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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 July 2020 (16.07.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/143832 A1

(51) International Patent Classification:
H04N 19/105 (2014.01)

(21) International Application Number:
PCT/CN2020/071774

(22) International Filing Date:
13 January 2020 (13.01.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2019/071503
12 January 2019 (12.01.2019) CN
PCT/CN2019/077171
06 March 2019 (06.03.2019) 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: **LIU, Hongbin**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). **ZHANG, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **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,

(54) Title: BI-PREDICTION CONSTRAINTS

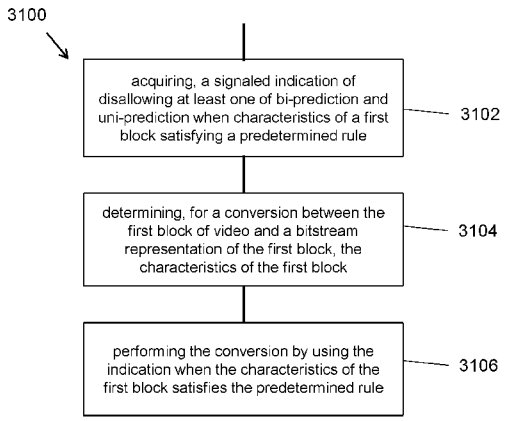


FIG. 31

(57) Abstract: Bi-prediction constraints are described. One example method includes acquiring, a signaled indication of disallowing at least one of bi-prediction and uni-prediction when the characteristics of a block satisfying a predetermined rule; determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; and performing the conversion by using the indication when the characteristics of the first block satisfies the predetermined rule.

WO 2020/143832 A1

[Continued on next page]

KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

BI-PREDICTION CONSTRAINTS

CROSS-REFERENCE TO RELATED APPLICATION

[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 benefits of International Patent Application No. PCT/CN2019/071503, filed on January 12, 2019, and No. PCT/CN2019/077171, filed on March 6, 2019. The entire disclosures of International Patent Application No. PCT/CN2019/071503 and No. PCT/CN2019/077171 are incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[002] This document is related to 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 video decoder or encoder embodiments for in which interpolation is improved using a block-shape interpolation order technique.

[005] In one example aspect, a method of video bitstream processing is disclosed. The method includes determining a shape of a first video block, determining an interpolation order based on the shape of the first video block, the interpolation order indicative of a sequence of performing a horizontal interpolation and a vertical interpolation, and performing the horizontal interpolation and the vertical interpolation for the first video block in the sequence in accordance with the interpolation order to reconstruct a decoded representation of the first video block.

[006] In another example aspect, a method of video bitstream processing includes determining characteristics of a motion vector related to a first video block, determining an interpolation order based on the characteristics of the motion vector, the interpolation order indicative of a sequence of performing a horizontal interpolation and a vertical interpolation, and performing the horizontal interpolation and the vertical interpolation for the first video block in the sequence in accordance with the interpolation order to reconstruct a decoded representation of the first video block.

[007] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, by a processor, dimension characteristics of a first video block; determining, by the processor, that a first interpolation filter is to be applied to the first video block based on the determination of the dimension characteristics; and performing further processing of the first video block using the first interpolation filter.

[008] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, by a processor, first characteristics of a first video block; determining, by the processor, that a first interpolation filter is to be applied to the first video block based on the first characteristics; performing further processing of the first video block using the first interpolation filter; determining, by a processor, second characteristics of a second video block; determining, by the processor, that a second interpolation filter is to be applied to the second video block based on the second characteristics, the first interpolation filter and the second interpolation filter being different short-tap filters; and performing further processing of the second video block using the second interpolation filter.

[009] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, by a processor, characteristics of a first video block, the characteristics including one or more of: a dimension information of a first video block, a prediction direction of the first video block, or a motion information of the first video block; rounding motion vectors (MVs) related to the first video block to integer-pel precision or half-pel precision based on the determination of the characteristics of the first video block; and performing further processing of the first video block using the motion vectors that are rounded.

[0010] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, by a processor, that a first video block is coded with a merge mode; rounding motion information related to the first video block to integer precision to generate modified motion information based on the determination that the first video block is coded with

the merge mode; and performing a motion compensation process for the first video block using the modified motion information.

[0011] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining characteristics of a first video block, the characteristics being one or both of: a size of the first video block, or a shape of the first video block; modifying motion vectors related to the first video block to integer-pel precision or half-pel precision to generate modified motion vectors; and performing further processing of the first video block using the modified motion vectors.

[0012] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining characteristics of a first video block, the characteristics being one or both of: a size dimension of the first video block, or a prediction direction of the first video block; determining MMVD side information based on the determination of the characteristics of the first video block; and performing further processing of the first video block using the MMVD side information.

[0013] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining characteristics of a first video block, the characteristics being one or both of: a size of the first video block, or a shape of the first video block; modifying motion vectors related to the first video block to integer-pel precision or half-pel precision to generate modified motion vectors; and performing further processing of the first video block using the modified motion vectors.

[0014] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining characteristics of a first video block, the characteristics being one or both of: a size of the first video block, or a shape of the first video block; determining a threshold number of half-pel motion vector (MV) components or quarter-pel MV components to be constrained based on the determination of the characteristics of the first video block; and performing further processing of the first video block using the threshold number.

[0015] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining characteristics of a first video block, the characteristics including a size of the first video block; modifying motion vectors (MVs) related to the first video block from fractional precision to integer precision based on the determination of the characteristics of the

first video block; and performing motion compensation for the first video block using the modified MVs.

[0016] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining a first dimension of a first video block; determining a first precision for motion vectors (MVs) related to the first video block based on the determination of the first dimension; determining a second dimension of a second video block, the first dimension and the second dimension being different dimensions; determining a second precision for MVs related to the second video block based on the determination of the second dimension, the first precision and the second precision being different precisions; and performing further processing of the first video block using the first dimension and of the second video block using the second dimension.

[0017] In another example aspect, a method of video processing, is disclosed. The method includes determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; determining filters with interpolation filter parameters used for interpolation of the first block based on the characteristics of the first block; and performing the conversion by using the filters with the interpolation filter parameters.

[0018] In another example aspect, a method of video processing, is disclosed. The method includes etching, for a conversion between a first block of video and a bitstream representation of the first block, reference pixels of a first reference block from reference picture, wherein the first reference block is smaller than a second reference block required for motion compensation of the first block; padding the first reference block with padding pixels to generate the second reference block; and performing the conversion by using the generated second reference block.

[0019] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; performing rounding process on motion vector (MV) of the first block based on the characteristics of the first block; and performing the conversion by using the rounded MV.

[0020] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; performing motion compensation for the first block using a MV with a first precision; and storing a MV with a second precision for the first block; wherein the first precision is different from the second precision.

[0021] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, for a conversion between a first block of video and a bitstream representation of the first block, coding mode of the first block; performing rounding process on motion vector (MV) of the first block if the coding mode of the first block satisfying a predetermined rule; and performing the motion compensation of the first block by using the rounded MV.

[0022] In another example aspect, a method for video bitstream processing is disclosed. The method includes generating, for a conversion between a first block of video and a bitstream representation of the first block, a first motion vector (MV) candidate list for the first block; performing rounding process on MV of at least one candidate before adding the at least one candidate into the first MV candidate list; and performing the conversion by using the first MV candidate list.

[0023] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; determining constraint parameter to be applied on the first block based on the characteristics of the first block, wherein the constraint parameter constrains a maximum number of fractional motion vector (MV) components of the first block; and performing the conversion by using the constraint parameter.

[0024] In another example aspect, a method for video bitstream processing is disclosed. The method includes

[0025] In another example aspect, a method for video bitstream processing is disclosed. The method includes acquiring, an a signaled indication of disallowing at least one of bi-prediction and uni-prediction when the characteristics of a block satisfying a predetermined rule; determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; and performing the conversion by using the indication when the characteristics of the first block satisfies the predetermined rule.

[0026] In another example aspect, a method for video bitstream processing is disclosed. The method includes signaling, an indication of disallowing at least one of bi-prediction and uni-prediction when the characteristics of a block satisfying a predetermined rule; determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; performing the conversion based on the characteristics of the first

block, wherein during the conversation, at least one of bi-prediction and uni-prediction is disabled when the characteristics of the first block satisfies a predetermined rule.

[0027] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, for a conversion between a first block of video and a bitstream representation of the first block, whether fractional motion vector (MV) or motion vector difference (MVD) precision is allowed for a first block; signaling, an Advanced Motion Vector Resolution (AMVR) parameter for the first block based on the determination; and performing the conversion by using the AMVR parameter.

[0028] In another example aspect, a method for video bitstream processing is disclosed. The method includes determining, for a conversion between a first block of video and a bitstream representation of the first block, whether fractional motion vector (MV) or motion vector difference (MVD) precision is allowed for a first block; acquiring, an Advanced Motion Vector Resolution (AMVR) parameter for the first block based on the determination; and performing the conversion by using the AMVR parameter.

[0029] In another example aspect, the above-described methods may be implemented by a video decoder apparatus that comprises a processor.

[0030] In another example aspect, the above-described methods may be implemented by a video encoder apparatus comprising a processor for decoding encoded video during video encoding process.

[0031] 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.

[0032] These, and other, aspects are further described in the present document.

BRIEF DESCRIPTION OF THE DRAWINGS

[0033] FIG. 1 is an illustration of a QUAD TREE BINARY TREE (QTBT) structure

[0034] FIG. 2 shows an example derivation process for merge candidates list construction.

[0035] FIG. 3 shows example positions of spatial merge candidates.

[0036] FIG. 4 shows an example of candidate pairs considered for redundancy check of spatial merge candidates.

[0037] FIG. 5A and 5B show examples of positions for the second prediction unit (PU) of N×2N and 2N×N partitions.

[0038] FIG. 6 is an illustration of motion vector scaling for temporal merge candidate.

[0039] FIG. 7 shows example candidate positions for temporal merge candidate, C0 and C1.

[0040] FIG. 8 shows an example of combined bi-predictive merge candidate.

[0041] FIG. 9 shows an example of a derivation process for motion vector prediction candidates.

[0042] FIG. 10 is an illustration of motion vector scaling for spatial motion vector candidate.

[0043] FIG. 11 shows an example of advanced temporal motion vector prediction (ATMVP) motion prediction for a coding unit (CU).

[0044] FIG. 12 shows an example of one CU with four sub-blocks (A-D) and its neighbouring blocks (a–d).

[0045] FIG. 13 illustrates proposed non-adjacent merge candidates in one example.

[0046] FIG. 14 illustrates proposed non-adjacent merge candidates in one example.

[0047] FIG. 15 illustrates proposed non-adjacent merge candidates in one example.

[0048] FIG. 16 shows an example of integer samples and fractional sample positions for quarter sample luma interpolation.

[0049] FIG. 17 is a block diagram of an example of a video processing apparatus.

[0050] FIG. 18 shows a block diagram of an example implementation of a video encoder.

[0051] FIG. 19 is a flowchart for an example of a video bitstream processing method.

[0052] FIG. 20 is a flowchart for an example of a video bitstream processing method.

[0053] FIG. 21 shows an example of repeat boundary pixels of a reference block before interpolation.

[0054] FIG. 22 is a flowchart for an example of a video bitstream processing method.

[0055] FIG. 23 is a flowchart for an example of a video bitstream processing method.

[0056] FIG. 24 is a flowchart for an example of a video bitstream processing method.

[0057] FIG. 25 is a flowchart for an example of a video bitstream processing method.

[0058] FIG. 26 is a flowchart for an example of a video bitstream processing method.

[0059] FIG. 27 is a flowchart for an example of a video bitstream processing method.

[0060] FIG. 28 is a flowchart for an example of a video bitstream processing method.

[0061] FIG. 29 is a flowchart for an example of a video bitstream processing method.

[0062] FIG. 30 is a flowchart for an example of a video bitstream processing method.

[0063] FIG. 31 is a flowchart for an example of a video bitstream processing method.

[0064] FIG. 32 is a flowchart for an example of a video bitstream processing method.

[0065] FIG. 33 is a flowchart for an example of a video bitstream processing method.

[0066] FIG. 34 is a flowchart for an example of a video bitstream processing method.

5 DETAILED DESCRIPTION

[0067] 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.

10 **[0068]** Section headings are used in the present document for ease of understanding and do not limit the embodiments and techniques to the corresponding sections. As such, embodiments from one section can be combined with embodiments from other sections.

[0069] 1. Summary

[0070] This invention is related to video coding technologies. Specifically, it is related to

15 interpolation in video coding. It may be applied to the existing video coding standard like HEVC, or the standard (Versatile Video Coding) to be finalized. It may be also applicable to future video coding standards or video codec.

[0071] 2. Background

[0072] Video coding standards have evolved primarily through the development of the well-

20 known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future

25 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

30 to HEVC.

[0073] FIG. 18 is a block diagram of an example implementation of a video encoder.

[0074] 2.1 Quadtree plus binary tree (QTBT) block structure with larger CTUs

[0075] In HEVC, a CTU is split into CUs by using a quadtree structure denoted as coding tree to adapt to various local characteristics. The decision whether to code a picture area using inter-picture (temporal) or intra-picture (spatial) prediction is made at the CU level. Each CU can be further split into one, two or four PUs according to the PU splitting type. Inside one PU, the same prediction process is applied and the relevant information is transmitted to the decoder on a PU basis. After obtaining the residual block by applying the prediction process based on the PU splitting type, a CU can be partitioned into transform units (TUs) according to another quadtree structure similar to the coding tree for the CU. One of key feature of the HEVC structure is that it has the multiple partition conceptions including CU, PU, and TU.

[0076] The QTBT structure removes the concepts of multiple partition types, i.e. it removes the separation of the CU, PU and TU concepts, and supports more flexibility for CU partition shapes. In the QTBT block structure, a CU can have either a square or rectangular shape. As shown in FIG. 1, a coding tree unit (CTU) is first partitioned by a quadtree structure. The quadtree leaf nodes are further partitioned by a binary tree structure. There are two splitting types, symmetric horizontal splitting and symmetric vertical splitting, in the binary tree splitting. The binary tree leaf nodes are called coding units (CUs), and that segmentation is used for prediction and transform processing without any further partitioning. This means that the CU, PU and TU have the same block size in the QTBT coding block structure. In the JEM, a CU sometimes consists of coding blocks (CBs) of different colour components, e.g. one CU contains one luma CB and two chroma CBs in the case of P and B slices of the 4:2:0 chroma format and sometimes consists of a CB of a single component, e.g., one CU contains only one luma CB or just two chroma CBs in the case of I slices.

[0077] The following parameters are defined for the QTBT partitioning scheme.

- CTU size: the root node size of a quadtree, the same concept as in HEVC
- *MinQTSize*: the minimum allowed quadtree leaf node size
- *MaxBTSize*: the maximum allowed binary tree root node size
- *MaxBTDepth*: the maximum allowed binary tree depth
- *MinBTSize*: the minimum allowed binary tree leaf node size

[0078] In one example of the QTBT partitioning structure, the CTU size is set as 128×128 luma samples with two corresponding 64×64 blocks of chroma samples, the *MinQTSize* is set as 16×16, the *MaxBTSize* is set as 64×64, the *MinBTSize* (for both width and height) is set as 4×4, and the *MaxBTDepth* is set as 4. The quadtree partitioning is applied to the CTU first to generate quadtree leaf nodes. The quadtree leaf nodes may have a size from 16×16 (i.e., the *MinQTSize*) to 128×128 (i.e., the CTU size). If the leaf quadtree node is 128×128, it will not be further split by the binary tree since the size exceeds the *MaxBTSize* (i.e., 64×64). Otherwise, the leaf quadtree node could be further partitioned by the binary tree. Therefore, the quadtree leaf node is also the root node for the binary tree and it has the binary tree depth as 0. When the binary tree depth reaches *MaxBTDepth* (i.e., 4), no further splitting is considered. When the binary tree node has width equal to *MinBTSize* (i.e., 4), no further horizontal splitting is considered. Similarly, when the binary tree node has height equal to *MinBTSize*, no further vertical splitting is considered. The leaf nodes of the binary tree are further processed by prediction and transform processing without any further partitioning. In the JEM, the maximum CTU size is 256×256 luma samples.

[0079] FIG. 1 illustrates an example of block partitioning by using QTBT, and FIG. 1 (right) illustrates the corresponding tree representation. The solid lines indicate quadtree splitting and dotted lines indicate binary tree splitting. In each splitting (i.e., non-leaf) node of the binary tree, one flag is signalled to indicate which splitting type (i.e., horizontal or vertical) is used, where 0 indicates horizontal splitting and 1 indicates vertical splitting. For the quadtree splitting, there is no need to indicate the splitting type since quadtree splitting always splits a block both horizontally and vertically to produce 4 sub-blocks with an equal size.

[0080] In addition, the QTBT scheme supports the ability for the luma and chroma to have a separate QTBT structure. Currently, for P and B slices, the luma and chroma CTBs in one CTU share the same QTBT structure. However, for I slices, the luma CTB is partitioned into CUs by a QTBT structure, and the chroma CTBs are partitioned into chroma CUs by another QTBT structure. This means that a CU in an I slice consists of a coding block of the luma component or coding blocks of two chroma components, and a CU in a P or B slice consists of coding blocks of all three colour components.

[0081] In HEVC, inter prediction for small blocks is restricted to reduce the memory access of motion compensation, such that bi-prediction is not supported for 4×8 and 8×4 blocks, and inter

prediction is not supported for 4×4 blocks. In the QTBT of the JEM, these restrictions are removed.

[0082] 2.2 Inter prediction in HEVC/H.265

[0083] Each inter-predicted PU has motion parameters for one or two reference picture lists. Motion parameters include a motion vector and a reference picture index. Usage of one of the two reference picture lists may also be signalled using *inter_pred_idc*. Motion vectors may be explicitly coded as deltas relative to predictors.

[0084] When a CU is coded with skip mode, one PU is associated with the CU, and there are no significant residual coefficients, no coded motion vector delta or reference picture index. A merge mode is specified whereby the motion parameters for the current PU are obtained from neighbouring PUs, including spatial and temporal candidates. The merge mode can be applied to any inter-predicted PU, not only for skip mode. The alternative to merge mode is the explicit transmission of motion parameters, where motion vector (to be more precise, motion vector difference compared to a motion vector predictor), corresponding reference picture index for each reference picture list and reference picture list usage are signalled explicitly per each PU. Such mode is named Advanced motion vector prediction (AMVP) in this disclosure.

[0085] 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.

[0086] 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.

[0087] The following text provides the details on the inter prediction modes specified in HEVC. The description will start with the merge mode.

[0088] 2.2.1 Merge Mode

[0089] 2.2.1.1 Derivation of candidates for merge mode

[0090] When a PU is predicted using merge mode, an index pointing to an entry in the *merge candidates list* is parsed from the bitstream and used to retrieve the motion information. The construction of this list is specified in the HEVC standard and can be summarized according to the following sequence of steps:

- Step 1: Initial candidates derivation

- Step 1.1: Spatial candidates derivation
- Step 1.2: Redundancy check for spatial candidates
- Step 1.3: Temporal candidates derivation

- Step 2: Additional candidates insertion
  - Step 2.1: Creation of bi-predictive candidates
  - Step 2.2: Insertion of zero motion candidates

[0091] These steps are also schematically depicted in FIG. 2. For spatial merge candidate derivation, a maximum of four merge candidates are selected among candidates that are located in five different positions. For temporal merge candidate derivation, a maximum of one merge candidate is selected among two candidates. Since constant number of candidates for each PU is assumed at decoder, additional candidates are generated when the number of candidates obtained from step 1 does not reach the maximum number of merge candidate (MaxNumMergeCand) which is signalled in slice header. Since the number of candidates is constant, index of best merge candidate is encoded using truncated unary binarization (TU). If the size of CU is equal to 8, all the PUs of the current CU share a single merge candidate list, which is identical to the merge candidate list of the 2N×2N prediction unit.

[0092] In the following, the operations associated with the aforementioned steps are detailed.

[0093] 2.2.1.2 Spatial candidates derivation

[0094] In the derivation of spatial merge candidates, a maximum of four merge candidates are selected among candidates located in the positions depicted in FIG. 3. The order of derivation is A_1, B_1, B_0, A_0 and B_2. Position B_2 is considered only when any PU of position A_1, B_1, B_0, A_0 is not available (e.g. because it belongs to another slice or tile) or is intra coded. After candidate at position A_1 is added, the addition of the remaining candidates is subject to a redundancy check which ensures that candidates with same motion information are excluded from the list so that coding efficiency is improved. To reduce computational complexity, not all possible candidate pairs are considered in the mentioned redundancy check. Instead only the pairs linked with an arrow in FIG. 4 are considered and a candidate is only added to the list if the corresponding candidate used for redundancy check has not the same motion information. Another source of duplicate motion information is the "*second PU*" associated with partitions different from 2Nx2N. As an example, FIG. 5 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 A_1 is not

considered for list construction. In fact, by adding this candidate will lead to two prediction units having the same motion information, which is redundant to just have one PU in a coding unit. Similarly, position B_1 is not considered when the current PU is partitioned as 2N×N.

[0095] 2.2.1.3 Temporal candidates derivation

[0096] 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 dashed line in FIG. 6, 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.

[0097] FIG. 6 is an illustration of motion vector scaling for temporal merge candidate.

[0098] 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. 7. If PU at position C_0 is not available, is intra coded, or is outside of the current CTU row, position C_1 is used. Otherwise, position C_0 is used in the derivation of the temporal merge candidate.

[0099] 2.2.1.4 Additional candidates insertion

[00100] 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. 8 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.

[00101] 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.

[00102] 2.2.1.5 Motion estimation regions for parallel processing

[00103] To speed up the encoding process, motion estimation can be performed in parallel whereby the motion vectors for all prediction units inside a given region are derived simultaneously. The derivation of merge candidates from spatial neighbourhood may interfere with parallel processing as one prediction unit cannot derive the motion parameters from an adjacent PU until its associated motion estimation is completed. To mitigate the trade-off between coding efficiency and processing latency, HEVC defines the motion estimation region (MER) whose size is signalled in the picture parameter set using the "log2_parallel_merge_level_minus2" syntax element. When a MER is defined, merge candidates falling in the same region are marked as unavailable and therefore not considered in the list construction.

[00104] 2.2.2 AMVP

[00105] AMVP exploits spatio-temporal correlation of motion vector with neighbouring PUs, which is used for explicit transmission of motion parameters. For each reference picture list, a motion vector candidate list is constructed by firstly checking availability of left, above temporally neighbouring PU positions, removing redundant candidates and adding zero vector to make the candidate list to be constant length. Then, the encoder can select the best predictor from the candidate list and transmit the corresponding index indicating the chosen candidate. Similarly with merge index signalling, the index of the best motion vector candidate is encoded using truncated unary. The maximum value to be encoded in this case is 2 (see FIG. 9). In the following sections, details about derivation process of motion vector prediction candidate are provided.

[00106] 2.2.2.1 Derivation of AMVP candidates

[00107] FIG. 9 summarizes derivation process for motion vector prediction candidate.

[00108] 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. 3.

[00109] 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.

[00110] 2.2.2.2 Spatial motion vector candidates

[00111] 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. 3, 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)

[00112] The no-spatial-scaling cases are checked first followed by the spatial scaling. Spatial scaling is considered when the POC is different between the reference picture of the neighbouring PU and that of the current PU regardless of reference picture list. If all PUs of left

candidates are not available or are intra coded, scaling for the above motion vector is allowed to help parallel derivation of left and above MV candidates. Otherwise, spatial scaling is not allowed for the above motion vector.

[00113] FIG. 10 is an illustration of motion vector scaling for spatial motion vector candidate.

[00114] In a spatial scaling process, the motion vector of the neighbouring PU is scaled in a similar manner as for temporal scaling, as depicted as FIG. 10. 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.

[00115] 2.2.2.3 Temporal motion vector candidates

[00116] 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. 7). The reference picture index is signalled to the decoder.

[00117] 2.3 New inter merge candidates in JEM

[00118] 2.3.1 Sub-CU based motion vector prediction

[00119] In the JEM with QTBT, each CU can have at most one set of motion parameters for each prediction direction. Two sub-CU level motion vector prediction methods are considered in the encoder by splitting a large CU into sub-CUs and deriving motion information for all the sub-CUs of the large CU. Alternative temporal motion vector prediction (ATMVP) method allows each CU to fetch multiple sets of motion information from multiple blocks smaller than the current CU in the collocated reference picture. In spatial-temporal motion vector prediction (STMVP) method motion vectors of the sub-CUs are derived recursively by using the temporal motion vector predictor and spatial neighbouring motion vector.

[00120] To preserve more accurate motion field for sub-CU motion prediction, the motion compression for the reference frames is currently disabled.

[00121] 2.3.1.1 Alternative temporal motion vector prediction

[00122] In the alternative temporal motion vector prediction (ATMVP) method, the motion vectors temporal motion vector prediction (TMVP) is modified by fetching multiple sets of motion information (including motion vectors and reference indices) from blocks smaller than the current CU. As shown in FIG. 11, the sub-CUs are square N×N blocks (N is set to 4 by default).

[00123] ATMVP predicts the motion vectors of the sub-CUs within a CU in two steps. The first step is to identify the corresponding block in a reference picture with a so-called temporal vector. The reference picture is called the motion source picture. The second step is to split the current CU into sub-CUs and obtain the motion vectors as well as the reference indices of each sub-CU from the block corresponding to each sub-CU, as shown in FIG. 11.

