



(51) International Patent Classification:

H04N 19/52 (2014.01) H04N 19/105 (2014.01)
H04N 19/593 (2014.01)

(21) International Application Number:

PCT/CN2024/072864

(22) International Filing Date:

17 January 2024 (17.01.2024)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2023/073261

19 January 2023 (19.01.2023) CN

(71) Applicants: **DOUYIN VISION 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: **WANG, Yang**; Jinritoutiao Post office, China
Satellite Communications Tower, No. 63, Zhichun Road,
Haidian District, Beijing 100080 (CN). **ZHANG, Kai**;
12655 West Jefferson Boulevard, Sixth Floor, Suite No.
137, Los Angeles, California 90066 (US). **ZHANG, Li**;
12655 West Jefferson Boulevard, Sixth Floor, Suite No.
137, Los Angeles, California 90066 (US).

(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, CV, CZ, DE, DJ, DK, DM,
DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG,

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

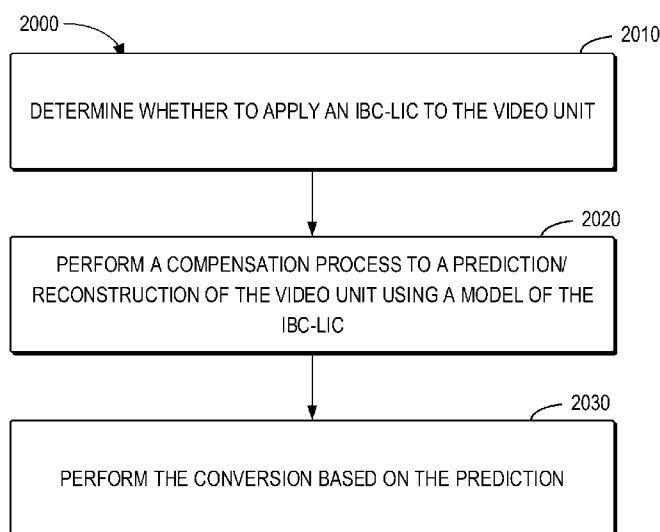


Fig. 20

(57) Abstract: Embodiments of the present disclosure provide a solution for video processing. A method for video processing is proposed. The method comprises: determining, for a conversion between a video unit of a video and a bitstream of the video unit, whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to the video unit based on a model used for the IBC-LIC; in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; and performing the conversion based on the compensated prediction/reconstruction of the video unit.



KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY,
MA, MD, MG, MK, MN, MU, 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, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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, APPARATUS, AND MEDIUM FOR VIDEO PROCESSING

FIELDS

[0001] Embodiments of the present disclosure relates generally to video processing
5 techniques, and more particularly, to intra block copy with local illumination
compensation.

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,
10 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 video coding techniques is generally expected to be further improved.

SUMMARY

15 [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: determining, for a conversion between a video unit of a video and a bitstream
of the video unit, whether an intra block copy (IBC) and local illumination compensation
(LIC) (IBC-LIC) is applied to the video unit; in accordance with a determination that the
20 IBC-LIC is applied to the video unit, performing a compensation process to a
prediction/reconstruction of the video unit using a model of the IBC-LIC; and performing
the conversion based on the compensated prediction/reconstruction of the video unit. In
this way, it can improve coding efficiency.

[0005] In a second aspect, an apparatus for video processing is proposed. The apparatus
25 comprises a processor and a non-transitory memory with instructions thereon. The
instructions upon execution by the processor, cause the processor to perform a method in
accordance with the first aspect of the present disclosure.

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

disclosure.

[0007] In a fourth aspect, another non-transitory computer-readable recording medium is proposed. The non-transitory computer-readable recording medium stores a bitstream of a video which is generated by a method performed by an apparatus for video processing.

5 The method comprises: determining whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to a video unit of the video; in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; and generating the bitstream based on the compensated
10 prediction/reconstruction of the video unit.

