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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
08 October 2020 (08.10.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/200238 A1

(51) International Patent Classification:
H04N 19/00 (2014.01)

(21) International Application Number:
PCT/CN2020/082754

(22) International Filing Date:
01 April 2020 (01.04.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2019/080754
01 April 2019 (01.04.2019) CN

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

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

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

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,

(54) Title: HALF-PEL INTERPOLATION FILTER IN INTER CODING MODE

(57) Abstract: A method of video processing is provided to comprise: performing a conversion between a current video block of a video and a coded representation of the video, wherein the coded representation includes a first parameter that indicates a motion information precision from a multiple precision set for the current video block that does not use a default motion information precision and/or a second parameter which identifies whether an alternative half-pel interpolation filter is used for the conversion, and wherein each of the first parameter and/or the second parameter is associated with one or more bins and a context-model based coding is only used for a first bin of the one or more bins in the coded representation.

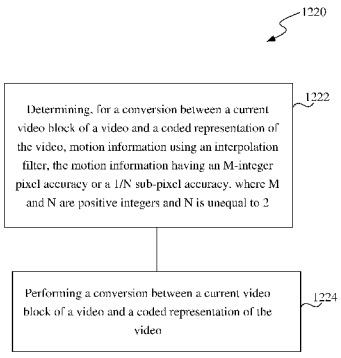


FIG. 12B

WO 2020/200238 A1

[Continued on next page]

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))*

HALF-PEL INTERPOLATION FILTER IN INTER CODING MODE

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefit of International Patent Application No. PCT/CN2019/080754, filed on April 1, 2019. For all purposes under the law, the entire disclosure of the aforementioned application is incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

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

BACKGROUND

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

SUMMARY

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

[0005] In one representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes performing a conversion between a current video block of a video and a coded representation of the video, wherein the coded representation includes a first parameter that indicates a motion information precision from a multiple precision set for the current video block that does not use a default motion information precision and/or a second parameter which identifies whether an alternative half-pel interpolation filter is used for the conversion, and wherein each of the first parameter and/or the second parameter is associated with one or more bins and a

context-model based coding is only used for a first bin of the one or more bins in the coded representation.

[0006] In another aspect, the disclosed technology may be used to provide a method for video processing. This method includes determining, for a conversion between a current video block of a video and a coded representation of the video, motion information using an interpolation filter, the motion information having an M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers and N is unequal to 2; and performing the conversion based on the motion information; wherein a syntax field in the coded representation corresponds to the interpolation filter.

[0007] In another aspect, the disclosed technology may be used to provide a method for video processing. This method includes generating, for a conversion between a current video block of a video and a coded representation of the video, a first motion candidate based on one or more motion candidates in a motion candidate list and one or more interpolation filters associated with the one or more motion candidates; wherein an interpolation filter is assigned to the first motion candidate according to a rule that depends on the one or more interpolation filters associated with the one or more motion candidates; inserting the first motion candidate into the motion candidate list; and performing the conversion based on the motion candidate list

[0008] In another aspect, the disclosed technology may be used to provide a method for video processing. This method includes associating, with each of multiple blocks comprising a first block in a first video region and a second block in a second video region of a video, motion information that includes information about a respective interpolation filter used for interpolating motion vector difference information for each of the multiple blocks; and performing a conversion between the video and a coded representation of the video using the motion information.

[0009] In another aspect, the disclosed technology may be used to provide a method for video processing. This method includes performing a conversion between a current video block of a video and a coded representation of the video, wherein the current video block is represented in the coded representation using a merge mode or a merge with motion vector different (MMVD) mode, wherein an interpolation filter having a 1/N precision is used for interpolating a prediction block associated with the current video block according to a rule, wherein N is a positive integer; wherein the rule

defines a condition for inheriting interpolation information in case that the interpolation filter is an alternative interpolation filter that is different from a default interpolation filter.

[0010] In another aspect, the disclosed technology may be used to provide a method for video processing. This method includes processing a current video block during a conversion between a video comprising the current video block and a coded representation of the video; and comparing and selectively updating a merge candidate list with motion information of the current video block, wherein the motion information includes interpolation filter information; wherein the interpolation filter information includes parameters of an interpolation filter used for interpolating motion vector difference values used for representing the current video block in the coded representation.

[0011] In one representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes maintaining, prior to a conversion between a current video block of a video region and a coded representation of the video, at least one history-based motion vector prediction (HMVP) table, wherein the HMVP table includes one or more entries corresponding to motion information of one or more previously processed blocks; and performing the conversion using the at least one HMVP table, and wherein the motion information of each entry is configured to include interpolation filter information for the one or more previously processed blocks, and wherein the interpolation filter information indicates interpolation filters used for interpolating prediction blocks of the one or more previously processed blocks.

[0012] In one representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes maintaining, prior to a conversion between a current video block of a video region and a coded representation of the video, at least one history-based motion vector prediction (HMVP) table that includes one or more entries corresponding to motion information of one or more previously processed blocks, and wherein the motion information of each entry is configured to include interpolation filter information for the one or more previously processed blocks; constructing, for the conversion, a motion candidate list which includes an HMVP merge candidate, wherein the HMVP candidate is derived by inheriting one entry from the HMVP table including corresponding interpolation filter information associated with the one entry; and performing the conversion based on the motion candidate list.

[0013] In one representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes deciding a coding mode used for representing a current video block of a video in a coded representation of the video; and coding the current video block into the coded representation according to the coding mode, wherein use of half-pel accuracy for representing motion information is disabled for the current video block due to use of the coding mode.

[0014] In one representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes parsing a coded representation of a video to determine that a current video block of the video is coded using a coding mode; and generating a decoded representation of the current video block from the coded representation according to the coding mode, wherein use of half-pel accuracy for representing motion information is disabled for the current video block due to use of the coding mode.

[0015] In one representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes determining, for a current video block that is processed using a generalized Bi-prediction (GBi) mode, to use a first weighting factor and a second, different weighting factor for a first prediction block and a second prediction block, respectively, the first weighting factor and the second weighting factor selected from weighting factor sets; and performing a conversion between the current video block of a video and a coded representation of the video based on the determining, wherein a first weighting factor set used for an inter mode is different from a second weighting factor set used for an affine inter mode.

[0016] In one representative aspect, the disclosed technology may be used to provide a method for video processing. This method includes determining, for a conversion between multiple video units of a video region of a video and a coded representation of the multiple video units, information related to an interpolation filter depending on a coding mode of a video unit, the interpolation filter used for interpolating motion vector difference values in the coded representation and having a M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers; and performing the conversion using the interpolation filter.

[0017] In one representative aspect, the disclosed technology may be used to provide a video encoding method. This method includes deciding a coding mode used for representing a current video block of a video in a coded representation of the video; and coding the current video block into the

coded representation according to the coding mode, wherein use of alternative half-pel accuracy filters in addition to a default half-pel accuracy filter for representing motion information is disabled for the current video block due to use of the coding mode.

[0018] In one representative aspect, the disclosed technology may be used to provide a video decoding method. This method includes parsing a coded representation of a video to determine that a current video block of the video is coded using a coding mode; and generating a decoded representation of the current video block from the coded representation according to the coding mode, wherein use of alternative half-pel accuracy filters in addition to a default half-pel accuracy filter for representing motion information is disabled for the current video block due to use of the coding mode.

[0019] In one representative aspect, the disclosed technology may be used to provide a video encoding method. This method includes determining, due to use of an alternative interpolation filter being enabled for a video block of a video, that a coding mode is disallowed for coding the video block into a coded representation of the video; and generating the coded representation of the video block based on the determining, wherein the alternative interpolation filter is used for interpolating inter prediction blocks of the current video block.

[0020] In one representative aspect, the disclosed technology may be used to provide a video decoding method. This method includes determining, due to use of an alternative interpolation filter being enabled for a video block of a video, that use of a coding mode for representing the video block in a coded representation of the video is disallowed; and generating a decoded video block by parsing the coded representation based on the determining.

[0021] Further, in a representative aspect, any of the disclosed methods is an encoder-side implementation.

[0022] Also, in a representative aspect, any of the disclosed methods is a decoder-side implementation.

[0023] One of the above-described methods are embodied in the form of processor-executable code and stored in a computer-readable program medium.

[0024] In yet another representative aspect, an apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon is disclosed. The instructions upon execution by the processor, cause the processor to implement any of the disclosed methods.

[0025] The above and other aspects and features of the disclosed technology are described in greater detail in the drawings, the description and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] FIG. 1 shows an illustration of a Quadtree plus binary tree (QTBT) block structure.

[0027] FIG. 2 shows an example of constructing a merge candidate list.

[0028] FIG. 3 shows an example of positions of spatial candidates.

[0029] FIG. 4 shows an example of candidate pairs subject to a redundancy check of spatial merge candidates.

[0030] FIGS. 5A and 5B show examples of the position of a second prediction unit (PU) based on the size and shape of the current block.

[0031] FIG. 6 shows an example of motion vector scaling for temporal merge candidates.

[0032] FIG. 7 shows an example of candidate positions for temporal merge candidates.

[0033] FIG. 8 shows an example of creating a combined bi-predictive merge candidate.

[0034] FIG. 9 shows an example of constructing motion vector prediction candidates.

[0035] FIG. 10 shows an example of motion vector scaling for spatial motion vector candidates.

[0036] FIGS. 11A and 11B are block diagrams of examples of a hardware platform for implementing a visual media decoding or a visual media encoding technique described in the present document.

[0037] FIGS. 12A and 12B show flowcharts of example methods for video processing based on some implementations of the disclosed technology.

[0038] FIGS. 13A to 13C show flowcharts of example methods for video processing based on some implementations of the disclosed technology.

[0039] FIGS. 14A and 14B show flowcharts of example methods for video processing based on some implementations of the disclosed technology.

[0040] FIGS. 15A to 15D show flowcharts of example methods for video processing based on some implementations of the disclosed technology.

[0041] FIGS. 16A to 16D show flowcharts of example methods for video processing based on some implementations of the disclosed technology.

DETAILED DESCRIPTION

1. Video coding in HEVC/H.265

[0042] Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM). In April 2018, the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC.

2.1. Quadtree plus binary tree (QTBT) block structure with larger CTUs

[0043] 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.

[0044] FIG. 1 shows an illustration of a Quadtree plus binary tree (QTBT) block structure. 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.

[0045] 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

[0046] 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.

[0047] FIG. 1 (left) 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.

[0048] 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.

[0049] 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.

2.2. Inter prediction in HEVC/H.265

[0050] 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.

[0051] 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.

[0052] 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.

[0053] 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.

[0054] The following text provides the details on the inter prediction modes specified in HEVC. The description will start with the merge mode.

2.2.1. Merge Mode

2.2.1.1. Derivation of candidates for merge mode

[0055] 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

[0056] These steps are also schematically depicted in Figure 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.

[0057] In the following, the operations associated with the aforementioned steps are detailed.

2.2.1.2. Spatial candidates derivation

[0058] In the derivation of spatial merge candidates, a maximum of four merge candidates are selected among candidates located in the positions depicted in Figure 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 Figure 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, Figure 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.

2.2.1.3. Temporal candidates derivation

[0059] In this step, only one candidate is added to the list. Particularly, in the derivation of this temporal merge candidate, a scaled motion vector is derived based on co-located PU belonging to the picture which has the smallest POC difference with current picture within the given reference picture list. The reference picture list to be used for derivation of the co-located PU is explicitly signalled in the slice header. The scaled motion vector for temporal merge candidate is obtained as illustrated by the dotted line in Figure 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.

[0060] 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 Figure 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.

2.2.1.4. Additional candidates insertion

[0061] 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, Figure 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.

[0062] 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.

2.2.1.5. Motion estimation regions for parallel processing

[0063] 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.

2.2.2. AMVP

[0064] 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 Figure 9). In the following sections, details about derivation process of motion vector prediction candidate are provided.

2.2.2.1. Derivation of AMVP candidates

[0065] Figure 9 summarizes derivation process for motion vector prediction candidate.

[0066] In motion vector prediction, two types of motion vector candidates are considered: spatial motion vector candidate and temporal motion vector candidate. For spatial motion vector candidate derivation, two motion vector candidates are eventually derived based on motion vectors of each PU located in five different positions as depicted in Figure 3.

[0067] For temporal motion vector candidate derivation, one motion vector candidate is selected from two candidates, which are derived based on two different co-located positions. After the first list of spatio-temporal candidates is made, duplicated motion vector candidates in the list are removed. If the number of potential candidates is larger than two, motion vector candidates whose reference picture index within the associated reference picture list is larger than 1 are removed from the list. If the number of spatio-temporal motion vector candidates is smaller than two, additional zero motion vector candidates is added to the list.

2.2.2.2. Spatial motion vector candidates

[0068] 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 Figure 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 can be four cases used as motion vector candidates, with two cases not associated with 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)

[0069] 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.

[0070] 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 Figure 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.

2.2.2.3. Temporal motion vector candidates

[0071] 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 Figure 7). The reference picture index is signalled to the decoder.

2.3. Adaptive motion vector difference resolution (AMVR)

[0072] In VVC, for regular inter mode, MVD can be coded in units of quarter luma samples, integer luma samples or four luma samples. The MVD resolution is controlled at the coding unit (CU) level, and MVD resolution flags are conditionally signalled for each CU that has at least one non-zero MVD components.

[0073] For a CU that has at least one non-zero MVD components, a first flag is signalled to indicate whether quarter luma sample MV precision is used in the CU. When the first flag (equal to 1) indicates that quarter luma sample MV precision is not used, another flag is signalled to indicate whether integer luma sample MV precision or four luma sample MV precision is used.

[0074] When the first MVD resolution flag of a CU is zero, or not coded for a CU (meaning all MVDs in the CU are zero), the quarter luma sample MV resolution is used for the CU. When a CU uses integer-luma sample MV precision or four-luma-sample MV precision, the MVPs in the AMVP candidate list for the CU are rounded to the corresponding precision.

2.4. Interpolation filters in VVC

[0075] For the luma interpolation filtering, 8-tap separable interpolation filters are used for 1/16-pel precision samples, as shown in Table 1.

Table 1: 8-tap coefficients f_L for 1/16-pel luma interpolation.

Fractional sample position p	interpolation filter coefficients							
	$f_L[p][0]$	$f_L[p][1]$	$f_L[p][2]$	$f_L[p][3]$	$f_L[p][4]$	$f_L[p][5]$	$f_L[p][6]$	$f_L[p][7]$
1	0	1	−3	63	4	−2	1	0
2	−1	2	−5	62	8	−3	1	0
3	−1	3	−8	60	13	−4	1	0
4	−1	4	−10	58	17	−5	1	0
5	−1	4	−11	52	26	−8	3	−1
6	−1	3	−9	47	31	−10	4	−1
7	−1	4	−11	45	34	−10	4	−1
8	−1	4	−11	40	40	−11	4	−1
9	−1	4	−10	34	45	−11	4	−1
10	−1	4	−10	31	47	−9	3	−1
11	−1	3	−8	26	52	−11	4	−1
12	0	1	−5	17	58	−10	4	−1
13	0	1	−4	13	60	−8	3	−1
14	0	1	−3	8	62	−5	2	−1
15	0	1	−2	4	63	−3	1	0

[0076] Similarly, 4-tap separable interpolation filters are used for the 1/32-pel precisions chroma interpolation, as shown in Table 2.

Table 2: 4-tap interpolation coefficients f_C for 1/32-pel chroma interpolation.

Fractional sample position p	interpolation filter coefficients			
	$f_C[p][0]$	$f_C[p][1]$	$f_C[p][2]$	$f_C[p][3]$
1	−1	63	2	0
2	−2	62	4	0

3	-2	60	7	−1
4	−2	58	10	−2
5	−3	57	12	−2
6	−4	56	14	−2
7	−4	55	15	−2
8	−4	54	16	−2
9	−5	53	18	−2
10	−6	52	20	−2
11	−6	49	24	−3
12	−6	46	28	−4
13	−5	44	29	−4
14	−4	42	30	−4
15	−4	39	33	−4
16	−4	36	36	−4
17	−4	33	39	−4
18	−4	30	42	−4
19	−4	29	44	−5
20	−4	28	46	−6
21	−3	24	49	−6
22	−2	20	52	−6
23	−2	18	53	−5
24	−2	16	54	−4
25	−2	15	55	−4
26	−2	14	56	−4
27	−2	12	57	−3
28	−2	10	58	−2
29	−1	7	60	−2
30	0	4	62	−2
31	0	2	63	−1

[0077] 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 1/16-pel chroma interpolation.

2.5. Alternative luma half-pel interpolation filters

[0078] In JVET-N0309, alternative half-pel interpolation filters are proposed.

[0079] The switching of the half-pel luma interpolation filter is done depending on the motion vector accuracy. In addition to the existing quarter-pel, full-pel, and 4-pel AMVR modes, a new half-pel accuracy AMVR mode is introduced. Only in case of half-pel motion vector accuracy, an alternative half-pel luma interpolation filter can be selected.

2.5.1. Half-pel AMVR mode

[0080] An additional AMVR mode for non-affine non-merge inter-coded CUs is proposed which allows signaling of motion vector differences at half-pel accuracy. The existing AMVR scheme of the current VVC draft is extended straightforward in the following way: Directly following the syntax element amvr_flag, if amvr_flag == 1, there is a new context-modeled binary syntax element hpel_amvr_flag which indicates usage of the new half-pel AMVR mode if hpel_amvr_flag == 1. Otherwise, i.e. if hpel_amvr_flag == 0, the selection between full-pel and 4-pel AMVR mode is indicated by the syntax element amvr_precision_flag as in the current VVC draft.

2.5.2. Alternative luma half-pel interpolation filters

[0081] For a non-affine non-merge inter-coded CU which uses half-pel motion vector accuracy (i.e., the half-pel AMVR mode), a switching between the HEVC/VVC half-pel luma interpolation filter and one or more alternative half-pel interpolation is made based on the value of a new syntax element if_idx. The syntax element if_idx is only signaled in case of half-pel AMVR mode. In case of skip/merge mode using a spatial merging candidate, the value of the syntax element if_idx is inherited from the neighbouring block.

2.5.2.1. Test 1: One alternative half-pel interpolation filter

[0082] In this test case, there is one 6-tap interpolation filter as an alternative to the ordinary HEVC/VVC half-pel interpolation filter. The following table shows the mapping between the value of the syntax element if_idx and the selected half-pel luma interpolation filter:

if_idx	Binarization	Filter	Interpolation filter coefficients

0	0	Gauss (6-tap)	[0, 3, 9, 20, 20, 9, 3, 0]
1	1	HEVC/VVC (8-tap)	[-1, 4, -11, 40, 40, -11, 4, -1]

2.5.2.2. Test 2: Two alternative half-pel interpolation filters

[0083] In this test case, there are two 8-tap interpolation filters as an alternative to the ordinary HEVC/VVC half-pel interpolation filter. The following table shows the mapping between the value of the syntax element if_idx and the selected half-pel luma interpolation filter:

if_idx	Binarization	Filter	Interpolation filter coefficients
0	0	Filter 1 (8-tap)	[3, 6, 10, 13, 13, 10, 6, 3]
1	10	Filter 2 (8-tap)	[-1, -1, 9, 25, 25, 9, -1, -1]
2	11	HEVC/VVC (8-tap)	[-1, 4, -11, 40, 40, -11, 4, -1]

[0084] amvr_precision_idx is signaled to indicate whether the current CU employs 1/2-pel, 1-pel or 4-pel MV preicision. There are 2 bins to be coded.

