

TABLE MAINTENANCE FOR HMVP CANDIDATE STORAGE

CROSS REFERENCE TO RELATED APPLICATIONS

[001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefits of International Patent Application No. PCT/CN2018/113716, filed on November 02, 2019 and International Patent Application No. PCT/CN2019/086174, filed on May 09, 2019. The entire disclosure of the aforementioned application is incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

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

BACKGROUND

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

SUMMARY

[004] The disclosed techniques may be used by video or image decoder or encoder embodiments for in which geometry partitions with history based motion vector prediction (HMVP) is used.

[005] In one example aspect, a method of processing video is disclosed. The method includes performing a determination, by a processor, that a first video block is intra-coded or non-merge inter-coded; determining, by the processor, a first sub-portion and a second sub-portion for the first video block based on the determination that the first video block is intra-coded or non-merge inter-coded, wherein one or both of the first sub-portion or the second sub-portion are non-rectangular and non-square portions of the first video block; and performing further processing of the first video block using the first sub-portion and the second sub-portion.

[006] In another example aspect, a method of processing video includes performing a determination, by a processor, that a first video block is intra-coded or non-merge inter-coded; determining, by the processor, a first sub-portion and a second sub-portion for the first video block, wherein one or both of the first sub-portion or the second sub-portion are non-rectangular and non-square portions of the first video block; and performing further processing of the first video block using the first sub-portion and the second sub-portion, wherein at least one sub-portion is merge or non-merge inter coded and using a current image as a reference image.

[007] In another example aspect, a method of processing video includes performing a determination, by a processor, that a first video block is intra-coded or non-merge inter-coded; determining, by the processor, a first sub-portion and a second sub-portion for the first video block, wherein one or both of the first sub-portion or the second sub-portion are non-rectangular and non-square portions of the first video block; and performing further processing of the first video block using the first sub-portion and the second sub-portion, wherein performing further processing of the first video block using the first sub-portion and the second sub-portion is based on inter or intra coded information of non-adjacent spatial video blocks in relation to the first video block.

[008] In another example aspect, a method of processing video includes performing a determination that a first video block is coded with triangular portion mode (TPM) using a triangular prediction portion of the first video block and that a second video block is coded using non-TPM using a non-triangular prediction portion of the second video block; performing further processing of the first video block and the second video block using stored HMVP candidates and storing HMVP candidates associated with the first video block and the second video block.

[009] In another example aspect, a method of processing video includes performing a determination, by a processor, that a first video block includes prediction portions that are non-rectangular and non-square portions of the first video block; identifying an HMVP candidate; adding one or more motion candidates derived from the HMVP candidate to a merge list associated with video blocks that include prediction portions that are non-rectangular and non-square; and performing further processing of the first video block using the merge list.

[0010] In another example aspect, a method for processing video includes: determining, during a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, multiple sub-portions for the current block; determining intra prediction information of the multiple sub-portions; performing the conversion of the current

block using the intra prediction information of the multiple sub-portions; and wherein the current block is intra-coded, and at least one of the multiple sub-portions is a non-rectangular and non-square sub-portion.

[0011] In another example aspect, a method for processing video includes: determining, during a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, multiple sub-portions for the current block; determining motion information of the multiple sub-portions; performing the conversion of the current block using the motion information of the multiple sub-portions; and wherein the current block is non-merge inter-coded, and at least one of the multiple sub-portions is a non-rectangular and non-square sub-portion.

[0012] In another example aspect, a method of video processing includes: performing a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple sub-portions according to a splitting pattern in which a first sub-portion has a non-rectangular, non-square shape; processing a first sub-portion with intra coding mode; and processing a second sub-portion with inter coding mode.

[0013] In another example aspect, a method of video processing includes: performing a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple sub-portions according to a splitting pattern in which a first sub-portion has a non-rectangular, non-square shape; wherein the at least one of the multiple sub-portions is merge or non-merge inter coded and uses a current picture as a reference picture.

[0014] In another example aspect, a method of video processing includes: performing a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple sub-portions according to a splitting pattern in which a first prediction partition has a non-rectangular, non-square shape; and performing the conversion using inter or intra coded information of one or more non-adjacent spatial blocks.

[0015] In another example aspect, a method for processing video includes: determining, during a conversion between a first block of visual media data and a corresponding coded representation of the visual media data, the first block being coded with geometry partition mode; determining,

based on at least one table storing history based motion vector prediction (HMVP) candidates which include motion information based on previously coded blocks, motion information of at least one sub-portion of the first block; performing the conversion of the first block using the determined motion information.

5 [0016] In another example aspect, a method for processing video includes: determining, during a conversion between a first block of visual media data and a corresponding coded representation of the visual media data, the first block being coded with geometry partition mode; determining motion information of at least one sub-portion of the first block; performing the conversion of the first block using the motion information of the at least one sub-portion; wherein determining
10 motion information of at least one sub-portion comprises using at least one history based motion vector prediction (HMVP) candidate which includes motion information based on a previously coded block to construct a motion candidate list and determining the motion information from the motion candidate list.

[0017] In another example aspect, the above-described method may be implemented by a video
15 encoder apparatus that comprises a processor.

[0018] In another example aspect, the above-described method may be implemented by a video decoder apparatus that comprises a processor.

[0019] In yet another example aspect, these methods may be embodied in the form of processor-executable instructions and stored on a computer-readable program medium.

20 [0020] These, and other, aspects are further described in the present document.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] FIG. 1 shows an example of a derivation process for merge candidates list construction.

[0022] FIG. 2 shows an example of positions of spatial merge candidates.

25 [0023] FIG. 3 shows an example of candidate pairs considered for redundancy check of spatial merge candidates.

[0024] FIG. 4 shows an example of positions for the second PU of $N \times 2N$ and $2N \times N$ partitions.

[0025] FIG. 5 shows an example of motion vector scaling for a temporal merge candidate.

[0026] FIG. 6 shows an example of candidate positions for temporal merge candidate, C0 and
30 C1.

[0027] FIG. 7 shows an example of a combined bi-predictive merge candidate.

- [0028] FIG. 8 shows an example of a derivation process for motion vector prediction candidates.
- [0029] FIG. 9 shows an example of motion vector scaling for a spatial motion vector candidate.
- [0030] FIG. 10 shows an example of ATMVP motion prediction for a CU.
- [0031] FIG. 11 shows an example of one CU with four sub-blocks (A-D) and its neighbouring
5 blocks (a-d).
- [0032] FIG. 12 shows an example of a flowchart of encoding with different MV precision.
- [0033] FIG. 13 shows examples of splitting a CU into two triangular prediction units (two partition types).
- [0034] FIG. 14 shows an example of a position of neighboring blocks.
- 10 [0035] FIG. 15 shows an example of a CU applying the 1st weighting factor group.
- [0036] FIG. 16 shows an example of motion vector storage.
- [0037] FIG. 17 shows an example of neighboring blocks (A and L) used for context selection in TPM flag coding.
- [0038] FIG. 18 shows an example of sub-blocks where OBMC applies.
- 15 [0039] FIG. 19 shows an example of neighboring samples used for deriving IC parameters.
- [0040] FIG. 20 shows an example of a simplified affine motion model.
- [0041] FIG. 21 shows an example of affine MVF per sub-block.
- [0042] FIG. 22 shows an example of 4-parameter affine model (a) and 6-parameter affine model (b).
- 20 [0043] FIG. 23 shows an example of MVP for AF_INTER.
- [0044] FIG. 24 shows an example of candidates for AF_MERGE.
- [0045] FIG. 25 shows an example of candidates position for affine merge mode.
- [0046] FIG. 26 shows an example of optical flow trajectory.
- [0047] FIG. 27 shows an example of BIO w/o block extension: a) access positions outside of the
25 block; b) padding is used in order to avoid extra memory access and calculation.
- [0048] FIG. 28 shows an example of DMVR based on bilateral template matching.
- [0049] FIG. 29 is a block diagram of an example of a video processing apparatus.
- [0050] FIG. 30 shows a block diagram of an example implementation of a video encoder.
- [0051] FIG. 31 is a flowchart for an example of a video processing method.
- 30 [0052] FIG. 32 is a flowchart for an example of a video processing method.
- [0053] FIG. 33 is a flowchart for an example of a video processing method.

[0054] FIG. 34 is a flowchart for an example of a video processing method.

[0055] FIG. 35 is a flowchart for an example of a video processing method.

[0056] FIG. 36 is a flowchart for an example of a video processing method.

[0057] FIG. 37 is a flowchart for an example of a video processing method.

5 [0058] FIG. 38 is a flowchart for an example of a video processing method.

[0059] FIG. 39 is a flowchart for an example of a video processing method.

[0060] FIG. 40 is a flowchart for an example of a video processing method.

[0061] FIG. 41 is a flowchart for an example of a video processing method.

[0062] FIG. 42 is a flowchart for an example of a video processing method.

10

DETAILED DESCRIPTION

[0063] The present document provides various techniques that can be used by a decoder of image or video bitstreams to improve the quality of decompressed or decoded digital video or images. For brevity, the term “video” is used herein to include both a sequence of pictures (traditionally
15 called video) and individual images. Furthermore, a video encoder may also implement these techniques during the process of encoding in order to reconstruct decoded frames used for further encoding.

15

[0064] Section headings are used in the present document for ease of understanding and do not limit the embodiments and techniques to the corresponding sections. As such, embodiments from
20 one section can be combined with embodiments from other sections.

20

[0065] 1. Summary

[0066] This patent document is related to video coding technologies. Specifically, it is related to motion vector coding under geometry partition in video coding. It may be applied to the existing video coding standard like HEVC, or the standard (Versatile Video Coding) to be finalized. It may
25 be also applicable to future video coding standards or video codec.

25

[0067] 2. Background

[0068] 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
30 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since

30

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.

[0069] FIG. 30 is a block diagram of an example implementation of a video encoder. FIG. 30 shows that the encoder implementation has a feedback path built in in which the video encoder also performs video decoding functionality (reconstructing compressed representation of video data for use in encoding of next video data).

[0070] 2.1 Inter prediction in HEVC/H.265

[0071] 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.

[0072] 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 a mode is named Advanced motion vector prediction (AMVP) in this disclosure.

[0073] 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.

[0074] 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.

[0075] The following text provides the details on the inter prediction modes specified in HEVC. The description will start with the merge mode.

[0076] 2.1.1. Reference picture list

[0077] In HEVC, the term inter prediction is used to denote prediction derived from data elements (e.g., sample values or motion vectors) of reference pictures other than the current decoded picture. Like in H.264/AVC, a picture can be predicted from multiple reference pictures. The reference pictures that are used for inter prediction are organized in one or more reference picture lists. The reference index identifies which of the reference pictures in the list should be used for creating the prediction signal.

[0078] A single reference picture list, List 0, is used for a P slice and two reference picture lists, List 0 and List 1 are used for B slices. It should be noted reference pictures included in List 0/1 could be from past and future pictures in terms of capturing/display order.

[0079] 2.1.2 Merge Mode

[0080] 2.1.2.1 Derivation of candidates for merge mode

[0081] 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:

[0082] Step 1: Initial candidates derivation

[0083] Step 1.1: Spatial candidates derivation

[0084] Step 1.2: Redundancy check for spatial candidates

[0085] Step 1.3: Temporal candidates derivation

[0086] Step 2: Additional candidates insertion

[0087] Step 2.1: Creation of bi-predictive candidates

[0088] Step 2.2: Insertion of zero motion candidates

[0089] These steps are also schematically depicted in Figure 1. For spatial merge candidate derivation, a maximum of four merge candidates are selected among candidates that are located in five different positions. For temporal merge candidate derivation, a maximum of one merge candidate is selected among two candidates. Since constant number of candidates for each PU is assumed at decoder, additional candidates are generated when the number of candidates obtained from step 1 does not reach the maximum number of merge candidate (MaxNumMergeCand) which

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

5 [0090] In the following, the operations associated with the aforementioned steps are detailed.

[0091] Figure 1 shows an example of a derivation process for merge candidates list construction.

[0092] 2.1.2.2. Spatial candidates derivation

[0093] In the derivation of spatial merge candidates, a maximum of four merge candidates are selected among candidates located in the positions depicted in Figure 2. The order of derivation is
 10 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
 15 pairs are considered in the mentioned redundancy check. Instead only the pairs linked with an arrow in Figure 3 are considered and a candidate is only added to the list if the corresponding candidate used for redundancy check has not the same motion information. Another source of duplicate motion information is the “*second PU*” associated with partitions different from $2N \times 2N$. As an example, Figure 4 depicts the second PU for the case of $N \times 2N$ and $2N \times N$, respectively.
 20 When the current PU is partitioned as $N \times 2N$, candidate at position A_1 is not considered for list construction. In fact, by adding this candidate will lead to two prediction units having the same motion information, which is redundant to just have one PU in a coding unit. Similarly, position B_1 is not considered when the current PU is partitioned as $2N \times N$.

[0094] Figure 2 shows an example of positions of spatial merge candidates.

25 [0095] Figure 3 shows an example of candidate pairs considered for redundancy check of spatial merge candidates.

[0096] Figure 4 shows an example of positions for the second PU of $N \times 2N$ and $2N \times N$ partitions.

[0097] 2.1.2.3. Temporal candidates derivations

[0098] In this step, only one candidate is added to the list. Particularly, in the derivation of this
 30 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 5, which is scaled from the motion vector of the co-located PU using the POC distances, t_b and t_d , where t_b is defined to be the POC difference between the reference picture of the current picture and the current picture and t_d is defined to be the POC difference between the reference picture of the co-located picture and the co-located picture. The reference picture index of temporal merge candidate is set equal to zero. A practical realization of the scaling process is described in the HEVC specification. For a B-slice, two motion vectors, one is for reference picture list 0 and the other is for reference picture list 1, are obtained and combined to make the bi-predictive merge candidate.

[0099] Figure 5 shows an example of of motion vector scaling for temporal merge candidate.

[00100] 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 6. 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.

[00101] Figure 6 shows an example of candidate positions for temporal merge candidate, C_0 and C_1 .

[00102] 2.1.2.4. Additional candidates insertion

[00103] 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 7 depicts the case when two candidates in the original list (on the left), which have mvL_0 and $refIdxL_0$ or mvL_1 and $refIdxL_1$, 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.

[00104] Figure 7 shows an example of combined bi-predictive merge candidate.

[00105] 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.

[00106] 2.1.2.5. Motion estimation regions for parallel processing

[00107] 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.

[00108] 2.1.3. AMVP

[00109] 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 8). In the following sections, details about derivation process of motion vector prediction candidate are provided.

[00110] 2.1.3.1 Derivation of AMVP candidates

[00111] Figure 8 summarizes derivation process for motion vector prediction candidate.

[00112] In motion vector prediction, two types of motion vector candidates are considered: spatial motion vector candidate and temporal motion vector candidate. For spatial motion vector candidate

derivation, two motion vector candidates are eventually derived based on motion vectors of each PU located in five different positions as depicted in Figure 2.

[00113] 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.

10 [00114] 2.1.3.2. Spatial motion vector candidates

[00115] 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 2, those positions being the same as those of motion merge. The order of derivation for the left side of the current PU is defined as A0, A1, and scaled A0, scaled A1. The order of derivation for the above side of the current PU is defined as B0, B1, B2, scaled B0, scaled B1, scaled B2. For each side there are therefore four cases that can be used as motion vector candidate, with two cases not required to use spatial scaling, and two cases where spatial scaling is used. The four different cases are summarized as follows.

[00116] No spatial scaling

20 [00117] (1) Same reference picture list, and same reference picture index (same POC)

[00118] (2) Different reference picture list, but same reference picture (same POC)

[00119] Spatial scaling

[00120] (3) Same reference picture list, but different reference picture (different POC)

[00121] (4) Different reference picture list, and different reference picture (different POC)

25 [00122] 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.

