



(51) International Patent Classification:
H04N 19/46 (2014.01)

(21) International Application Number:
PCT/CN2022/088127

(22) International Filing Date:
21 April 2022 (21.04.2022)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2021/088798
21 April 2021 (21.04.2021) CN

(71) Applicants: **BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD.** [CN/CN]; Room B-0035, 2/F, No. 3 Building, No. 30, Shixing Road, Shijingshan District, Beijing 100041 (CN). **BYTEDANCE INC.** [US/US];

12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

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

(74) Agent: **SHIHUI PARTNERS**; 42/F, Tower C, Beijing Yintai Centre, No. 2 Jianguomenwai Avenue, Chaoyang District, Beijing 100022 (CN).

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,

(54) Title: METHOD, DEVICE, AND MEDIUM FOR VIDEO PROCESSING

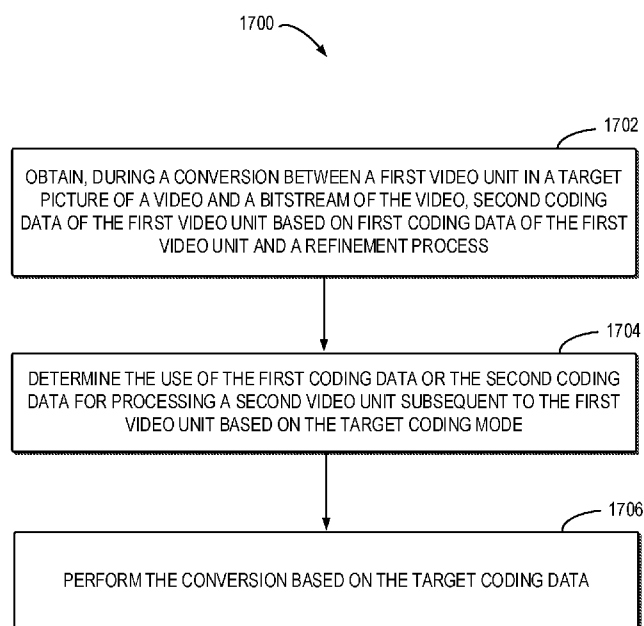


Fig. 17

(57) Abstract: A method for video processing is proposed. The method comprises: obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process (1702); determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode (1704); and performing the conversion based on the target coding data (1706). Compared with the conventional solution, the proposed method can advantageously improve the coding performance and efficiency.

DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

Published:

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

METHOD, DEVICE, AND MEDIUM FOR VIDEO PROCESSING

FIELD

[0001] Embodiments of the present disclosure relates generally to video coding techniques, and more particularly, to refinement in image or video coding.

BACKGROUND

[0002] In nowadays, digital video capabilities are being applied in various aspects of peoples' lives. Multiple types of video compression technologies, such as MPEG-2, MPEG-4, ITU-TH.263, ITU-TH.264/MPEG-4 Part 10 Advanced Video Coding (AVC), ITU-TH.265 high efficiency video coding (HEVC) standard, versatile video coding (VVC) standard, have been proposed for video encoding/decoding. However, coding efficiency of conventional video coding techniques is generally very low, which is undesirable.

SUMMARY

[0003] Embodiments of the present disclosure provide a solution for video processing.

[0004] In a first aspect, a method for video processing is proposed. The method comprises: obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and performing the conversion based on the determination. With the method in accordance with the first aspect of the present disclosure, whether to use refined coding data or original coding data before refined for processing video units, may be dependent on what coding technique is applied to the video units. Compared with the conventional solution, the proposed method can advantageously improve the coding efficiency and enable a more precise prediction.

[0005] In a second aspect, an apparatus for video processing is proposed. The apparatus comprises a processor and a non-transitory memory coupled to the processor and having

instructions stored thereon, wherein the instructions upon execution by the processor, cause the processor to: obtain, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determine the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and generate the bitstream based on the determination.

[0006] In a third aspect, a non-transitory computer-readable storage medium is proposed. The non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with the first aspect of the present disclosure.

[0007] In a fourth aspect, a non-transitory computer-readable recording medium is proposed. The non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by an apparatus for video processing, wherein the method comprises: obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determine the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and generating the bitstream based on the determination.

[0008] In a fifth aspect, a method for storing a bitstream of a video is proposed. The method comprises: obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; generating the bitstream based on the determination; and storing the bitstream in a non-transitory computer-readable recording medium.

[0009] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to

identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Through the following detailed description with reference to the accompanying drawings, the above and other objectives, features, and advantages of example embodiments of the present disclosure will become more apparent. In the example embodiments of the present disclosure, the same reference numerals usually refer to the same components.

[0011] Fig. 1 illustrates a block diagram of an example video coding system in accordance with some embodiments of the present disclosure;

[0012] Fig. 2 illustrates a block diagram of an example video encoder in accordance with some embodiments of the present disclosure;

[0013] Fig. 3 illustrates a block diagram of an example video decoder in accordance with some embodiments of the present disclosure;

[0014] Fig. 4 is a schematic diagram illustrating positions of a spatial merge candidate;

[0015] Fig. 5 is a schematic diagram illustrating candidate pairs considered for redundancy check of spatial merge candidates;

[0016] Fig. 6 illustrates motion vector scaling for temporal merge candidate;

[0017] Fig. 7 is a schematic diagram illustrating candidate positions for temporal merge candidate, C0 and C1;

[0018] Fig. 8 is a schematic diagram illustrating a merge mode with motion vector differences (MMVD) search point;

[0019] Fig. 9 is a schematic diagram illustrating the decoding side motion vector refinement;

[0020] Fig. 10 illustrates examples of the geometric partitioning mode (GPM) splits grouped by identical angles;

[0021] Fig. 11 is a schematic diagram illustrating the uni-prediction MV selection for

geometric partitioning mode;

[0022] Fig. 12 is a schematic diagram illustrating the exemplified generation of a bending weight w_0 using GPM;

[0023] Fig. 13 is a schematic diagram illustrating the top and left neighboring blocks used in combined inter and intra prediction (CIIP) weight derivation;

[0024] Fig. 14 is a schematic diagram illustrating the template matching that performs on a search area around initial MV;

[0025] Fig. 15 is a schematic diagram illustrating diamond regions in the search area;

[0026] Fig. 16 is a schematic diagram illustrating spatial neighboring blocks used to derive the spatial merge candidates;

[0027] Fig. 17 illustrates a flowchart of a method for video processing in accordance with some embodiments of the present disclosure; and

[0028] Fig. 18 illustrates a block diagram of a computing device in which various embodiments of the present disclosure can be implemented.

[0029] Throughout the drawings, the same or similar reference numerals usually refer to the same or similar elements.

DETAILED DESCRIPTION

[0030] Principle of the present disclosure will now be described with reference to some embodiments. It is to be understood that these embodiments are described only for the purpose of illustration and help those skilled in the art to understand and implement the present disclosure, without suggesting any limitation as to the scope of the disclosure. The disclosure described herein can be implemented in various manners other than the ones described below.

[0031] In the following description and claims, unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skills in the art to which this disclosure belongs.

[0032] References in the present disclosure to “one embodiment,” “an embodiment,” “an

example embodiment,” and the like indicate that the embodiment described may include a particular feature, structure, or characteristic, but it is not necessary that every embodiment includes the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an example embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0033] It shall be understood that although the terms “first” and “second” etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and similarly, a second element could be termed a first element, without departing from the scope of example embodiments. As used herein, the term “and/or” includes any and all combinations of one or more of the listed terms.

[0034] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of example embodiments. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises”, “comprising”, “has”, “having”, “includes” and/or “including”, when used herein, specify the presence of stated features, elements, and/or components etc., but do not preclude the presence or addition of one or more other features, elements, components and/or combinations thereof.

Example Environment

[0035] Fig. 1 is a block diagram that illustrates an example video coding system 100 that may utilize the techniques of this disclosure. As shown, the video coding system 100 may include a source device 110 and a destination device 120. The source device 110 can be also referred to as a video encoding device, and the destination device 120 can be also referred to as a video decoding device. In operation, the source device 110 can be configured to generate encoded video data and the destination device 120 can be configured to decode the encoded video data generated by the source device 110. The source device 110 may include a video

source 112, a video encoder 114, and an input/output (I/O) interface 116.

[0036] The video source 112 may include a source such as a video capture device. Examples of the video capture device include, but are not limited to, an interface to receive video data from a video content provider, a computer graphics system for generating video data, and/or a combination thereof.

[0037] The video data may comprise one or more pictures. The video encoder 114 encodes the video data from the video source 112 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. The I/O interface 116 may include a modulator/demodulator and/or a transmitter. The encoded video data may be transmitted directly to the destination device 120 via the I/O interface 116 through a network 130A. The encoded video data may also be stored onto a storage medium/server 130B for access by destination device 120.

[0038] The destination device 120 may include an I/O interface 126, a video decoder 124, and a display device 122. The I/O interface 126 may include a receiver and/or a modem. The I/O interface 126 may acquire encoded video data from the source device 110 or the storage medium/server 130B. The video decoder 124 may decode the encoded video data. The display device 122 may display the decoded video data to a user. The display device 122 may be integrated with the destination device 120, or may be external to the destination device 120 which is configured to interface with an external display device.

[0039] The video encoder 114 and the video decoder 124 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding (VVC) standard and other current and/or future standards.

[0040] Fig. 2 is a block diagram illustrating an example of a video encoder 200, which may be an example of the video encoder 114 in the system 100 illustrated in Fig. 1, in accordance with some embodiments of the present disclosure.

[0041] The video encoder 200 may be configured to implement any or all of the techniques of this disclosure. In the example of Fig. 2, the video encoder 200 includes a plurality of

functional components. The techniques described in this disclosure may be shared among the various components of the video encoder 200. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0042] In some embodiments, the video encoder 200 may include a partition unit 201, a predication unit 202 which may include a mode selection unit 203, a motion estimation unit 204, a motion compensation unit 205 and an intra prediction unit 206, a residual generation unit 207, a transform unit 208, a quantization unit 209, an inverse quantization unit 210, an inverse transform unit 211, a reconstruction unit 212, a buffer 213, and an entropy encoding unit 214.

[0043] In other examples, the video encoder 200 may include more, fewer, or different functional components. In an example, the predication unit 202 may include an intra block copy (IBC) unit. The IBC unit may perform predication in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0044] Furthermore, although some components, such as the motion estimation unit 204 and the motion compensation unit 205, may be integrated, but are represented in the example of Fig. 2 separately for purposes of explanation.

[0045] The partition unit 201 may partition a picture into one or more video blocks. The video encoder 200 and a video decoder 300 (which will be discussed in detail below) may support various video block sizes.

[0046] The mode selection unit 203 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra-coded or inter-coded block to the residual generation unit 207 to generate residual block data and to the reconstruction unit 212 to reconstruct the encoded block for use as a reference picture. In some examples, the mode selection unit 203 may select a combination of intra and inter predication (CIIP) mode in which the predication is based on an inter predication signal and an intra predication signal. The mode selection unit 203 may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter-predication.

[0047] To perform inter prediction on a current video block, the motion estimation unit 204 may generate motion information for the current video block by comparing one or more reference frames from buffer 213 to the current video block. The motion compensation unit

205 may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from the buffer 213 other than the picture associated with the current video block.

[0048] The motion estimation unit 204 and the motion compensation unit 205 may perform different operations for a current video block, for example, depending on whether the current video block is in an I-slice, a P-slice, or a B-slice. As used herein, an “I-slice” may refer to a portion of a picture composed of macroblocks, all of which are based upon macroblocks within the same picture. Further, as used herein, in some aspects, “P-slices” and “B-slices” may refer to portions of a picture composed of macroblocks that are not dependent on macroblocks in the same picture.

[0049] In some examples, the motion estimation unit 204 may perform uni-directional prediction for the current video block, and the motion estimation unit 204 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. The motion estimation unit 204 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. The motion estimation unit 204 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. The motion compensation unit 205 may generate the predicted video block of the current video block based on the reference video block indicated by the motion information of the current video block.

[0050] Alternatively, in other examples, the motion estimation unit 204 may perform bi-directional prediction for the current video block. The motion estimation unit 204 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. The motion estimation unit 204 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. The motion estimation unit 204 may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. The motion compensation unit 205 may generate the predicted video block of the current video

block based on the reference video blocks indicated by the motion information of the current video block.

[0051] In some examples, the motion estimation unit 204 may output a full set of motion information for decoding processing of a decoder. Alternatively, in some embodiments, the motion estimation unit 204 may signal the motion information of the current video block with reference to the motion information of another video block. For example, the motion estimation unit 204 may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

[0052] In one example, the motion estimation unit 204 may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder 300 that the current video block has the same motion information as the another video block.

[0053] In another example, the motion estimation unit 204 may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder 300 may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

[0054] As discussed above, the video encoder 200 may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by the video encoder 200 include advanced motion vector prediction (AMVP) and merge mode signaling.

[0055] The intra prediction unit 206 may perform intra prediction on the current video block. When performing intra prediction on the current video block, the intra prediction unit 206 may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

[0056] The residual generation unit 207 may generate residual data for the current video block by subtracting (e.g., indicated by the minus sign) the predicted video block (s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the

samples in the current video block.

[0057] In other examples, there may be no residual data for the current video block for the current video block, for example in a skip mode, and the residual generation unit 207 may not perform the subtracting operation.

[0058] The transform unit 208 may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

[0059] After the transform unit 208 generates a transform coefficient video block associated with the current video block, the quantization unit 209 may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[0060] The inverse quantization unit 210 and the inverse transform unit 211 may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. The reconstruction unit 212 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the predication unit 202 to produce a reconstructed video block associated with the current video block for storage in the buffer 213.

[0061] After the reconstruction unit 212 reconstructs the video block, a loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[0062] The entropy encoding unit 214 may receive data from other functional components of the video encoder 200. When the data is received, the entropy encoding unit 214 may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[0063] Fig. 3 is a block diagram illustrating an example of a video decoder 300, which may be an example of the video decoder 124 in the system 100 illustrated in Fig. 1, in accordance with some embodiments of the present disclosure.

[0064] The video decoder 300 may be configured to perform any or all of the techniques of

this disclosure. In the example of Fig. 3, the video decoder 300 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 300. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0065] In the example of Fig. 3, the video decoder 300 includes an entropy decoding unit 301, a motion compensation unit 302, an intra prediction unit 303, an inverse quantization unit 304, an inverse transform unit 305, a reconstruction unit 306, and a buffer 307. The video decoder 300 may, in some examples, perform a decoding pass that is generally reciprocal to the encoding pass as described with respect to the video encoder 200.

[0066] The entropy decoding unit 301 may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). The entropy decoding unit 301 may decode the entropy coded video data, and from the entropy decoded video data, the motion compensation unit 302 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. The motion compensation unit 302 may, for example, determine such information by performing the AMVP and merge mode. AMVP is used, including derivation of several most probable candidates based on data from adjacent PBs and the reference picture. Motion information typically includes the horizontal and vertical motion vector displacement values, one or two reference picture indices, and, in the case of prediction regions in B slices, an identification of which reference picture list is associated with each index. As used herein, in some aspects, a “merge mode” may refer to deriving the motion information from spatially or temporally neighboring blocks.

[0067] The motion compensation unit 302 may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

[0068] The motion compensation unit 302 may use the interpolation filters as used by the video encoder 200 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. The motion compensation unit 302 may determine the interpolation filters used by the video encoder 200 according to the received syntax information

and use the interpolation filters to produce predictive blocks.

[0069] The motion compensation unit 302 may use at least part of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter-encoded block, and other information to decode the encoded video sequence. As used herein, in some aspects, a “slice” may refer to a data structure that can be decoded independently from other slices of the same picture, in terms of entropy coding, signal prediction, and residual signal reconstruction. A slice can either be an entire picture or a region of a picture.

[0070] The intra prediction unit 303 may use intra prediction modes, which, for example, are received in the bitstream, to form a prediction block from spatially adjacent blocks. The inverse quantization unit 304 inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by the entropy decoding unit 301. The inverse transform unit 305 applies an inverse transform.

[0071] The reconstruction unit 306 may obtain the decoded blocks, e.g., by summing the residual blocks with the corresponding prediction blocks generated by the motion compensation unit 302 or intra-prediction unit 303. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in the buffer 307, which provides reference blocks for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

[0072] Some example embodiments of the present disclosure will be described in detailed hereinafter. It should be understood that section headings are used in the present document to facilitate ease of understanding and do not limit the embodiments disclosed in a section to only that section. Furthermore, while certain embodiments are described with reference to Versatile Video Coding or other specific video codecs, the disclosed techniques are applicable to other video coding technologies also. Furthermore, while some embodiments describe video coding steps in detail, it will be understood that corresponding steps decoding that undo

the coding will be implemented by a decoder. Furthermore, the term video processing encompasses video coding or compression, video decoding or decompression and video transcoding in which video pixels are represented from one compressed format into another compressed format or at a different compressed bitrate.