[0008] In a fifth aspect, a method for storing a bitstream of a video is proposed. The method comprises: determining whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to a video unit of the video; in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation
15 process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; generating the bitstream based on the compensated prediction/reconstruction of the video unit; 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
20 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
25 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 that illustrates an example video coding system, in accordance with some embodiments of the present disclosure;

30 [0012] Fig. 2 illustrates a block diagram that illustrates a first example video encoder, in accordance with some embodiments of the present disclosure;

- [0013] Fig. 3 illustrates a block diagram that illustrates an example video decoder, in accordance with some embodiments of the present disclosure;
- [0014] Fig. 4 illustrates an example of encoder block diagram of VVC;
- [0015] Fig. 5 illustrates several intra prediction modes;
- 5 [0016] Fig. 6A and Fig. 6B illustrate reference samples for wide-angular intra prediction;
- [0017] Fig. 7 illustrates problem of discontinuity in case of directions beyond 45° ;
- [0018] Fig. 8 illustrates example of motion vector scaling for temporal merge candidate;
- [0019] Fig. 9A and Fig. 9B illustrate MMVD search point;
- 10 [0020] Fig. 10 illustrates an example of local illustration compensation;
- [0021] Fig. 11 illustrates no subsampling for the short side;
- [0022] Fig. 12 illustrates IBC reference region depending on current CU position;
- [0023] Fig. 13 illustrates examples of symmetry in screen content pictures;
- [0024] Fig. 14A illustrates an illustration of BV adjustment for horizontal flip;
- 15 [0025] Fig. 14B illustrates an illustration of BV adjustment for vertical flip;
- [0026] Fig. 15 illustrates an intra template matching search area used;
- [0027] Fig. 16 illustrates templates used to derive the parameters of LIC for IBC;
- [0028] Fig. 17 illustrates samples used to drive IBC-LIC parameters for the templates;
- [0029] Fig. 18 illustrates adjusted reference template when RR-IBC is horizontal flip;
- 20 [0030] Fig. 19 illustrates adjusted reference template when RR-IBC is vertical flip;
- [0031] Fig. 20 illustrates a flowchart of a method for video processing in accordance with embodiments of the present disclosure; and
- [0032] Fig. 21 illustrates a block diagram of a computing device in which various embodiments of the present disclosure can be implemented.
- 25 [0033] Throughout the drawings, the same or similar reference numerals usually refer to the same or similar elements.

DETAILED DESCRIPTION

[0034] 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.

[0035] 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.

[0036] 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.

[0037] 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.

[0038] 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

[0039] 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.

[0040] 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.

[0041] 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 destination device 120 via the I/O interface 116 through the network 130A. The encoded video data may also be stored onto a storage medium/server 130B for access by destination device 120.

[0042] 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.

[0043] 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 further standards.

[0044] 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.

[0045] 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.

[0046] In some embodiments, the video encoder 200 may include a partition unit 201, a predication unit 202 which may include a mode select 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.

[0047] 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.

[0048] 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.

[0049] The partition unit 201 may partition a picture into one or more video blocks. The video encoder 200 and the video decoder 300 may support various video block sizes.

[0050] The mode select 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 a residual generation unit 207 to generate residual block data and to a reconstruction unit 212 to reconstruct the encoded block for use as a reference picture. In some examples, the mode select 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 select 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.

[0051] 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.

[0052] 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.

[0053] 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.

[0054] 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.

[0055] 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.

[0056] 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.

[0057] 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.

[0058] As discussed above, video encoder 200 may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder 200 include advanced motion vector prediction (AMVP) and merge mode signaling.

[0059] The intra prediction unit 206 may perform intra prediction on the current video block. When the intra prediction unit 206 performs 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.

[0060] 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.

[0061] 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.

[0062] The transform processing 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.

[0063] After the transform processing 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.

[0064] 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 prediction unit 202 to produce a reconstructed video block associated with the current video block for storage in the buffer 213.

[0065] After the reconstruction unit 212 reconstructs the video block, loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[0066] The entropy encoding unit 214 may receive data from other functional

components of the video encoder 200. When the entropy encoding unit 214 receives the data, 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.

5 [0067] 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.

[0068] 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
10 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.

[0069] 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
15 quantization unit 304, an inverse transformation unit 305, and a reconstruction unit 306 and a buffer 307. The video decoder 300 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 200.

[0070] 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
20 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,
25 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
30 to deriving the motion information from spatially or temporally neighboring blocks.

[0071] 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.

[0072] 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.

[0073] 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.