30 [00123] Figure 9 shows an example of motion vector scaling for spatial motion vector candidate.

[00124] 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 9. The main difference is that the reference picture list and index of current PU is given as input; the actual scaling process is the same as that of temporal scaling.

5 **[00125] 2.1.3.3. Temporal motion vector candidates**

[00126] 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 6). The reference picture index is signalled to the decoder.

[00127] 2.2. New inter prediction methods in JEM

10 **[00128] 2.2.1. Sub-CU based motion vector prediction**

[00129] In the JEM with QTBT, each CU can have at most one set of motion parameters for each prediction direction. Two sub-CU level motion vector prediction methods are considered in the encoder by splitting a large CU into sub-CUs and deriving motion information for all the sub-CUs of the large CU. Alternative temporal motion vector prediction (ATMVP) method allows each CU
15 to fetch multiple sets of motion information from multiple blocks smaller than the current CU in the collocated reference picture. In spatial-temporal motion vector prediction (STMVP) method motion vectors of the sub-CUs are derived recursively by using the temporal motion vector predictor and spatial neighbouring motion vector.

[00130] To preserve more accurate motion field for sub-CU motion prediction, the motion
20 compression for the reference frames is currently disabled.

[00131] Figure 10 shows an example ATMVP motion prediction for a CU.

[00132] 2.2.1.1. Alternative temporal motion vector prediction

[00133] In the alternative temporal motion vector prediction (ATMVP) method, the motion vectors temporal motion vector prediction (TMVP) is modified by fetching multiple sets of motion
25 information (including motion vectors and reference indices) from blocks smaller than the current CU. The sub-CUs are square $N \times N$ blocks (N is set to 4 by default).

[00134] ATMVP predicts the motion vectors of the sub-CUs within a CU in two steps. The first step is to identify the corresponding block in a reference picture with a so-called temporal vector. The reference picture is called the motion source picture. The second step is to split the current
30 CU into sub-CUs and obtain the motion vectors as well as the reference indices of each sub-CU from the block corresponding to each sub-CU.

[00135] In the first step, a reference picture and the corresponding block is determined by the motion information of the spatial neighbouring blocks of the current CU. To avoid the repetitive scanning process of neighbouring blocks, the first merge candidate in the merge candidate list of the current CU is used. The first available motion vector as well as its associated reference index are set to be the *temporal vector* and the index to the motion source picture. This way, in ATMVP, the corresponding block may be more accurately identified, compared with TMVP, wherein the corresponding block (sometimes called collocated block) is always in a bottom-right or center position relative to the current CU.

[00136] In the second step, a corresponding block of the sub-CU is identified by the temporal vector in the motion source picture, by adding to the coordinate of the current CU the temporal vector. For each sub-CU, the motion information of its corresponding block (the smallest motion grid that covers the center sample) is used to derive the motion information for the sub-CU. After the motion information of a corresponding $N \times N$ block is identified, it is converted to the motion vectors and reference indices of the current sub-CU, in the same way as TMVP of HEVC, wherein motion scaling and other procedures apply. For example, the decoder checks whether the low-delay condition (i.e. the POCs of all reference pictures of the current picture are smaller than the POC of the current picture) is fulfilled and possibly uses motion vector MV_x (the motion vector corresponding to reference picture list X) to predict motion vector MV_y (with X being equal to 0 or 1 and Y being equal to $1-X$) for each sub-CU.

[00137] 2.2.1.2. Spatial-temporal motion vector prediction (STMVP)

[00138] In this method, the motion vectors of the sub-CUs are derived recursively, following raster scan order. Figure 11 illustrates this concept. Let us consider an 8×8 CU which contains four 4×4 sub-CUs A, B, C, and D. The neighbouring 4×4 blocks in the current frame are labelled as a, b, c, and d.

[00139] The motion derivation for sub-CU A starts by identifying its two spatial neighbours. The first neighbour is the $N \times N$ block above sub-CU A (block c). If this block c is not available or is intra coded the other $N \times N$ blocks above sub-CU A are checked (from left to right, starting at block c). The second neighbour is a block to the left of the sub-CU A (block b). If block b is not available or is intra coded other blocks to the left of sub-CU A are checked (from top to bottom, starting at block b). The motion information obtained from the neighbouring blocks for each list is scaled to the first reference frame for a given list. Next, temporal motion vector predictor (TMVP) of sub-

block A is derived by following the same procedure of TMVP derivation as specified in HEVC. The motion information of the collocated block at location D is fetched and scaled accordingly. Finally, after retrieving and scaling the motion information, all available motion vectors (up to 3) are averaged separately for each reference list. The averaged motion vector is assigned as the motion vector of the current sub-CU.

[00140] 2.2.1.3. Sub-CU motion prediction mode signaling

[00141] The sub-CU modes are enabled as additional merge candidates and there is no additional syntax element required to signal the modes. Two additional merge candidates are added to merge candidates list of each CU to represent the ATMVP mode and STMVP mode. Up to seven merge candidates are used, if the sequence parameter set indicates that ATMVP and STMVP are enabled. The encoding logic of the additional merge candidates is the same as for the merge candidates in the HM, which means, for each CU in P or B slice, two more RD checks is needed for the two additional merge candidates.

[00142] In the JEM, all bins of merge index is context coded by CABAC. While in HEVC, only the first bin is context coded and the remaining bins are context by-pass coded.

[00143] 2.2.2. Adaptive motion vector difference resolution

[00144] In HEVC, motion vector differences (MVDs) (between the motion vector and predicted motion vector of a PU) are signalled in units of quarter luma samples when `use_integer_mv_flag` is equal to 0 in the slice header. In the JEM, a locally adaptive motion vector resolution (LAMVR) is introduced. In the JEM, MVD can be coded in units of quarter luma samples, integer luma samples or four luma samples. The MVD resolution is controlled at the coding unit (CU) level, and MVD resolution flags are conditionally signalled for each CU that has at least one non-zero MVD components.

[00145] 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.

[00146] 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.

[00147] In the encoder, CU-level RD checks are used to determine which MVD resolution is to be used for a CU. That is, the CU-level RD check is performed three times for each MVD resolution. To accelerate encoder speed, the following encoding schemes are applied in the JEM.

[00148] During RD check of a CU with normal quarter luma sample MVD resolution, the motion information of the current CU (integer luma sample accuracy) is stored. The stored motion information (after rounding) is used as the starting point for further small range motion vector refinement during the RD check for the same CU with integer luma sample and 4 luma sample MVD resolution so that the time-consuming motion estimation process is not duplicated three times.

[00149] RD check of a CU with 4 luma sample MVD resolution is conditionally invoked. For a CU, when RD cost integer luma sample MVD resolution is much larger than that of quarter luma sample MVD resolution, the RD check of 4 luma sample MVD resolution for the CU is skipped.

[00150] The encoding process is shown in Figure 12. First, 1/4 pel MV is tested and the RD cost is calculated and denoted as RDCost0, then integer MV is tested and the RD cost is denoted as RDCost1. If $RDCost1 < th * RDCost0$ (wherein th is a positive value), then 4-pel MV is tested; otherwise, 4-pel MV is skipped. Basically, motion information and RD cost etc. are already known for 1/4 pel MV when checking integer or 4-pel MV, which can be reused to speed up the encoding process of integer or 4-pel MV.

[00151] 2.2.3. Triangular prediction mode

[00152] The concept of the triangular prediction mode (TPM) is to introduce a new triangular partition for motion compensated prediction. As shown in Figure 13, it splits a CU into two triangular prediction units, in either diagonal or inverse diagonal direction. Each triangular prediction unit in the CU is inter-predicted using its own uni-prediction motion vector and reference frame index which are derived from a uni-prediction candidate list. An adaptive weighting process is performed to the diagonal edge after predicting the triangular prediction units. Then, the transform and quantization process are applied to the whole CU. It is noted that this mode is only applied to skip and merge modes.

[00153] 2.2.3.1 Uni-prediction candidate list for TPM

[00154] The uni-prediction candidate list consists of five uni-prediction motion vector candidates.

It is derived from seven neighboring blocks including five spatial neighboring blocks (1 to 5) and two temporal co-located blocks (6 to 7), as shown in Figure 14. The motion vectors of the seven

neighboring blocks are collected and put into the uni-prediction candidate list according in the order of **uni-prediction motion vectors, L0 motion vector of bi-prediction motion vectors, L1 motion vector of bi-prediction motion vectors, and averaged motion vector of the L0 and L1 motion vectors of bi-prediction motion vectors**. If the number of candidates is less than five, zero motion vector is added to the list.

[00155] More specifically, the following steps are involved:

[00156] Obtain motion candidates from A1, B1, B0, A0, B2, Col and Col2 (corresponding to block 1-7 in Figure 14) without any pruning operations.

[00157] Set variable numCurrMergeCand = 0

10 [00158] For each motion candidates derived from A1, B1, B0, A0, B2, Col and Col2 and numCurrMergeCand is less than 5, if the motion candidate is uni-prediction (either from List 0 or List 1), it is added to the merge list with numCurrMergeCand increased by 1. Such added motion candidates is named 'originally uni-predicted candidate'.

[00159] Full pruning is applied.

15 [00160] For each motion candidates derived from A1, B1, B0, A0, B2, Col and Col2 and numCurrMergeCand is less than 5, if the motion candidate is bi-prediction, the motion information from List 0 is added to the merge list (that is, modified to be uni-prediction from List 0) and numCurrMergeCand increased by 1. Such added motion candidates is named 'Truncated List0-predicted candidate'.

20 [00161] Full pruning is applied.

[00162] For each motion candidates derived from A1, B1, B0, A0, B2, Col and Col2 and numCurrMergeCand is less than 5, if the motion candidate is bi-prediction, the motion information from List 1 is added to the merge list (that is, modified to be uni-prediction from List 1) and numCurrMergeCand increased by 1. Such added motion candidates is named 'Truncated List1-predicted candidate'.

25 [00163] Full pruning is applied.

[00164] For each motion candidates derived from A1, B1, B0, A0, B2, Col and Col2 and numCurrMergeCand is less than 5, if the motion candidate is bi-prediction,

[00165] If List 0 reference picture's slice QP is smaller than List 1 reference picture's slice QP, the motion information of List 1 is firstly scaled to List 0 reference picture, and the average of the two MVs (one is from original List 0, and the other is the scaled MV from List 1) is added to the

merge list, that is averaged uni-prediction from List 0 motion candidate and numCurrMergeCand increased by 1.

[00166] Otherwise, the motion information of List 0 is firstly scaled to List 1 reference picture, and the average of the two MVs (one is from original List 1, and the other is the scaled MV from List 0) is added to the merge list, that is averaged uni-prediction from List 1 motion candidate and numCurrMergeCand increased by 1.

[00167] Full pruning is applied.

[00168] If numCurrMergeCand is less than 5, zero motion vector candidates are added.

[00169] 2.2.3.2. Adaptive weighting process

[00170] After predicting each triangular prediction unit, an adaptive weighting process is applied to the diagonal edge between the two triangular prediction units to derive the final prediction for the whole CU. Two weighting factor groups are defined as follows:

[00171] 1st weighting factor group: $\{7/8, 6/8, 4/8, 2/8, 1/8\}$ and $\{7/8, 4/8, 1/8\}$ are used for the luminance and the chrominance samples, respectively;

[00172] 2nd weighting factor group: $\{7/8, 6/8, 5/8, 4/8, 3/8, 2/8, 1/8\}$ and $\{6/8, 4/8, 2/8\}$ are used for the luminance and the chrominance samples, respectively.

[00173] Weighting factor group is selected based on the comparison of the motion vectors of two triangular prediction units. The 2nd weighting factor group is used when the reference pictures of the two triangular prediction units are different from each other or their motion vector difference is larger than 16 pixels. Otherwise, the 1st weighting factor group is used. An example is shown in Figure 15.

[00174] 2.2.3.3. Motion vector storage

The motion vectors (Mv1 and Mv2 in Figure 16) of the triangular prediction units are stored in 4×4 grids. For each 4×4 grid, either uni-prediction or bi-prediction motion vector is stored depending on the position of the 4×4 grid in the CU. As shown in Figure 16, uni-prediction motion vector, either Mv1 or Mv2, is stored for the 4×4 grid located in the **non-weighted area** (that is, not located at the diagonal edge). On the other hand, a bi-prediction motion vector is stored for the 4×4 grid located in the **weighted area**. The bi-prediction motion vector is derived from Mv1 and Mv2 according to the following rules:

[00175] In the case that Mv1 and Mv2 have motion vector from different directions (L0 or L1), Mv1 and Mv2 are simply combined to form the bi-prediction motion vector.

[00176] In the case that both Mv1 and Mv2 are from the same L0 (or L1) direction,

[00177] If the reference picture of Mv2 is the same as a picture in the L1 (or L0) reference picture list, Mv2 is scaled to the picture. Mv1 and the scaled Mv2 are combined to form the bi-prediction motion vector.

[00178] If the reference picture of Mv1 is the same as a picture in the L1 (or L0) reference picture list, Mv1 is scaled to the picture. The scaled Mv1 and Mv2 are combined to form the bi-prediction motion vector.

[00179] Otherwise, only Mv1 is stored for the weighted area.

[00180] 2.2.3.4. Signaling of triangular prediction mode (TPM)

[00181] One bit flag to indicate whether TPM is used may be firstly signaled. Afterwards, the indications of two splitting patterns (as depicted in Figure 13), and selected merge indices for each of the two partitions are further signaled.

[00182] 2.2.3.4.1 Signaling of TPM flag

[00183] Let's denote one luma block's width and height by W and H, respectively. If $W \cdot H < 64$, triangular prediction mode is disabled.

[00184] When one block is coded with affine mode, triangular prediction mode is also disabled.

[00185] When one block is coded with merge mode, one bit flag may be signaled to indicate whether the triangular prediction mode is enabled or disabled for the block.

[00186] The flag is coded with 3 contexts, based on the following equation:

[00187] $\text{Ctx index} = ((\text{left block } L \text{ available \&\& } L \text{ is coded with TPM? }) 1 : 0)$

[00188] $+ ((\text{Above block } A \text{ available \&\& } A \text{ is coded with TPM? }) 1 : 0);$

[00189] 2.2.3.4.2. Signaling of an indication of two splitting patterns (as depicted in Figure 13), and selected merge indices for each of the two partitions

[00190] It is noted that splitting patterns, merge indices of two partitions are jointly coded. It is restricted that the two partitions couldn't use the same reference index. Therefore, there are 2 (splitting patterns) * N (maximum number of merge candidates) * (N-1) possibilities wherein N is

set to 5. One indication is coded and the mapping between the partition type, two merge indices are derived from the array defined below:

```
[00191] const uint8_t g_TriangleCombination[TRIANGLE_MAX_NUM_CANDS][3] = {
[00192] { 0, 1, 0 }, { 1, 0, 1 }, { 1, 0, 2 }, { 0, 0, 1 }, { 0, 2, 0 },
5 [00193] { 1, 0, 3 }, { 1, 0, 4 }, { 1, 1, 0 }, { 0, 3, 0 }, { 0, 4, 0 },
[00194] { 0, 0, 2 }, { 0, 1, 2 }, { 1, 1, 2 }, { 0, 0, 4 }, { 0, 0, 3 },
[00195] { 0, 1, 3 }, { 0, 1, 4 }, { 1, 1, 4 }, { 1, 1, 3 }, { 1, 2, 1 },
[00196] { 1, 2, 0 }, { 0, 2, 1 }, { 0, 4, 3 }, { 1, 3, 0 }, { 1, 3, 2 },
[00197] { 1, 3, 4 }, { 1, 4, 0 }, { 1, 3, 1 }, { 1, 2, 3 }, { 1, 4, 1 },
10 [00198] { 0, 4, 1 }, { 0, 2, 3 }, { 1, 4, 2 }, { 0, 3, 2 }, { 1, 4, 3 },
[00199] { 0, 3, 1 }, { 0, 2, 4 }, { 1, 2, 4 }, { 0, 4, 2 }, { 0, 3, 4 } };
```

[00200] Partition type (45 degree or 135 degree) = g_TriangleCombination[signaled indication][0];

[00201] Merge index of candidate A = g_TriangleCombination[signaled indication][1];

[00202] Merge index of candidate B = g_TriangleCombination[signaled indication][2];

15 [00203] Once the two motion candidates A and B are derived, the two partitions' (PU1 and PU2) motion information could be set either from A or B. Whether PU1 uses the motion information of merge candidate A or B is dependent on the prediction directions of the two motion candidates. Table 1 Table 1 shows the relationship between two derived motion candidates A and B, with the two partitions.