[00124] In the first step, a reference picture and the corresponding block is determined by the motion information of the spatial neighbouring blocks of the current CU. To avoid the repetitive scanning process of neighbouring blocks, the first merge candidate in the merge candidate list of the current CU is used. The first available motion vector as well as its associated reference index are set to be the ***temporal vector*** and the index to the motion source picture. This way, in ATMVP, the corresponding block may be more accurately identified, compared with TMVP, wherein the corresponding block (sometimes called collocated block) is always in a bottom-right or center position relative to the current CU.

[00125] In the second step, a corresponding block of the sub-CU is identified by the temporal vector in the motion source picture, by adding to the coordinate of the current CU the temporal vector. For each sub-CU, the motion information of its corresponding block (the smallest motion grid that covers the center sample) is used to derive the motion information for the sub-CU. After the motion information of a corresponding N×N block is identified, it is converted to the motion vectors and reference indices of the current sub-CU, in the same way as TMVP of HEVC, wherein motion scaling and other procedures apply. For example, the decoder checks whether the low-delay condition (i.e. the POCs of all reference pictures of the current picture are smaller than the POC of the current picture) is fulfilled and possibly uses motion vector MV_x (the motion vector corresponding to reference picture list X) to predict motion vector MV_y (with X being equal to 0 or 1 and Y being equal to 1−X) for each sub-CU.

[00126] 2.3.1.2 Spatial-temporal motion vector prediction

[00127] In this method, the motion vectors of the sub-CUs are derived recursively, following raster scan order. FIG. 12 illustrates this concept. Let us consider an 8×8 CU which contains four 4×4 sub-CUs A, B, C, and D. The neighbouring 4×4 blocks in the current frame are labelled as a, b, c, and d.

[00128] The motion derivation for sub-CU A starts by identifying its two spatial neighbours. The first neighbour is the N×N block above sub-CU A (block c). If this block c is not available

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

[00129] 2.3.1.3 Sub-CU motion prediction mode signalling

[00130] 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.

[00131] In the JEM, all bins of merge index is context coded by CABAC. While in HEVC, only the first bin is context coded and the remaining bins are context by-pass coded.

[00132] 2.3.2 Non-adjacent merge candidates

[00133] Qualcomm proposes to derive additional spatial merge candidates from non-adjacent neighboring positions which are marked as 6 to 49 as in FIG. 13. The derived candidates are added after TMVP candidates in the merge candidate list.

[00134] Tencent proposes to derive additional spatial merge candidates from positions in an outer reference area which has an offset of (-96, -96) to the current block.

[00135] As shown in FIG. 14, the positions are marked as A(i,j), B(i,j), C(i,j), D(i,j) and E(i,j). Each candidate B (i, j) or C (i, j) has an offset of 16 in the vertical direction compared to its previous B or C candidates. Each candidate A (i, j) or D (i, j) has an offset of 16 in the horizontal direction compared to its previous A or D candidates. Each E (i, j) has an offset of 16 in both horizontal direction and vertical direction compared to its previous E candidates. The candidates

are checked from inside to the outside. And the order of the candidates is A (i, j), B (i, j), C (i, j), D (i, j), and E (i, j). To further study whether the number of merge candidates can be further reduced. The candidates are added after TMVP candidates in the merge candidate list.

[00136] In some examples, the extended spatial positions from 6 to 27 as in FIG. 15 are checked according to their numerical order after the temporal candidate. To save the MV line buffer, all the spatial candidates are restricted within two CTU lines.

[00137] 2.4 Intra prediction in JEM

[00138] 2.4.1 Intra mode coding with 67 intra prediction modes

[00139] For the luma interpolation filtering, an 8-tap separable DCT-based interpolation filter is used for 2/4 precision samples and a 7-tap separable DCT-based interpolation filter is used for 1/4 precisions samples, as shown in Table 1.

Table 1: 8-tap DCT-IF coefficients for 1/4th luma interpolation.

Position	Filter coefficients
1/4	{ -1, 4, -10, 58, 17, -5, 1 }
2/4	{ -1, 4, -11, 40, 40, -11, 4, -1 }
3/4	{ 1, -5, 17, 58, -10, 4, -1 }

[00140] Similarly, a 4-tap separable DCT-based interpolation filter is used for the chroma interpolation filter, as shown in Table 2.

Table 2: 4-tap DCT-IF coefficients for 1/8th chroma interpolation.

Position	Filter coefficients
1/8	{ -2, 58, 10, -2 }

2/8	{ -4, 54, 16, -2 }
3/8	{ -6, 46, 28, -4 }
4/8	{ -4, 36, 36, -4 }
5/8	{ -4, 28, 46, -6 }
6/8	{ -2, 16, 54, -4 }
7/8	{ -2, 10, 58, -2 }

[00141] For the vertical interpolation for 4:2:2 and the horizontal and vertical interpolation for 4:4:4 chroma channels, the odd positions in Table 2 are not used, resulting in 1/4th chroma interpolation.

[00142] For the bi-directional prediction, the bit-depth of the output of the interpolation filter is maintained to 14-bit accuracy, regardless of the source bit-depth, before the averaging of the two prediction signals. The actual averaging process is done implicitly with the bit-depth reduction process as:

[00143] predSamples[x, y]= (predSamplesL0[x, y] + predSamplesL1[x, y] + offset) >> shift

[00144] where shift = (15 − BitDepth) and offset = 1 << (shift − 1)

[00145] If both horizonal component and vertical component of a motion vector point to sub-pixel positions, horizonal interpolation is always performed firstly, and then the vertical interpolation is performed. For example, to interpolate the subpixel $j_{0,0}$ shown in FIG. 16, first, $b_{0,k}$ (k = -3, -2, … 3) is interpolated according to equation 2-1, then $j_{0,0}$ is interpolated according to equation 2-2. Here, shift1 = Min(4, BitDepthY − 8), and shift2 = 6.

[00146] $b_{0,k} = (-A_{-3,k} + 4 * A_{-2,k} - 11 * A_{-1,k} + 40 * A_{0,k} + 40 * A_{1,k} - 11 * A_{2,k} + 4 * A_{3,k} - A_{4,k}) >> shift1$ (2-1)

[00147] $j_{0,0} = (-b_{0,-3} + 4 * b_{0,-2} - 11 * b_{0,-1} + 40 * b_{0,0} + 40 * b_{0,1} - 11 * b_{0,2} + 4 * b_{0,3} - b_{0,4}) >> shift2$ (2-2)

[00148] Alternatively, we can first perform vertical interpolation and then perform horizonal interpolation. In this case, to interpolation $j_{0,0}$, first, $h_{k,0}$ (k = -3, -2, … 3) is interpolated according to equation 2-3, then, $j_{0,0}$ is interpolated according to equation 2-4. When BitDepthY is smaller than or equal to 8, shift1 is 0, nothing is lost in the first interpolation stage, therefore, the final interpolation result is not changed by the interpolation order. However, when BitDepthY is greater than 8, shift1 is greater than 0. In this case, the final interpolation result can be different when different interpolation orders are applied.

[00149] $h_{k,0} = (-A_{k,-3} + 4 * A_{k,-2} - 11 * A_{k,-1} + 40 * A_{k,0} + 40 * A_{k,1} - 11 * A_{k,2} + 4 * A_{k,3} - A_{k,4}) >> shift1$ (2-3)

[00150] $j_{0,0} = (-h_{-3,0} + 4 * h_{-2,0} - 11 * h_{-1,0} + 40 * h_{0,0} + 40 * h_{1,0} - 11 * h_{2,0} + 4 * h_{3,0} - h_{4,0}) >> shift2$ (2-4)

[00151] 3. Examples of Problems solved by embodiments

[00152] For luma block size WxH, if we always perform horizonal interpolation firstly, the required interpolation (per pixel) is shown in Table 3.

Table 3: interpolation required for WxH luma component by HEVC/JEM

Horizonal component of MV	Vertical component of MV	interpolation
1/4 pel	1/4 pel	((H + 6) x W + W x H) / (W x H) = 2 + 6 / H
1/4 pel	1/2 pel	((H + 7) x W + W x H) / (W x H) = 2 + 7 / H
1/2 pel	1/4 pel	((H + 6) x W + W x H) / (W x H) = 2 + 6 / H
1/2 pel	1/2 pel	((H + 7) x W + W x H) / (W x H) = 2 + 7 / H

[00153] On the other hand, if we perform vertical interpolation firstly, the required interpolation is shown in Table 4. Apparently, the optimal interpolation order is the one which requires smaller interpolation times between Table 3 and Table 4.

Table 4: interpolation required for WxH luma component when the interpolation order is reversed.

Horizonal component of MV	Vertical component of MV	interpolation
1/4 pel	1/4 pel	((W + 6) x H + W x H) / (W x H) = 2 + 6 / W
1/4 pel	1/2 pel	((W + 6) x H + W x H) / (W x H) = 2 + 6 / W
1/2 pel	1/4 pel	((W + 7) x H + W x H) / (W x H) = 2 + 7 / W
1/2 pel	1/2 pel	((W + 7) x H + W x H) / (W x H) = 2 + 7 / W

[00154] For chroma component, if we always perform horizonal interpolation firstly, the required interpolation is ((H + 3) x W + W x H) / (W x H) = 2 + 3 / H. if we always perform vertical interpolation firstly, the required interpolation is ((W + 3) x H + W x H) / (W x H) = 2 + 3 / W.

[00155] As mentioned above, different interpolation order can lead to different interpolation result when bitdepth of the input video is greater than 8. Therefore, the interpolation order shall be defined implicitly in both encoder and decoder.

[00156] 4. Examples of embodiments

[00157] To tackle the problems, and provide other benefits, we propose shape dependent interpolation order. Suppose the interpolation filter tap (in motion compensation) is N (for example, 8, 6, 4, or 2), and the current block size is WxH.

[00158] Suppose the number of allowed MVD in MMVD (such as the number of entry to the distance table) is M. Note that triangle mode is considered as a bi-prediction mode, and the following techniques related to bi-prediction may be applied to triangle mode too.

[00159] The detailed examples below should be considered as examples to explain general concepts. These examples should not be interpreted in a narrow way. Furthermore, these examples can be combined in any manner.

1. It is proposed that the interpolation order depends on the current coding block shape (e.g., the coding block is a CU).
    a. In one example, for block (such as CU, PU or sub-block used in sub-block based prediction like affine, ATMVP or BIO) with width > height, vertical interpolation is firstly performed, and then horizonal interpolation is performed, e.g., pixels $d_{k,0}$, $h_{k,0}$ and $n_{k,0}$ are firstly interpolated and $e_{0,0}$ to $r_{0,0}$ are then interpolated. An example of $j_{0,0}$ is shown in equation 2-3 and 2-4.
        i. Alternatively, for a block (such as CU, PU or sub-block used in sub-block based prediction like affine, ATMVP or BIO) with width >= height, vertical interpolation is firstly performed, and then horizonal interpolation is performed
    b. In one example, for a block (such as CU, PU or sub-block used in sub-block based prediction like affine, ATMVP or BIO) with width <= height, horizonal interpolation is firstly performed, and then vertical interpolation is performed.
        i. Alternatively, for a block (such as CU, PU or sub-block used in sub-block based prediction like affine, ATMVP or BIO) with width < height, horizonal interpolation is firstly performed, and then vertical interpolation is performed
    c. In one example, both the luma component and the chroma components follow the same interpolation order.
    d. Alternatively, when one chroma coding block corresponds to multiple luma coding blocks (e.g., for 4:2:0 color format, one chroma 4x4 block may correspond to two 8x4 or 4x8 luma blocks), luma and chroma may use different interpolation orders.

   e. In one example, when different interpolation orders are utilized, the scaling factors in the multiple stages (i.e., shift1 and shift2) may be further changed accordingly.
2. Alternatively, in addition, it is proposed that the interpolation order of luma component can further depend on the MV.
   a. In one example, if the vertical MV component points to a quarter-pel position and the horizonal MV component points to a half-pel position, horizonal interpolation is firstly performed, and then vertical interpolation is performed.
   b. In one example, if the vertical MV component points to a half-pel position and the horizonal MV component points to a quarter-pel position, vertical interpolation is firstly performed, and then horizonal interpolation is performed.
   c. In one example, the proposed methods are only applied to square coding blocks.
3. It is proposed that for a block coded with merge mode (e.g., regular merge list, triangular merge list, affine merge list, or other non-intra/non-AMVP modes), the associated motion information may be modified to integer precision (e.g., via rounding) before invoking motion compensation process.
   a. Alternatively, merge candidates with fractional merge candidates may be excluded from the merge list.
   b. Alternatively, when a merge candidate derived from spatial or temporal blocks or other ways (such as HMVP, pairwise bi-prediction merge candidates) is associated with fractional motion vectors, the fractional motion vectors may be firstly modified to integer precision (e.g., via rounding) before being added to the merge list.
   c. In one example, a separate HMVP table may be kept on-the-fly to store motion candidates with integer precisions.
   d. Alternatively, the above methods may be only applied when the merge candidate is a bi-prediction candidate.
   e. In one example, the above methods may be applied to certain block dimensions, such as 4x16, 16x4, 4x8, 8x4, 4x4.
   f. In one example, the above methods may be applied to the AMVP coded blocks wherein the merge candidate may be replaced by an AMVP candidate.
   g. In one example, the above methods may be applied to certain block modes, such as non-affine mode.

4. It is proposed that the MMVD side information (such as distance table, directions) may be dependent on block dimension and/or prediction direction (e.g., uni-prediction or bi-prediction).
   a. In one example, a distance table with all integer precisions may be defined or signaled.
   b. In one example, if the base merge candidate is associated with motion vectors of fractional precision, it may be firstly modified (such as via rounding) to integer precision and then used to derive the final motion vectors for motion compensation.
5. It is proposed that MV in MMVD mode may be constrained to be with integer-pel precision or half-pel precision for some block sizes or block shapes.

a. In one example, if integer-pel precision is selected for an MMVD coded block, the base merge candidates used in MMVD may be firstly modified to integer-pel precision (such as via rounding).
b. In one example, if half-pel precision is selected for an MMVD coded block, the base merge candidates used in MMVD may be modified to half-pel precision (such as via rounding).
    i. In one example, rounding may be performed in the base merge list construction process, therefore, rounded MVs are used in pruning.
    ii. In one example, rounding may be performed after the base merge list construction process, therefore, unrounded MVs are used in pruning.
c. In one example, if integer-pel precision or half-pel precision is used for MMVD mode, only MVD with same or lower precision are allowed.
    i. For example, if integer-pel precision is used for MMVD mode, only integer-pel precision, 2-pel precision or N-pel precision (N >= 1) MVD are allowed.
d. In one example, if K MVD are not allowed in MMVD mode, binarization of MVD index may be modified because the maximum MVD index is M – K – 1 instead of M – 1. Meanwhile, different context may be used in CABAC coding.
e. In one example, rounding may be performed after deriving the MV in MMVD mode.
f. The constraint may be different for bi-prediction and uni-prediction. For example, the constraint may be not applied in uni-prediction.
g. The constraint may be different for different block sizes or block shapes.

6. It is proposed that the maximum number of half-pel MV components or/and quarter-pel MV components (e.g., horizonal MV or vertical MV) may be constrained for some block sizes or block shapes.
    a. In one example, bitstream shall conform to the constraint.
    b. The constraint may be different for bi-prediction and uni-prediction. For example, the constraint may be not applied in uni-prediction.
        i. For example, such constraint may be applied to bi-predicted 4x8 or/and 8x4 or/and 4x16 or/and 16x4 block, however, it may be not applied to uni-predicted 4x8 or/and 8x4 or/and 4x16 or/and 16x4 block.
        ii. For example, such constraint may be applied to both bi-predicted and uni-predicted 4x4 block.
    c. The constraint may be different for different block sizes or block shapes.
    d. The constraint may be applied to triangle mode.
        i. For example, such constraint may be applied to 4x16 or/and 16x4 block coded in triangle mode.
    e. In one example, for bi-predicted blocks, at most 3 quarter-pel MV components may be allowed.
    f. In one example, for bi-predicted blocks, at most 2 quarter-pel MV components may be allowed.
    g. In one example, for bi-predicted blocks, at most 1 quarter-pel MV components may be allowed.

h. In one example, for bi-predicted blocks, at most 0 quarter-pel MV components may be allowed.
i. In one example, for uni-predicted blocks, at most 1 quarter-pel MV components may be allowed.
j. In one example, for uni-predicted blocks, at most 0 quarter-pel MV components may be allowed.
k. In one example, for bi-predicted blocks, at most 3 fractional MV components may be allowed.
l. In one example, for bi-predicted blocks, at most 2 fractional MV components may be allowed.
m. In one example, for bi-predicted blocks, at most 1 fractional MV components may be allowed.
n. In one example, for bi-predicted blocks, at most 0 fractional MV components may be allowed.
o. In one example, for uni-predicted blocks, at most 1 fractional MV components may be allowed.
p. In one example, for uni-predicted blocks, at most 0 fractional MV components may be allowed.

7. It is proposed that some components of a MV may be rounded to integer-pel precision or half-pel precision depending on the dimension (e.g., width and/or height, ratios of width and height), or/and prediction direction or/and motion information of a block.

a. In one example, MV is rounded to the nearest integer-pel precision MV or/and half-pel precision MV.
b. In one example, different rounding method may be used. For example, rounding down, rounding up, rounding towards zero or rounding away from zero may be used.
c. In one example, if the size (i.e., width * height) of a block is smaller (or larger than) than (and/or equal to) a threshold L (e.g., L=16 or 64), MV rounding may be applied to horizonal or/and vertical MV component.
d. In one example, if the width (or height) of a block is smaller than (and/or equal to) a threshold L1 (e.g., L1 = 4, 8), MV rounding may be applied to horizonal (or vertical) MV component.
e. In one example, thresholds L and L1 may be different for bi-predicted blocks and uni-predicted blocks. For example, smaller thresholds may be used for bi-predicted blocks.
f. In one example, if the ratio between width and height is larger than a first threshold or smaller than a second threshold (such as for narrow blocks like 4x16 or 16x4), MV rounding may be applied.
g. In one example, MV rounding may be applied only when both horizonal and vertical components of the MV are fractional, i.e., they point to fractional pixel position instead of integer pixel position.

h. Whether MV rounding is applied or not may depend on whether the current block is bi-predicted or uni-predicted.
    i. For example, MV rounding may be applied only when the current block is bi-predicted.
i. Whether MV rounding is applied or not may depend on the prediction direction (e.g., from List 0 or list 1) and/or the associated motion vectors. In one example, for bi-predicted blocks, whether MV rounding is applied or not may be different for different prediction directions.
    i. In one example, if MV of prediction direction X (X = 0 or 1) have fractional components in both horizonal and vertical directions, then MV rounding may be applied to N MV components for prediction direction X; otherwise, MV rounding may be not applied. Here, N = 0, 1 or 2.
    ii. In one example, if N (N >= 0) MV components are with fractional precision, MV rounding may be applied to M (0 <= M <= N) of the N MV components.
        1. N and M may be different for bi-predicted blocks and uni-predicted blocks.
        2. N and M may be different for different block sizes (width or/and height or/and width * height).
        3. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 4.
        4. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 3.
        5. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 2.
        6. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 1.
        7. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 3.
        8. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 2.
        9. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 1.
        10. For example, for bi-predicted blocks, N is equal to 2 and M is equal to 2.
        11. For example, for bi-predicted blocks, N is equal to 2 and M is equal to 1.
        12. For example, for bi-predicted blocks, N is equal to 1 and M is equal to 1.
        13. For example, for uni-predicted blocks, N is equal to 2 and M is equal to 2.
        14. For example, for uni-predicted blocks, N is equal to 2 and M is equal to 1.

15. For example, for uni-predicted blocks, N is equal to 1 and M is equal to 1.

iii. In one example, K of the M MV components are rounded to integer-pel precision and M – K MV components are rounded to half-pel precision, wherein K = 0, 1, …, M – 1.

j. Whether MV rounding is applied or not may be different for different color components such as Y, Cb and Cr.

i. For example, whether to and how to apply MV rounding may depend on color formats such as 4:2:0, 4:2:2 or 4:4:4.

k. Whether and/or how MV rounding is applied or not may depend on the block size (or width, height), block shapes, prediction direction etc.

i. In one example, some MV components of 4x16 or/and 16x4 bi-predicted or/and uni-predicted luma blocks may be rounded to half-pel precision.

ii. In one example, some MV components of 4x16 or/and 16x4 bi-predicted or/and uni-predicted luma blocks may be rounded to integer-pel precision.

iii. In one example, some MV components of of 4x4 uni-predicted or/and bi-predicted luma blocks may be rounded to integer-pel precision.

iv. In one example, some MV components of 4x8 or/and 8x4 bi-predicted or/and uni-predicted luma blocks may be rounded to integer-pel precision.

l. In one example, the MV rounding may be not applied on sub-block prediction, such as affine prediction.

i. In an alternative example, the MV rounding may be applied on sub-block prediction, such as ATMVP prediction. In such a case, each sub-block is treated as a coding block to judge whether and how to apply MV rounding.

8. It is proposed that for certain block sizes, motion vectors of one block shall be modified to integer precision before being utilized for motion compensation, for example, if they are fractional precisions.

9. In one example, for certain block dimensions, the stored motion vectors and those utilized for motion compensation may be in different precisions.

a. In one example, sub-pel precision (a.k.a., fractional precision, such as ¼-pel, 1/16-pel) may be stored for blocks with certain block dimensions, but the motion compensation process is based on integer version of those motion vectors (such as via rounding).

10. It is proposed that an indication of disallowing bi-prediction for certain block dimensions may be signaled in sequence parameter set/picture parameter set/sequence header/picture header/tile header/tile group header/CTU rows/regions/other high-level syntax.

a. Alternatively, an indication of disallowing bi-prediction for certain block dimensions may be signaled in sequence parameter set/picture parameter set/sequence header/picture header/tile header/tile group header/CTU rows/regions/other high-level syntax.

b. Alternatively, an indication of disallowing bi-prediction and/or uni-prediction for certain block dimensions may be signaled in sequence parameter set/picture

parameter set/sequence header/picture header/tile header/tile group header/CTU rows/regions/other high-level syntax.

c. Alternatively, furthermore, such indications may be only applied to certain modes, such as non-affine mode.

d. Alternatively, furthermore, when uni-/bi-prediction is disallowed for a block, the signaling of AMVR indices may be modified accordingly, such as only integer-pel precisions are allowed, or different MV precisions may be utilized instead.

e. Alternatively, furthermore, above methods (such as bullets 3-9) may be also applicable.

11. It is proposed that a conformance bitstream shall follow the rule that for certain block dimensions, only integer-pel motion vectors are allowed for bi-prediction coded blocks.
    a. It is proposed that a conformance bitstream shall follow the rule that for certain block dimensions, only integer-pel motion vectors are allowed for bi-prediction coded blocks.

12. Signaling of AMVR flag may depend on whether fractional motion vectors are allowed for a block.
    a. In one example, if fractional (i.e., 1/4-pel) MV/MVD precision is disallowed for a block, the flag indicating whether MV/MVD precision of the current block is 1/4-pel may be skipped and derived to be false implicitly.

13. In one example, the block dimensions mentioned above are, for example, 4x16, 16x4, 4x8, 8x4, 4x4.

14. It is proposed that filters with different interpolation filters (e.g., different filter taps, and/or different filter interpolation filter coefficients) may be used in interpolation depending on the dimension (e.g., width and/or height, ratios of width and height) of a block.
    a. Different filters may be used for vertical interpolation and horizontal interpolation. For example, shorter tap filter may be applied for vertical interpolation compared to that for horizontal interpolation.
    b. In one example, interpolation filters with less taps than the interpolation filters in VTM-3.0 may be applied in some cases. These interpolation filters with less taps are also called “short-tap filters”.
    c. In one example, if the size (i.e., width * height) of a block is smaller (or larger than) than (and/or equal to) a threshold L (e.g., L=16 or 64), different filters (e.g., short-tap filters) may be used for horizonal or/and vertical interpolation.
    d. In one example, if the width (or height) of a block is smaller than (and/or equal to) a threshold L1 (e.g., L1 = 4, 8), different filters (e.g., short-tap filters) may be used for horizonal (or vertical) interpolation.
    e. In one example, if the ratio between width and height is larger than a first threshold or smaller than a second threshold (such as for narrow blocks like 4x16 or 16x4), a different filter from those used for other kinds of blocks (e.g., short-tap filter) may be selected.
    f. In one example, the short-tap filters may be used only when both horizonal and vertical components of the MV are fractional, i.e., they point to fractional pixel position instead of integer pixel position.

g. Which filter to be used (e.g., the short-tap filters may be used or not) may depend on whether the current block is bi-predicted or uni-predicted.
    i. For example, the short-tap filters may be used only when the current block is bi-predicted.
h. Which filter to be used (e.g., the short-tap filters may be used or not) may depend on the prediction direction (e.g., from List 0 or list 1) and/or the associated motion vectors. In one example, for bi-predicted blocks, whether short-tap filters are used or not may be different for different prediction direction.
    i. In one example, if MV of prediction direction X (X = 0 or 1) have fractional components in both horizonal and vertical directions, then short-tap filters are used for prediction direction X; otherwise, short-tap filters are not used.
    ii. In one example, if N (N >= 0) MV components are with fractional precision, short-tap filter may be applied to M (0 <= M <= N) of the N MV components.
        1. N and M may be different for bi-predicted blocks and uni-predicted blocks.
        2. N and M may be different for different block sizes (width or/and height or/and width * height).
        3. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 4.
        4. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 3.
        5. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 2.
        6. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 1.
        7. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 3.
        8. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 2.
        9. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 1.
        10. For example, for bi-predicted blocks, N is equal to 2 and M is equal to 2.
        11. For example, for bi-predicted blocks, N is equal to 2 and M is equal to 1.
        12. For example, for bi-predicted blocks, N is equal to 1 and M is equal to 1.
        13. For example, for uni-predicted blocks, N is equal to 2 and M is equal to 2.
        14. For example, for uni-predicted blocks, N is equal to 2 and M is equal to 1.

