigned any interpolated motion, the motion of the reference block is scaled to the current picture according to the temporal distance TD0 and TD1 (the same way as that of MV scaling of TMVP in HEVC) and the scaled motion is assigned to the block in the current frame. If no scaled MV is assigned to a 4×4 block, the block's motion is marked as unavailable in the interpolated motion field.

Interpolation and matching cost

[0093] When a motion vector points to a fractional sample position, motion compensated interpolation is needed. To reduce complexity, bi-linear interpolation instead of regular 8-tap HEVC interpolation is used for both bilateral matching and template matching.

[0094] The calculation of matching cost is a bit different at different steps. When selecting the candidate from the candidate set at the CU level, the matching cost is the absolute sum difference (SAD) of bilateral matching or template matching. After the starting MV is determined, the matching cost C of bilateral matching at sub-CU level search is calculated as follows:

$$C = SAD + w \cdot (|MV_x - MV_x^s| + |MV_y - MV_y^s|) \qquad \textbf{(Equation 2)}$$

where w is a weighting factor which is empirically set to 4, MV and MV^s indicate the current MV and the starting MV, respectively. SAD is still used as the matching cost of template matching at sub-CU level search.

[0095] In FRUC mode, MV is derived by using luma samples only. The derived motion will be used for both luma and chroma for MC inter prediction. After MV is decided, final MC is performed using 8-taps interpolation filter for luma and 4-taps interpolation filter for chroma.

MV refinement

[0096] MV refinement is a pattern based MV search with the criterion of bilateral matching cost or template matching cost. In the JEM, two search patterns are supported – an unrestricted center-biased diamond search (UCBDS) and an adaptive cross search for MV refinement at the CU level and sub-CU level, respectively. For both CU and sub-CU level MV refinement, the MV is directly searched at quarter luma sample MV accuracy, and this is followed by one-eighth luma sample MV refinement. The search range of MV refinement for the CU and sub-CU step are set equal to 8 luma samples.

Selection of prediction direction in template matching FRUC merge mode

[0097] In the bilateral matching merge mode, bi-prediction is always applied since the motion information of a CU is derived based on the closest match between two blocks along the motion trajectory of the current CU in two different reference pictures. There is no such limitation for the template matching merge mode. In the template matching merge mode, the encoder can choose among uni-prediction from list0, uni-prediction from list1 or bi-prediction for a CU. The selection is based on a template matching cost as follows:

If *costBi* <= *factor* * min (*cost*0, *cost1*)

bi-prediction is used;

Otherwise, if *cost*0 <= *cost1*

uni-prediction from list0 is used;

Otherwise,

uni-prediction from list1 is used;

where cost0 is the SAD of list0 template matching, cost1 is the SAD of list1 template matching and costBi is the SAD of bi-prediction template matching. The value of *factor* is equal to 1.25, which means that the selection process is biased toward bi-prediction.

[0098] The inter prediction direction selection is only applied to the CU-level template matching process.

Decoder-side motion vector refinement

[0099] 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 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 by a bilateral template matching process. The bilateral template matching applied in the decoder to perform a distortion-based search between a bilateral template and the reconstruction samples in the reference pictures to obtain a refined MV without transmission of additional motion information.

[00100] In DMVR, a 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 Figure 10. The template matching operation consists of calculating cost measures between the generated 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 Figure 10, are used for generating the final bi-prediction results. A sum of absolute differences (SAD) is used as the cost measure.

[00101] DMVR is applied for the merge mode of bi-prediction with one MV from a reference picture in the past and another from a reference picture in the future, without the transmission of additional syntax elements. In the JEM, when LIC, affine motion, FRUC, or sub-CU merge candidate is enabled for a CU, DMVR is not applied.

Adaptive motion vector difference resolution

[00102] In HEVC, motion vector differences (MVDs) (between the motion vector and predicted motion vector of a PU) are signaled in units of quarter luma samples when use_integer_mv_flag is equal to 0 in the slice header. In the JEM, a locally adaptive motion vector resolution (LAMVR) is introduced. In the JEM, MVD can be coded in units of quarter luma samples, integer luma samples or four luma samples. The MVD resolution is controlled at the coding unit (CU) level, and MVD resolution flags are conditionally signaled for each CU that has at least one non-zero MVD components.

[00103] For a CU that has at least one non-zero MVD components, a first flag is signaled 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 signaled to indicate whether integer luma sample MV precision or four luma sample MV precision is used.

[00104] 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.

[00105] In the encoder, CU-level RD checks are used to determine which MVD resolution is to be used for a CU. That is, the CU-level RD check is performed three times for each MVD resolution. To accelerate encoder speed, the following encoding schemes are applied in the JEM.

[00106] During RD check of a CU with normal quarter luma sample MVD resolution, the motion information of the current CU (integer luma sample accuracy) is stored. The stored motion information (after rounding) is used as the starting point for further small range motion vector refinement during the RD check for the same CU with integer luma sample and 4 luma sample MVD resolution so that the time-consuming motion estimation process is not duplicated three times.

[00107] RD check of a CU with 4 luma sample MVD resolution is conditionally invoked. For a CU, when RD cost integer luma sample MVD resolution is much larger than that of quarter luma sample MVD resolution, the RD check of 4 luma sample MVD resolution for the CU is skipped.

Sub-CU based motion vector prediction

[00108] In JEM, each CU can have at most one set of motion parameters for each prediction direction. Two sub-CU level motion vector prediction methods are considered in the encoder by splitting a large CU into sub-CUs and deriving motion information for all the sub-CUs of the large CU. Alternative temporal motion vector prediction (ATMVP) method allows each CU to fetch multiple sets of motion information from multiple blocks smaller than the current CU in the collocated reference picture. In spatial-temporal motion vector prediction (STMVP) method motion vectors of the sub-CUs are derived recursively by using the temporal motion vector predictor and spatial neighboring motion vector.

[00109] To preserve more accurate motion field for sub-CU motion prediction, the motion compression for the reference frames is currently disabled.

Alternative temporal motion vector prediction

[00110] In the alternative temporal motion vector prediction (ATMVP) method, the motion vectors temporal motion vector prediction (TMVP) is modified by fetching multiple sets of motion information (including motion vectors and reference indices) from blocks smaller than the current CU. As shown in Figure 11, the sub-CUs are square N×N blocks (N is set to 4 by default).

[00111] Figure 13 illustrates an example of a bilateral template matching process. In a first step, a bilateral template is generated from prediction blocks. In a second step, bilateral template matching is used to find the best matched blocks.

[00112] ATMVP predicts the motion vectors of the sub-CUs within a CU in two steps. The first step is to identify the corresponding block in a reference picture with a so-called temporal vector. The reference picture is called the motion source picture. The second step is to split the current CU into sub-CUs and obtain the motion vectors as well as the reference indices of each sub-CU from the block corresponding to each sub-CU, as shown in Figure 14.

[00113] In the first step, a reference picture and the corresponding block is determined by the motion information of the spatial neighboring blocks of the current CU. To avoid the repetitive scanning process of neighboring blocks, the first merge candidate in the merge candidate list of the current CU is used. The first available motion vector as well as its associated reference index are set to be the temporal vector and the index to the motion source picture. This way, in ATMVP, the corresponding block may be more accurately identified, compared with TMVP, wherein the corresponding block (sometimes called collocated block) is always in a bottom-right or center position relative to the current CU. In one example, if the first merge candidate is from the left neighboring block (i.e., A1 in Figure 15), the associated MV and reference picture are utilized to identify the source block and source picture.

[00114] In the second step, a corresponding block of the sub-CU is identified by the temporal vector in the motion source picture, by adding to the coordinate of the current CU the temporal vector. For each sub-CU, the motion information of its corresponding block (the smallest motion grid that covers the center sample) is used to derive the motion information for the sub-CU. After the motion information of a corresponding N×N block is identified, it is converted to the motion vectors and reference indices of the current sub-CU, in the same way as TMVP of HEVC, wherein motion scaling and other procedures apply. For example, the decoder checks whether the low-delay condition (i.e. the POCs of all reference pictures of the current picture are smaller than the POC of the current picture) is fulfilled and possibly uses motion vector MVx (the motion vector corresponding to reference picture list X) to predict motion vector MVy (with X being equal to 0 or 1 and Y being equal to 1−X) for each sub-CU.

Spatial-temporal motion vector prediction

[00115] In this method, the motion vectors of the sub-CUs are derived recursively, following raster scan order. Figure 16 illustrates this concept. Let us consider an 8×8 CU which contains four 4×4 sub-CUs A, B, C, and D. The neighboring 4×4 blocks in the current frame are labelled as a, b, c, and d.

[00116] The motion derivation for sub-CU A starts by identifying its two spatial neighbors. The first neighbor is the N×N block above sub-CU A (block c). If this block c is not available or is intra coded the other N×N blocks above sub-CU A are checked (from left to right, starting at block c). The second neighbor is a block to the left of the sub-CU A (block b). If block b is not available or is intra coded other blocks to the left of sub-CU A are checked (from top to bottom, staring at block b). The motion information obtained from the neighboring blocks for each list is scaled to the first reference frame for a given list. Next, temporal motion vector predictor (TMVP) of sub-block A is derived by following the same procedure of TMVP derivation as specified in HEVC. The motion information of the collocated block at location D is fetched and scaled accordingly. Finally, after retrieving and scaling the motion information, all available motion vectors (up to 3) are averaged separately for each reference list. The averaged motion vector is assigned as the motion vector of the current sub-CU.

Sub-CU motion prediction mode signaling

[00117] The sub-CU modes are enabled as additional merge candidates and there is no additional syntax element required to signal the modes. Two additional merge candidates are added to merge candidates list of each CU to represent the ATMVP mode and STMVP mode. Up to seven merge candidates are used, if the sequence parameter set indicates that ATMVP and STMVP are enabled. The encoding logic of the additional merge candidates is the same as for the merge candidates in the HM, which means, for each CU in P or B slice, two more RD checks is needed for the two additional merge candidates.