1. Summary

This disclosure is related to video coding technologies. Specifically, it is about prediction mode refinement, motion information refinement, prediction samples refinement related techniques in video coding. It may be applied to the existing video coding standard like HEVC, VVC, and etc. It may be also applicable to future video coding standards or video codec.

2. Background

Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards (ITU-T and ISO/IEC, “High efficiency video coding”, Rec. ITU-T H.265 | ISO/IEC 23008-2 (in force edition)). Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, the Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. The JVET meeting is concurrently held once every quarter, and the new video coding standard was officially named as Versatile Video Coding (VVC) in the April 2018 JVET meeting, and the first version of VVC test model (VTM) was released at that time. The VVC working draft and test model VTM are then updated after every meeting. The VVC project achieved technical completion (FDIS) at the July 2020 meeting.

2.1. Explanation of coding tools

2.1.1. Extended merge prediction

In VVC, the merge candidate list is constructed by including the following five types of candidates in order:

- 1) Spatial MVP from spatial neighbour CUs
- 2) Temporal MVP from collocated CUs
- 3) History-based MVP from an FIFO table
- 4) Pairwise average MVP
- 5) Zero MVs.

The size of merge list is signalled in sequence parameter set header and the maximum allowed size of merge list is 6. For each CU code in merge mode, an index of best merge candidate is encoded using truncated unary binarization (TU). The first bin of the merge index is coded with context and bypass coding is used for other bins.

The derivation process of each category of merge candidates is provided in this session. As done in HEVC, VVC also supports parallel derivation of the merging candidate lists for all CUs within a certain size of area.

2.1.1.1. Spatial candidates derivation

The derivation of spatial merge candidates in VVC is same to that in HEVC except the positions of first two merge candidates are swapped. A maximum of four merge candidates are selected among candidates located in the positions depicted in Fig. 4. The order of derivation is B_0 , A_0 , B_1 , A_1 and B_2 . Position B_2 is considered only when one or more than one CUs of position B_0 , A_0 , B_1 , A_1 are not available (e.g. because it belongs to another slice or tile) or is intra coded. After candidate at position A_1 is added, the addition of the remaining candidates is subject to a redundancy check which ensures that candidates with same motion information are excluded from the list so that coding efficiency is improved. To reduce computational complexity, not all possible candidate pairs are considered in the mentioned redundancy check. Instead only the pairs linked with an arrow in Fig. 5 are considered and a candidate is only added to the list if the corresponding candidate used for redundancy check has not the same motion information.

2.1.1.2. Temporal candidates derivation

In this step, only one candidate is added to the list. Particularly, in the derivation of this temporal merge candidate, a scaled motion vector is derived based on co-located CU belonging to the collocated referenncee picture. The reference picture list to be used for derivation of the co-located CU is explicitly signalled in the slice header. The scaled motion vector for temporal merge candidate is obtained as illustrated by the dotted line in Fig. 6, which is scaled from the motion vector of the co-located CU 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.

The position for the temporal candidate is selected between candidates C_0 and C_1 , as depicted in Fig. 7. If CU at position C_0 is not available, is intra coded, or is outside of the current row of CTUs, position C_1 is used. Otherwise, position C_0 is used in the derivation of the temporal merge candidate.

2.1.1.3. History-based merge candidates derivation

The history-based MVP (HMVP) merge candidates are added to merge list after the spatial MVP and TMVP. In this method, the motion information of a previously coded block is stored in a table and used as MVP for the current CU. The table with multiple HMVP candidates is maintained during the encoding/decoding process. The table is reset (emptied) when a new CTU row is encountered. Whenever there is a non-subblock inter-coded CU, the associated motion information is added to the last entry of the table as a new HMVP candidate.

The HMVP table size S is set to be 6, which indicates up to 6 History-based MVP (HMVP) candidates may be added to the table. When inserting a new motion candidate to the table, a constrained first-in-first-out (FIFO) rule is utilized wherein redundancy check is firstly applied to find whether there is an

identical HMVP in the table. If found, the identical HMVP is removed from the table and all the HMVP candidates afterwards are moved forward,

HMVP candidates could be used in the merge candidate list construction process. The latest several HMVP candidates in the table are checked in order and inserted to the candidate list after the TMVP candidate. Redundancy check is applied on the HMVP candidates to the spatial or temporal merge candidate.

To reduce the number of redundancy check operations, the following simplifications are introduced:

1. Number of HMPV candidates is used for merge list generation is set as $(N \leq 4) ? M : (8 - N)$, wherein N indicates number of existing candidates in the merge list and M indicates number of available HMVP candidates in the table.
2. Once the total number of available merge candidates reaches the maximally allowed merge candidates minus 1, the merge candidate list construction process from HMVP is terminated.

2.1.1.4. Pair-wise average merge candidates derivation

Pairwise average candidates are generated by averaging predefined pairs of candidates in the existing merge candidate list, and the predefined pairs are defined as $\{(0, 1), (0, 2), (1, 2), (0, 3), (1, 3), (2, 3)\}$, where the numbers denote the merge indices to the merge candidate list. The averaged motion vectors are calculated separately for each reference list. If both motion vectors are available in one list, these two motion vectors are averaged even when they point to different reference pictures; if only one motion vector is available, use the one directly; if no motion vector is available, keep this list invalid.

When the merge list is not full after pair-wise average merge candidates are added, the zero MVPs are inserted in the end until the maximum merge candidate number is encountered.

2.1.1.5. Merge estimation region

Merge estimation region (MER) allows independent derivation of merge candidate list for the CUs in the same merge estimation region (MER). A candidate block that is within the same MER to the current CU is not included for the generation of the merge candidate list of the current CU. In addition, the updating process for the history-based motion vector predictor candidate list is updated only if $(x_{Cb} + cbWidth) \gg \text{Log2ParMrgLevel}$ is greater than $x_{Cb} \gg \text{Log2ParMrgLevel}$ and $(y_{Cb} + cbHeight) \gg \text{Log2ParMrgLevel}$ is great than $y_{Cb} \gg \text{Log2ParMrgLevel}$ and where (x_{Cb}, y_{Cb}) is the top-left luma sample position of the current CU in the picture and $(cbWidth, cbHeight)$ is the CU size. The MER size is selected at encoder side and signalled as `log2_parallel_merge_level_minus2` in the sequence parameter set.

2.1.2. Merge mode with MVD (MMVD)

In addition to merge mode, where the implicitly derived motion information is directly used for prediction samples generation of the current CU, the merge mode with motion vector differences (MMVD) is introduced in VVC. A MMVD flag is signalled right after sending a skip flag and merge flag to specify whether MMVD mode is used for a CU.

In MMVD, after a merge candidate is selected, it is further refined by the signalled MVDs information. The further information includes a merge candidate flag, an index to specify motion magnitude, and an index for indication of motion direction. In MMVD mode, one for the first two candidates in the merge list is selected to be used as MV basis. The merge candidate flag is signalled to specify which one is used.

Distance index specifies motion magnitude information and indicate the pre-defined offset from the starting point. As shown in Fig. 8, an offset is added to either horizontal component or vertical component of starting MV. The relation of distance index and pre-defined offset is specified in Table 1

Table 1 – The relation of distance index and pre-defined offset

Distance IDX	0	1	2	3	4	5	6	7
Offset (in unit of luma sample)	1/4	1/2	1	2	4	8	16	32

Direction index represents the direction of the MVD relative to the starting point. The direction index can represent of the four directions as shown in Table 2. It's noted that the meaning of MVD sign could be variant according to the information of starting MVs. When the starting MVs is an un-prediction MV or bi-prediction MVs with both lists point to the same side of the current picture (i.e. POCs of two references are both larger than the POC of the current picture, or are both smaller than the POC of the current picture), the sign in Table 2 specifies the sign of MV offset added to the starting MV. When the starting MVs is bi-prediction MVs with the two MVs point to the different sides of the current picture (i.e. the POC of one reference is larger than the POC of the current picture, and the POC of the other reference is smaller than the POC of the current picture), the sign in Table 2 specifies the sign of MV offset added to the list0 MV component of starting MV and the sign for the list1 MV has opposite value.

Table 2 – Sign of MV offset specified by direction index

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

2.1.3. Decoder side motion vector refinement (DMVR)

In order to increase the accuracy of the MVs of the merge mode, a bilateral-matching based decoder side motion vector refinement is applied in VVC. In bi-prediction operation, a refined MV is searched around the initial MVs in the reference picture list L0 and reference picture list L1. The BM method calculates the distortion between the two candidate blocks in the reference picture list L0 and list L1. As illustrated in Fig. 9, the SAD between the blocks 901 and 902 based on each MV candidate around the initial MV is calculated. The MV candidate with the lowest SAD becomes the refined MV and used to generate the bi-predicted signal.

In VVC, the DMVR can be applied for the CUs which are coded with following modes and features:

- CU level merge mode with bi-prediction MV
- One reference picture is in the past and another reference picture is in the future with respect to the current picture
- The distances (i.e. POC difference) from two reference pictures to the current picture are same
- Both reference pictures are short-term reference pictures
- CU has more than 64 luma samples
- Both CU height and CU width are larger than or equal to 8 luma samples
- BCW weight index indicates equal weight
- WP is not enabled for the current block
- CIIP mode is not used for the current block

The refined MV derived by DMVR process is used to generate the inter prediction samples and also used in temporal motion vector prediction for future pictures coding. While the original MV is used in deblocking process and also used in spatial motion vector prediction for future CU coding.

The additional features of DMVR are mentioned in the following sub-clauses.

2.1.3.1. Searching scheme

In DMVR, the search points are surrounding the initial MV and the MV offset obey the MV difference mirroring rule. In other words, any points that are checked by DMVR, denoted by candidate MV pair (MV0, MV1) obey the following two equations:

$$MV0' = MV0 + MV_offset \quad (1)$$

$$MV1' = MV1 - MV_offset \quad (2)$$

Where MV_offset represents the refinement offset between the initial MV and the refined MV in one of the reference pictures. The refinement search range is two integer luma samples from the initial MV. The searching includes the integer sample offset search stage and fractional sample refinement stage.

25 points full search is applied for integer sample offset searching. The SAD of the initial MV pair is first calculated. If the SAD of the initial MV pair is smaller than a threshold, the integer sample stage of DMVR is terminated. Otherwise SADs of the remaining 24 points are calculated and checked in raster scanning order. The point with the smallest SAD is selected as the output of integer sample offset searching stage. To reduce the penalty of the uncertainty of DMVR refinement, it is proposed to favor the original MV during the DMVR process. The SAD between the reference blocks referred by the initial MV candidates is decreased by 1/4 of the SAD value.

The integer sample search is followed by fractional sample refinement. To save the calculational complexity, the fractional sample refinement is derived by using parametric error surface equation, instead of additional search with SAD comparison. The fractional sample refinement is conditionally invoked based on the output of the integer sample search stage. When the integer sample search stage is

terminated with center having the smallest SAD in either the first iteration or the second iteration search, the fractional sample refinement is further applied.

In parametric error surface based sub-pixel offsets estimation, the center position cost and the costs at four neighboring positions from the center are used to fit a 2-D parabolic error surface equation of the following form

$$E(x, y) = A(x - x_{min})^2 + B(y - y_{min})^2 + C \quad (3)$$

where (x_{min}, y_{min}) corresponds to the fractional position with the least cost and C corresponds to the minimum cost value. By solving the above equations by using the cost value of the five search points, the (x_{min}, y_{min}) is computed as:

$$x_{min} = (E(-1,0) - E(1,0)) / (2(E(-1,0) + E(1,0) - 2E(0,0))) \quad (4)$$

$$y_{min} = (E(0,-1) - E(0,1)) / (2(E(0,-1) + E(0,1) - 2E(0,0))) \quad (5)$$

The value of x_{min} and y_{min} are automatically constrained to be between -8 and 8 since all cost values are positive and the smallest value is $E(0,0)$. This corresponds to half pel offset with 1/16th-pel MV accuracy in VVC. The computed fractional (x_{min}, y_{min}) are added to the integer distance refinement MV to get the sub-pixel accurate refinement delta MV.

2.1.3.2. Bilinear-interpolation and sample padding

In VVC, the resolution of the MVs is 1/16 luma samples. The samples at the fractional position are interpolated using a 8-tap interpolation filter. In DMVR, the search points are surrounding the initial fractional-pel MV with integer sample offset, therefore the samples of those fractional position need to be interpolated for DMVR search process. To reduce the calculation complexity, the bi-linear interpolation filter is used to generate the fractional samples for the searching process in DMVR. Another important effect is that by using bi-linear filter is that with 2-sample search range, the DVMR does not access more reference samples compared to the normal motion compensation process. After the refined MV is attained with DMVR search process, the normal 8-tap interpolation filter is applied to generate the final prediction. In order to not access more reference samples to normal MC process, the samples, which is not needed for the interpolation process based on the original MV but is needed for the interpolation process based on the refined MV, will be padded from those available samples.

2.1.3.3. Maximum DMVR processing unit

When the width and/or height of a CU are larger than 16 luma samples, it will be further split into subblocks with width and/or height equal to 16 luma samples. The maximum unit size for DMVR searching process is limit to 16x16.

2.1.4. Geometric partitioning mode (GPM) for Inter Prediction

In VVC, a geometric partitioning mode is supported for inter prediction. The geometric partitioning mode is signalled using a CU-level flag as one kind of merge mode, with other merge modes including the regular merge mode, the MMVD mode, the CIIP mode and the subblock merge mode. In total 64

partitions are supported by geometric partitioning mode for each possible CU size $w \times h = 2^m \times 2^n$ with $m, n \in \{3 \dots 6\}$ excluding 8×64 and 64×8 .

When this mode is used, a CU is split into two parts by a geometrically located straight line (Fig. 10). The location of the splitting line is mathematically derived from the angle and offset parameters of a specific partition. Each part of a geometric partition in the CU is inter-predicted using its own motion; only uni-prediction is allowed for each partition, that is, each part has one motion vector and one reference index. The uni-prediction motion constraint is applied to ensure that same as the conventional bi-prediction, only two motion compensated prediction are needed for each CU. The uni-prediction motion for each partition is derived using the process described in 3.4.1.

If geometric partitioning mode is used for the current CU, then a geometric partition index indicating the partition mode of the geometric partition (angle and offset), and two merge indices (one for each partition) are further signalled. The number of maximum GPM candidate size is signalled explicitly in SPS and specifies syntax binarization for GPM merge indices. After predicting each of part of the geometric partition, the sample values along the geometric partition edge are adjusted using a blending processing with adaptive weights as in 3.4.2. This is the prediction signal for the whole CU, and transform and quantization process will be applied to the whole CU as in other prediction modes. Finally, the motion field of a CU predicted using the geometric partition modes is stored as in 3.4.3.

2.1.4.1. Uni-prediction candidate list construction

The uni-prediction candidate list is derived directly from the merge candidate list constructed according to the extended merge prediction process in 3.4.1. Denote n as the index of the uni-prediction motion in the geometric uni-prediction candidate list. The LX motion vector of the n -th extended merge candidate, with X equal to the parity of n , is used as the n -th uni-prediction motion vector for geometric partitioning mode. These motion vectors are marked with “x” in Fig. 11. In case a corresponding LX motion vector of the n -th extended merge candidate does not exist, the $L(1 - X)$ motion vector of the same candidate is used instead as the uni-prediction motion vector for geometric partitioning mode.

2.1.4.2. Blending along the geometric partitioning edge

After predicting each part of a geometric partition using its own motion, blending is applied to the two prediction signals to derive samples around geometric partition edge. The blending weight for each position of the CU are derived based on the distance between individual position and the partition edge.

The distance for a position (x, y) to the partition edge are derived as:

$$d(x, y) = (2x + 1 - w) \cos(\varphi_i) + (2y + 1 - h) \sin(\varphi_i) - \rho_j \quad (6)$$

$$\rho_j = \rho_{x,j} \cos(\varphi_i) + \rho_{y,j} \sin(\varphi_i) \quad (7)$$

$$\rho_{x,j} = \begin{cases} 0 & i \% 16 = 8 \text{ or } (i \% 16 \neq 0 \text{ and } h \geq w) \\ \pm(j \times w) \gg 2 & \text{otherwise} \end{cases} \quad (8)$$

$$\rho_{y,j} = \begin{cases} \pm(j \times h) \gg 2 & i \% 16 = 8 \text{ or } (i \% 16 \neq 0 \text{ and } h \geq w) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where i, j are the indices for angle and offset of a geometric partition, which depend on the signaled geometric partition index. The sign of $\rho_{x,j}$ and $\rho_{y,j}$ depend on angle index i .