20 [00204] Table 1: Derivation of partitions' motion information from derived two merge candidates (A, B)

Prediction direction of A	Prediction direction of B	PU1's motion information	PU2's motion information
L0	L0	A (L0)	B (L0)
L1	L1	B (L1)	A (L1)
L0	L1	A (L0)	B (L1)
L1	L0	B (L0)	A (L1)

[00205] 2.2.3.4.3. Entropy coding of the indication (denoted by merge_triangle_idx)

[00206] merge_triangle_idx is within the range [0, 39], inclusively. K-th order Exponential Golomb (EG) code is used for binarization of merge_triangle_idx wherein K is set to 1.

[00207] K-th order EG

[00208] To encode larger numbers in fewer bits (at the expense of using more bits to encode smaller numbers), this can be generalized using a nonnegative integer parameter k . To encode a
5 nonnegative integer x in an order- k exp-Golomb code:

[00209] Encode $\lfloor x/2^k \rfloor$ using order-0 exp-Golomb code described above, then

[00210] Encode $x \bmod 2^k$ in binary

[00211] Table 2: Exp-Golomb-k coding examples

x	$k=0$	$k=1$	$k=2$		x	$k=0$	$k=1$	$k=2$
0	1	10	100		10	0001011	001100	01110
1	010	11	101		11	0001100	001101	01111
2	011	0100	110		12	0001101	001110	0010000
3	00100	0101	111		13	0001110	001111	0010001
4	00101	0110	01000		14	0001111	00010000	0010010
5	00110	0111	01001		15	000010000	00010001	0010011
6	00111	001000	01010		16	000010001	00010010	0010100
7	0001000	001001	01011		17	000010010	00010011	0010101
8	0001001	001010	01100		18	000010011	00010100	0010110
9	0001010	001011	01101		19	000010100	00010101	0010111

10

[00212] **2.2.4. Overlapped block motion compensation**

[00213] Overlapped Block Motion Compensation (OBMC) has previously been used in H.263. In the JEM, unlike in H.263, OBMC can be switched on and off using syntax at the CU level. When OBMC is used in the JEM, the OBMC is performed for all motion compensation (MC) block
15 boundaries except the right and bottom boundaries of a CU. Moreover, it is applied for both the luma and chroma components. In the JEM, a MC block is corresponding to a coding block. When a CU is coded with sub-CU mode (includes sub-CU merge, affine and FRUC mode), each sub-block of the CU is a MC block. To process CU boundaries in a uniform fashion, OBMC is

performed at sub-block level for all MC block boundaries, where sub-block size is set equal to 4×4 , as illustrated in Figure 18.

[00214] When OBMC applies to the current sub-block, besides current motion vectors, motion vectors of four connected neighbouring sub-blocks, if available and are not identical to the current motion vector, are also used to derive prediction block for the current sub-block. These multiple prediction blocks based on multiple motion vectors are combined to generate the final prediction signal of the current sub-block.

[00215] Prediction block based on motion vectors of a neighbouring sub-block is denoted as P_N , with N indicating an index for the neighbouring *above*, *below*, *left* and *right* sub-blocks and prediction block based on motion vectors of the current sub-block is denoted as P_C . When P_N is based on the motion information of a neighbouring sub-block that contains the same motion information to the current sub-block, the OBMC is not performed from P_N . Otherwise, every sample of P_N is added to the same sample in P_C , i.e., four rows/columns of P_N are added to P_C . The weighting factors $\{1/4, 1/8, 1/16, 1/32\}$ are used for P_N and the weighting factors $\{3/4, 7/8, 15/16, 31/32\}$ are used for P_C . The exception are small MC blocks, (i.e., when height or width of the coding block is equal to 4 or a CU is coded with sub-CU mode), for which only two rows/columns of P_N are added to P_C . In this case weighting factors $\{1/4, 1/8\}$ are used for P_N and weighting factors $\{3/4, 7/8\}$ are used for P_C . For P_N generated based on motion vectors of vertically (horizontally) neighbouring sub-block, samples in the same row (column) of P_N are added to P_C with a same weighting factor.

[00216] In the JEM, for a CU with size less than or equal to 256 luma samples, a CU level flag is signalled to indicate whether OBMC is applied or not for the current CU. For the CUs with size larger than 256 luma samples or not coded with AMVP mode, OBMC is applied by default. At the encoder, when OBMC is applied for a CU, its impact is taken into account during the motion estimation stage. The prediction signal formed by OBMC using motion information of the top neighbouring block and the left neighbouring block is used to compensate the top and left boundaries of the original signal of the current CU, and then the normal motion estimation process is applied.

[00217] **2.2.5. Local illumination compensation**

[00218] Local Illumination Compensation (LIC) is based on a linear model for illumination changes, using a scaling factor a and an offset b . And it is enabled or disabled adaptively for each inter-mode coded coding unit (CU).

[00219] When LIC applies for a CU, a least square error method is employed to derive the parameters a and b by using the neighbouring samples of the current CU and their corresponding reference samples. More specifically, as illustrated in Figure 19, the subsampled (2:1 subsampling) neighbouring samples of the CU and the corresponding samples (identified by motion information of the current CU or sub-CU) in the reference picture are used. The IC parameters are derived and applied for each prediction direction separately.

[00220] When a CU is coded with merge mode, the LIC flag is copied from neighbouring blocks, in a way similar to motion information copy in merge mode; otherwise, an LIC flag is signalled for the CU to indicate whether LIC applies or not.

[00221] When LIC is enabled for a picture, additional CU level RD check is needed to determine whether LIC is applied or not for a CU. When LIC is enabled for a CU, mean-removed sum of absolute difference (MR-SAD) and mean-removed sum of absolute Hadamard-transformed difference (MR-SATD) are used, instead of SAD and SATD, for integer pel motion search and fractional pel motion search, respectively.

[00222] To reduce the encoding complexity, the following encoding scheme is applied in the JEM.

[00223] LIC is disabled for the entire picture when there is no obvious illumination change between a current picture and its reference pictures. To identify this situation, histograms of a current picture and every reference picture of the current picture are calculated at the encoder. If the histogram difference between the current picture and every reference picture of the current picture is smaller than a given threshold, LIC is disabled for the current picture; otherwise, LIC is enabled for the current picture.

[00224] 2.2.6. Affine motion compensation prediction

[00225] In HEVC, only translation motion model is applied for motion compensation prediction (MCP). While in the real world, there are many kinds of motion, e.g. zoom in/out, rotation, perspective motions and the other irregular motions. In the JEM, a simplified affine transform motion compensation prediction is applied. As shown in Figure 20, the affine motion field of the block is described by two control point motion vectors.

[00226] The motion vector field (MVF) of a block is described by the following equation:

$$[00227] \begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w} x - \frac{(v_{1y} - v_{0y})}{w} y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w} x + \frac{(v_{1x} - v_{0x})}{w} y + v_{0y} \end{cases} \quad (1-a)$$

[00228] For 6-parameter affine,

$$[00229] \begin{cases} v_x = \frac{v_{1x} - v_{0x}}{w} x + \frac{v_{2x} - v_{0x}}{h} y + v_{0x} \\ v_y = \frac{v_{1y} - v_{0y}}{w} x + \frac{v_{2y} - v_{0y}}{h} y + v_{0y} \end{cases} \quad (1-b)$$

[00230] Where (v_{0x}, v_{0y}) is motion vector of the top-left corner control point, and (v_{1x}, v_{1y}) is motion vector of the top-right corner control point and (v_{2x}, v_{2y}) is motion vector of the bottom-left corner control point, (x, y) represents the coordinate of a **representative point** relative to the top-left sample within current block. In VTM, the **representative point** is defined to be the center position of a sub-block, e.g., when the coordinate of the left-top corner of a sub-block relative to the top-left sample within current block is (xs, ys) , the coordinate of the representative point is defined to be $(xs+2, ys+2)$.

[00231] In order to further simplify the motion compensation prediction, sub-block based affine transform prediction is applied. The sub-block size $M \times N$ is derived as in Equation 2, where $MvPre$ is the motion vector fraction accuracy (1/16 in JEM), (v_{2x}, v_{2y}) is motion vector of the bottom-left control point, calculated according to Equation 1.

$$[00232] \begin{cases} M = clip3(4, w, \frac{w \times MvPre}{\max(abs(v_{1x} - v_{0x}), abs(v_{1y} - v_{0y}))}) \\ N = clip3(4, h, \frac{h \times MvPre}{\max(abs(v_{2x} - v_{0x}), abs(v_{2y} - v_{0y}))}) \end{cases} \quad (2)$$

[00233] After derived by Equation 2, M and N should be adjusted downward if necessary to make it a divisor of w and h, respectively.

[00234] To derive motion vector of each $M \times N$ sub-block, the motion vector of the center sample of each sub-block, as shown in Figure 21, is calculated according to Equation 1, and rounded to 1/16 fraction accuracy. Then the motion compensation interpolation filters mentioned elsewhere herein are applied to generate the prediction of each sub-block with derived motion vector.

[00235] After MCP, the high accuracy motion vector of each sub-block is rounded and saved as the same accuracy as the normal motion vector.

[00236] **2.2.6.1. AF_INTER mode**

[00237] In the JEM, there are two affine motion modes: AF_INTER mode and AF_MERGE mode. For CUs with both width and height larger than 8, AF_INTER mode can be applied. An affine flag in CU level is signalled in the bitstream to indicate whether AF_INTER mode is used. In this mode, a candidate list with motion vector pair $\{(v_0, v_1) | v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}\}$ is constructed using the neighbour blocks. As shown in Figure 23, v_0 is selected from the motion vectors of the block A, B or C. The motion vector from the neighbour block is scaled according to the reference list and the relationship among the POC of the reference for the neighbour block, the POC of the reference for the current CU and the POC of the current CU. And the approach to select v_1 from the neighbour block D and E is similar. If the number of candidate list is smaller than 2, the list is padded by the motion vector pair composed by duplicating each of the AMVP candidates. When the candidate list is larger than 2, the candidates are firstly sorted according to the consistency of the neighbouring motion vectors (similarity of the two motion vectors in a pair candidate) and only the first two candidates are kept. An RD cost check is used to determine which motion vector pair candidate is selected as the control point motion vector prediction (CPMVP) of the current CU. And an index indicating the position of the CPMVP in the candidate list is signalled in the bitstream. After the CPMVP of the current affine CU is determined, affine motion estimation is applied and the control point motion vector (CPMV) is found. Then the difference of the CPMV and the CPMVP is signalled in the bitstream.

[00238] In AF_INTER mode, when 4/6 parameter affine mode is used, 2/3 control points are required, and therefore 2/3 MVD needs to be coded for these control points, as shown in Figure 22. It is proposed to derive the MV as follows, i.e., mvd_1 and mvd_2 are predicted from mvd_0 .

$$[00239] \quad mv_0 = \overline{mv}_0 + mvd_0$$

$$[00240] \quad mv_1 = \overline{mv}_1 + mvd_1 + mvd_0$$

$$[00241] \quad mv_2 = \overline{mv}_2 + mvd_2 + mvd_0$$

[00242] Wherein $\overline{mv}_i$, mvd_i and mv_i are the predicted motion vector, motion vector difference and motion vector of the top-left pixel ($i = 0$), top-right pixel ($i = 1$) or left-bottom pixel ($i = 2$) respectively, as shown in Figure 22 (b). Please note that the addition of two motion vectors (e.g., $mv_A(x_A, y_A)$ and $mv_B(x_B, y_B)$) is equal to summation of two components separately, that is, $newMV = mv_A + mv_B$ and the two components of newMV is set to $(x_A + x_B)$ and $(y_A + y_B)$, respectively.

[00243] 2.2.6.2. AF_MERGE mode

[00244] When a CU is applied in AF_MERGE mode, it gets the first block coded with affine mode from the valid neighbour reconstructed blocks. And the selection order for the candidate block is from left, above, above right, left bottom to above left as shown in Figure 24 a. If the neighbour left bottom block A is coded in affine mode as shown in Figure 24 b, the motion vectors v_2 , v_3 and v_4 of the top left corner, above right corner and left bottom corner of the CU which contains the block A are derived. And the motion vector v_0 of the top left corner on the current CU is calculated according to v_2 , v_3 and v_4 . Secondly, the motion vector v_1 of the above right of the current CU is calculated.

[00245] After the CPMV of the current CU v_0 and v_1 are derived, according to the simplified affine motion model Equation 1, the MVF of the current CU is generated. In order to identify whether the current CU is coded with AF_MERGE mode, an affine flag is signalled in the bitstream when there is at least one neighbour block is coded in affine mode.

[00246] An affine merge candidate list is constructed with following steps:

[00247] Insert inherited affine candidates

[00248] Inherited affine candidate means that the candidate is derived from the affine motion model of its valid neighbor affine coded block. In the common base, as shown in Figure 25, the scan order for the candidate positions is: A1, B1, B0, A0 and B2.

[00249] After a candidate is derived, full pruning process is performed to check whether same candidate has been inserted into the list. If a same candidate exists, the derived candidate is discarded.

[00250] Insert constructed affine candidates

[00251] If the number of candidates in affine merge candidate list is less than MaxNumAffineCand (set to 5 in this contribution), constructed affine candidates are inserted into the candidate list.

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

[00252] The motion information for the control points is derived firstly from the specified spatial neighbors and temporal neighbor shown in Figure 25. CP_k (k=1, 2, 3, 4) represents the k-th control point. A0, A1, A2, B0, B1, B2 and B3 are spatial positions for predicting CP_k (k=1, 2, 3); T is temporal position for predicting CP4.

[00253] The coordinates of CP1, CP2, CP3 and CP4 is (0, 0), (W, 0), (H, 0) and (W, H), respectively, where W and H are the width and height of current block.

[00254] The motion information of each control point is obtained according to the following priority order:

- 5 [00255] For CP1, the checking priority is B2->B3->A2. B2 is used if it is available. Otherwise, if B2 is available, B3 is used. If both B2 and B3 are unavailable, A2 is used. If all the three candidates are unavailable, the motion information of CP1 cannot be obtained.

[00256] For CP2, the checking priority is B1->B0.

[00257] For CP3, the checking priority is A1->A0.

- 10 [00258] For CP4, T is used.

[00259] Secondly, the combinations of controls points are used to construct an affine merge candidate.

- [00260] Motion information of three control points are needed to construct a 6-parameter affine candidate. The three control points can be selected from one of the following four combinations
15 ({CP1, CP2, CP4}, {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4}). Combinations {CP1, CP2, CP3}, {CP2, CP3, CP4}, {CP1, CP3, CP4} will be converted to a 6-parameter motion model represented by top-left, top-right and bottom-left control points.