15. For example, for uni-predicted blocks, N is equal to 1 and M is equal to 1.

iii. Different short-tap filters may be used for the M MV components.
1. In one example, K of the M MV components use S1-tap filter, and M – K MV components use S2-tap filter, wherein K = 0, 1, …, M – 1. For example, S1 is equal to 6 and S2 is equal to 4.

i. In one example, different filters (e.g., the short-tap filters) may be used only for some pixels. For example, they are used only for boundary pixels of the block.
i. For example, they are only used for the N1 right column or/and N2 left column or/and N3 top row or/and N4 bottom row of the block.

j. Whether short-tap filters are used or not may be different for uni-predicted blocks and bi-predicted blocks.

k. Whether short-tap filters are used or not may be different for different color components such as Y, Cb and Cr.
i. For example, whether to and how to apply short-tap filters may depend on color formats such as 4:2:0, 4:2:2 or 4:4:4.

l. Different short-tap filters may be used for different blocks. The selected short-tap filters may depend on the block size (or width, height), block shapes, prediction direction etc.
i. In one example, 7-tap filter is used for horizonal and vertical interpolation of 4x16 or/and 16x4 bi-predicted or/and uni-predicted luma blocks.
ii. In one example, 7-tap filter is used for horizonal (or vertical) interpolation of 4x4 uni-predicted or/and bi-predicted luma blocks.
iii. In one example, 6-tap filter is used for horizonal and vertical interpolation of 4x8 or/and 8x4 bi-predicted or/and uni-predicted luma blocks.
1. Alternatively, 6-tap filter and 5-tap filter (or 5-tap filter and 6-tap filter) are used in horizonal interpolation and vertical interpolation respectively for 4x8 or/and 8x4 bi-predicted or/and uni-predicted luma blocks.

m. Different short-tap filters may be used for different kinds of motion vectors.
i. In one example, longer tap length filters may be used for motion vectors that only have fractional components in one direction (i.e., either horizonal or vertical direction), and shorter tap length filters may be used for motion vectors that have fractional components in both horizonal and vertical directions.
ii. For example, 8-tap filter is used for 4x16 or/and 16x4 or/and 4x8 or/and 8x4 or/and 4x4 bi-predicted or/and uni-predicted blocks that only have fractional MV components in one direction, and short-tap filters described in bullet 3.h is used for 4x16 or/and 16x4 or/and 4x8 or/and 8x4 or/and 4x4 bi-predicted or/and uni-predicted blocks that have fractional MV components in both directions.

iii. In one example, interpolation filters used for affine motion may be different from that used for translational motion vectors.
iv. In one example, short-tap interpolation filters may be used for affine motion compared to those used for translational motion vectors.

n. In one example, the short-tap filters may not be applied on sub-block prediction, such as affine prediction.
   i. In an alternative example, the short-tap filters may be applied on sub-block prediction, such as ATMVP prediction. In such a case, each sub-block is treated as a coding block to judge whether and how to apply short-tap filters.

o. In one example, whether to apply short-tap filters and/or how to apply short-tap filters may depend on the block dimension, coded information, etc. al.
   i. In one example, when certain mode is enabled for a block, such as OBMC, interweaved affine prediction mode, short-tap filters may be applied.

15. It is proposed that (W + N – 1 – PW) * (W + N – 1 – PH) reference pixels (instead of (W + N – 1) * (H + N – 1) reference pixels) may be fetched for motion compensation of WxH block wherein PW and PH couldn't be both equal to 0.
   a. In one example, furthermore, for the remaining reference pixels (not fetched, but required for motion compensation), padding or derivation from fetched reference samples may be applied.
   b. Alternatively, furthermore, pixels at the reference block boundaries (top, left, bottom and right boundary) are repeated to generate a (W + N – 1) * (H + N – 1) block, which is used for the final interpolation. An example is shown in Figure 21, in the figure, W = 8, H = 4, N = 7, PW = 2 and PH = 3.
   c. The fetched reference pixels may be identified by (x + MVXInt – N/2 + offSet1, y + MVYInt – N/2 + offSet2), wherein (x, y) is the top-left position of the current block, (MVXInt, MVYInt) is the integer part of the MV, offSet1 and offSet2 are integers such as -2, -1, 0, 1, 2 etc.
   d. In one example, PH is zero, and only left or/and right boundaries are repeated.
   e. In one example, PW is zero, and only top or/and bottom boundaries are repeated.
   f. In one example, both PW and PH are greater than zero, and first the left or/and the right boundaries are repeated, and then the top or/and bottom boundaries are repeated.
   g. In one example, both PW and PH are greater than zero, and first the top or/and bottom boundaries are repeated, and then the left or/and right boundaries are repeated.
   h. In one example, the left boundary is repeated by M1 times and the right boundary is repeated by PW – M1 times, wherein M1 is an integer and M1 >= 0.
      i. Alternatively, if M1 (or PW – M1) is greater than 1, instead of repeating the first left (or right) column M1 times, multiple columns may be utilized, such as the M1 left columns (or PW – M1 right columns) may be repeated.

i. In one example, the top boundary is repeated by M2 times and the bottom boundary is repeated by PH – M2 times, wherein M2 is an integer and M2 >= 0.
    i. Alternatively, if M2 (or PH – M2) is greater than 1, instead of repeating the first top (or bottom) row M2 times, multiple rows may be utilized, such as the M2 top rows (or PH – M2 bottom rows) may be repeated.
j. In one example, some default values may be used for boundary padding.
k. In one example, such boundary pixels repeating method may be used only when both horizonal and vertical components of the MV are fractional, i.e., they point to fractional pixel position instead of integer pixel position.
l. In one example, such boundary pixels repeating method may be applied to some of or all reference blocks.
    i. In one example, if MV of prediction direction X (X = 0 or 1) have fractional components in both horizonal and vertical directions, then such boundary pixels repeating method is used for prediction direction X; otherwise, it is not used.
    ii. In one example, if N (N >= 0) MV components are with fractional precision, boundary pixels repeating method may be applied to M (0 <= M <= N) of the N MV components.
        1. N and M may be different for bi-predicted blocks and uni-predicted blocks.
        2. N and M may be different for different block sizes (width or/and height or/and width * height).
        3. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 4.
        4. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 3.
        5. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 2.
        6. For example, for bi-predicted blocks, N is equal to 4 and M is equal to 1.
        7. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 3.
        8. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 2.
        9. For example, for bi-predicted blocks, N is equal to 3 and M is equal to 1.
        10. For example, for bi-predicted blocks, N is equal to 2 and M is equal to 2.
        11. For example, for bi-predicted blocks, N is equal to 2 and M is equal to 1.
        12. For example, for bi-predicted blocks, N is equal to 1 and M is equal to 1.

13. For example, for uni-predicted blocks, N is equal to 2 and M is equal to 2.
14. For example, for uni-predicted blocks, N is equal to 2 and M is equal to 1.
15. For example, for uni-predicted blocks, N is equal to 1 and M is equal to 1.

iii. Different boundary pixel repeating method may be used for the M MV components.

m. PW and/or PH may be different for different color components such as Y, Cb and Cr.

i. For example, whether to and how to apply boundary pixel repeating may depend on color formats such as 4:2:0, 4:2:2 or 4:4:4.

n. In one example, PW and/or PH may be different for different block size or shape.

iv. In one example, PW and PH are set equal to 1 for 4x16 or/and 16x4 bi-predicted or/and uni-predicted blocks.

v. In one example, PW and PH are set equal to 0 and 1 (or 1 and 0), respectively, for 4x4 bi-predicted or/and uni-predicted blocks.

vi. In one example, PW and PH are set equal to 2 for 4x8 or/and 8x4 bi-predicted or/and uni-predicted blocks.

1. Alternatively, PW and PH are set equal to 2 and 3 (or 3 and 2) respectively for 4x8 or/and 8x4 bi-predicted or/and uni-predicted blocks.

o. In one example, PW and PH may be different for uni-prediction and bi-prediction.

p. PW and PH may be different for different kinds of motion vectors.

vii. In one example, PW and PH may be smaller (or even zero) for motion vectors that only have fractional components in one direction (i.e., either horizonal or vertical direction), and they may be larger for motion vectors that have fractional components in both horizonal and vertical directions.

viii. For example, PW and PH are set equal to 0 for 4x16 or/and 16x4 or/and 4x8 or/and 8x4 or/and 4x4 bi-predicted or/and uni-predicted blocks that only have fractional MV components in one direction, and PW and PH described bullet 4.i is used for 4x16 or/and 16x4 or/and 4x8 or/and 8x4 or/and 4x4 bi-predicted or/and uni-predicted blocks that have fractional MV components in both direction.

Figure 21 shows an example of repeat boundary pixels of a reference block before interpolation.

16. The proposed methods may be applied to certain modes, block sizes/shapes, and/or certain sub-block sizes.

a. The proposed methods may be applied to certain modes, such as bi-predicted mode.

b. The proposed methods may be applied to certain block sizes.

i. In one example, it is only applied to a block with w×n<=T, where w and n are the width and height of the current block.
ii. In one example, it is only applied to a block with h <=T.

c. The proposed methods may be applied to certain color component (such as only luma component).

17. The rounding operations mentioned above may be defined as:
    a. Shift(x, s) is defined as

$$Shift(x,s) = (x + off) >> s$$

    b. SignShift(x, s) is defined as

$$SignShift(x,s) = \begin{cases} (x+off) >> s & x \geq 0 \\ -((-x+off) >> s) & x < 0 \end{cases},$$

where *off* is an integer such as 0 or 2^{s-1}.

    c. It may be defined as those used for motion vector rounding in the AMVR process, affine process or other process modules.

18. In one example, how to round the MVs may be dependent of MV components.
    a. For example, y-component of MV is rounded to integer pixel but x-component of MV is not rounded.
    b. In one example, the MV may be rounded to integer pixels before motion compensation for luma component, but rounded to 2-pel pixels before motion compensation for chroma components when the color format is 4:2:0.

19. It is proposed that bi-linear filter is used to do interpolation filtering for one or multiple specific cases, such as:
    a. 4x4 uni-prediction;
    b. 4x8 bi-prediction;
    c. 8x4-bi-prediction;
    d. 4x16 bi-prediction;
    e. 16x4 bi-prediction;
    f. 8x8 bi-prediction;
    g. 8x4 uni-prediction;
    h. 4x8 uni-prediction;

20. It is proposed that, when multi-hypothesis prediction is applied to one block, short-tap or different interpolation filters may be applied compared to those filters applied to normal prediction mode.
    a. In one example, bilinear filter may be used.
    b. short-tap or a second interpolation filter may be applied to a reference picture list which involves multiple reference blocks while for another reference picture with only one reference block, the same filter as that used for normal prediction mode may be applied.
    c. The proposed method may be applied under certain conditions, such as certain temporal layer(s), quantization parameters of a block/a tile/a slice/a picture containing the block is within a range (such as larger than a threshold).

[00160] FIG. 17 is a block diagram of a video processing apparatus 1700. The apparatus 1700 may be used to implement one or more of the methods described herein. The apparatus 1700 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1700 may include one or more processors 1702, one or more memories 1704 and video processing hardware 1706. The processor(s) 1702 may be configured to implement one or more methods described in the present document. The memory (memories) 1704 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1706 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00161] FIG. 19 is a flowchart for a method 1900 of video bitstream processing. The method 1900 includes determining (1905) a shape of a video block, determining (1910) an interpolation order based on the video block, the interpolation order being indicative of a sequence of performing horizontal interpolation and vertical interpolation, and performing the horizontal interpolation and the vertical interpolation in accordance with the interpolation order for the video block to reconstruct (1915) a decoded representation of the video block.

[00162] FIG. 20 is a flowchart for a method 2000 of video bitstream processing. The method 2000 includes determining (2005) characteristics of a motion vector related to a video block, determining (2010) an interpolation order of the video block based on the characteristics of the motion vector, the interpolation order being indicative of a sequence of performing horizontal interpolation and vertical interpolation, and performing the horizontal interpolation and the vertical interpolation in accordance with the interpolation order for the video block to reconstruct (2015) a decoded representation of the video block.

[00163] FIG. 22 is a flowchart for a method 2200 of video bitstream processing. The method 2200 includes determining (2205) dimension characteristics of a first video block, determining (2210) that a first interpolation filter is to be applied to the first video block based on the determination of the dimension characteristics, and performing (2215) further processing of the first video block using the first interpolation filter.

[00164] FIG. 23 is a flowchart for a method 2300 of video bitstream processing. The method 2300 includes determining (2305) first characteristics of a first video block, determining (2310) that a first interpolation filter is to be applied to the first video block based on the determination of the first characteristics, performing (2315) further processing of the first video block using the

first interpolation filter, determining (2320) second characteristics of a second video block, determining (2325) that a second interpolation filter is to be applied to the first video block based on the second characteristics, the first interpolation filter and the second interpolation filter being different short-tap filters, and performing (2330) further processing of the second video block using the second interpolation filter.

[00165] With reference to methods 1900, 2000, 2200, and 2300, some examples of sequences of performing horizontal interpolation and vertical interpolation and their use are described in Section 4 of the present document. For example, as described in Section 4, under different shapes of the video block, a preference may be given to performing one of the horizontal interpolation or vertical interpolation first. In some embodiments, the horizontal interpolation is performed before the vertical interpolation, and in some embodiments the vertical interpolation is performed before the horizontal interpolation.

[00166] With reference to methods 1900, 2000, 2200, and 2300, the video block may be encoded in the video bitstream in which bit efficiency may be achieved by using a bitstream generation rule related to interpolation orders that also depends on the shape of the video block.

[00167] The methods can include wherein rounding the motion vectors includes one or more of: rounding to a nearest integer-pel precision MV, or rounding to a half-pel precision MV.

[00168] The methods can include wherein rounding the MVs includes one or more of: rounding down, rounding up, rounding towards zero, or rounding away from zero.

[00169] The methods can include wherein wherein the dimension information represents that a size of the first video block is less than a threshold value, and rounding the MVs is applied to one or both of a horizontal MV component or a vertical MV component based on the dimension information representing that the size of the first video block is less than the threshold value.

[00170] The methods can include wherein the dimension information represents that a width or a height of the first video block is less than a threshold value, and rounding the MVs is applied to one or both of a horizontal MV component or a vertical MV component based on the dimension information representing that the width or the height of the first video block is less than the threshold value.

[00171] The methods can include wherein the threshold value is different for bi-predicted blocks and uni-predicted blocks.

[00172] The methods can include wherein the dimension information represents a ratio between

a width and a height of the first video block is larger than a first threshold value or smaller than a second threshold value, and wherein the rounding of the MVs is based on the determination of the dimension information.

[00173] The methods can include wherein rounding the MVs is further based on both horizontal and vertical components of the MVs being fractional.

[00174] The methods can include wherein rounding the MVs is further based on the first video block being bi-predicted or uni-predicted.

[00175] The methods can include wherein rounding the MVs is further based on a prediction direction related to the first video block.

[00176] The methods can include wherein rounding the MVs is further based on color components of the first video block.

[00177] The methods can include wherein rounding the MVs is further based on a size of the first video block, a shape of the first video block, or a prediction shape of the first video block.

[00178] The methods can include wherein rounding the MVs is applied on sub-block prediction.

[00179] The methods can include wherein a short-tap filter is applied to MV components based on the MV components having fractional precision.

[00180] The methods can include wherein short-tap filters are applied based on a dimension of the first video block, or coded information of the first video block.

[00181] The methods can include wherein short-tap filters are applied based on a mode of the first video block.

[00182] The methods can include wherein default values are used for boundary padding related to the first video block.

[00183] The methods can include wherein the merge mode is one or more of: a regular merge list, a triangular merge list, an affine merge list, or other non-intra or non-AMVP mode.

[00184] The methods can include wherein merge candidates with fractional merge candidates are excluded from a merge list.

[00185] The methods can include wherein rounding the motion information includes rounding a merge candidate associated with fractional motion vectors to integer precision, and the modified motion information is inserted into a merge list.

[00186] The methods can include wherein the motion information is a bi-prediction candidate.

[00187] The methods can include wherein MMVD is mean magnitude of vector difference.

[00188] The methods can include wherein the motion vectors are in MMVD mode.

[00189] The methods can include wherein the first video block is an MMVD coded block to be associated with integer-pel precision, and wherein base merge candidates used in MMVD are modified to integer-pel precision via rounding.

[00190] The methods can include wherein the first video block is an MMVD coded block to be associated with half-pel precision, and wherein base merge candidates used in MMVD are modified to half-pel precision via rounding.

[00191] The methods can include wherein the threshold number is a maximum number of allowed half-pel MV components or quarter-pel MV components.

[00192] The methods can include wherein the threshold number is different between bi-prediction and uni-prediction.

[00193] The methods can include wherein an indication disallowing bi-prediction is signaled in a sequence parameter set, a picture parameter set, a sequence header, a picture header, a tile header, a tile group header, a CTU row, a region, or other high-level syntax.

[00194] The methods can include wherein the methods are in conformance with a bitstream rule that allows for only integer-pel motion vectors for bi-prediction coded blocks having particular dimensions.

[00195] The methods can include wherein the first video block has a size of: 4x6, 16x4, 4x8, 8x4, or 4x4.

[00196] The methods can include wherein modifying or rounding the motion information includes modifying different MV components differently.

[00197] The methods can include wherein a y-component of a first MV is modified or rounded to integer-pixel, and an x-component of the first MV is not modified or rounded.

[00198] The methods can include wherein a luma component of a first MV is rounded to integer pixels, and a chroma component of the first MV is rounded to 2-pel pixels.

[00199] The methods can include wherein the first MV is related to a video block having a color format that is 4:2:0.

[00200] The methods can include wherein the bilateral filter is used for 4x4 uni-prediction, 4x8 bi-prediction, 8x4-bi-prediction, 4x16 bi-prediction, 16x4 bi-prediction, 8x8 bi-prediction, 8x4 uni-prediction, or 4x8 uni-prediction.

[00201] FIG. 24 is a flowchart for a method 2400 of video processing. The method 2400

includes determining (2402), for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; determining (2404) filters with interpolation filter parameters used for interpolation of the first block based on the characteristics of the first block; and performing (2406) the conversion by using the filters with the interpolation filter parameters.

[00202] In some examples, the interpolation filter parameters includes filter taps and/or interpolation filter coefficients, and the interpolation includes at least one of vertical interpolation and horizontal interpolation.

[00203] In some examples, the filters includes short-tap filters with less taps than regular interpolation filters.

[00204] In some examples, the regular interpolation filters have 8 taps.

[00205] In some examples, the characteristics of the first block includes dimension parameters including at least one of width, height, a ratio of width and height, a size of width*height of the first block.

[00206] In some examples, the filter used for the vertical interpolation is different from the filter used for the horizontal interpolation in number of taps.

[00207] In some examples, the filter used for the vertical interpolation has less taps than the filter used for the horizontal interpolation.

[00208] In some examples, the filter used for the horizontal interpolation has less taps than the filter used for the vertical interpolation.

[00209] In some examples, when the size of the first block is smaller than and/or equal to a threshold, the short-tap filters are used for the horizontal interpolation or/and the vertical interpolation.

[00210] In some examples, when the size of the first block is larger than and/or equal to a threshold, the short-tap filters are used for the horizontal interpolation or/and the vertical interpolation.

[00211] In some examples, when the width of the first block is smaller than and/or equal to a threshold, the short-tap filters are used for the horizontal interpolation, or when the height of the first block is smaller than and/or equal to a threshold, the short-tap filters are used for the vertical interpolation.

[00212] In some examples, when the ratio between the width and the height is larger than a first threshold or smaller than a second threshold, the short-tap filters are used for the vertical interpolation and/or horizontal interpolation.

[00213] In some examples, the characteristics of the first block includes at least one motion vector (MV) associated with the first block.

[00214] In some examples, only when both horizontal and vertical components of the MV are fractional, the short-tap filters are used for the interpolation.

[00215] In some examples, the characteristics of the first block includes prediction parameter indicating whether the first block is bi-predicted or uni-predicted.

[00216] In some examples, whether the short-tap filters are used or not depends on the prediction parameter.

[00217] In some examples, only when the first block is bi-predicted, the short-tap filters are used for the interpolation.

[00218] In some examples, the characteristics of the first block includes prediction direction indicating from List 0 or List 1 and/or associated motion vectors (MVs).

[00219] In some examples, whether the short-tap filters are used or not depends on prediction direction of the first block and/or the MVs.

[00220] In some examples, in a case that the first block is a bi-predicted block, whether the short-tap filters are used or not is different for different prediction direction.

[00221] In some examples, if MV of prediction direction X, X being 0 or 1, has fractional components in both horizontal and vertical directions, the short-tap filters are used for the prediction direction X; otherwise, the short-tap filters are not used.

[00222] In some examples, if N MV components are with fractional precision, the short-tap filters are used for M MV components of the N MV components, wherein N, M are integers, and 0 <= M <= N.

[00223] In some examples, N and M are different for bi-predicted blocks and uni-predicted blocks.

[00224] In some examples, for bi-predicted blocks, N is equal to 4 and M is equal to 4, or N is equal to 4 and M is equal to 3, or N is equal to 4 and M is equal to 2, or N is equal to 4 and M is equal to 1, or N is equal to 3 and M is equal to 3, or N is equal to 3 and M is equal to 2, or N is equal to 3 and M is equal to 1, or N is equal to 2 and M is equal to 2, or N is equal to 2 and M is equal to 1, or N is equal to 1 and M is equal to 1.

[00225] In some examples, for uni-predicted blocks, N is equal to 2 and M is equal to 2, or N is equal to 2 and M is equal to 1, or N is equal to 1 and M is equal to 1.

[00226] In some examples, the short-tap filters includes first short-tap filters with S1 tap and second short-tap filters with S2 tap, and wherein K MV components of the M MV components use the first short-tap filters, and (M – K) MV components of the M MV components use the second short-tap filters, wherein K is an integer in a range from 0 to M – 1, S1 and S2 are integers.

[00227] In some examples, N and M are different for different dimension parameters of blocks, wherein the dimension parameters includes width or/and height or/and width * height of the blocks.

[00228] In some examples, the characteristics of the first block includes position of the pixels of the first block.

[00229] In some examples, whether the short-tap filters are used or not depends on the position of

the pixels.

[00230] In some examples, the short-tap filters are used only for boundary pixels of the first block.

[00231] In some examples, the short-tap filters are used only for N1 right column or/and N2 left column or/and N3 top row or/and N4 bottom row of the first block, N1, N2, N3, N4 being integers.

[00232] In some examples, the characteristics of the first block includes color components of the first block.

[00233] In some examples, whether the short-tap filters are used or not is different for different color components of the first block.

[00234] In some examples, the color components include Y, Cb and Cr.

[00235] In some examples, the characteristics of the first block includes color formats of the first block.

[00236] In some examples, whether to and how to apply the short-tap filters depend on color formats of the first block.

[00237] In some examples, the color formats include 4:2:0, 4:2:2 or 4:4:4.

[00238] In some examples, the filters includes different short-tap filters with different taps, and selection of the different short-tap filters is based on the characteristics of the blocks.

[00239] In some examples, a 7-tap filter is selected for horizontal and vertical interpolation of 4x16 or/and 16x4 bi-predicted or/and uni-predicted luma blocks.

[00240] In some examples, a 7-tap filter is selected for horizontal or vertical interpolation of 4x4 uni-predicted or/and bi-predicted luma blocks.

[00241] In some examples, a 6-tap filter is selected for horizontal and vertical interpolation of 4x8 or/and 8x4 bi-predicted or/and uni-predicted luma blocks.

[00242] In some examples, a 6-tap filter and a 5-tap filter or a 5-tap filter and a 6-tap filter are

selected for horizontal interpolation and vertical interpolation respectively for 4x8 or/and 8x4 bi-predicted or/and uni-predicted luma blocks.

[00243] In some examples, the filters includes different short-tap filters with different taps, and the different short-tap filters are used for different kinds of motion vectors (MVs).

[00244] In some examples, longer tap length filters from the different short-tap filters are used for MVs that only have fractional components in one of horizontal or vertical direction, and shorter tap length filters from the different short-tap filters are used for MVs that have fractional components in both horizontal and vertical directions.

[00245] In some examples, a 8-tap filter is used for 4x16 or/and 16x4 or/and 4x8 or/and 8x4 or/and 4x4 bi-predicted or/and uni-predicted blocks that only have fractional MV components in one of horizontal or vertical direction, and short-tap filters is used for 4x16 or/and 16x4 or/and 4x8 or/and 8x4 or/and 4x4 bi-predicted or/and uni-predicted blocks that have fractional MV components in both directions.

[00246] In some examples, filters used for affine motion are different from that used for translational motion vectors.

[00247] In some examples, filters used for affine motion have less taps compared to those used for translational motion vectors.

[00248] In some examples, the short-tap filters are not applied to sub-block based prediction including affine prediction.

[00249] In some examples, the short-tap filters are applied to sub-block based prediction including Advanced Temporal Motion Vector Prediction (ATMVP) prediction.

[00250] In some examples, each sub-block is used as a coding block to determine whether to and how to apply the short-tap filters.