[0085] hpel_if_idx is signaled to indicate whether the default half-pel interpolation filter or alternative half-pel interpolation filters are used. When 2 alternative half-pel interpolation filters are used, there are 2 bins to be coded.

2.6. Generalized Bi-prediction

[0086] In conventional bi-prediction, the predictors from L0 and L1 are averaged to generate the final predictor using the equal weight 0.5. The predictor generation formula is shown as in Equ. (3)

$$P_{TraditionalBiPred} = (P_{L0} + P_{L1} + RoundingOffset) >> shiftNum, \quad (1)$$

In Equ. (3), $P_{TraditionalBiPred}$ is the final predictor for the conventional bi-prediction, P_{L0} and P_{L1} are predictors from L0 and L1, respectively, and *RoundingOffset and shiftNum* are used to normalize the final predictor.

[0087] Generalized Bi-prediction (GBI) is proposed to allow applying different weights to predictors from L0 and L1. The predictor generation is shown in Equ. (4).

$$P_{GBi} = ((1-w_1)* P_{L0} + w_1 * P_{L1} + RoundingOffset_{GBi}) >> shiftNum_{GBi}, \quad (2)$$

In Equ. (4), P_{GBi} is the final predictor of GBi. $(1-w_1)$ and w_1 are the selected GBI weights applied to the predictors of L0 and L1, respectively. $RoundingOffset_{GBi}$ and $shiftNum_{GBi}$ are used to normalize the final predictor in GBi.

[0088] The supported weights of w_1 is {-1/4, 3/8, 1/2, 5/8, 5/4}. One equal-weight set and four unequal-weight sets are supported. For the equal-weight case, the process to generate the final predictor is exactly the same as that in the conventional bi-prediction mode. For the true bi-prediction cases in random access (RA) condition, the number of candidate weight sets is reduced to three.

[0089] For advanced motion vector prediction (AMVP) mode, the weight selection in GBI is explicitly signaled at CU-level if this CU is coded by bi-prediction. For merge mode, the weight selection is inherited from the merge candidate.

3. Problems in Conventional Implementations

[0090] Alternative half-pel interpolation filters may be inherited in merge with motion vector difference (MMVD) mode even though the MV derived in MMVD mode are not of 1/2-pel precision, which is not reasonable.

[0091] When coding **amvr_precision_idx** and **hpel_if_idx**, all the bins are context coded.

4. Example embodiments and techniques

[0092] The embodiments detailed below should be considered as examples to explain general concepts. These embodiments should not be interpreted in a narrow way. Furthermore, these embodiments can be combined in any manner.

[0093] Decoder-side motion vector derivation (DMVD) is used to represent BDOF (bi-direction optical flow) or/and DMVR (decoder-side motion vector refinement) or/and other tools associated with refinement motion vector or prediction sample at decoder.

[0094] In the following text, the default interpolation filter may indicate the one defined in HEVC/VVC. Newly introduced interpolation filters (such as those proposed in JVET-N0309) may be also referred as alternative interpolation filters in the following description.

1. Alternative 1/N-pel interpolation filters may be used for different N with N unequal to 2.

a. In one example, N may be equal to 4, 16 etc.
b. In one example, the index of 1/N-pel interpolation filters may be signaled in AMVR mode.
    i. Alternatively, furthermore, the index of 1/N-pel interpolation filters may be signaled only when 1/N-pel MV/MVD precision is selected by the block.
c. In one example, alternative 1/N-pel interpolation filters may be not inherited in merge mode or/and MMVD mode.
    i. Alternatively, furthermore, only default 1/N-pel interpolation filter may be used in merge mode or/and MMVD mode.
d. In one example, alternative 1/N-pel interpolation filters may be inherited in merge mode.
e. In one example, alternative 1/N-pel interpolation filters may be inherited in MMVD mode.
    i. In one example, alternative 1/N-pel interpolation filters may be inherited in MMVD mode when the final derived MVs are of 1/N-pel precision, i.e., no MV component is of finer MV precisions.
    ii. In one example, alternative 1/N-pel interpolation filters may be inherited in MMVD mode when K (K >=1) MV components of the final derived MVs are of 1/N-pel precision.
f. In one example, alternative 1/N-pel interpolation filters may be inherited in MMVD mode or/and merge mode, however, alternative 1/N-pel interpolation filters are only used in motion compensation. The index of alternative 1/N-pel interpolation filters may not be stored for the block and may be not used by following coded blocks.

2. Indications of interpolation filters (such as default half-pel interpolation filter, alternative half-pel interpolation filters) may be stored together with other motion information, such as motion vectors, reference indices.
    a. In one example, for one block to be encoded/decoded, when it accesses a 2nd block which is located in a different region (such as in different CTU rows, in different VPDU), the interpolation filters associated with the 2nd block is disallowed to be utilized for encoding/decoding the current block.
3. Alternative half-pel interpolation filters may be not inherited in merge mode or/and MMVD mode.
    a. In one example, alternative half-pel interpolation filters may be not inherited in MMVD mode.
        i. Alternatively, furthermore, default half-pel interpolation filter in VVC may be always used for MMVD mode.

ii. Alternatively, the alternative half-pel interpolation filters may be inherited in MMVD mode. That is, for MMVD mode, the alternative half-pel interpolation filter associated with the base merge candidate may be inherited.

b. In one example, alternative half-pel interpolation filters may be inherited in MMVD mode under certain conditions.

i. In one example, alternative half-pel interpolation filters may be inherited when the final derived MVs are of 1/2-pel precision, i.e., no MV component is of finer MV precision like 1/4-pel precision, 1/16-pel precision.

ii. In one example, alternative half-pel interpolation filters may be inherited in MMVD mode when K (K >=1) MV components of the final derived MVs are of 1/2-pel precision.

c. In one example, alternative half-pel interpolation filters may be inherited in MMVD mode or/and merge mode, however, alternative half-pel interpolation filters are only used in motion compensation. The index of default half-pel interpolation filters, instead of alternative half-pel interpolation filters, may be stored for the block and may be used by following coded blocks.

d. The above methods may be applicable to the other cases wherein multiple interpolation filters for the 1/N-pel precision may be applied.

4. Interpolation filter information may be stored in history-based motion vector prediction (HMVP) table and may be inherited by HMVP merge candidate. In some implementations, a HMVP table is updated, selectively after the conversion between a video block of a video region and a coded representation. Whether to update the table, i.e., selectivity in updating the table, is based on determining, after the conversion of the video block, whether there is a motion candidate used in the conversion that is suitable for updating the HMVP table (e.g., by applying pruning, etc.).

a. In one example, when inserting a new candidate into the HMVP lookup table, the interpolation filter information may be considered. For example, two candidates with same motion information but different interpolation filter information may be considered as two different candidates.

b. In one example, when inserting a new candidate into the HMVP lookup table, two candidates with same motion information but different interpolation filter information may be considered as same candidates.

5. When inserting merge candidates into the merge candidate list, interpolation filter information may be considered in the pruning process.

a. In one example, two merge candidates with different interpolation filters may be considered as two different merge candidates.
b. In one example, when inserting the HMVP merge candidates into the merge list, interpolation filter information may be considered in the pruning process.
c. In one example, when inserting the HMVP merge candidates into the merge list, interpolation filter information may be not considered in the pruning process.

6. When generating pair-wise merge candidates or/and combined merge candidates or/and zero motion vector candidates or/and other default candidates, interpolation filter information may be taken into consideration instead of always using the default interpolation filter.
    a. In one example, if both candidates (involved in generating pair-wise merge candidates or/and combined merge candidates) employ same alternative interpolation filter, such interpolation filter may be inherited in the pair-wise merge candidates or/and combined merge candidates.
    b. In one example, if one of two candidates (involved in generating pair-wise merge candidates or/and combined merge candidates) does not employ default interpolation filter, its interpolation filter may be inherited in the pair-wise merge candidates or/and combined merge candidates.
    c. In one example, if one of the two candidates (involved in generating combined merge candidates) does not employ default interpolation filter, its interpolation filter may be inherited in the combined merge candidates. However, such interpolation filter may be only used for the corresponding prediction direction.
    d. In one example, if the two candidates (involved in generating combined merge candidates) employ different interpolation filters, their interpolation filters may be both inherited in the combined merge candidates. In this case, different interpolation filter may be used for different prediction directions.
    e. In one example, no more than K (K>=0) pair-wise merge candidates or/and combined merge candidates may use alternative interpolation filters.
    f. In one example, default interpolation filters are always used for pair-wise merge candidates or/and combined merge candidates.
7. It is proposed to disable the usage of half-pel motion vector/motion vector difference precision when the current block is coded with IBC mode.
    a. Alternatively, furthermore, there is no need to signal the indication of usage of half-pel MV/MVD precision.

b. In one example, alternative half-pel interpolation filter is always disabled if the current block is coded with IBC mode.
c. Alternatively, furthermore, there is no need to signal the indication of half-pel interpolation filter.
d. In one example, the condition of 'the current block is coded with IBC mode' may be replaced by 'the current block is coded with a mode'. Such a mode may be defined as triangular mode, merge mode, etc. al. One example mode corresponds to a coding mode in which a block is divided into two parts, each part is associated with one motion candidate and two motion candidates are derived from a same motion candidate list.

8. When encoding **amvr_precision_idx** and/or **hpel_if_idx**, only the first bin may be context-coded.
   a. Alternatively, furthermore, other bins may be bypass-coded.
   b. In one example, the first bin of **amvr_precision_idx** may be bypass-coded.
   c. In one example, the first bin of **hpel_if_idx** may be bypass-coded.
   d. In one example, only 1 context may be used to code the first bin of **amvr_precision_idx**.
   e. In one example, only 1 context may be used to code the first bin of **hpel_if_idx**.
   f. In one example, all bins of **amvr_precision_idx** may share one same context.
   g. In one example, all bins of **hpel_if_idx** may share one same context.
9. Some coding tools may be disallowed when alternative interpolation filters are employed.
   a. In one example, bi-directional optical flow (BDOF) may be disallowed when alternative interpolation filters are employed.
   b. In one example, DMVR or/and DMVD may be disallowed when alternative interpolation filters are employed.
   c. In one example, CIIP (combined inter-intra prediction) may be disallowed when alternative interpolation filters are employed.
      i. In one example, CIIP flag may be skipped and inferred to be false when a merge candidate inherits alternative interpolation filters.
      ii. Alternatively, when CIIP flag is true, default interpolation filter may be always employed.
   d. In one example, SMVD (symmetric motion vector difference) may be disallowed when alternative interpolation filters are employed.
      i. In one example, when SMVD is employed, default interpolation filters are always used, and syntax elements related to alternatively interpolation filters are not signaled.

ii. Alternatively, when syntax elements related to alternatively interpolation filters indicate that alternative interpolation filters are employed, SMVD related syntax elements may be not signaled and SMVD mode is not used.

e. In one example, SBT (subblock transform) may be disallowed when alternative interpolation filters are employed.

i. In one example, when SBT is employed, default interpolation filters are always used, and syntax elements related to alternatively interpolation filters are not signaled.

ii. Alternatively, when syntax elements related to alternatively interpolation filters indicate that alternative interpolation filters are employed, SBT related syntax elements may be not signaled and SBT is not used.

f. In one example, triangle prediction and other coding modes, in which a block is divided into two parts, may be disallowed when alternative interpolation filters are employed. In some implementations, when the block is divided into two parts, each part is associated with one motion candidate and two motion candidates are derived from a same motion candidate list.

i. In one example, interpolation filter information may be not inherited in triangle prediction and only default interpolation filters may be used.

g. Alternatively, triangle prediction and other coding modes, in which a block is divided into two parts, may be allowed when alternative interpolation filters are employed. In some implementations, when the block is divided into two parts, each part is associated with one motion candidate and two motion candidates are derived from a same motion candidate list.

i. In one example, interpolation filter information may be inherited in triangle prediction.

h. Alternatively, for a coding tool which is mentioned above, if it is enabled, then the alternative half-pel interpolation filter may be disabled.

10. A filter may be applied to N-pel precision MVs.

a. In one example, N may be equal to 1, 2 or 4 etc.

b. In one example, the filter may be a low-pass filter.

c. In one example, the filter may be a 1-d filter.

i. For example, the filter may be a 1-d horizonal filter.

ii. For example, the filter may be a 1-d vertical filter.

d. In one example, a flag may be signaled to indicate whether such filter is employed or not.

i. Alternatively, furthermore, such flag may be signaled only when the N-pel MVD precision (signaled in AMVR mode) is used for the block.

11. Different weighing factor sets may be used for regular inter mode and affine mode in GBI mode.

a. In one example, weighting factor sets used for regular inter mode and affine mode may be signaled in SPS/tile group header/slice header/VPS/PPS etc.
b. In one example, weighting factor sets used for regular inter mode and affine mode may be predefined at encoder and decoder.

12. How to define/select alternative interpolation filters may depend on the coded mode information.
    a. In one example, for the affine mode, and non-affine mode, the allowed sets of alternative interpolation filters may be different.
    b. In one example, for the IBC mode, and non-IBC mode, the allowed sets of alternative interpolation filters may be different.

5. Example implementations of the disclosed technology

[0095] FIG. 11A is a block diagram of a video processing apparatus 1100. The apparatus 1100 may be used to implement one or more of the methods described herein. The apparatus 1100 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1100 may include one or more processors 1102, one or more memories 1104 and video processing hardware 1106. The processor(s) 1102 may be configured to implement one or more methods described in the present document. The memory (memories) 1104 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1106 may be used to implement, in hardware circuitry, some techniques described in the present document, and may be partly or completely be a part of the processors 1102 (e.g., graphics processor core GPU or other signal processing circuitry).

[0096] FIG. 11B is another example of a block diagram of a video processing system in which disclosed techniques may be implemented. FIG. 11B is a block diagram showing an example video processing system 4100 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 4100. The system 4100 may include input 4102 for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input 4102 may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as Wi-Fi or cellular interfaces.

[0097] The system 4100 may include a coding component 4104 that may implement the various coding or encoding methods described in the present document. The coding component 4104 may reduce the average bitrate of video from the input 4102 to the output of the coding component 4104 to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component 4104 may be either stored, or transmitted via a communication connected, as represented by the component 4106. The stored or communicated bitstream (or coded) representation of the video received at the input 4102 may be used by the component 4108 for generating pixel values or displayable video that is sent to a display interface 4110. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as "coding" operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

[0098] Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on. Examples of storage interfaces include SATA (serial advanced technology attachment), PCI, IDE interface, and the like. The techniques described in the present document may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

[0099] Some embodiments of the disclosed technology include making a decision or determination to enable a video processing tool or mode. In an example, when the video processing tool or mode is enabled, the encoder will use or implement the tool or mode in the processing of a block of video, but may not necessarily modify the resulting bitstream based on the usage of the tool or mode. That is, a conversion from the block of video to the bitstream representation of the video will use the video processing tool or mode when it is enabled based on the decision or determination. In another example, when the video processing tool or mode is enabled, the decoder will process the bitstream with the knowledge that the bitstream has been modified based on the video processing tool or mode. That is, a conversion from the bitstream representation of the video to the block of video will be performed using the video processing tool or mode that was enabled based on the decision or determination.

[0100] Some embodiments of the disclosed technology include making a decision or determination to disable a video processing tool or mode. In an example, when the video processing tool or mode is disabled, the encoder will not use the tool or mode in the conversion of the block of video to the bitstream representation of the video. In another example, when the video processing tool or mode is disabled, the decoder will process the bitstream with the knowledge that the bitstream has not been modified using the video processing tool or mode that was disabled based on the decision or determination.

[0101] In the present document, the term "video processing" may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream.

[0102] It will be appreciated that the disclosed methods and techniques will benefit video encoder and/or decoder embodiments incorporated within video processing devices such as smartphones, laptops, desktops, and similar devices by allowing the use of the techniques disclosed in the present document.

[0103] FIG. 12A is a flowchart for an example method 1210 of video processing. The method 1210 includes, at 1212, performing a conversion between a current video block of a video and a coded representation of the video. In some implementations, the coded representation includes a first parameter that indicates a motion information precision from a multiple precision set for the current video block that does not use a default motion information precision and/or a second parameter which identifies whether an alternative half-pel interpolation filter is used for the conversion. In some implementations, each of the first parameter and/or the second parameter is associated with one or more bins and a context-model based coding is only used for a first bin of the one or more bins in the coded representation. In some implementations, the current video block is represented in the coded representation using a merge mode or a merge with motion vector different (MMVD) mode, wherein an interpolation filter having a 1/N precision is used for interpolating a prediction block associated

with the current video block according to a rule, wherein N is a positive integer; wherein the rule defines a condition for inheriting interpolation information in case that the interpolation filter is an alternative interpolation filter that is different from a default interpolation filter.

[0104] FIG. 12B is a flowchart for an example method 1220 of video processing. The method 1220 includes, at 1222, determining, for a conversion between a current video block of a video and a coded representation of the video, motion information using an interpolation filter, the motion information having an M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers and N is unequal to 2. The method 1220 further includes, at 1224, performing the conversion based on the motion information.

[0105] FIG. 13A is a flowchart for an example method 1310 of video processing. The method 1310 includes, at 1312, generating, for a conversion between a current video block of a video and a coded representation of the video, a first motion candidate based on one or more motion candidates in a motion candidate list and one or more interpolation filters associated with the one or more motion candidates, wherein an interpolation filter is assigned to the first motion candidate according to a rule that depends on the one or more interpolation filters associated with the one or more motion candidates. The method 1310 further includes, at 1314, inserting the first motion candidate into the motion candidate list. The method 1310 further includes, at 1316, performing the conversion based on the motion candidate list

[0106] FIG. 13B is a flowchart for an example method 1320 of video processing. The method 1320 includes, at 1322, associating, with each of multiple blocks comprising a first block in a first video region and a second block in a second video region of a video, motion information that includes information about a respective interpolation filter used for interpolating motion vector difference information for each of the multiple blocks. The method 1320 further includes, at 1324, performing a conversion between the video and a coded representation of the video using the motion information.

[0107] FIG. 13C is a flowchart for an example method 1330 of video processing. The method 1330 includes, at 1332, processing a current video block during a conversion between a video comprising the current video block and a coded representation of the video. The method 1330 further includes, at 1334, comparing and selectively updating a merge candidate list with motion information of the current video block. In some implementations, the motion information includes interpolation filter

information and the interpolation filter information includes parameters of an interpolation filter used for interpolating motion vector difference values used for representing the current video block in the coded representation.

[0108] FIG. 14A is a flowchart for an example method 1410 of video processing. The method 1410 includes, at 1412, maintaining, prior to a conversion between a current video block of a video region and a coded representation of the video, at least one history-based motion vector prediction (HMVP) table, wherein the HMVP table includes one or more entries corresponding to motion information of one or more previously processed blocks. The method 1410 further includes, at 1414, performing the conversion using the at least one HMVP table. In some implementations, the motion information of each entry is configured to include interpolation filter information for the one or more previously processed blocks. In some implementations, the interpolation filter information indicates interpolation filters used for interpolating prediction blocks of the one or more previously processed blocks.