- [00261] Motion information of two control points are needed to construct a 4-parameter affine candidate. The two control points can be selected from one of the following six combinations
20 ({CP1, CP4}, {CP2, CP3}, {CP1, CP2}, {CP2, CP4}, {CP1, CP3}, {CP3, CP4}). Combinations {CP1, CP4}, {CP2, CP3}, {CP2, CP4}, {CP1, CP3}, {CP3, CP4} will be converted to a 4-parameter motion model represented by top-left and top-right control points.

[00262] The combinations of constructed affine candidates are inserted into to candidate list as following order:

- 25 [00263] {CP1, CP2, CP3}, {CP1, CP2, CP4}, {CP1, CP3, CP4}, {CP2, CP3, CP4}, {CP1, CP2}, {CP1, CP3}, {CP2, CP3}, {CP1, CP4}, {CP2, CP4}, {CP3, CP4}

[00264] For reference list X (X being 0 or 1) of a combination, the reference index with highest usage ratio in the control points is selected as the reference index of list X, and motion vectors point to difference reference picture will be scaled.

[00265] After a candidate is derived, full pruning process is performed to check whether same candidate has been inserted into the list. If a same candidate exists, the derived candidate is discarded.

[00266] Padding with zero motion vectors

- 5 [00267] If the number of candidates in affine merge candidate list is less than 5, zero motion vectors with zero reference indices are insert into the candidate list, until the list is full.

[00268] 2.2.7. Bi-directional optical flow

- 10 [00269] Bi-directional Optical flow (BIO) is sample-wise motion refinement which is performed on top of block-wise motion compensation for bi-prediction. The sample-level motion refinement doesn't use signalling.

[00270] Let $I^{(k)}$ be the luma value from reference k ($k=0, 1$) after block motion compensation, and $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ are horizontal and vertical components of the $I^{(k)}$ gradient, respectively. Assuming the optical flow is valid, the motion vector field (v_x, v_y) is given by an equation

$$[00271] \partial I^{(k)}/\partial t + v_x \partial I^{(k)}/\partial x + v_y \partial I^{(k)}/\partial y = 0. \quad (3)$$

- 15 [00272] Combining this optical flow equation with Hermite interpolation for the motion trajectory of each sample results in a unique third-order polynomial that matches both the function values $I^{(k)}$ and derivatives $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ at the ends. The value of this polynomial at $t=0$ is the BIO prediction:

$$[00273] pred_{BIO} = 1/2 \cdot (I^{(0)} + I^{(1)} + v_x/2 \cdot (\tau_1 \partial I^{(1)}/\partial x - \tau_0 \partial I^{(0)}/\partial x) + v_y/2 \cdot (\tau_1 \partial I^{(1)}/\partial y - \tau_0 \partial I^{(0)}/\partial y)) \quad (3)$$

- 20 [00274] Here, τ_0 and τ_1 denote the distances to the reference frames as shown on a Figure 26. Distances τ_0 and τ_1 are calculated based on POC for Ref0 and Ref1: $\tau_0 = \text{POC}(\text{current}) - \text{POC}(\text{Ref0})$, $\tau_1 = \text{POC}(\text{Ref1}) - \text{POC}(\text{current})$. If both predictions come from the same time direction (either both from the past or both from the future) then the signs are different (i.e., $\tau_0 \cdot \tau_1 < 0$). In this case, BIO is applied only if the prediction is not from the same time moment (i.e., $\tau_0 \neq \tau_1$),
- 25 both referenced regions have non-zero motion ($MVx_0, MVy_0, MVx_1, MVy_1 \neq 0$) and the block motion vectors are proportional to the time distance ($MVx_0/MVx_1 = MVy_0/MVy_1 = -\tau_0/\tau_1$).

[00275] The motion vector field (v_x, v_y) is determined by minimizing the difference Δ between values in points A and B (intersection of motion trajectory and reference frame planes on Fig. 9). Model uses only first linear term of a local Taylor expansion for Δ :

$$[00276] \Delta = (I^{(0)} - I^{(1)})_0 + v_x (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) + v_y (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \quad (5)$$

5 [00277] All values in Equation 5 depend on the sample location (i', j') , which was omitted from the notation so far. Assuming the motion is consistent in the local surrounding area, we minimize Δ inside the $(2M+1) \times (2M+1)$ square window Ω centered on the currently predicted point (i, j) , where M is equal to 2:

$$[00278] (v_x, v_y) = \underset{v_x, v_y}{\operatorname{argmin}} \sum_{[i', j'] \in \Omega} \Delta^2 [i', j'] \quad (6)$$

10 [00279] For this optimization problem, the JEM uses a simplified approach making first a minimization in the vertical direction and then in the horizontal direction. This results in

$$[00280] v_x = (s_1 + r) > m ? \operatorname{clip3} \left(-thBIO, thBIO, -\frac{s_3}{(s_1 + r)} \right) : 0 \quad (7)$$

$$[00281] v_y = (s_5 + r) > m ? \operatorname{clip3} \left(-thBIO, thBIO, -\frac{s_6 - v_x s_2 / 2}{(s_5 + r)} \right) : 0 \quad (8)$$

[00282] where,

$$15 \quad [00283] \begin{aligned} s_1 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x)^2; s_3 = \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x), \\ s_2 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial x + \tau_0 \partial I^{(0)} / \partial x) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y), \\ s_5 &= \sum_{[i', j'] \in \Omega} (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y)^2; s_6 = \sum_{[i', j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)} / \partial y + \tau_0 \partial I^{(0)} / \partial y) \end{aligned} \quad (9)$$

[00284] In order to avoid division by zero or a very small value, regularization parameters r and m are introduced in Equations 7 and 8.

$$[00285] r = 500 \cdot 4^{d-8} \quad (10)$$

$$[00286] m = 700 \cdot 4^{d-8} \quad (11)$$

20 [00287] Here d is bit depth of the video samples.

[00288] In order to keep the memory access for BIO the same as for regular bi-predictive motion compensation, all prediction and gradients values, $I^{(k)}$, $\partial I^{(k)} / \partial x$, $\partial I^{(k)} / \partial y$, are calculated only for positions inside the current block. In Equation 9, $(2M+1) \times (2M+1)$ square window Ω centered in currently predicted point on a boundary of predicted block needs to access positions outside

of the block (as shown in Figure 27 (a)). In the JEM, values of $I^{(k)}$, $\partial I^{(k)}/\partial x$, $\partial I^{(k)}/\partial y$ outside of the block are set to be equal to the nearest available value inside the block. For example, this can be implemented as padding, as shown in Figure 27 (b).

[00289] With BIO, it's possible that the motion field can be refined for each sample. To reduce the computational complexity, a block-based design of BIO is used in the JEM. The motion refinement is calculated based on 4×4 block. In the block-based BIO, the values of s_n in Equation 9 of all samples in a 4×4 block are aggregated, and then the aggregated values of s_n in are used to derived BIO motion vectors offset for the 4×4 block. More specifically, the following formula is used for block-based BIO derivation:

[00290]

$$s_{1,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega(x,y)} (\tau_1 \partial I^{(1)}/\partial x + \tau_0 \partial I^{(0)}/\partial x)^2; s_{3,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)}/\partial x + \tau_0 \partial I^{(0)}/\partial x),$$

$$[00291] s_{2,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (\tau_1 \partial I^{(1)}/\partial x + \tau_0 \partial I^{(0)}/\partial x) (\tau_1 \partial I^{(1)}/\partial y + \tau_0 \partial I^{(0)}/\partial y),$$

$$s_{5,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (\tau_1 \partial I^{(1)}/\partial y + \tau_0 \partial I^{(0)}/\partial y)^2; s_{6,b_k} = \sum_{(x,y) \in b_k} \sum_{[i',j'] \in \Omega} (I^{(1)} - I^{(0)}) (\tau_1 \partial I^{(1)}/\partial y + \tau_0 \partial I^{(0)}/\partial y)$$

(12)

[00292] where b_k denotes the set of samples belonging to the k-th 4×4 block of the predicted block. s_n in Equations 7 and 8 are replaced by $(s_{n,b_k} \gg 4)$ to derive the associated motion vector offsets.

[00293] In some cases, MV regiment of BIO might be unreliable due to noise or irregular motion. Therefore, in BIO, the magnitude of MV regiment is clipped to a threshold value th_{BIO} . The threshold value is determined based on whether the reference pictures of the current picture are all from one direction. If all the reference pictures of the current picture are from one direction, the value of the threshold is set to $12 \times 2^{14-d}$; otherwise, it is set to $12 \times 2^{13-d}$.

[00294] Gradients for BIO are calculated at the same time with motion compensation interpolation using operations consistent with HEVC motion compensation process (2D separable FIR). The input for this 2D separable FIR is the same reference frame sample as for motion compensation process and fractional position $(fracX, fracY)$ according to the fractional part of block motion vector. In case of horizontal gradient $\partial I/\partial x$ signal first interpolated vertically using *BIOfilterS* corresponding to the fractional position $fracY$ with de-scaling shift $d-8$, then gradient filter *BIOfilterG* is applied in horizontal direction corresponding to the fractional position $fracX$ with

de-scaling shift by $18-d$. In case of vertical gradient $\partial I / \partial y$ first gradient filter is applied vertically using *BIOfilterG* corresponding to the fractional position *fracY* with de-scaling shift $d-8$, then signal displacement is performed using *BIOfilterS* in horizontal direction corresponding to the fractional position *fracX* with de-scaling shift by $18-d$. The length of interpolation filter for gradients calculation BIOfilterG and signal displacement BIOfilterF is shorter (6-tap) in order to maintain reasonable complexity. Table 3 shows the filters used for gradients calculation for different fractional positions of block motion vector in BIO. Table 4 shows the interpolation filters used for prediction signal generation in BIO.

[00295] Table 1: Filters for gradients calculation in BIO

Fractional pel position	Interpolation filter for gradient(BIOfilterG)
0	{ 8, -39, -3, 46, -17, 5}
1/16	{ 8, -32, -13, 50, -18, 5}
1/8	{ 7, -27, -20, 54, -19, 5}
3/16	{ 6, -21, -29, 57, -18, 5}
1/4	{ 4, -17, -36, 60, -15, 4}
5/16	{ 3, -9, -44, 61, -15, 4}
3/8	{ 1, -4, -48, 61, -13, 3}
7/16	{ 0, 1, -54, 60, -9, 2}
1/2	{ -1, 4, -57, 57, -4, 1}

[00296] Table 2: Interpolation filters for prediction signal generation in BIO

Fractional pel position	Interpolation filter for prediction signal(BIOfilterS)
0	{ 0, 0, 64, 0, 0, 0}
1/16	{ 1, -3, 64, 4, -2, 0}
1/8	{ 1, -6, 62, 9, -3, 1}
3/16	{ 2, -8, 60, 14, -5, 1}
1/4	{ 2, -9, 57, 19, -7, 2}
5/16	{ 3, -10, 53, 24, -8, 2}
3/8	{ 3, -11, 50, 29, -9, 2}
7/16	{ 3, -11, 44, 35, -10, 3}
1/2	{ 3, -10, 35, 44, -11, 3}

[00297] In the JEM, BIO is applied to all bi-predicted blocks when the two predictions are from different reference pictures. When LIC is enabled for a CU, BIO is disabled.

[00298] In the JEM, OBMC is applied for a block after normal MC process. To reduce the computational complexity, BIO is not applied during the OBMC process. This means that BIO is only applied in the MC process for a block when using its own MV and is not applied in the MC process when the MV of a neighboring block is used during the OBMC process.

[00299] 2.2.8. Decoder-side motion vector refinement

[00300] In bi-prediction operation, for the prediction of one block region, two prediction blocks, formed using a motion vector (MV) of list0 and a MV of list1, respectively, are combined to form a single prediction signal. In the decoder-side motion vector refinement (DMVR) method, the two motion vectors of the bi-prediction are further refined by a bilateral template matching process. The bilateral template matching applied in the decoder to perform a distortion-based search between a bilateral template and the reconstruction samples in the reference pictures in order to obtain a refined MV without transmission of additional motion information.

[00301] In DMVR, a bilateral template is generated as the weighted combination (i.e. average) of the two prediction blocks, from the initial MV0 of list0 and MV1 of list1, respectively, as shown in Figure 28. The template matching operation consists of calculating cost measures between the generated template and the sample region (around the initial prediction block) in the reference picture. For each of the two reference pictures, the MV that yields the minimum template cost is considered as the updated MV of that list to replace the original one. In the JEM, nine MV candidates are searched for each list. The nine MV candidates include the original MV and 8 surrounding MVs with one luma sample offset to the original MV in either the horizontal or vertical direction, or both. Finally, the two new MVs, i.e., MV0' and MV1' as shown in Figure 28, are used for generating the final bi-prediction results. A sum of absolute differences (SAD) is used as the cost measure. Please note that when calculating the cost of a prediction block generated by one surrounding MV, the rounded MV (to integer pel) is actually used to obtain the prediction block instead of the real MV.

[00302] DMVR is applied for the merge mode of bi-prediction with one MV from a reference picture in the past and another from a reference picture in the future, without the transmission of additional syntax elements. In the JEM, when LIC, affine motion, FRUC, or sub-CU merge candidate is enabled for a CU, DMVR is not applied.

[00303] The LUT-based motion vector prediction based on the disclosed technology, which may enhance both existing and future video coding standards, is elucidated in the following examples described for various implementations. Because the LUTs allow the encoding/decoding process to be performed based on historical data (e.g., the blocks that have been processed), the LUT-based motion vector prediction can also be referred to as History-based Motion Vector Prediction (HMVP) method. In the LUT-based motion vector prediction method, one or multiple tables with motion information from previously coded blocks are maintained during the encoding/decoding process. These motion candidates stored in the LUTs are named HMVP candidates. During the encoding/decoding of one block, the associated motion information in LUTs may be added to the motion candidate lists (e.g., merge/AMVP candidate lists), and after encoding/decoding one block, LUTs may be updated. The updated LUTs are then used to code the subsequent blocks. That is, the updating of motion candidates in the LUTs are based on the encoding/decoding order of blocks. The examples below should be considered as examples to explain general concepts. These

examples should not be interpreted in a narrow way. Furthermore, these examples can be combined in any manner.

[00304] 3. Examples of Problems Solved by Embodiments

[00305] In the design of triangle partition, one block could be split to two partitions. To save memory bandwidth due to motion compensation, it is required that the two partitions shall be uni-predicted. During the motion compensation process, bi-prediction is used for the diagonal edge and uni-prediction is used for all other remaining parts. If bi-prediction is allowed for each partition, then samples located at the diagonal edge will have four MVs with two of them from one partition and two of them from another partition. Such a design has the following problems:

[00306] Only spatial neighboring blocks and temporal blocks are checked in the merge list construction process.

[00307] History-based motion vector prediction technology is disallowed for triangle partition.

[00308] How to handle triangular partition mode with intra block copy mode is unknown.

[00309] 4. Examples of embodiments

[00310] The proposed techniques could be applied to any of non-square/non-rectangular partitions, e.g., geometry partitions. In the following descriptions, we use ‘triangular partition mode’ to represent one as an example of the non-square/non-rectangular partition mode (TPM). It should be noted other kinds of partitions may be also applicable.

[00311] The detailed techniques below should be considered as examples to explain general concepts. These techniques should not be interpreted in a narrow way. Furthermore, these inventions can be combined in any manner. Denote a block size as $W \times H$.

1. Instead of always applying triangular prediction mode with the merge mode, it is proposed to enable triangular prediction mode for intra-coded blocks, or non-merge inter-coded blocks
 - a. In one example, the two partitions are predicted with different intra prediction modes.
 - b. Alternatively, furthermore, the prediction values of samples along edge boundaries may be further filtered.
 - c. In one example, the motion information (such as MV) of the two triangular partitions cannot predict each other;
 - i. Alternatively, the motion information (such as MV) of one triangular partition can be used to predict the other partition.