[00251] In some examples, the characteristics of the first block includes dimension parameters and coded information of the first block, and whether to and how to apply the short-tap filters depend on the block dimension and coded information of the first block.

[00252] In some examples, when certain mode including at least one of OBMC and interweaved affine prediction mode is enabled for the first block, the short-tap filters are applied.

[00253] In some examples, the conversion generates the first/second block of video from the bitstream representation.

[00254] In some examples, the conversion generates the bitstream representation from the first/second block of video.

[00255] FIG. 25 is a flowchart for a method 2500 of video processing. The method 2500 includes fetching (2502), for a conversion between a first block of video and a bitstream representation of the first block, reference pixels of a first reference block from reference picture, wherein the first reference block is smaller than a second reference block required for motion compensation of the first block; padding (2504) the first reference block with padding pixels to generate the second reference block required for motion compensation of the first block; and performing (2506) the conversion by using the generated second reference block.

[00256] In some examples, the first block has a size of W*H, the first reference block has a size of (W + N – 1 – PW) * (H + N – 1 – PH), and the second reference block has a size of (W + N – 1) * (H + N – 1), wherein W is width of the first block, H is height of the first block, N is the number of interpolation filter taps used for the first block, PW and PH are integers.

[00257] In some examples, the step of padding the first reference block with padding pixels to generate the second reference block includes: repeating pixels at one or more boundaries of the first reference block as the padding pixels to generate the second reference block.

[00258] In some examples, the boundaries are top, left, bottom and right boundary of the first reference block.

[00259] In some examples, W = 8, H = 4, N = 7, PW = 2 and PH = 3.

[00260] In some examples, the pixels at the top, left and right boundary are repeated once, and the pixels at the bottom boundary are repeated twice.

[00261] In some examples, the fetched reference pixels are identified by (x + MVXInt – N/2 + offSet1, y + MVYInt – N/2 + offSet2), wherein (x, y) is the top-left position of the first block, (MVXInt, MVYInt) is the integer part of motion vector (MV) for the first block, and offSet1 and offSet2 are integers.

[00262] In some examples, when PH is zero, only the pixels at the left or/and right boundaries of the first reference block are repeated.

[00263] In some examples, when PW is zero, only the pixels at the top or/and bottom boundaries of the first reference block are repeated.

[00264] In some examples, when both PW and PH are greater than zero, first the pixels at the left or/and the right boundaries of the first reference block are repeated, and then the pixels at the top or/and bottom boundaries of the first reference block are repeated, or first the top or/and bottom boundaries of the first reference block are repeated, and then the left or/and right boundaries of the first reference block are repeated.

[00265] In some examples, the pixels at the left boundary of the first reference block is repeated by M1 times and the pixels at the right boundary of the first reference block is repeated by (PW – M1) times, wherein M1 is an integer and M1 >= 0.

[00266] In some examples, the pixels of M1 left columns of the first reference block, or the pixels of (PW – M1) right columns of the first reference block are repeated, wherein M1 >1 or PW – M1>1.

[00267] In some examples, the pixels at the top boundary of the first reference block is repeated by M2 times and the pixels at the bottom boundary of the first reference block is repeated by (PH – M2) times, wherein M2 is an integer and M2 >= 0.

[00268] In some examples, the pixels of M2 top rows of the first reference block, or the pixels of (PH – M2) bottom rows of the first reference block are repeated, wherein M2 >1 or PW – M2>1.

[00269] In some examples, when both horizontal and vertical components of MV for the first block are fractional, pixels at one or more boundaries of the first reference block are repeated as the padding pixels to generate the second reference block.

[00270] In some examples, when MV of prediction direction X, X being 0 or 1, has fractional components in both horizontal and vertical directions, pixels at one or more boundaries of the first reference block are repeated as the padding pixels to generate the second reference block.

[00271] In some examples, the first reference block is any one of partial or all reference blocks of the first block.

[00272] In some examples, if MV of prediction direction X, X being 0 or 1, has fractional components in both horizontal and vertical directions, pixels at one or more boundaries of the first reference block are repeated as the padding pixels to generate the second reference block for prediction direction X; otherwise, the pixels are not repeated.

[00273] In some examples, if N2 MV components are with fractional precision, pixels at one or more boundaries of the first reference block are repeated as the padding pixels to generate the second reference block for M MV components of the N2 MV components, wherein N2, M are integers, and 0 <= M <= N2.

[00274] In some examples, N2 and M are different for bi-predicted blocks and uni-predicted blocks.

[00275] In some examples, N2 and M are different for different block sizes, the block size being associated with width or/and height or/and width * height of the block.

[00276] In some examples, for bi-predicted blocks, N2 is equal to 4 and M is equal to 4, or N2 is equal to 4 and M is equal to 3, or N2 is equal to 4 and M is equal to 2, or N2 is equal to 4 and M is equal to 1, or N2 is equal to 3 and M is equal to 3, or N2 is equal to 3 and M is equal to 2, or

[00277] N2 is equal to 3 and M is equal to 1, or N2 is equal to 2 and M is equal to 2, or N2 is equal to 2 and M is equal to 1, or N2 is equal to 1 and M is equal to 1.

[00278] In some examples, for uni-predicted blocks, N2 is equal to 2 and M is equal to 2, or N2 is equal to 2 and M is equal to 1, or N2 is equal to 1 and M is equal to 1.

[00279] In some examples, pixels at different boundaries of the first reference block are repeated as the padding pixels in different ways to generate the second reference block for the M MV components.

[00280] In some examples, when pixel padding is not used for a horizontal MV component, PW is set equal to zero when fetching the first reference block using the MV.

[00281] In some examples, when pixel padding is not used for a vertical MV component, PH is set equal to zero when fetching the first reference block using the MV.

[00282] In some examples, PW and/or PH are different for different color components of the first block.

[00283] In some examples, the color components includes Y, Cb and Cr.

[00284] In some examples, PW and/or PH are different for different block size or shape.

[00285] In some examples, PW and PH are set equal to 1 for 4x16 or/and 16x4 bi-predicted or/and uni-predicted blocks.

[00286] In some examples, PW and PH are set equal to 0 and 1, or 1 and 0 respectively, for 4x4 bi-predicted or/and uni-predicted blocks.

[00287] In some examples, PW and PH are set equal to 2 for 4x8 or/and 8x4 bi-predicted or/and uni-predicted block.

[00288] In some examples, PW and PH are set equal to 2 and 3, or 3 and 2 respectively, for 4x8 or/and 8x4 bi-predicted or/and uni-predicted blocks.

[00289] In some examples, PW and PH are different for uni-prediction and bi-prediction.

[00290] In some examples, PW and PH are different for different kinds of motion vectors.

[00291] In some examples, PW and PH are set to a smaller value or equal to zero for motion vectors (MVs) that only have fractional components in one of horizontal or vertical direction, and PW and PH are set to a larger value for MVs that have fractional components in both horizontal and vertical directions.

[00292] In some examples, PW and PH are set equal to 0 for 4x16 or/and 16x4 or/and 4x8 or/and 8x4 or/and 4x4 bi-predicted or/and uni-predicted blocks that only have fractional MV components in one of horizontal or vertical direction.

[00293] In some examples, the PW and PH are used for 4x16 or/and 16x4 or/and 4x8 or/and 8x4 or/and 4x4 bi-predicted or/and uni-predicted blocks that have fractional MV components in both horizontal and vertical direction

[00294] In some examples, whether to and how to repeat pixels at the boundaries depend on color formats of the first block.

[00295] In some examples, the color formats includes 4:2:0, 4:2:2 or 4:4:4.

[00296] In some examples, the step of padding the first reference block with padding pixels to generate the second reference block includes: padding default values as the padding pixels to generate the second reference block.

[00297] In some examples, the conversion generates the first block of video from the bitstream representation.

[00298] In some examples, the conversion generates the bitstream representation from the first/second block of video.

[00299] FIG. 26 is a flowchart for a method 2600 of video processing. The method 2600 includes determining (2602), for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; performing (2604) rounding process on motion vector (MV) of the first block based on the characteristics of the first block; and performing (2606) the conversion by using the rounded MV.

[00300] In some examples, the performing rounding process on the MV includes rounding the MV to integer-pel precision or half-pel precision.

[00301] In some examples, the MV is rounded to a nearest integer-pel precision MV or half-pel precision MV.

[00302] In some examples, the performing rounding process on the MV includes rounding up, rounding down, rounding towards zero or rounding away from zero of the MV.

[00303] In some examples, the characteristics of the first block includes dimension parameters including at least one of width, height, a ratio of width and height, a size of width*height of the first block.

[00304] In some examples, when the size of the first block is smaller than and/or equal to a threshold L, rounding process is performed on horizontal or/and vertical component of the MV.

[00305] In some examples, when the size of the first block is larger than and/or equal to a threshold L, rounding process is performed on horizontal or/and vertical component of the MV.

[00306] In some examples, when the width of the first block is smaller than and/or equal to a second threshold L1, rounding process is performed on horizontal component of the MV, or when the height of the first block is smaller than and/or equal to the second threshold L1, rounding process is performed on vertical component of the MV.

[00307] In some examples, the thresholds L and L1 are different for bi-predicted blocks and uni-predicted blocks.

[00308] In some examples, when the ratio between width and height is larger than a third threshold L3 or smaller than a fourth threshold L4, rounding process is performed on the MV.

[00309] In some examples, when both horizontal and vertical components of the MV are fractional, rounding process is performed on the MV.

[00310] In some examples, the characteristics of the first block includes prediction parameter indicating whether the first block is bi-predicted or uni-predicted.

[00311] In some examples, whether performing the rounding process on the MV depends on the prediction parameter.

[00312] In some examples, only when the first block is bi-predicted, rounding process is performed on the MV.

[00313] In some examples, the characteristics of the first block includes prediction direction indicating from List 0 or List 1 and/or associated MVs.

[00314] In some examples, whether performing the rounding process on the MV depends on prediction direction of the first block and/or the MVs.

[00315] In some examples, in a case that the first block is a bi-predicted block, whether performing the rounding process on the MV or not is different for different prediction direction.

[00316] In some examples, if MV of prediction direction X, X being 0 or 1, has fractional components in both horizontal and vertical directions, the rounding process is performed on N MV components for the prediction direction X, N is an integer in a range from 0 to 2; otherwise, the rounding process is not performed.

[00317] In some examples, if N1 MV components are with fractional precision, the rounding process is performed on M MV components of the N1 MV components, wherein N1, M are integers, and 0 <= M <= N1.

[00318] In some examples, N1 and M are different for bi-predicted blocks and uni-predicted blocks.

[00319] In some examples, for bi-predicted blocks,

N1 is equal to 4 and M is equal to 4, or

N1 is equal to 4 and M is equal to 3, or

N1 is equal to 4 and M is equal to 2, or

N1 is equal to 4 and M is equal to 1, or

N1 is equal to 3 and M is equal to 3, or

N1 is equal to 3 and M is equal to 2, or

N1 is equal to 3 and M is equal to 1, or

N1 is equal to 2 and M is equal to 2, or

N1 is equal to 2 and M is equal to 1, or

N1 is equal to 1 and M is equal to 1.

[00320] In some examples, for uni-predicted blocks,

N1 is equal to 2 and M is equal to 2, or

N1 is equal to 2 and M is equal to 1, or

N1 is equal to 1 and M is equal to 1.

[00321] In some examples, N1 and M are different for different dimension parameters including at least one of width, height, a ratio of width and height, a size of width*height of the first block.

[00322] In some examples, K MV components of the M MV components are rounded to integer-pel precision and M – K MV components are rounded to half-pel precision, wherein K is an integer in a range from 0 to M – 1.

[00323] In some examples, the characteristics of the first block includes color components of the first block.

[00324] In some examples, whether performing rounding process on the MV is different for different color components of the first block.

[00325] In some examples, the color components include Y, Cb and Cr.

[00326] In some examples, the characteristics of the first block includes color formats of the first block.

[00327] In some examples, whether performing rounding process on the MV depends on color formats of the first block.

[00328] In some examples, the color formats include 4:2:0, 4:2:2 or 4:4:4.

[00329] In some examples, whether and/or how to perform rounding process on the MV depend on the characteristics of the block.

[00330] In some examples, one or more MV components of 4x16 or/and 16x4 bi-predicted or/and uni-predicted luma blocks are rounded to half-pel precision.

[00331] In some examples, one or more MV components of 4x16 or/and 16x4 bi-predicted or/and uni-predicted luma blocks are rounded to integer-pel precision.

[00332] In some examples, one or more MV components of 4x4 uni-predicted or/and bi-predicted luma blocks are rounded to integer-pel precision.

[00333] In some examples, one or more MV components of 4x8 or/and 8x4 bi-predicted or/and uni-predicted luma blocks are rounded to integer-pel precision.

[00334] In some examples, the characteristics of the first block includes whether the first block is coded with sub-block based prediction method including affine prediction mode and Sub-block based Temporal Motion Vector Prediction (SbTMVP) mode.

[00335] In some examples, the rounding process on the MV is not applied if the first block is coded with affine prediction mode.

[00336] In some examples, the rounding process on the MV is applied if the first block is coded with SbTMVP mode, and the rounding process is performed for each sub-block of the first block.

[00337] In some examples, the performing rounding process on motion vector (MV) of the first block based on the characteristics of the first block comprises: determining whether at least one MV of the first block are fractional precisions when the dimension parameters of the first block satisfy a predetermined rule; and in response to the determination that the at least one MV of the first block are fractional precisions, performing rounding process on the at least one MV to generate rounded MVs having integer precision.

[00338] In some examples, the bitstream representation of the first block follows the rule depending on the dimension parameters of the first block, wherein only integer-pel MVs are allowed for bi-prediction coded blocks.

[00339] In some examples, the dimensions parameters of the first block are 4x16, 16x4, 4x8, 8x4, or 4x4.

[00340] In some examples, the performing the conversion by using the rounded MV comprises: performing motion compensation for the first block by using the rounded MVs.

[00341] FIG. 27 is a flowchart for a method 2700 of video processing. The method 2700 includes determining (2702), for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; performing (2704) motion compensation for the first block using a MV with a first precision; and storing (2706) a MV with a second precision for the first block; wherein the first precision is different from the second precision.

[00342] In some examples, the characteristics of the first block includes dimension parameters including at least one of width, height, a ratio of width and height, a size of width*height of the first block.

[00343] In some examples, the first precision is integer precision and the second precision is fractional precision.

[00344] FIG. 28 is a flowchart for a method 2800 of video processing. The method 2800 includes determining (2802), for a conversion between a first block of video and a bitstream

representation of the first block, coding mode of the first block; performing (2804) rounding process on motion vector (MV) of the first block if the coding mode of the first block satisfying a predetermined rule; and performing (2806) the motion compensation of the first block by using the rounded MV.

[00345] In some examples, the predetermined rule comprises: the first block is coded with merge mode, non-intra modes or non-Advanced motion vector prediction (AMVP) mode.

[00346] FIG. 29 is a flowchart for a method 2900 of video processing. The method 2900 includes generating (2902), for a conversion between a first block of video and a bitstream representation of the first block, a first motion vector (MV) candidate list for the first block; performing (2904) rounding process on MV of at least one candidate before adding the at least one candidate into the first MV candidate list; and performing (2906) the conversion by using the first MV candidate list.

[00347] In some examples, the first block is coded with merge mode, non-intra modes or non-Advanced motion vector prediction (AMVP) mode, and the MV candidate list includes merge candidate list and non-merge candidate list.

[00348] In some examples, the candidates with fractional MVs are excluded from the first MV candidate list.

[00349] In some examples, the at least one candidate comprises: a candidate derived from a spatial block, a candidate derived from a temporal block, a candidate derived from a History motion vector prediction (HMVP) table or a pairwise bi-prediction merge candidate.

[00350] In some examples, the method further comprises: providing a separate HMVP table to store the candidates with MV of integer precision.

[00351] In some examples, the method further comprises: performing the rounding process on the MV or the rounding process on the MV of candidate in the candidate list based on characteristics of the first block.

[00352] In some examples, the characteristics of the first block includes dimension parameters including at least one of width, height, a ratio of width and height, a size of width*height of the first block.

[00353] In some examples, the dimension parameters include at least one of 4x16, 16x4, 4x8, 8x4, 4x4.

[00354] In some examples, the characteristics of the first block includes prediction parameter indicating whether the first block is bi-predicted or uni-predicted, and performing rounding process on MV comprises: performing the rounding process on the MV or the rounding process on the MV of candidate in the candidate list only when the candidate is a bi-prediction candidate.

[00355] In some examples, the first block is coded with AMVP mode, and the candidate is AMVP candidate.

[00356] In some examples, the first block is non-affine mode.

[00357] FIG. 30 is a flowchart for a method 3000 of video processing. The method 3000 includes determining (3002), for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; determining (3004) constraint parameter to be applied on the first block based on the characteristics of the first block, wherein the constraint parameter constrains a maximum number of fractional motion vector (MV) components of the first block; and performing (3006) the conversion by using the constraint parameter.

[00358] In some examples, the MV components include at least one of horizontal MV component and/or vertical MV component, and the fractional MV components include at least one of half-pel MV components, quarter-pel MV components, MV components with finer precision than quarter-pel.

[00359] In some examples, the characteristics of the first block includes prediction parameter indicating whether the first block is bi-predicted or uni-predicted.

[00360] In some examples, the constraint parameter is different for bi-prediction and uni-prediction.

[00361] In some examples, the constraint parameter is not applied in uni-prediction.

[00362] In some examples, the constraint parameter is applied when the first block is bi-predicted 4x8, 8x4, 4x16, or16x4 block.

[00363] In some examples, the constraint parameter is not applied when the first block is uni-predicted 4x8, 8x4, 4x16 or16x4 block.

[00364] In some examples, the constraint parameter is applied when the first block is a uni-predicted 4x4 or a bi-predicted 4x4 block.

[00365] In some examples, for bi-predicted blocks, the maximum number of the fractional MV components is 3, 2, 1 or 0.

[00366] In some examples, for uni-predicted blocks, the maximum number of the fractional MV components is 1 or 0.

[00367] In some examples, for bi-predicted blocks, the maximum number of the quarter-pel MV components is 3, 2, 1 or 0.

[00368] In some examples, for uni-predicted blocks, the maximum number of the quarter-pel MV components is 1 or 0.

[00369] In some examples, the characteristics of the first block includes at least one of shapes and dimension parameters including at least one of width, height, a ratio of width and height, a size of width*height, and shape of the first block.

[00370] In some examples, the constraint parameter is different for different sizes or shapes of the first block.

[00371] In some examples, the characteristics of the first block includes mode parameter indicating coding mode of the first block.

[00372] In some examples, the coding mode includes a triangle mode in which the current is split into two partitions, wherein each partition has at least one MV.

[00373] In some examples, the constraint parameter is applied when the first block is 4x16 or 16x4 block coded in the triangle mode.

[00374] FIG. 31 is a flowchart for a method 3100 of video processing. The method 3100 includes acquiring (3102), an a signaled indication of disallowing at least one of bi-prediction and uni-prediction when the characteristics of a block satisfying a predetermined rule; determining (3104), for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; and performing (3106) the conversion by using the indication when the characteristics of the first block satisfies the predetermined rule.

[00375] FIG. 32 is a flowchart for a method 3200 of video processing. The method 3200 includes signaling (3202), an indication of disallowing at least one of bi-prediction and uni-prediction when the characteristics of a block satisfying a predetermined rule; determining (3204), for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block; performing (3206) the conversion based on the characteristics of the first block, wherein during the conversation, at least one of bi-prediction

and uni-prediction is disabled when the characteristics of the first block satisfies a predetermined rule.

[00376] In some examples, the indication is signaled in sequence parameter set/picture parameter set/sequence header/picture header/tile header/tile group header/coding tree unit (CTU) rows/regions/other high-level syntax.

[00377] In some examples, the characteristics of the first block includes dimension parameters including at least one of width, height, a ratio of width and height, a size of width*height, and shape of the first block.

[00378] In some examples, the predetermined rules comprises: the first block is of certain block dimensions.

[00379] In some examples, the characteristics of the first block includes mode parameter indicating coding mode of the first block.

[00380] In some examples, the predetermined rule comprises: the first block is coded with non-affine mode.

[00381] In some examples, when at least one of uni-prediction and bi-prediction is disallowed for the first block, the signaling of Advanced Motion Vector Resolution (AMVR) parameter for the first block is modified accordingly.

[00382] In some examples, the signaling of Advanced Motion Vector Resolution (AMVR) parameter is modified so that only integer-pel precisions are allowed for the first block.

[00383] In some examples, the signaling of Advanced Motion Vector Resolution (AMVR) parameter is modified so that different motion vector (MV) precisions are utilized.

[00384] In some examples, the block dimension of the first block is at least one of 4x16, 16x4, 4x8, 8x4, 4x4.

[00385] In some examples, the bitstream representation of the first block follows the rule depending on the dimensions parameters of the first block, wherein only integer-pel MVs are allowed for bi-prediction coded blocks.

[00386] FIG. 33 is a flowchart for a method 3300 of video processing. The method 3300 includes determining (3302), for a conversion between a first block of video and a bitstream representation of the first block, whether fractional motion vector (MV) or motion vector difference (MVD) precision is allowed for a first block; signaling (3304), an Advanced Motion

Vector Resolution (AMVR) parameter for the first block based on the determination; and performing (3306) the conversion by using the AMVR parameter.

[00387] FIG. 34 is a flowchart for a method 3400 of video processing. The method 3400 includes determining (3402), for a conversion between a first block of video and a bitstream representation of the first block, whether fractional motion vector (MV) or motion vector difference (MVD) precision is allowed for a first block; acquiring (3404), an Advanced Motion Vector Resolution (AMVR) parameter for the first block based on the determination; and performing (3406) the conversion by using the AMVR parameter.

[00388] In some examples, if the fractional MV or MVD precision is disallowed for the first block, the AMVR parameter indicating whether MV/MVD precision of the current block is fractional precision is skipped and derived to be false implicitly.

[00389]

[00390] 5. An Embodiment

[00391] In following embodiments, PW and PH are designed for 4x16, 16x4, 4x4, 8x4 and 4x8 blocks.

[00392] Suppose the MV of the block in reference list X is MVX, and the horizonal and vertical components of MVX are MVX[0] and MVX[1] respectively, and the integer part of MVX[0] and MVX[1] are MVXInt[0] and MVXInt[1] respectively, wherein X = 0 or 1. Suppose the interpolation filter tap (in motion compensation) is N (for example, 8, 6, 4, or 2), and the current block size is WxH, and the position (i.e., position of the top-left pixel) of current block is (x, y). The index of the rows and columns start from 1, for example, H rows include the 1st, …, (H – 1)th row.

[00393] The following boundary pixel repeating process is performed only when both MVX[0] and MVX[1] are fractional.

[00394] 5.1 An Embodiment

[00395] For 4x16 and 16x4 uni-predicted and bi-predicted blocks, PW and PH are both set equal to 1 for prediction direction X. First, (W + N – 2) * (H + N – 2) reference pixels are fetched from the reference picture, wherein the top-left position of the reference pixels is identified by (MVXInt[0] + x – N/2 + 1, MVXInt[1] + y – N/2 + 1). Then, the (W + N – 1)th column is generated by copying the (W + N – 2)th column. Finally, the (H + N – 1)th row is generated by copying the (H + N – 2)th row.

[00396] For 4x4 uni-predicted block, PW and PH are set equal to 0 and 1 respectively. First, (W + N – 1) * (H + N – 2) reference pixels are fetched from the reference picture, wherein the top-left position of the reference pixels is identified by (MVXInt[0] + x – N/2 + 1, MVXInt[1] + y – N/2 + 1). Then, the (H + N – 1)th row is generated by copying the (H + N – 2)th row.

[00397] For 4x8 and 8x4 ui-predicted and bi-predicted blocks, PW and PH are set equal to 2 and 3 respectively. First, (W + N – 3) * (H + N – 4) reference pixels are fetched from the reference picture, wherein the top-left position of the reference pixels is identified by (MVXInt[0] + x – N/2 + 2, MVXInt[1] + y – N/2 + 2). Then, the 1st column is copied to its left side to obtain W + N – 2 columns, after that, the (W + N – 1)th column is generated by copying the (W + N – 2)th column. Finally, the 1st row is copied to its upside to obtain H + N – 3 rows, after that, the (H + N – 2)th row and (H + N – 1)th row are generated by copying the (H + N – 3)th row.

[00398] 5.2 An Embodiment

[00399] For 4x16 and 16x4 ui-predicted and bi-predicted blocks, PW and PH are both set equal to 1 for prediction direction X. First, (W + N – 2) * (H + N – 2) reference pixels are fetched from the reference picture, wherein the top-left position of the reference pixels is identified by (MVXInt[0] + x – N/2 + 2, MVXInt[1] + y – N/2 + 2). Then, the 1st column is copied to its left side to obtain W + N – 1 columns. Finally, the 1st row is copied to its upside to obtain H + N – 1 rows.

[00400] For 4x4 uni-predicted block, PW and PH are set equal to 0 and 1 respectively. First, (W + N – 1) * (H + N – 2) reference pixels are fetched from the reference picture, wherein the top-left position of the reference pixels is identified by (MVXInt[0] + x – N/2 + 1, MVXInt[1] + y – N/2 + 2). Then, the 1st row is copied to its upside to obtain H + N – 1 rows.