[0109] FIG. 14B is a flowchart for an example method 1420 of video processing. The method 1420 includes, at 1422, maintaining, prior to a conversion between a current video block of a video region and a coded representation of the video, at least one history-based motion vector prediction (HMVP) table that includes one or more entries corresponding to motion information of one or more previously processed blocks, and wherein the motion information of each entry is configured to include interpolation filter information for the one or more previously processed blocks. The method 1420 further includes, at 1424, constructing, for the conversion, a motion candidate list which includes an HMVP merge candidate, wherein the HMVP candidate is derived by inheriting one entry from the HMVP table including corresponding interpolation filter information associated with the one entry. The method 1420 further includes, at 1426, performing the conversion based on the motion candidate list.

[0110] FIG. 15A is a flowchart for an example method 1510 of video processing. The method 1510 includes, at 1512, deciding a coding mode used for representing a current video block of a video in a coded representation of the video. The method 1510 further includes, at 1514, coding the current video block into the coded representation according to the coding mode, wherein use of half-pel

accuracy for representing motion information is disabled for the current video block due to use of the coding mode.

[0111] FIG. 15B is a flowchart for an example method 1520 of video processing. The method 1520 includes, at 1522, parsing a coded representation of a video to determine that a current video block of the video is coded using a coding mode. The method 1520 further includes, at 1524, generating a decoded representation of the current video block from the coded representation according to the coding mode, wherein use of half-pel accuracy for representing motion information is disabled for the current video block due to use of the coding mode.

[0112] FIG. 15C is a flowchart for an example method 1530 of video processing. The method 1530 includes, at 1532, determining, for a current video block that is processed using a generalized Bi-prediction (GBi) mode, to use a first weighting factor and a second, different weighting factor for a first prediction block and a second prediction block, respectively, the first weighting factor and the second weighting factor selected from weighting factor sets. The method 1530 further includes, at 1534, performing a conversion between the current video block of a video and a coded representation of the video based on the determining. In some implementations, a first weighting factor set used for an inter mode is different from a second weighting factor set used for an affine inter mode.

[0113] FIG. 15D is a flowchart for an example method 1540 of video processing. The method 1540 includes, at 1542, determining, for a conversion between multiple video units of a video region of a video and a coded representation of the multiple video units, information related to an interpolation filter depending on a coding mode of a video unit, the interpolation filter used for interpolating motion vector difference values in the coded representation and having a M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers. The method 1540 further includes, at 1544, performing the conversion using the interpolation filter.

[0114] FIG. 16A is a flowchart for an example method 1610 of video processing. The method 1610 includes, at 1612, deciding a coding mode used for representing a current video block of a video in a coded representation of the video. The method 1610 further includes, at 1614, coding the current video block into the coded representation according to the coding mode. In some implementations, use of alternative half-pel accuracy filters in addition to a default half-pel accuracy filter for representing motion information is disabled for the current video block due to use of the coding mode.

[0115] FIG. 16B is a flowchart for an example method 1620 of video processing. The method 1620 includes, at 1622, parsing a coded representation of a video to determine that a current video block of the video is coded using a coding mode. The method 1620 further includes, at 1624, generating a decoded representation of the current video block from the coded representation according to the coding mode, In some implementations, use of alternative half-pel accuracy filters in addition to a default half-pel accuracy filter for representing motion information is disabled for the current video block due to use of the coding mode.

[0116] FIG. 16C is a flowchart for an example method 1630 of video processing. The method 1630 includes, at 1632, determining, due to use of an alternative interpolation filter being enabled for a video block of a video, that a coding mode is disallowed for coding the video block into a coded representation of the video. The method 1630 further includes, at 1634, generating the coded representation of the video block based on the determining, wherein the alternative interpolation filter is used for interpolating inter prediction blocks of the current video block.

[0117] FIG. 16D is a flowchart for an example method 1640 of video processing. The method 1640 includes, at 1642, determining, due to use of an alternative interpolation filter being enabled for a video block of a video, that use of a coding mode for representing the video block in a coded representation of the video is disallowed. The method 1640 further includes, at 1644, generating a decoded video block by parsing the coded representation based on the determining.

[0118] Various techniques and embodiments may be described using the following clause-based format.

[0119] The first set of clauses describe certain features and aspects of the disclosed techniques in the previous section.

[0120] 1. A method of video processing, comprising: determining, for a current video block, a single set of motion information based on applying at least one interpolation filter to a set of neighboring blocks, wherein the at least one interpolation filter is configurable to an integer pixel accuracy or a sub-pixel accuracy of the single set of motion information; and performing a conversion between the current video block and a bitstream representation of the current video block, wherein the conversion

includes a decoder motion vector refinement (DMVR) step for refining the single set of motion information signaled in the bitstream representation.

[0121] 2. The method of clause 1, wherein when the single set of motion information is associated with an adaptive motion vector resolution (AMVR) mode, an index of the at least one interpolation filter is signaled in the bitstream representation.

[0122] 3. The method of clause 1, wherein when the single set of motion information is associated with a merge mode, an index of the at least one interpolation filter is inherited from a previous video block.

[0123] 4. The method of clause 1, wherein when the single set of motion information is associated with a merge with motion vector difference (MMVD) mode, an index of the at least one interpolation filter is inherited from a previous video block.

[0124] 5. The method of clause 1, wherein the at least one interpolation filter corresponds to a default filter with the sub-pixel accuracy, and wherein the single set of motion information is associated with a merge with motion vector difference (MMVD) mode.

[0125] 6. The method of any of clauses 2-4, wherein coefficients of the at least one interpolation filter for the current video block are inherited from the previous video block.

[0126] 7. The method of any of clauses 1-6, wherein the sub-pixel accuracy of the single set of motion information equals 1/4 pixel or 1/16 pixel.

[0127] 8. The method of any of clauses 1-7, wherein one or more components of the single set of motion information have the sub-pixel accuracy.

[0128] 9. The method of clause 1, wherein the at least one interpolation filter is expressed using 6 coefficients or 8 coefficients.

[0129] 10. The method of clause 5, wherein an index of the at least one interpolation filter is associated exclusively for the current video block and not associated for subsequent video blocks.

[0130] 11. The method of clause 1, wherein information related to the at least one interpolation filter is stored along with the single set of motion information.

[0131] 12. The method of clause 11, wherein the information related to the at least one interpolation filter identifies the at least one interpolation filter as a default filter.

[0132] 13. The method of clause 1, wherein coefficients of the at least one interpolation filter for the current video block are prevented from utilization by an interpolation filter of another video block.

[0133] 14. The method of any of clause 1-13, wherein the at least one interpolation filter corresponds to a plurality of filters, each filter in the plurality of filters associated with the sub-pixel accuracy of the single set of motion information.

[0134] 15. The method of any of clauses 1-14, wherein a precision of each component of the single set of motion information is equal to or lower than the sub-pixel accuracy the of the single set of motion information.

[0135] 16. The method of any of clauses 1-15, wherein the at least one interpolation filter is stored in a history-based motion vector prediction (HMVP) lookup table.

[0136] 17. The method of clause 16, further comprising:

[0137] upon detecting that motion information of another video block is same as the single set of motion of the current video block, inserting the single set of motion information into the HMVP table without inserting the motion information of the another video block.

[0138] 18. The method of clause 16, further comprising: upon detecting that motion information of another video block is same as the single set of motion of the current video block, inserting, into the HMVP table, the single set of motion information of the current video block and the motion information of the another video block.

[0139] 19. The method of any of clauses 17-18, wherein the another video block is associated with an interpolation filter, and wherein the inserting is based, at least in part on the at least one interpolation filter of the current video block and/or the interpolation filter of the another video block.

[0140] 20. The method of any of clauses 17-19, wherein the current video block and the another video block correspond to pair-wise candidates or combined merge candidates.

[0141] 21. The method of any of clauses 17-20, wherein the at least one interpolation filter of the current video block and the interpolation filter of the another video block are same.

[0142] 22. The method of any of clauses 17-20, wherein the at least one interpolation filter of the current video block and the interpolation filter of the another video block are different.

[0143] 23. The method of clause 1, wherein the current video block is coded in an Intra Block Copy (IBC) mode, and wherein a usage of the sub-pixel accuracy in representation of the single set of motion information is disabled.

[0144] 24. The method of any of clauses 1-23, wherein a usage of the at least one interpolation filter is disabled.

[0145] 25. The method of clause 1, wherein the at least one interpolation filter is associated with a amvr_precision_idx flag and/or a hpel_if_idx flag.

[0146] 26. The method of clause 25, wherein the amvr_precision_idx flag and/or the hpel_if_idx flag are/is associated with bypass-coded bins or context-coded bins.

[0147] 27. The method of clause 26, wherein a first bin is a bypass-coded bin or a context-coded bin.

[0148] 28. The method of any of clauses 25-27, wherein all bins share a same context.

[0149] 29. The method of clause 1, wherein one or more video processing steps are disabled based on employing the at least one interpolation filter.

[0150] 30. The method of clause 29, wherein the one or more video processing steps include the decoder motion vector refinement (DMVR) step, a bi-directional optical flow (BDOF) step, a combined inter-intra prediction (CIIP) step, a symmetric motion vector difference (SMVD) step, a subblock transform (SBT) step, or a triangle prediction step.

[0151] 31. The method of clause 30, wherein the at least one interpolation filter corresponds to a default filter.

[0152] 32. The method of any of clauses 30-31, wherein disabling the one or more video processing steps include disabling, in the bitstream representation, an indication of the one or more video processing steps.

[0153] 33. The method of any of clauses 30-31, wherein disabling the one or more video processing steps include disabling inheritance of the at least one interpolation filter of the current video block to another video block.

[0154] 34. The method of clause 1, wherein the integer pixel accuracy of the single set of motion information corresponds to 1 pixel, 2 pixels, or 4 pixels.

[0155] 35. The method of clause 34, wherein the at least one interpolation filter is a low-pass filter.

[0156] 36. The method of clause 34, wherein the at least one interpolation filter is a one-dimensional filter.

[0157] 37. The method of clause 36, wherein the one-dimensional filter is a horizontal filter or a vertical filter.

[0158] 38. The method of clause 34, wherein a flag in the bitstream representation indicates whether the at least one interpolation filter is employed or not.

[0159] 39. The method of any of clauses 1-38, wherein the current video block is associated with an adaptive motion vector resolution (AMVR) mode.

[0160] 40. The method of any of clauses 1-39, wherein the at least one interpolation filter for an inter mode of the current video block is different from a generalized bi-prediction (GBI) mode of the current video block.

[0161] 41. The method of any of clauses 1-40, wherein the at least one interpolation filter is pre-determined.

[0162] 42. The method of any of clauses 1-41, wherein the bitstream representation includes a video parameter set (VPS), a picture parameter set (PPS), a picture header, a tile group header, or a slice header associated with the current video block.

[0163] 43. The method of any one or more of clauses 1 through 42, wherein the video processing is an encoder-side implementation.

[0164] 44. The method of any one or more of clauses 1 through 70, wherein the video processing is a decoder-side implementation.

[0165] 45. 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 of clauses 1 to 44.

[0166] 46. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one or more of clauses 1 to 45.

[0167] The second set of clauses describe certain features and aspects of the disclosed techniques listed in the previous section, including, for example, Example Implementations 1, 8, and 10.

[0168] 1. A video processing method, comprising: performing a conversion between a current video block of a video and a coded representation of the video, wherein the coded representation includes a first parameter that indicates a motion information precision from a multiple precision set for the current video block that does not use a default motion information precision and/or a second parameter which identifies whether an alternative half-pel interpolation filter is used for the conversion, and wherein each of the first parameter and/or the second parameter is associated with one or more bins and a context-model based coding is only used for a first bin of the one or more bins in the coded representation.

[0169] 2. The method of clause 1, wherein the motion information comprises at least one of: motion vector prediction, motion vector difference and motion vector.

[0170] 3. The method of clause 1, wherein the first parameter is amvr_precision_idx.

[0171] 4. The method of clause 1, wherein the default motion information precision is 1/4-luma sample.

[0172] 5. The method of clause 1, wherein the multiple precision set comprises at least one of: 1/16-luma sample, 1/2-luma sample, 1 luma sample and 4 luma sample.

[0173] 6. The method of clause 1, wherein the current block is coded in inter mode, affine inter mode or intra block copy mode.

[0174] 7. The method of clause 1, wherein one context is used for the first bin of the first parameter.

[0175] 8. The method of clause 1, wherein remaining bins are bypass-coded.

[0176] 9. The method of clause 1, wherein the first bin of the first parameter and/or the first bin of the second parameter is bypass-coded.

[0177] 10. The method of clause 1, wherein one context is used for the first bin of the second parameter.

[0178] 11. The method of clause 1, wherein one context is shared by all bins of the first parameter.

[0179] 12. The method of clause 1, wherein one context is shared by all bins of the second parameter.

[0180] 13. The method of clause 1, wherein the second parameter has a first value that indicate use of a default half-pel interpolation filter.

[0181] 14. The method of clause 1, wherein the second parameter has a second value that use of the alternative half-pel interpolation filter.

[0182] 15. The method of clause 13, wherein the default half-pel interpolation filter has a coefficient set that is [-1, 4, -11, 40, 40, -11, 4, -1].

[0183] 16. The method of clause 1, wherein the alternative half-pel interpolation filter has a coefficient set that is [0, 3, 9, 20, 20, 9, 3, 0].

[0184] 17. A video processing method, comprising: determining, for a conversion between a current video block of a video and a coded representation of the video, motion information using an interpolation filter, the motion information having an M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers and N is unequal to 2; and performing the conversion based on the motion information; wherein a syntax field in the coded representation corresponds to the interpolation filter.

[0185] 18. The method of clause 17, wherein N is equal to 4. Alternatively, in some embodiments N is equal to 16.

[0186] 19. The method of clause 17, wherein, due to the current video block coded with an adaptive motion vector difference resolution (AMVR) mode, the coded representation includes an index of the interpolation filter.

[0187] 20. The method of clause 17, wherein, due to the current video block coded with a merge mode and/or a merge with motion vector difference (MMVD) mode, an index of the interpolation filter is not inherited from a previous video block.

[0188] 21. The method of clause 17, wherein, due to the current video block being coded with a merge mode and/or a merge with motion vector difference (MMVD) mode, the syntax field indicates a default interpolation filter.

[0189] 22. The method of clause 17, wherein, for the current video block coded with a merge mode and/or a merge with motion vector difference (MMVD) mode, an index of the interpolation filter is inherited from a previous video block.

[0190] 23. The method of clause 22, wherein the motion information has the sub-pixel accuracy.

[0191] 24. The method of clause 22, wherein the motion information includes one or more components having the sub-pixel accuracy.

[0192] 25. The method of clause 22, wherein an index of the interpolation filter is associated exclusively for the current video block and not associated for subsequent video blocks.

[0193] 26. The method of clause 17, wherein M is equal to 1, 2 or 4.

[0194] 27. The method of clause 17, wherein the interpolation filter is a low-pass filter.

[0195] 28. The method of clause 17, wherein the interpolation filter is a one-dimensional filter.

[0196] 29. The method of clause 28, wherein the one-dimensional filter is a horizontal filter or a vertical filter.

[0197] 30. The method of clause 17, wherein the coded representation selectively includes a flag indicating whether the interpolation filter having M-integer pixel accuracy is enabled.

[0198] 31. The method of clause 30, wherein the flag is signaled if and only if the current video block is encoded using an integer pixel motion vector precision.

[0199] 32. The method of any of clauses 1 to 31, wherein the performing of the conversion includes generating the video from the coded representation.

[0200] 33. The method of any of clauses 1 to 31, wherein the performing of the conversion includes generating the coded representation from the video.

[0201] 34. 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 clauses 1 to 33.

[0202] 35. 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 clauses 1 to 34.

[0203] The third set of clauses describe certain features and aspects of the disclosed techniques listed in the previous section, including, for example, Example Implementations 2, 3, 5, and 6.

[0204] 1. A video processing method, comprising: generating, for a conversion between a current video block of a video and a coded representation of the video, a first motion candidate based on one or more motion candidates in a motion candidate list and one or more interpolation filters associated with the one or more motion candidates; wherein an interpolation filter is assigned to the first motion candidate according to a rule that depends on the one or more interpolation filters associated with the one or more motion candidates, inserting the first motion candidate into the motion candidate list; and performing the conversion based on the motion candidate list.

[0205] 2. The method of clause 1, wherein the first motion candidate corresponds to at least one of a pair-wise merge candidate, a combined merge candidate, a zero motion vector candidate, and/or other default candidates.

[0206] 3. The method of clause 1, wherein each motion candidate is associated with a half-sample interpolation filter index and a half-sample interpolation filter of the first motion candidate is determined by half-sample interpolation filter indices of at least two motion candidates.

[0207] 4. The method of clause 1, wherein, for two motion candidates being associated with a same half-sample interpolation filter, the rule specifies that the same half-sample interpolation filter is inherited in the first motion candidate.

[0208] 5. The method of clause 4, wherein the same half-sample interpolation filter corresponds to an alternative half-sample interpolation filter associated with a half-sample interpolation filter index being equal to 1, the alternative half-sample interpolation filter different from a default half-sample interpolation filter associated with a half-sample interpolation filter index being equal to 0.

[0209] 6. The method of clause 1, wherein, for one of the motion candidates not being associated with a default interpolation filter, the rule specifies that the interpolation filter is inherited in the first motion candidate.

[0210] 7. The method of clause 6, wherein the interpolation filter is only used for a corresponding prediction direction.

[0211] 8. The method of clause 1, wherein, for two motion candidates being associated with different interpolation filters, the rule specifies that the different interpolation filters are inherited in the first motion candidate.

[0212] 9. The method of clause 1, wherein the different interpolation filters are used for different prediction directions.

[0213] 10. The method of clause 1, wherein a number of the first motion candidate whose assigned interpolation filter corresponds to an alternative interpolation filter is not more than K, where K being equal to or greater than 0.

[0214] 11. The method of clause 1, wherein a default interpolation filter is always used for the first motion candidate.

[0215] 12. A video processing method, comprising: associating, with each of multiple blocks comprising a first block in a first video region and a second block in a second video region of a video, motion information that includes information about a respective interpolation filter used for interpolating motion vector difference information for each of the multiple blocks; and performing a conversion between the video and a coded representation of the video using the motion information.

[0216] 13. The method of clause 12, wherein the motion information comprises information indicative of use of a default motion interpolation filter.

[0217] 14. The method of clause 12, wherein the motion information identifies an alternative interpolation filter that is different from a default interpolation filter.

[0218] 15. The method of clause 12, wherein the motion information includes one or more motion vectors and one or more reference frame indices.

[0219] 16. The method of any of clauses 12 to 15, wherein the first video region and the second video region are coding tree unit rows.

[0220] 17. The method of any of clause 12 to 15, wherein the first video region and the second video region are virtual pipeline data units.

[0221] 18. The method of any of clauses 12 to 17, wherein the first video region and the second video region are different regions.

[0222] 19. The method of any of clauses 12 to 18, wherein the second block is coded or decoded by accessing the first block and wherein the motion information is according to a rule that specifies that an interpolation filter used for the second block is disallowed to be same as an interpolation filter used for the first block in case that the first video region is different from the second video region.

[0223] 20. The method of any of clauses 12 to 17, wherein the first video region and the second video region are same.