The weights for each part of a geometric partition are derived as following:

$$wIdxL(x, y) = partIdx ? 32 + d(x, y) : 32 - d(x, y) \quad (10)$$

$$w_0(x, y) = \frac{clip3(0, 8, (wIdxL(x, y) + 4) \gg 3)}{8} \quad (11)$$

$$w_1(x, y) = 1 - w_0(x, y) \quad (12)$$

The partIdx depends on the angle index i . One example of weigh w_0 is illustrated in Fig. 12.

2.1.4.3. Motion field storage for geometric partitioning mode

Mv1 from the first part of the geometric partition, Mv2 from the second part of the geometric partition and a combined Mv of Mv1 and Mv2 are stored in the motion filed of a geometric partitioning mode coded CU.

The stored motion vector type for each individual position in the motion filed are determined as:

$$sType = abs(motionIdx) < 32 ? 2 : (motionIdx \leq 0 ? (1 - partIdx) : partIdx) \quad (13)$$

where motionIdx is equal to $d(4x + 2, 4y + 2)$. The partIdx depends on the angle index i .

If sType is equal to 0 or 1, Mv0 or Mv1 are stored in the corresponding motion field, otherwise if sType is equal to 2, a combined Mv from Mv0 and Mv2 are stored. The combined Mv are generated using the following process:

- 1) If Mv1 and Mv2 are from different reference picture lists (one from L0 and the other from L1), then Mv1 and Mv2 are simply combined to form the bi-prediction motion vectors.
- 2) Otherwise, if Mv1 and Mv2 are from the same list, only uni-prediction motion Mv2 is stored.

2.2. Geometric prediction mode with motion vector differences (GMVD)

With Geometric prediction mode with Motion Vector Difference (GMVD), each geometric partition in GPM can decide to use GMVD or not. If GMVD is chosen for a geometric region, the MV of the region is calculated as a sum of the MV of a merge candidate and an MVD. All other processing is kept the same as in GPM.

With GMVD, an MVD is signaled as a pair of direction and distance, following the current design of MMVD. That is, there are eight candidate distances (1/4-pel, 1/2-pel, 1-pel, 2-pel, 4-pel, 8-pel, 16-pel, 32-pel), and four candidate directions (toward-left, toward-right, toward-above, and toward-below). In addition, when pic_fpel_mmvd_enabled_flag is equal to 1, the MVD in GMVD is also left shifted by 2 as in MMVD.

2.3. Combined inter and intra prediction (CIIP)

In VVC, when a CU is coded in merge mode, if the CU contains at least 64 luma samples (that is, CU width times CU height is equal to or larger than 64), and if both CU width and CU height are less than 128 luma samples, an additional flag is signalled to indicate if the combined inter/intra prediction (CIIP) mode is applied to the current CU. As its name indicates, the CIIP prediction combines an inter prediction signal with an intra prediction signal. The inter prediction signal in the CIIP mode P_{inter} is derived using the same inter prediction process applied to regular merge mode; and the intra prediction signal P_{intra} is derived following the regular intra prediction process with the planar mode. Then, the intra and inter prediction signals are combined using weighted averaging, where the weight value is calculated depending on the coding modes of the top and left neighbouring blocks (depicted in Fig. 13) as follows:

- If the top neighbor is available and intra coded, then set isIntraTop to 1, otherwise set isIntraTop to 0;
- If the left neighbor is available and intra coded, then set isIntraLeft to 1, otherwise set isIntraLeft to 0;
- If (isIntraLeft + isIntraTop) is equal to 2, then wt is set to 3;
- Otherwise, if (isIntraLeft + isIntraTop) is equal to 1, then wt is set to 2;
- Otherwise, set wt to 1.

The CIIP prediction is formed as follows:

$$P_{CIIP} = ((4 - wt) * P_{inter} + wt * P_{intra} + 2) \gg 2 \quad (3-1)$$

2.4. Multi-hypothesis prediction (MHP)

The multi-hypothesis prediction is adopted in this contribution. Up to two additional predictors are signalled on top of inter AMVP mode, regular merge mode, and MMVD mode. The resulting overall prediction signal is accumulated iteratively with each additional prediction signal.

$$p_{n+1} = (1 - \alpha_{n+1})p_n + \alpha_{n+1}h_{n+1}$$

The weighting factor α is specified according to the following table:

add_hyp_weight_idx	α
0	1/4
1	-1/8

For inter AMVP mode, MHP is only applied if non-equal weight in BCW is selected in bi-prediction mode.

2.5. Template matching (TM)

Template matching (TM) is a decoder-side MV derivation method to refine the motion information of the current CU by finding the closest match between a template (i.e., top and/or left neighbouring blocks of the current CU) in the current picture and a block (i.e., same size to the template) in a reference picture. As illustrated in Fig. 14, a better MV is to be searched around the initial motion of the current CU within

a $[-8, +8]$ -pel search range. The template matching is adopted in this contribution with two modifications: search step size is determined based on AMVR mode and TM can be cascaded with bilateral matching process in merge modes.

In AMVP mode, an MVP candidate is determined based on template matching error to pick up the one which reaches the minimum difference between current block template and reference block template, and then TM performs only for this particular MVP candidate for MV refinement. TM refines this MVP candidate, starting from full-pel MVD precision (or 4-pel for 4-pel AMVR mode) within a $[-8, +8]$ -pel search range by using iterative diamond search. The AMVP candidate may be further refined by using cross search with full-pel MVD precision (or 4-pel for 4-pel AMVR mode), followed sequentially by half-pel and quarter-pel ones depending on AMVR mode as specified in Table 3. This search process ensures that the MVP candidate still keeps the same MV precision as indicated by AMVR mode after TM process.

Table 3. Search patterns of AMVR and merge mode with AMVR.

Search pattern	AMVR mode				Merge mode	
	4-pel	Full-pel	Half-pel	Quarter-pel	AltIF=0	AltIF=1
4-pel diamond	v					
4-pel cross	v					
Full-pel diamond		v	v	v	v	v
Full-pel cross		v	v	v	v	v
Half-pel cross			v	v	v	v
Quarter-pel cross				v	v	
1/8-pel cross					v	

In merge mode, similar search method is applied to the merge candidate indicated by the merge index. As Table 3 shows, TM may perform all the way down to 1/8-pel MVD precision or skipping those beyond half-pel MVD precision, depending on whether the alternative interpolation filter (that is used when AMVR is of half-pel mode) is used according to merged motion information. Besides, when TM mode is enabled, template matching may work as an independent process or an extra MV refinement process between block-based and subblock-based bilateral matching (BM) methods, depending on whether BM can be enabled or not according to its enabling condition check.

2.6. Multi-pass decoder-side motion vector refinement

In this contribution, a multi-pass decoder-side motion vector refinement is applied. In the first pass, bilateral matching (BM) is applied to the coding block. In the second pass, BM is applied to each 16x16 subblock within the coding block. In the third pass, MV in each 8x8 subblock is refined by applying bi-directional optical flow (BDOF). The refined MVs are stored for both spatial and temporal motion vector prediction.

▪ First pass – Block based bilateral matching MV refinement

In the first pass, a refined MV is derived by applying BM to a coding block. Similar to decoder-side motion vector refinement (DMVR), in bi-prediction operation, a refined MV is searched around the two initial MVs (MV0 and MV1) in the reference picture lists L0 and L1. The refined MVs (MV0_pass1 and MV1_pass1) are derived around the initiate MVs based on the minimum bilateral matching cost between the two reference blocks in L0 and L1.

BM performs local search to derive integer sample precision intDeltaMV. The local search applies a 3×3 square search pattern to loop through the search range [−sHor, sHor] in horizontal direction and [−sVer, sVer] in vertical direction, wherein, the values of sHor and sVer are determined by the block dimension, and the maximum value of sHor and sVer is 8.

The bilateral matching cost is calculated as: $\text{bilCost} = \text{mvDistanceCost} + \text{sadCost}$. When the block size $\text{cbW} * \text{cbH}$ is greater than 64, MRSAD cost function is applied to remove the DC effect of distortion between reference blocks. When the bilCost at the center point of the 3×3 search pattern has the minimum cost, the intDeltaMV local search is terminated. Otherwise, the current minimum cost search point becomes the new center point of the 3×3 search pattern and continue to search for the minimum cost, until it reaches the end of the search range.

The existing fractional sample refinement is further applied to derive the final deltaMV. The refined MVs after the first pass is then derived as:

- $\text{MV0_pass1} = \text{MV0} + \text{deltaMV}$
- $\text{MV1_pass1} = \text{MV1} - \text{deltaMV}$

▪ Second pass – Subblock based bilateral matching MV refinement

In the second pass, a refined MV is derived by applying BM to a 16×16 grid subblock. For each subblock, a refined MV is searched around the two MVs (MV0_pass1 and MV1_pass1), obtained on the first pass, in the reference picture list L0 and L1. The refined MVs (MV0_pass2(sbIdx2) and MV1_pass2(sbIdx2)) are derived based on the minimum bilateral matching cost between the two reference subblocks in L0 and L1.

For each subblock, BM performs full search to derive integer sample precision intDeltaMV. The full search has a search range [−sHor, sHor] in horizontal direction and [−sVer, sVer] in vertical direction, wherein, the values of sHor and sVer are determined by the block dimension, and the maximum value of sHor and sVer is 8.

The bilateral matching cost is calculated by applying a cost factor to the SATD cost between two reference subblocks, as: $\text{bilCost} = \text{satdCost} * \text{costFactor}$. The search area $(2 * \text{sHor} + 1) * (2 * \text{sVer} + 1)$ is divided up to 5 diamond shape search regions shown on Fig. 15. Each search region is assigned a costFactor, which is determined by the distance (intDeltaMV) between each search point and the starting MV, and each diamond region is processed in the order starting from the center of the search area. In each region, the search points are processed in the raster scan order starting from the top left going to the bottom right corner of the region. When the minimum bilCost within the current search region is less than a threshold equal to $\text{sbW} * \text{sbH}$, the int-pel full search is terminated, otherwise, the int-pel full search continues to the next search region until all search points are examined.

The existing VVC DMVR fractional sample refinement is further applied to derive the final $\text{deltaMV}(\text{sbIdx2})$. The refined MVs at second pass is then derived as:

- $\text{MV0_pass2}(\text{sbIdx2}) = \text{MV0_pass1} + \text{deltaMV}(\text{sbIdx2})$
- $\text{MV1_pass2}(\text{sbIdx2}) = \text{MV1_pass1} - \text{deltaMV}(\text{sbIdx2})$

▪ Third pass – Subblock based bi-directional optical flow MV refinement

In the third pass, a refined MV is derived by applying BDOF to an 8×8 grid subblock. For each 8×8 subblock, BDOF refinement is applied to derive scaled V_x and V_y without clipping starting from the refined MV of the parent subblock of the second pass. The derived $\text{bioMv}(V_x, V_y)$ is rounded to 1/16 sample precision and clipped between -32 and 32.

The refined MVs ($\text{MV0_pass3}(\text{sbIdx3})$ and $\text{MV1_pass3}(\text{sbIdx3})$) at third pass are derived as:

- $\text{MV0_pass3}(\text{sbIdx3}) = \text{MV0_pass2}(\text{sbIdx2}) + \text{bioMv}$
- $\text{MV1_pass3}(\text{sbIdx3}) = \text{MV0_pass2}(\text{sbIdx2}) - \text{bioMv}$

2.7. Non-adjacent spatial candidate

The non-adjacent spatial merge candidates are inserted after the TMVP in the regular merge candidate list. The pattern of spatial merge candidates is shown in Fig. 16. The distances between non-adjacent spatial candidates and current coding block are based on the width and height of current coding block. The line buffer restriction, however, is not applied in this contribution.

2.8. GPM motion refinement

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

The term ‘GPM’ may represent a coding method that split one block into two or more sub-regions wherein at least one sub-region is non-rectangular, or non-square, or it couldn’t be generated by any of existing partitioning structure (e.g., QT/BT/TT) which splits one block into multiple rectangular sub-regions. In one example, for the GPM coded blocks, one or more weighting masks are derived for a coding block based on how the sub-regions are split, and the final prediction signal of the coding block is generated by a weighted-sum of two or more auxiliary prediction signals associated with the sub-regions.

The term ‘GPM’ may indicate the geometric merge mode (GEO), and/or geometric partition mode (GPM), and/or wedge prediction mode, and/or triangular prediction mode (TPM), and/or a GPM block with motion vector difference (GMVD), and/or a GPM block with motion refinement, and/or any variant based on GPM.

The term ‘block’ may represent a coding block (CB), a CU, a PU, a TU, a PB, a TB.

The phrase “normal/regular merge candidate” may represent the merge candidates generated by the extended merge prediction process (as illustrated in section 3.1). It may also represent any other advanced merge candidates except GEO merge candidates and subblock based merge candidates.

Note that a part/partition of a GPM/GMVD block means a part of a geometric partition in the CU, e.g., the two parts of a GPM block in Fig. 10 are split by a geometrically located straight line. Each part of a geometric partition in the CU is inter-predicted using its own motion, but the transform is performed for the whole CU rather than each part/partition of a GPM block.

It should also be noticed that GPM/GMVD applied to other modes (e.g., AMVP mode) may also use the following methods wherein the motion for merge mode may be replaced by motion for AMVP mode.

It is noticed that in the following descriptions, the term ‘GPM merge list’ is given as an example. However, the proposed solutions could also be extended to other GPM candidate list, such as GPM AMVP candidate list.

In the disclosure, a merge candidate is called to be “refined” if the motion information of the merge candidate is modified according to information signaled from the encoder or derived at the decoder. For example, a merge candidate may be refined by DVMR, FRUC, TM, MMVD, BDOF and so on.

1. In one example, during the GPM merge list construction process, the GPM motion information may be generated from a refined regular merge candidate.
 - 1) For example, the refinement process may be conducted on a regular merge candidate list, before the GPM merge list construction process. E.g., the GPM merge list may be constructed based on refined regular merge candidates.
 - 2) For example, refined L0 motion and/or L1 motion of a regular merge candidate may be used as a GPM merge candidate.
 - a) For example, a bi-prediction regular merge candidate may be firstly refined by a decoder side motion derivation/refinement process, and then being used for derivation of GPM motion information.
 - b) For example, a uni-prediction regular merge candidate may be firstly refined by a decoder side motion derivation/refinement process, and then being used for derivation of GPM motion information.
 - 3) Whether to refine a merge candidate or a merge candidate list may depend on the motion information of the candidates.
 - a) For example, if a normal merge candidate satisfies the condition of the decoder side motion derivation/refinement method, then this normal merge candidate may be firstly refined by such method, and then being used for derivation of GPM motion information.
2. In one example, after deriving the GPM motion information according to candidate indices (e.g., using the parity and candidate indices to a regular merge candidate list in VVC), the motion information may be further refined by another process.

- 1) Alternatively, furthermore, the final prediction of a GPM coded video unit may be dependent on the refined motion information.
- 2) For example, the refinement process may be conducted on a GPM merge candidate list, after the GPM merge list construction process. E.g., the GPM merge list may be constructed based on non-refined regular merge candidates.
- 3) For example, a GPM merge candidate list (e.g., uni-prediction) is firstly build from a regular merge candidate list, and then any of the GPM merge candidates may be further refined through decoder side motion derivation methods.
3. In one example, a two-stage refinement process may be applied.
 - 1) For example, a first refinement process may be conducted on a regular merge candidate list, before the GPM merge list construction process. E.g., the GPM merge list may be constructed based on regular merge candidates refined by the first refinement process.
 - 2) For example, a second refinement process may be conducted on a GPM merge candidate list, after the GPM merge list construction process.
4. In one example, the motion refinement of a GPM block may be conducted for multiple candidates (e.g., corresponding to multiple parts, e.g., both part-0 motion and part-1 motion), simultaneously.
 - 1) Alternatively, the motion refinement of a GPM block may be conducted for part-0 motion and part-1 motion, respectively.
5. In one example, the motion refinement of a GPM block may be applied to at least one part of a GPM block.
 - 1) For example, the motion refinement of a GPM block may be applied to both parts of a GPM block.
 - 2) For example, the motion refinement of a GPM block may be applied to a certain part (not both) of a GPM block, wherein the part index may be predefined or determined by a rule.
6. In one example, the aforementioned motion refinement (e.g., decoder side motion derivation) process may be based on a bilateral matching method (such as DMVR which measures the prediction sample difference between L0 prediction block and L1 prediction block).
 - 1) For example, the L0/L1 prediction in the bilateral matching of a GPM block may take into account the whole block's information regardless of the GPM split mode information, e.g., a reference block with the same size of the whole GPM block is used a L0/L1 prediction.
 - a) Alternatively, the L0/L1 prediction in the bilateral matching of a GPM block may take into account the GPM split mode information, e.g., a reference block with the block shape as same as the part-0/1 associated with a specific GPM split mode may be taken into account.
 - 2) Alternatively, the aforementioned motion refinement (e.g., decoder side motion derivation) process may be based on a template matching method (e.g., measures the prediction sample difference between template samples in the current picture and template samples in the reference picture, wherein template samples may be the above/left neighbors of the current video unit).
 - a) Furthermore, the template may be uni-directional and/or bi-directional.