[0074] The intra prediction unit 303 may use intra prediction modes for example 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 entropy decoding unit 301. The inverse transform unit 305 applies an inverse transform.

[0075] 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 predication and also produces decoded video for presentation on a display device.

[0076] Some exemplary 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. Brief Summary

The present disclosure is related to video coding technologies. Specifically, it is related to intra block copy (IBC), how to and/or whether to combine IBC with local illumination compensation, and other coding tools in image/video coding. It may be applied to the existing video coding standard like HEVC, or Versatile Video Coding (VVC). It may be also applicable to future video coding standards or video codec.

2. Introduction

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

2.1 Coding flow of a typical video codec

Fig. 4 shows an example of encoder block diagram of VVC, which contains three in-loop filtering blocks: deblocking filter (DF), sample adaptive offset (SAO) and ALF. Unlike DF, which uses predefined filters, SAO and ALF utilize the original samples of the current picture to reduce the mean square errors between the original samples and the reconstructed samples by adding an offset and by applying a finite impulse response (FIR) filter, respectively, with

coded side information signalling the offsets and filter coefficients. ALF is located at the last processing stage of each picture and can be regarded as a tool trying to catch and fix artifacts created by the previous stages.

2.2 Intra mode coding with 67 intra prediction modes

5 Fig. 5 illustrates 67 intra prediction modes. To capture the arbitrary edge directions presented in natural video, the number of directional intra modes is extended from 33, as used in HEVC, to 65, as shown in Fig. 5, and the planar and DC modes remain the same. These denser directional intra prediction modes apply for all block sizes and for both luma and chroma intra predictions.

10 In the HEVC, every intra-coded block has a square shape and the length of each of its side is a power of 2. Thus, no division operations are required to generate an intra-predictor using DC mode. In VVC, blocks can have a rectangular shape that necessitates the use of a division operation per block in the general case. To avoid division operations for DC prediction, only the longer side is used to compute the average for non-square blocks.

15 2.2.1 Wide angle intra prediction

Although 67 modes are defined in the VVC, the exact prediction direction for a given intra prediction mode index is further dependent on the block shape. Conventional angular intra prediction directions are defined from 45 degrees to -135 degrees in clockwise direction. In VVC, several conventional angular intra prediction modes are adaptively replaced with wide-angle intra prediction modes for non-square blocks. The replaced modes are signalled using the original mode indexes, which are remapped to the indexes of wide angular modes after parsing. The total number of intra prediction modes is unchanged, i.e., 67, and the intra mode coding method is unchanged.

20 Fig. 6A and Fig. 6B illustrate reference samples for wide-angular intra prediction To support these prediction directions, the top reference with length $2W+1$, and the left reference with length $2H+1$, are defined as shown in Fig. 6A and Fig. 6B. Fig. 6A and Fig. 6B illustrate reference samples for wide-angular intra prediction.

The number of replaced modes in wide-angular direction mode depends on the aspect ratio of a block. The replaced intra prediction modes are illustrated in Table 1.

Table 1 – Intra prediction modes replaced by wide-angular modes

Aspect ratio	Replaced intra prediction modes
$W / H == 16$	Modes 2,3,4,5,6,7,8,9,10,11,12, 13,14,15
$W / H == 8$	Modes 2,3,4,5,6,7,8,9,10,11,12, 13
$W / H == 4$	Modes 2,3,4,5,6,7,8,9,10,11
$W / H == 2$	Modes 2,3,4,5,6,7,8,9
$W / H == 1$	None
$W / H == 1/2$	Modes 59,60,61,62,63,64,65,66
$W / H == 1/4$	Mode 57,58,59,60,61,62,63,64,65,66
$W / H == 1/8$	Modes 55, 56,57,58,59,60,61,62,63,64,65,66
$W / H == 1/16$	Modes 53, 54, 55, 56,57,58,59,60,61,62,63,64,65,66

As shown in Fig. 7, two vertically adjacent predicted samples may use two non-adjacent reference samples in the case of wide-angle intra prediction. Hence, low-pass reference samples filter and side smoothing are applied to the wide-angle prediction to reduce the negative effect of the increased gap Δp_α . If a wide-angle mode represents a non-fractional offset. There are 8 modes in the wide-angle modes satisfy this condition, which are $[-14, -12, -10, -6, 72, 76, 78, 80]$. When a block is predicted by these modes, the samples in the reference buffer are directly copied without applying any interpolation. With this modification, the number of samples needed to be smoothing is reduced. Besides, it aligns the design of non-fractional modes in the conventional prediction modes and wide-angle modes.