[0224] 21. A video processing method, comprising: performing a conversion between a current video block of a video and a coded representation of the video, wherein the current video block is represented in the coded representation using a merge mode or a merge with motion vector different (MMVD) mode, wherein an interpolation filter having a 1/N precision is used for interpolating a prediction block associated with the current video block according to a rule, wherein N is a positive integer; wherein the rule defines a condition for inheriting interpolation information in case that the interpolation filter is an alternative interpolation filter that is different from a default interpolation filter.

[0225] 22. The method of clause 21, wherein the rule determines to disallow the inheriting of coefficients of the alternative interpolation filter.

[0226] 23. The method of clause 21, wherein a syntax element in the coded representation always indicates the default interpolation filter.

[0227] 24. The method of clause 21, wherein the rule determines to allow the inheriting of coefficients of the alternative interpolation filter.

[0228] 25. The method of clause 24, wherein the coefficients of the alternative interpolation filter are inherited when final derived motion vectors are of 1/2-pel precision and no motion vector component has a finer motion vector precision.

[0229] 26. The method of clause 24, wherein the coefficients of the alternative interpolation filter are inherited when final derived motion vectors include one or more motion vector components that have 1/2-pel precision.

[0230] 27. The method of clause 24, wherein the alternative interpolation filter is only used for motion compensation.

[0231] 28. The method of clause 24, wherein an index of the default interpolation filter is stored for the current video block and used by coded blocks subsequent to the current video block.

[0232] 29. A video processing method, comprising: processing a current video block during a conversion between a video comprising the current video block and a coded representation of the video; and comparing and selectively updating a merge candidate list with motion information of the current video block, wherein the motion information includes interpolation filter information; wherein the interpolation filter information includes parameters of an interpolation filter used for interpolating motion vector difference values used for representing the current video block in the coded representation.

[0233] 30. The method of clause 29, wherein two merge candidates associated with different interpolation filters are considered as two different merge candidates.

[0234] 31. The method of clause 29, wherein comparing and selectively updating the merge list includes inserting a history-based motion vector prediction (HMVP) merge candidate to the merge candidate list by considering the interpolation filter information.

[0235] 32. The method of clause 29, wherein comparing and selectively updating the merge candidate list includes inserting a history-based motion vector prediction (HMVP) merge candidate to the merge candidate list by not considering the interpolation filter information.

[0236] 33. The method of any of clauses 1 to 32, wherein the default motion interpolation filter has a coefficient set that is [-1, 4, -11, 40, 40, -11, 4, -1].

[0237] 34. The method of any of clauses 1 to 32, wherein the alternative interpolation filter has a coefficient set that is [0, 3, 9, 20, 20, 9, 3, 0].

[0238] 35. The method of any of clauses 1 to 34, wherein the performing of the conversion includes generating the coded representation from the current video block.

[0239] 36. The method of any of clauses 1 to 34, wherein the performing of the conversion includes generating the current video block from the coded representation.

[0240] 37. 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 clauses 1 to 36.

[0241] 38. 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 clauses 1 to 36.

[0242] The fourth set of clauses describe certain features and aspects of the disclosed techniques listed in the previous section, including, for example, Example Implementation 4.

[0243] 1. A video processing method, comprising: maintaining, prior to a conversion between a current video block of a video region and a coded representation of the video, at least one history-based motion vector prediction (HMVP) table, wherein the HMVP table includes one or more entries corresponding to motion information of one or more previously processed blocks; and performing the conversion using the at least one HMVP table; and wherein the motion information of each entry is configured to include interpolation filter information for the one or more previously processed blocks, wherein the interpolation filter information indicates interpolation filters used for interpolating prediction blocks of the one or more previously processed blocks.

[0244] 2. A video processing method, comprising: maintaining, prior to a conversion between a current video block of a video region and a coded representation of the video, at least one history-based motion vector prediction (HMVP) table that includes one or more entries corresponding to motion information of one or more previously processed blocks, and wherein the motion information of each entry is configured to include interpolation filter information for the one or more previously processed blocks; constructing, for the conversion, a motion candidate list which includes an HMVP

merge candidate, wherein the HMVP candidate is derived by inheriting one entry from the HMVP table including corresponding interpolation filter information associated with the one entry; and performing the conversion based on the motion candidate list.

[0245] 3. The method of clause 1 or 2, the method further comprises updating, selectively after the conversion, the HMVP table.

[0246] 4. The method of clause 1 or 2, the method further comprises updating selectively the HMVP table using the motion information derived during the conversion for the current video block, wherein the motion information derived for the current video block includes interpolation filter information for the current video block.

[0247] 5. The method of clause 3 or 4, wherein the updating includes determining whether to insert a new candidate into the HMVP table without considering the interpolation filter information.

[0248] 6. The method of clause 4, wherein the new candidate and an entry in the HMVP table before being updated that have different interpolation filter information but same part or all of remaining motion information excluding the interpolation filter information are considered as a same candidate.

[0249] 7. The method of clause 3 or 4, wherein the updating includes determining whether to insert a new candidate into the HMVP table by considering the interpolation filter information.

[0250] 8. The method of clause 7, wherein the new candidate and an entry in the HMVP table before being updated that have same part or all of remaining motion information excluding the interpolation filter information are considered as two different candidates.

[0251] 9. The method of any of clauses 5 to 8, wherein the new candidate is derived based on the motion information of the current video block;

[0252] 10. The method of any of clauses 5 to 8, wherein the updating includes, for the new candidate that is different from an entry in the HMVP table, adding the new candidate to the HMVP table.

[0253] 11. The method of any of clauses 5 to 8, wherein the updating includes, for the new candidate that is same as an entry in the HMVP table, removing the entry from the HMVP table and adding the new candidate to the HMVP table.

[0254] 12. The method of any of clauses 1 to 11, wherein the interpolation filter information includes a half sample interpolation filter index that indicates use of an alternative interpolation filter or a default interpolation filter.

[0255] 13. The method of clause 12, wherein the half sample interpolation filter index is equal to 0 to indicate the use of the default interpolation filter.

[0256] 14. The method of clause 12, wherein the half sample interpolation filter index is equal to 1 to indicates the use of the alternative interpolation filter.

[0257] 15. The method of clause 12, wherein the default interpolation filter has a coefficient set that is [-1, 4, -11, 40, 40, -11, 4, -1].

[0258] 16. The method of clause 12, wherein the alternative interpolation filter has a coefficient set that is [0, 3, 9, 20, 20, 9, 3, 0].

[0259] 17. The method of any of clauses 1 to 16, wherein the performing of the conversion includes generating the coded representation from the current video block.

[0260] 18. The method of any of clauses 1 to 16, wherein the performing of the conversion includes generating the current video block from the coded representation.

[0261] 19. 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 clauses 1 to 18.

[0262] 20. 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 clauses 1 to 18.

[0263] The fifth set of clauses describe certain features and aspects of the disclosed techniques listed in the previous section, including, for example, Example Implementations 7 and 11-12.

[0264] 1. A video encoding method, comprising: deciding a coding mode used for representing a current video block of a video in a coded representation of the video; and coding the current video block into the coded representation according to the coding mode, wherein use of half-pel accuracy for representing motion information is disabled for the current video block due to use of the coding mode.

[0265] 2. A video decoding method, comprising: parsing a coded representation of a video to determine that a current video block of the video is coded using a coding mode; and generating a decoded representation of the current video block from the coded representation according to the coding mode, wherein use of half-pel accuracy for representing motion information is disabled for the current video block due to use of the coding mode.

[0266] 3. The method of clause 1 or 2, wherein the motion information corresponds to one or more motion vectors for the current video block.

[0267] 4. The method of clause 1 or 2, wherein the motion information corresponds to one or more motion vector difference values for the current video block.

[0268] 5. The method of clause 1 or 2, wherein the motion information corresponds to one or more motion vectors for the current video block and one or more motion vector difference values for the current video block.

[0269] 6. The method of any of clauses 1 to 4, wherein, due to use of the coding mode, the coded representation omits bits and/or bins for signaling whether half-pel resolution is applicable to the current video block.

[0270] 7. The method of any of clauses 1 to 6, wherein the coding mode corresponds to an intra block copy (IBC) mode that generates a prediction block using at least a block vector pointing to a video frame containing the current video block for coding the current video block.

[0271] 8. The method of clause 7, wherein an indication of a half-pel interpolation filter is not signaled.

[0272] 9. The method of clause 1, wherein the coding mode corresponds to a mode in which the current video block is divided into two parts, each part is associated with one motion candidate and two motion candidates are derived from a same motion candidate list.

[0273] 10. The method of clause 9, wherein the coding mode corresponds to a triangular prediction mode.

[0274] 11. The method of clause 1, wherein the coding mode corresponds to a merge mode in which the current video block and neighboring blocks share the motion information.

[0275] 12. A video processing method, comprising: determining, for a current video block that is processed using a generalized Bi-prediction (GBi) mode, to use a first weighting factor and a second, different weighting factor for a first prediction block and a second prediction block, respectively, the first weighting factor and the second weighting factor selected from weighting factor sets; and performing a conversion between the current video block of a video and a coded representation of the video using the final predictor, wherein a first weighting factor set used for an inter mode is different from a second weighting factor set used for an affine inter mode.

[0276] 13. The method of clause 12, wherein a sum of the first weighting factor and the second weighting factor equals to 1.

[0277] 14. The method of clause 12, wherein the weighing factor sets are signaled in a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a picture header, a tile group header, or a slice header associated with the current video block.

[0278] 15. The method of clause 12, wherein the weighing factor sets are predefined.

[0279] 16. A video processing method, comprising: determining, for a conversion between multiple video units of a video region of a video and a coded representation of the multiple video units, information related to an interpolation filter depending on a coding mode of a video unit, the interpolation filter used for interpolating motion vector difference values in the coded representation and having a M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers; and performing the conversion using the interpolation filter.

[0280] 17. The method of clause 16, wherein determining determines a first set of coefficients of the interpolation filter for the video unit coded with an affine mode and wherein determining determined a different, second set of coefficients of the interpolation for another video unit coded with a non-affine mode.

[0281] 18. The method of clause 16, wherein determining determines a first set of coefficients of the interpolation filter for the video unit coded with an IBC mode and wherein determining determined a different, second set of coefficients of the interpolation for another video unit coded with a non-IBC mode.

[0282] 19. The method of any of clauses 1 to 18, wherein the performing of the conversion includes generating the coded representation from the current video block.

[0283] 20. The method of any of clauses 1 to 18, wherein the performing of the conversion includes generating the current video block from the coded representation.

[0284] 21. 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 clauses 1 to 20.

[0285] 22. 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 clauses 1 to 20.

[0286] The sixth set of clauses describe certain features and aspects of the disclosed techniques listed in the previous section, including, for example, Example Implementation 9.

[0287] 1. A video encoding method, comprising: deciding a coding mode used for representing a current video block of a video in a coded representation of the video; and coding the current video block into the coded representation according to the coding mode, wherein use of alternative half-pel accuracy filters in addition to a default half-pel accuracy filter for representing motion information is disabled for the current video block due to use of the coding mode.

[0288] 2. A video decoding method, comprising: parsing a coded representation of a video to determine that a current video block of the video is coded using a coding mode; and generating a decoded representation of the current video block from the coded representation according to the coding mode, wherein use of alternative half-pel accuracy filters in addition to a default half-pel accuracy filter for representing motion information is disabled for the current video block due to use of the coding mode.

[0289] 3. The method of clause 1 or 2, wherein the coding mode corresponds to a triangle prediction mode in which the video block is partitioned into multiple partitions, at least one of which is a non-rectangular partition.

[0290] 4. The method of clause 3, wherein information related to the alternative half-pel accuracy filter is not inherited in the triangle prediction mode and only the default half-pel accuracy filter is used.

[0291] 5. The method of clause 1 or 2, wherein the coding mode corresponds to a bi-direction optical flow (BDOF) in which the current block is bi-directionally predicted.

[0292] 6. The method of clause 1 or 2, wherein the coding mode corresponds to a decoder-side motion vector refinement (DMVR) that refines prediction information from motion information derived at a decoder side.

[0293] 7. The method of clause 1 or 2, wherein the coding mode corresponds to a decoder-side motion vector derivation (DMVD) that derives motion information at a decoder side.

[0294] 8. The method of clause 1 or 2, wherein the coding mode corresponds to a combined inter-intra prediction (CIIP) in which inter and intra prediction is combined to generate a final prediction signal.

[0295] 9. The method of clause 1 or 2, wherein the coding mode corresponds to a SMVD (symmetric motion vector difference) that processes motion vector difference symmetrically.

[0296] 10. The method of clause 1 or 2, wherein the coding mode corresponds to a SBT (subblock transform) that transforms a sub-block into a transform block.

[0297] 11. A video encoding method, comprising: determining, due to use of an alternative interpolation filter being enabled for a video block of a video, that a coding mode is disallowed for coding the video block into a coded representation of the video; and generating the coded representation of the video block based on the determining, wherein the alternative interpolation filter is used for interpolating inter prediction blocks of the current video block.

[0298] 12. A video decoding method, comprising: determining, due to use of an alternative interpolation filter being enabled for a video block of a video, that use of a coding mode for

representing the video block in a coded representation of the video is disallowed; and generating a decoded video block by parsing the coded representation based on the determining.

[0299] 13. The method of clause 11 or 12, wherein the alternative interpolation filter is an alternative half-sample interpolation filter.

[0300] 14. The method of clause 11, wherein the determining determines to disallow the coding mode corresponding to a coding mode in which the current video block is divided into two parts, each part is associated with one motion candidate and two motion candidates are derived from a same motion candidate list.

[0301] 15. The method of clause 14, wherein the coding mode corresponds to a triangle prediction mode.

[0302] 16. The method of clause 13 or 14, wherein information related to the alternative interpolation filter is not inherited in the coding mode and only a default half-sample interpolation filter is used.

[0303] 17. The method of clause 11 or 12, wherein the determining determines to disallow the coding mode corresponding to a bi-direction optical flow (BDOF) in which the current block is bi-directionally predicted.

[0304] 18. The method of clause 17, wherein the determining determines to disallow the coding mode corresponding to a decoder-side motion vector refinement (DMVR) that refines prediction information from motion information derived at a decoder side.

[0305] 19. The method of clause 12, wherein the determining determines to disallow the coding mode corresponding to a decoder-side motion vector derivation (DMVD) that derives motion information at a decoder side.

[0306] 20. The method of clause 11, wherein the determining determines to disallow the coding mode corresponding to a combined inter-intra prediction (CIIP) in which inter and intra prediction is combined to generate a final prediction signal.

[0307] 21. The method of clause 20, wherein a CIIP flag is skipped and inferred to be false in case that a merge candidate inherits the interpolation filter.

[0308] 22. The method of clause 20, wherein a default interpolation filter is always used in case that a CIIP flag is true.

[0309] 23. The method of clause 11, wherein the determining determines to disallow the coding mode corresponding to a SMVD (symmetric motion vector difference) that processes motion vector difference symmetrically.

[0310] 24. The method of clause 23, wherein a default interpolation filter is always used for the usage of the SMVD and wherein a syntax element related to the interpolation filter is not signaled.

[0311] 25. The method of clause 23, wherein a syntax element related to the interpolation filter indicates that the interpolation filter is not used, a syntax element related to the SMVD is not signaled, and the SMVD is not used.

[0312] 26. The method of clause 12, wherein the determining determines to disallow the coding mode corresponding to a SBT (subblock transform) that transforms a sub-block into a transform block.

[0313] 27. The method of clause 26, wherein a default interpolation filter is always used for the usage of the SBT and a syntax element related to the interpolation filter is not signaled.

[0314] 28. The method of clause 26, wherein a syntax element related to the interpolation filter indicates that the interpolation filter is not used, a syntax element related to the SBT is not signaled, and the SBT is not used.

[0315] 29. The method of clause 11, wherein the generating the coded representation includes generating the coded representation by coding the video block using a triangle coding mode in which the video block is partitioned into multiple partitions, at least one of which is a non-rectangular partition.

[0316] 30. The method of clause 29, wherein information related to the interpolation filter is inherited in the triangle coding mode.

[0317] 31. The method of clause 11, wherein the generating the coded representation includes generating the coded representation by omitting syntax elements that signal use of the coding mode.

[0318] 32. The method of clause 12, wherein the parsing the coded representation includes parsing the coded representation in absence of syntax elements that signal use of the coding mode.

[0319] 33. 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 clauses 1 to 32.

[0320] 34. 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 clauses 1 to 32.

[0321] In some embodiments, the above-disclosed methods may be implemented by an apparatus that comprises a processor that is configurable to perform the method.

[0322] 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.

[0323] 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.

[0324] 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).

[0325] 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.

[0326] While this patent document contains many specifics, these should not be construed as limitations on the scope of any subject matter or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular techniques. Certain features that are described in this patent document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described

in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0327] 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.

[0328] 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 documents.

CLAIMS

What is claimed is:

1. A video processing method, comprising:

performing a conversion between a current video block of a video and a coded representation of the video,

wherein the coded representation includes a first parameter that indicates a motion information precision from a multiple precision set for the current video block that does not use a default motion information precision and/or a second parameter which identifies whether an alternative half-pel interpolation filter is used for the conversion, and

wherein each of the first parameter and/or the second parameter is associated with one or more bins and a context-model based coding is only used for a first bin of the one or more bins in the coded representation.

2. The method of claim 1, wherein the motion information comprises at least one of: motion vector prediction, motion vector difference and motion vector.

3. The method of claim 1, wherein the first parameter is amvr_precision_idx.

4. The method of claim 1, wherein the default motion information precision is 1/4-luma sample.

5. The method of claim 1, wherein the multiple precision set comprises at least one of: 1/16-luma sample, 1/2-luma sample, 1 luma sample and 4 luma sample.

6. The method of claim 1, wherein the current block is coded in inter mode, affine inter mode or intra block copy mode.

7. The method of claim 1, wherein one context is used for the first bin of the first parameter.

8. The method of claim 1, wherein remaining bins are bypass-coded.

9. The method of claim 1, wherein the first bin of the first parameter and/or the first bin of the second parameter is bypass-coded.

10. The method of claim 1, wherein one context is used for the first bin of the second parameter.

11. The method of claim 1, wherein one context is shared by all bins of the first parameter.

12. The method of claim 1, wherein one context is shared by all bins of the second parameter.

13. The method of claim 1, wherein the second parameter has a first value that indicate use of a default half-pel interpolation filter.

14. The method of claim 1, wherein the second parameter has a second value that use of the alternative half-pel interpolation filter.

15. The method of claim 13, wherein the default half-pel interpolation filter has a coefficient set that is [-1, 4, -11, 40, 40, -11, 4, -1].

16. The method of claim 1, wherein the alternative half-pel interpolation filter has a coefficient set that is [0, 3, 9, 20, 20, 9, 3, 0].

17. A video processing method, comprising:

determining, for a conversion between a current video block of a video and a coded representation of the video, motion information using an interpolation filter, the motion information having an M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers and N is unequal to 2; and

performing the conversion based on the motion information;

wherein a syntax field in the coded representation corresponds to the interpolation filter.

18. The method of claim 17, wherein N is equal to 4 or 16.

19. The method of claim 17, wherein, due to the current video block coded with an adaptive motion vector difference resolution (AMVR) mode, the coded representation includes an index of the interpolation filter.

20. The method of claim 17, wherein, due to the current video block coded with a merge mode and/or a merge with motion vector difference (MMVD) mode, an index of the interpolation filter is not inherited from a previous video block.