- b) For example, the template for part-0 and part-1 may be based on different rules.
 - c) For example, the template matching process may be applied to a whole block, but the refinement information derived from the template matching process is applied to one part of the block.
 - d) For example, the template matching may be applied to a part individually (not applying template matching on the whole block for two parts).
 - a. In one example, the shape of a template for a part may depend on the shape of the part.
- 3) Furthermore, whether to use bilateral matching method or template matching method to refine a regular merge candidate may be dependent on the motion data of the regular /GPM merge candidate (such as prediction direction, how different the L0 and L1 motion vectors are, POC distances of L0 and L1 motion, and etc.).
- 4) Additionally, the refinement process may be applied for GPM motion, without explicit signalling.
 - a) Alternatively, whether to allow the refinement or not may be explicitly signalled.
- 7. In one example, the refined motion may be used for the motion compensation for a GPM block.
 - 1) Alternatively, the original motion without the refinement may be used for the motion compensation for a GPM block.
- 8. In one example, the refined motion may be used for the subblock (e.g., 4x4) based motion vector storage for a GPM block.
 - 1) Alternatively, the original motion without the refinement may be used for the subblock based motion vector storage for a GPM block.
 - 2) In one example, the refined motion may be used for the deblocking strength determination for a GPM block.
 - a) Alternatively, the original motion without the refinement may be used for the deblocking strength determination for a GPM block.
 - 3) In one example, when generating the AMVP/Merge candidate list for a succeeding block, which may be GPM-coded or non-GPM-coded, the refined motion of a GPM block may be used as 1) a temporal motion vector candidate when the temporal neighbor block is the GPM block, and/or 2) a spatial motion vector candidate when the spatial neighbor block is the GPM block.
 - a) Alternatively, the original motion without the refinement may be used in any of the above-mentioned case.
- 9. In one example, MVD may be added to a refined MV for a block with GMVD mode.
 - 1) Alternatively, MVD may be added to a non-refined MV for a block with GMVD mode, and then the resulted MV is to be refined.
- 10. How to conduct the refinement process may be dependent on whether GPM and/or GMVD is used.
 - 1) For example, less searching points are checked in the refinement process if GPM and/or GMVD is used.

2.9. GPM prediction sample refinement

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

The term ‘GPM’ may represent a coding method that split one block into two or more sub-regions wherein at least one sub-region is non-rectangular, or non-square, or it couldn’t be generated by any of existing partitioning structure (e.g., QT/BT/TT) which splits one block into multiple rectangular sub-regions. In one example, for the GPM coded blocks, one or more weighting masks are derived for a coding block based on how the sub-regions are split, and the final prediction signal of the coding block is generated by a weighted-sum of two or more auxiliary prediction signals associated with the sub-regions.

The term ‘GPM’ may indicate the geometric merge mode (GEO), and/or geometric partition mode (GPM), and/or wedge prediction mode, and/or triangular prediction mode (TPM), and/or a GPM block with motion vector difference (GMVD), and/or a GPM block with motion refinement, and/or any variant based on GPM.

The term ‘block’ may represent a coding block (CB), a CU, a PU, a TU, a PB, a TB.

The phrase “normal/regular merge candidate” may represent the merge candidates generated by the extended merge prediction process (as illustrated in section 3.1). It may also represent any other advanced merge candidates except GEO merge candidates and subblock based merge candidates.

Note that a part/partition of a GPM/GMVD block means a part of a geometric partition in the CU, e.g., the two parts of a GPM block in Fig. 10 are split by a geometrically located straight line. Each part of a geometric partition in the CU is inter-predicted using its own motion, but the transform is performed for the whole CU rather than each part/partition of a GPM block.

It should also be noticed that GPM/GMVD applied to other modes (e.g., AMVP mode) may also use the following methods wherein the motion for merge mode may be replaced by motion for AMVP mode.

1. In one example, motion-compensated prediction sample refinement process may be applied to a GPM block.
 - a. For example, at least one prediction sample of a GPM prediction block may be refined by an overlapped block-based motion compensation (e.g., OBMC) technique, in which the prediction samples are refined using neighboring block’s motion information with a weighted prediction.
 - b. For example, at least one prediction sample of a GPM prediction block may be refined by a multi-hypothesis prediction (e.g., MHP) technique in which the resulting overall prediction samples are weighted from accumulating more than one prediction signals from multiple hypothetical motion data.

- c. For example, at least one prediction sample of a GPM prediction block may be refined by a local illumination compensation (e.g., LIC) technique in which a linear model is used to compensate illumination change for the motion compensated luma samples.
 - d. For example, at least one prediction sample of a GPM prediction block may be refined by a Combined Inter-Intra Prediction (CIIP) technique in which intra-prediction is used to refine the motion compensated luma samples.
 - e. For example, at least one prediction sample of a GPM prediction block may be refined by a bi-directional optical-flow based motion refinement (e.g., BDOF or BIO) technique in which a pixel-wise motion refinement performed on top of block-wise motion compensation in a case of bi-prediction.
 - 1) For example, only if the two motion vectors of the two parts of a GPM block are from two different directions, the bi-directional optical-flow based motion refinement may be performed.
2. In one example, OBMC may be performed for all subblocks of a block coded with GPM.
- a. Alternatively, OBMC may be performed for some subblocks or some samples of a block coded with GPM.
 - 1) For example, OBMC may only be performed for subblocks at block boundaries of a block when the block is coded with GPM.
 - 2) For example, OBMC may only be performed for samples at block boundaries of a block when the block is coded with GPM.
3. In one example, when performing OBMC to a GPM block, the OBMC is applied based on the stored subblock (e.g., 4x4) based motion data of the current and neighboring GPM coded blocks.
- a. For example, the OBMC blending weights are determined based on the motion similarities between the stored subblock based motion of the current GPM subblock and the motion of the neighbor subblocks.
 - b. Alternatively, in such case, the OBMC may be applied based the motion data derived from the GPM merge candidates (e.g., without considering the subblock based GPM motion derived from the motion index of each subblock), rather than the stored subblock based motion of a GPM block.
4. In one example, whether to apply a feature/tool on top of GPM block may be dependent on the temporal layer identifier (e.g., layer ID) of the current picture among the group of pictures (GOP) structure.
- a. For example, the aforementioned feature/tool may be based on any of the following techniques:
 - 1) MMVD
 - 2) OBMC
 - 3) MHP
 - 4) LIC
 - 5) CIIP
 - 6) Non-adjacent spatial merge candidate

- 7) Decoder side motion refinement/derivation (e.g., template matching, bilateral matching, etc.)
- b. For example, a feature/tool may be applied to a GPM block when the current picture locates at pre-defined layer IDs, without extra signalling.
- c. For example, pictures of what layer IDs would have a feature/tool on a GPM block, may be explicit signalled.
5. In one example, in case motion vector difference is allowed to be used to a GPM block (named as GMVD), suppose M merge candidates are allowed for GPM without motion vector difference (named as GPM), and N merge candidates are allowed for GMVD, the following approaches are disclosed:
 - a. In one example, the maximum allowed merge candidates' number of GMVD may be different from that of GPM without motion vector difference.
 - 1) For example, M may be greater than N.
 - a) Alternatively, the maximum allowed merge candidates' numbers of GMVD and GPM are the same (e.g., $M = N$).
 - b) Alternatively, M may be less than N.
 - 2) For example, the maximum allowed merge candidates' numbers of a GMVD coded block may be signalled in the bitstream, e.g., by a syntax element.
 - a) Alternatively, the maximum allowed merge candidates' numbers of a GMVD coded block may be a predefined fixed value, such as $N = 2$.
 - 3) The signalling of GPM merge candidates index (e.g., `merge_gpm_idx0`, `merge_gpm_idx1`) may be dependent on whether GMVD is used for the current video unit.
 - a) For example, whether the current video block uses GMVD or not may be signalled before the GPM merge candidate index signalling.
 - b) For example, when the current video block uses GMVD (e.g., either part of a GPM block uses GMVD), then the input parameters (e.g., `cMax`) for GPM merge candidate index binarization may be based on the maximum allowed merge candidates number of GMVD (e.g., N).
 - c) For example, when the current video block doesn't use GMVD (e.g., both parts of a GPM block don't use GMVD), then the input parameters (e.g., `cMax`) for GPM merge candidate index binarization may be based on the maximum allowed merge candidates number of GPM without motion vector difference (e.g., N).
 - 4) In one example, a first syntax element (SE) to indicate whether GMVD is applied may depend on at least one GPM merge candidate index.
 - a) For example, the first SE may not be signaled if the largest GPM merge candidate index signaled for the current block is larger than a threshold.
 - b) For example, the first SE may not be signaled if the smallest GPM merge candidate index signaled for the current block is smaller than a threshold.

- c) If the first SE is not signaled, it may be inferred that GMVD is applied.
 - d) If the first SE is not signaled, it may be inferred that GMVD is not applied.
- b. In one example, GMVD may select base candidate(s) from the K (such as $K \leq M$) GPM merge candidates, and then add a motion vector difference on that base candidate.
 - 1) For example, the K GPM merge candidates may be the first K candidates in the list.
 - 2) For example, $K = 2$.
 - 3) For example, the base candidate index of a GPM block/part may be signalled, and its binarization input parameter cMax may be determined based on the value of K.
 - 4) For example, multiple parts (e.g. all parts) of a GPM block may share a same base candidate.
 - 5) For example, each part of a GPM block uses its own base candidate.
- c. In one example, not all the MVD parameters for a GPM block (e.g., the MVD distances and MVD directions) of two parts of a GMVD block are signalled.
 - 1) In one example, the MVD parameters of a first part of a GPM block may be signalled.
 - a) For example, the MVD parameters of the second part of a GPM block may be derived, e.g., based on the signalled MVD of the first part.
 - b) For example, the method that only signal MVD for one of the two parts of a GPM block may be based on a rule.
 - a) For example, the rule may be dependent on whether the motions of the two parts are pointing to different directions.
 - b) For example, the rule may be dependent on whether two parts of a GPM block are coded with GMVD.
 - 2) For example, if the base candidate of GMVD is a bi-prediction candidate, the MVD parameters may be signalled for a first prediction direction.
 - a) For example, the MVD derived from the signalled MVD parameters (such as MVD direction and MVD offset) may be applied to the LX motion, wherein $X = 0$ or 1 , while the $L(1-X)$ motion is derived, e.g., based on the signalled MVD of the first prediction direction LX.
 - 3) For example, the derivation of MVD in the second part/direction may be based on a scaled or a mirrored style.
 - a) For example, the derived MVD direction is based on mirroring the signalled MVD direction.
 - a) For example, suppose the first signalled GMVD direction index (for the first part or prediction direction of a GMVD block) can be interpreted by `gmvdSign[0][0]` and `gmvdSign[0][1]` in horizontal direction and vertical direction, respectively. Therefore, the second derived GMVD direction (for the second part or prediction direction of a GMVD block) in horizontal may be equal to an opposite direction (such as `gmvdSign[1][0] = -gmvdSign[0][0]`)

- and/or the second derived GMVD direction in vertical may be equal to an opposite vertical direction (such as $\text{gmvdSign}[1][1] = -\text{gmvdSign}[0][1]$).
- b) For example, at least one GMVD direction (e.g., horizontal or vertical) of the second derived GMVD direction is opposite to those interpreted from the first signalled GMVD direction index.
 - b) For example, the scaling factor of L(1-X) MVD offset is derived based on the POC distances of current-picture-to-L0-reference and current-picture-to-L1-reference.
 - a) For example, suppose the first signalled GMVD distance (for the first part or prediction direction of a GMVD block) is denoted by $\text{gmvdDistance}[0]$, and the POC distance between the first motion's reference picture and the current GMVD block is denoted by $\text{PocDiff}[0]$, and, the POC distance between the second motion's reference picture and the current GMVD block is denoted by $\text{PocDiff}[1]$. Then the derived GMVD distance, $\text{gmvdDistance}[1]$, may be derived based on $\text{PocDiff}[0]$, $\text{PocDiff}[1]$, and $\text{gmvdDistance}[0]$.
 - i. For example, $\text{gmvdDistance}[1] = (\text{gmvdDistance}[0] \gg a) \ll b$, wherein the value a is dependent on $\text{PocDiff}[0]$, and the value b is dependent on $\text{PocDiff}[1]$.
 - ii. For example, $\text{gmvdDistance}[1] = (\text{gmvdDistance}[0] \ll b) / a$, wherein the value a is dependent on $\text{PocDiff}[0]$, and the value b is dependent on $\text{PocDiff}[1]$.
 - 4) Alternatively, both LX and L(1-X) MVD offset are directly derived from the signalled MVD offset (e.g., without scaling or mirroring).
 - a) For example, the second derived GMVD distance is equal to the first signalled GMVD distance, e.g., $\text{gmvdDistance}[1] = \text{gmvdDistance}[0]$.
 - d. In one example, more than one set of GMVD tables (e.g., GMVD directions, and/or GMVD offsets) may be defined for GPM mode.
 - 1) For example, which set of GMVD tables is allowed/used for a video unit may be explicitly signalled.
 - 2) For example, which set of GMVD tables is allowed/used for a video unit may be hard coded based on a pre-defined rule (such as picture resolutions).
 - e. In one example, the final motion vector (e.g., GPM merge candidate plus the MVD offset) of at least one of the two GMVD parts must be different from the final MV of any one of the GPM merge candidate (which may be added by an MVD) in the GPM merge list.
 - 1) Alternatively, furthermore, the final motion vector of both GMVD parts are not allowed to be same with any of the GPM merge candidate in the GPM merge list.
 - 2) For example, if the final MV is the same to that of another GPM merge candidate, the final MV may be modified.
 - 3) For example, if the final MV is the same to that of another GPM merge candidate, the specific GPM merge candidate or MVD may be not allowed to be signaled.

- f. In one example, the final motion vectors of the two GMVD parts must be different from each other.
- 1) Alternatively, the final motion vectors of the two GMVD parts may be the same but different from any one of the GPM merge candidate in the GPM merge list.
 - 2) For example, if the final MV of a part is the same to that of the other part, the final MV may be modified.
 - 3) For example, if the final MV of a first part is the same to that of the other part, the specific GPM merge candidate or MVD of the first part may be not allowed to be signaled.

3. Problems

There are several flaws in the VVC v1 standard, which would be further improved for higher coding gain.

- 1) In the VVC v1 standard, the motion data of some type of coded blocks (such as CIIP, GPM, Affine, MMVD, and SbTMVP etc.) is generated from a merge/AMVP candidate, without motion refinement. Considering the motion refinement before or after the motion compensation (e.g., MMVD, decoder side motion derivation/refinement such as DMVR, FRUC, template matching TM merge, TM AMVP and etc.), it would be more efficient if a motion vector of such kind of coding block is refined.
- 2) The prediction mode of some type of coded blocks (such as intra mode in the CIIP, regular intra mode etc.) may be refined using the decoded information, in order to generate more precise prediction.
- 3) The prediction samples of some type of coded blocks (such as AMVP, GPM, CIIP, SbTMVP, Affine, MMVD, DMVR, FRUC, TM merge, TM AMVP and etc.) may be refined using the decoded information (e.g., BDOF, OBMC, and etc.), in order to generate more precise prediction.
- 4) For new coding techniques (e.g., multi-hypothesis prediction, MHP, and etc.) introduced beyond VVC, the coding data (such as motion, mode, prediction samples) of the new coding tool coded video units may be further refined using the signalled/decoded information.

4. Embodiments of the present disclosure

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

The terms ‘video unit’ or ‘coding unit’ or ‘block’ may represent a coding tree block (CTB), a coding tree unit (CTU), a coding block (CB), a CU, a PU, a TU, a PB, a TB.

A block may be rectangular or non-rectangular.

In the disclosure, the phrase “regular motion candidate” may represent a merge motion candidate in a regular/extended merge list indicated by a merge candidate index, or an AMVP motion vector, or an AMVP motion candidate in regular/extended AMVP list indicated by an AMVP candidate index.

In the disclosure, a motion candidate is called to be “refined” if the motion information of the candidate is modified according to information signaled from the encoder or derived at the decoder. For example, a motion vector may be refined by DMVR, FRUC, TM merge, TM AMVP, MMVD, GMVD, affine MMVD BDOF and so on.

In the disclosure, the phrase “coding data refinement” may represent a refinement process in order to refine the signalled/decoded/derived prediction modes, prediction directions, or signalled/decoded/derived motion information, prediction and/or reconstruction samples for a video unit.