[00118] In the JEM, all bins of merge index are context coded by CABAC. While in HEVC, only the first bin is context coded and the remaining bins are context by-pass coded.

Examples of problems solved by embodiments

[00119] While it provides great flexibility to adapt to the versatile motion in video signal, MVD constitutes a large portion of bitstream. Especially during bi-prediction, MVDs of both L0 and L1 need to be signaled and they introduce large overhead, especially for low-rate visual communication. Some properties about motion symmetricity can be utilized to save the rate spent on coding of motion information.

[00120] Current AMVP mode (including both the MVP index and the reference index) is separately signaled for both L0 and L1, while they may be more efficiently represented when the motion follows a symmetric model.

Embodiment examples

1. During bi-prediction, the property of symmetricity of motion vector can be utilized to generate the basis MV set for AMVP mode. Specifically, the MVD is only signaled for a single direction (list), and the MV of the other direction is set using a mirrored condition. Alternatively, furthermore, the MV may be further refined. Such a mode is called sym-bi-mode. Herein, the bi-prediction refers to prediction by using one reference frame from the past and the other reference frame from the future in display order. In some example embodiments, versatile video coding (VVC) (e.g., JVET-N1001-v5 and other versions and standards) includes a symmetric motion vector difference (SMVD) mode which may skip the signaling of L1 MVD. The skipped L1 MVD may be set to the mirror of the L0 MVD without scaling.

    a. In one example, when L(*1-N*) (*N=0 or 1*) MVD is sent, MVD value of L*N* is not sent (i.e., inherited to be (0, 0)), and the MVP value is set to the mirrored MV value from L(*1-N*) MV. After that, a motion refinement may be applied to L*N* motion vector.

(i) In one example, the DMVR refinement process may be applied. Alternatively, the FRUC refinement process may be applied to refine L*N* motion vector.

(ii) In one example, the search range of the refinement can be pre-defined or signalled through the SPS (Sequence Parameter Set), PPS (Picture Parameter Set), VPS (Video Parameter set), or slice headers.

(iii) In one example, the motion refinement can be applied to a specific grid. For example, the uniform sampling grid with grid distance *d* can be used to define the search points. The grid distance *d* can be pre-defined, or signalled via SPS, PPS, VPS, or slice headers. The use of the sampling grid can be considered as a sub-sampled search region and hence has the benefit of reducing memory bandwidth required by the search.

(iv) In one example, the signalling of the mirror mode can be done in either the CU level, CTU level, region level (covering multiple CUs/CTUs) or slice level. When it is done at CU level, a one-bit flag needs to be signalled when it is sym-bi-mode. That is, when this flag is signalled to be 1, the associated L*N* MVD as well as its MVP index can be skipped. When it is done in the CTU level, region level, or slice level, all sym-bi-mode will not signal the L*N* MVD values and their MVP indexes. In some example embodiments, the signaling of a SMVD flag occurs at the CU level.

b. In one example, there is a one-bit flag in the slice header/picture parameter set/sequence parameter set signalling whether the refinement process should be invoked or not. Alternatively, it may be also signaled in CU/CTU/region level.

c. In one example, during bi-prediction, which list of MVD to be skipped can be signalled. The signalling can occur at CU level, region level, CTU level, or slice level. When it is signalled at CU level, a one-bit flag needs to be signaled in sym-bi-mode. When it is signalled at region level, CTU level, or slice level, all belonging bi-predictive CUs will skip the MVD signalling of the specified list and use the mirrored MVP as its starting point to find the final motion vector.

d. In one example, only the mirrored MVP needs to be stored in the MV buffer for motion prediction (AMVP, merge) of subsequent blocks. The refined motion vectors do not need to be stored in the MV buffer.

e. In one example, the MVP can be placed along with the conventional MVP indexes, and one extra bit (as a total of 2) is required to signal the three MVP indexes. In some embodiments, in SMVD mode, both MVP indices are signaled as a regular AMVP mode.

f. In one example, the mirrored MVP candidate is added in place of the second AMVP candidate. Still, only one-bit is required to signal the MVP index.

g. In one example, the mirrored MVP mode can be applied when the POC distances between the two reference frames are equal. In some embodiments, in SMVD mode, two references are derived as the closest reference frames to the current frame in both L0 and L1.

h. In one example, the scaling introduced by the mirroring can use the relative temporal distance between the source and the destination frames. For example, if reference frame of L(*1-N*) and reference frame of L*N* are used and it is decided to skip the MVD signaling of L*N*, the initial motion vector of L*N* ($N = 0$ *or* 1) can be calculated as: MVP*N* = ($\tau N/\tau(1-N)$)·MV(*1-N*), where $\tau 0$ and $\tau 1$ represent the POC distances between current frame and reference frame of L0 and that of L1, respectively.

2. The refinement process can be done using various matching schemes. Let the patches from L0 and L1 pictures be P0 and P1, respectively. A patch is defined as the prediction samples generated by the interpolation process of a MV.

   a. The similarity between P0 and P1 is used as the criteria to select the refined MV. In one example, the refinement finds the MV*N* *(N=0 or 1)* which minimizes the sum of absolute difference (SAD) between P0 and P1.

   b. A temporary patch is generated by the P0 and P1 and the criteria could be defined as finding the MV with the highest correlation between the prediction patch and the temporary patch. For example, a separate patch P' = (P0+P1)/2 can be created and be used to find the MV*N* *(N=0 or 1)* which minimizes the SAD of between P' and P*N*. More generally, P' can be generated by formula: $P'=\omega\cdot P0+(1-\omega)\cdot P1$, where ω is a weighting factor between 0 and 1.

   c. In one example, template-based matching scheme can be used to define the refinement process. The top-template, left-template, or both top and left template can be used to find MV*N* *(N=0 or 1)*. The procedure of finding MV*N* *(N=0 or 1)* resembles the procedure described in the above two examples.

   d. In one example, depending on the distance of search points to the initial mirrored MVP position, the interpolation process can be skipped for some of the search points. No interpolation process is involved when searching for points where their distances to the MVP*N* *(N=0 or 1)* exceed a threshold *T*. Only the integer-pixel reference samples are used as patches to derive motion vector. *T* can be predefined, or signalled via SPS, PPS, VPS, or slice headers.

   e. In one example, the cost metric used to find the MV*N* include the estimated rate introduced by the search point to the mirrored MVP: $C = SAD + \lambda\cdot R$, where λ is a weighting factor to weight the importance of estimated rate during the refinement process. The value of λ can be pre-defined, signaled through SPS, PPS, VPS, or slice headers. Note that the MVDN, MVN, and MVPN defined below are two-dimensional vectors.

i. In one example, R = ||MVDN||, where MVDN = MVN – MVPN. Here, the function ||·|| represents the L1 norm.

ii. In one example, R = round($\log_2$(||MVDN||)), where the function round indicates the rounding function of the input argument to the nearest integers.

iii. In one example, R = mvd_coding(MVDN), where the function mvd_coding indicates standard-compliant binarization process of the input MVD value.

3. The MVD_L1_ZERO_FLAG is a slice-level flag which imposes a strong constraint on L1 MVD signaling by removing all the L1 MVD values. The mirrored MV and refinement can be used in conjunction with such design in the following ways.

    a. In one example, when MVD_L1_ZERO_FLAG is enabled, no MVP index is signalled, and the mirrored MVP constraint and refinement process can still be applied.

    b. In one example, when MVD_L1_ZERO_FLAG is enabled, MVP index is still signalled (e.g., as in 1.e or 1.f) and the mirrored MVP constraint is not imposed. However, the MV refinement process can still be applied.

    c. In one example, when MVD_L1_ZERO_FLAG is enabled, the mirrored MVP is added to the MVP candidate lists, followed by the MV refinement process.

4. When it comes to the signaling of reference indexes and MVP indexes of LN ($N = 0$ *or* 1), a joint MVP list can be created to support the mirrored MVD mode. That is, the MVP list is derived jointly for L0 and L1 (given a pair of specific reference indexes), and only a single index needs to be signalled.

    a. In one example, the signaling of refIdx$N$ can be skipped and only the reference frame which is closest to the mirrored location of L(1-N) reference frame is selected as its reference frame for MVP scaling. In some embodiments, in SMVD mode, both reference indices are skipped as they are chosen to be the closest reference frames to the current frame in both lists.

    b. In one example, MVP candidates which are unable to create Bi-predictors should be considered invalid during the derivation process.

    c. In one example, the derivation can be done by following the existing procedure of MVP derivation for L(1-N), except that when the scaling occurs, the candidate pairs which result in motion vectors lying on both reference frames of L0 and L1 in the Decoded Picture Buffer (DPB) are considered as valid candidates.

    d. Mirrored MVD mode may be expressed including:

```
if( sym_mvd_flag[ x0 ][ y0 ] ) {
        MvdL1[ x0 ][ y0 ][ 0 ] = −MvdL0[ x0 ][ y0 ][ 0 ]
        MvdL1[ x0 ][ y0 ][ 1 ] = −MvdL0[ x0 ][ y0 ][ 1 ]
    } else
```

5. The proposed methods may be also applied to the multi hypothesis mode.
    a. In this case, when there are two sets MV information for each reference picture list, MV information may be signaled for one reference picture list. However, the MVD of the sets of MV information of the other reference picture list may be derived. For each set of MV information of one reference picture list, it may be treated in the same way as the sym-bi-mode.
    b. Alternatively, when there are two sets of MV information for each reference picture list, one set of MV information of the two reference picture lists may be signaled. While the other two sets of MV information of the two reference picture lists may be derived on the fly using the sym-bi-mode.

[00121] Many video coding standards are based on a hybrid video coding structure wherein temporal prediction plus transform coding are utilized. An example of a typical HEVC encoder framework is depicted in FIG. 17.

[00122] FIG. 18 is a block diagram 1800of a video processing apparatus. The apparatus 1800 may be used to implement one or more of the methods described herein. The apparatus 1800 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1800 may include one or more processors 1802, one or more memories 1804 and video processing hardware 1806. The processor(s) 1802 may be configured to implement one or more methods described in the present document. The memory (memories) 1804 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1806 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00123] FIG. 19 is a flowchart for an example method 1900 of video bitstream processing. The method 1900 includes generating (1902), in response to a mirror mode flag in the video bitstream, a second motion vector difference information based on a symmetry rule and a first motion vector difference information; and reconstructing (1904) a video block using the first motion vector difference and the second motion vector difference information, wherein the reconstruction is performed bi-predictively.