21. The method of claim 17, wherein, due to the current video block being coded with a merge mode and/or a merge with motion vector difference (MMVD) mode, the syntax field indicates a default interpolation filter.

22. The method of claim 17, wherein, for the current video block coded with a merge mode and/or a merge with motion vector difference (MMVD) mode, an index of the interpolation filter is inherited from a previous video block.

23. The method of claim 22, wherein the motion information has the sub-pixel accuracy.

24. The method of claim 22, wherein the motion information includes one or more components having the sub-pixel accuracy.

25. The method of claim 22, wherein an index of the interpolation filter is associated exclusively for the current video block and not associated for subsequent video blocks.

26. The method of claim 17, wherein M is equal to 1, 2 or 4.

27. The method of claim 17, wherein the interpolation filter is a low-pass filter.

28. The method of claim 17, wherein the interpolation filter is a one-dimensional filter.

29. The method of claim 28, wherein the one-dimensional filter is a horizontal filter or a vertical filter.

30. The method of claim 17, wherein the coded representation selectively includes a flag indicating whether the interpolation filter having M-integer pixel accuracy is enabled.

31. The method of claim 30, wherein the flag is signaled if and only if the current video block is encoded using an integer pixel motion vector precision.

32. The method of any of claims 1 to 31, wherein the performing of the conversion includes generating the video from the coded representation.

33. The method of any of claims 1 to 31, wherein the performing of the conversion includes generating the coded representation from the video.

34. 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 recited in one or more of claims 1 to 33.

35. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method recited in one or more of claims 1 to 34.

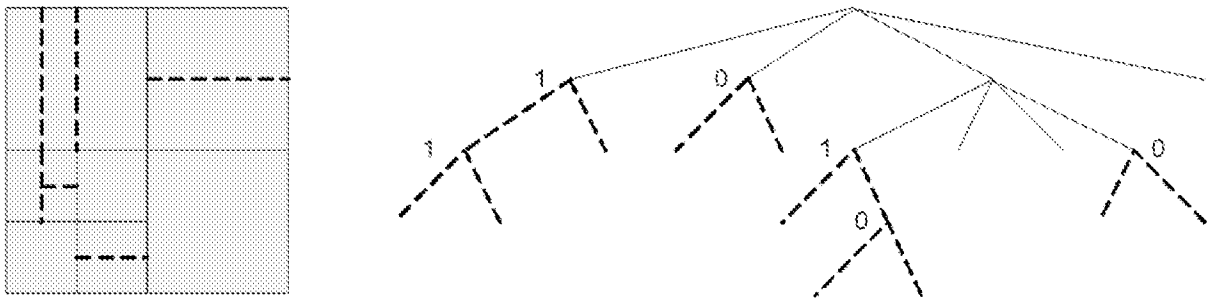
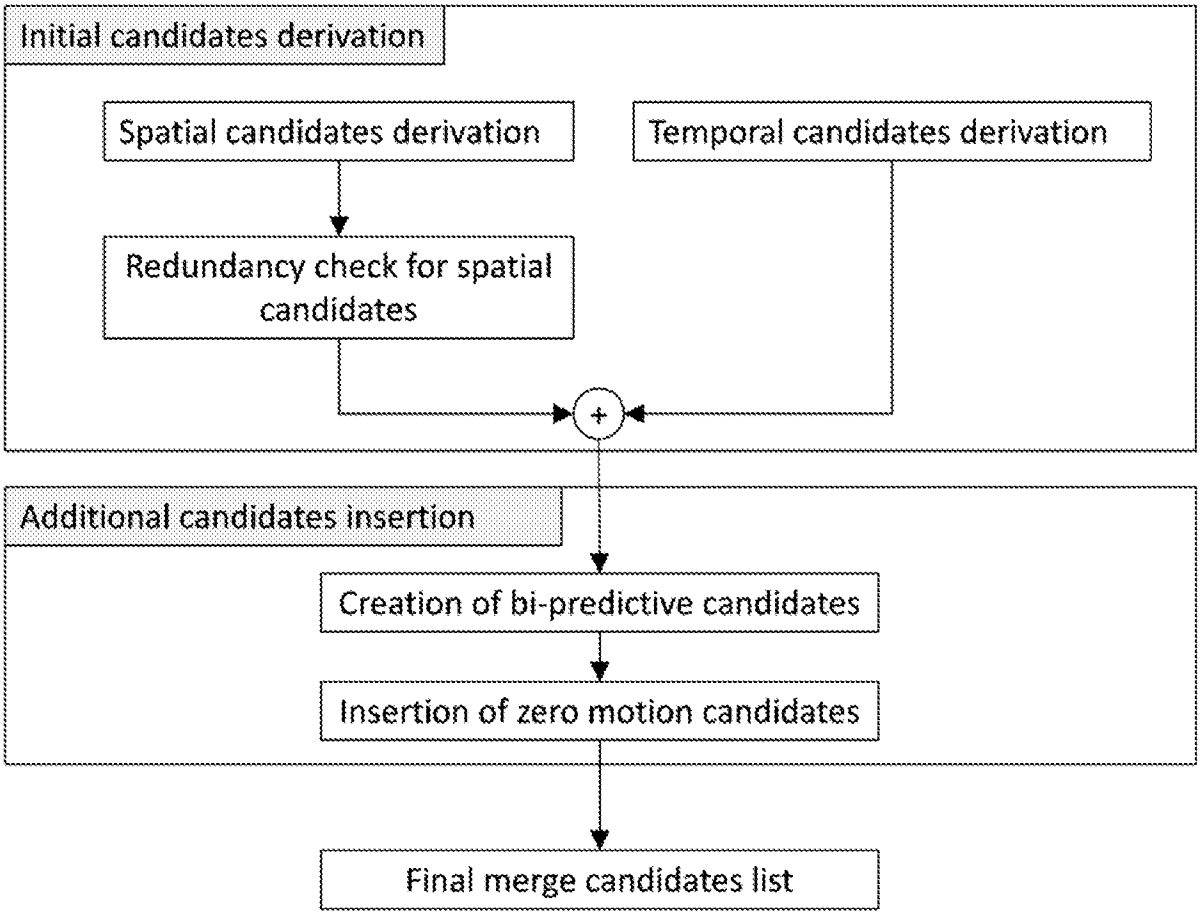


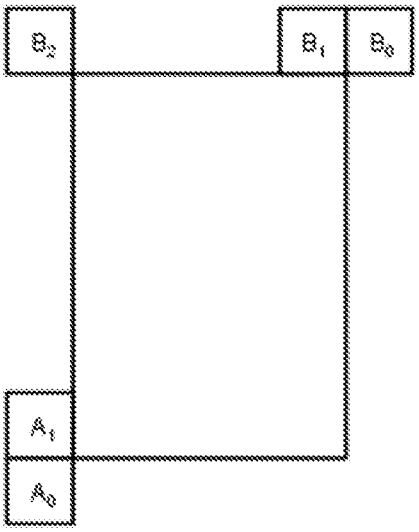
Illustration of a QTBT structure

FIG. 1



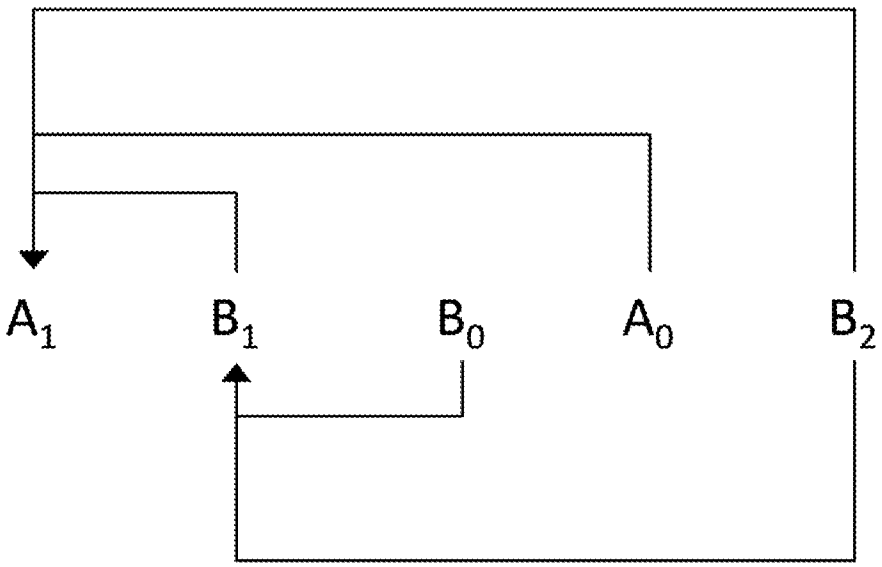
Derivation process for merge candidates list construction.

FIG. 2



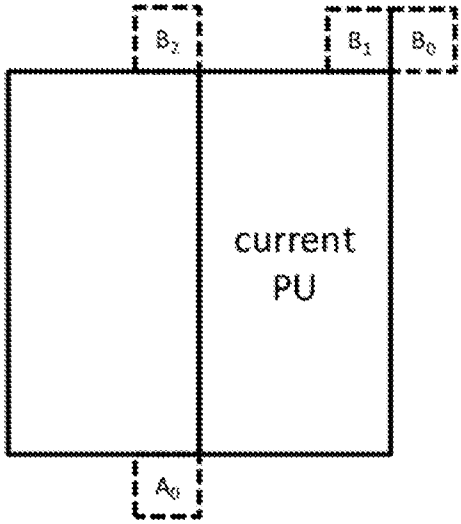
Positions of spatial merge candidates

FIG. 3



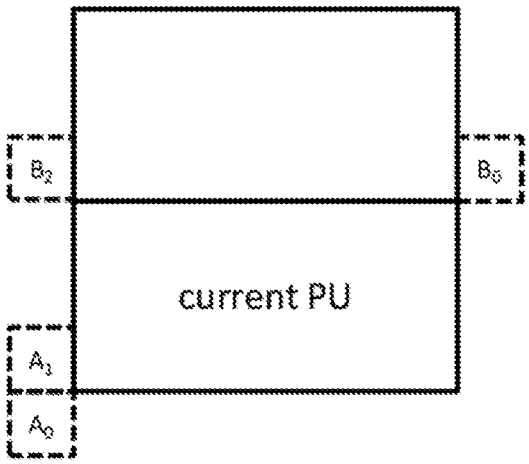
Candidate pairs considered for redundancy check of spatial merge candidates

FIG. 4



second PU of Nx2N

FIG. 5A



second PU of 2NxN

FIG. 5B

Positions for the second PU of N×2N and 2N×N partitions

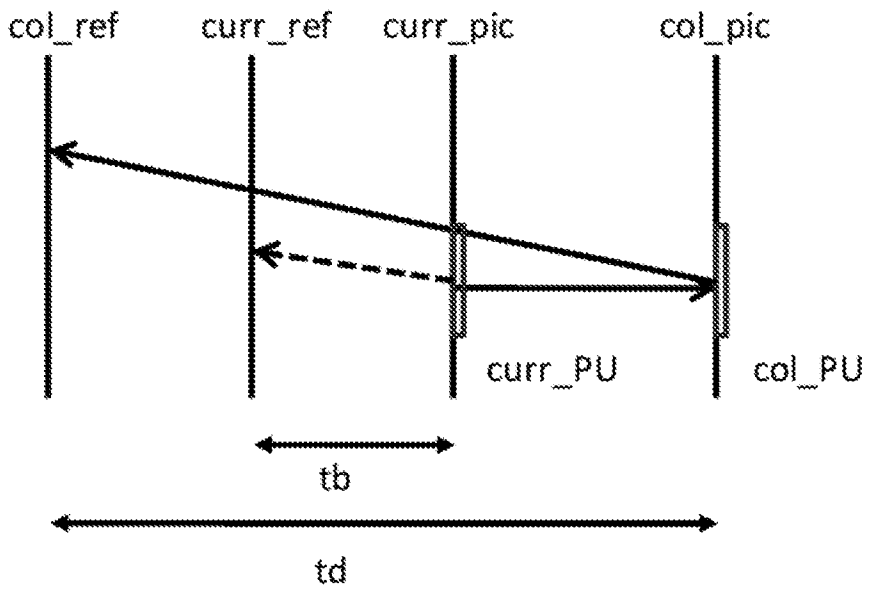
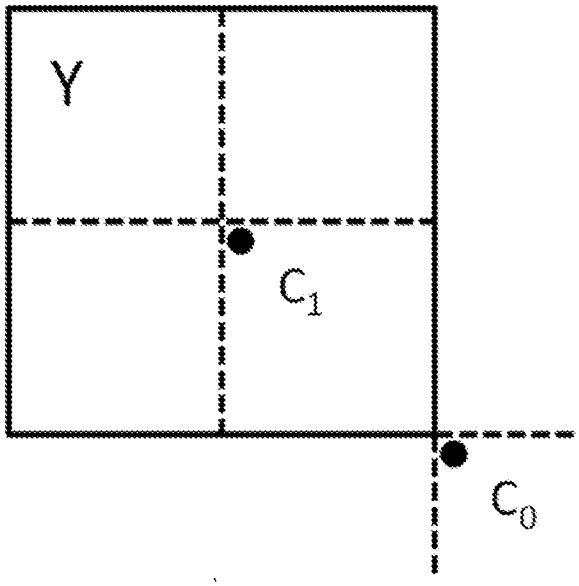


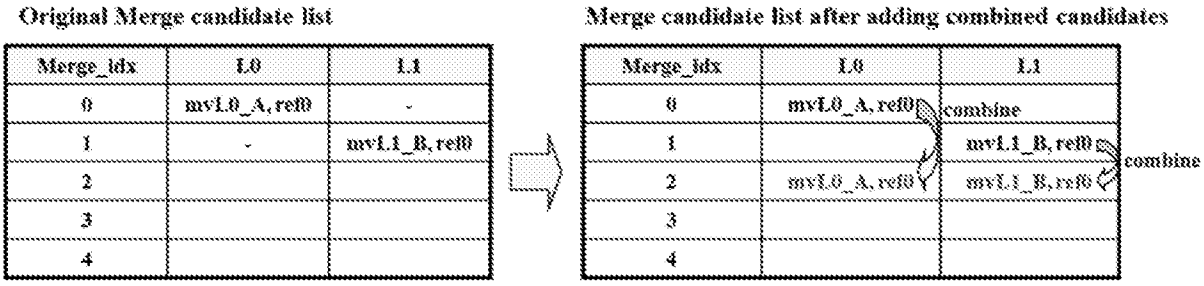
Illustration of motion vector scaling for temporal merge candidate

FIG. 6



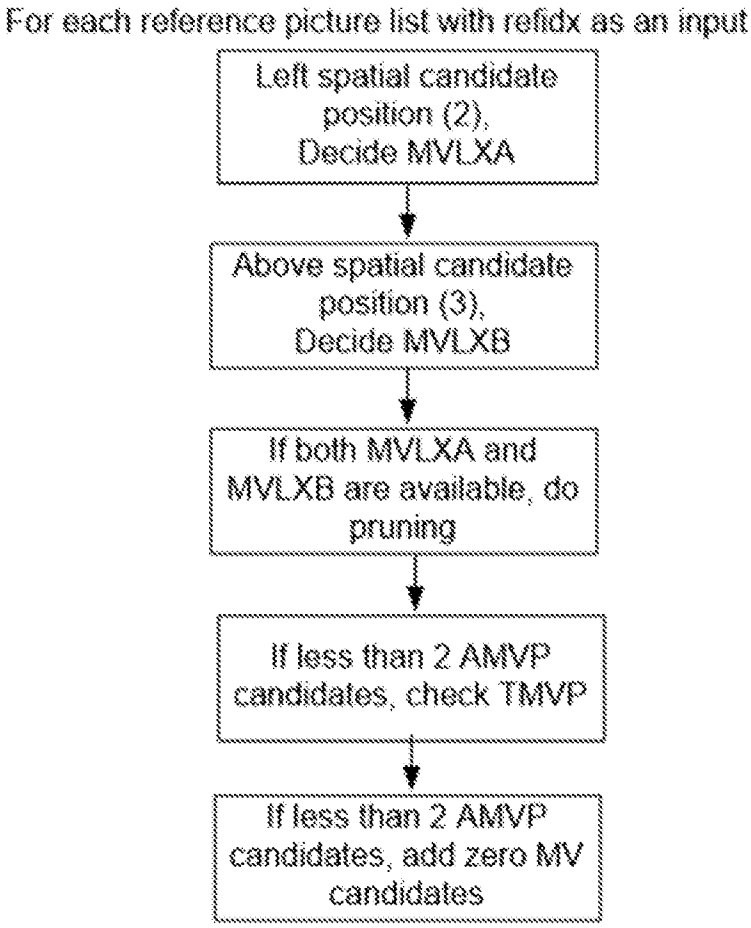
Candidate positions for temporal merge candidate, C0 and C1