1. In one example, the coding data Z of a video unit coded by a particular coding technique X may be further refined by another process Y.
 - 1) For example, the coding data Z may be a signalled/decoded/derived prediction mode and/or prediction directions of the video unit.
 - 2) For example, the coding data Z may be signalled/decoded/derived motion information of the video unit.
 - a) In one example, the coding data Z may be the motion information of a given reference picture list X (X being 0 or 1).
 - 3) For example, the coding data Z may be prediction samples or reconstruction samples of a video unit.
 - 4) For example, the particular coding technique X may be an AMVP candidate-based technique.
 - 5) For example, the particular coding technique X may be a merge candidate-based technique.
 - 6) For example, the particular coding technique X may be CIIP, MMVD, GPM, MHP, and etc.
 - 7) For example, the particular coding technique X may be a whole-block-based technique wherein all samples in the video unit share the same coding information.
 - a) In one example, X may be regular merge, regular AMVP, CIIP, MHP, and etc.
 - 8) For example, the particular coding technique X may be a subblock-based technique wherein two sub-blocks in the video unit may use different coding information.
 - a) In one example, X may be Affine, SbTMVP, and etc.
 - b) In one example, X may be ISP, and etc.
 - c) In one example, X may be GPM, GEO, TPM, and etc.
 - 9) For example, the particular coding technique X may be an inter prediction-based technique.
 - 10) For example, the particular coding technique X may be an intra prediction-based technique such as regular intra mode, MIP, CIIP, ISP, LM, IBC, BDPCM, and etc.
 - 11) For example, the particular refinement process Y may be based on explicitly signalling based method such as signalling motion vector differences, or intra mode delta values, or prediction and/or reconstruction block/sample delta values in a bitstream.
 - a) In one example, for a particular coding technique X1 coded video unit, delta information may be explicitly signalled in a bitstream.

- i. Alternatively, for another particular coding technique X2 coded video unit, delta information may be derived by using decoded/reconstructed information which is available at the decoder side.
 - ii. For example, the delta information may be one or more motion vector difference.
 - a) For example, one or more motion vector differences may be added to a X-coded video unit.
 - b) For example, more than one look up tables may be defined in a codec, to derive the actual motion vector difference for different MMVD based coding techniques.
 - c) For example, a unified look up table may be defined in a codec, for all different MMVD based coding techniques.
 - iii. For example, the delta information may be a delta value which can be used to generate a new prediction mode by adding up this delta value to a signalled/derived prediction mode.
 - a) For example, the intra mode information of a video unit coded by CIIP (or, ISP, or regular Intra angular mode, or regular intra mode, and etc) may be refined by adding the delta value to a signalled/derived prediction mode.
 - iv. For example, the delta information may be one or more delta values which can be used to generate one or more new prediction and/or reconstruction sample values.
 - b) For example, the particular refinement process Y may be based on a filtering method.
 - i. In one example, at least one filtering parameter is signaled to the decoder.
 - ii. In one example, at least one filtering parameter is derived at the decoder.
- 12) For example, the particular refinement process Y may be based on implicitly derivation related techniques.
- a) In one example, the Y may be based on the motion information of neighbouring video units (adjacent, or non-adjacent).
 - i. In one example, the Y may be the OBMC process.
 - b) For example, the particular refinement process Y may be based on a bilateral matching method such as DMVR which measures the prediction sample difference between L0 prediction block and L1 prediction block.
 - c) In one example, the Y may be based on the reconstruction samples of neighbouring video units (adjacent, or non-adjacent).
 - i. For example, the particular refinement process Y may be based on templated matching related techniques such as FRUC, TM merge, TM AMVP, TM IBC, BDOF, and etc.
 - a) For example, the template may be constructed based on neighboring reconstructed samples on the top and/or left neighboring of the video unit, and, prediction/reconstructed samples at predefined locations in the reference area (e.g., within the current picture, or in a reference picture).

- b) For example, the reference samples of the template in the reference area may be derived based on a subblock based motion (e.g., each reference sub-template may be retrieved with individual motion information).
 - c) For example, the reference samples of the template in the reference area may be derived based on single motion information.
 - d) For example, whether the template matching is done in a way of uni-prediction or bi-prediction, may be dependent on the signalled motion information.
 - i. For example, if the decoded/signalled motion information indicates that the current video unit is uni-predicted, the template matching based refinement may be conducted in a uni-prediction way (e.g., optimizing the motion vector according to a criteria based on the differences between a uni-predicted reference template and the template in the current picture).
 - ii. For example, if the decoded/signalled motion information indicates that the current unit is bi-predicted, the template matching based refinement may be conducted in a bi-prediction way (e.g., optimizing the motion vector according to a criteria based on the differences between more than one reference templates/the combination of more than one reference templates and the template in the current picture).
 - iii. For example, the template matching based refinement may be always conducted in a bi-prediction way, regardless the prediction direction obtained from the decoded/signalled motion information.
 - 1. Furthermore, alternatively, whether to take this solution may be dependent on the type of the coding technique X that applied the video unit.
 - iv. For example, the template matching based refinement may be always conducted in a uni-prediction way, regardless the prediction direction obtained from the decoded/signalled motion information.
 - 1. Furthermore, alternatively, whether to take this solution may be dependent on the type of the coding technique X that applied the video unit.
2. For one video unit, multiple refinement processes may be applied.
- 1) In one example, at least two refinement processes may be applied wherein each of the two is used to refine one kind of coding data.
 - a) In one example, the motion information and intra prediction modes may be both refined.
 - i. Alternatively, furthermore, the above methods may be applied to CIIP-coded blocks.
 - ii. Alternatively, furthermore, the above methods may be applied to video unit with inter and intra combined prediction mode.
 - 2) In one example, at least two refinement processes may be applied wherein both of them are used to refine the same kind of coding data.

- a) In one example, the motion information may be refined using multiple ways, e.g., DMVR and TM based methods.
 - b) Alternatively, furthermore, the final refined motion information to be applied may be further determined from the temporary refined motion information from the multiple ways.
 - a) In one example, the temporary refined motion information from one among the multiple ways may be utilized as the final refined motion information to be applied.
 - c) Alternatively, furthermore, the final reconstruction/prediction block generation process may be dependent on the temporary refined motion information from the multiple ways.
- 3. In one example, the refinement process may be applied to one or more parts within a video unit.
 - 1) For example, the refinement process may be applied to a video unit in a whole-block-based way (e.g., the coding data of the whole CU may be refined).
 - 2) For example, the refinement process may be applied to a video unit in a subblock/part/partition-based way.
 - a) For example, the refinement process may be applied to one or more parts/partitions of a video unit (in case of the coding unit contains more than one parts/partitions) rather than all partitions of the coding unit.
 - b) For example, the refinement process may be applied to one or more subblocks of the coding unit rather than the entire coding unit.
 - i. For example, the subblocks may be represented by a size of $M \times N$ (such as $M=N=4$ or 8) samples which is less than the size of the whole coding unit.
 - ii. For example, the subblocks at pre-defined positions (such as above or left border) may be taken into account.
 - c) Alternatively, the refinement process may be applied to all parts/partitions/subblocks of the coding unit.
 - d) In one example, whether to and/or how to apply a refinement process on a sub-block may depend on the location of the sub-block.
 - i. For example, a first refinement process is applied on a sub-block at boundaries of a block, and a second refinement process is applied on a sub-block not at boundaries of a block.
 - e) In one example, the refined results of a first subblock may be used to refine a second subblock of the block.
 - i. Alternatively, the refined results of a first subblock cannot be used to refine a second subblock of the block.
- 4. In one example, whether or not apply the refinement process to a video unit and/or how to apply the refinement process may be controlled by one or multiple syntax elements (e.g., a flag).
 - 1) For example, whether to signal syntax elements related to a refinement process Y may be dependent on the type of the coding technique X that applied the video unit.

- a) For example, for a particular coding technique X1 coded video unit, whether to use refinement process Y1 may be indicated by a syntax flag.
 - b) Alternatively, for another particular coding technique X2 coded video unit, a refinement process Y2 may be mandatorily applied, without explicit signalling.
- 5. In one example, whether to use refined coding data or original coding data (before being refined) for processing the current video unit and/or proceeding video units, may be dependent on what coding technique X is applied to the video unit.
 - 1) In one example, a refined motion of a video unit may be used to generate the motion compensated prediction samples.
 - a) Alternatively, the original motion without the refinement may be used to generate the motion compensated prediction samples for a video unit.
 - 2) In one example, a refined motion of a video unit may be used to determine parameters in a loop filter process.
 - a) For example, the refined motion may be used for the deblocking strength determination for a video unit.
 - b) Alternatively, the original motion without the refinement may be used for the deblocking strength determination for a video unit.
 - 3) In one example, refined coding data of a first video unit may be stored for coding information derivation for a second video unit.
 - a) For example, the refined motion vector may be stored in a $M \times N$ (such as $M=N=4$ or 8 or 16) subblock basis.
 - i. Alternatively, the refined motion vector may be stored in a CU basis.
 - b) For example, the refinement motion vector for a first video unit may be stored for the spatial motion candidate derivation for a second video unit.
 - i. Alternatively, the original motion without the refinement for a first video unit may be stored for the spatial motion candidate derivation for a second video unit.
 - c) For example, the refinement motion vector for a first video unit may be stored for the temporal motion candidate derivation for a second video unit.
 - d) For example, the refinement intra prediction mode for a first video unit may be stored for the intra MPM list generation for a second video unit.
- 6. In one example, whether to and/or how to apply a refinement process may depend on color format and/or color component.
 - 1) In one example, a refinement process is applied on a first color component but not on a second color component.
- 7. In one example, whether to and/or how to apply a refinement process may depend on dimensions $W \times H$ of a block.
 - 1) For example, a refinement process may not be applied if $W \geq T1$ and/or $H \geq T2$.
 - 2) For example, a refinement process may not be applied if $W \leq T1$ and/or $H \leq T2$.
 - 3) For example, a refinement process may not be applied if $W > T1$ and/or $H > T2$.

- 4) For example, a refinement process may not be applied if $W < T1$ and/or $H < T2$.
- 5) For example, a refinement process may not be applied if $W \times H \geq T$.
- 6) For example, a refinement process may not be applied if $W \times H > T$.
- 7) For example, a refinement process may not be applied if $W \times H \leq T$.
- 8) For example, a refinement process may not be applied if $W \times H < T$.

[0073] Fig. 17 illustrates a flowchart of a method 1700 for video processing in accordance with some embodiments of the present disclosure. As shown in Fig. 17, the method 1700 comprises: obtaining 1702, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determining 1704 the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and performing 1706 the conversion based on the determination.

[0074] According to the method 1700, whether to use refined coding data or original coding data before refined for processing the current video unit and/or proceeding video units, may be dependent on what coding technique is applied to the respective video units. As such, the coding efficiency can be improved. In a case where the refined coding data is used, prediction samples of the coded block can be refined by using the signaled or decoded information, which generates more precise prediction. Compared with the conventional solution, the method 1700 in accordance with some embodiments of the present disclosure can advantageously improve the coding performance and efficiency.

[0075] In some embodiments, the second coding data may comprise refined motion information of the first video unit, and the refined motion information is used for generating motion compensated prediction samples the first video unit.

[0076] In some embodiments, the first coding data may comprise original motion information of the first video unit without refinement, and the original motion information is used for generating motion compensated prediction samples the first video unit.

[0077] In some embodiments, the second coding data may comprise refined motion information of the first video unit, and the refined motion information is used for determining

parameters in a loop filter process for the video.

[0078] In some embodiments, the refined motion information may be used for deblocking strength determination for the first video unit.

[0079] In some embodiments, the first coding data may comprise original motion information of the first video unit without refinement, and the original motion information is used for deblocking strength determination for the first video unit.

[0080] In some embodiments, the second coding data may comprise refined motion information of the first video unit, and the refined motion information is used for deriving motion information of the second video unit in the target picture.

[0081] In some embodiments, the refined motion information may comprise a refined motion vector of the first video unit and is stored on a subblock basis. By way of example, the refined motion vector may be stored on a $M \times N$ subblock basis, where M may be equal to N , and a value of M and N may be 4, 8 or 16. It should be understood that the values and configurations are given for the illustrative purpose, and any other suitable values and configurations are also applicable to implementations of the present disclosure. Therefore, the scope of the present disclosure is not limited in this regard.

[0082] In some embodiments, the refined motion information may comprise a refined motion vector of the first video unit and is stored on a coded unit (CU) basis.

[0083] In some embodiments, the refined motion information may comprise a refined motion vector of the first video unit and is stored for deriving spatial motion candidate of the second video unit.

[0084] In some embodiments, the first coding data may comprise original motion information of the first video unit without refinement, and the original motion information is stored for deriving spatial motion candidate of the second video unit.

[0085] In some embodiments, the refined motion information may comprise a refined motion vector of the first video unit and is stored for deriving a temporal motion candidate of the second video unit.

[0086] In some embodiments, the refined motion information may comprise a refinement intra prediction mode for the first video unit and is stored for generating an intra most probable mode (MPM) list of the second video unit.

[0087] In some embodiments, the target coding mode may be based on an adaptive motion vector resolution prediction (AMVP) candidate-based coding technique.

[0088] In some embodiments, the target coding mode may be based on a merge candidate-based coding technique.

[0089] In some embodiments, the target coding mode may comprise one of a combined inter-intra prediction (CIIP) mode, a merge mode with motion vector differences (MMVD), a geometric partitioning mode (GPM), or a multi-hypothesis prediction (MHP) mode.

[0090] In some embodiments, the target coding mode may be based on a whole-block-based coding technique wherein all samples of the target video unit have the same coding information.

[0091] In some embodiments, the whole-block-based coding technique may comprise one of a regular merge mode, a regular adaptive motion vector resolution prediction (AMVP) mode, a combined inter-intra prediction (CIIP) mode, or a multi-hypothesis prediction (MHP) mode.

[0092] In some embodiments, the target coding mode may be based on a subblock-based coding technique wherein at least two of sub-blocks in the target video unit have different first coding data.

[0093] In some embodiments, the target coding mode may comprise one of an affine mode or a subblock-based temporal motion vector prediction (SbTMVP) mode.

[0094] In some embodiments, the target coding mode may comprise an intra sub-partitions (ISP) mode.

[0095] In some embodiments, the target coding mode may comprise one of a geometric partitioning mode (GPM), a geometric merge mode (GEO), a triangular prediction mode (TPM).

[0096] In some embodiments, the target coding mode may be based on an inter prediction-based technique.

[0097] In some embodiments, the target coding mode may be based on an intra prediction-

based technique and comprise one of an intra coding mode, a matrix weighted intra prediction (MIP) mode, a combined inter-intra prediction (CIIP) mode, an intra sub-partitions (ISP) mode, a linear model (LM) mode, an intra block copy (IBC) mode, or a block-based differential pulse-code modulation (BDPCM).

[0098] As a result, the prediction samples of some type of coded blocks (such as AMVP, GPM, CIIP, SbTMVP, Affine, MMVD, DMVR, FRUC, TM merge, TM AMVP and etc.) may be refined by using the decoded information (e.g., BDOF, OBMC, and etc.), which generates more precise prediction.

[0099] In some embodiments, the refinement process may be based on a method explicitly indicated in the bitstream.

[00100] In some embodiments, the refinement process may be based on delta information of the target video unit, and the delta information may comprise one of the following at least one motion vector difference, at least one intra mode delta value, at least one prediction block or sample delta value, or at least one reconstruction block or sample delta value.

[00101] In some embodiments, the refinement process may be based on at least one filtering parameter for filtering the first coding data.

[00102] In some embodiments, the refinement process may be based on motion information of at least one neighboring video unit, and the at least one neighboring video unit comprises at least one of video units adjacent or non-adjacent to the target video unit.

[00103] In some embodiments, the refinement process may be based on an overlapped block-based motion compensation (OBMC) technique.

[00104] In some embodiments, the refinement process may be based on a bilateral matching technique comprising at least a decoder side motion vector refinement (DMVR) mode.

[00105] In some embodiments, the refinement process may comprise the DMVR mode, and the second coding data comprises a prediction sample difference between a L0 prediction block and a L1 prediction block of the video unit.

[00106] In some embodiments, the refinement process may be based on reconstruction

samples of at least one neighboring video unit, and the at least one neighboring video unit comprises at least one of video units adjacent or non-adjacent to the target video unit.

[00107] In some embodiments, the refinement process may be based on a templated matching related technique comprising one of a frame-rate up conversion (FRUC) mode, TM merge, a temporal motion (TM) mode, an adaptive motion vector resolution prediction (AMVP) mode, a TM intra block copy (IBC) mode, or a bi-directional optical flow (BDOF) mode.

[00108] In some embodiments, the conversion may comprise decoding the target picture from the bitstream of the video.

[00109] In some embodiments, the conversion may comprise encoding the target picture into the bitstream of the video.