[00401] For 4x8 and 8x4 ui-predicted and bi-predicted blocks, PW and PH are set equal to 2 and 3 respectively. First, (W + N – 3) * (H + N – 4) reference pixels are fetched from the reference picture, wherein the top-left position of the reference pixels is identified by (MVXInt[0] + x – N/2 + 2, MVXInt[1] + y – N/2 + 2). Then, the 1st column is copied to its left side to obtain W + N – 2 columns, after that, the (W + N – 1)th column is generated by copying the (W + N – 2)th column. Finally, the 1st row is copied to its upside to obtain H + N – 3 rows, after that, the (H + N – 2)th row and (H + N – 1)th row are generated by copying the (H + N – 3)th row.

[00402] It will be appreciated that the disclosed techniques may be embodied in video encoders or decoders to improve compression efficiency when the coding units being compressed have shaped that are significantly different than the traditional square shaped blocks

or rectangular blocks that are half-square shaped. For example, new coding tools that use long or tall coding units such as 4x32 or 32x4 sized units may benefit from the disclosed techniques.

[00403] 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.

[00404] 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.

[00405] 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).

[00406] 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.

[00407] 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.

[00408] 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.

[00409] 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:

acquiring, a signaled indication of disallowing at least one of bi-prediction and uni-prediction when characteristics of a first block satisfying a predetermined rule;

determining, for a conversion between the first block of video and a bitstream representation of the first block, the characteristics of the first block; and

performing the conversion by using the indication when the characteristics of the first block satisfies the predetermined rule.

2. A method of video processing, comprising:

signaling, an indication of disallowing at least one of bi-prediction and uni-prediction when characteristics of a first block satisfying a predetermined rule;

determining, for a conversion between the first block of video and a bitstream representation of the first block, the characteristics of the first block;

performing the conversion based on the characteristics of the first block, wherein during the conversation, at least one of bi-prediction and uni-prediction is disabled when the characteristics of the first block satisfies a predetermined rule.

3. The method of claim 1 or 2, wherein the indication is signaled in sequence parameter set/picture parameter set/sequence header/picture header/tile header/tile group header/coding tree unit (CTU) rows/regions/other high-level syntax.

4. The method of anyone of claims 1 to 3, wherein the characteristics of the first block includes dimension parameters including at least one of width, height, a ratio of width and height, a size of width*height, and shape of the first block.

5. The method of claim 4, wherein the predetermined rules comprises: the first block is of certain block dimensions.

6. The method of any one of claims 1-5, wherein the characteristics of the first block includes mode parameter indicating coding mode of the first block.

7. The method of claim 6, wherein the predetermined rule comprises: the first block is coded with non-affine mode.

8. The method of any one of claim 1-7, wherein when at least one of uni-prediction and bi-prediction is disallowed for the first block, the signaling of Advanced Motion Vector Resolution (AMVR) parameter for the first block is modified accordingly.

9. The method of claim 8, wherein the signaling of Advanced Motion Vector Resolution (AMVR) parameter is modified so that only integer-pel precisions are allowed for the first block.

10. The method of claim 8, wherein the signaling of Advanced Motion Vector Resolution (AMVR) parameter is modified so that different motion vector (MV) precisions are utilized.

11. The method of anyone of claims 1-10, wherein the block dimension of the first block is at least one of 4x16, 16x4, 4x8, 8x4, 4x4.

12. The method of anyone of claims 1-11, wherein the bitstream representation of the first block follows the rule depending on the dimensions parameters of the first block, wherein only integer-pel MVs are allowed for bi-prediction coded blocks.

13. A method of video processing, comprising:

determining, for a conversion between a first block of video and a bitstream representation of the first block, whether fractional motion vector (MV) or motion vector difference (MVD) precision is allowed for a first block;

signaling, an Advanced Motion Vector Resolution (AMVR) parameter for the first block based on the determination; and

performing the conversion by using the AMVR parameter.

14. A method of video processing, comprising:

determining, for a conversion between a first block of video and a bitstream representation of the first block, whether fractional motion vector (MV) or motion vector difference (MVD) precision is allowed for a first block;

acquiring, an Advanced Motion Vector Resolution (AMVR) parameter for the first block based on the determination; and

performing the conversion by using the AMVR parameter.

15. The method of claim 13 or 14, wherein if the fractional MV or MVD precision is disallowed for the first block, the AMVR parameter indicating whether MV/MVD precision of the current block is fractional precision is skipped and derived to be false implicitly.

16. The method of anyone of claims 1 to 15, wherein the conversion generates the first block of video from the bitstream representation.

17. The method of anyone of claims 1 to 15, wherein the conversion generates the bitstream representation from the first block of video.

18. 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 17.

19. 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 17.

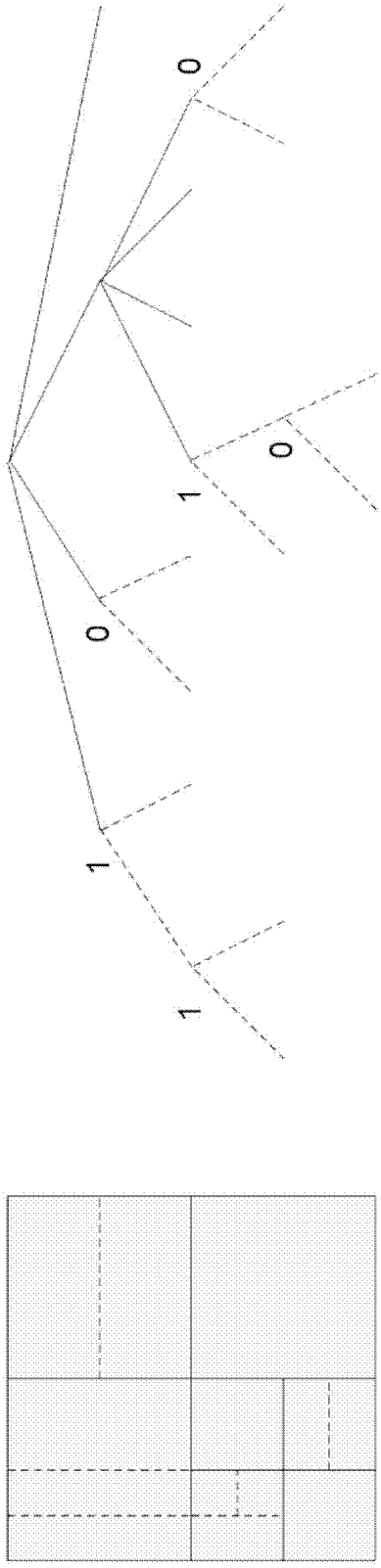


FIG. 1

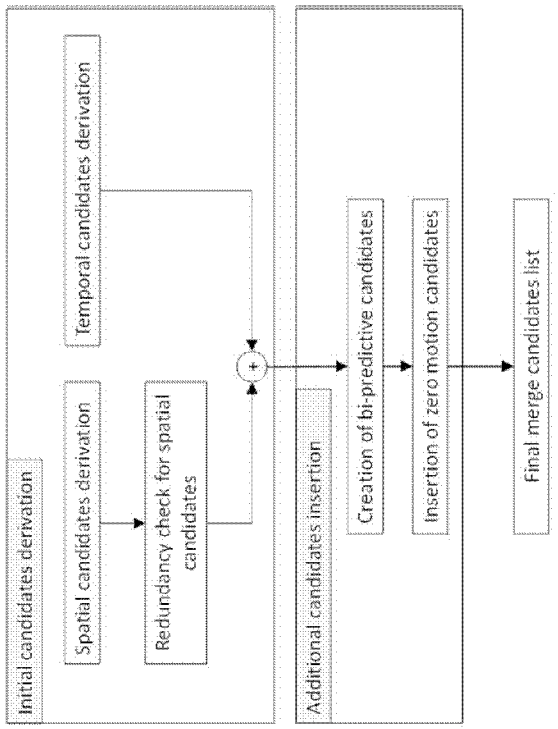

Initial candidates derivation
Spatial candidates derivation
Redundancy check for spatial candidates
Temporal candidates derivation
+
Additional candidates insertion
Creation of bi-predictive candidates
Insertion of zero motion candidates
Final merge candidates list

FIG. 2

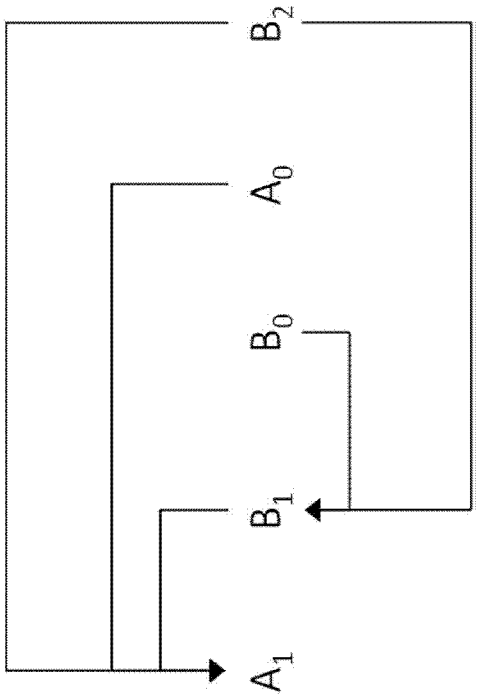


FIG. 4

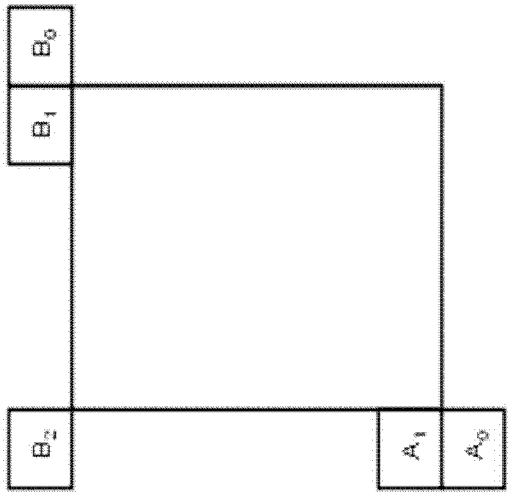
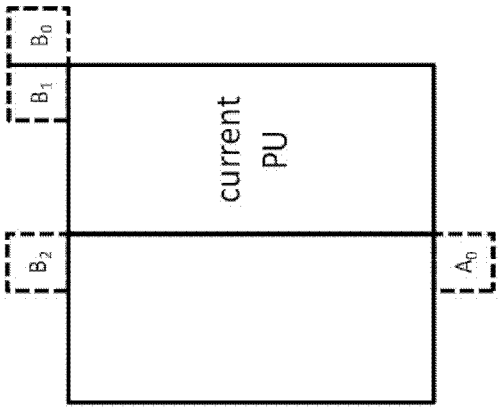
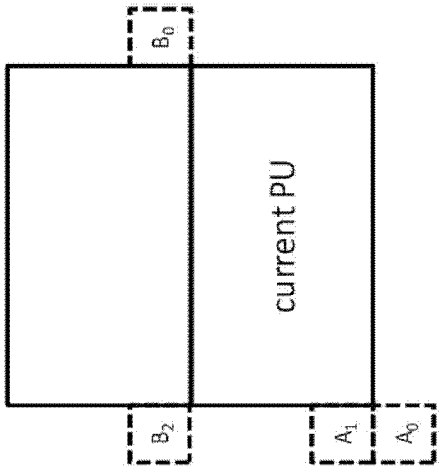


FIG. 3



(a) second PU of Nx2N



(b)second PU of 2NxN

FIG. 5

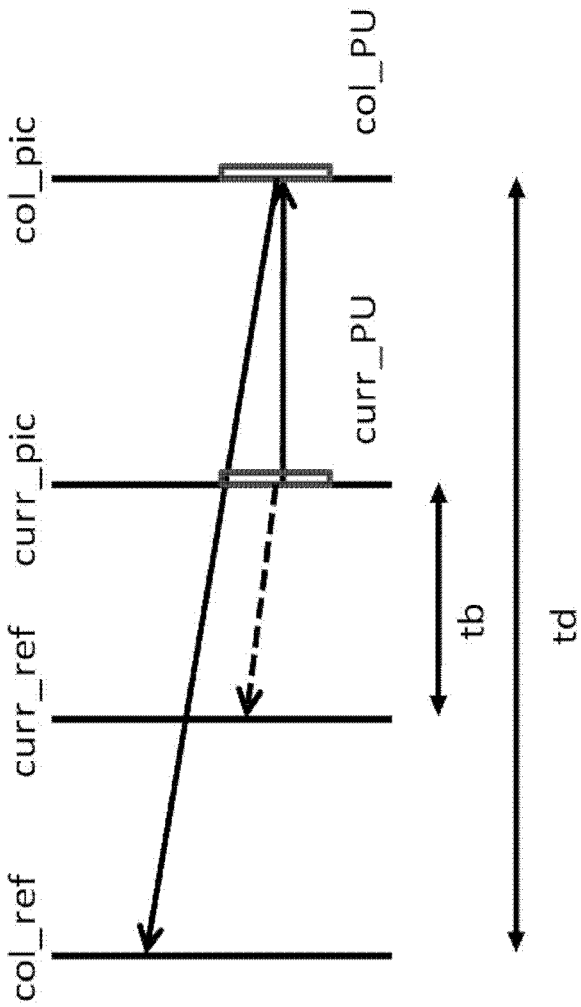


FIG. 6

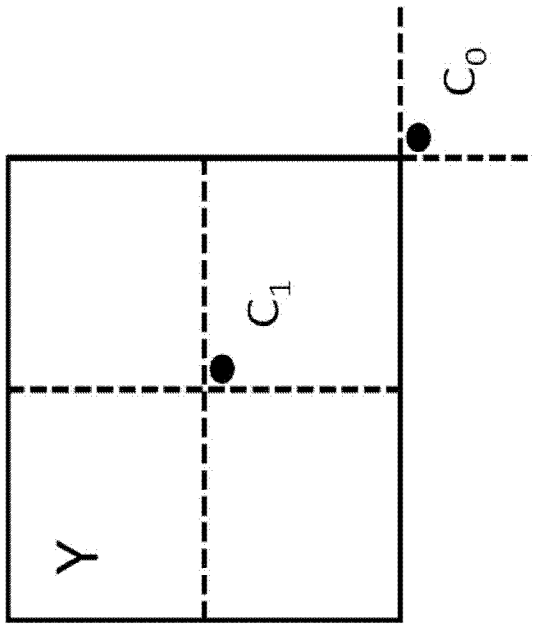


FIG. 7

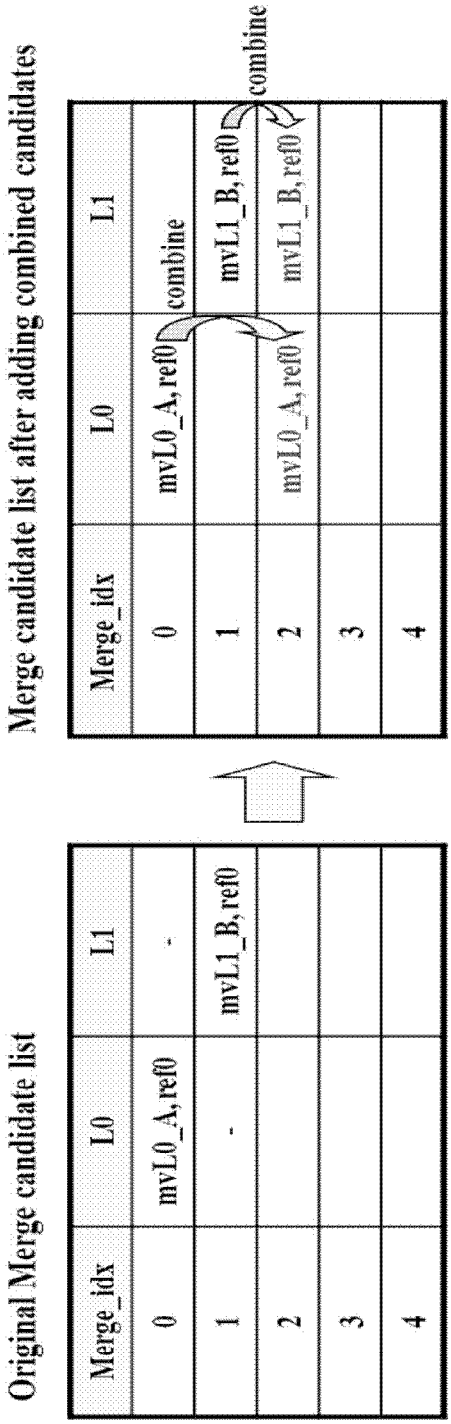

Original Merge candidate list
Merge_idx
L0
L1
0
mvL0_A, ref0
-
1
-
mvL1_B, ref0
2
3
4
Merge candidate list after adding combined candidates
Merge_idx
L0
L1
0
mvL0_A, ref0
combine
1
mvL1_B, ref0
combine
2
mvL0_A, ref0
mvL1_B, ref0
3
4

FIG. 8

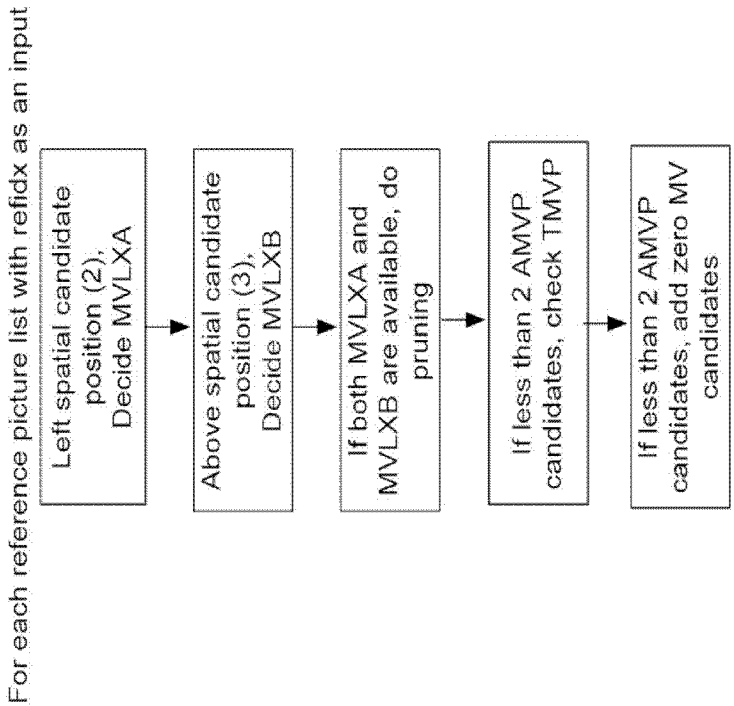

For each reference picture list with refidx as an input
Left spatial candidate position (2), Decide MVLXA
Above spatial candidate position (3), Decide MVLXB
If both MVLXA and MVLXB are available, do pruning
If less than 2 AMVP candidates, check TMVP
If less than 2 AMVP candidates, add zero MV candidates

FIG. 9

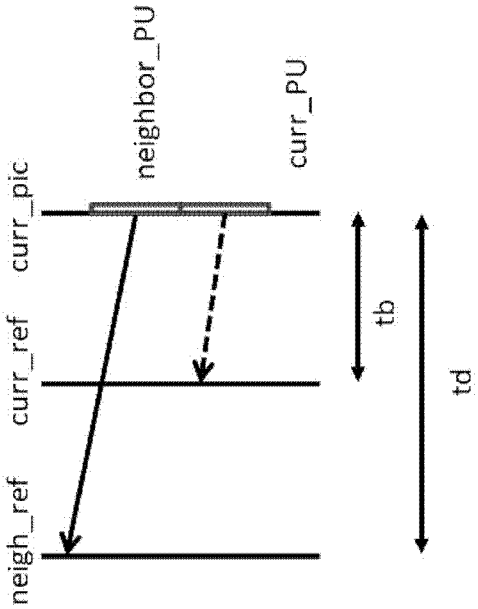


FIG. 10

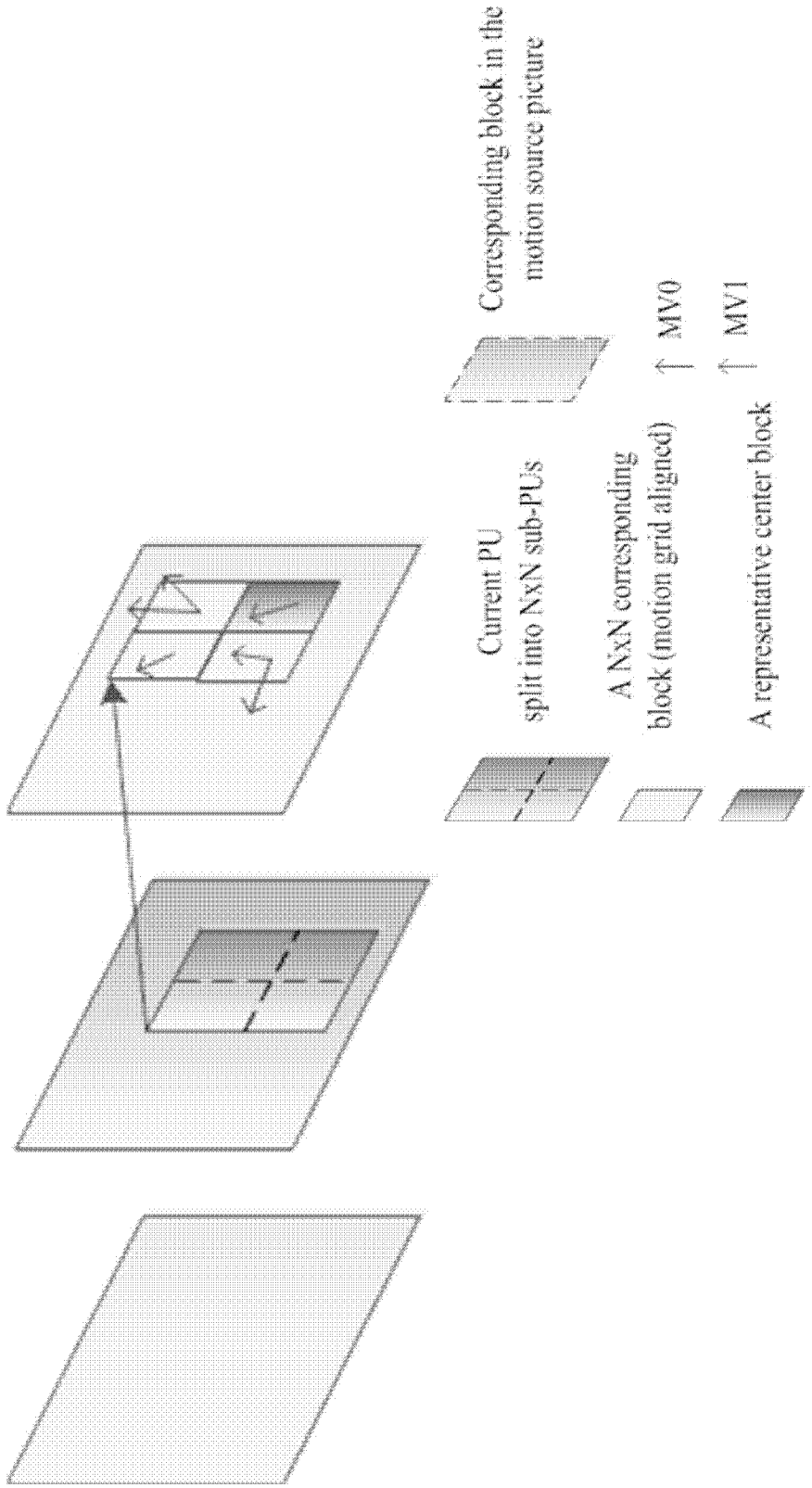

Current PU
split into NxN sub-PUs
A NxN corresponding
block (motion grid aligned)
A representative center block
Corresponding block in the
motion source picture
MV0
MV1