2. It is proposed that when one block is split into two geometry partitions, one partition may be coded with intra mode and the other may be coded with inter mode.
 - a. In one example, for the inter-coded partition, bi-prediction may be also applied.
 - b. In one example, such method may be signaled when current block is coded with merge mode, i.e., for the inter-coded partition, motion information signaled with an merge index.
 - c. Alternatively, furthermore, the prediction values of samples along edge boundaries may be further filtered.
 - d. In one example, only a subset of intra prediction modes may be applied.
 - i. The subsets may be different for the two partitions.
 - ii. The subsets may be dependent on the position of a partition.
 - iii. The subsets may be dependent on the block size and/or block shape.
3. As disclosed in Bullet 1 and Bullet 2, the merge or non-merge inter-coded block can use the current picture as the reference picture.
4. It is proposed that when coding one block with triangular partition mode, inter/intra coded information of non-adjacent spatial blocks may be also treated as predictors for coding current block.
 - a. In one example, the motion information of non-adjacent spatial blocks may be utilized.
 - b. In one example, intra prediction mode of non-adjacent spatial blocks may be utilized.
 - c. Alternatively, coded information of temporal blocks may be further utilized for coding one block with TPM.
5. It is proposed to add motion candidates derived from HMVP candidates in the merge list construction process for TPM coded blocks wherein a HMVP (History based motion vector prediction) candidate is the motion information inherited or derived from previously coded blocks.

Table maintenance for HMVP candidate storage

- a. The maintenance of HMVP candidates and/or updating of tables for storing and/or HMVP candidates may be same as HMVP for normal motion vector..
- b. In one example, the same table may be maintained for HMVP candidates storage which may be utilized for both non-TPM coded blocks and TPM coded blocks.
 - i. In one example, motion information used by TPM are not put into the HMVP storage. HMVP table(s) are not updated after encoding/decoding a TPM coded block.
- c. In one example, separate tables may be maintained to store the HMVP candidates for coding TPM coded blocks.
 - i. Tables may be maintained for HMVP candidates storage with motion information purely from TPM coded blocks.
 - ii. Two tables may be maintained to store uni- and bi-predicted HMVP candidates, respectively.

- iii. Two tables may be maintained to store motion information of the first and second partitions, respectively.
- iv. Two tables may be maintained to store motion information of List 0 and List 1 motion information, respectively.
- 5 v. Alternatively, three tables may be maintained to store uni-predicted from List 0, uni-predicted from L1 and bi-predicted HMVP candidates, respectively.
- d. After coding with TPM coded block, tables for HMVP candidates storage may not be updated.
- 10 i. Alternatively, one/multiple tables for HMVP candidates storage may be updated with the motion information of the first partition.
- ii. Alternatively, one/multiple tables for HMVP candidates storage may be updated with the motion information of the second partition.
- 15 iii. Alternatively, one/multiple tables for HMVP candidates storage may be updated with the motion information from both partitions, by adding two HMVP candidates
- iv. Alternatively, one/multiple tables for HMVP candidates storage may be updated with the motion information from both partitions, by adding one HMVP candidate (e.g., when the two partitions are predicted from two reference picture lists).
- 20 v. Whether to add one or two motion candidates to the HMVP table may depend on whether the two partitions are predicted from the same reference picture lists and/or from same reference picture.
- vi. Whether to add motion information of the first partition or the second partition to the HMVP table may depend on POC difference between the reference pictures and the current picture.
- 25 e. When separate tables are kept for coding TPM- and non-TPM coded blocks, the updating of tables may be invoked based on the mode of one block.
- 30 i. In one example, after decoding a TPM coded block, the motion information may be used to update the table for coding TPM-coded blocks.
- ii. In one example, after decoding a non-TPM coded block, the motion information may be used to update the table for coding non-TPM-coded blocks.
- 35 iii. Alternatively, furthermore, after decoding a non-TPM coded block, the motion information may be used to update the table for coding TPM-coded blocks.

Usage of HMVP candidates in TPM coded blocks

- f. A HMVP candidate may be directly added to the merge list for TPM-coded blocks.
- 40 i. Alternatively, one HMVP candidate may be utilized to derive two motion candidates, e.g., one is uni-prediction with List 0 motion information and the other is uni-prediction with List 1 motion information of the HMVP candidate.

- ii. Pruning may be applied when inserting a motion candidate derived/inherited from a HMVP candidate with other motion candidates derived/inherited from spatial or temporal blocks.
 - iii. Pruning may be applied when inserting a motion candidate derived/inherited from a HMVP candidate with a motion candidate derived/inherited from another HMVP candidate.
 - iv. Pruning may be applied when inserting a motion candidate derived/inherited from a HMVP candidate with another motion candidate derived/inherited from the same HMVP candidate.
- g. In one example, one or multiple motion candidates derived from one or multiple HMVP candidates may be added after motion candidates derived from spatial and/or temporal blocks such as block 1-7 in Figure 14. In this case, the HMVP candidate is treated in the same way as the other spatial/temporal merge candidates.
- h. Motion candidates inherited/derived from HMVP candidates may be added to the merge list right after or before a specific step in the merge list construction process which relies on motion information of spatial/temporal blocks.
 - i. Alternatively, motion candidates inherited/derived from HMVP candidates may be added to the merge list right after or before several pre-defined steps.
 - ii. In one example, one or multiple HMVP candidates with uni-prediction may be added right after all **originally uni-predicted candidates** with current design.
 - iii. In one example, one or multiple HMVP candidates with uni-prediction from L0 or bi-prediction may be added right after all **Truncated List0-predicted candidates**. Alternatively, furthermore, if a HMVP candidate is with bi-prediction, only List 0 motion information may be kept.
 - iv. In one example, one or multiple HMVP candidates with uni-prediction from L1 or bi-prediction may be added right after all **Truncated List1-predicted candidates**. Alternatively, furthermore, if a HMVP candidate is with bi-prediction, only List 1 motion information may be kept.
 - v. In one example, one or multiple HMVP candidates may be added right after all **averaged uni-prediction from List 0 or List 1 motion candidates**.
 - vi. In one example, one or multiple HMVP candidates may be added right before all **averaged uni-prediction from List 0 or List 1 motion candidates**.
- i. Motion candidates inherited/derived from HMVP candidates may be added to the merge list in an interleaved way with motion candidates inherited/derived from motion information of spatial/temporal blocks.
- j. In one example, motion candidates derived from List 0 and List 1 of each HMVP candidate may be added in order before motion candidates derived from List 0 and List 1 of another HMVP candidate.

- k. In one example, motion candidates derived from List 0 of a first set of HMVP candidates may be added in order before those derived from List 1 of a second set of HMVP candidate.
 - i. In one example, the first set and the second set are the same set, i.e., all available HMVP candidates.
 - ii. Alternatively, the first set and the second set could be different, such as the first set of HMVP candidates include HMVP candidates with uni-prediction from List 0 and bi-prediction; while the second set of HMVP candidates include HMVP candidates with uni-prediction from List 1 and bi-prediction.
 - iii. The number of HMVP candidates associated with the first and the second set may be different.
- l. HMVP candidates of list LX may be first scaled to list $L(1-X)$ and then used for list $L(1-X)$.
 - i. The scaled candidates may be inserted after all other candidates.
 - ii. The scaled candidates may be inserted after all other candidates except the temporal candidates.
- m. The checking order of HMVP candidates to be used to derive motion candidates to be added to the TPM merge list may be dependent on the indices of the HMVP candidates.
 - i. Alternatively, it may be dependent on the prediction directions.
 - ii. Alternatively, it may be dependent on the motion information.
6. How many HMVP to be checked in TPM merge list construction process may be pre-defined, e.g., 5.
 - a. The number of HMVP candidate to be checked in TPM merge list construction process may depend on block size/block shape/how many candidates are available before checking HMVP candidates.
 - b. The number of HMVP candidate to be checked in TPM merge list construction process may be signaled in SPS/VPS/PPS/picture header/slice header/tile group header/CTU rows/CTU/group of CTUs.
 - c. When partial of available HMVP candidates are to be checked in TPM merge list construction process, the selection of HMVP candidates may depend on the prediction direction/MV information/reference picture index/POC distances of reference picture and current picture of a candidate and/or candidate index.
7. Whether to and how to apply HMVP on a TPM coded block may depend on the block size (e.g., width and/or height, ratio of width and height) or block shape (e.g., square or non-square).
8. The proposed methods could also be applicable to other kinds of motion candidate lists (such as AMVP list) for geometry partitions.
9. The HMVP table size (i.e., maximum number of history motion candidates could be stored) may depend on a function of one or multiple motion candidate list sizes.

- 5
- a. In one example, the HMVP table size may depend on the regular merge list size, the TPM merge list size, the IBC merge list size, the regular inter AMVP list size, the regular inter IBC list size etc. al.
 - b. In one example, the HMVP table size may depend on those motion candidate lists which access the same HMVP table.
 - c. In one example, the function may be the function $\text{Max}(\text{inputs})$ which returns the maximum value among several inputs.
 - d. In one example, the HMVP table size may be defined as $\text{Max}(\text{regular merge list size Minus } K0, \text{ TPM merge list size Minus } K1, \text{ IBC merge list size Minus } K2)$.
- 10
- i. In one example, $K0=K1=K2 = 1$.
10. The HMVP table size (i.e., maximum number of history motion candidates could be stored) may depend on a function of one or multiple HMVP candidate numbers (denoted by numHMVP) that may be added to motion candidate lists.
- 15
- a. In one example, it may depend on the number of HMVP candidates that may be added to the motion candidate lists (e.g., the regular merge list, the TPM merge list, the IBC merge list, the regular inter AMVP list, the regular inter IBC list etc. al).
 - b. In one example, the function may be the function $\text{Max}(\text{inputs})$ which returns the maximum value among several inputs.
 - c. In one example, the HMVP table size may be defined as $\text{Max}(\text{numHMVP for regular merge list Minus } K0, \text{ numHMVP for regular AMVP list Minus } K1)$.
- 20
- ii. In one example, $K0=1$ and $K1=0$.
 - d. In one example, the HMVP table size may be defined as $\text{Max}(\text{numHMVP for regular merge list Minus } K0, \text{ numHMVP for TPM merge list Minus } K1, \text{ numHMVP for IBC merge list size Minus } K2)$.
- 25
- iii. In one example, $K0=K1=K2 = 1$.
11. The HMVP table size for one coding mode may be different from another coding mode, and it may be dependent on the motion candidate list size.
- 30
- a. In one example, the HMVP table size for the regular merge and TPM merge lists may be dependent on the regular merge list size.
 - iv. In one example, the HMVP table size may be equal to regular merge list size minus $K0$, e.g. $K0 = 1$.
 - b. In one example, the HMVP table size for the regular merge and TPM merge lists may be dependent on the regular merge list size and the TPM merge list size.
 - v. In one example, it may be defined as $\text{Max}(\text{regular merge list size Minus } K0, \text{ TPM merge list size Minus } K1)$.
- 35
- vi. In one example, the HVMP table for the normal inter-AMVP follows that used for the regular merge and TPM merge lists.
 - c. In one example, the HMVP table size for the IBC-coded blocks depend on that for the regular merge list and/or the TPM merge list.
- 40
- vii. For example, the HMVP table size for the IBC-coded blocks is equal to that for the regular merge list and/or the TPM merge list.

- viii. Alternatively, the HMVP table size for the IBC-coded merge/AMVP blocks may depend on the size of IBC merge/AMVP candidate list size.
 - d. Indications of the HMVP table sizes may be signaled in the bitstream.
 - ix. When the signaled size is equal to 0, HMVP candidates are not inserted to any motion candidate lists.
 - x. In one example, HMVP table size is not allowed to be 0.
 - xi. Alternatively, HMVP table size minus K0 is signaled, e.g. K0=1.
- 12. Indications of number of allowed HMVP candidates that may be added to motion candidate lists may be signaled in the bitstream.
 - a. In one example, for each mode (e.g., IBC or non-IBC; merge no AMVP), an indication of number of allowed HMVP candidates may be signaled independently.
 - b. Alternatively, the indications of numbers of allowed HMVP candidates may be signaled in a predictive way.
 - c. Alternatively, furthermore, the HMVP table sizes may depend on the allowed number of HMVP candidates.
 - d. For example, the number of allowed HMVP candidates that may be added to motion candidate lists is not allowed to be 0.
 - e. Alternatively, the number of allowed HMVP candidates that may be added to motion candidate lists minus K0 may be signaled, e.g. K0=1.
 - f. Alternatively, predictive coding of the number of allowed HMVP candidates that may be added to a motion candidate list (denoted by numHMVP) and the number of maximum motion candidate in the motion candidate list (denoted by numMotionList) may be applied.
 - xii. In one example, the difference between numMotionList and numHMVP may be coded.
 - xiii. In one example, the difference between (numMotionList-K0) and numHMVP may be coded, e.g., K0 = 1.
- 13. Number of allowed HMVP candidates that may be added to motion candidate lists may be dependent on the motion candidate list sizes.
 - a. In one example, it may be derived from the corresponding motion candidate list size, such as (list size – K0), e.g., K0=0 or 1.
 - b. In one example, it may be derived from the size of the corresponding HMVP table that is used by the motion candidate list, such as (table size – K0), e.g., K0=0 or 1.
- 14. A conformance bitstream shall satisfy that the number of allowed HMVP candidates that may be added to a motion candidate list is no larger than the number of maximum motion candidate in the motion candidate list.
 - a. Alternatively, a conformance bitstream shall satisfy that the number of allowed HMVP candidates that may be added to a motion candidate list is no larger than (the number of maximum motion candidate in the motion candidate list minus K0), e.g., K0=1.

[00312] FIG. 29 is a block diagram of a video processing apparatus 2900. The apparatus 2900 may be used to implement one or more of the methods described herein. The apparatus 2900 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 2900 may include one or more processors 2902, one or more memories 2904 and video processing hardware 2906. The processor(s) 2902 may be configured to implement one or more methods described in the present document. The memory (memories) 2904 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 2906 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00313] FIG. 31 is a flowchart for a method 3100 of processing a video. The method 3100 includes performing a determination (3105) that a first video block is intra-coded or non-merge inter-coded, determining (3110) a first prediction portion and a second prediction portion for the first video block based on the determination that the first video block is intra-coded or non-merge inter-coded, wherein one or both of the first prediction portion or the second prediction portion are non-rectangular and non-square portions of the first video block, and performing (3115) further processing of the first video block using the first prediction portion and the second prediction portion.

[00314] FIG. 32 is a flowchart for a method 3200 of processing a video. The method 3200 includes performing a determination (3205) that a first video block is intra-coded or non-merge inter-coded, determining (3210) a first prediction portion and a second prediction portion for the first video block, wherein one or both of the first prediction portion or the second prediction portion are non-rectangular and non-square portions of the first video block, and performing (3215) further processing of the first video block using the first prediction portion and the second prediction portion, wherein at least one prediction portion is merge or non-merge inter coded using a current image as a reference image.

[00315] FIG. 33 is a flowchart for a method 3300 of processing a video. The method 3300 includes performing a determination (3305) that a first video block is intra-coded or non-merge inter-coded, determining (3310) a first prediction portion and a second prediction portion for the first video block, wherein one or both of the first prediction portion or the second prediction portion are non-rectangular and non-square portions of the first video block, and performing (3315) further

processing of the first video block using the first prediction portion and the second prediction portion, wherein performing further processing of the first video block using the first prediction portion and the second prediction portion is based on inter or intra coded information of non-adjacent spatial blocks in relation to the first video block.