[00124] FIG. 20 is a flowchart for an example method 2000 of video bitstream processing. The method 2000 includes receiving (2002), for a first reference picture list associated with a video block, motion vector difference information for a first set of motion vectors; and deriving (2004), from the motion vector difference information for the first set of motion vectors, using a multi-hypothesis symmetry rule, the motion vector difference information associated with a second set of motion vectors for a second reference picture list associated with the video block. This information may be generated using the received motion vector difference information for the first set of motion vectors.

[00125] In some embodiments, a method of video bitstream processing may include a variation of the method 2000 in which the partial motion vector difference information in case of multi-hypothesis is signaled in an interweaved manner. Such a method includes receiving, for a video block, a first motion vector difference information associated with a first reference picture list, receiving, for the video block, a second motion vector difference information associated with a second reference picture list; deriving, from the first motion vector difference information and the second motion vector difference information, using a multi-hypothesis symmetry rule, a third motion vector difference information associated with the first reference picture list and a fourth motion vector difference information associated with the second reference picture list.

[00126] With respect to methods 1900 and 2000, the bitstream processing may include generation of bitstream that represents the video in a compressed form. Alternatively, bitstream processing may include using the bitstream to reconstruct video from its compressed form representation.

[00127] With respect to methods 1900 and 2000, in some embodiments, the symmetry rule and the multi-hypothesis symmetry rules may be same or different. In particular, the multi-hypothesis symmetry rule may only be used when a video block (or picture) is encoded using multi-hypothesis motion prediction.

[00128] With respect to methods 1900 and 2000, the symmetry rule may specify that the second motion vector prediction difference value is to be (0,0), and a corresponding motion vector predictor is set to a mirrored motion vector whose value is derived from the first motion vector difference information. Furthermore, motion vector refinement may further be performed on the mirrored motion vector value. As described in the examples above, the mirror mode may be selectively used based on an indication in the bitstream at the CU/CTU/region level. Similarly, motion vector refinement may also be controlled to be used (or not used) by signaling of a refinement flag. The refinement flag may be used at a slice header, or picture parameter set, or sequence parameter set or region level or coding unit or coding tree unit level.

[00129] With respect to methods 1900 and 2000, the use of the symmetry rule-based technique for generating mirrored motion vectors may enable skipping sending motion vector difference information in the bitstream (because this information can be generated by the decoder). The skipping operation may be selectively controlled via a flag in the bitstream. In one advantageous aspect, mirrored MVP calculations using the above-described techniques may be used at the decoder side for improved decoding of subsequent blocks, without suffering from the dependency of calculation that may occur if refined motion vectors are used for prediction of subsequent blocks.

[00130] With respect to methods 1900 and 2000, in some embodiments, the symmetry rule may only be used for generating mirrored motion vectors in the case that the two reference frames have a same distance. Otherwise, scaling of motion vectors may be performed based on relative temporal distances of the reference frames.

[00131] With respect to methods 1900 and 2000, in some embodiments, the mirrored motion vectors may be calculated using a patch-based technique and may include generating a first patch of prediction samples using the first motion vector difference from list 0 of reference frames, generating a second patch of prediction samples using the first motion vector difference from list 1 of reference frames, and determining the motion vector refinement to be a value that minimizes an error function between the first patch and the second patch. Various optimization criteria (e.g., rate distortion, SAD, and so on) may be used to determine the refined motion vectors.

[00132] It will be appreciated that techniques for reducing the quantity of bits used for representing motion in a compressed video bitstream are disclosed. Using the disclosed technique, bi-directional prediction may be signaled using only half the motion information of a conventional technique, and the other half of the motion information may be generated at the decoder using a mirror symmetry of motion of objects in a video. A symmetry flag and a refinement flag may be used to signal use (or no-use) of this mode and further refinement of motion vectors. Mirrored motion vectors may be calculated using symmetry rules. One assumption made in symmetry rule is that an object maintains its translational motion between the time of the current block and the times of reference blocks used for bi-prediction. For example, using one symmetry rule, a motion vector pointing to a reference region that is delx and dely displaced from the current block in one temporal direction may be assumed to change to a scaled version of delx and dely in another direction (scaling may also include negative scaling, which may be due to changing direction of motion vectors). The scaling may depend on temporal distances, and other considerations and described in the present document.

[00133] 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.

[00134] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[00135] The processes and logic flows described in this document can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

[00136] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[00137] While this patent document contains many specifics, these should not be construed as limitations on the scope of any invention or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this patent document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[00138] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described in this patent document should not be understood as requiring such separation in all embodiments.

[00139] 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

What is claimed is:

1. A method of processing a video bitstream, comprising:

generating, in response to a mirror mode flag in the video bitstream, a second motion vector difference information based on a symmetry rule and a first motion vector difference information; and

reconstructing a video block in a current picture using the first motion vector difference information and the second motion vector difference information, wherein the reconstruction is performed using a bi-predictive prediction.

2. The method of claim 1, wherein the symmetry rule specifies that the second motion vector difference information is not sent.

3. The method of claim 2, further comprising:

performing motion vector refinement of a mirrored motion vector value to generate a motion vector refinement value.

4. The method of claim 1, wherein the mirror mode flag is present at either the coding unit (CU) level, coding tree unit (CTU) level, region level covering multiple CUs/CTUs, or slice level.

5. The method of claim 3, wherein the motion vector refinement is selectively performed based on a refinement flag in the video bitstream.

6. The method of claim 5, wherein the refinement flag is included in at least a slice header, a picture parameter set, a sequence parameter set, a region level, a coding unit, or a coding tree unit level.

7. The method of claim 1, wherein the video bitstream includes skipping information indicative of a list of motion vector differences skipped from signaling in the video bitstream.

8. The method of claim 7, wherein the skipping information is at a coding unit level, a region level, a coding tree unit level, or a slice level.

9. The method of claim 8, wherein, in case that the skipping information is at the region level, the coding tree unit level, or the slice level, coding units use the symmetry rule for generating second motion vector information.

10. The method of claim 1, further comprising:

storing the motion vector predictor generated using the symmetry rule for processing prediction information of a subsequent video block.

11. The method of claim 10, wherein the motion vector predictor is used with conventional motion vector predictors, and wherein a two-bit field signals the motion vector predictor in the video bitstream.

12. The method of claim 10, wherein the motion vector predictor is used in place of one of the conventional motion vector predictors and signaling is performed with a single bit in the video bitstream.

13. The method of any of claims 1 to 12, wherein the symmetry rule is only used in case of a picture order count distance between two reference frames used for the bi-predictive prediction being equal.

14. The method of claim 1, wherein the video bitstream omits signaling of a motion vector difference value for a reference picture list l1, and wherein the bi-predication is performed using: (a first picture order count (POC) of a first reference picture in a reference picture list 0) - (a second POC of the current picture) = (the POC of the current picture) - (a third POC of another reference picture in the reference picture list 1).

15. The method of claim 1, wherein two reference pictures for the bi-predictive prediction are derived as closest reference pictures to the current picture from both past frames and future frames.

16. The method of claim 1, wherein the video bitstream jointly signals reference indexes and motion vector predictor indexes of reference list 0 and reference list 1 using a single reference index and a single motion vector prediction index per video block.

17. The method of claim 1, wherein a mirrored motion vector value is determined using scaling that is proportional to relative temporal distances between a source frame of the video block and a destination reference frame.

18. The method of claim 3, wherein the performing the motion vector refinement includes:

generating a first patch of prediction samples using a third motion vector from a reference frame associated with a first reference picture list;

generating a second patch of prediction samples using the mirrored motion vector value from the reference frame associated with the second reference picture list; and

determining the motion vector refinement value to be a value that minimizes an error function between the first patch and the second patch.

19. The method of claim 18, wherein the error function comprises a sum of absolute difference measure.

20. The method of claim 18, wherein the error function comprises a correlation between the motion vector refinement value and a weighted linear average of the first patch and the second patch.

21. The method of claim 18, wherein the error function is a rate distortion function using the motion vector refinement value.

22. The method of claim 3, wherein the performing the motion vector refinement includes:

determining the motion vector refinement value to be a value that minimizes an error function using the top and left reference or interpolated samples between the reference frames associated with the two reference picture lists.

23. The method of claim 3, wherein the performing the motion vector refinement includes:

determining the motion vector refinement value to be a value that minimizes an error function using integer reference samples between the two reference frames associated with the two reference picture lists, when the motion vector refinement value is greater than a threshold.

24. The method of claim 1, wherein the symmetry rule is responsive to a flag including MVD_L1_ZERO_FLAG in a slice level signaling for the video block.

25. A method of video bitstream processing, comprising:

receiving, for a first reference picture list associated with a video block, motion vector difference information for a first set of motion vectors; and

deriving, from the motion vector difference information for the first set of motion vectors, using a multi-hypothesis symmetry rule, a motion vector information associated with a second set of motion vectors for a second reference picture list associated with the video block, wherein the multi-hypothesis symmetry rule specifies that the second motion vector difference value is (0,0), and a corresponding motion vector predictor is set to a mirrored motion vector value derived from the first motion vector difference information; and

performing, using a result of the deriving, a conversion between the video block and a bitstream representation of the video block.

26. The method of claim 25, comprising:

deriving with the multi-hypothesis symmetry rule another motion vector difference information associated with the first reference picture list associated with the video block; and

deriving with the multi-hypothesis symmetry rule another motion vector difference information associated with the second reference picture list associated with the video block.