In VVC, 4:2:2 and 4:4:4 chroma formats are supported as well as 4:2:0. Chroma derived mode (DM) derivation table for 4:2:2 chroma format was initially ported from HEVC extending the number of entries from 35 to 67 to align with the extension of intra prediction modes. Since HEVC specification does not support prediction angle below -135 degree and above 45 degree, luma intra prediction modes ranging from 2 to 5 are mapped to 2. Therefore, chroma DM derivation table for 4:2:2: chroma format is updated by replacing some values of the entries of the mapping table to convert prediction angle more precisely for chroma blocks.

2.3 Inter prediction

For each inter-predicted CU, motion parameters consisting of motion vectors, reference picture indices and reference picture list usage index, and additional information needed for the new coding feature of VVC to be used for inter-predicted sample generation. The motion parameter can be signalled in an explicit or implicit manner. When a CU is coded with skip mode, the CU is associated with one PU and has no significant residual coefficients, no coded motion vector delta or reference picture index. A merge mode is specified whereby the motion parameters for

the current CU are obtained from neighbouring CUs, including spatial and temporal candidates, and additional schedules introduced in VVC. The merge mode can be applied to any inter-predicted CU, not only for skip mode. The alternative to merge mode is the explicit transmission of motion parameters, where motion vector, corresponding reference picture index for each reference picture list and reference picture list usage flag and other needed information are signalled explicitly per each CU.

2.4 Intra block copy (IBC)

Intra block copy (IBC) is a tool adopted in HEVC extensions on SCC. It is well known that it significantly improves the coding efficiency of screen content materials. Since IBC mode is implemented as a block level coding mode, block matching (BM) is performed at the encoder to find the optimal block vector (or motion vector) for each CU. Here, a block vector is used to indicate the displacement from the current block to a reference block, which is already reconstructed inside the current picture. The luma block vector of an IBC-coded CU is in integer precision. The chroma block vector rounds to integer precision as well. When combined with AMVR, the IBC mode can switch between 1-pel and 4-pel motion vector precisions. An IBC-coded CU is treated as the third prediction mode other than intra or inter prediction modes. The IBC mode is applicable to the CUs with both width and height smaller than or equal to 64 luma samples.

At the encoder side, hash-based motion estimation is performed for IBC. The encoder performs RD check for blocks with either width or height no larger than 16 luma samples. For non-merge mode, the block vector search is performed using hash-based search first. If hash search does not return valid candidate, block matching based local search will be performed.

In the hash-based search, hash key matching (32-bit CRC) between the current block and a reference block is extended to all allowed block sizes. The hash key calculation for every position in the current picture is based on 4×4 sub-blocks. For the current block of a larger size, a hash key is determined to match that of the reference block when all the hash keys of all 4×4 sub-blocks match the hash keys in the corresponding reference locations. If hash keys of multiple reference blocks are found to match that of the current block, the block vector costs of each matched reference are calculated and the one with the minimum cost is selected.

In block matching search, the search range is set to cover both the previous and current CTUs. At CU level, IBC mode is signalled with a flag and it can be signalled as IBC AMVP mode or IBC skip/merge mode as follows:

- IBC skip/merge mode: a merge candidate index is used to indicate which of the block vectors in the list from neighbouring candidate IBC coded blocks is used to predict the current block. The merge list consists of spatial, HMVP, and pairwise candidates.
- IBC AMVP mode: block vector difference is coded in the same way as a motion vector difference. The block vector prediction method uses two candidates as predictors, one from left neighbour and one from above neighbour (if IBC coded). When either neighbour is not

available, a default block vector will be used as a predictor. A flag is signalled to indicate the block vector predictor index.

2.5 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 regular 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 MMVD candidate flag is signalled to specify which one is used between the first and second merge candidates. Fig. 8 shows an illustration of motion vector scaling for temporal merge candidate. Fig. 9A and Fig. 9B illustrate examples of MMVD search point. Distance index specifies motion magnitude information and indicate the pre-defined offset from the starting point. As shown in Fig. 9A and Fig. 9B, 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 2.