FIG. 11

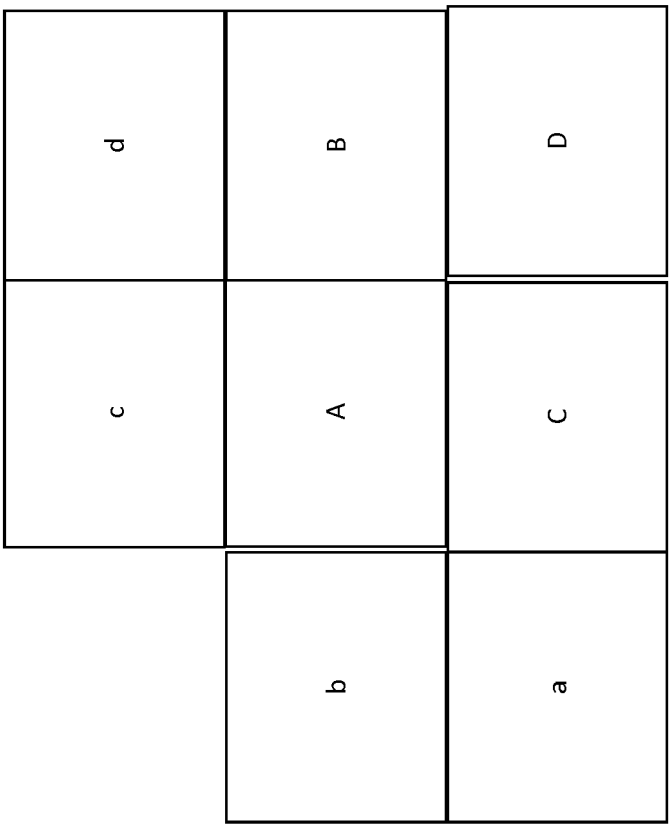


FIG. 12

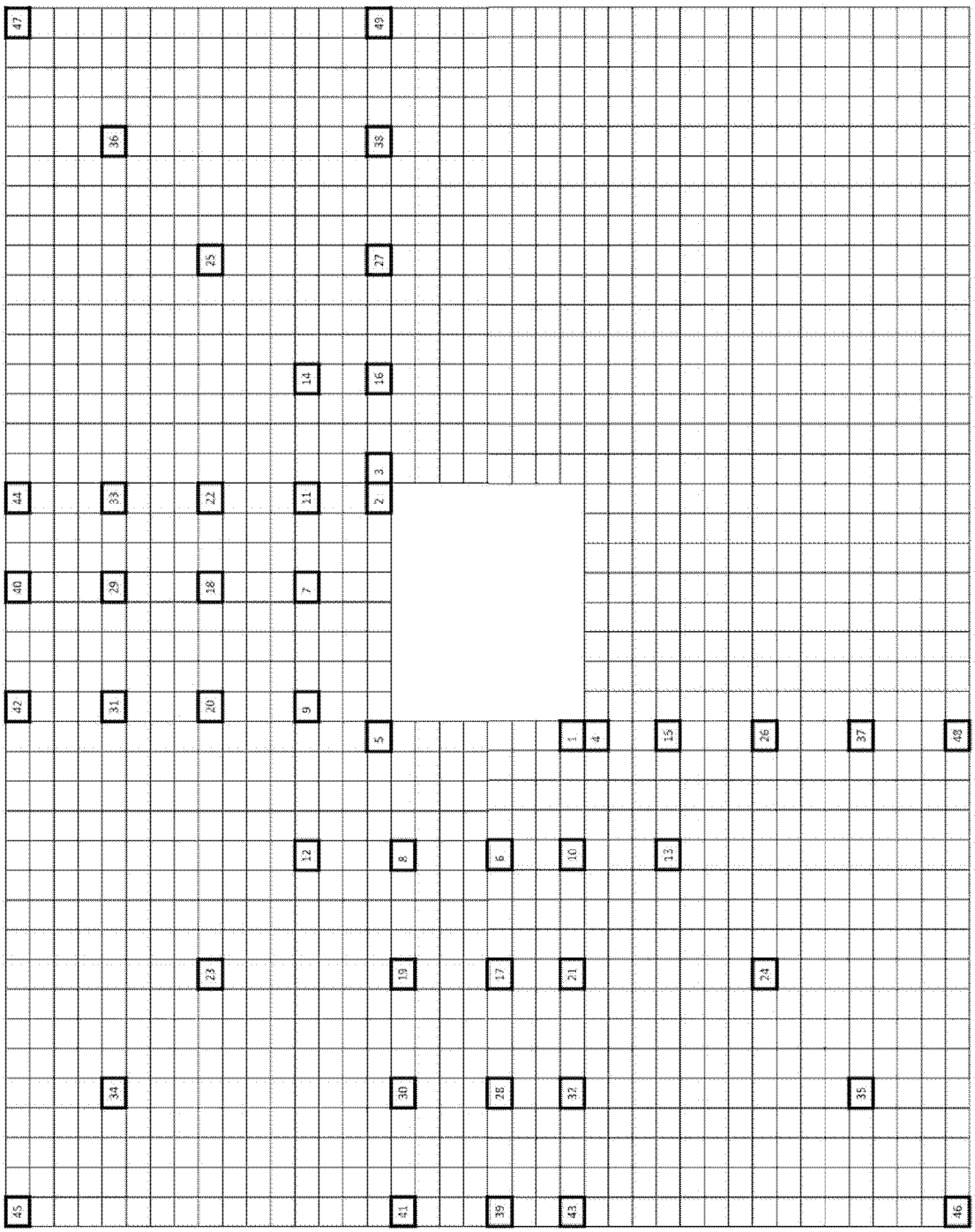

47
44
40
42
45
36
33
29
31
34
25
22
18
20
23
14
11
7
9
12
49
38
27
16
3
2
5
8
19
30
41
1
4
6
17
28
39
10
21
32
43
15
13
26
24
37
35
48
46

FIG. 13

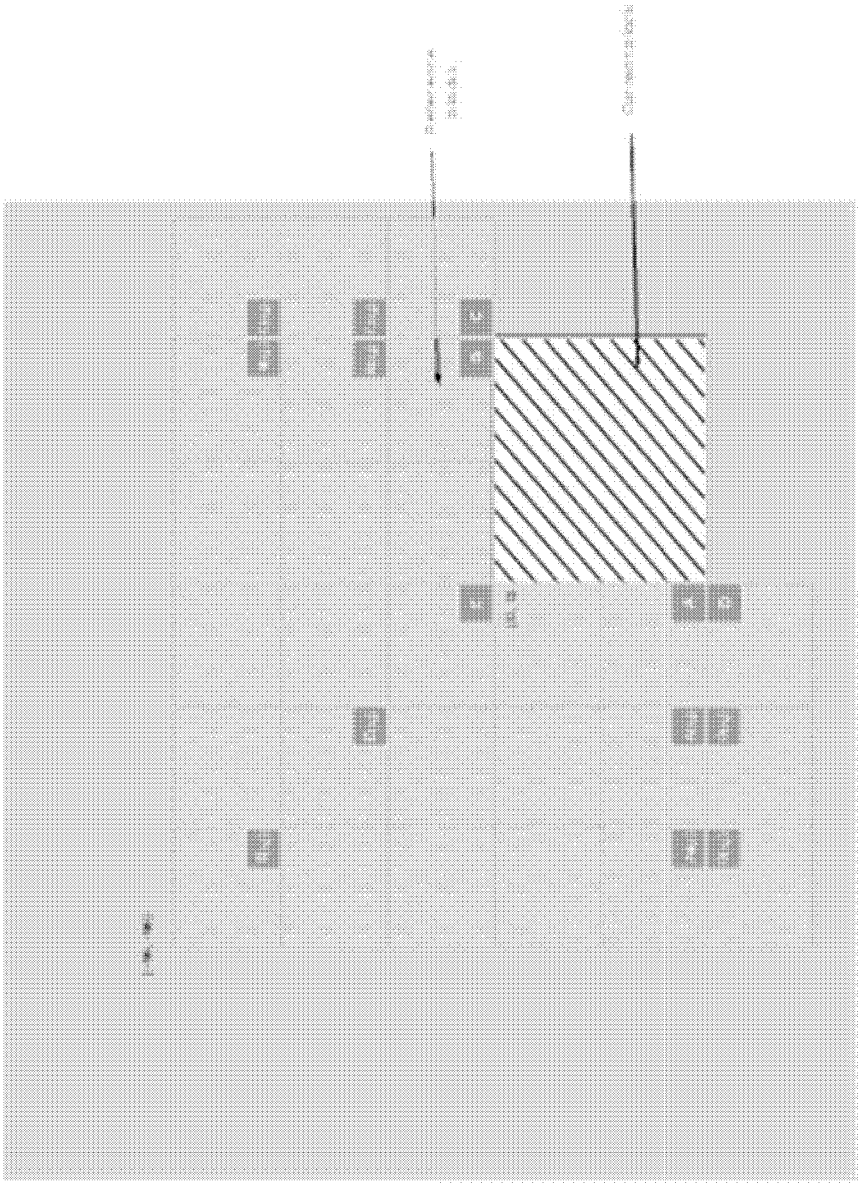


FIG. 14

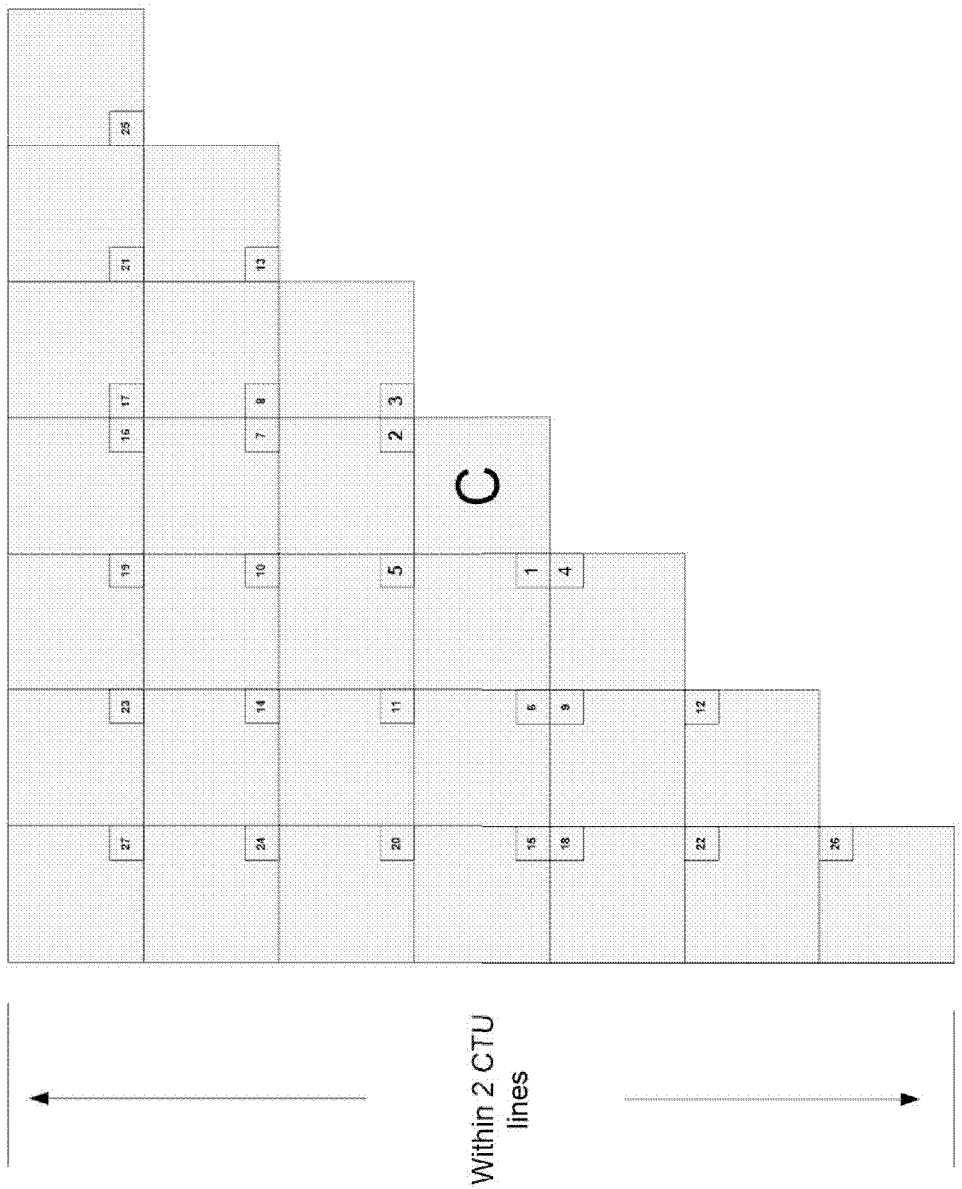

Within 2 CTU lines
C
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

FIG. 15

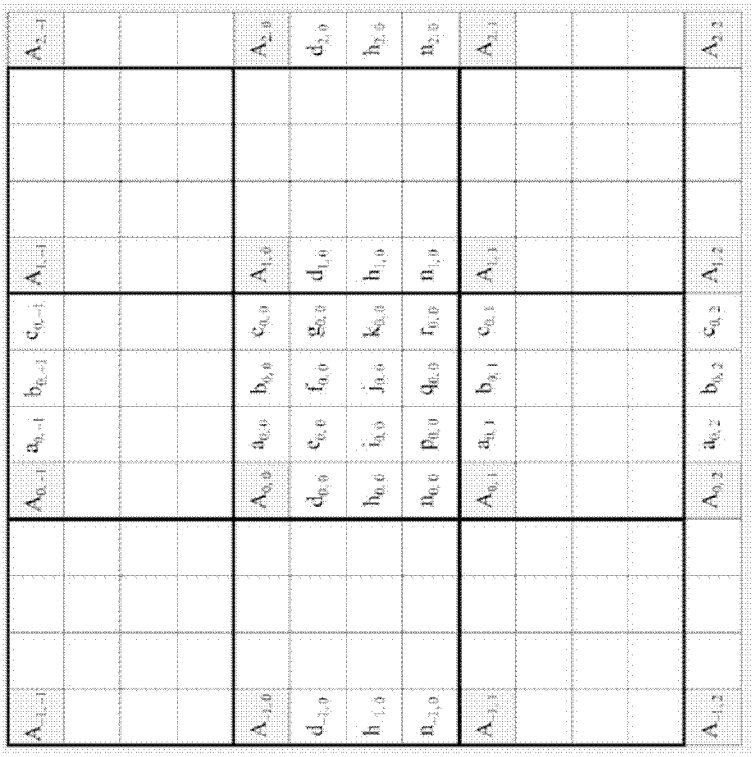

$A_{-1,-1}$
$A_{0,-1}$
$a_{0,-1}$
$b_{0,-1}$
$c_{0,-1}$
$A_{1,-1}$
$A_{-1,0}$
$d_{-1,0}$
$h_{-1,0}$
$n_{-1,0}$
$A_{0,0}$
$a_{0,0}$
$b_{0,0}$
$c_{0,0}$
$d_{0,0}$
$e_{0,0}$
$f_{0,0}$
$g_{0,0}$
$h_{0,0}$
$i_{0,0}$
$j_{0,0}$
$k_{0,0}$
$n_{0,0}$
$p_{0,0}$
$q_{0,0}$
$r_{0,0}$
$A_{1,0}$
$d_{1,0}$
$h_{1,0}$
$n_{1,0}$
$A_{2,0}$
$d_{2,0}$
$h_{2,0}$
$n_{2,0}$
$A_{-1,1}$
$A_{0,1}$
$a_{0,1}$
$b_{0,1}$
$c_{0,1}$
$A_{1,1}$
$A_{2,1}$
$A_{-1,2}$
$A_{0,2}$
$a_{0,2}$
$b_{0,2}$
$c_{0,2}$
$A_{1,2}$
$A_{2,2}$

FIG. 16

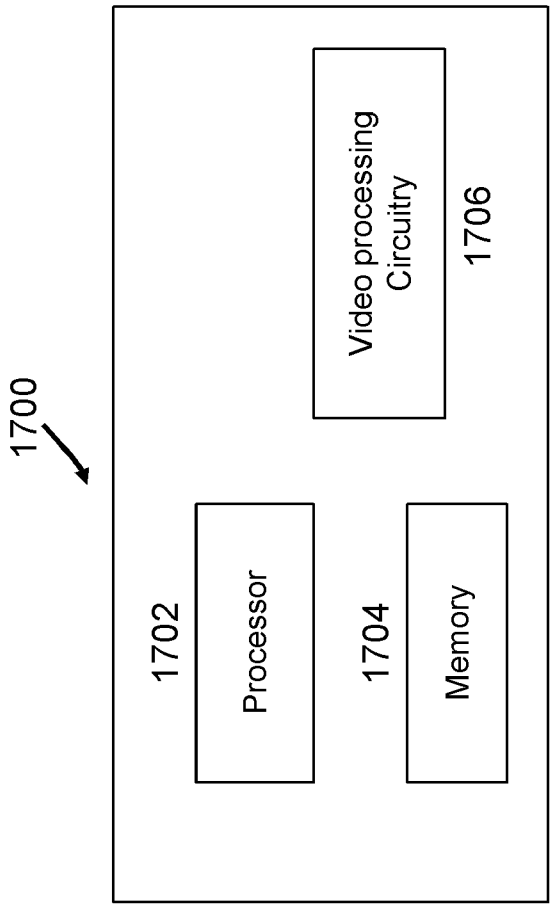

1700
1702
Processor
1704
Memory
Video processing Circuitry
1706

FIG. 17

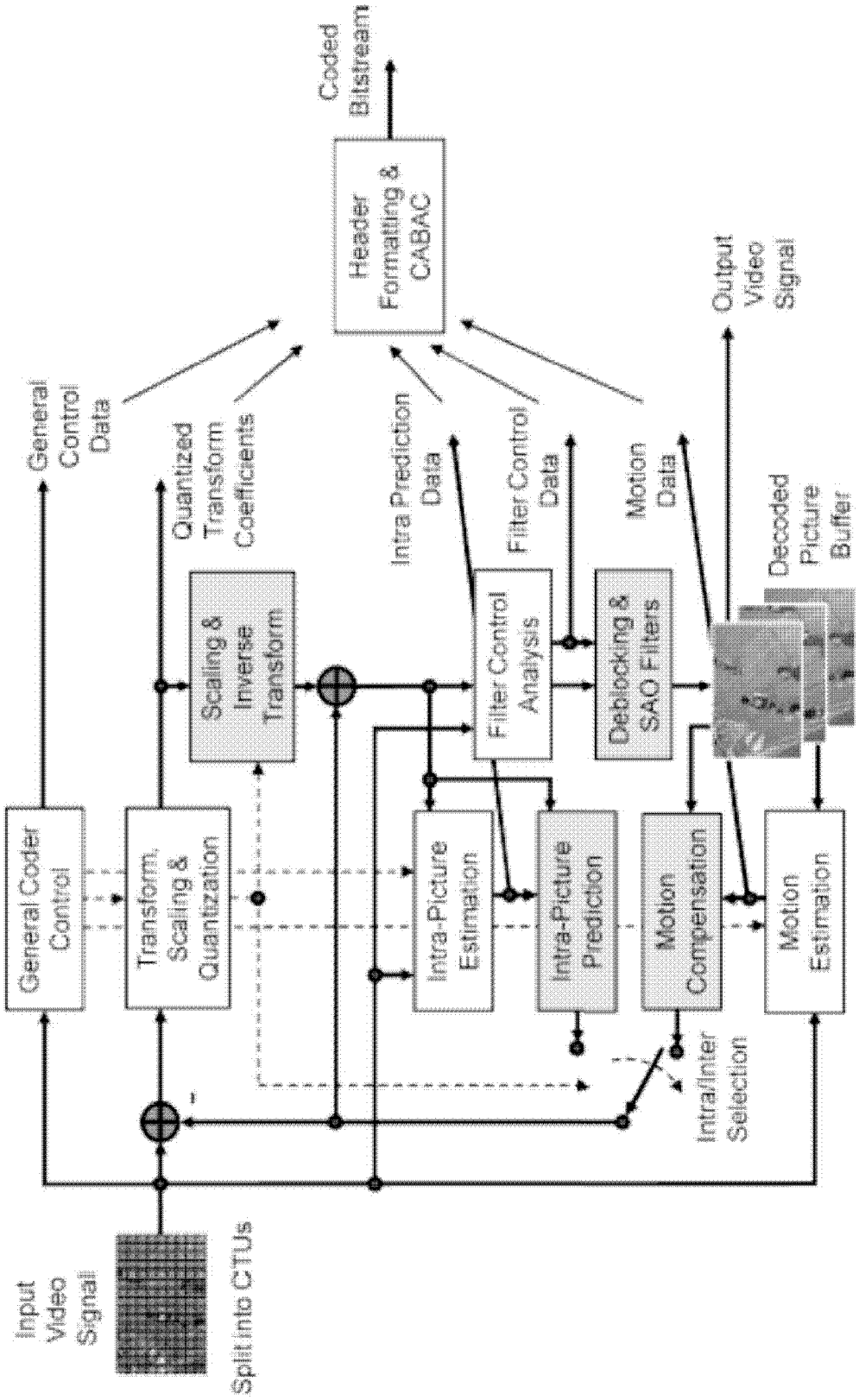

Coded Bitstream
Header Formatting & CABAC
General Control Data
Quantized Transform Coefficients
Intra Prediction Data
Filter Control Data
Motion Data
Output Video Signal
Decoded Picture Buffer
Scaling & Inverse Transform
Filter Control Analysis
Deblocking & SAO Filters
General Coder Control
Transform,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

FIG. 18

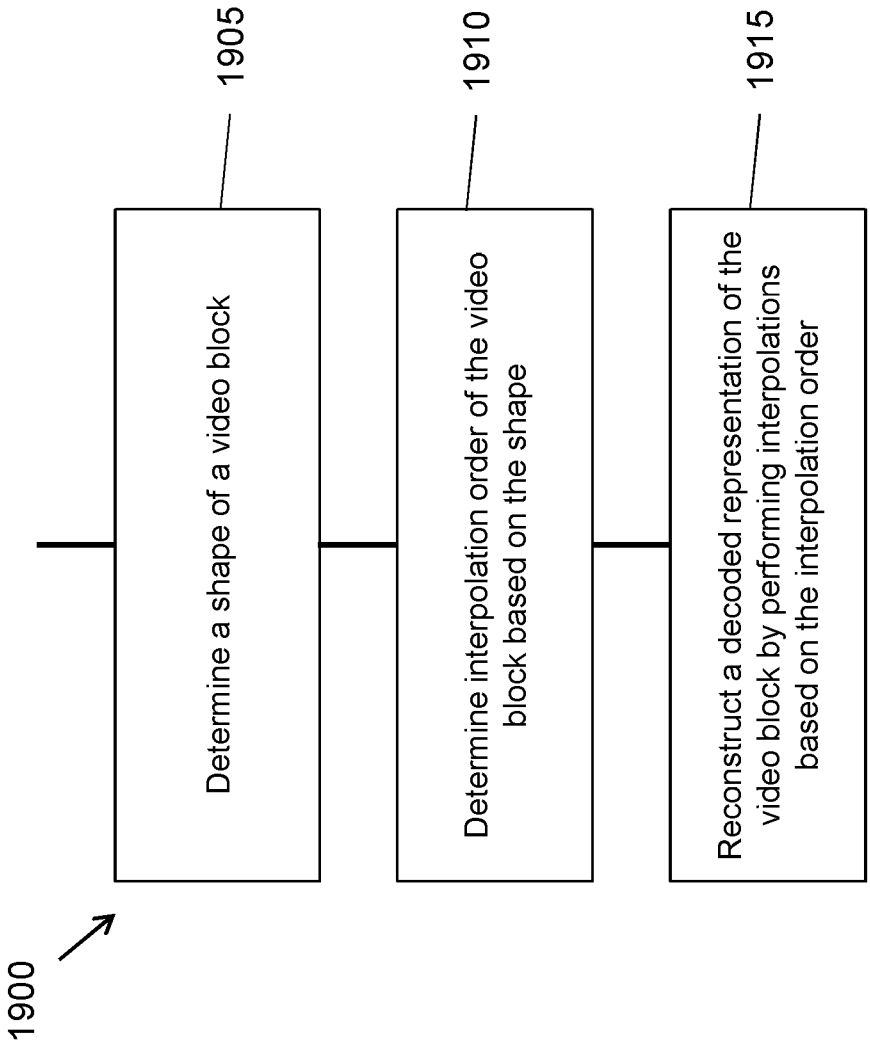

1900
1905
Determine a shape of a video block
1910
Determine interpolation order of the video block based on the shape
1915
Reconstruct a decoded representation of the video block by performing interpolations based on the interpolation order

FIG. 19

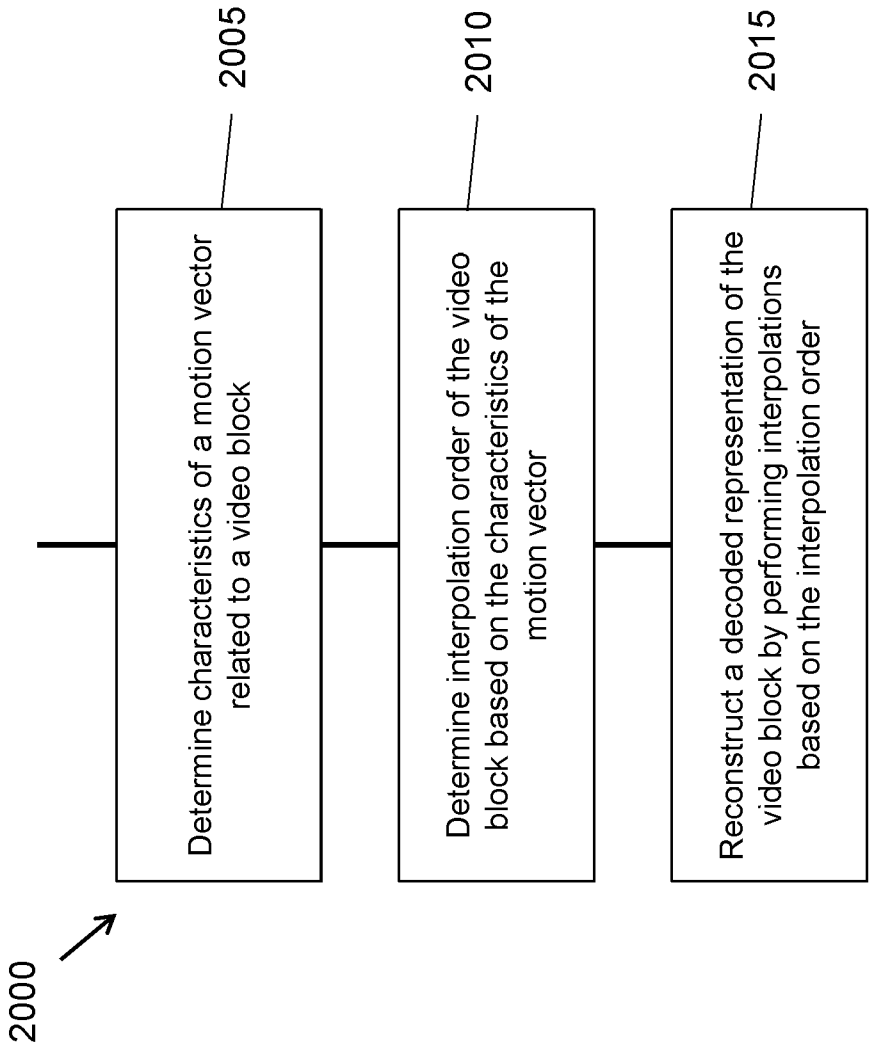

2000
2005
Determine characteristics of a motion vector related to a video block
2010
Determine interpolation order of the video block based on the characteristics of the motion vector
2015
Reconstruct a decoded representation of the video block by performing interpolations based on the interpolation order