5 **[00316]** FIG. 34 is a flowchart for a method 3400 of processing a video. The method 3400 includes performing a determination (3405) that a first video block is coded with triangular portion mode (TPM) using a triangular prediction portion of the first video block and that a second video block is coded using non-TPM using a non-triangular prediction portion of the second video block, performing (3410) further processing of the first video block and the second video block using
10 stored HMVP candidates, and storing (3415) HMVP candidates associated with the first video block and the second video block.

[00317] FIG. 35 is a flowchart for a method 3500 of processing a video. The method 3500 includes performing a determination (3505) that a first video block includes prediction portions that are non-rectangular and non-square portions of the first video block, identifying (3510) an HMVP
15 candidate, adding (3515) one or more motion candidates derived from the HMVP candidate to a merge list associated with video blocks that include prediction portions that are non-rectangular and non-square, and performing (3520) further processing of the first video block using the merge list.

[00318] With reference to methods 3100, 3200, 3300, 3400, and 3500, some examples of
20 determining a candidate for encoding and their use are described in Section 4 of the present document. For example, as described in Section 4, video blocks can be processed using prediction portions that are non-square and non-rectangular.

[00319] With reference to methods 3100, 3200, 3300, 3400, and 3500, a video block may be encoded in the video bitstream in which bit efficiency may be achieved by using a bitstream
25 generation rule related to motion information prediction.

[00320] The methods can include determining image information of the first prediction portion using a first intra prediction mode; and determining image information of the second prediction portion using a second intra prediction mode, the first intra prediction mode being different than the second intra prediction mode.

[00321] The methods can include filtering prediction values related to the first prediction portion and the second prediction portion along an edge boundary between the first prediction portion and the second prediction portion.

5 [00322] The methods can include determining, by the processor, first motion information of the first prediction portion; and determining, by the processor, second motion information of the second prediction portion, wherein the first motion information is determined without using the second motion information, and the second motion information is determined without using the first motion information.

10 [00323] The methods can include determining, by the processor, first motion information of the first prediction portion; and determining, by the processor, second motion information of the second prediction portion, wherein the second motion information is determined using the first motion information.

[00324] The methods can include processing the first prediction portion with intra mode coding; and processing the second prediction portion with inter mode coding.

15 [00325] The methods can include wherein processing the second prediction portion includes applying bi-prediction.

[00326] The methods can include wherein processing the second prediction portion includes determining motion information signaled with a merge index.

20 [00327] The methods can include filtering prediction values related to the first prediction portion and the second prediction portion along an edge boundary between the first prediction portion and the second prediction portion.

[00328] The methods can include wherein a subset of intra mode coding is used.

[00329] The methods can include wherein the first prediction portion and the second prediction portion are associated with different subsets of intra mode coding.

25 [00330] The methods can include wherein the first prediction portion and the second prediction portion are associated with different subsets of intra mode coding based on positions of the first prediction portion and the second prediction portion.

[00331] The methods can include wherein the first prediction portion and the second prediction portion are associated with different subsets of intra mode coding based on one or more of: size of the first video block, or shape of the first video block.

5 [00332] The methods can include wherein the inter or intra coded information of non-adjacent spatial video blocks includes motion information of the non-adjacent spatial video blocks.

[00333] The methods can include wherein an intra prediction mode of the non-adjacent spatial video blocks is used.

[00334] The methods can include wherein coded information of temporal blocks is used.

10 [00335] The methods can include adding history based motion vector prediction (HMVP) candidates in a motion vector prediction candidate list, wherein the HMVP candidates includes motion information based on previously coded video blocks; and decoding the first video block based on the HMVP candidates.

[00336] The methods can include wherein the HMVP candidates are stored in a table, and wherein motion information used by TPM is not stored in the table.

15 [00337] The methods can include wherein the table is not updated after encoding or decoding the first video block using TPM.

[00338] The methods can include wherein the HMVP candidates are stored in a first table and a second table, the first table storing HMVP candidates for TPM, and the second table storing HMVP candidates for non-TPM.

20 [00339] The methods can include wherein one of the first table or the second table stores uni-predicted HMVP candidates, and the other stores bi-predicted HMVP candidates.

[00340] The methods can include wherein one of the first table or the second table stores motion information of the first partition portion, and the other stores motion information of the second partition portion.

25 [00341] The methods can include wherein one of the first table or the second table stores motion information of List 0, and the other stores motion information of List 1.

[00342] The methods can include wherein the first table stores uni-predicted HMVP candidates from List 0, the second table stores uni-predicted HMVP candidates from List 1, and a third table stores bi-predicted HMVP candidates.

5 [00343] The methods can include wherein tables for HMVP candidate storage is not updated based on one or both of the first prediction portion or the second prediction portion being non-rectangular and non-square geometry portions of the first video block.

[00344] The methods can include updating one or more of the first table or the second table for HMVP candidates storage with motion information of the first partition portion.

10 [00345] The methods can include updating one or more of the first table or the second table for HMVP candidates storage with motion information of the second partition portion.

[00346] The methods can include updating one or more of the first table or the second table for HMVP candidates storage with motion information of the first partition portion and motion information of the second partition portion.

15 [00347] The methods can include determining that the first partition portion and the second partition portion are predicted from two reference picture lists; and updating one or more tables for HMVP candidates storage with motion information of one HMVP candidate based on the determination that the first partition portion and the second partition portion are predicted from two reference picture lists.

20 [00348] The methods can include determining that the first partition portion and the second partition portion are predicted using a same reference picture list or a same reference picture , wherein the tables are updated based on the determination that the first partition portion and the second partition portion are predicted using the same reference picture list or the same reference picture.

25 [00349] The methods can include determining a picture order count (POC) difference between a reference picture and a current picture , wherein motion information of the first partition portion or the second partition portion are added to the one or more tables based on the POC difference.

[00350] The methods can include determining characteristics regarding coding of the first video block; and updating one or both of the first table or the second table based on the characteristics regarding coding of the first video block.

5 [00351] The methods can include wherein motion information is used to update a table associated with TPM video blocks after decoding of a TPM video block.

[00352] The methods can include wherein motion information is used to update a table associated with non-TPM video blocks after decoding of a non-TPM video block.

[00353] The methods can include wherein motion information is used to update a table associated with TPM video blocks after decoding of a non-TPM video block.

10 [00354] The methods can include determining a first motion candidate and a second motion candidate from the HMVP candidate, one of the first merge candidate or the second merge candidate being uni-prediction with List 0 motion information of the HMVP candidate, and the other being uni-prediction with List 1 motion information of the HMVP candidate.

15 [00355] The methods can include pruning based on an insertion of a motion candidate derived from the HMVP candidate with other motion candidates that are based on spatial or temporal video blocks.

[00356] The methods can include pruning based on an insertion of a motion candidate derived from the HMVP candidate with other motion candidates that are based on other HMVP candidates.

20 [00357] The methods can include pruning based on an insertion of a motion candidate derived from the HMVP candidate with other motion candidates that are based on the HMVP candidate.

[00358] The methods can include adding the motion candidates derived from the HMVP candidate associated with the first video block to a merge list after motion candidates derived from one or both of spatial or temporal video blocks.

25 [00359] The methods can include adding a motion candidate derived from the HMVP candidate to the merge list after or before a merge list construction process based on motion information of spatial or temporal video blocks.

[00360] The methods can include adding a motion candidate derived from the HMVP candidate to the merge list after or before a pre-defined step.

[00361] The methods can include adding the motion candidates derived from the HMVP candidate with uni-prediction after other originally uni-predicted candidates.

[00362] The methods can include adding the motion candidates derived from the HMVP candidate with uni-prediction from List 0 or bi-prediction after truncated List 0 predicted candidates.

5 [00363] The methods can include adding the motion candidates derived from the HMVP candidate with uni-prediction from List 1 or bi-prediction after truncated List 1 predicted candidates.

[00364] The methods can include adding the motion candidates derived from the HMVP candidate after averaged uni-prediction from List 0 or List 1 motion candidates.

10 [00365] The methods can include adding the motion candidates derived from the HMVP candidate before averaged uni-prediction from List 0 or List 1 motion candidates.

[00366] The methods can include wherein motion candidates derived from HMVP candidates are added to the merge list in accordance with interleaving with motion candidates that are based on motion information of spatial or temporal video blocks in relation to the first video block.

15 [00367] The methods can include wherein motion candidates derived from List 0 and List 1 of the HMVP candidate is added in order before motion candidates derived from List 0 and List 1 of another HMVP candidate.

[00368] The methods can include herein motion candidates derived from List 0 of a first set of HMVP candidates are added in order before motion candidates derived from List 1 of a second set of HMVP candidates.

20 [00369] The methods can include wherein the first set and the second set are the same, wherein the first set and the second set include all available HMVP candidates.

[00370] The methods can include wherein the first set and the second set are different, the first set including HMVP candidates with uni-prediction from List 0 and bi-prediction, the second set including HMVP candidates with uni-prediction from List 1 and bi-prediction.

25 [00371] The methods can include wherein the first set includes a first number of HMVP candidates, the second set includes a second number of HMVP candidates, the first number and the second number being different.

[00372] The methods can include wherein a scaled motion candidate may be derived from a HMVP candidate by scaling its motion vector of list LX to list L(1-X) and the scaled motion candidate is assigned with the scaled motion vector and list L(1-X).

5 [00373] The methods can include wherein the scaled motion candidates derived from HMVP candidates are inserted after other motion candidates derived from HMVP candidates.

[00374] The methods can include wherein the scaled motion candidates derived from HMVP candidates are inserted before temporal candidates.

[00375] The methods can include wherein a checking order of HMVP candidates to be added to the merge list is based on indices of the HMVP candidates.

10 [00376] The methods can include wherein a checking order of HMVP candidates to be added to the merge list is based on prediction directions.

[00377] The methods can include wherein a checking order of HMVP candidates to be added to the merge list is based on motion information.

15 [00378] The methods can include wherein a number of HMVP candidates to be checked in a TPM merge list construction process is pre-defined.

[00379] The methods can include wherein a number of HMVP candidates to be checked in a TPM merge list construction process is based on a block size of the first video block, a block shape of the first video block, or a number of candidates available before checking HMVP candidates.

20 [00380] The methods can include wherein a number of HMVP candidates to be checked in a TPM merge list construction process is signaled in a video parameter set (VPS), a sequence parameter set (SPS), a picture parameter set (PPS), a picture header, a tile group header, a slice header, a group of rows of coding tree units (CTUs), a CTU, or a group of CTUs.

25 [00381] The methods can include wherein selection of the HMVP candidates is based on a prediction direction, a motion vector (MV) information, a reference picture index, a POC distances of reference picture and current picture of one or both of: a candidate, or a candidate index.

[00382] The methods can include wherein application of HMVP in association with the first video block is based on a block size of the first video block or a block shape of the first video block.

[00383] The methods can include wherein the non-rectangular and non-square portions are triangular.

[00384] It will be appreciated that the disclosed techniques may be embodied in video encoders or decoders to improve compression efficiency when the coding units being compressed have shaped that are significantly different than the traditional square shaped blocks or rectangular blocks that are half-square shaped. For example, new coding tools that use long or tall coding units such as 4x32 or 32x4 sized units may benefit from the disclosed techniques.

[00385] FIG. 36 is a flowchart for a method 3600 of processing a video. The method 3600 includes determining (3605), during a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, multiple sub-portions for the current block; determining (3610) intra prediction information of the multiple sub-portions; and performing (3615) performing the conversion of the current block using the intra prediction information of the multiple sub-portions; and wherein the current block is intra-coded, and at least one of the multiple sub-portions is a non-rectangular and non-square sub-portion. It should be noted that the sub-portion may equivalent to the prediction portion above.

[00386] In some implementations, additional modifications can be performed to method 3600. For example, determining intra prediction information of the multiple sub-portions comprises: determining a first intra prediction information of a first sub-portion in the multiple sub-portions using a first intra prediction mode; determining a second intra prediction information of a second sub-portion in the multiple sub-portions using a second intra prediction mode; wherein the first intra prediction mode is different from the second intra prediction mode. The first intra prediction information is determined by using a first subset of intra prediction modes, and the second intra prediction information is determined by using second subset of intra prediction modes; wherein the first subset of intra prediction modes is different from the second subset of intra prediction modes. The subset of intra prediction modes is based on at least one of a position, a size and a shape of the first sub-portion and/or second sub-portion. Determining intra prediction information of the multiple sub-portions further comprises: filtering prediction values of samples along an edge boundary of the at least one of the multiple sub-portions. First intra prediction information of a first sub-portion of the multiple sub-portions is determined without using the second intra prediction information of a second sub-portion of the multiple sub-portions, and the second intra

prediction information of the second sub-portion is determined without using the first intra prediction information of the first sub-portion. First intra prediction information of a first sub-portion of the multiple sub-portions is determined using second intra prediction information of a second sub-portion of the multiple sub-portions, and/or the second intra prediction information of the second sub-portion is determined using the first intra prediction information of the first sub-portion.

[00387] FIG. 37 is a flowchart for a method 3700 of processing a video. The method 3700 includes determining (3705), during a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, multiple sub-portions for the current block; determining (3710) motion information of the multiple sub-portions; performing (3715) the conversion of the current block using the motion information of the multiple sub-portions; and wherein the current block is non-merge inter-coded, and at least one of the multiple sub-portions is a non-rectangular and non-square sub-portion.

[00388] In some implementations, additional modifications can be performed to method 3700.

For example, determining motion information of the multiple sub-portions comprises: determining a first motion information of a first sub-portion in the multiple sub-portions using a first intra prediction mode; determining a second motion information of a second sub-portion in the multiple sub-portions using a second intra prediction mode. Determining intra prediction of the multiple sub-portions further comprises: filtering prediction values of samples along an edge boundary of the at least one of the multiple sub-portions. The first motion information of the first sub-portion is determined without using the second motion information of the second sub-portion, and the second motion information of the second sub-portion is determined without using the first motion information of the first sub-portion. The first motion information of the first sub-portion is determined using the second motion information of the second prediction, and/or the second motion information of the second sub-portion is determined using the first motion information of the first sub-portion.

[00389] FIG. 38 is a flowchart for a method 3800 of processing a video. The method 3800 includes performing (3805) a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple sub-portions according to a splitting pattern in which a first sub-portion has a non-rectangular,

non-square shape; processing (3810) a first sub-portion with intra coding mode; and processing (3815) a second sub-portion with inter coding mode.

[00390] In some implementations, additional modifications can be performed to method 3800. For example, the second sub-portion is processed by applying bi-prediction. Signaling motion information with an merge index in response to the current block being coded with merge mode. Determining motion information of the first predication portion using a subset of intra prediction mode. The subset of intra prediction modes is based on at least one of a position of the first predication portion, a size of a first predication portion and a shape of a first predication portion.

[00391] FIG. 39 is a flowchart for a method 3900 of processing a video. The method 3900 includes performing (3905) a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple sub-portions according to a splitting pattern in which a first sub-portion has a non-rectangular, non-square shape; wherein the at least one of the multiple sub-portions is merge or non-merge inter coded and uses a current picture as a reference picture.

[00392] FIG. 40 is a flowchart for a method 4000 of processing a video. The method 4000 includes performing (4005) a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple sub-portions according to a splitting pattern in which a first prediction partition has a non-rectangular, non-square shape; and performing (4010) the conversion using inter or intra coded information of one or more non-adjacent spatial blocks.

[00393] In some implementations, additional modifications can be performed to method 4000. For example, the inter or intra coded information of one or more non-adjacent spatial blocks comprises motion information of the one or more non-adjacent spatial blocks. Performing the conversion of the current block using inter or intra coded information of one or more non-adjacent spatial blocks comprises: performing the conversion of the current block using intra prediction mode of the one or more non-adjacent spatial blocks. Coded information of temporal blocks is used.