27. A method of processing a video bitstream, comprising:

receiving, for a video block, a first motion vector difference information associated with a first reference picture list;

receiving, for the video block, a second motion vector difference information associated with a second reference picture list;

deriving, from the first motion vector difference information and the second motion vector difference information, using a multi-hypothesis symmetry rule, a third motion vector difference information associated with the first reference picture list and a fourth motion vector difference information associated with the second reference picture list, wherein the multi-hypothesis symmetry rule specifies that the second motion vector difference value is (0,0), and a corresponding motion vector predictor is set to a mirrored motion vector value derived from the first motion vector difference information.

28. The method of any of claims 25 to 27, further including:

performing motion vector refinement of the mirrored motion vector value to generate a motion vector refinement value.

29. A method of video processing, comprising:

receiving a future frame of video relative to a reference frame of video;

receiving a motion vector related to the future frame of video and a past frame of video;

applying a predetermined relationship between the future frame of video and the past frame of video;

reconstructing the past frame of video based on the future frame of video, the motion vector, and the predetermined relationship between the past frame of video and the future frame of video, wherein the predetermined relationship is the future frame of video and the past frame of video are related by a mirror condition.

30. The method of claim 29, wherein the mirror condition means that an object with coordinates (x, y) in the future frame of video has coordinates (-x, -y) in the past frame of video.

31. A method of video processing, comprising:

receiving a past frame of video relative to a reference frame of video;

receiving a motion vector related to the past frame of video and a future frame of video;

applying a predetermined relationship between the future frame of video and the past frame of video;

reconstructing the future frame of video based on the past frame of video, the motion vector, and the predetermined relationship between the past frame of video and the future frame of video, wherein the predetermined relationship is the future frame of video and the past frame of video are related by a mirror condition.

32. The method of claim 31, wherein the mirror condition means that an object with coordinates (x, y) in the past frame of video has coordinates (-x, -y) in the future frame of video.

33. A video decoding apparatus comprising a processor configured to implement a method recited in one or more of claims 1 to 32.

34. A video encoding apparatus comprising a processor configured to implement a method recited in one or more of claims 1 to 32.

35. A computer program product having computer code stored thereon, the code, when executed by a processor, causes the processor to implement a method recited in one or more of claims 1 to 32.

36. A method, apparatus or system described in the present document.

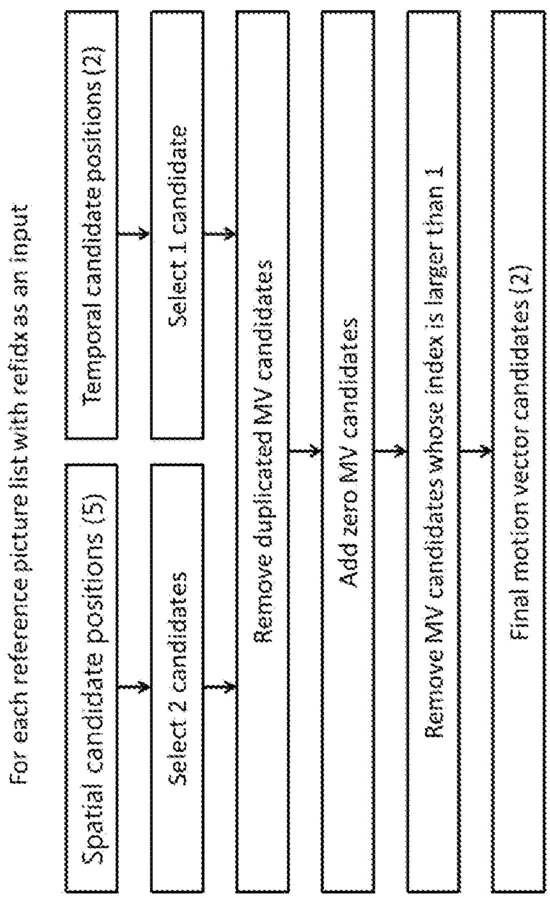

For each reference picture list with refidx as an input
Spatial candidate positions (5)
Temporal candidate positions (2)
Select 2 candidates
Select 1 candidate
Remove duplicated MV candidates
Add zero MV candidates
Remove MV candidates whose index is larger than 1
Final motion vector candidates (2)

FIG. 1

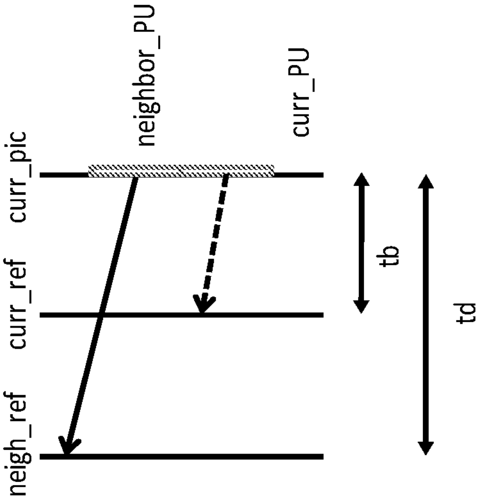

neighbor_PU
curr_PU
curr_pic
curr_ref
neigh_ref
tb
td

FIG. 3

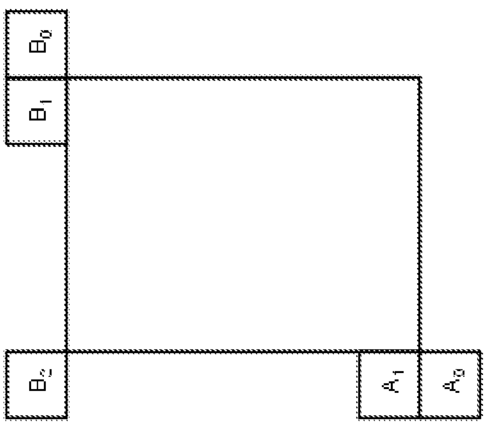

B_0
B_1
B_2
A_1
A_0

FIG. 2

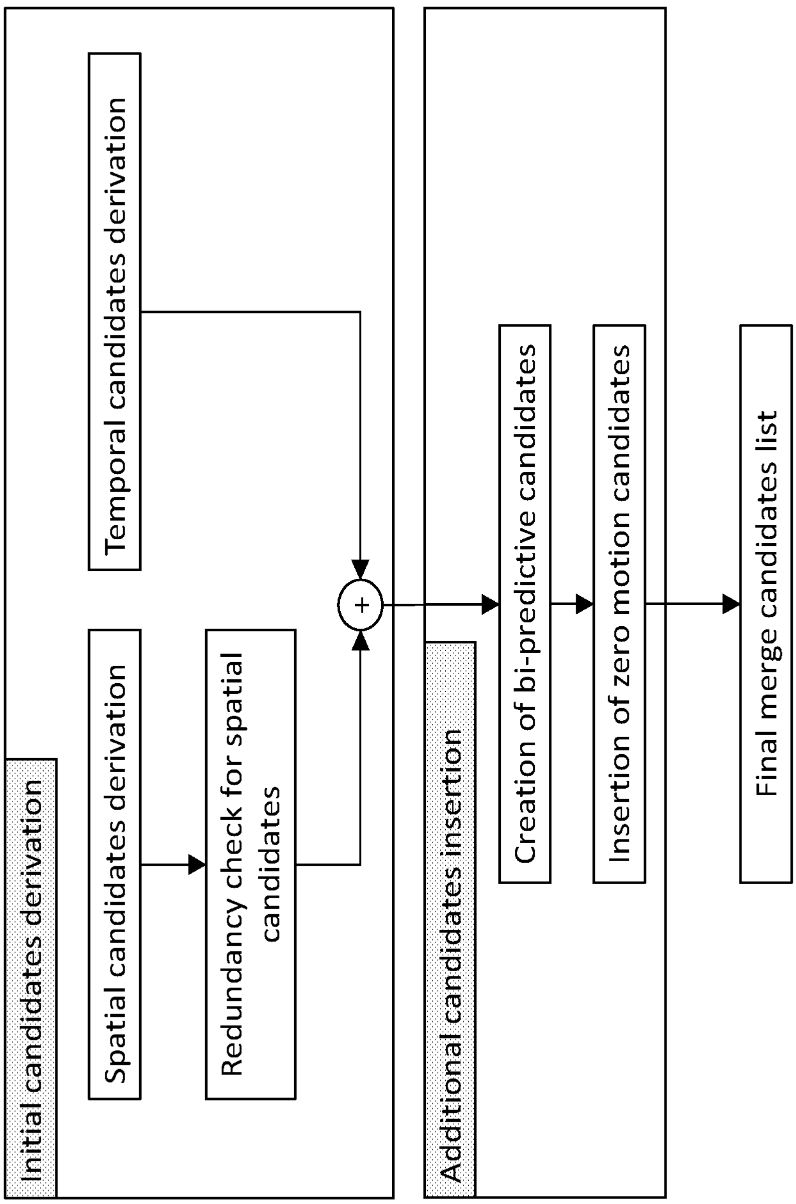

Initial candidates derivation
Spatial candidates derivation
Temporal candidates derivation
Redundancy check for spatial candidates
+
Additional candidates insertion
Creation of bi-predictive candidates
Insertion of zero motion candidates
Final merge candidates list