Table 2 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 3. 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 3 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), and the difference of POC in list 0 is greater than the one in list 1, the sign in Table 3 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. Otherwise, if the difference of POC in list 1 is greater than list 0, the sign in Table 3 specifies the sign of MV offset added to the list1 MV component of starting MV and the sign for the list0 MV has opposite value.

The MVD is scaled according to the difference of POCs in each direction. If the differences of

POCs in both lists are the same, no scaling is needed. Otherwise, if the difference of POC in list 0 is larger than the one of list 1, the MVD for list 1 is scaled, by defining the POC difference of L0 as td and POC difference of L1 as tb , described in Fig. 8. If the POC difference of L1 is greater than L0, the MVD for list 0 is scaled in the same way. If the starting MV is uni-predicted, the MVD is added to the available MV.

Table 3 – 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	+	–

2.6 Local illumination compensation (LIC)

Local illumination compensation (LIC) is a coding tool to address the issue of local illumination changes between current picture and its temporal reference pictures. The LIC is based on a linear model where a scaling factor and an offset are applied to the reference samples to obtain the prediction samples of a current block. Specifically, the LIC can be mathematically modeled by the following equation:

$$P(x, y) = \alpha \cdot P_r(x + v_x, y + v_y) + \beta$$

where $P(x, y)$ is the prediction signal of the current block at the coordinate (x, y) ; $P_r(x + v_x, y + v_y)$ is the reference block pointed by the motion vector (v_x, v_y) ; α and β are the corresponding scaling factor and offset that are applied to the reference block. Fig. 10 illustrates the LIC process. Fig. 10 illustrates an example of local illustration compensation. In Fig. 10, when the LIC is applied for a block, a least mean square error (LMSE) method is employed to derive the values of the LIC parameters (i.e., α and β) by minimizing the difference between the neighboring samples of the current block (i.e., the template T in Fig. 10) and their corresponding reference samples in the temporal reference pictures (i.e., either $T0$ or $T1$ in Fig. 10). Additionally, to reduce the computational complexity, both the template samples and the reference template samples are subsampled (adaptive subsampling) to derive the LIC parameters, i.e., only the shaded samples in Fig. 10 are used to derive α and β .

Fig. 11 illustrates no subsampling for the short side. To improve the coding performance, no subsampling for the short side is performed as shown in Fig. 11.

2.7 IBC with Template Matching

It is proposed to also use Template Matching with IBC for both IBC merge mode and IBC AMVP mode.

The IBC-TM merge list has been modified compared to the one used by regular IBC merge

mode such that the candidates are selected according to a pruning method with a motion distance between the candidates as in the regular TM merge mode. The ending zero motion fulfillment (which is a nonsense regarding Intra coding) has been replaced by motion vectors to the left $(-W, 0)$, top $(0, -H)$ and top-left $(-W, -H)$ CUs, then, if necessary, the list is fulfilled with the left one without pruning.

In the IBC-TM merge mode, the selected candidates are refined with the Template Matching method prior to the RDO or decoding process. The IBC-TM merge mode has been put in competition with the regular IBC merge mode and a TM-merge flag is signaled.

In the IBC-TM AMVP mode, up to 3 candidates are selected from the IBC merge list. Each of those 3 selected candidates are refined using the Template Matching method and sorted according to their resulting Template Matching cost. Only the 2 first ones are then considered in the motion estimation process as usual.

Fig. 12 illustrates IBC reference region depending on current CU position. The Template Matching refinement for both IBC-TM merge and AMVP modes is quite simple since IBC motion vectors are constrained to be integer and within a reference region as shown in Fig. 12. So, in IBC-TM merge mode, all refinements are performed at integer precision, and in IBC-TM AMVP mode, they are performed either at integer or 4-pel precision. In both cases, the refined motion vectors in each refinement step must respect the constraint of the reference region.

2.8 IBC Merge Mode with Block Vector Differences

IBC merge mode with block vector differences is shown as follows.