[00394] FIG. 41 is a flowchart for a method 4100 of processing a video. The method 4100 includes determining (4105), during a conversion between a first block of visual media data and a corresponding coded representation of the visual media data, the first block being coded with geometry partition mode; determining (4110) based on at least one table storing history based

motion vector prediction (HMVP) candidates which include motion information based on previously coded blocks, motion information of at least one sub-portion of the first block; performing (4115) the conversion of the first block using the determined motion information.

[00395] In some implementations, additional modifications can be performed to method 4100.

- 5 For example, the first block being coded with geometry partition mode comprises: splitting the first block to multiple sub-portions; wherein at least one of the multiple sub-portions is a non-rectangular and non-square portion. Determining, during a conversion between a second block of the visual media data and the corresponding coded representation of the visual media data, motion information of the second block based on the same table that is used for the first block, and the
- 10 second block is not using the geometry partition mode. The motion information used by the first block is refrained from being stored in the at least one table. The at least one table is not updated after the conversion of the first block. The at least one table includes multiple tables maintained to store the HMVP candidates for blocks with the geometry partition mode. For the at least one table, the stored HMVP candidates only have motion information according to that used by blocks with
- 15 the geometry partition mode. The at least one table includes two tables maintained to store uni-predicted HMVP candidates and bi-predicted HMVP candidates, respectively. The at least one table includes two tables maintained to store motion information of a first sub-portion and a second sub-portion of the first block, respectively. The at least one table includes two tables maintained to store motion information of List 0 and List 1, respectively. The at least one table includes three
- 20 tables maintained to store uni-predicted HMVP candidates from List 0, uni-predicted HMVP candidates from List 1, and bi-predicted HMVP candidates, respectively. The at least one table is updated with the motion information of part of sub-portions of the first block. The at least one table is updated with the motion information of a first sub-portion and the motion information of a second sub-portion, the first sub-portion and the second sub-portion being sub-portion s of the
- 25 first block. The at least one table is updated with the motion information of a first sub-portion and the motion information of a second sub-portion by adding one HMVP candidate. Whether to update the at least one table by adding one or two motions candidates depends on whether the first sub-portion and the second sub-portion are using a same reference picture list or a same reference picture. Whether to update the at least one table by adding motion information of the first sub-
- 30 portion or the second sub-portion depends on a picture order count (POC) difference between a reference picture and a current picture. Ffter processing the block with the non-rectangular and

non-square portion, the motion information of the block with the non-rectangular and non-square portion is used to update the table maintained to store the HMVP candidates for block with the non-rectangular and non-square portion. After processing the block without the non-rectangular and non-square portion, the motion information of the block without the non-rectangular and non-square portion is used to update the table maintained to store the HMVP candidates for block without the non-rectangular and non-square portion. After processing the block without the non-rectangular and non-square portion, the motion information of the block without the non-rectangular and non-square portion is used to update the table maintained to store the motion vector prediction (HMVP) candidates for block with the non-rectangular and non-square portion. The geometry partition mode comprises a triangular partitioning mode. The method is applicable to other kinds of motion candidate lists.

[00396] FIG. 42 is a flowchart for a method 4200 of processing a video. The method 4200 includes determining (4205), during a conversion between a first block of visual media data and a corresponding coded representation of the visual media data, the first block being coded with geometry partition mode; determining (4210) motion information of at least one sub-portion of the first block; performing (4215) performing the conversion of the first block using the motion information of the at least one sub-portion; wherein determining motion information of at least one sub-portion comprises using at least one history based motion vector prediction (HMVP) candidate which includes motion information based on a previously coded block to construct a motion candidate list and determining the motion information from the motion candidate list.

In some implementations, additional modifications can be performed to method 4200. For example, the first block being coded with geometry partition mode comprises: splitting the first block to multiple sub-portions; wherein at least one of the multiple sub-portions is a non-rectangular and non-square portion. The at least one HMVP candidate is directly added to the motion candidate list. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: using one HMVP candidate to determine a first motion candidate and a second motion candidate, one of the first motion candidate and the second motion candidate being uni-prediction with List 0 motion information of the HMVP candidate, and the other being uni-prediction with List 1 motion information of the HMVP candidate. Performing a pruning in response to inserting a motion candidate derived from the HMVP candidate with other motion candidates that are based on spatial

or temporal video blocks. Performing a pruning in response to inserting a motion candidate derived from the HMVP candidate with other motion candidates that are based on other HMVP candidates. Performing a pruning in response to inserting a motion candidate derived from the HMVP candidate with other motion candidates that are based on the HMVP candidate. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding at least one motion candidate derived from the HMVP candidate to the motion candidate list after motion candidates derived from one or both of spatial or temporal blocks. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding at least one motion candidate derived from the HMVP candidate to the motion candidate list after or before a pre-defined step in a motion candidate list construction process based on motion information of spatial or temporal blocks. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding at least one motion candidate derived from the HMVP candidate to the motion candidate list after or before multiple pre-defined steps in a motion candidate list construction process based on motion information of spatial or temporal blocks. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding at least one motion candidate derived from the HMVP candidate with uni-prediction to the motion candidate list after all originally uni-predicted candidates. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding at least one motion candidate derived from the HMVP candidate with uni-prediction from List 0 or bi-prediction to the motion candidate list after truncated List 0 predicted candidates. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding at least one motion candidate derived from the HMVP candidate with uni-prediction from List 1 or bi-prediction to the motion candidate list after truncated List 1 predicted candidates. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding at least one motion candidate derived from the at least one HMVP candidate to the motion candidate list after averaged uni-prediction from List 0 or List 1 motion candidates. The using at least one HMVP candidate which includes motion information based on a previously

coded block to construct a motion candidate list comprises: adding at least one motion candidates derived from the at least one HMVP candidate to the motion candidate list before averaged uni-prediction from List 0 or List 1 motion candidates. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate

5 list comprises: adding at least one motion candidate derived from the HMVP candidate to the motion candidate list in accordance with interleaving with motion candidates that are based on motion information of spatial or temporal blocks in relation to the current block. The derived motion candidate from a HMVP candidate is set equal to the HMVP candidate. The using at least one HMVP candidate which includes motion information based on a previously coded block to

10 construct a motion candidate list comprises: adding at least one motion candidate derived from List 0 and List 1 of the HMVP candidate to the motion candidate list before motion candidates derived from List 0 and List 1 of another HMVP candidate. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding at least one motion candidate derived from List 0 of a first

15 set of HMVP candidates to the motion candidate list before the motion candidates derived from List 1 of a second set of HMVP candidates. The first set and the second set are the same. The first set and the second set are different, and the first set includes HMVP candidates with uni-prediction from List 0 and bi-prediction, the second set includes HMVP candidates with uni-prediction from List 1 and bi-prediction. The first set includes a first number of HMVP candidates, the second set

20 includes a second number of HMVP candidates, and the first number and the second number are different. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: scaling the HMVP candidates of list LX to list L(1-X) to generate the scaled HMVP candidates which are used for the L(1-X). The using at least one HMVP candidate which includes motion information based on

25 a previously coded block to construct a motion candidate list comprises: adding the scaled HMVP candidates after other motion candidates derived from HMVP candidates. The using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises: adding the scaled HMVP candidates after other motion candidates derived from HMVP candidates and before temporal candidates. A checking

30 order of HMVP candidates to be added to the motion candidate list is based on indices of the HMVP candidates. A checking order of HMVP candidates to be added to the motion candidate list

is based on prediction directions. A checking order of HMVP candidates to be added to the motion candidate list is based on motion information. A number of HMVP candidates to be checked in the motion candidate list construction process is pre-defined. A number of HMVP candidates to be checked in the motion candidate list construction process is based on a block size of the current block, a block shape of the current block, or a number of candidates available before checking HMVP candidates. A number of HMVP candidates to be checked in the motion candidate list construction process is signaled in a video parameter set (VPS), a sequence parameter set (SPS), a picture parameter set (PPS), a picture header, a tile group header, a slice header, a group of rows of coding tree units (CTUs), a CTU, or a group of CTUs. The HMVP candidates selected to be checked in the motion candidate list are based on at least one of a prediction direction, a motion vector (MV) information, a reference picture index, a POC distances of reference picture and current picture of a candidate and a candidate index. An application of HMVP to the current block is based on a block size of the current block or a block shape of the current block. The motion candidate list comprises a merge list. The motion candidate list is applicable to other kinds of motion candidate lists exclude the merge candidate list.

[00397] Some features preferably implemented by some embodiments are now disclosed in clause-based format.

[00398] 1. A method for processing video, comprising:

[00399] determining, during a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, multiple sub-portions for the current block;

[00400] determining intra prediction information of the multiple sub-portions;

[00401] performing the conversion of the current block using the intra prediction information of the multiple sub-portions; and

[00402] wherein the current block is intra-coded, and at least one of the multiple sub-portions is a non-rectangular and non-square sub-portion.

[00403] 2. The method of claim 1, wherein determining intra prediction information of the multiple sub-portions comprises:

[00404] determining a first intra prediction information of a first sub-portion in the multiple sub-portions using a first intra prediction mode;

[00405] determining a second intra prediction information of a second sub-portion in the multiple sub-portions using a second intra prediction mode;

5 [00406] wherein the first intra prediction mode is different from the second intra prediction mode.

[00407] 3. The method of claim 2, wherein the first intra prediction information is determined by using a first subset of intra prediction modes, and the second intra prediction information is determined by using second subset of intra prediction modes;

10 [00408] wherein the first subset of intra prediction modes is different from the second subset of intra prediction modes.

[00409] 4. The method of claim 3, wherein the subset of intra prediction modes is based on at least one of a position, a size and a shape of the first sub-portion and/or second sub-portion.

[00410] 5. The method of anyone of claims 1 to 4, wherein determining intra prediction information of the multiple sub-portions further comprises:

15 [00411] filtering prediction values of samples along an edge boundary of the at least one of the multiple sub-portions.

[00412] 6. The method of anyone of claims 1 to 5, wherein first intra prediction information of a first sub-portion of the multiple sub-portions is determined without using the second intra prediction information of a second sub-portion of the multiple sub-portions, and the second intra prediction information of the second sub-portion is determined without using the first intra prediction information of the first sub-portion.

20 [00413] 7. The method of anyone of claims 1 to 5, wherein first intra prediction information of a first sub-portion of the multiple sub-portions is determined using second intra prediction information of a second sub-portion of the multiple sub-portions, and/or the second intra prediction information of the second sub-portion is determined using the first intra prediction information of the first sub-portion.

25 [00414] 8. A method for processing video, comprising:

[00415] determining, during a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, multiple sub-portions for the current block;

[00416] determining motion information of the multiple sub-portions;

5 [00417] performing the conversion of the current block using the motion information of the multiple sub-portions; and

[00418] wherein the current block is non-merge inter-coded, and at least one of the multiple sub-portions is a non-rectangular and non-square sub-portion.

10 [00419] 9. The method of claim 8, wherein determining motion information of the multiple sub-portions comprises:

[00420] determining a first motion information of a first sub-portion in the multiple sub-portions using a first intra prediction mode;

[00421] determining a second motion information of a second sub-portion in the multiple sub-portions using a second intra prediction mode.

15 [00422] 10. The method of anyone of claims 8-9, wherein determining intra prediction of the multiple sub-portions further comprises:

[00423] filtering prediction values of samples along an edge boundary of the at least one of the multiple sub-portions.

20 [00424] 11. The method of anyone of claims 8-9, wherein the first motion information of the first sub-portion is determined without using the second motion information of the second sub-portion, and the second motion information of the second sub-portion is determined without using the first motion information of the first sub-portion.

25 [00425] 12. The method of anyone of claims 8-9, wherein the first motion information of the first sub-portion is determined using the second motion information of the second prediction, and/or the second motion information of the second sub-portion is determined using the first motion information of the first sub-portion.

[00426] 13. A method of video processing, comprising:

[00427] performing a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple sub-portions according to a splitting pattern in which a first sub-portion has a non-rectangular, non-square shape;

5 [00428] processing a first sub-portion with intra coding mode; and

[00429] processing a second sub-portion with inter coding mode.

[00430] 14. The method of claim 13, wherein the second sub-portion is processed by applying bi-prediction.

[00431] 15. The method of anyone of claims 13-14, further comprises:

10 [00432] signaling motion information with an merge index in response to the current block being coded with merge mode.

[00433] 16. The method of anyone of claims 13-15, further comprises:

[00434] determining motion information of the first predication portion using a subset of intra prediction mode.

15 [00435] 17. The method of claim 16, wherein the subset of intra prediction modes is based on at least one of a position of the first predication portion, a size of a first predication portion and a shape of a first predication portion.

[00436] 18. A method of video processing, comprising:

20 [00437] performing a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple sub-portions according to a splitting pattern in which a first sub-portion has a non-rectangular, non-square shape;

[00438] wherein the at least one of the multiple sub-portions is merge or non-merge inter coded and uses a current picture as a reference picture.

25 [00439] 19. A method of video processing, comprising:

[00440] performing a conversion between a current block of visual media data and a corresponding coded representation of the visual media data, wherein the current block is partitioned into multiple

sub-portions according to a splitting pattern in which a first prediction partition has a non-rectangular, non-square shape;

[00441] and performing the conversion using inter or intra coded information of one or more non-adjacent spatial blocks.

- 5 [00442] 20. The method of claim 19, wherein the inter or intra coded information of one or more non-adjacent spatial blocks comprises motion information of the one or more non-adjacent spatial blocks.

[00443] 21. The method of claim 19, wherein performing the conversion of the current block using inter or intra coded information of one or more non-adjacent spatial blocks comprises:

- 10 [00444] performing the conversion of the current block using intra prediction mode of the one or more non-adjacent spatial blocks.

[00445] 22. The method of claim 19, wherein coded information of temporal blocks is used.

[00446] 23. A video processing apparatus comprising a processor configured to implement a method recited in anyone of claims 1 to 22.

- 15 [00447] 24. The apparatus of claim 23, wherein the apparatus is a video encoder.

[00448] 25. The apparatus of claim 23, wherein the apparatus is a video decoder.

[00449] 26. A computer readable recoding media on which a program comprising code is recorded, the program is for a processor to carry out a method recited in anyone of claims 1 to 22.

- 20 [00450] Some features preferably implemented by some embodiments are now disclosed in clause-based format.

[00451] 1. A method for processing video, comprising:

[00452] determining, during a conversion between a first block of visual media data and a corresponding coded representation of the visual media data, the first block being coded with geometry partition mode;

- 25 [00453] determining, based on at least one table storing history based motion vector prediction (HMVP) candidates which include motion information based on previously coded blocks, motion information of at least one sub-portion of the first block;

[00454] performing the conversion of the first block using the determined motion information.

[00455] 2 A method of claim 1, wherein the first block being coded with geometry partition mode comprises:

[00456] splitting the first block to multiple sub-portions;

[00457] wherein at least one of the multiple sub-portions is a non-rectangular and non-square portion.

[00458] 3. The method of claim 1, further comprising:

[00459] determining, during a conversion between a second block of the visual media data and the corresponding coded representation of the visual media data, motion information of the second block based on the same table that is used for the first block,

[00460] and the second block is not using the geometry partition mode.

[00461] 4. The method of anyone of claims 1-3, wherein the motion information used by the first block is refrained from being stored in the at least one table.

[00462] 5. The method of anyone of claims 1-4, wherein the at least one table is not updated after the conversion of the first block.

[00463] 6. The method of anyone of claims 1-5, wherein the at least one table includes multiple tables maintained to store the HMVP candidates for blocks with the geometry partition mode.