FIG. 4

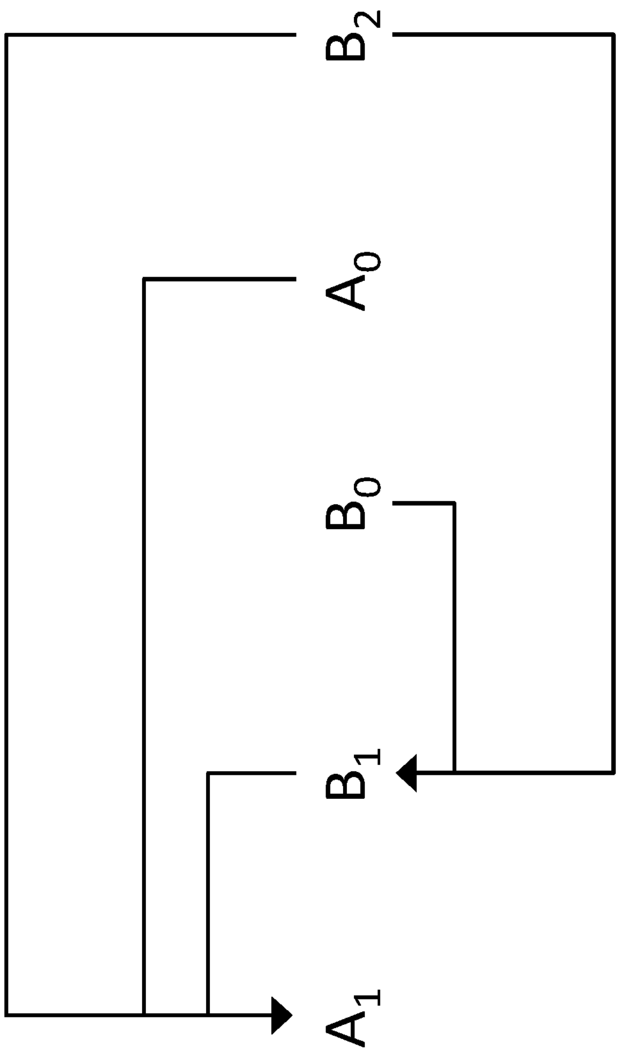
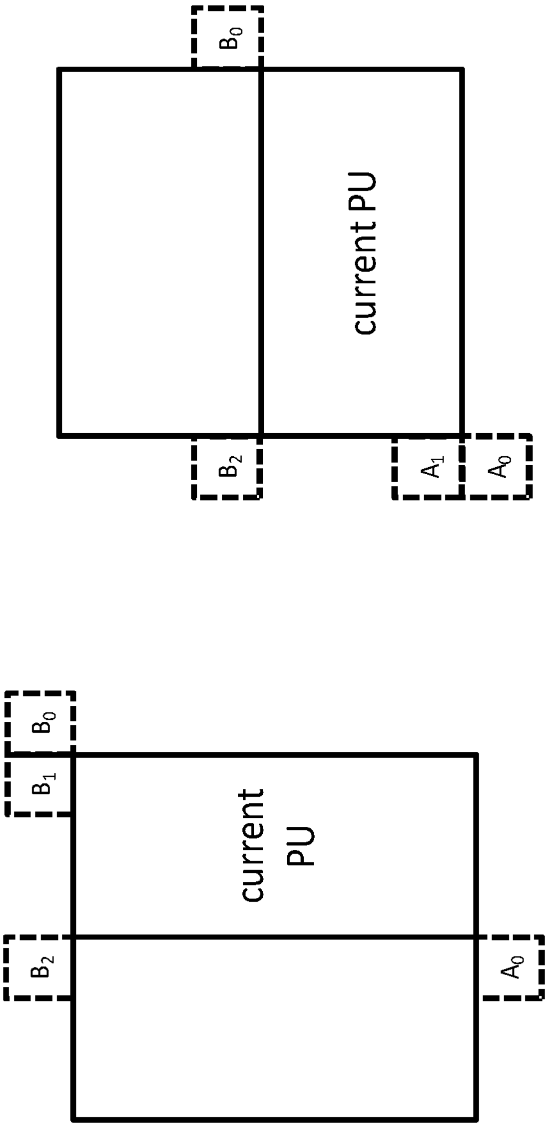


FIG. 5



(a) second PU of Nx2N

(b)second PU of 2NxN

FIG. 6

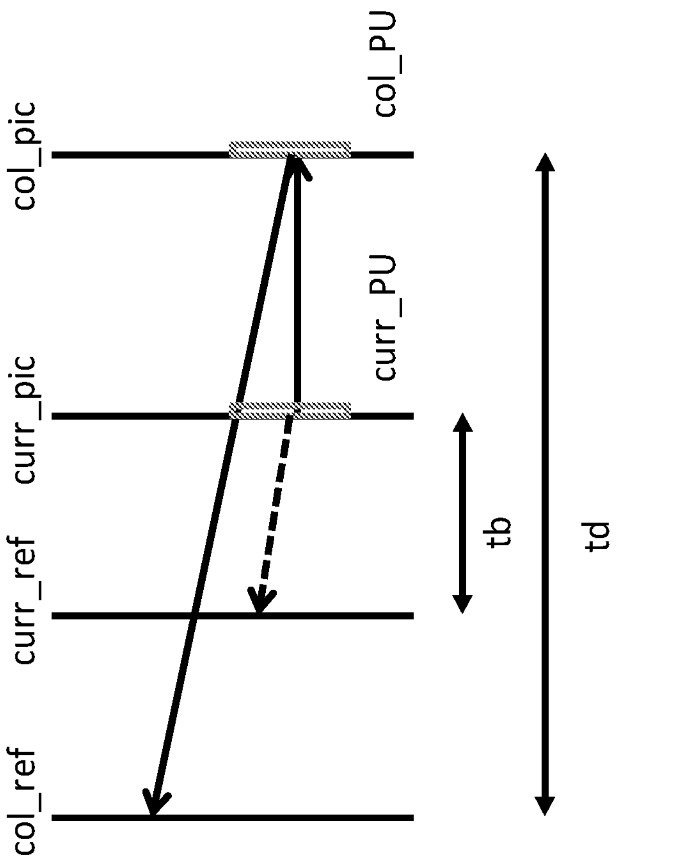


FIG. 7

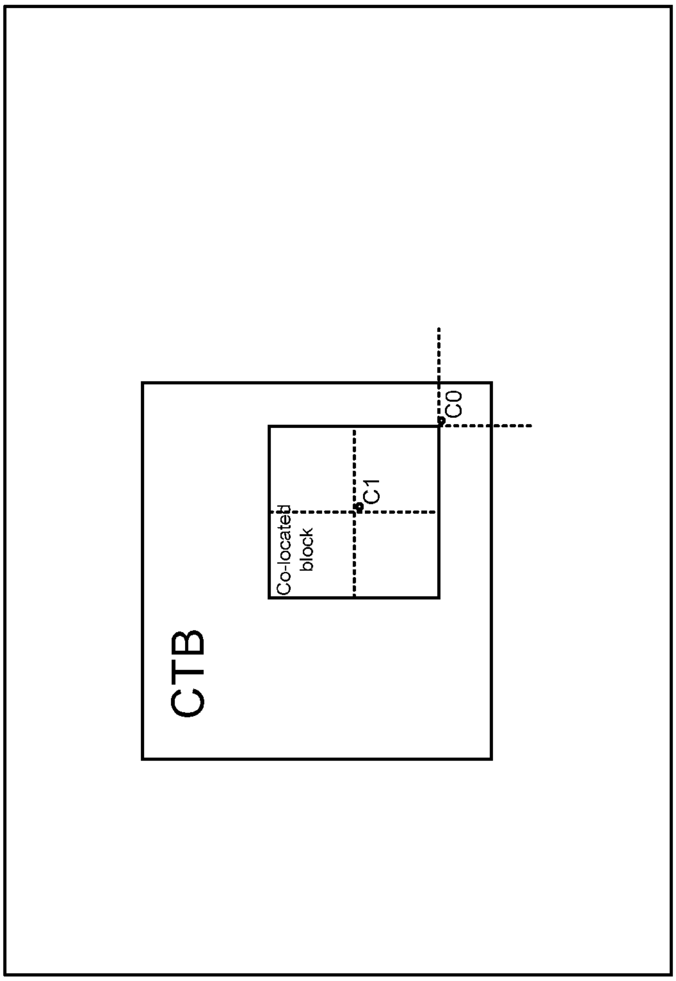


FIG. 8

Original Merge candidate list

Merge_idx	L0	L1
0	mvL0_A, ref0	-
1	-	mvL1_B, ref0
2		
3		
4		

Merge candidate list after adding combined candidates

Merge_idx	L0	L1
0	mvL0_A, ref0	combine
1		mvL1_B, ref0 combine
2	mvL0_A, ref0	mvL1_B, ref0
3		
4		