The distance set is $\{1\text{-pel}, 2\text{-pel}, 4\text{-pel}, 8\text{-pel}, 12\text{-pel}, 16\text{-pel}, 24\text{-pel}, 32\text{-pel}, 40\text{-pel}, 48\text{-pel}, 56\text{-pel}, 64\text{-pel}, 72\text{-pel}, 80\text{-pel}, 88\text{-pel}, 96\text{-pel}, 104\text{-pel}, 112\text{-pel}, 120\text{-pel}, 128\text{-pel}\}$, and the BVD directions are two horizontal and two vertical directions.

The base candidates are selected from the first five candidates in the reordered IBC merge list. And based on the SAD cost between the template (one row above and one column left to the current block) and its reference for each refinement position, all the possible MBVD refinement positions (20×4) for each base candidate are reordered. Finally, the top 8 refinement positions with the lowest template SAD costs are kept as available positions, consequently for MBVD index coding.

2.9 Reconstruction-Reordered IBC (RR-IBC)

Screen content coding tools like Intra Block Copy (IBC) generate a prediction block by directly copying a prior coded reference region in the same picture. Fig. 13 illustrates examples of symmetry in screen content pictures. Symmetry is often observed in video content, especially in text character regions and computer-generated graphics in screen content sequences, as shown in Fig. 13. Therefore, a specific screen content coding tool considering the symmetry would be efficient to compress such kinds of video contents.

A Reconstruction-Reordered IBC (RR-IBC) mode is proposed for screen content video coding.

When it is applied, the samples in a reconstruction block are flipped according to a flip type of the current block. At the encoder side, the original block is flipped before motion search and residual calculation, while the prediction block is derived without flipping. At the decoder side, the reconstruction block is flipped back to restore the original block.

- 5 Two flip methods, horizontal flip and vertical flip, are supported for RR-IBC coded blocks. A syntax flag is firstly signalled for an IBC AMVP coded block, indicating whether the reconstruction is flipped, and if it is flipped, another flag is further signaled specifying the flip type. For IBC merge, the flip type is inherited from neighbouring blocks, without syntax signalling. Considering the horizontal or vertical symmetry, the current block and the reference
10 block are normally aligned horizontally or vertically. Therefore, when a horizontal flip is applied, the vertical component of the BV is not signaled and inferred to be equal to 0. Similarly, the horizontal component of the BV is not signaled and inferred to be equal to 0 when a vertical flip is applied.

- Fig. 14A illustrates an illustration of BV adjustment for horizontal flip. Fig. 14B illustrates an
15 illustration of BV adjustment for vertical flip. To better utilize the symmetry property, a flip-aware BV adjustment approach is applied to refine the block vector candidate. For example, as shown in Fig. 14A and Fig. 14B, (x_{nbr}, y_{nbr}) and (x_{cur}, y_{cur}) represent the coordinates of the center sample of the neighboring block and the current block, respectively, BV^{nbr} and BV^{cur} denotes the BV of the neighboring block and the current block, respectively. Instead of directly
20 inheriting the BV from a neighbouring block, the horizontal component of BV^{cur} is calculated by adding a motion shift to the horizontal component of BV^{nbr} (denoted as BV^{nbr}_h) in case that the neighbouring block is coded with a horizontal flip, i.e., $BV^{cur}_h = 2(x_{nbr} - x_{cur}) + BV^{nbr}_h$. Similarly, the vertical component of BV^{cur} is calculated by adding a motion shift to the vertical component of BV^{nbr} (denoted as BV^{nbr}_v) in case that the neighbouring block is coded with a
25 vertical flip, i.e., $BV^{cur}_v = 2(y_{nbr} - y_{cur}) + BV^{nbr}_v$.

2. 10 Intra template matching

- Intra template matching prediction (Intra TMP) is a special intra prediction mode that copies the best prediction block from the reconstructed part of the current frame, whose L-shaped template matches the current template. For a predefined search range, the encoder searches for
30 the most similar template to the current template in a reconstructed part of the current frame and uses the corresponding block as a prediction block. The encoder then signals the usage of this mode, and the same prediction operation is performed at the decoder side.

- Fig. 15 illustrates an intra template matching search area used. The prediction signal is generated by matching the L-shaped causal neighbor of the current block with another block in
35 a predefined search area in Fig. 15 consisting of:

- R1: current CTU,
- R2: top-left CTU,
- R3: above CTU,

R4: left CTU.