[00110] It should be understood that values, parameters, or configurations in the above embodiments are given for the illustrative purpose, and any other suitable values, parameters, or configurations are also applicable to the implementations of the present disclosure. Therefore, the scope of the present disclosure is not limited in this regard.

[00111] Implementations of the present disclosure can be described in view of the following clauses, the features of which can be combined in any reasonable manner.

[00112] Clause 1. A method for video processing, comprising: obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and performing the conversion based on the determination.

[00113] Clause 2. The method of clause 1, wherein the second coding data comprises refined motion information of the first video unit, and the refined motion information is used for generating motion compensated prediction samples the first video unit.

[00114] Clause 3. The method of clause 1, wherein the first coding data comprises original motion information of the first video unit without refinement, and the original motion

information is used for generating motion compensated prediction samples the first video unit.

[00115] Clause 4. The method of clause 1, wherein the second coding data comprises refined motion information of the first video unit, and the refined motion information is used for determining parameters in a loop filter process for the video.

[00116] Clause 5. The method of clause 4, wherein the refined motion information is used for deblocking strength determination for the first video unit.

[00117] Clause 6. The method of clause 4, wherein the first coding data comprises original motion information of the first video unit without refinement, and the original motion information is used for deblocking strength determination for the first video unit.

[00118] Clause 7. The method of clause 1, wherein the second coding data comprises refined motion information of the first video unit, and the refined motion information is used for deriving motion information of the second video unit in the target picture.

[00119] Clause 8. The method of clause 7, wherein the refined motion information comprises a refined motion vector of the first video unit and is stored on a subblock basis.

[00120] Clause 9. The method of clause 7, wherein the refined motion information comprises a refined motion vector of the first video unit and is stored on a coded unit (CU) basis.

[00121] Clause 10. The method of clause 7, wherein the refined motion information comprises a refined motion vector of the first video unit and is stored for deriving spatial motion candidate of the second video unit.

[00122] Clause 11. The method of clause 1, wherein the first coding data comprises original motion information of the first video unit without refinement, and the original motion information is stored for deriving spatial motion candidate of the second video unit.

[00123] Clause 12. The method of clause 7, wherein the refined motion information comprises a refined motion vector of the first video unit and is stored for deriving a temporal motion candidate of the second video unit.

[00124] Clause 13. The method of clause 7, wherein the refined motion information comprises a refinement intra prediction mode for the first video unit and is stored for generating

an intra most probable mode (MPM) list of the second video unit.

[00125] Clause 14. The method of clause 1, wherein the target coding mode is based on an adaptive motion vector resolution prediction (AMVP) candidate-based coding technique.

[00126] Clause 15. The method of clause 1, wherein the target coding mode is based on a merge candidate-based coding technique.

[00127] Clause 16. The method of clause 1, wherein the target coding mode comprises one of a combined inter-intra prediction (CIIP) mode, a merge mode with motion vector differences (MMVD), a geometric partitioning mode (GPM), or a multi-hypothesis prediction (MHP) mode.

[00128] Clause 17. The method of clause 1, wherein the target coding mode is based on a whole-block-based coding technique wherein all samples of the target video unit have the same coding information.

[00129] Clause 18. The method of clause 17, wherein the whole-block-based coding technique comprises one of a regular merge mode, a regular adaptive motion vector resolution prediction (AMVP) mode, a combined inter-intra prediction (CIIP) mode, or a multi-hypothesis prediction (MHP) mode.

[00130] Clause 19. The method of clause 1, wherein the target coding mode is based on a subblock-based coding technique wherein at least two of sub-blocks in the target video unit have different first coding data.

[00131] Clause 20. The method of clause 19, wherein the target coding mode comprises one of an affine mode or a subblock-based temporal motion vector prediction (SbTMVP) mode.

[00132] Clause 21. The method of clause 19, wherein the target coding mode comprises an intra sub-partitions (ISP) mode.

[00133] Clause 22. The method of clause 19, wherein the target coding mode comprises one of a geometric partitioning mode (GPM), a geometric merge mode (GEO), a triangular prediction mode (TPM).

[00134] Clause 23. The method of clause 1, wherein the target coding mode is based on an inter prediction-based technique.

[00135] Clause 24. The method of clause 1, wherein the target coding mode is based on an intra prediction-based technique and comprises one of an intra coding mode, a matrix weighted intra prediction (MIP) mode, a combined inter-intra prediction (CIIP) mode, an intra sub-partitions (ISP) mode, a linear model (LM) mode, an intra block copy (IBC) mode, or a block-based differential pulse-code modulation (BDPCM).

[00136] Clause 25. The method of clause 1, wherein the refinement process is based on a method explicitly indicated in the bitstream.

[00137] Clause 26. The method of clause 1, wherein the refinement process is based on delta information of the target video unit, and the delta information comprises one of the following: at least one motion vector difference, at least one intra mode delta value, at least one prediction block or sample delta value, or at least one reconstruction block or sample delta value.

[00138] Clause 27. The method of clause 1, wherein the refinement process is based on at least one filtering parameter for filtering the first coding data.

[00139] Clause 28. The method of clause 1, wherein the refinement process is based on motion information of at least one neighboring video unit, and the at least one neighboring video unit comprises at least one of video units adjacent or non-adjacent to the target video unit.

[00140] Clause 29. The method of clause 28, wherein the refinement process is based on an overlapped block-based motion compensation (OBMC) technique.

[00141] Clause 30. The method of clause 1, wherein the refinement process is based on a bilateral matching technique comprising at least a decoder side motion vector refinement (DMVR) mode.

[00142] Clause 31. The method of clause 30, wherein the refinement process comprises the DMVR mode, and the second coding data comprises a prediction sample difference between a L0 prediction block and a L1 prediction block of the video unit.

[00143] Clause 32. The method of clause 1, wherein the refinement process is based on reconstruction samples of at least one neighboring video unit, and the at least one neighboring video unit comprises at least one of video units adjacent or non-adjacent to the target video unit.

[00144] Clause 33. The method of clause 32, wherein the refinement process is based on a templated matching related technique comprising one of a frame-rate up conversion (FRUC) mode, TM merge, a temporal motion (TM) mode, an adaptive motion vector resolution prediction (AMVP) mode, a TM intra block copy (IBC) mode, or a bi-directional optical flow (BDOF) mode.

[00145] Clause 34. The method of any of clauses 1 to 33, wherein the conversion comprises decoding the target picture into the bitstream of the video.

[00146] Clause 35. The method of any of clauses 1 to 33, wherein the conversion comprises encoding the target picture into the bitstream of the video.

[00147] Clause 36. An apparatus for video processing, comprising a processor and a non-transitory memory coupled to the processor and having instructions stored thereon, wherein the instructions upon execution by the processor, cause the processor to: obtain, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determine the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and generate the bitstream based on the determination.

[00148] Clause 37. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of Clauses 1-35.

[00149] Clause 38. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by an apparatus for video processing, wherein the method comprises: obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and generating the bitstream based on the determination.

[00150] Clause 39. A method for storing a bitstream of a video, comprising: obtaining, during

a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode; determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; generating the bitstream based on the determination; and storing the bitstream in a non-transitory computer-readable recording medium.

Example Device

[00151] Fig. 18 illustrates a block diagram of a computing device 1800 in which various embodiments of the present disclosure can be implemented. The computing device 1800 may be implemented as or included in the source device 110 (or the video encoder 114 or 200) or the destination device 120 (or the video decoder 124 or 300).

[00152] It would be appreciated that the computing device 1800 shown in Fig. 18 is merely for purpose of illustration, without suggesting any limitation to the functions and scopes of the embodiments of the present disclosure in any manner.

[00153] As shown in Fig. 18, the computing device 1800 includes a general-purpose computing device 1800. The computing device 1800 may at least comprise one or more processors or processing units 1810, a memory 1820, a storage unit 1830, one or more communication units 1840, one or more input devices 1850, and one or more output devices 1860.

[00154] In some embodiments, the computing device 1800 may be implemented as any user terminal or server terminal having the computing capability. The server terminal may be a server, a large-scale computing device or the like that is provided by a service provider. The user terminal may for example be any type of mobile terminal, fixed terminal, or portable terminal, including a mobile phone, station, unit, device, multimedia computer, multimedia tablet, Internet node, communicator, desktop computer, laptop computer, notebook computer, netbook computer, tablet computer, personal communication system (PCS) device, personal navigation device, personal digital assistant (PDA), audio/video player, digital camera/video

camera, positioning device, television receiver, radio broadcast receiver, E-book device, gaming device, or any combination thereof, including the accessories and peripherals of these devices, or any combination thereof. It would be contemplated that the computing device 1800 can support any type of interface to a user (such as “wearable” circuitry and the like).

[00155] The processing unit 1810 may be a physical or virtual processor and can implement various processes based on programs stored in the memory 1820. In a multi-processor system, multiple processing units execute computer executable instructions in parallel so as to improve the parallel processing capability of the computing device 1800. The processing unit 1810 may also be referred to as a central processing unit (CPU), a microprocessor, a controller or a microcontroller.

[00156] The computing device 1800 typically includes various computer storage medium. Such medium can be any medium accessible by the computing device 1800, including, but not limited to, volatile and non-volatile medium, or detachable and non-detachable medium. The memory 1820 can be a volatile memory (for example, a register, cache, Random Access Memory (RAM)), a non-volatile memory (such as a Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), or a flash memory), or any combination thereof. The storage unit 1830 may be any detachable or non-detachable medium and may include a machine-readable medium such as a memory, flash memory drive, magnetic disk or another other media, which can be used for storing information and/or data and can be accessed in the computing device 1800.

[00157] The computing device 1800 may further include additional detachable/non-detachable, volatile/non-volatile memory medium. Although not shown in Fig. 18, it is possible to provide a magnetic disk drive for reading from and/or writing into a detachable and non-volatile magnetic disk and an optical disk drive for reading from and/or writing into a detachable non-volatile optical disk. In such cases, each drive may be connected to a bus (not shown) via one or more data medium interfaces.

[00158] The communication unit 1840 communicates with a further computing device via the communication medium. In addition, the functions of the components in the computing device 1800 can be implemented by a single computing cluster or multiple computing machines

that can communicate via communication connections. Therefore, the computing device 1800 can operate in a networked environment using a logical connection with one or more other servers, networked personal computers (PCs) or further general network nodes.

[00159] The input device 1850 may be one or more of a variety of input devices, such as a mouse, keyboard, tracking ball, voice-input device, and the like. The output device 1860 may be one or more of a variety of output devices, such as a display, loudspeaker, printer, and the like. By means of the communication unit 1840, the computing device 1800 can further communicate with one or more external devices (not shown) such as the storage devices and display device, with one or more devices enabling the user to interact with the computing device 1800, or any devices (such as a network card, a modem and the like) enabling the computing device 1800 to communicate with one or more other computing devices, if required. Such communication can be performed via input/output (I/O) interfaces (not shown).

[00160] In some embodiments, instead of being integrated in a single device, some or all components of the computing device 1800 may also be arranged in cloud computing architecture. In the cloud computing architecture, the components may be provided remotely and work together to implement the functionalities described in the present disclosure. In some embodiments, cloud computing provides computing, software, data access and storage service, which will not require end users to be aware of the physical locations or configurations of the systems or hardware providing these services. In various embodiments, the cloud computing provides the services via a wide area network (such as Internet) using suitable protocols. For example, a cloud computing provider provides applications over the wide area network, which can be accessed through a web browser or any other computing components. The software or components of the cloud computing architecture and corresponding data may be stored on a server at a remote position. The computing resources in the cloud computing environment may be merged or distributed at locations in a remote data center. Cloud computing infrastructures may provide the services through a shared data center, though they behave as a single access point for the users. Therefore, the cloud computing architectures may be used to provide the components and functionalities described herein from a service provider at a remote location. Alternatively, they may be provided from a conventional server or installed directly or otherwise on a client device.

[00161] The computing device 1800 may be used to implement video encoding/decoding in embodiments of the present disclosure. The memory 1820 may include one or more video coding modules 1825 having one or more program instructions. These modules are accessible and executable by the processing unit 1810 to perform the functionalities of the various embodiments described herein.

[00162] In the example embodiments of performing video encoding, the input device 1850 may receive video data as an input 1870 to be encoded. The video data may be processed, for example, by the video coding module 1825, to generate an encoded bitstream. The encoded bitstream may be provided via the output device 1860 as an output 1880.

[00163] In the example embodiments of performing video decoding, the input device 1850 may receive an encoded bitstream as the input 1870. The encoded bitstream may be processed, for example, by the video coding module 1825, to generate decoded video data. The decoded video data may be provided via the output device 1860 as the output 1880.

[00164] While this disclosure has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present application as defined by the appended claims. Such variations are intended to be covered by the scope of this present application. As such, the foregoing description of embodiments of the present application is not intended to be limiting.

I/We Claim:

1. A method for video processing, comprising:
obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode;
determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and
performing the conversion based on the determination.
2. The method of claim 1, wherein the second coding data comprises refined motion information of the first video unit, and the refined motion information is used for generating motion compensated prediction samples the first video unit.
3. The method of claim 2, wherein the first coding data comprises original motion information of the first video unit without refinement, and the original motion information is used for generating motion compensated prediction samples the first video unit.
4. The method of claim 1, wherein the second coding data comprises refined motion information of the first video unit, and the refined motion information is used for determining parameters in a loop filter process for the video.
5. The method of claim 4, wherein the refined motion information is used for deblocking strength determination for the first video unit.
6. The method of claim 4, wherein the first coding data comprises original motion information of the first video unit without refinement, and the original motion information is used for deblocking strength determination for the first video unit.
7. The method of claim 1, wherein the second coding data comprises refined motion information of the first video unit, and the refined motion information is used for deriving motion information of the second video unit in the target picture.

8. The method of claim 7, wherein the refined motion information comprises a refined motion vector of the first video unit and is stored on a subblock basis.

9. The method of claim 7, wherein the refined motion information comprises a refined motion vector of the first video unit and is stored on a coded unit (CU) basis.

10. The method of claim 7, wherein the refined motion information comprises a refined motion vector of the first video unit and is stored for deriving spatial motion candidate of the second video unit.

11. The method of claim 1, wherein the first coding data comprises original motion information of the first video unit without refinement, and the original motion information is stored for deriving spatial motion candidate of the second video unit.

12. The method of claim 7, wherein the refined motion information comprises a refined motion vector of the first video unit and is stored for deriving a temporal motion candidate of the second video unit.

13. The method of claim 7, wherein the refined motion information comprises a refinement intra prediction mode for the first video unit and is stored for generating an intra most probable mode (MPM) list of the second video unit.

14. The method of claim 1, wherein the target coding mode is based on an adaptive motion vector resolution prediction (AMVP) candidate-based coding technique.

15. The method of claim 1, wherein the target coding mode is based on a merge candidate-based coding technique.

16. The method of claim 1, wherein the target coding mode comprises one of a combined inter-intra prediction (CIIP) mode, a merge mode with motion vector differences (MMVD), a geometric partitioning mode (GPM), or a multi-hypothesis prediction (MHP) mode.

17. The method of claim 1, wherein the target coding mode is based on a whole-block-based coding technique wherein all samples of the target video unit have the same coding information.

18. The method of claim 17, wherein the whole-block-based coding technique comprises one of a regular merge mode, a regular adaptive motion vector resolution prediction (AMVP) mode, a combined inter-intra prediction (CIIP) mode, or a multi-hypothesis prediction (MHP) mode.

19. The method of claim 1, wherein the target coding mode is based on a subblock-based coding technique wherein at least two of sub-blocks in the target video unit have different first coding data.

20. The method of claim 19, wherein the target coding mode comprises one of an affine mode or a subblock-based temporal motion vector prediction (SbTMVP) mode.

21. The method of claim 19, wherein the target coding mode comprises an intra sub-partitions (ISP) mode.

22. The method of claim 19, wherein the target coding mode comprises one of a geometric partitioning mode (GPM), a geometric merge mode (GEO), a triangular prediction mode (TPM).

23. The method of claim 1, wherein the target coding mode is based on an inter prediction-based technique.

24. The method of claim 1, wherein the target coding mode is based on an intra prediction-based technique and comprises one of an intra coding mode, a matrix weighted intra prediction (MIP) mode, a combined inter-intra prediction (CIIP) mode, an intra sub-partitions (ISP) mode, a linear model (LM) mode, an intra block copy (IBC) mode, or a block-based differential pulse-code modulation (BDPCM).

25. The method of claim 1, wherein the refinement process is based on a method explicitly indicated in the bitstream.

26. The method of claim 1, wherein the refinement process is based on delta information of the target video unit, and the delta information comprises one of the following:

- at least one motion vector difference,
- at least one intra mode delta value,
- at least one prediction block or sample delta value, or
- at least one reconstruction block or sample delta value.

27. The method of claim 1, wherein the refinement process is based on at least one filtering parameter for filtering the first coding data.