FIG. 9

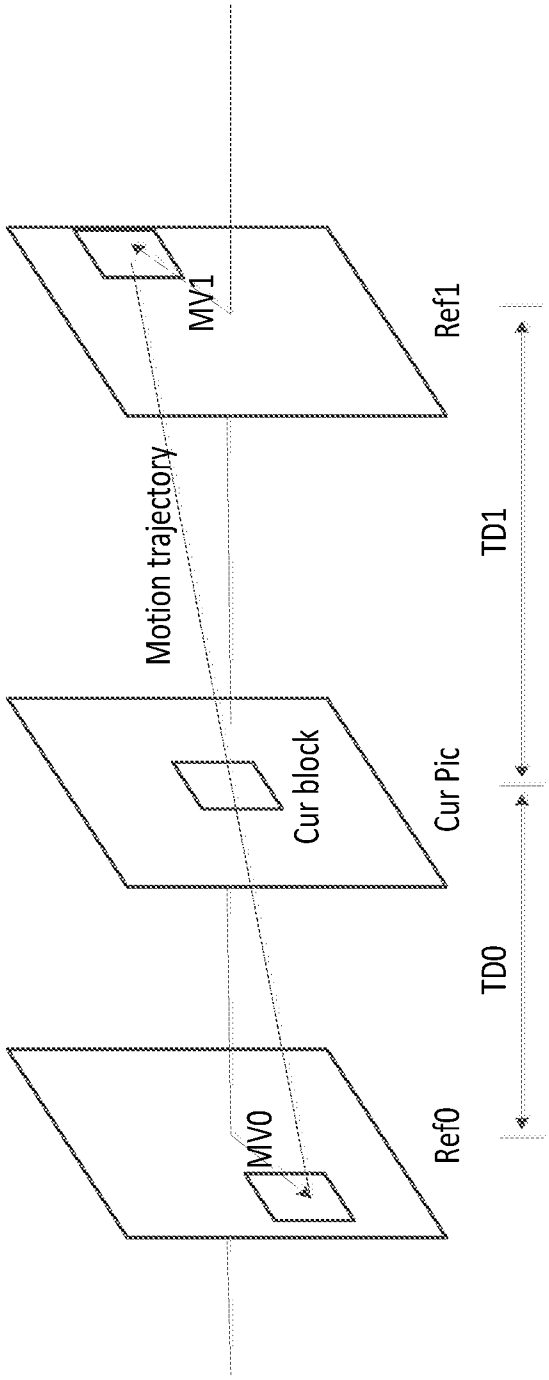

MV1
Ref1
Motion trajectory
TD1
Cur block
Cur Pic
TD0
MV0
Ref0

FIG. 10

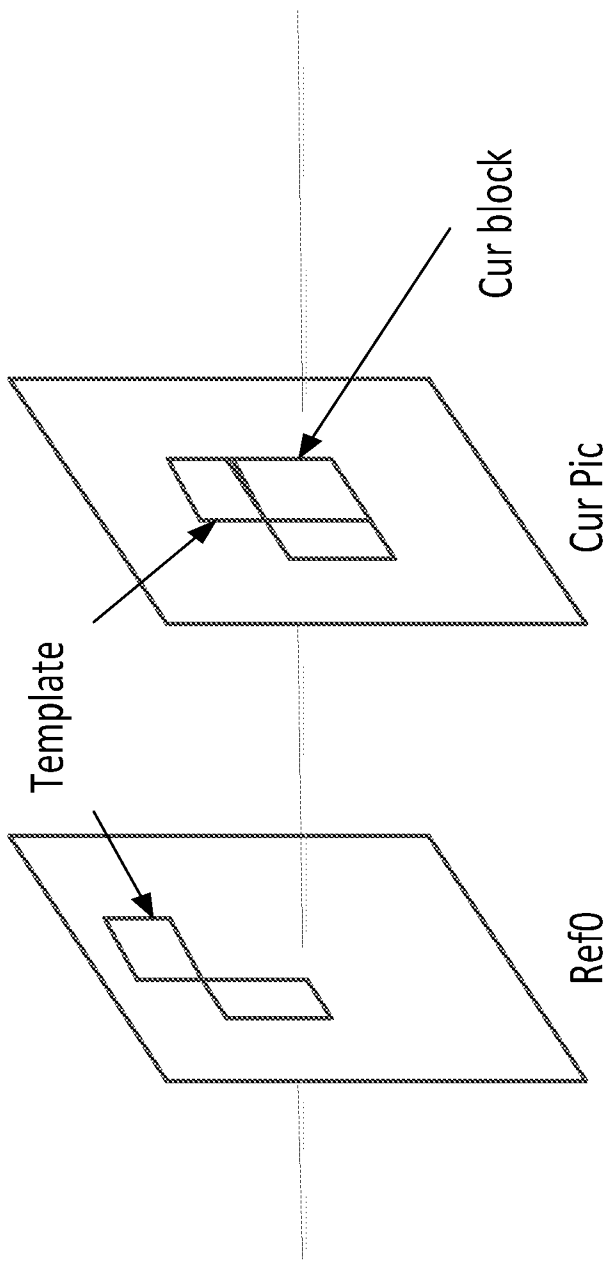


FIG. 11

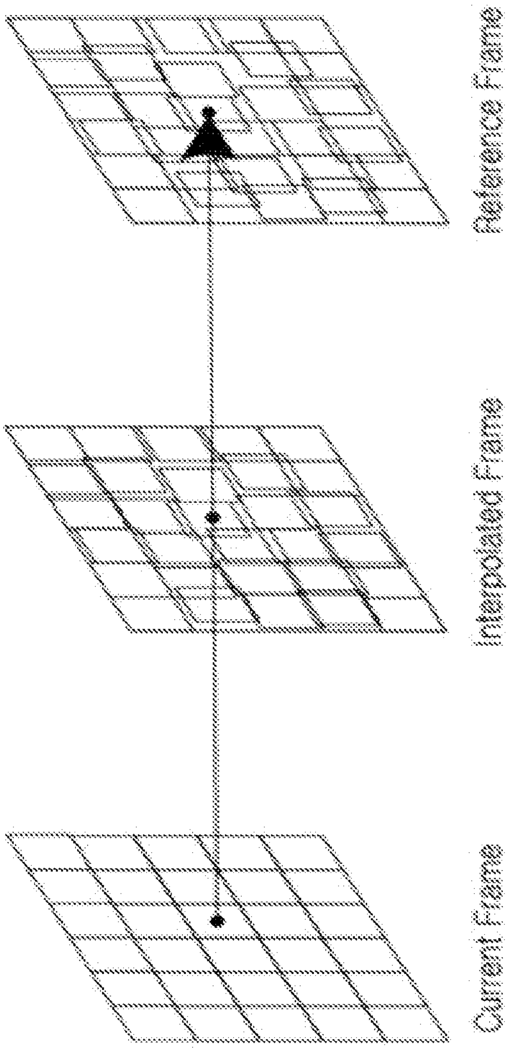
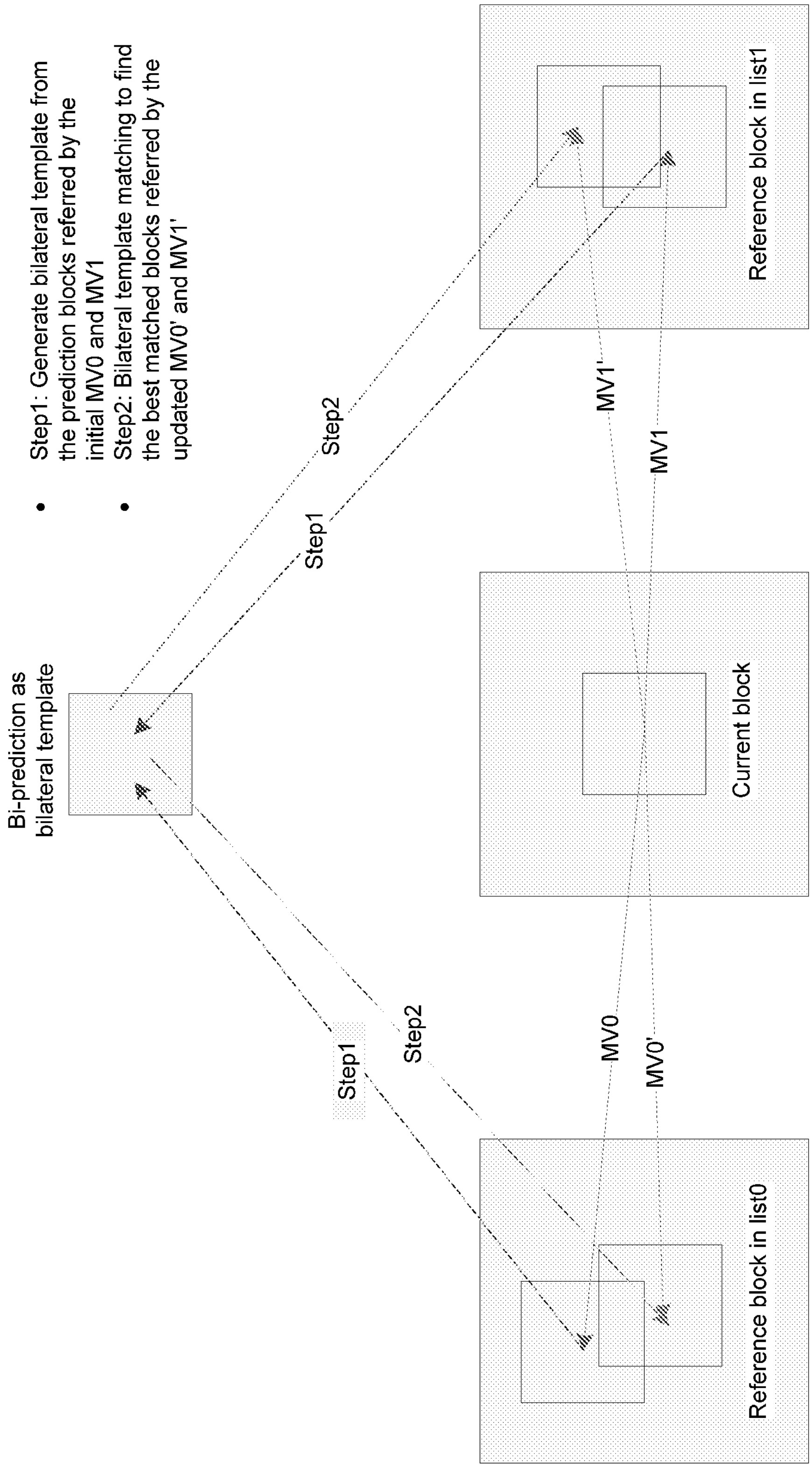


FIG. 12



Step1: Generate bilateral template from the prediction blocks referred by the initial MV0 and MV1
Step2: Bilateral template matching to find the best matched blocks referred by the updated MV0' and MV1'
Bi-prediction as bilateral template
Step1
Step2
Step1
Step2
MV0
MV0'
MV1'
MV1
Reference block in list0
Current block
Reference block in list1