FIG. 7



Example of combined bi-predictive merge candidate

FIG. 8



Derivation process for motion vector prediction candidates

FIG. 9

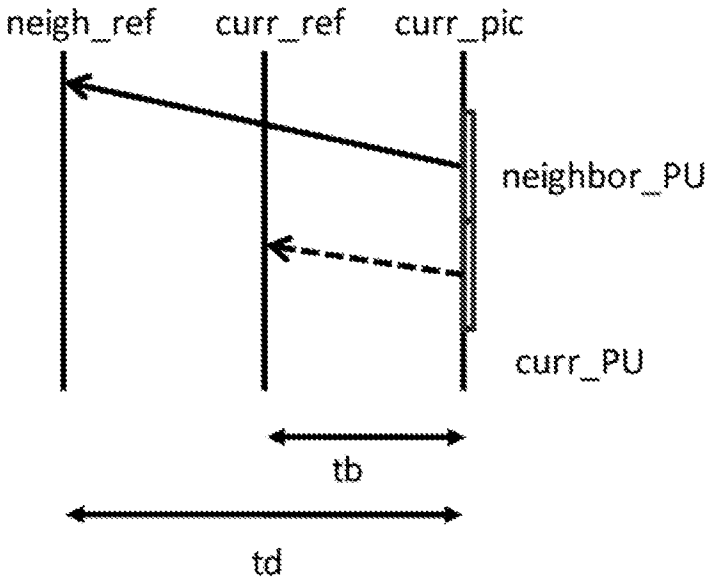


Illustration of motion vector scaling for spatial motion vector candidate

FIG. 10

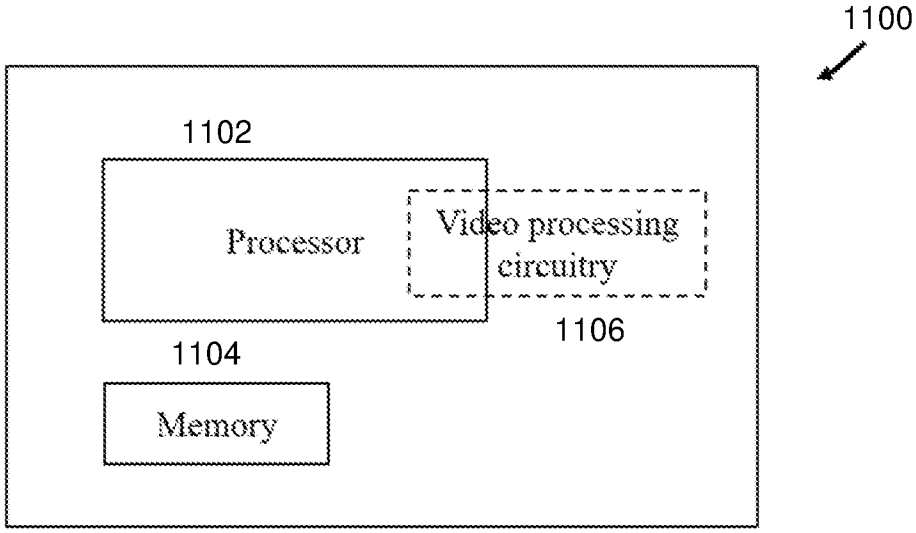

1100
1102
Processor
Video processing circuitry
1106
1104
Memory

FIG. 11A

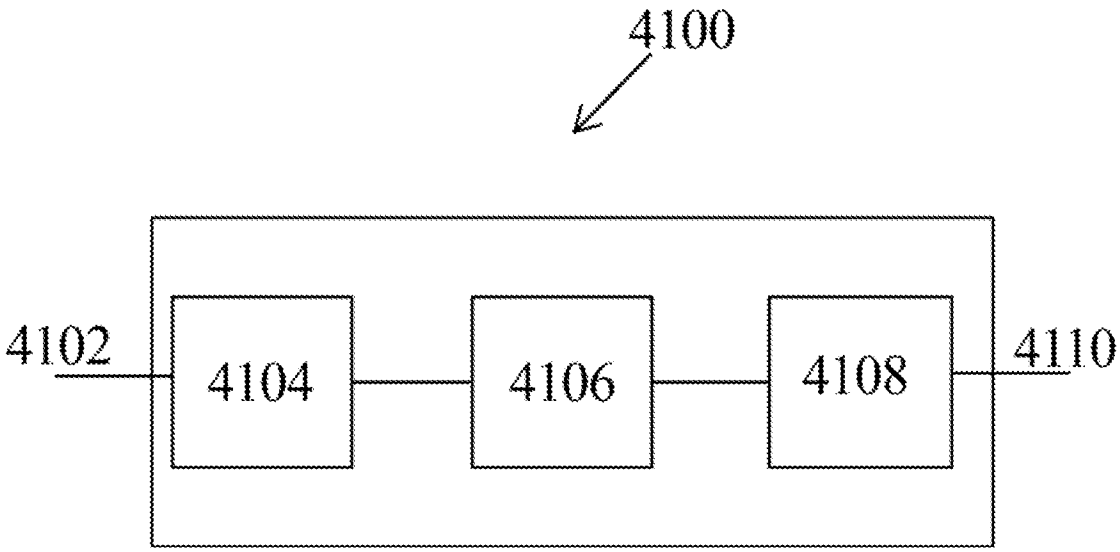


FIG. 11B

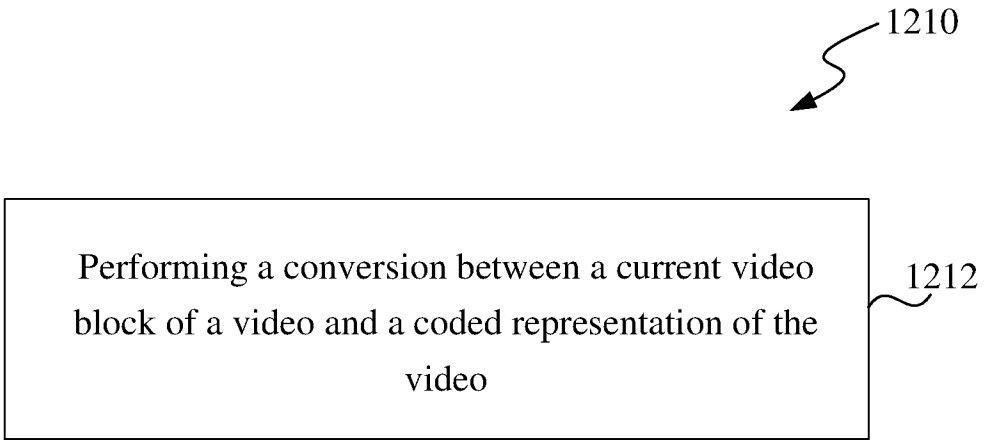

1210
Performing a conversion between a current video block of a video and a coded representation of the video
1212

FIG. 12A

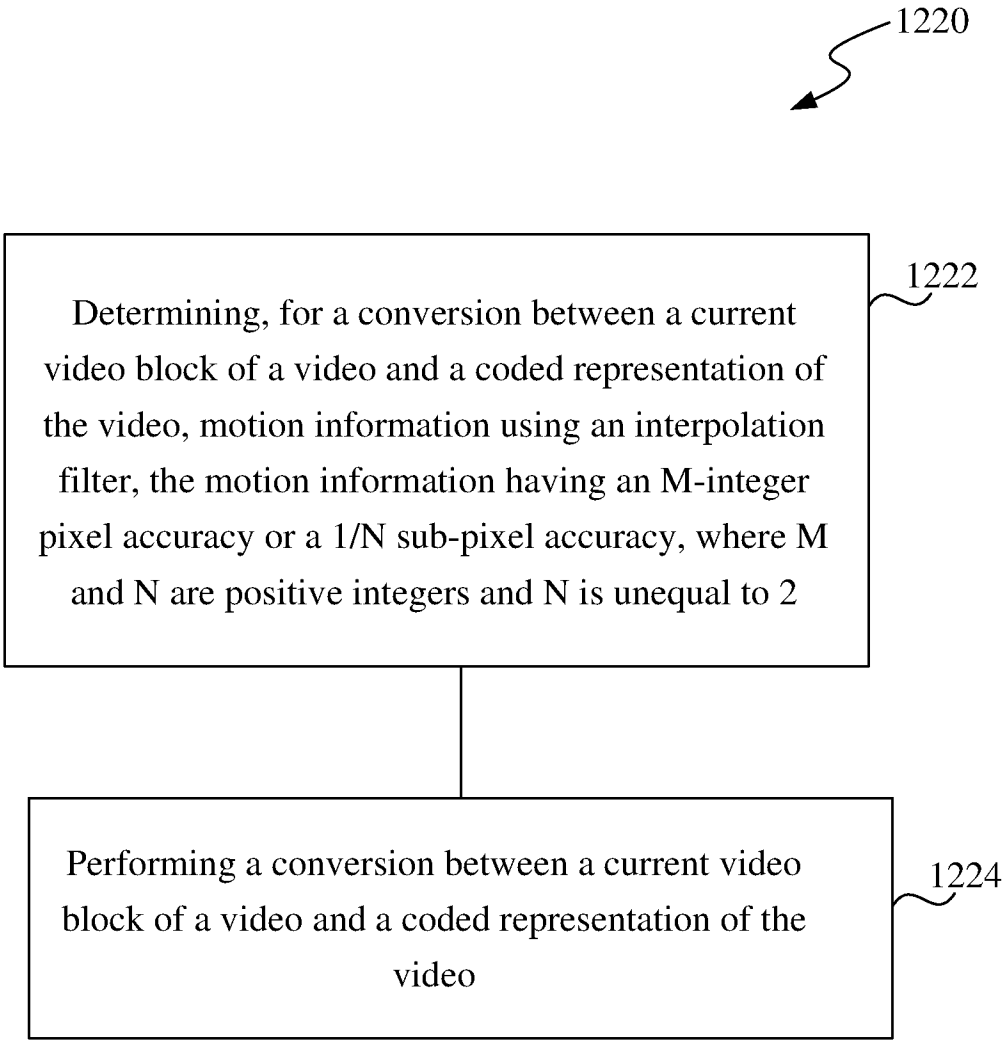

1220
Determining, for a conversion between a current video block of a video and a coded representation of the video, motion information using an interpolation filter, the motion information having an M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers and N is unequal to 2
1222
Performing a conversion between a current video block of a video and a coded representation of the video
1224

FIG. 12B

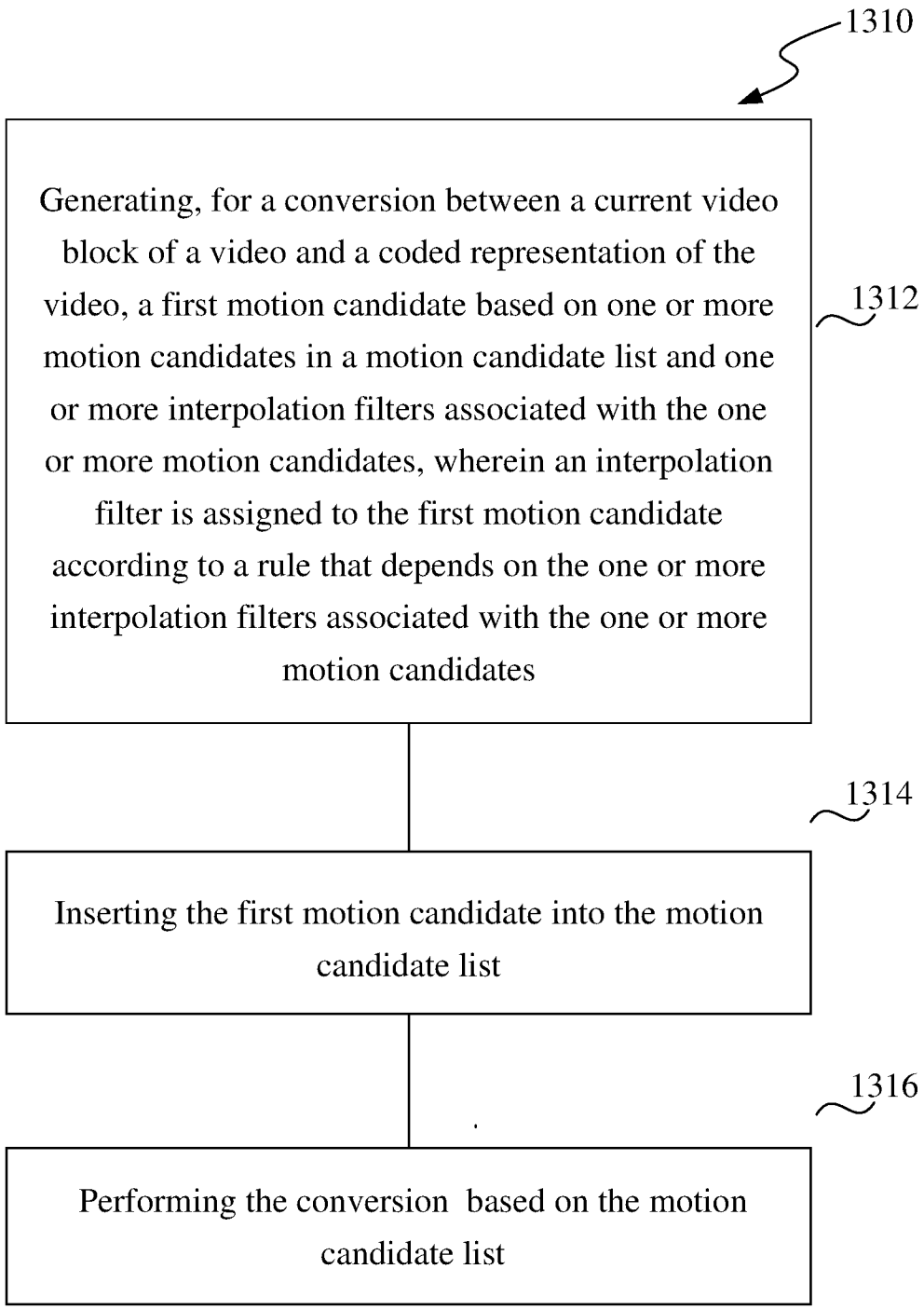

1310
Generating, for a conversion between a current video block of a video and a coded representation of the video, a first motion candidate based on one or more motion candidates in a motion candidate list and one or more interpolation filters associated with the one or more motion candidates, wherein an interpolation filter is assigned to the first motion candidate according to a rule that depends on the one or more interpolation filters associated with the one or more motion candidates
1312
1314
Inserting the first motion candidate into the motion candidate list
1316
Performing the conversion based on the motion candidate list

FIG. 13A

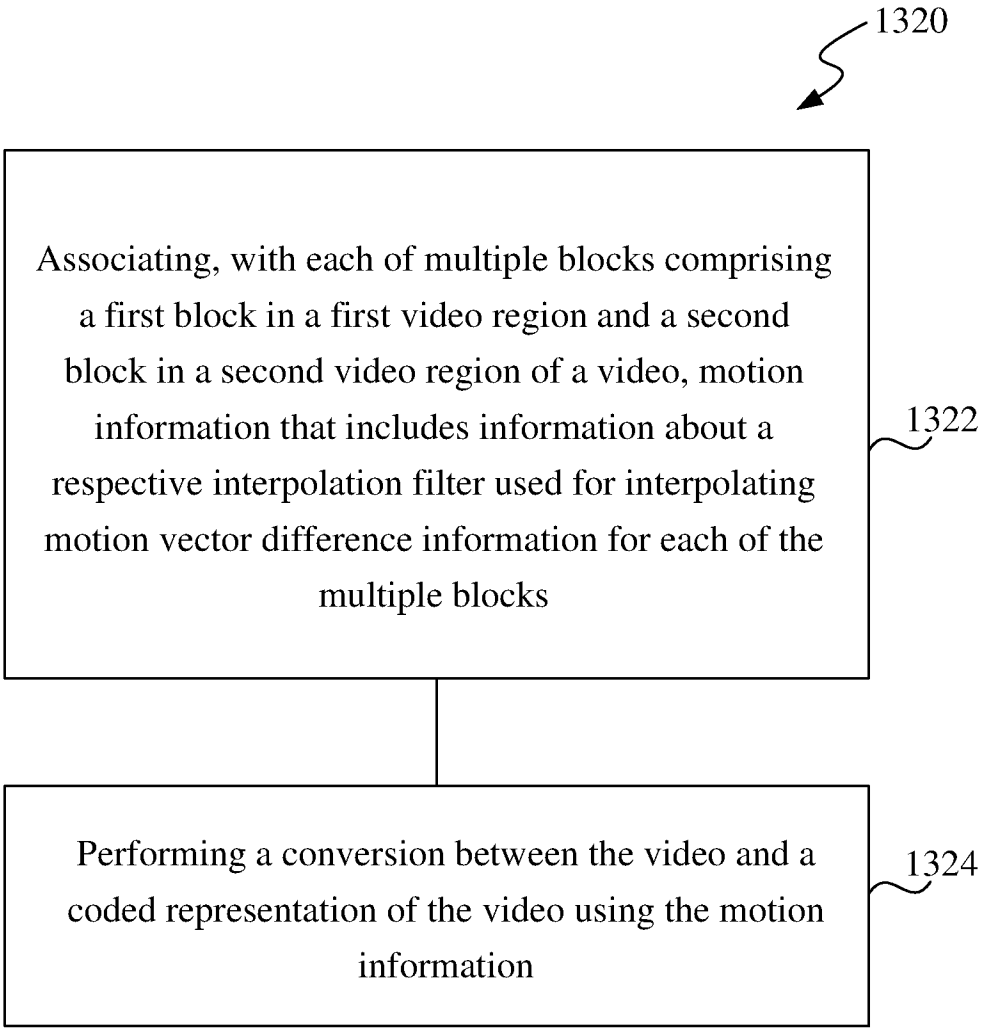

1320
Associating, with each of multiple blocks comprising a first block in a first video region and a second block in a second video region of a video, motion information that includes information about a respective interpolation filter used for interpolating motion vector difference information for each of the multiple blocks
1322
Performing a conversion between the video and a coded representation of the video using the motion information
1324

FIG. 13B

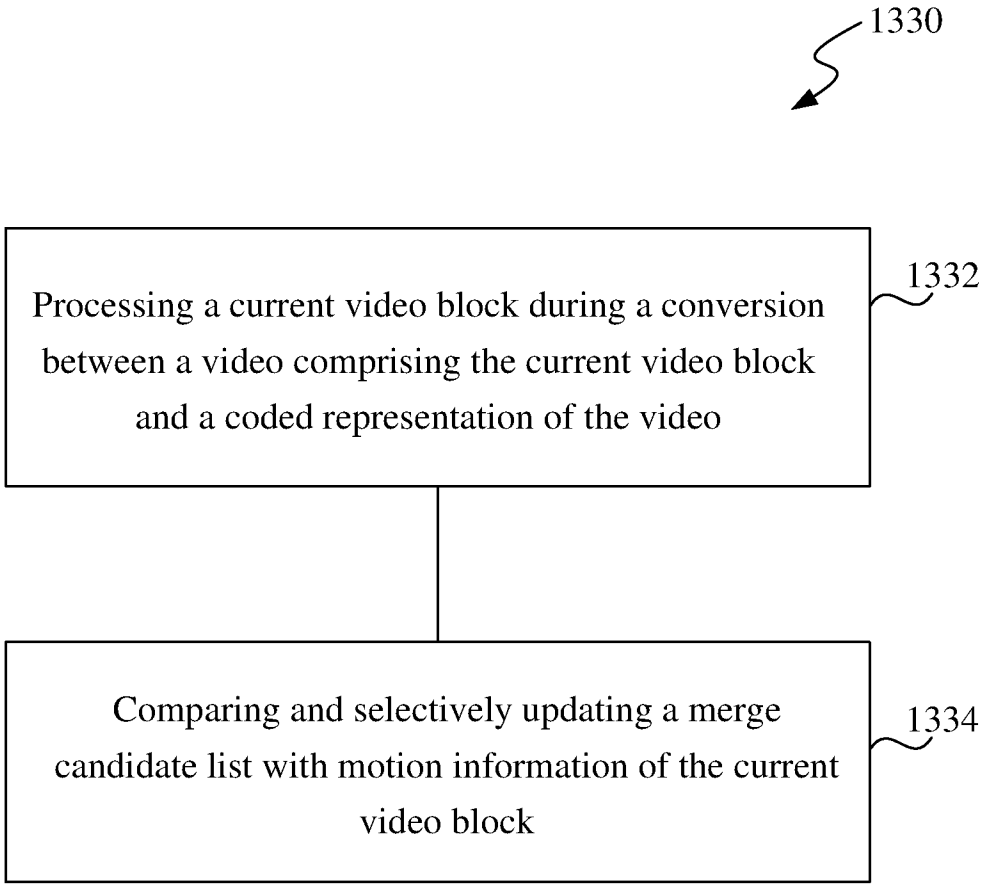

1330
Processing a current video block during a conversion between a video comprising the current video block and a coded representation of the video
1332
Comparing and selectively updating a merge candidate list with motion information of the current video block
1334