[00464] 7. The method of claim 6, wherein for the at least one table, the stored HMVP candidates only have motion information according to that used by blocks with the geometry partition mode.

[00465] 8. The method of anyone of claims 1-7, wherein the at least one table includes two tables maintained to store uni-predicted HMVP candidates and bi-predicted HMVP candidates, respectively.

[00466] 9. The method of anyone of claims 1-8, wherein the at least one table includes two tables maintained to store motion information of a first sub-portion and a second sub-portion of the first block, respectively.

[00467] 10. The method of anyone of claims 1-9, wherein the at least one table includes two tables maintained to store motion information of List 0 and List 1, respectively.

[00468] 11. The method of anyone of claims 1-10, wherein the at least one table includes three tables maintained to store uni-predicted HMVP candidates from List 0, uni-predicted HMVP candidates from List 1, and bi-predicted HMVP candidates, respectively.

5 [00469] 12. The method of anyone of claims 1-11, wherein the at least one table is updated with the motion information of part of sub-portions of the first block.

[00470] 13. The method of anyone of claims 1-12, wherein the at least one table is updated with the motion information of a first sub-portion and the motion information of a second sub-portion, the first sub-portion and the second sub-portion being sub-portion s of the first block.

10 [00471] 14. The method of claim 13, wherein the at least one table is updated with the motion information of a first sub-portion and the motion information of a second sub-portion by adding one HMVP candidate.

[00472] 15. The method of claim 14, wherein whether to update the at least one table by adding one or two motions candidates depends on whether the first sub-portion and the second sub-portion are using a same reference picture list or a same reference picture.

15 [00473] 16. The method of claim 14, wherein whether to update the at least one table by adding motion information of the first sub-portion or the second sub-portion depends on a picture order count (POC) difference between a reference picture and a current picture.

[00474] 17. The method of any one of claims 1 to 16, wherein the method further comprises:

20 [00475] after processing the block with the non-rectangular and non-square portion, the motion information of the block with the non-rectangular and non-square portion is used to update the table maintained to store the HMVP candidates for block with the non-rectangular and non-square portion.

[00476] 18. The method of claim 1 to 16, the method further comprises:

25 [00477] after processing the block without the non-rectangular and non-square portion, the motion information of the block without the non-rectangular and non-square portion is used to update the table maintained to store the HMVP candidates for block without the non-rectangular and non-square portion.

[00478] 19. The method of claim 1 to 16, the method further comprises:

[00479] after processing the block without the non-rectangular and non-square portion, the motion information of the block without the non-rectangular and non-square portion is used to update the table maintained to store the motion vector prediction (HMVP) candidates for block with the non-rectangular and non-square portion.

5 [00480] 20. The method of anyone of claims 1-19, wherein the geometry partition mode comprises a triangular partitioning mode.

[00481] 21. The method of anyone of claims 1-20, wherein the method is applicable to other kinds of motion candidate lists.

10 [00482] 22. A video processing apparatus comprising a processor configured to implement a method recited in anyone of claims 1 to 21.

[00483] 23. The apparatus of claim 22, wherein the apparatus is a video encoder.

[00484] 24. The apparatus of claim 22, wherein the apparatus is a video decoder.

[00485] 25. A computer readable recoding media on which a program comprising code is recorded, the program is for a processor to carry out a method recited in anyone of claims 1 to 21.

15 [00486] Some features preferably implemented by some embodiments are now disclosed in clause-based format.

[00487] 1. A method for processing video, comprising:

20 [00488] determining, during a conversion between a first block of visual media data and a corresponding coded representation of the visual media data, the first block being coded with geometry partition mode;

[00489] determining motion information of at least one sub-portion of the first block;

[00490] performing the conversion of the first block using the motion information of the at least one sub-portion;

25 [00491] wherein determining motion information of at least one sub-portion comprises using at least one history based motion vector prediction (HMVP) candidate which includes motion information based on a previously coded block to construct a motion candidate list and determining the motion information from the motion candidate list.

[00492] 2 The method of claim 1, wherein the first block being coded with geometry partition mode comprises:

[00493] splitting the first block to multiple sub-portions;

[00494] wherein at least one of the multiple sub-portions is a non-rectangular and non-square portion.

[00495] 3. The method of claim 1, wherein

[00496] the at least one HMVP candidate is directly added to the motion candidate list.

[00497] 4. The method of anyone of claims 1-3, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00498] using one HMVP candidate to determine a first motion candidate and a second motion candidate, one of the first motion candidate and the second motion candidate being uni-prediction with List 0 motion information of the HMVP candidate, and the other being uni-prediction with List 1 motion information of the HMVP candidate.

[00499] 5. The method of anyone of claims 1-4, further comprising:

[00500] performing a pruning in response to inserting a motion candidate derived from the HMVP candidate with other motion candidates that are based on spatial or temporal video blocks.

[00501] 6. The method of anyone of claims 1- 5, further comprising:

[00502] performing a pruning in response to inserting a motion candidate derived from the HMVP candidate with other motion candidates that are based on other HMVP candidates.

[00503] 7. The method of anyone of claims 1-6, further comprising:

[00504] performing a pruning in response to inserting a motion candidate derived from the HMVP candidate with other motion candidates that are based on the HMVP candidate.

[00505] 8. The method of anyone of claims 1-7, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00506] adding at least one motion candidate derived from the HMVP candidate to the motion candidate list after motion candidates derived from one or both of spatial or temporal blocks.

[00507] 9. The method of anyone of claims 1-8, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00508] adding at least one motion candidate derived from the HMVP candidate to the motion candidate list after or before a pre-defined step in a motion candidate list construction process based on motion information of spatial or temporal blocks.

[00509] 10. The method of anyone of claims 1-9, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises::

[00510] adding at least one motion candidate derived from the HMVP candidate to the motion candidate list after or before multiple pre-defined steps in a motion candidate list construction process based on motion information of spatial or temporal blocks.

[00511] 11. The method of anyone of claims 1-10, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00512] adding at least one motion candidate derived from the HMVP candidate with uni-prediction to the motion candidate list after all originally uni-predicted candidates.

[00513] 12. The method of anyone of claims 1-11, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00514] adding at least one motion candidate derived from the HMVP candidate with uni-prediction from List 0 or bi-prediction to the motion candidate list after truncated List 0 predicted candidates.

[00515] 13. The method of anyone of claims 1-12, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00516] adding at least one motion candidate derived from the HMVP candidate with uni-prediction from List 1 or bi-prediction to the motion candidate list after truncated List 1 predicted candidates.

[00517] 14. The method of anyone of claims 1-13, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00518] adding at least one motion candidate derived from the at least one HMVP candidate to the motion candidate list after averaged uni-prediction from List 0 or List 1 motion candidates.

[00519] 15. The method of anyone of claims 1-14, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00520] adding at least one motion candidates derived from the at least one HMVP candidate to the motion candidate list before averaged uni-prediction from List 0 or List 1 motion candidates.

[00521] 16. The method of anyone of claims 1-15, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00522] adding at least one motion candidate derived from the HMVP candidate to the motion candidate list in accordance with interleaving with motion candidates that are based on motion information of spatial or temporal blocks in relation to the current block.

[00523] 17. The method of anyone of claims 8-16, wherein

[00524] the derived motion candidate from a HMVP candidate is set equal to the HMVP candidate

[00525] 18. The method of anyone of claims 1-17, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00526] adding at least one motion candidate derived from List 0 and List 1 of the HMVP candidate to the motion candidate list before motion candidates derived from List 0 and List 1 of another HMVP candidate.

[00527] 19. The method of anyone of claims 1-18, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00528] adding at least one motion candidate derived from List 0 of a first set of HMVP candidates to the motion candidate list before the motion candidates derived from List 1 of a second set of HMVP candidates.

[00529] 20. The method of claim 19, wherein the first set and the second set are the same.

[00530] 21. The method of claim 19, wherein the first set and the second set are different, and the first set includes HMVP candidates with uni-prediction from List 0 and bi-prediction, the second set includes HMVP candidates with uni-prediction from List 1 and bi-prediction.

[00531] 22. The method of claim 19, wherein the first set includes a first number of HMVP candidates, the second set includes a second number of HMVP candidates, and the first number and the second number are different.

[00532] 23. The method of anyone of claims 1-22, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00533] scaling the HMVP candidates of list LX to list L(1-X) to generate the scaled HMVP candidates which are used for the L(1-X).

[00534] 24. The method of claim 23, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00535] adding the scaled HMVP candidates after other motion candidates derived from HMVP candidates.

[00536] 25. The method of claim 23, wherein the using at least one HMVP candidate which includes motion information based on a previously coded block to construct a motion candidate list comprises:

[00537] adding the scaled HMVP candidates after other motion candidates derived from HMVP candidates and before temporal candidates.

[00538] 26. The method of anyone of claims 1-25, wherein a checking order of HMVP candidates to be added to the motion candidate list is based on indices of the HMVP candidates.

[00539] 27. The method of anyone of claims 1-25, wherein a checking order of HMVP candidates to be added to the motion candidate list is based on prediction directions.

5 [00540] 28. The method of anyone of claims 1-25, wherein a checking order of HMVP candidates to be added to the motion candidate list is based on motion information.

[00541] 29. The method of anyone of claims 1-28, wherein a number of HMVP candidates to be checked in the motion candidate list construction process is pre-defined.

10 [00542] 30. The method of anyone of claims 1-28, wherein a number of HMVP candidates to be checked in the motion candidate list construction process is based on a block size of the current block, a block shape of the current block, or a number of candidates available before checking HMVP candidates.

15 [00543] 31. The method of anyone of claims 1-28, wherein a number of HMVP candidates to be checked in the motion candidate list construction process is signaled in a video parameter set (VPS), a sequence parameter set (SPS), a picture parameter set (PPS), a picture header, a tile group header, a slice header, a group of rows of coding tree units (CTUs), a CTU, or a group of CTUs.

20 [00544] 32. The method of anyone of claims 1-28, wherein the HMVP candidates selected to be checked in the motion candidate list are based on at least one of a prediction direction, a motion vector (MV) information, a reference picture index, a POC distances of reference picture and current picture of a candidate and a candidate index.

[00545] 33. The method of anyone of claims 1-28, wherein an application of HMVP to the current block is based on a block size of the current block or a block shape of the current block.

[00546] 34. The method of anyone of claims 1-33, wherein the motion candidate list comprises a merge list.

25 [00547] 35. The method of anyone of claims 1-33, wherein the motion candidate list is applicable to other kinds of motion candidate lists exclude the merge candidate list.

[00548] 36. A video processing apparatus comprising a processor configured to implement a method recited in anyone of claims 1 to 35.

[00549] 37. The apparatus of claim 36, wherein the apparatus is a video encoder.

[00550] 38. The apparatus of claim 36, wherein the apparatus is a video decoder.

[00551] 39. A computer readable recoding media on which a program comprising code is recorded, the program is for a processor to carry out a method recited in anyone of claims 1 to 35.

5 [00552] 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
10 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
15 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.,
20 a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

[00553] 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,
25 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
30 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.

[00554] 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).

[00555] 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.

[00556] 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.

[00557] 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.

- 5 [00558] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this patent document.

CLAIMS

1. A method for coding video data, comprising:

5 determining, during a conversion between a first block of visual media data and a corresponding bitstream of the visual media data, the first block which is a coding unit being coded with a geometric partitioning mode which is a prediction mode for the coding unit;

 constructing, based on a table storing one or multiple motion candidates which are associated with motion information based on previously coded blocks, a motion candidate list which
10 corresponds to a merge candidate list, wherein the table includes a history based motion vector prediction (HMVP) table;

 determining a geometric partitioning mode merge index for geometric partitions of the first block being included in the corresponding bitstream, wherein the geometric partitioning mode merge index and the motion candidate list are related to motion information of the first
15 block; and

 performing the conversion of the first block using the motion information of the first block.

2. The method of claim 1, wherein the motion information used by the first block is refrained from being stored in the table.

20 3. The method of claim 1, wherein the table is not updated after the conversion of the first block.

4. The method of claim 1, further comprising:

25 determining, during a conversion between a second block of the visual media data and the corresponding bitstream of the visual media data, motion information of the second block based on the same table that is used for the first block,

 wherein the second block is not coded using the geometric partitioning mode.

5. The method of claim 1, wherein constructing the motion candidate list further comprises constructing the motion candidate list based on motion candidates derived from spatial and/or temporal blocks.

5

6. The method of claim 1, wherein the constructing, based on the table storing one or multiple motion candidates which are associated with motion information based on previously coded blocks, the motion candidate list which corresponds to the merge candidate list comprising:

adding at least one motion candidate derived from the table for the first block.

10

7. The method of claim 6, wherein the adding motion candidates derived from the table for the first block, comprising:

adding at least one motion candidate derived from the table to the motion candidate list after motion candidates derived from one or both of spatial blocks or temporal blocks.

15

8. The method of claim 6, wherein the derived at least one motion candidate from the table is set equal to corresponding at least one motion candidate in the table.

9. The method of claim 6, wherein a checking order of motion candidates in the table to be added to the motion candidate list is based on indices of motion candidates in the table.

20

10. The method of claim 1, wherein the geometric partitioning mode includes multiple partition schemes and at least one partition scheme divides the first block in to two geometric partitions, at least one of which is non-square and non-rectangular.

25

11. The method of claim 1, wherein the geometric partitioning mode comprises a triangular partitioning mode.

12. The method of claim 1, wherein the conversion comprises decoding the first block from the corresponding bitstream.

5 13. The method of claim 1, wherein the conversion comprises encoding the first block into the corresponding bitstream.

14. An apparatus for coding video data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the
10 processor to:

 determine, during a conversion between a first block of visual media data and a corresponding bitstream of the visual media data, the first block which is a coding unit being coded with a geometric partitioning mode which is a prediction mode for the coding unit;

 construct, based on a table storing one or multiple motion candidates which are
15 associated with motion information based on previously coded blocks, a motion candidate list which corresponds to a merge candidate list, wherein the table includes a history based motion vector prediction (HMVP) table;

 determine a geometric partitioning mode merge index for geometric partitions of the first block being included in the corresponding bitstream, wherein the geometric partitioning mode merge
20 index and the motion candidate list are related to motion information of the first block; and
 perform the conversion of the first block using the motion information of the first block.

15. A non-transitory computer-readable storage medium storing instructions that cause a processor to:

25 determine, during a conversion between a first block of visual media data and a corresponding bitstream of the visual media data, the first block which is a coding unit being coded with a geometric partitioning mode which is a prediction mode for the coding unit;

 construct, based on a table storing one or multiple motion candidates which are associated with motion information based on previously coded blocks, a motion candidate list

which corresponds to a merge candidate list, wherein the table includes a history based motion vector prediction (HMVP) table;

determine a geometric partitioning mode merge index for geometric partitions of the first block being included in the corresponding bitstream, wherein the geometric partitioning mode merge index and the motion candidate list are related to motion information of the first block; and
perform the conversion of the first block using the motion information of the first block.

16. A non-transitory computer-readable recording medium storing a bitstream which is generated by a method performed by a video processing apparatus, wherein the method comprises:

determining, during generating a corresponding bitstream of visual media data from a first block of the visual media data, the first block which is a coding unit being coded with a geometric partitioning mode which is a prediction mode for the coding unit;

constructing, based on a table storing one or multiple motion candidates which are associated with motion information based on previously coded blocks, a motion candidate list which corresponds to a merge candidate list, wherein the table includes a history based motion vector prediction (HMVP) table;

determining a geometric partitioning mode merge index for geometric partitions of the first block being included in the corresponding bitstream, wherein the geometric partitioning mode merge index and the motion candidate list are related to motion information of the first block; and

generating the corresponding bitstream from the first block using the motion information of the first block.