FIG. 13

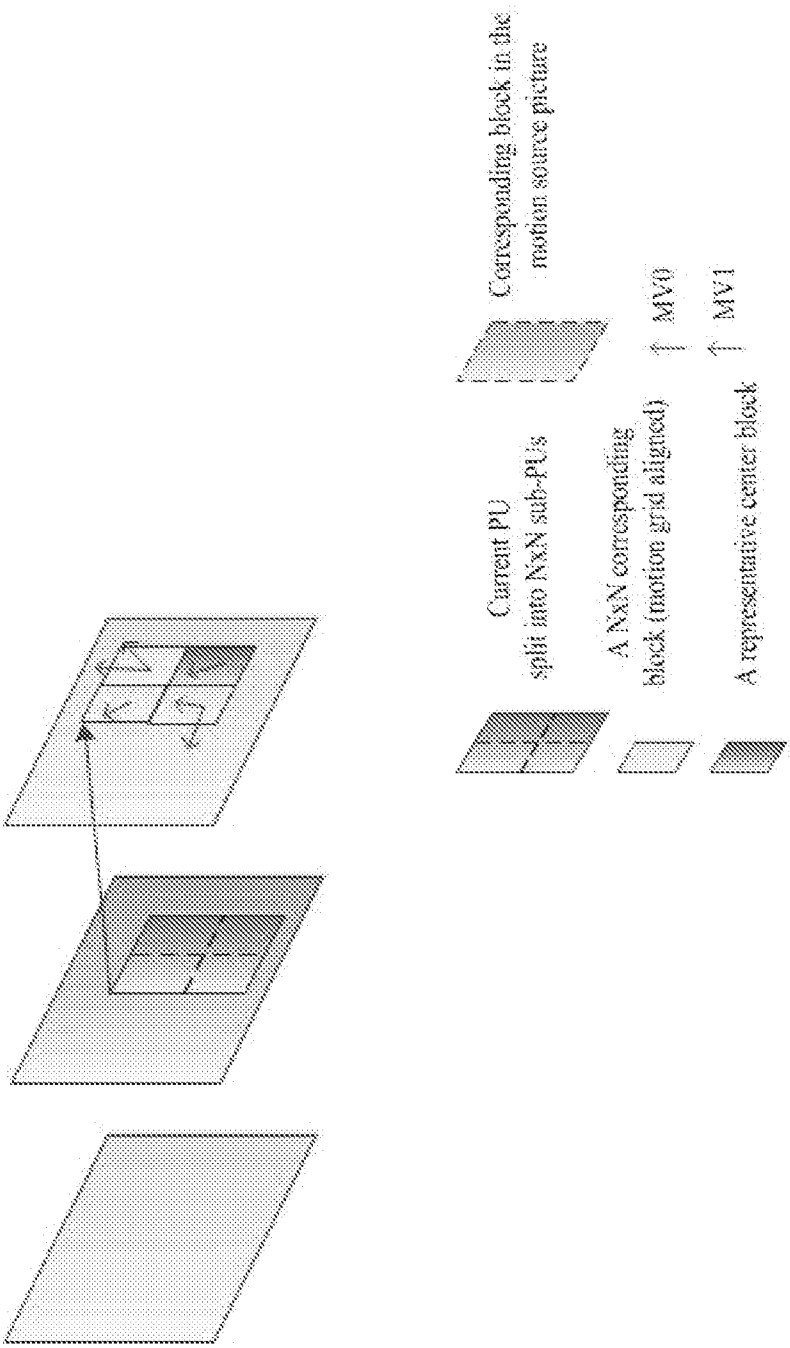

Current PU
split into NxN sub-PUs
A NxN corresponding
block (motion grid aligned)
A representative center block
Corresponding block in the
motion source picture
MV0
MV1

FIG. 14

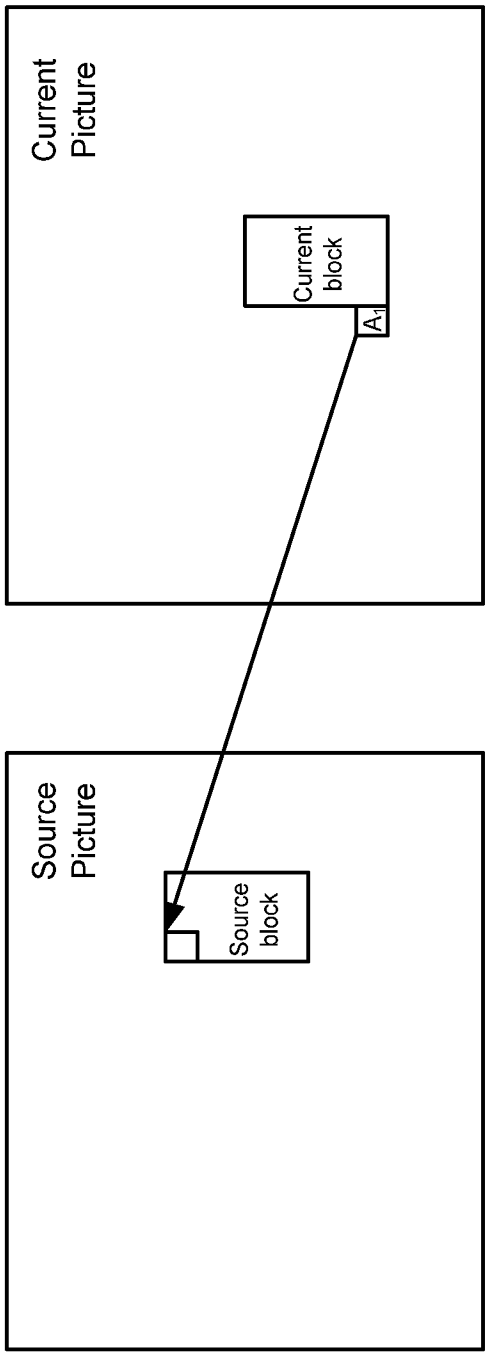

Current Picture
Current block
A_1
Source Picture
Source block

FIG. 15

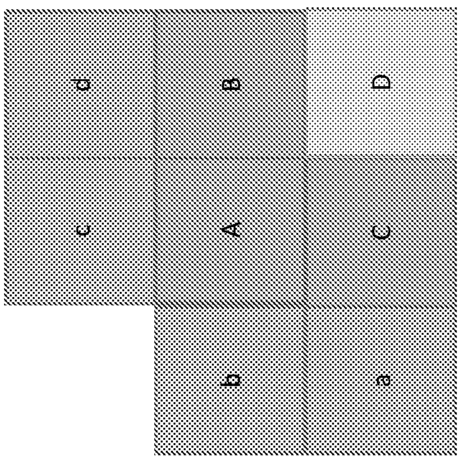


FIG. 16

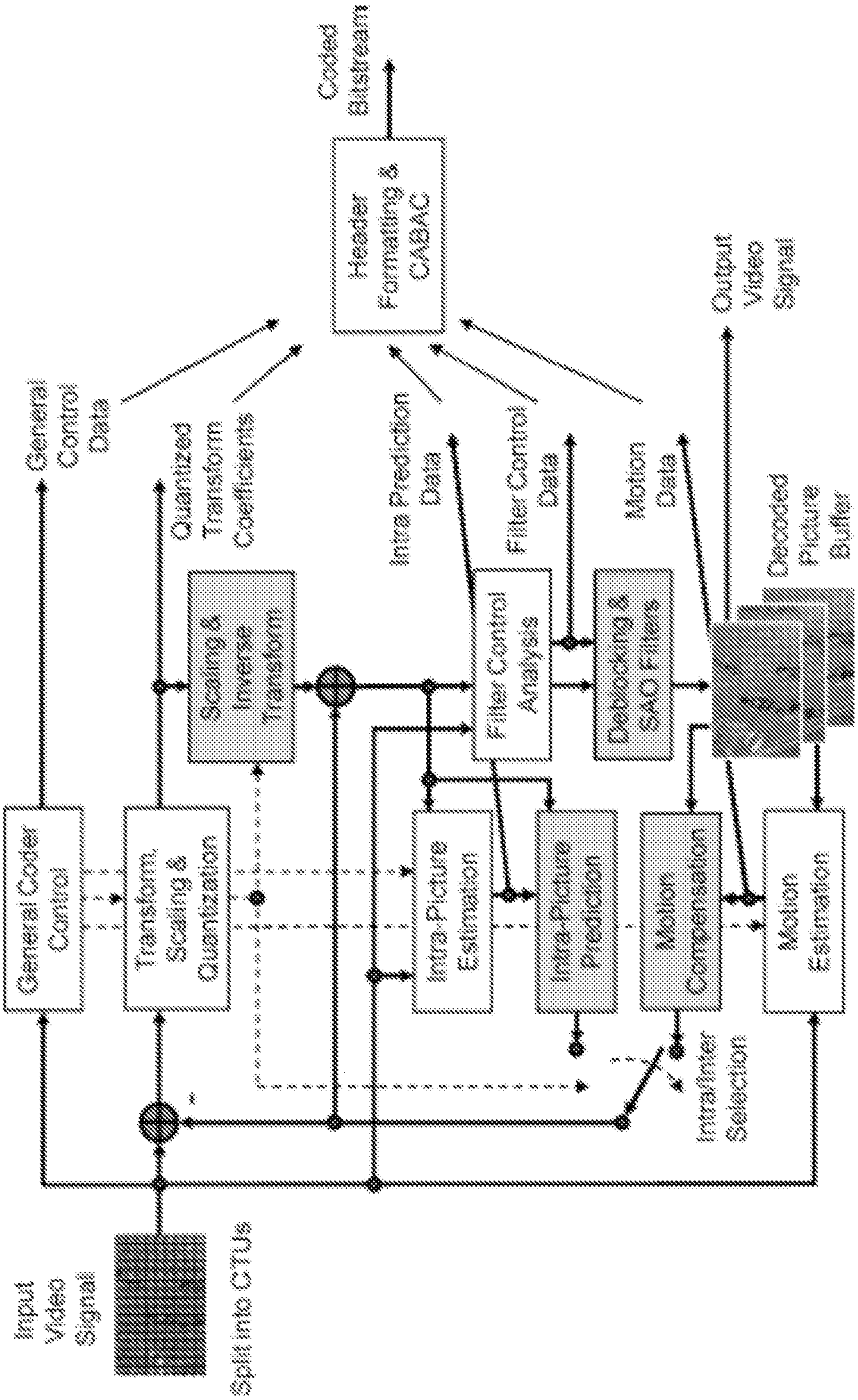

Input Video Signal
Split into CTUs
General Coder Control
Transform, Scaling & Quantization
Scaling & Inverse Transform
Intra-Picture Estimation
Intra-Picture Prediction
Motion Compensation
Motion Estimation
Intra/Inter Selection
Filter Control Analysis
Deblocking & SAO Filters
Decoded Picture Buffer
Header Formatting & CABAC
Coded Bitstream
General Control Data
Quantized Transform Coefficients
Intra Prediction Data
Filter Control Data
Motion Data
Output Video Signal