FIG. 13C

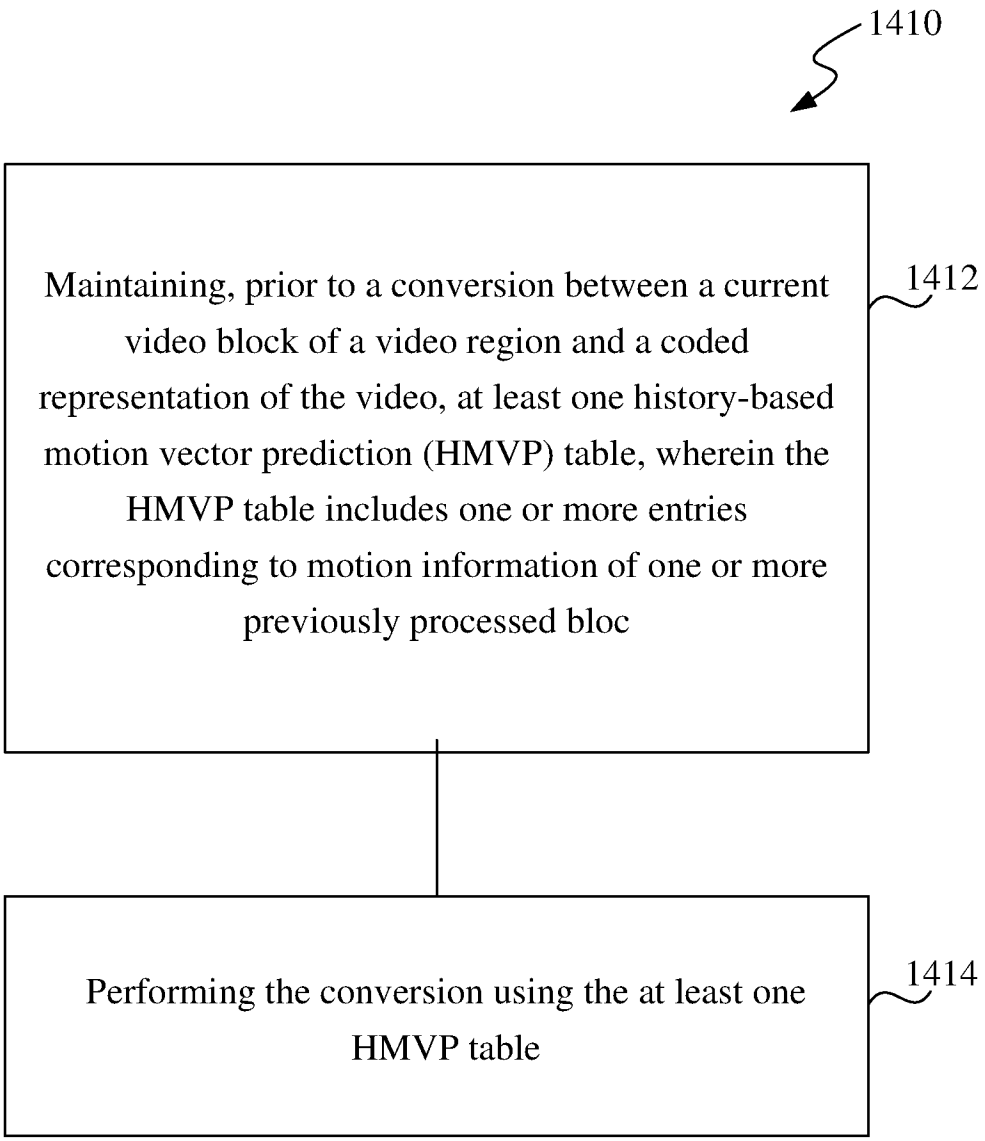

1410
Maintaining, prior to a conversion between a current video block of a video region and a coded representation of the video, at least one history-based motion vector prediction (HMVP) table, wherein the HMVP table includes one or more entries corresponding to motion information of one or more previously processed bloc
1412
Performing the conversion using the at least one HMVP table
1414

FIG. 14A

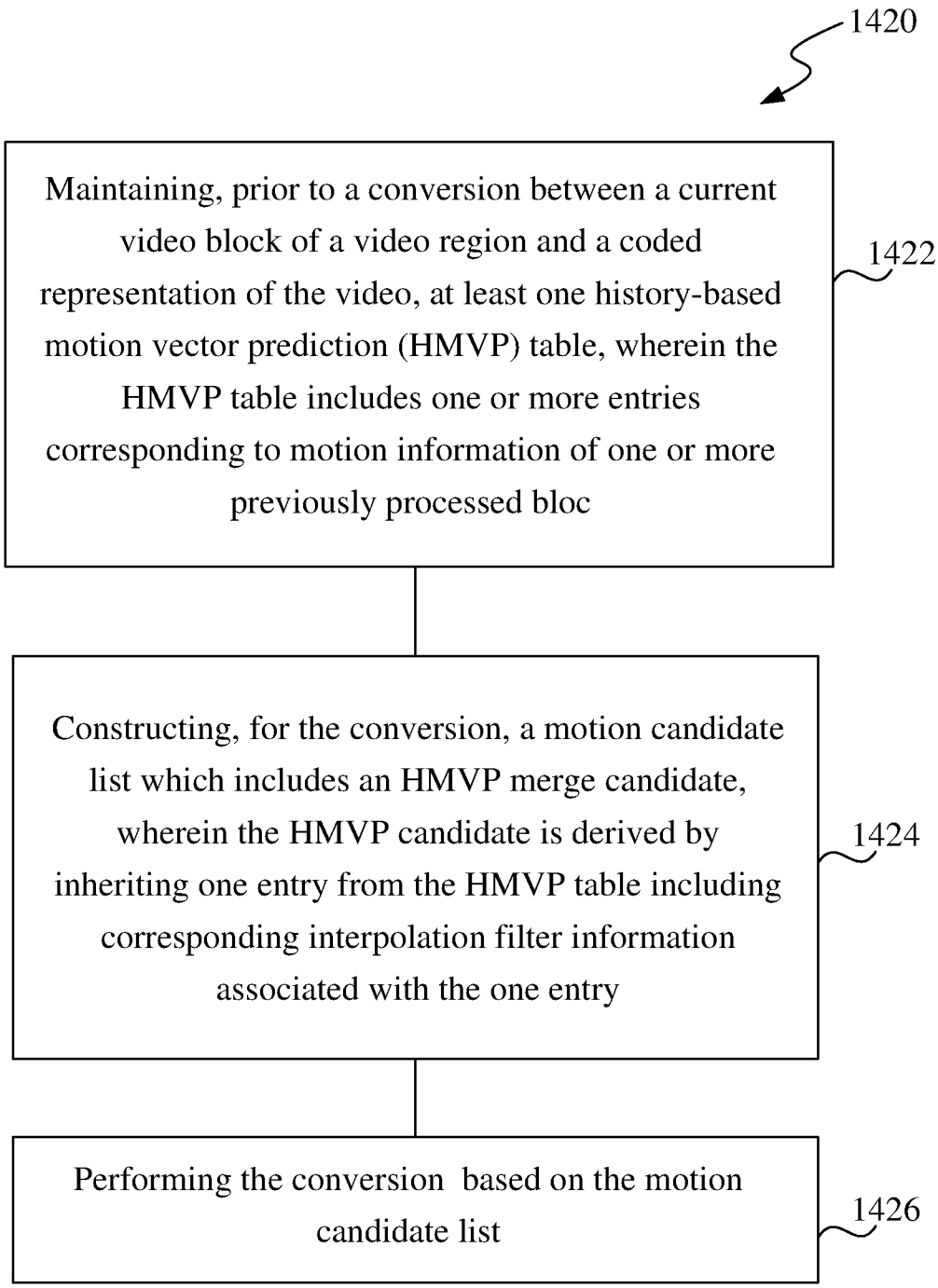

1420
Maintaining, prior to a conversion between a current video block of a video region and a coded representation of the video, at least one history-based motion vector prediction (HMVP) table, wherein the HMVP table includes one or more entries corresponding to motion information of one or more previously processed bloc
1422
Constructing, for the conversion, a motion candidate list which includes an HMVP merge candidate, wherein the HMVP candidate is derived by inheriting one entry from the HMVP table including corresponding interpolation filter information associated with the one entry
1424
Performing the conversion based on the motion candidate list
1426

FIG. 14B

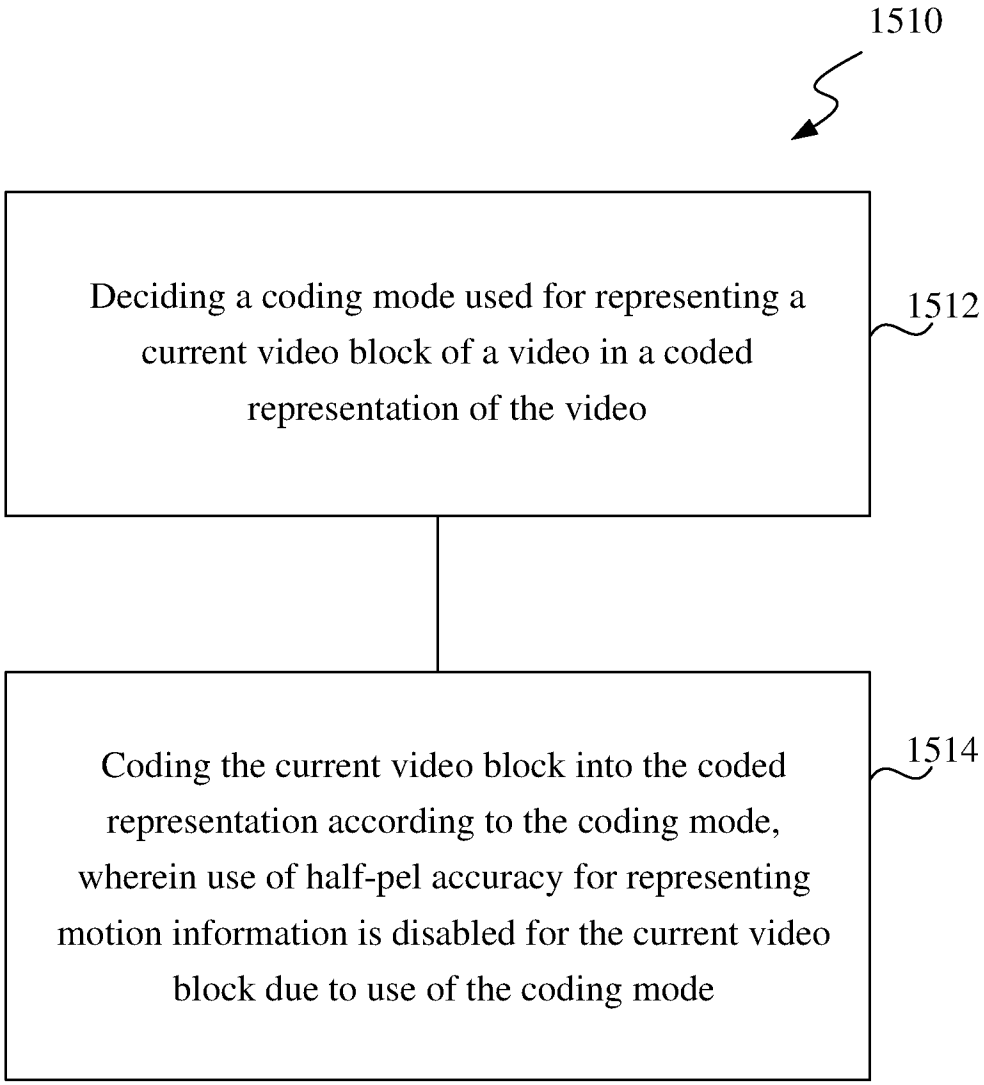

1510
Deciding a coding mode used for representing a current video block of a video in a coded representation of the video
1512
Coding the current video block into the coded representation according to the coding mode, wherein use of half-pel accuracy for representing motion information is disabled for the current video block due to use of the coding mode
1514

FIG. 15A

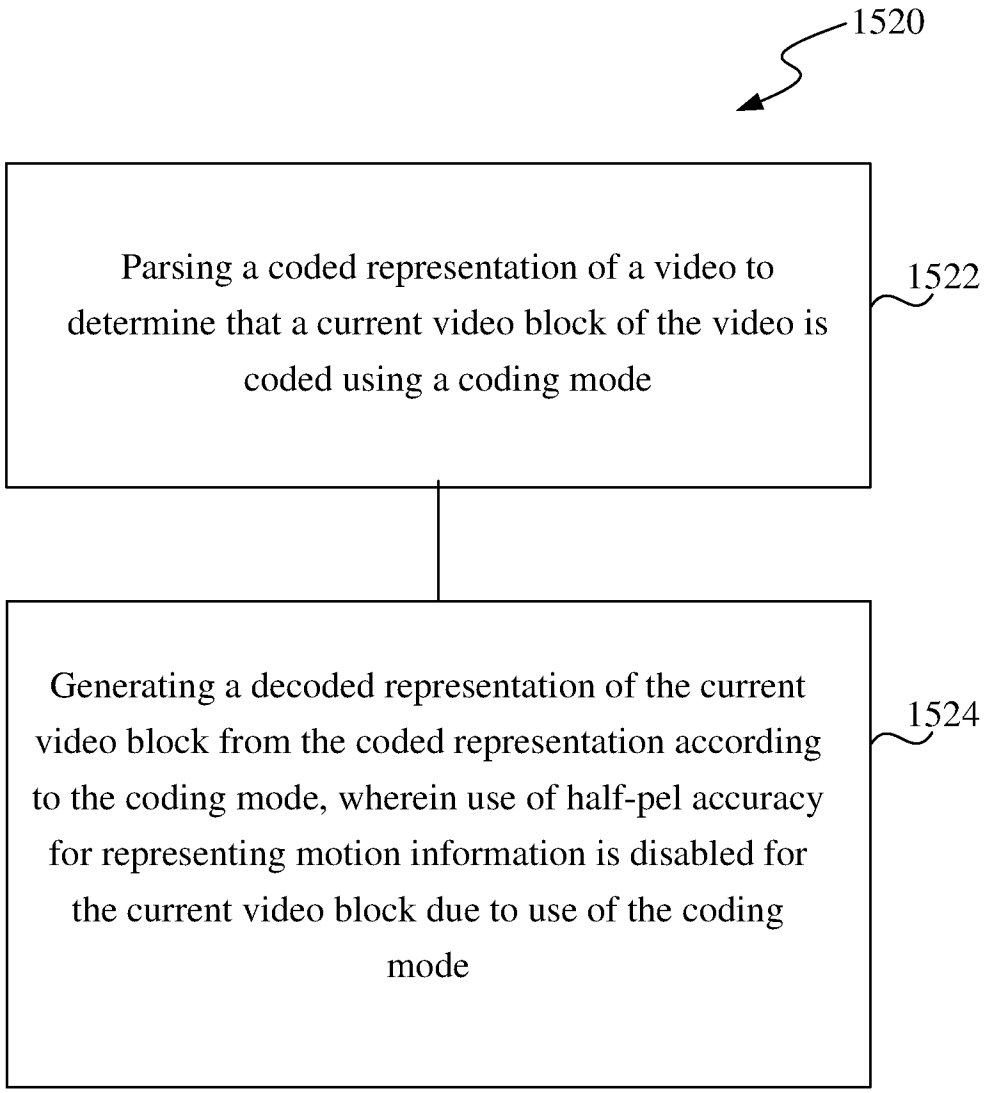

1520
Parsing a coded representation of a video to determine that a current video block of the video is coded using a coding mode
1522
Generating a decoded representation of the current video block from the coded representation according to the coding mode, wherein use of half-pel accuracy for representing motion information is disabled for the current video block due to use of the coding mode
1524

FIG. 15B

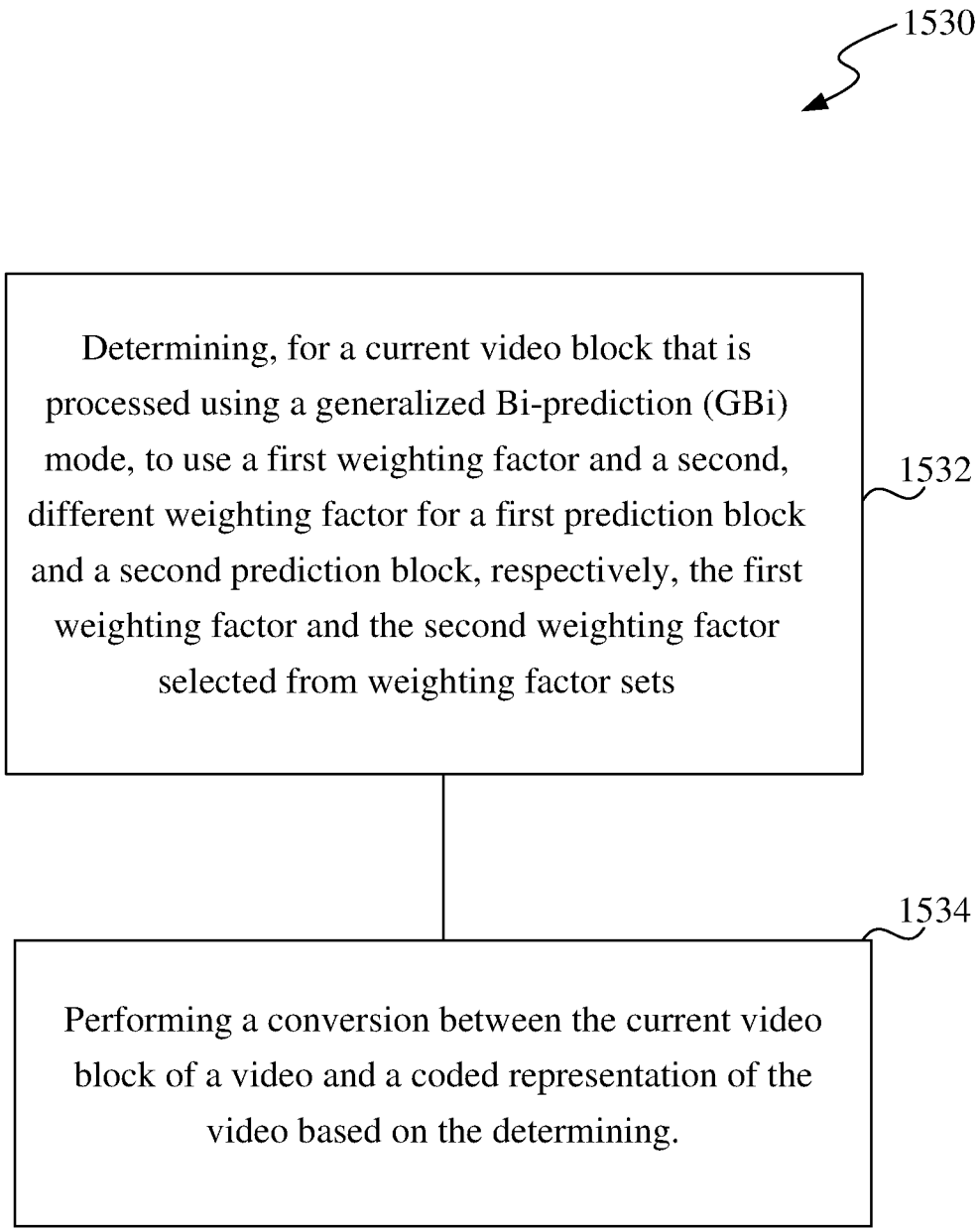

1530
Determining, for a current video block that is processed using a generalized Bi-prediction (GBi) mode, to use a first weighting factor and a second, different weighting factor for a first prediction block and a second prediction block, respectively, the first weighting factor and the second weighting factor selected from weighting factor sets
1532
1534
Performing a conversion between the current video block of a video and a coded representation of the video based on the determining.