FIG. 20

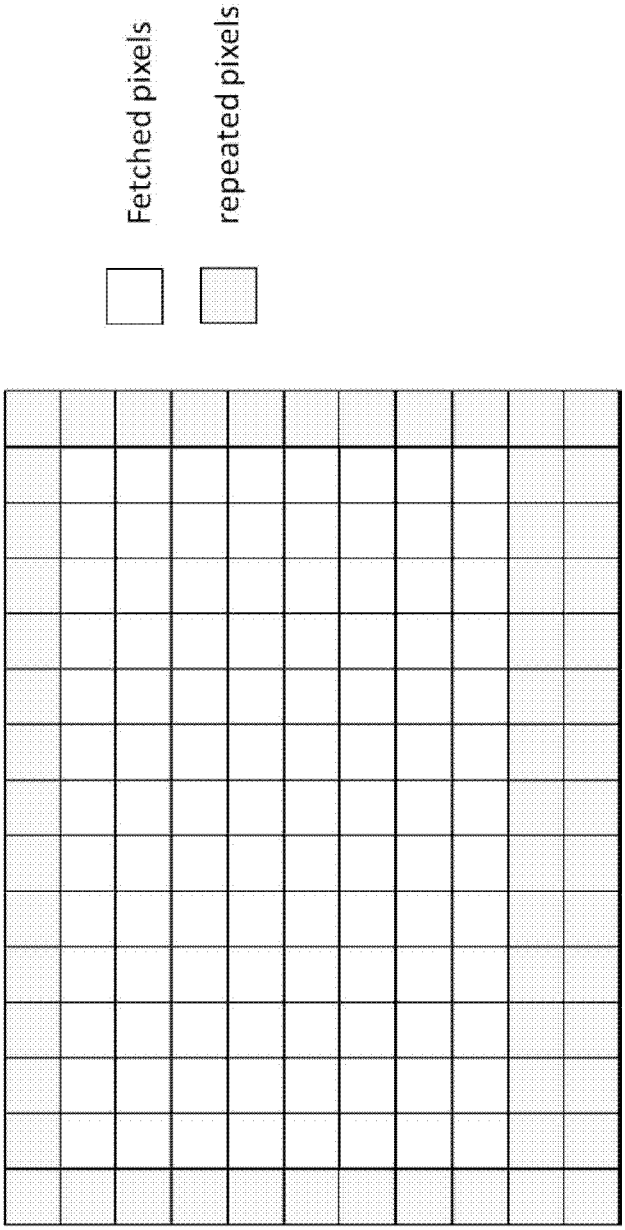


FIG. 21

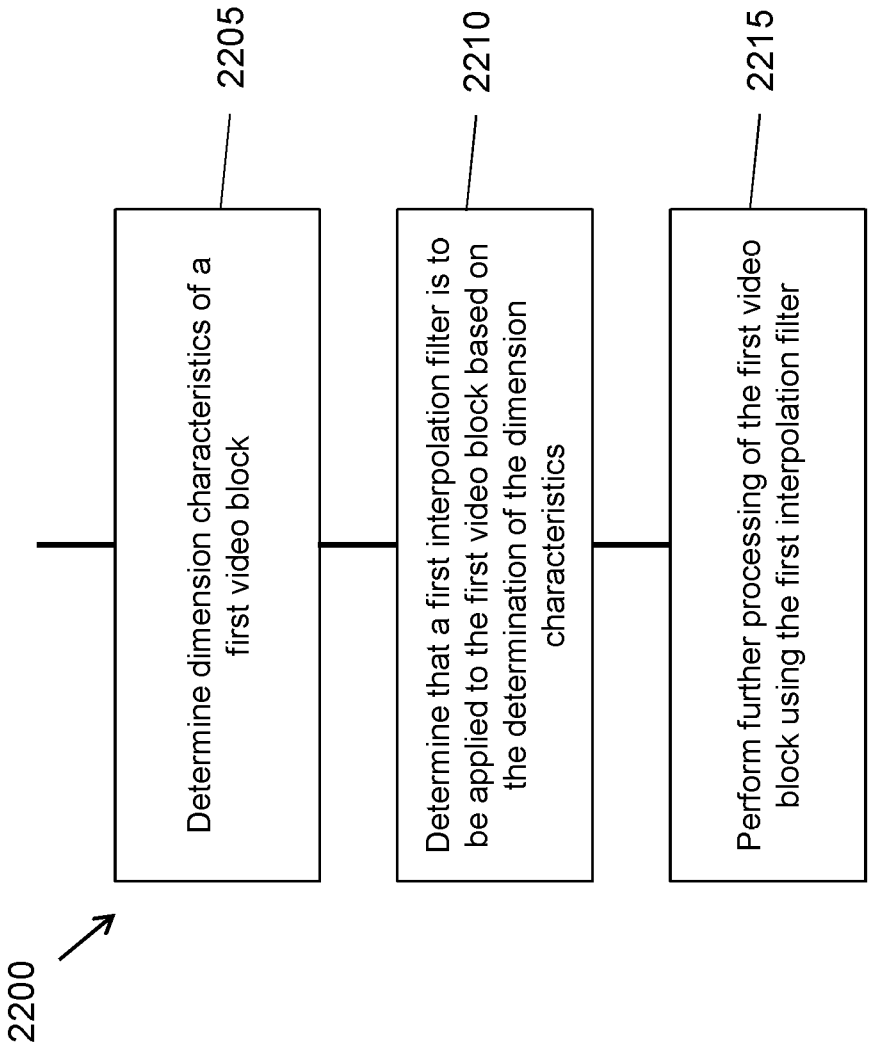

2200
2205
Determine dimension characteristics of a first video block
2210
Determine that a first interpolation filter is to be applied to the first video block based on the determination of the dimension characteristics
2215
Perform further processing of the first video block using the first interpolation filter

FIG. 22

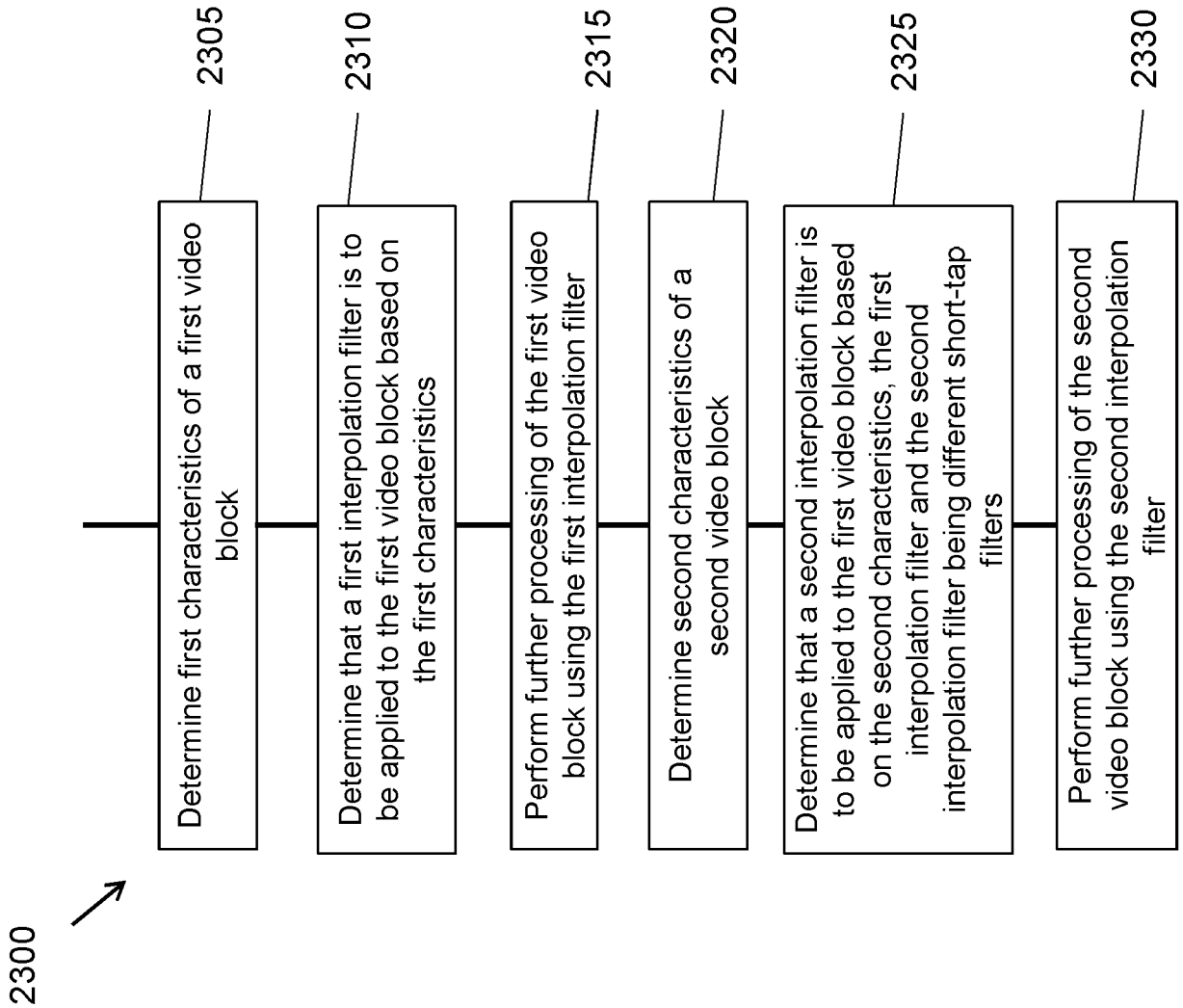

2300
Determine first characteristics of a first video block
2305
Determine that a first interpolation filter is to be applied to the first video block based on the first characteristics
2310
Perform further processing of the first video block using the first interpolation filter
2315
Determine second characteristics of a second video block
2320
Determine that a second interpolation filter is to be applied to the first video block based on the second characteristics, the first interpolation filter and the second interpolation filter being different short-tap filters
2325
Perform further processing of the second video block using the second interpolation filter
2330

FIG. 23

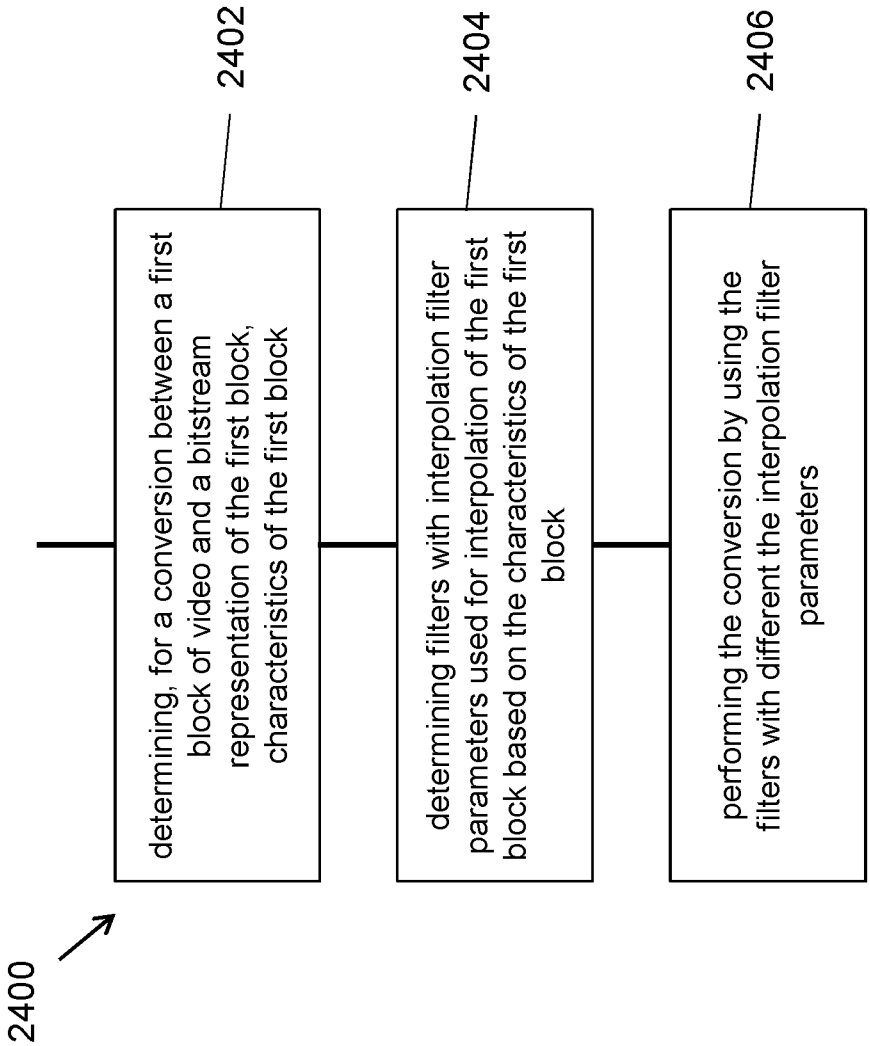

2400
2402
determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block
2404
determining filters with interpolation filter parameters used for interpolation of the first block based on the characteristics of the first block
2406
performing the conversion by using the filters with different the interpolation filter parameters

FIG. 24

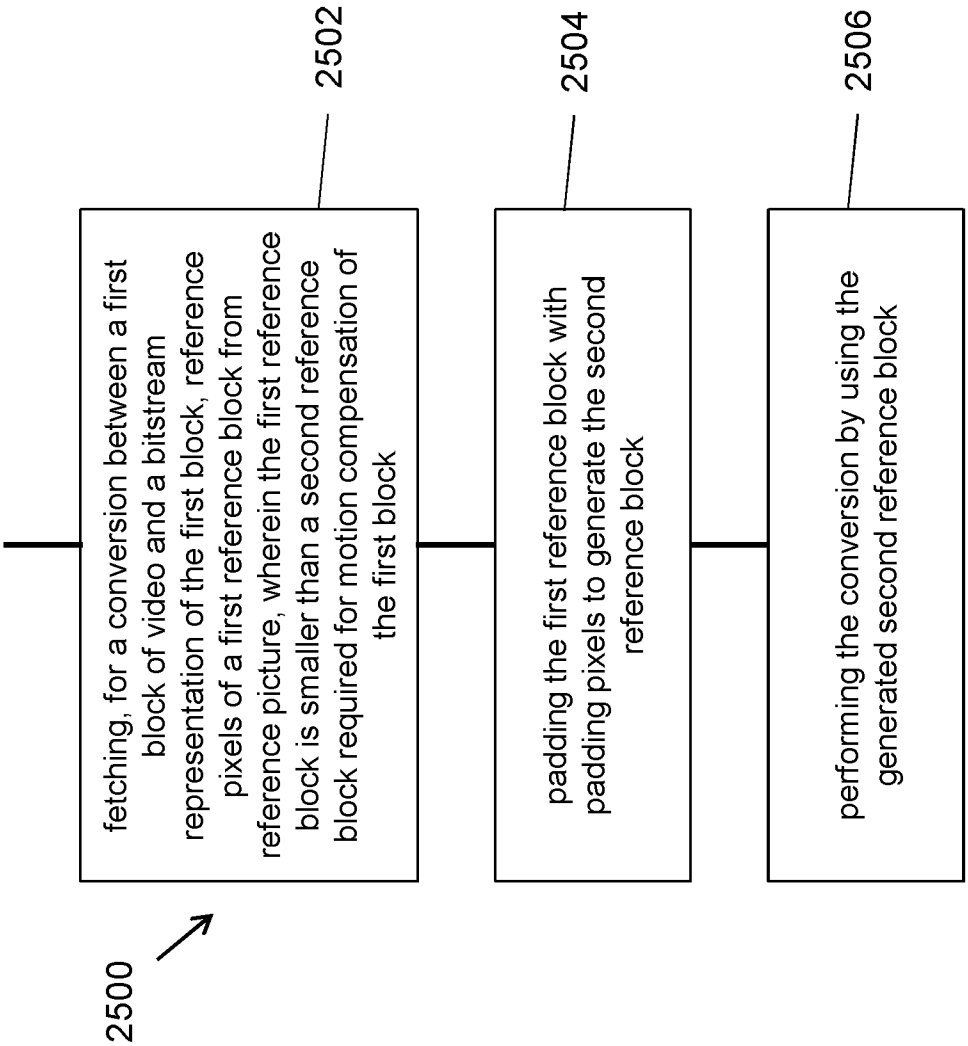

2500
fetching, for a conversion between a first block of video and a bitstream representation of the first block, reference pixels of a first reference block from reference picture, wherein the first reference block is smaller than a second reference block required for motion compensation of the first block
2502
padding the first reference block with padding pixels to generate the second reference block
2504
performing the conversion by using the generated second reference block
2506

FIG. 25

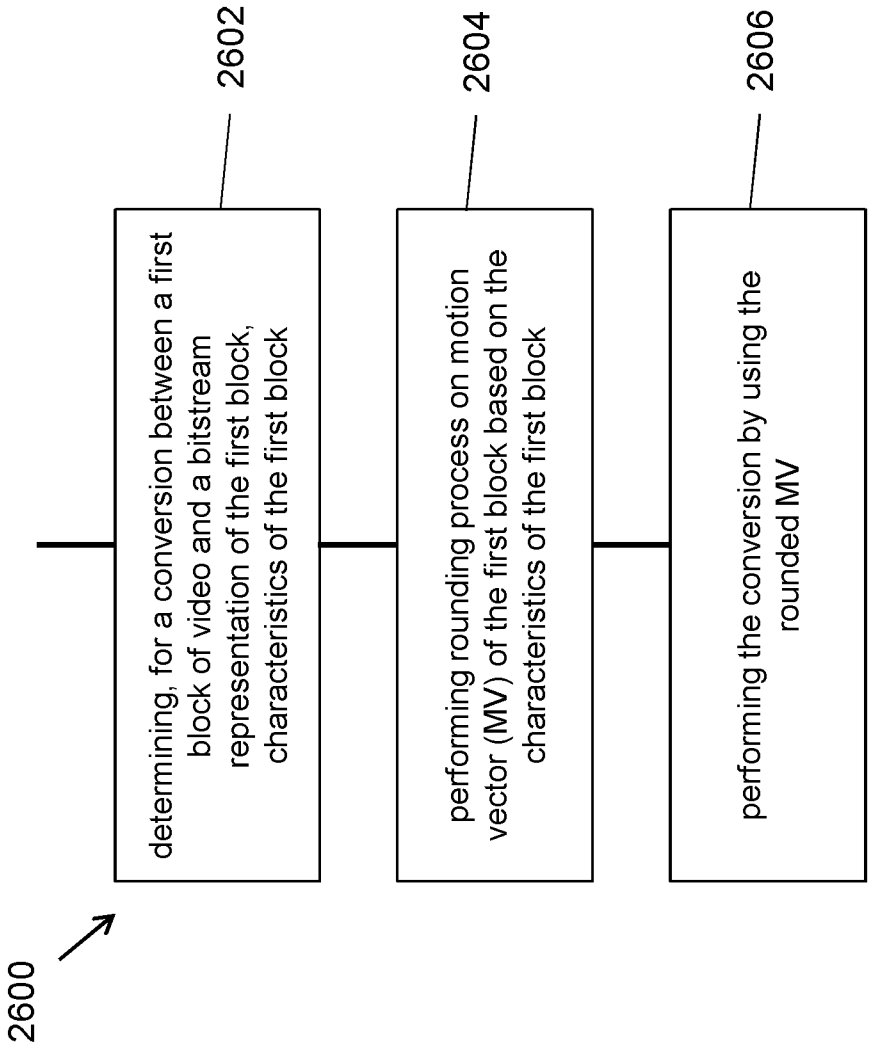

2600
determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block
2602
performing rounding process on motion vector (MV) of the first block based on the characteristics of the first block
2604
performing the conversion by using the rounded MV
2606

FIG. 26

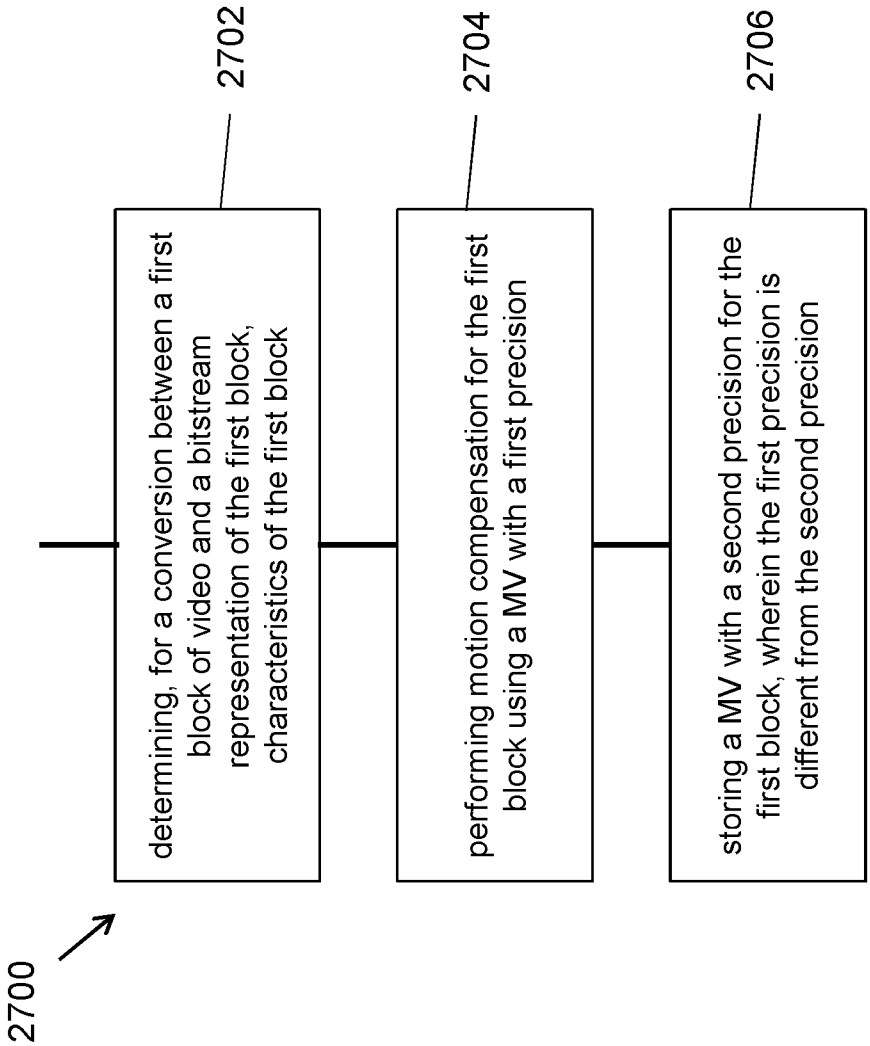

2700
2702
determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block
2704
performing motion compensation for the first block using a MV with a first precision
2706
storing a MV with a second precision for the first block, wherein the first precision is different from the second precision

FIG. 27

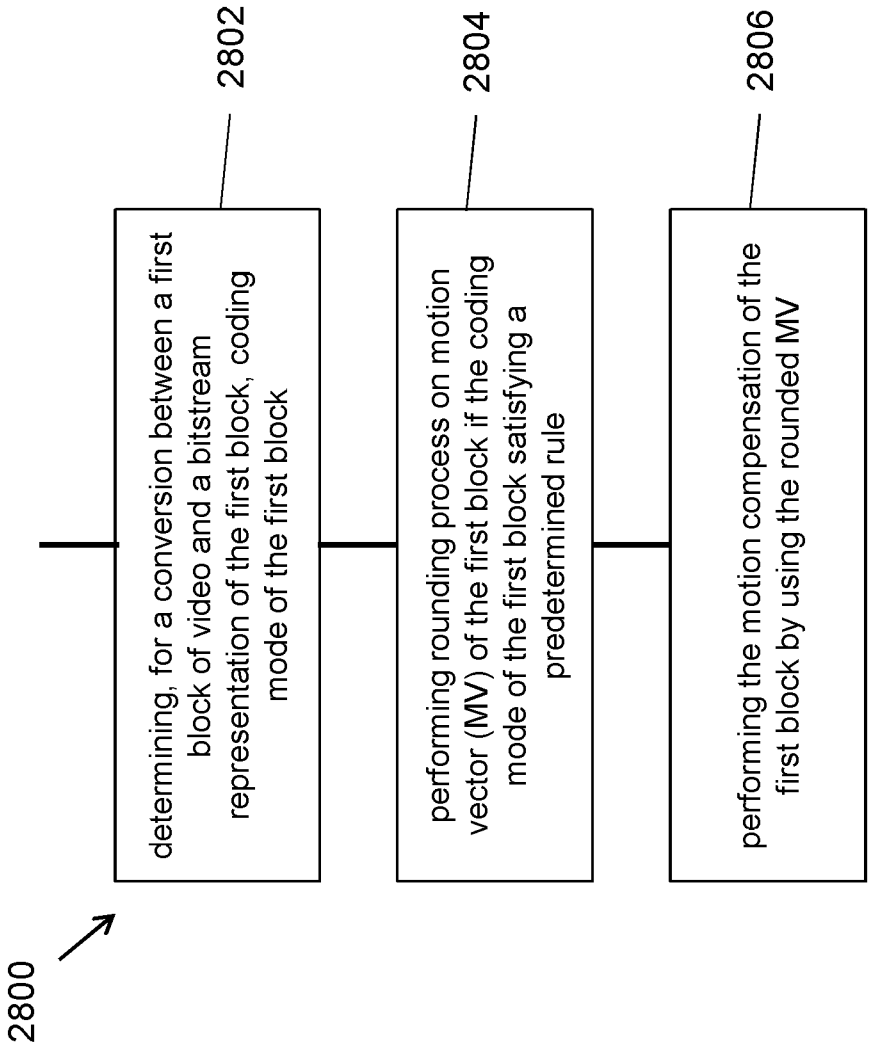

2800
2802
determining, for a conversion between a first block of video and a bitstream representation of the first block, coding mode of the first block
2804
performing rounding process on motion vector (MV) of the first block if the coding mode of the first block satisfying a predetermined rule
2806
performing the motion compensation of the first block by using the rounded MV