28. The method of claim 1, wherein the refinement process is based on motion information of at least one neighboring video unit, and the at least one neighboring video unit comprises at least one of video units adjacent or non-adjacent to the target video unit.

29. The method of claim 28, wherein the refinement process is based on an overlapped block-based motion compensation (OBMC) technique.

30. The method of claim 1, wherein the refinement process is based on a bilateral matching technique comprising at least a decoder side motion vector refinement (DMVR) mode.

31. The method of claim 30, wherein the refinement process comprises the DMVR mode, and the second coding data comprises a prediction sample difference between a L0 prediction block and a L1 prediction block of the video unit.

32. The method of claim 1, wherein the refinement process is based on reconstruction samples of at least one neighboring video unit, and the at least one neighboring video unit comprises at least one of video units adjacent or non-adjacent to the target video unit.

33. The method of claim 32, wherein the refinement process is based on a templated matching related technique comprising one of a frame-rate up conversion (FRUC) mode, TM

merge, a temporal motion (TM) mode, an adaptive motion vector resolution prediction (AMVP) mode, a TM intra block copy (IBC) mode, or a bi-directional optical flow (BDOF) mode.

34. The method of any of claims 1 to 33, wherein the conversion comprises decoding the target picture from the bitstream of the video.

35. The method of any of claims 1 to 33, wherein the conversion comprises encoding the target picture into the bitstream of the video.

36. An apparatus for video processing, comprising a processor and a non-transitory memory coupled to the processor and having instructions stored thereon, wherein the instructions upon execution by the processor, cause the processor to:

obtain, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode;

determine the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and

generate the bitstream based on the determination.

37. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of claims 1 to 35.

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

obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode;

determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode; and

generating the bitstream based on the determination.

39. A method for storing a bitstream of a video, comprising:

obtaining, during a conversion between a first video unit in a target picture of a video and a bitstream of the video, second coding data of the first video unit based on first coding data of the first video unit and a refinement process, the first coding data being coded by a target coding mode;

determining the use of the first coding data or the second coding data for processing a second video unit subsequent to the first video unit based on the target coding mode;

generating the bitstream based on the determination; and

storing the bitstream in a non-transitory computer-readable recording medium.

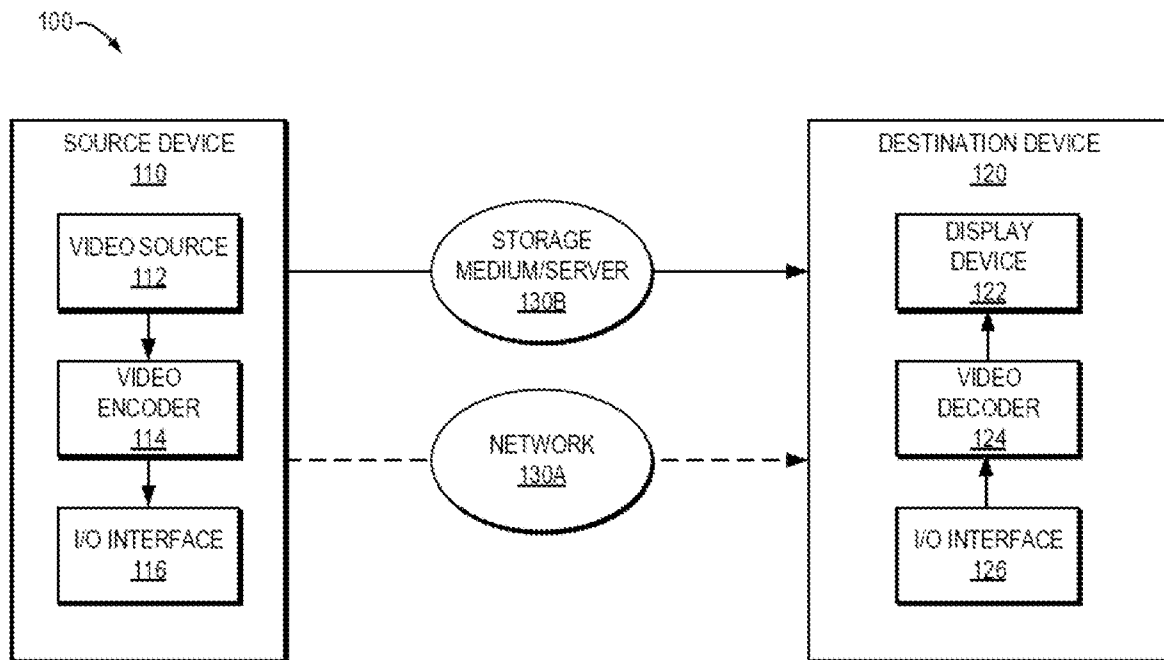


Fig. 1

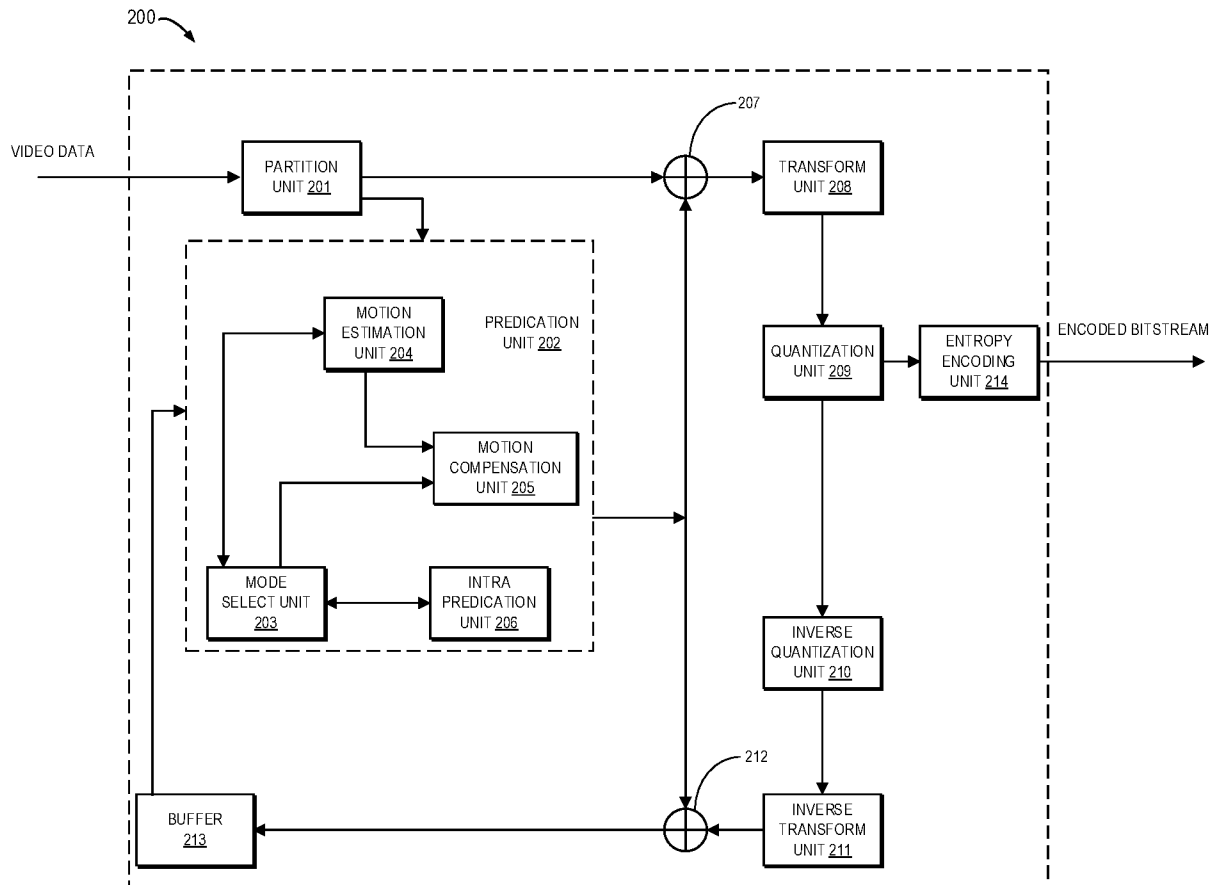
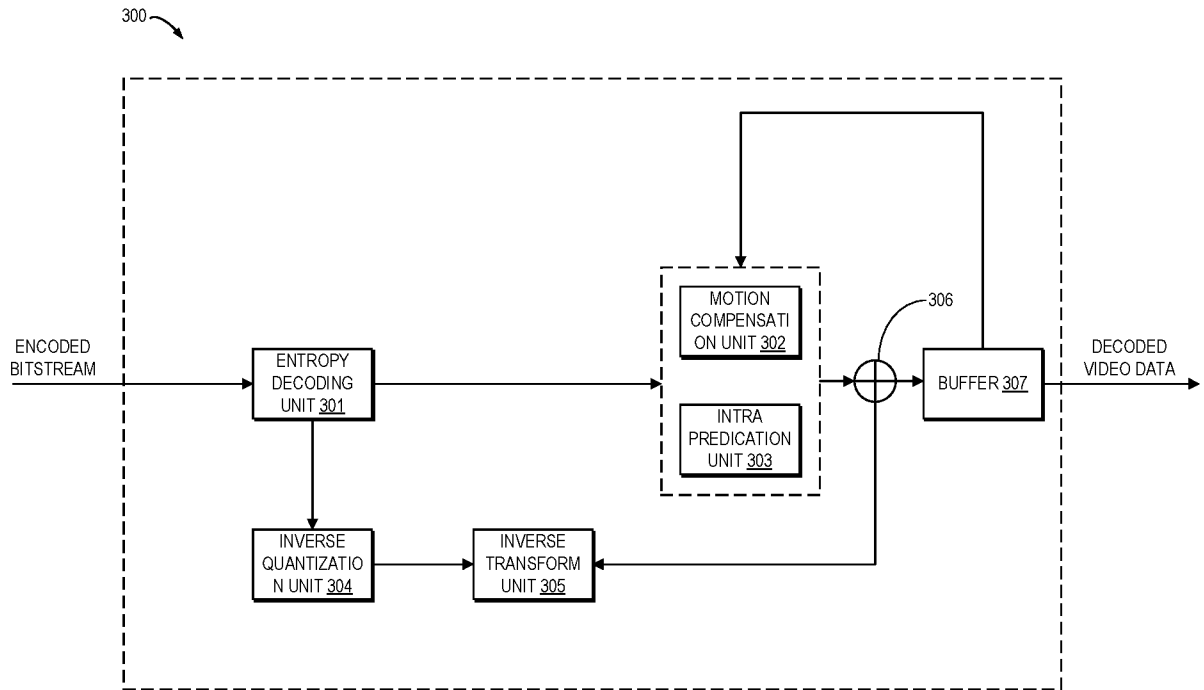