FIG. 15C

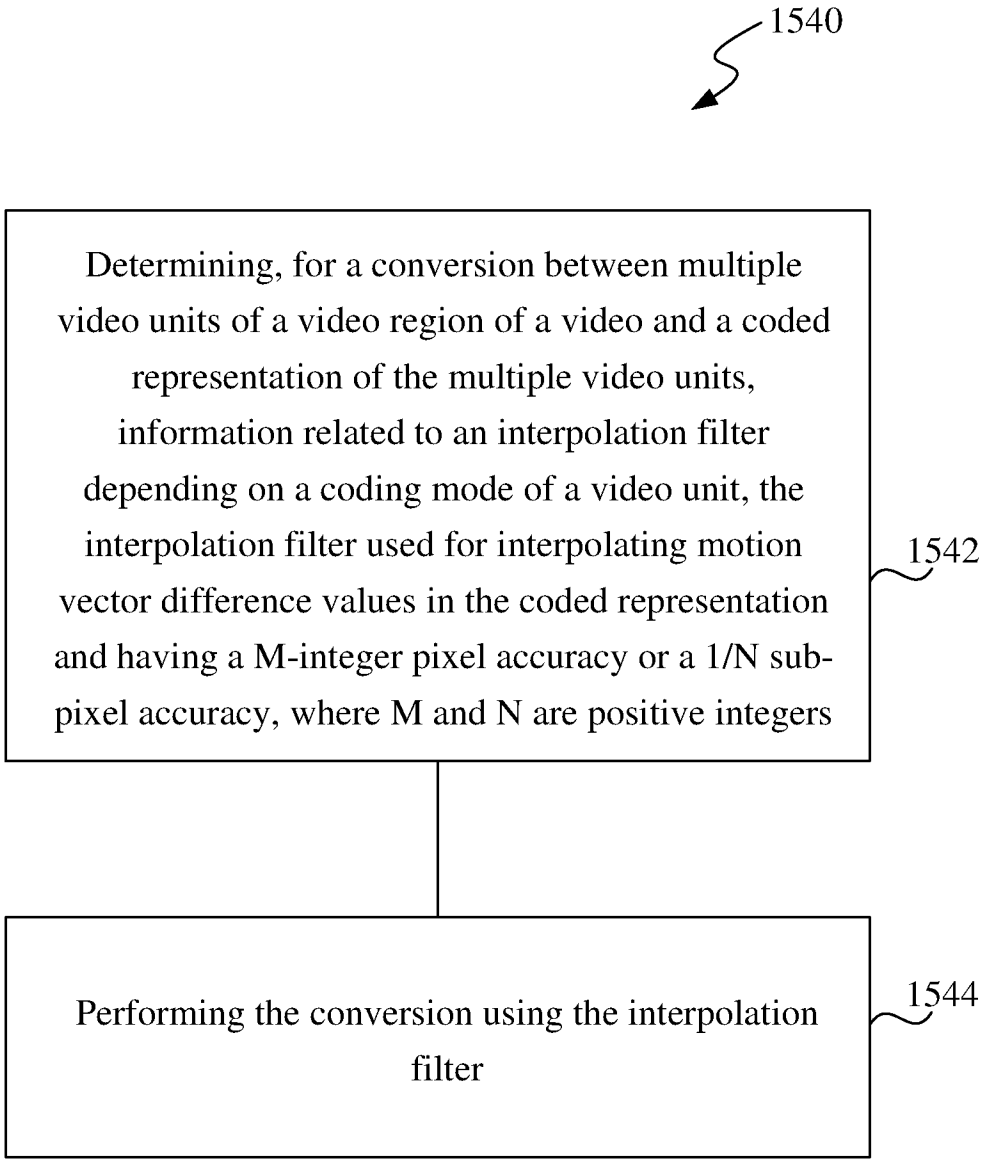

1540
Determining, for a conversion between multiple video units of a video region of a video and a coded representation of the multiple video units, information related to an interpolation filter depending on a coding mode of a video unit, the interpolation filter used for interpolating motion vector difference values in the coded representation and having a M-integer pixel accuracy or a 1/N sub-pixel accuracy, where M and N are positive integers
1542
Performing the conversion using the interpolation filter
1544

FIG. 15D

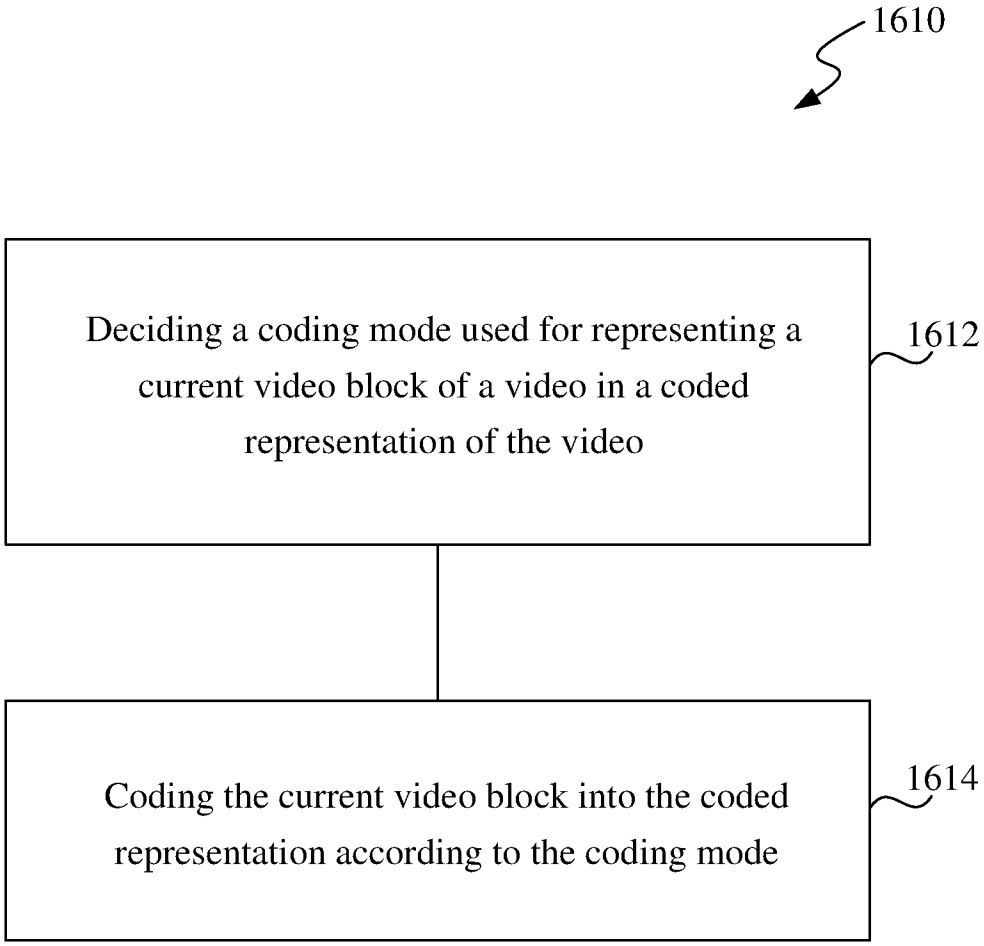

1610
Deciding a coding mode used for representing a current video block of a video in a coded representation of the video
1612
Coding the current video block into the coded representation according to the coding mode
1614

FIG. 16A

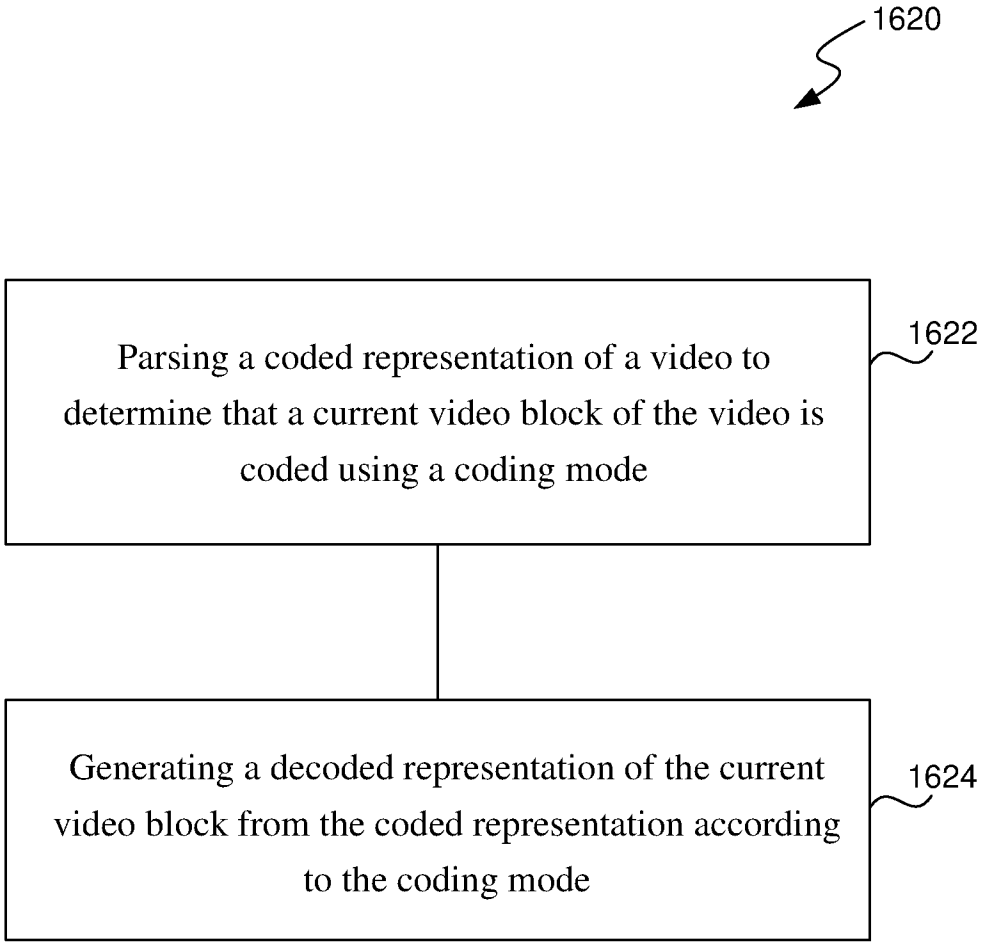

1620
Parsing a coded representation of a video to determine that a current video block of the video is coded using a coding mode
1622
Generating a decoded representation of the current video block from the coded representation according to the coding mode
1624

FIG. 16B

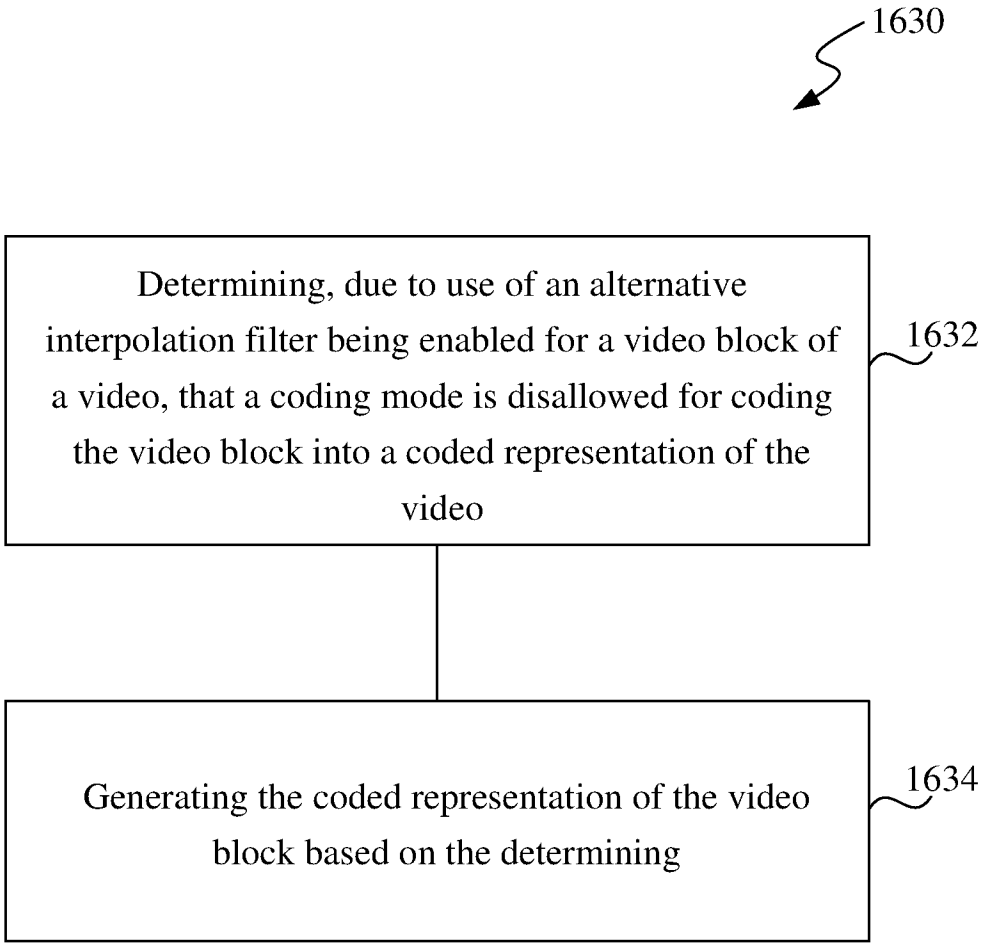

1630
Determining, due to use of an alternative interpolation filter being enabled for a video block of a video, that a coding mode is disallowed for coding the video block into a coded representation of the video
1632
Generating the coded representation of the video block based on the determining
1634

FIG. 16C

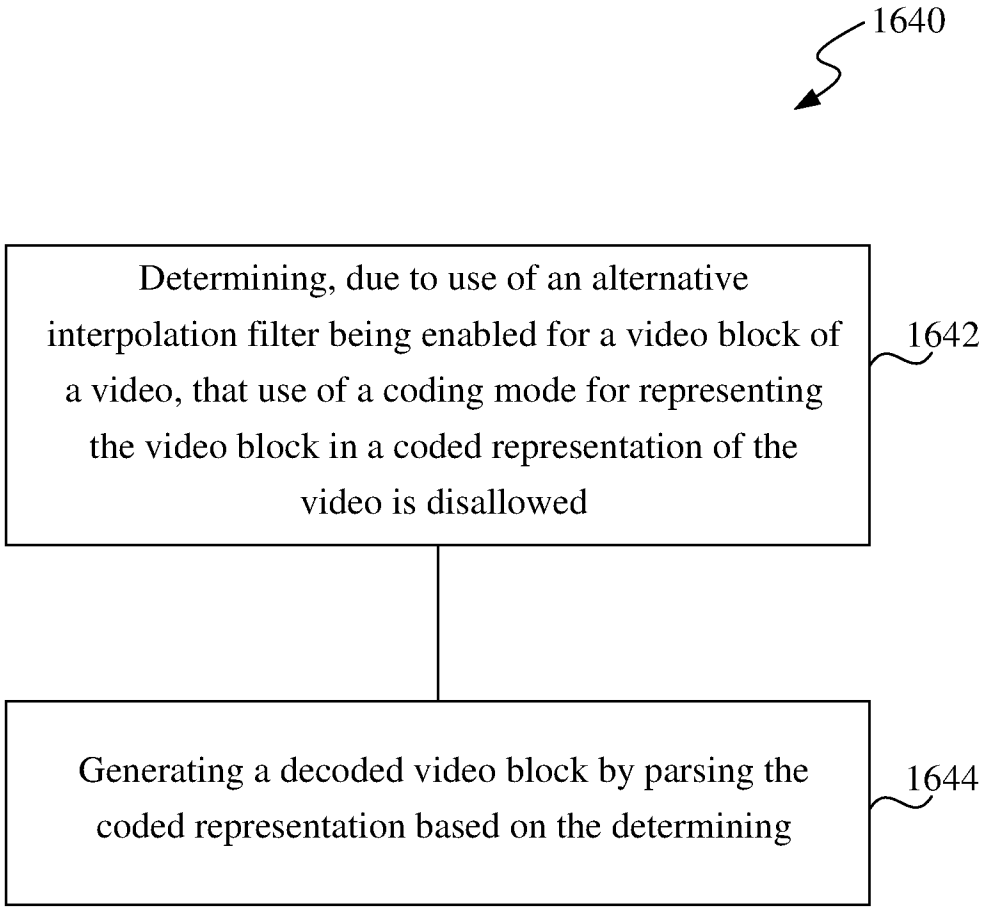

1640
Determining, due to use of an alternative interpolation filter being enabled for a video block of a video, that use of a coding mode for representing the video block in a coded representation of the video is disallowed
1642
Generating a decoded video block by parsing the coded representation based on the determining
1644

FIG. 16D

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/082754

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/00(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

CNPAT,WPI,EPODOC,CNKI,IEEE,JVET:encode, code, decode, interpolate, filter, half, pixel

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 108353176 A (LG ELECTRONICS INC.) 31 July 2018 (2018-07-31) the whole document	1-35
A	CN 106331722 A (HUAWEI TECHNOLOGIES CO., LTD.) 11 January 2017 (2017-01-11) the whole document	1-35
A	CN 106464863 A (MEDIATEK INC.) 22 February 2017 (2017-02-22) the whole document	1-35
A	US 2012183068 A1 (LOU, Jian et al.) 19 July 2012 (2012-07-19) the whole document	1-35
A	HENKEL, Anastasia et al. "Non-CE4: Switched half-pel interpolation filter" *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, 19–27 March 2019,* 27 March 2019 (2019-03-27), page 2, paragraph 1	1-35

☐ Further documents are listed in the continuation of Box C.

☑ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 June 2020	**02 July 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 **YANG,Shuangyi** Telephone No. **86-(10)-53961811**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/082754

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	108353176	A	31 July 2018	WO	2017052009	A1	30 March 2017
				KR	20180059444	A	04 June 2018
				US	2018270485	A1	20 September 2018
				EP	3355582	A1	01 August 2018
				US	10547847	B2	28 January 2020
				JP	2018533298	A	08 November 2018
				EP	3355582	A4	17 April 2019
CN	106331722	A	11 January 2017	CN	106331722	B	26 April 2019
				AU	2016290017	B2	11 July 2019
				CN	107710761	A	16 February 2018
				RU	2018143851	A	21 February 2019
				EP	3306934	A4	23 May 2018
				AU	2016290017	A1	25 January 2018
				US	10560714	B2	11 February 2020
				ZA	201800017	B	28 August 2019
				MX	2018000067	A	16 March 2018
				US	2018139468	A1	17 May 2018
				RU	2018143851	A3	17 July 2019
				RU	2675092	C1	14 December 2018
				KR	102073638	B1	05 February 2020
				JP	2018524918	A	30 August 2018
				EP	3306934	A1	11 April 2018
				KR	20200013266	A	06 February 2020
				KR	20180019688	A	26 February 2018
				WO	2017005128	A1	12 January 2017
				BR	112018000085	A2	28 August 2018
				RU	2699258	C2	04 September 2019
				CN	107710761	B	14 February 2020
				SG	11201710800Q	A	30 January 2018
CN	106464863	A	22 February 2017	EP	3111643	A1	04 January 2017
				US	2017085913	A1	23 March 2017
				EP	3111643	A4	25 October 2017
				CN	110225360	A	10 September 2019
				US	2019253731	A1	15 August 2019
				CN	106464863	B	12 July 2019
				WO	2015149699	A1	08 October 2015
				EP	3111643	B1	25 September 2019
				US	10321151	B2	11 June 2019
US	2012183068	A1	19 July 2012	US	9049454	B2	02 June 2015