FIG. 28

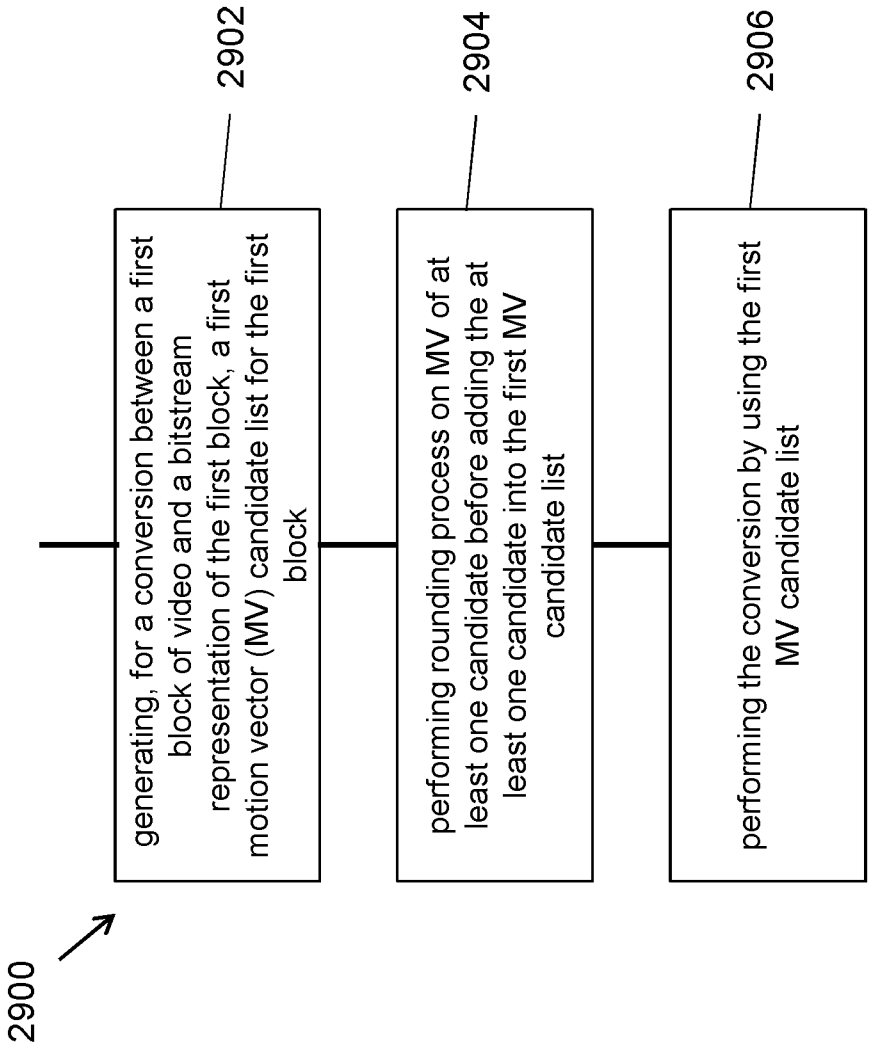

2900
generating, for a conversion between a first block of video and a bitstream representation of the first block, a first motion vector (MV) candidate list for the first block
2902
performing rounding process on MV of at least one candidate before adding the at least one candidate into the first MV candidate list
2904
performing the conversion by using the first MV candidate list
2906

FIG. 29

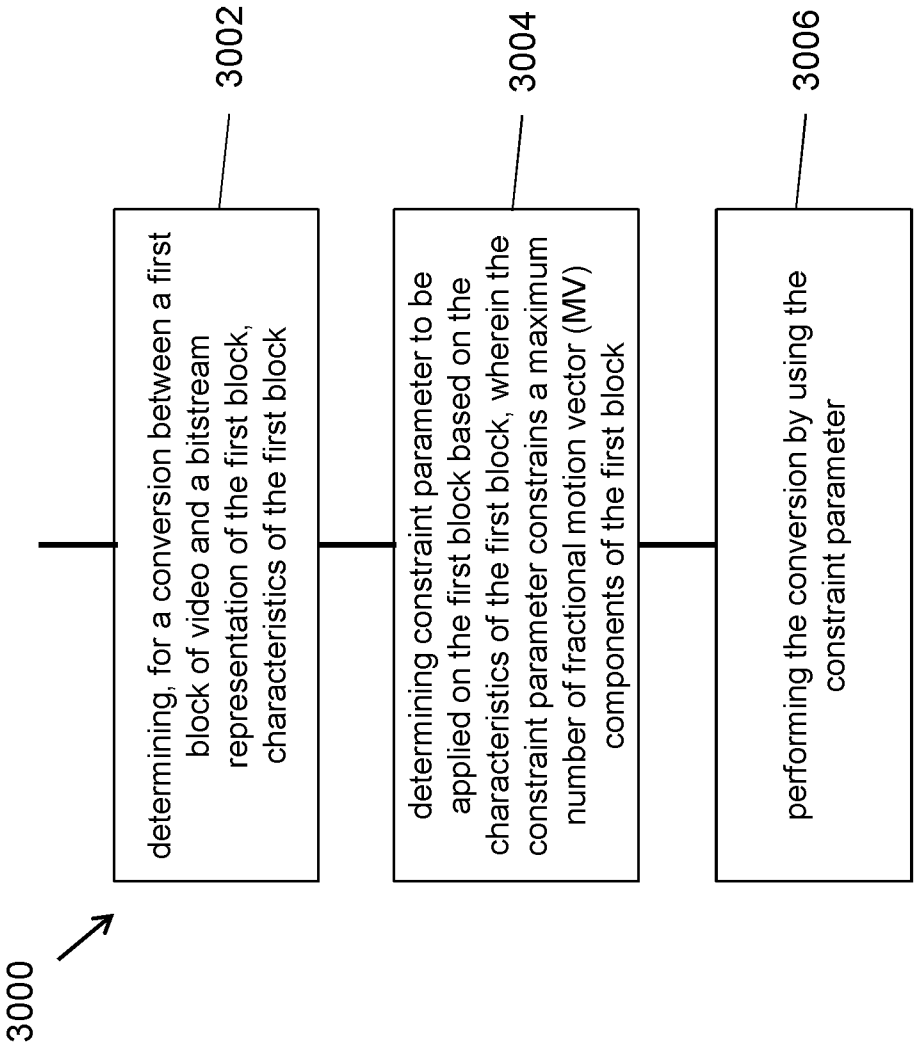

3000
3002
determining, for a conversion between a first block of video and a bitstream representation of the first block, characteristics of the first block
3004
determining constraint parameter to be applied on the first block based on the characteristics of the first block, wherein the constraint parameter constrains a maximum number of fractional motion vector (MV) components of the first block
3006
performing the conversion by using the constraint parameter

FIG. 30

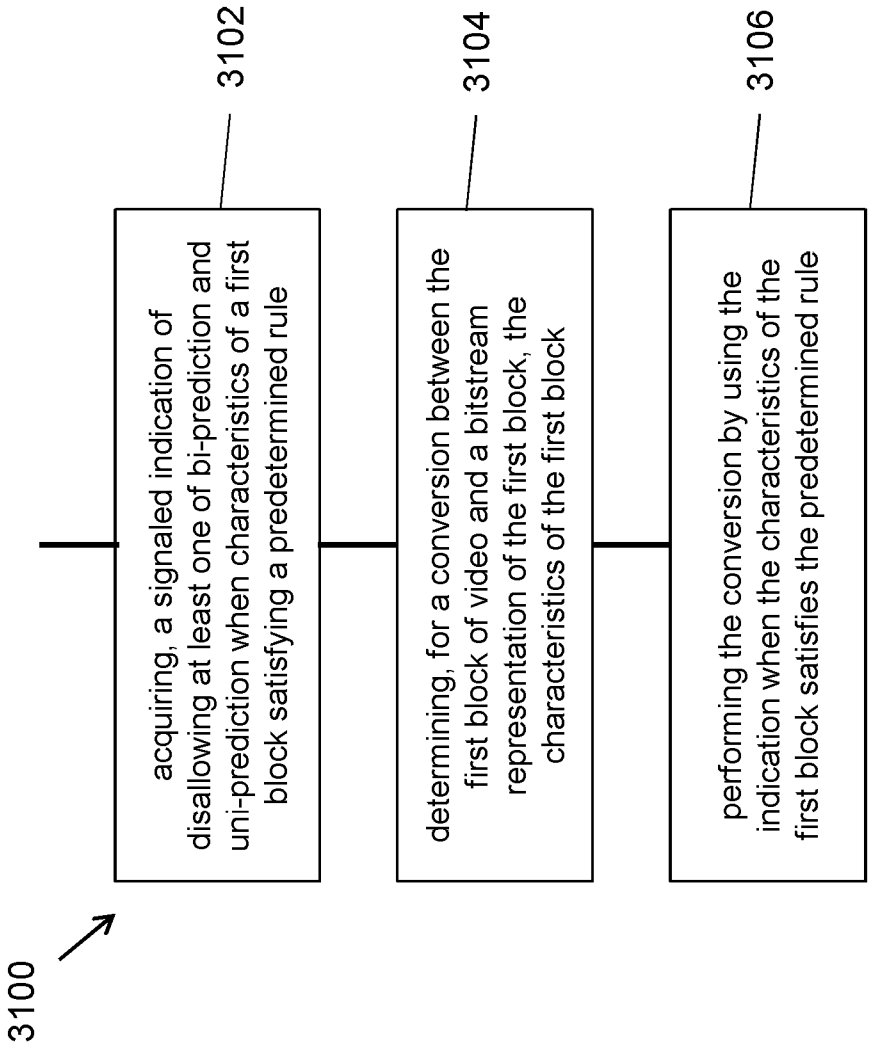

3100
acquiring, a signaled indication of disallowing at least one of bi-prediction and uni-prediction when characteristics of a first block satisfying a predetermined rule
3102
determining, for a conversion between the first block of video and a bitstream representation of the first block, the characteristics of the first block
3104
performing the conversion by using the indication when the characteristics of the first block satisfies the predetermined rule
3106

FIG. 31

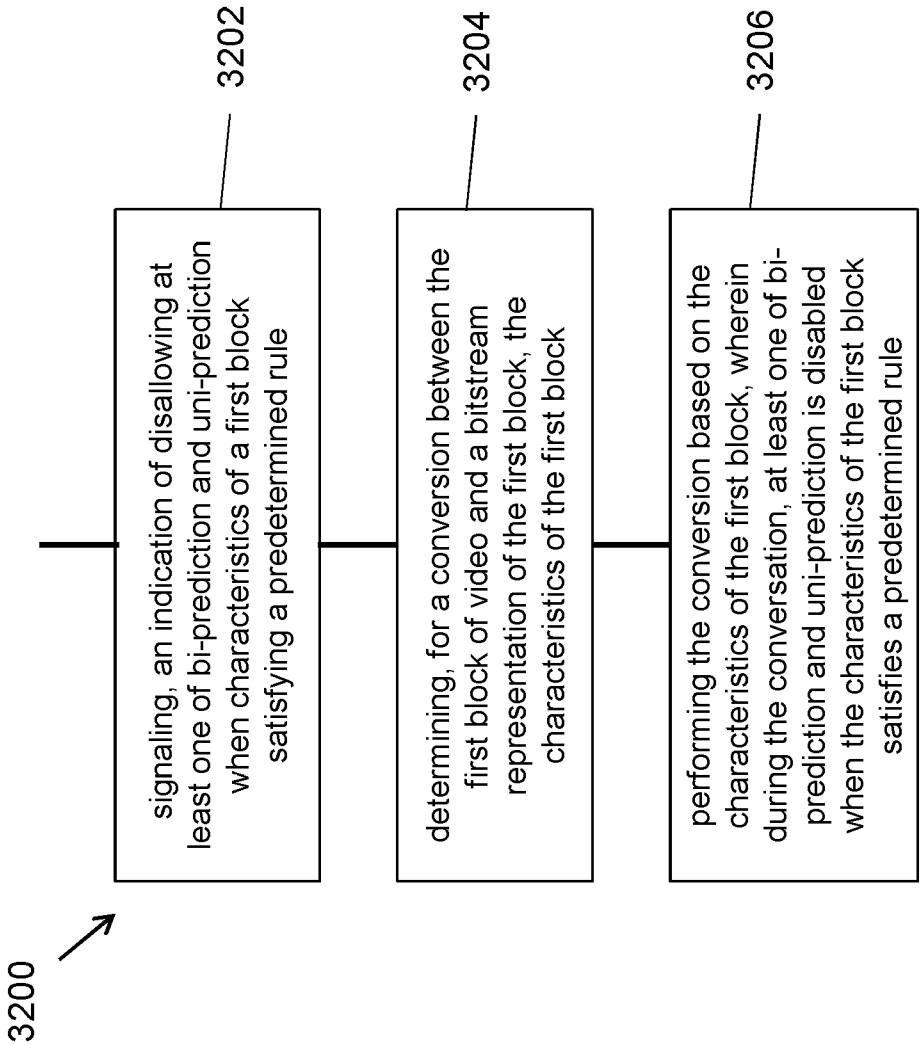

3200
signaling, an indication of disallowing at least one of bi-prediction and uni-prediction when characteristics of a first block satisfying a predetermined rule
3202
determining, for a conversion between the first block of video and a bitstream representation of the first block, the characteristics of the first block
3204
performing the conversion based on the characteristics of the first block, wherein during the conversation, at least one of bi-prediction and uni-prediction is disabled when the characteristics of the first block satisfies a predetermined rule
3206

FIG. 32

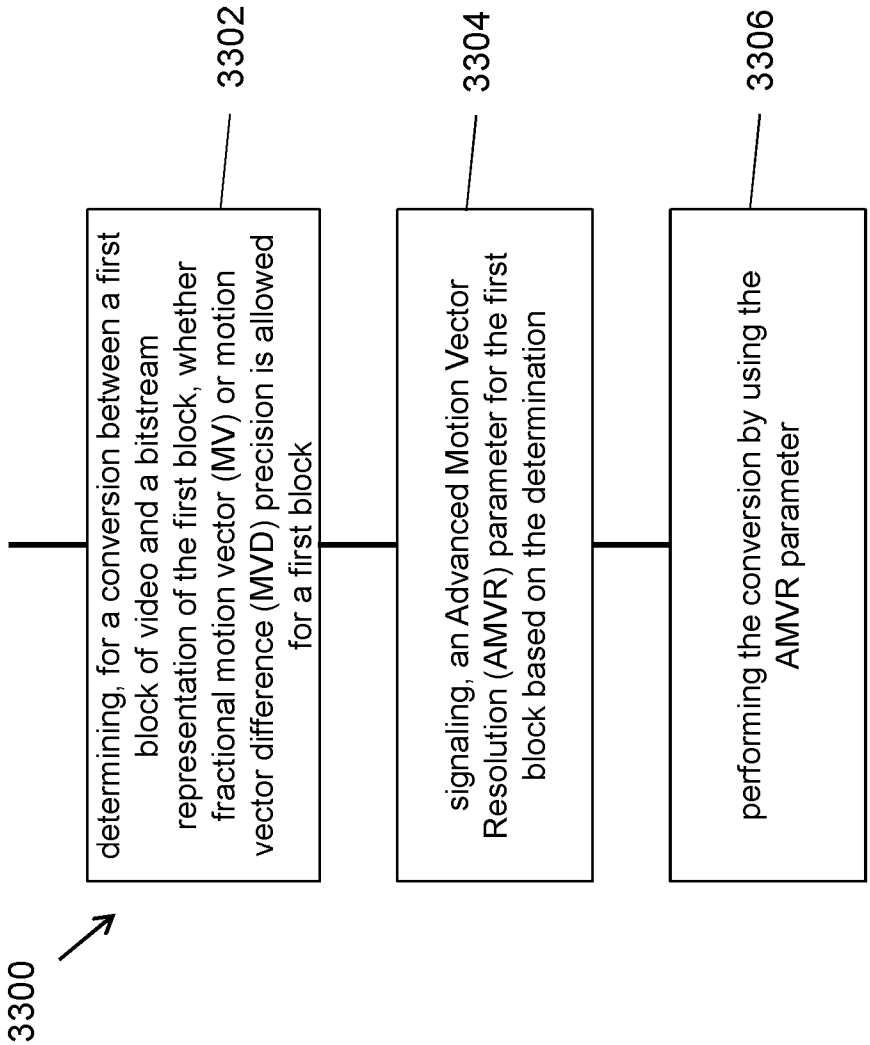

3300
determining, for a conversion between a first block of video and a bitstream representation of the first block, whether fractional motion vector (MV) or motion vector difference (MVD) precision is allowed for a first block
3302
signaling, an Advanced Motion Vector Resolution (AMVR) parameter for the first block based on the determination
3304
performing the conversion by using the AMVR parameter
3306

FIG. 33

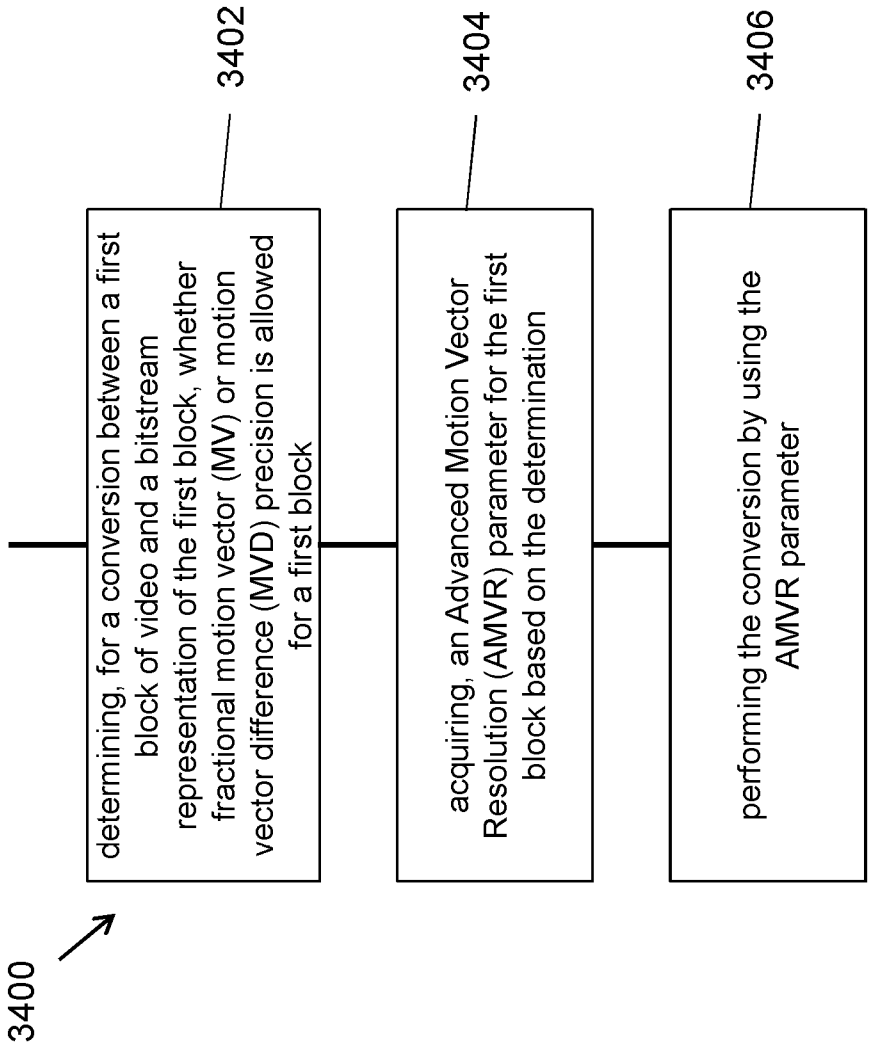

3400
determining, for a conversion between a first block of video and a bitstream representation of the first block, whether fractional motion vector (MV) or motion vector difference (MVD) precision is allowed for a first block
3402
acquiring, an Advanced Motion Vector Resolution (AMVR) parameter for the first block based on the determination
3404
performing the conversion by using the AMVR parameter
3406

FIG. 34

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/071774

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/105(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)

DWPI;EPTXT;VEN;USTXT;WOTXT;CNABS;CNTXT;JVET:disallow, prohibit, forbid, bi, uni, direction, prediction, AMVR, fractional, shape, size, affine

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2015373358 A1 (QUALCOMM INC) 24 December 2015 (2015-12-24) description paragraphs [0002]-[0281], figures 5-6	1-7, 11-12, 16-19
Y	US 2015373358 A1 (QUALCOMM INC) 24 December 2015 (2015-12-24) description paragraphs [0002]-[0281], figures 5-6	8-10
X	WO 2017157259 A1 (MEDIATEK INC) 21 September 2017 (2017-09-21) description paragraphs [0058]-[0060]	13-15
Y	WO 2017157259 A1 (MEDIATEK INC) 21 September 2017 (2017-09-21) description paragraphs [0058]-[0060]	8-10
X	US 2017034526 A1 (QUALCOMM INC) 02 February 2017 (2017-02-02) description paragraphs [0002]-[0200]	1-7, 11-12, 16-19
X	US 2013272415 A1 (TEXAS INSTRUMENTS INC) 17 October 2013 (2013-10-17) description paragraphs [0002]-[0087]	1-7, 11-12, 16-19
A	WO 2016165623 A1 (MEDIATEK INC) 20 October 2016 (2016-10-20) the whole document	1-19

☐ 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
26 March 2020	**03 April 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 **SHANG,Qin** Telephone No. **86-(010)-62089385**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/071774

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

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

[1] I: Claims 1,2 directed to disallowing signal of bi-prediction and uni-prediction;

[2] II: Claims 13,14 directed to AMVR parameter for allowing fractional MV or MVD precision;

[3] The only one same technical features among these claims are: "determining, for a conversion between the first block of video and a bitstream representation of the first block" and "performing the conversion". However these features are the common knowledge in the art. Therefore the common or corresponding special technical features are not included in the two groups of claims. The inventions are not so linked as to form a single general inventive concept, as required by Rule 13.1 PCT.

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

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

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

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

Remark on Protest

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

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

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/071774

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2015373358	A1	24 December 2015	CA	2952457	A1	23 December 2015
				JP	2017522789	A	10 August 2017
				US	10327002	B2	18 June 2019
				WO	2015196030	A1	23 December 2015
				US	10327001	B2	18 June 2019
				CA	2952279	A1	23 December 2015
				WO	2015196029	A1	23 December 2015
				KR	20170019364	A	21 February 2017
				BR	112016029539	A2	22 August 2017
				EP	3158736	B1	27 November 2019
				US	2015373357	A1	24 December 2015
				CN	106416245	B	02 August 2019
				CN	106464874	B	02 August 2019
				CN	106416245	A	15 February 2017
				JP	6527891	B2	05 June 2019
				EP	3158736	A1	26 April 2017
				JP	2017522790	A	10 August 2017
				KR	20170019365	A	21 February 2017
				CN	106464874	A	22 February 2017
				BR	112016029387	A2	22 August 2017
				EP	3158737	A1	26 April 2017
WO	2017157259	A1	21 September 2017	RU	2696551	C1	02 August 2019
				US	2019082191	A1	14 March 2019
				EP	3414900	A4	11 December 2019
				EP	3414900	A1	19 December 2018
				SG	11201806865Y	A	27 September 2018
				CN	108781284	A	09 November 2018
US	2017034526	A1	02 February 2017	KR	20180035810	A	06 April 2018
				EP	3588955	A1	01 January 2020
				TW	201709731	A	01 March 2017
				EP	3329675	B1	11 September 2019
				BR	112018001519	A2	11 September 2018
				CN	107852498	A	27 March 2018
				AU	2016298172	A1	18 January 2018
				WO	2017019656	A1	02 February 2017
				US	10404992	B2	03 September 2019
				JP	2018526881	A	13 September 2018
				EP	3329675	A1	06 June 2018
US	2013272415	A1	17 October 2013	US	10362326	B2	23 July 2019
				US	9591312	B2	07 March 2017
				US	2017180748	A1	22 June 2017
WO	2016165623	A1	20 October 2016	US	2018131963	A1	10 May 2018
				US	10432966	B2	01 October 2019
				EP	3269142	A4	24 October 2018
				US	2019373286	A1	05 December 2019
				EP	3269142	A1	17 January 2018
				CA	2981916	A1	20 October 2016
				CN	107852499	A	27 March 2018
				KR	20170135886	A	08 December 2017