FIG. 17

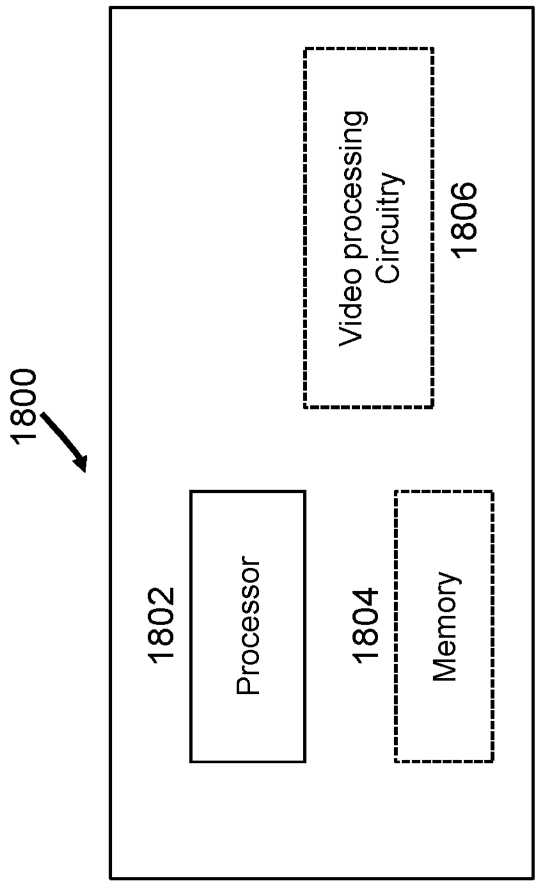

1800
1802
Processor
1804
Memory
Video processing Circuitry
1806

FIG. 18

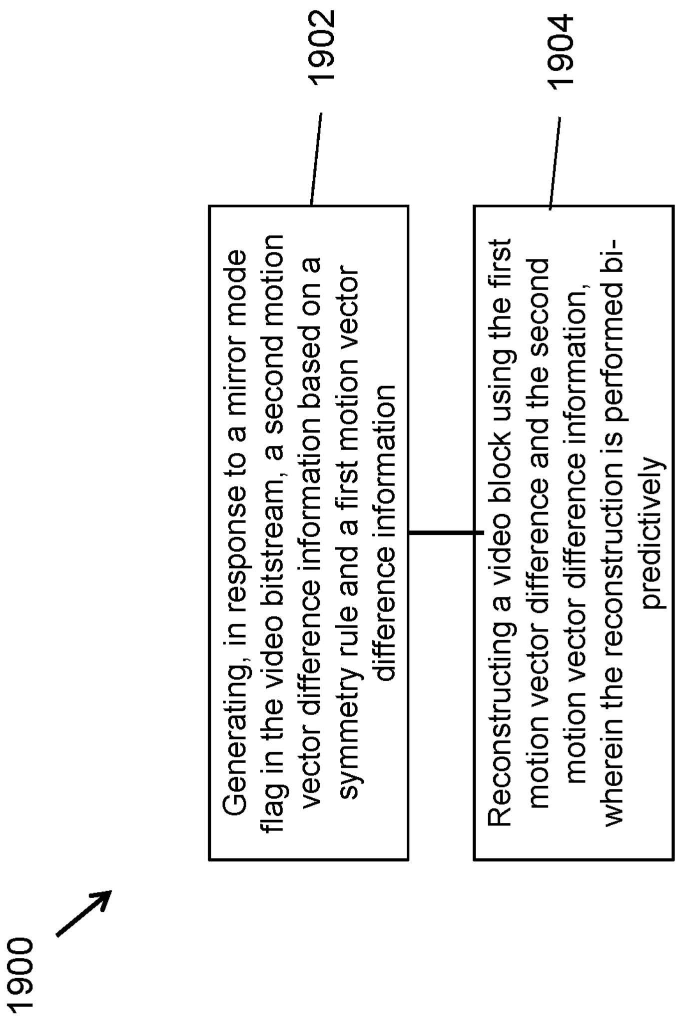

1900
1902
Generating, in response to a mirror mode flag in the video bitstream, a second motion vector difference information based on a symmetry rule and a first motion vector difference information
1904
Reconstructing a video block using the first motion vector difference and the second motion vector difference information, wherein the reconstruction is performed bi-predictively

FIG. 19

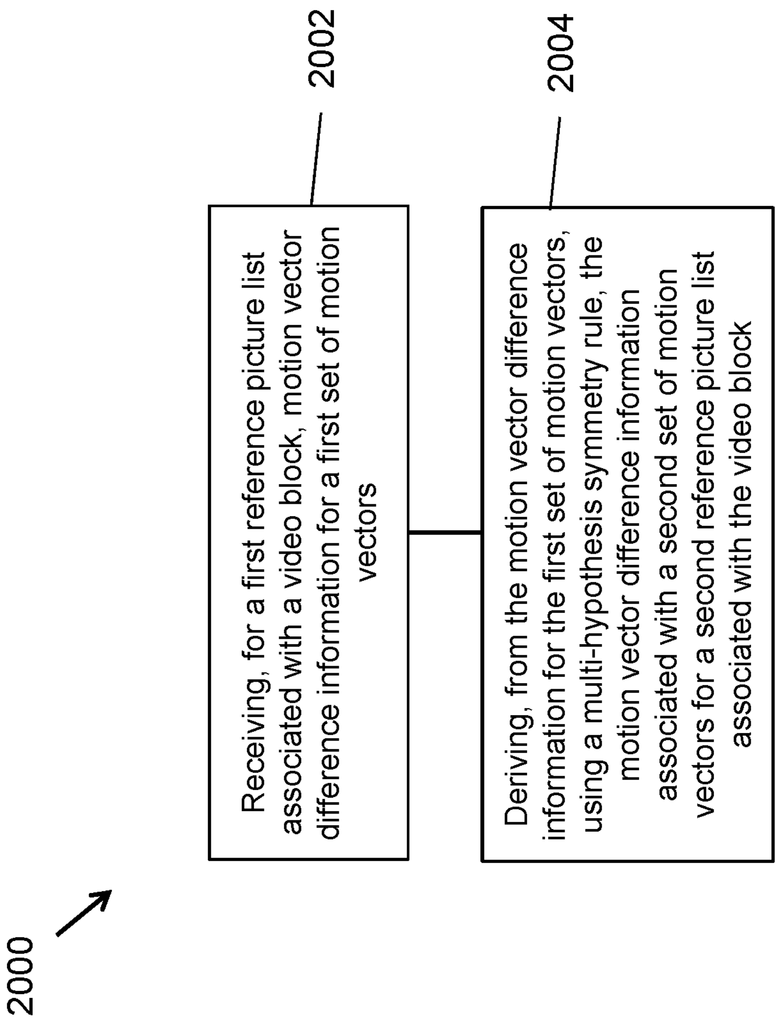

2000
2002
2004
Receiving, for a first reference picture list associated with a video block, motion vector difference information for a first set of motion vectors
Deriving, from the motion vector difference information for the first set of motion vectors, using a multi-hypothesis symmetry rule, the motion vector difference information associated with a second set of motion vectors for a second reference picture list associated with the video block

FIG. 20

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2019/055550

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/463 H04N19/513 H04N19/52 H04N19/577 H04N19/105
H04N19/139 H04N19/176
ADD.

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)

EPO-Internal, INSPEC, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	CHEN H ET AL: "Symmetrical mode for bi-prediction", 10. JVET MEETING; 10-4-2018 - 20-4-2018; SAN DIEGO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-J0063-v3, 15 April 2018 (2018-04-15), XP030151252, the whole document ----- -/--	1,2,4, 7-17,25, 27,29, 31,33-36 3,5,6, 18-24, 26,28, 30,32

[X] Further documents are listed in the continuation of Box C.

[ ] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
6 September 2019	16/09/2019

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Cyranka, Oliver

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Y-W CHEN ET AL: "Description of SDR, HDR and 360° video coding technology proposal by Qualcomm and Technicolor "" low and high complexity versions", 10. JVET MEETING; 10-4-2018 - 20-4-2018; SAN DIEGO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-J0021-v5, 14 April 2018 (2018-04-14), XP030151184,	1,3,5,6, 18-36
A	page 15 - page 18	2,4,7-17
X	SJÖBERG R ET AL: "Description of SDR and HDR video coding technology proposal by Ericsson and Nokia", 10. JVET MEETING; 10-4-2018 - 20-4-2018; SAN DIEGO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-J0012-v2, 13 April 2018 (2018-04-13), XP030151169,	1,24,25, 27,29, 31,33-36
A	page 5	2-23,26, 28,30,32
X,P	CHOI (SHARPLABS) B ET AL: "CE4-related: Bilateral Motion Vector Prediction", 11. JVET MEETING; 20180711 - 20180718; LJUBLJANA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), , no. JVET-K0302 10 July 2018 (2018-07-10), XP030199334, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end_user/documents/11_Ljubljana/wg11/JVET-K0302-v3.zip JVET-K0302-v3.doc [retrieved on 2018-07-10]	1,25,27, 29,31, 33-36
A,P	the whole document	2-24,26, 28,30,32

-/--

INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2019/055550

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	ESENLIK (HUAWEI) S ET AL: "CE9: DMVR with Motion Vector Difference Mirroring (Test 1.6)", 11. JVET MEETING; 20180711 - 20180718; LJUBLJANA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), , no. JVET-K0216 11 July 2018 (2018-07-11), XP030199427, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end_user/documents/11_Ljubljana/wg11/JVET-K0216-v4.zip JVET-K0216-v3.docx [retrieved on 2018-07-11] the whole document	1-36
X,P	YANG (INTERDIGITAL) H ET AL: "CE2: Symmetric MVD for affine bi-prediction coding (CE2-1.2)", 14. JVET MEETING; 20190319 - 20190327; GENEVA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), , no. JVET-N0319 13 March 2019 (2019-03-13), XP030202965, Retrieved from the Internet: URL:http://phenix.int-evry.fr/jvet/doc_end_user/documents/14_Geneva/wg11/JVET-N0319-v2.zip JVET-N0319_v1.docx [retrieved on 2019-03-13]	1,24,25, 27,29, 31,33-36
A,P	the whole document	2-23,26, 28,30,32