Fig. 2

**Fig. 3**

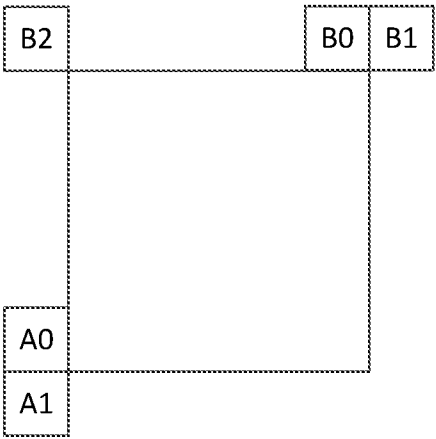


Fig. 4

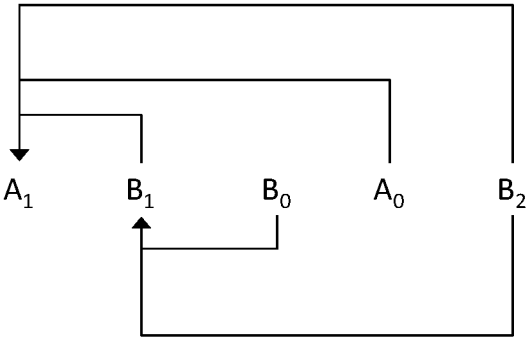


Fig. 5

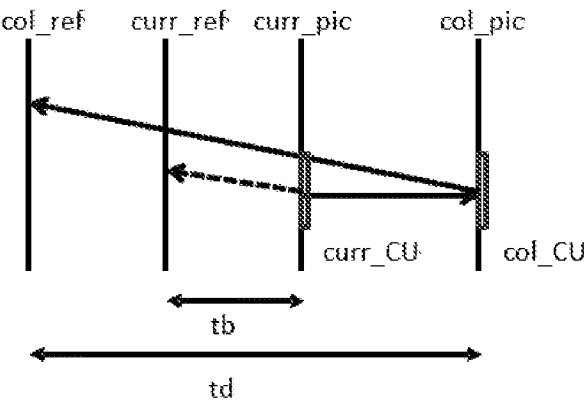


Fig. 6

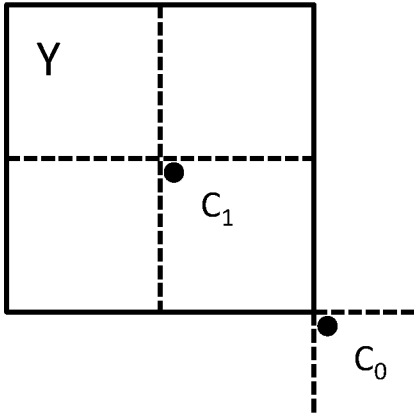


Fig. 7

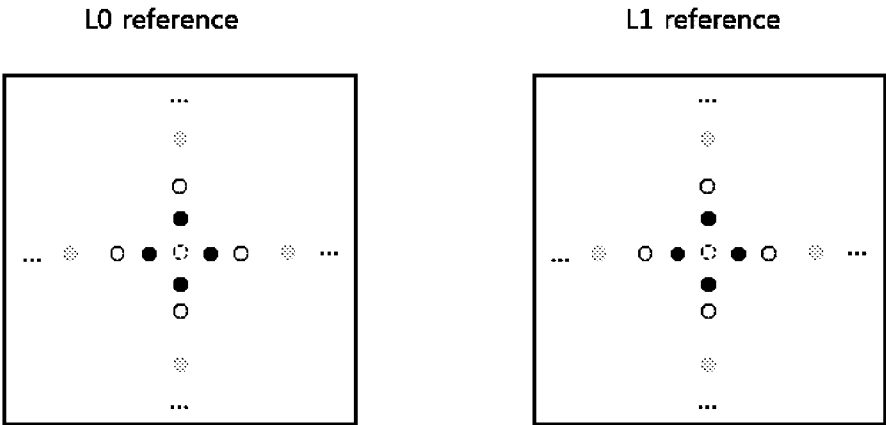


Fig. 8

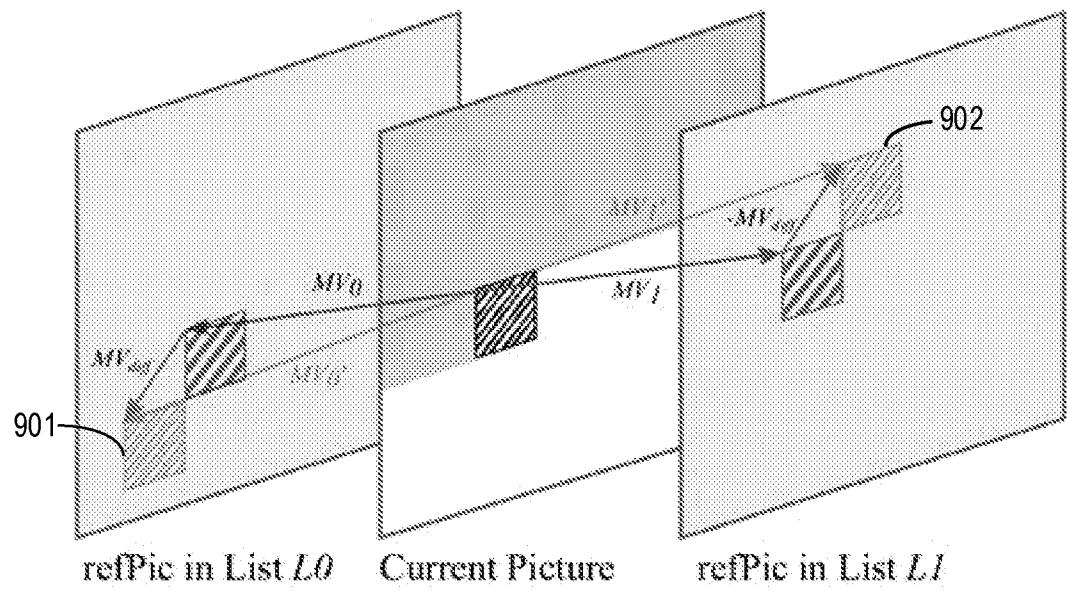


Fig. 9

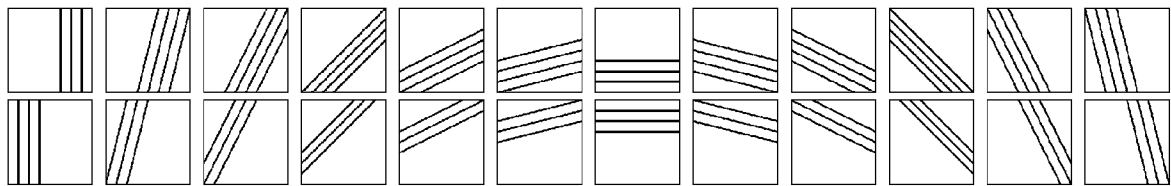


Fig. 10

	L0 MV	L1 MV
Merge Index		
0	x	
1		x
2	x	
3		x
4	x	

Fig. 11

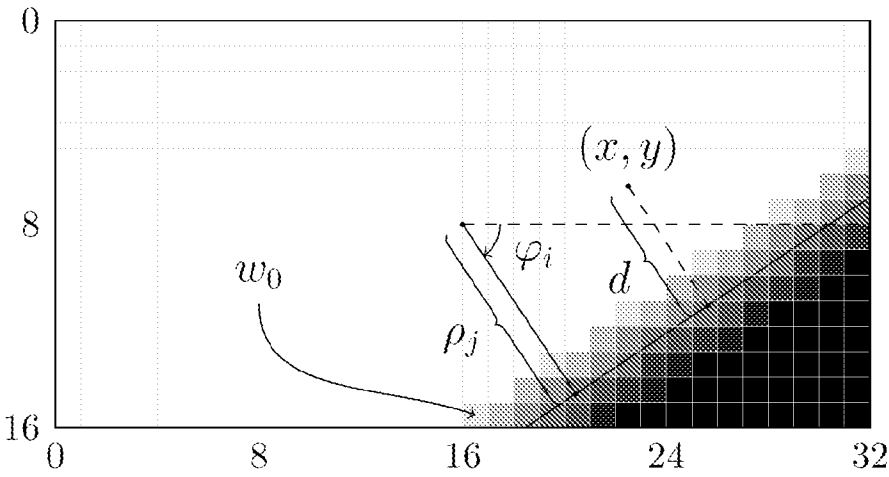


Fig. 12

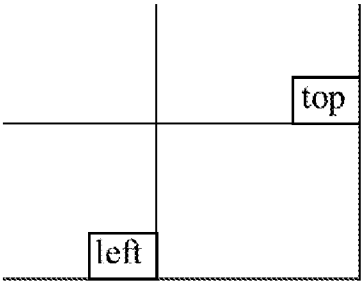


Fig. 13

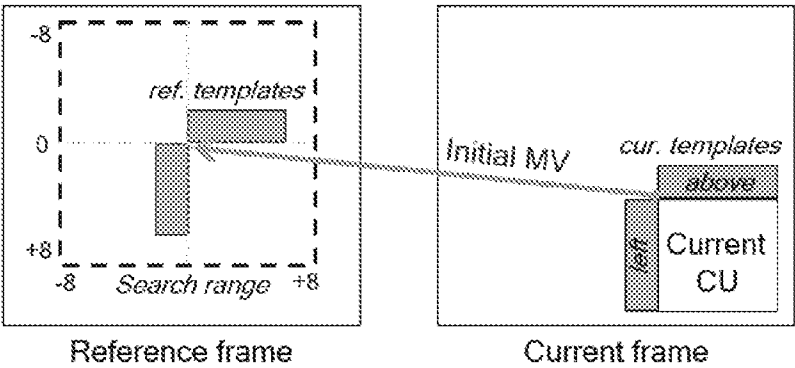


Fig. 14

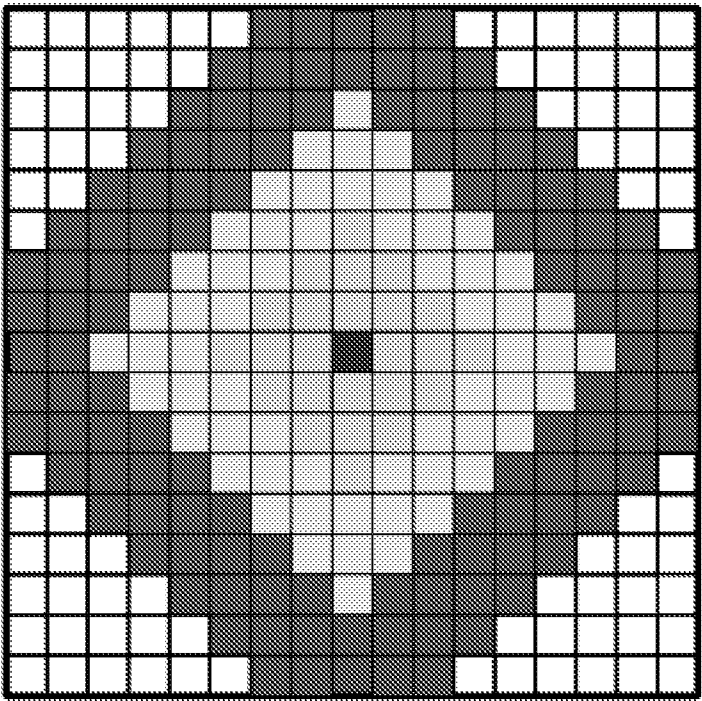


Fig. 15

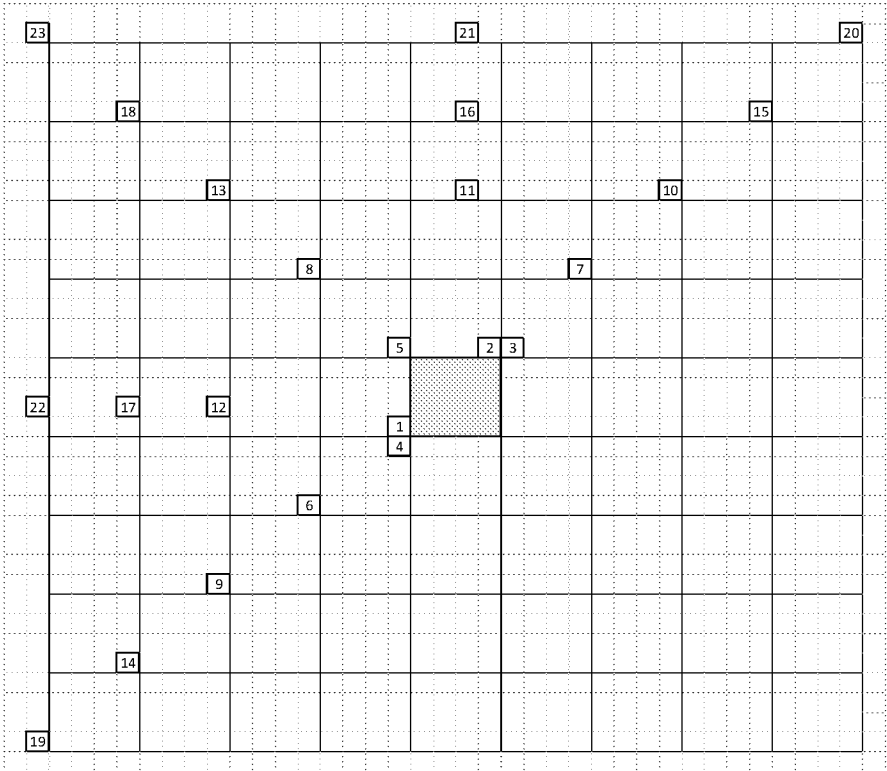
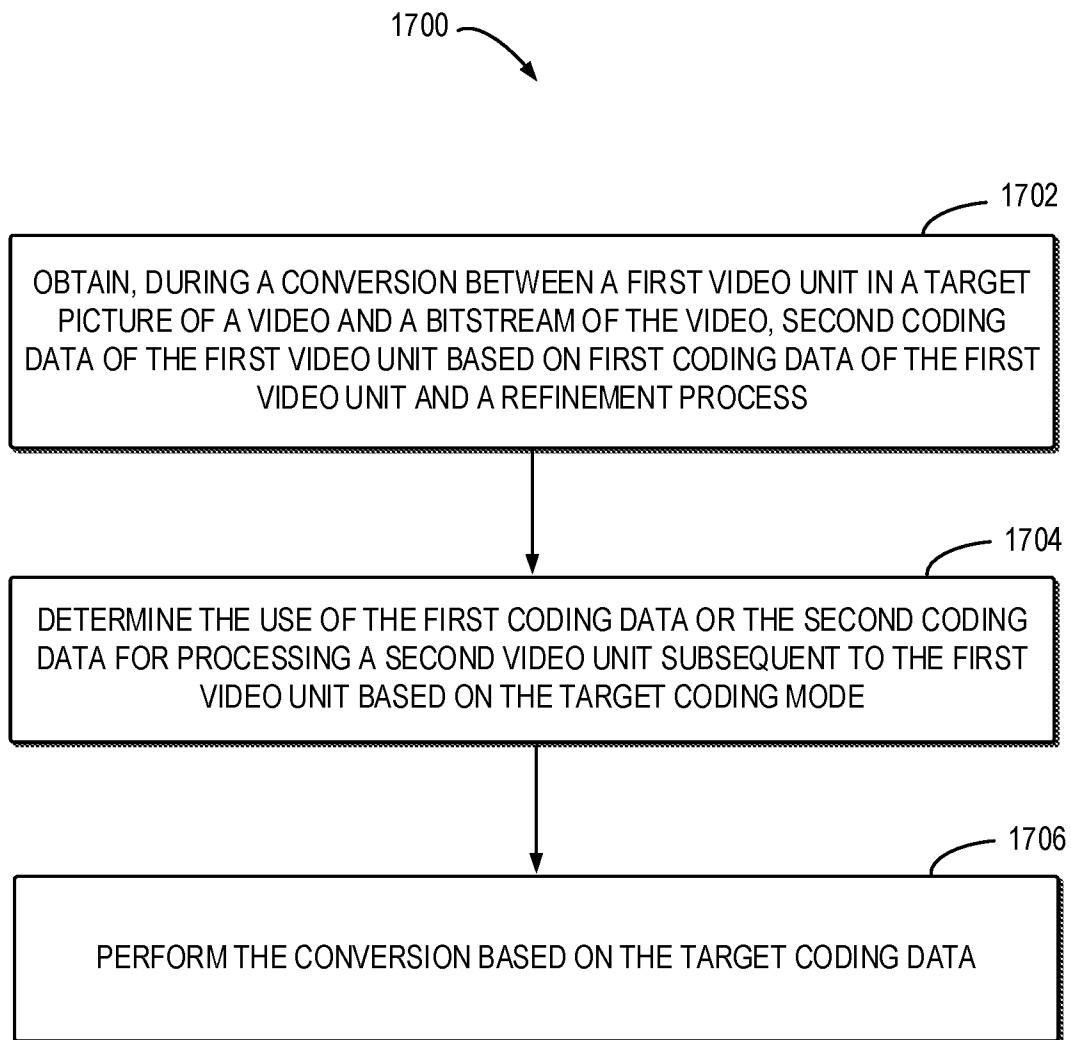
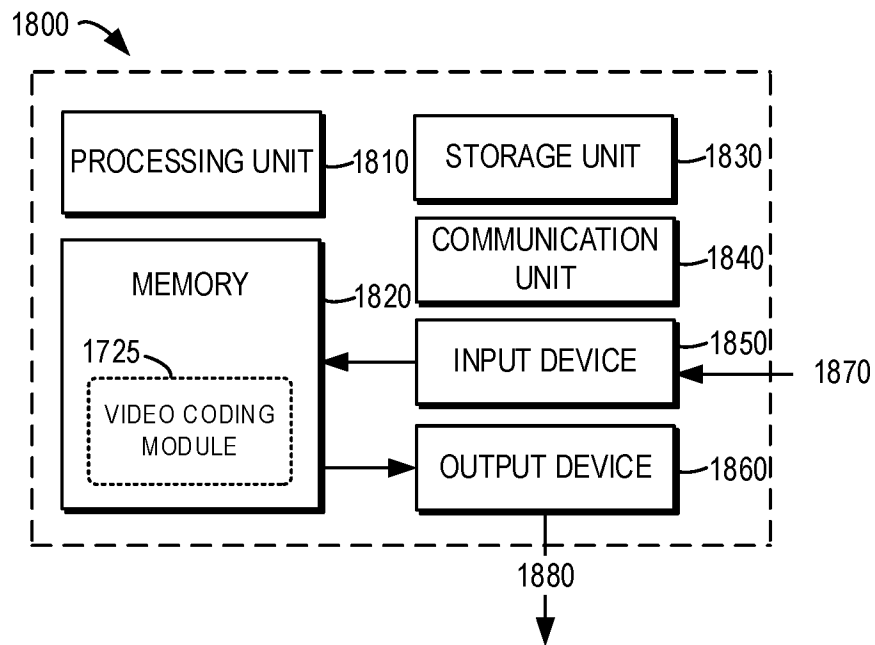


Fig. 16

**Fig. 17**

**Fig. 18**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/088127

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/46(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

CNPAT;CNKI;WPI;EPODOC:video, process, conversion, bitstream, coding, refinement, coding mode, first, second, motion

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 2020228718 A1 (BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. et al.) 19 November 2020 (2020-11-19) description, paragraphs [0004]-[0034]	1-39
A	US 2008212677 A1 (CHEN, Peisong et al.) 04 September 2008 (2008-09-04) the whole document	1-39
A	US 2013182774 A1 (QUALCOMM INCORPORATED) 18 July 2013 (2013-07-18) the whole document	1-39
A	US 2021084322 A1 (ALIBABA GROUP HOLDING LIMITED) 18 March 2021 (2021-03-18) the whole document	1-39
A	WO 2014107183 A1 (INTEL CORPORATION et al.) 10 July 2014 (2014-07-10) the whole document	1-39



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

“&” document member of the same patent family

Date of the actual completion of the international search

05 July 2022

Date of mailing of the international search report

21 July 2022

Name and mailing address of the ISA/CN

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

Facsimile No. (86-10)62019451

Authorized officer

JIAO, Yue

Telephone No. 86-(10)-53961306

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2022/088127

Patent document cited in search report				Publication date (day/month/year)				Patent family member(s)				Publication date (day/month/year)			
WO	2020228718	A1	19 November 2020	WO	2020228716	A1	19 November 2020								
				CN	113826398	A	21 December 2021								
				CN	113826405	A	21 December 2021								
US	2008212677	A1	04 September 2008	JP	2010520714	A	10 June 2010								
				EP	2132938	A1	16 December 2009								
				WO	2008109471	A1	12 September 2008								
				KR	20090123914	A	02 December 2009								
				CN	101622880	A	06 January 2010								
				TW	200904197	A	16 January 2009								
US	2013182774	A1	18 July 2013	AU	2012366142	A1	07 August 2014								
				CN	104054348	A	17 September 2014								
				PH	12014501531	A1	08 October 2014								
				EP	2805489	A1	26 November 2014								
				CA	2860762	A1	25 July 2013								
				BR	112014017379	A2	13 June 2017								
				RU	2014133720	A	10 March 2016								
				UA	111879	C2	24 June 2016								
				MY	165064	A	28 February 2018								
				WO	2013109382	A2	25 July 2013								
				JP	2015507907	A	12 March 2015								
				KR	20140120339	A	13 October 2014								
				HK	1198236	A1	13 March 2015								
				PH	12014501528	B1	08 October 2014								
				TW	201338560	A	16 September 2013								
				TW	201336312	A	01 September 2013								
				AU	2012366143	A1	07 August 2014								
				ZA	201406023	B	27 September 2017								
				HK	1198403	A1	17 April 2015								
				EP	2805488	A2	26 November 2014								
				SG	11201403349S	A	26 September 2014								
				CN	104054347	A	17 September 2014								
				SG	11201403351V	A	28 August 2014								
				IL	233360	D0	31 August 2014								
				UA	112675	C2	10 October 2016								
				WO	2013109383	A1	25 July 2013								
				MY	168215	A	15 October 2018								
				BR	112014017521	A2	13 June 2017								
				IL	233321	D0	31 August 2014								
				CA	2860840	A1	25 July 2013								
				JP	2015507906	A	12 March 2015								
				KR	20140120338	A	13 October 2014								
				US	2013182775	A1	18 July 2013								
				SG	10201605295X	A	30 August 2016								
US	2021084322	A1	18 March 2021	WO	2021050234	A1	18 March 2021								
				KR	20220062588	A	17 May 2022								
WO	2014107183	A1	10 July 2014	EP	2941866	A1	11 November 2015								
				KR	20150082527	A	15 July 2015								
				CN	104854865	A	19 August 2015								
				SG	11201503954Y	A	29 June 2015								

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/CN2022/088127

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
		JP 2016508318 A	17 March 2016