SAD is used as a cost function.

Within each region, the decoder searches for the template that has least SAD with respect to the current one and uses its corresponding block as a prediction block.

- 5 The dimensions of all regions (SearchRange_w, SearchRange_h) are set proportional to the block dimension (BlkW, BlkH) to have a fixed number of SAD comparisons per pixel. That is:

$$\text{SearchRange_w} = a * \text{BlkW}$$

$$\text{SearchRange_h} = a * \text{BlkH}$$

where 'a' is a constant that controls the gain/complexity trade-off. In practice, 'a' is equal to 5.

- 10 The Intra template matching tool is enabled for CUs with size less than or equal to 64 in width and height. This maximum CU size for Intra template matching is configurable.

The Intra template matching prediction mode is signaled at CU level through a dedicated flag when DIMD is not used for current CU.

3. Problems

- 15 In current design of IBC, the whole block is directly copied from the reconstructed region in the current picture. However, when illumination change occurs within the current picture, the coding efficiency of IBC may be limited.

4. Detailed Solutions

The detailed embodiments below should be considered as examples to explain general concepts.

- 20 These embodiments should not be interpreted in a narrow way. Furthermore, these embodiments can be combined in any manner.

In the present disclosure, intra block copy (IBC) may not be limited to the current IBC technology, but may be interpreted as the technology that reference (or prediction) block is obtained with samples in the current slice/tile/subpicture/picture/other video unit (e.g., CTU

- 25 row) excluding the conventional intra prediction methods.

In the present disclosure, local illumination compensation (LIC) may not be limited to the current LIC technology. LIC may refer to an inter prediction technique to model local illumination variation between current block and its prediction block as a function of that between current block template and reference block template. The parameters of the function

- 30 may be denoted by a linear equation (e.g., $\alpha \times p[x] + \beta$) or a non-linear equation.

In the present disclosure, CIBCIP (or IBC-CIIP) may refer to a coding tool which combines of intra block copy (IBC) and intra prediction. It's a coding tool which obtain the prediction of a block using both IBC and intra prediction.

In the present disclosure, IBC-GPM may refer to a coding tool which obtains the prediction of

at least one sub-partitions using IBC in a video unit when the video unit is divided into more than one sub-partitions geometrically.

In the following discussion, IBC may be replaced by other coding tools that rely on coded/decoded/reconstructed information within the same region, e.g., palette, intra template matching.

IBC with LIC

1. It is proposed that a refined prediction sample may be derived as $f(p[x])$, wherein $p[x]$ denotes a prediction sample of the video unit, and f is any function.

a. In one example, $f(p[x]) = \alpha \times p[x] + \beta$, wherein α and β denotes the parameters of the linear equation.

b. In one example, the function or at least one parameter of the function may be derived based on a template of the current block.

c. In one example, the prediction sample may be derived by IBC.

i. In one example, the function or at least one parameter of the function may be derived based on a template of the reference block of the current block, wherein the reference block may be located by a block vector (BV).

d. In one example, one or more offsets may be used to modify one or more derived parameters.

i. In one example, the one or more offsets may be signalled, or derived, or pre-defined.

ii. In one example, whether to and/or how to use the offsets to modify one or more parameters may be signalled, or pre-defined, or determined using coding information.

2. It is proposed that LIC may be applied to compensate the prediction (reconstruction) of a video unit, wherein IBC is used to obtain the prediction (reconstruction) of the video unit. It is denoted as IBC-LIC.

a. In one example, a linear or non-linear equation/model may be used for IBC-LIC to compensate the prediction of the video unit.

i. In one example, the linear equation may be $\alpha \times p[x] + \beta$, wherein $p[x]$ denotes the prediction of the video unit, and α and β denotes the parameters of the linear equation.

ii. In one example, the equation may be $c_0 \times x_0 + c_1 \times x_1 + c_2 \times x_2 + \dots + c_N \times x_N + B$, wherein c_i are the parameters and x_i denote the coding information.

1) In one example, the coding information may refer to reconstruction/reference/prediction samples, or the values calculated using the reconstruction/reference/prediction samples (e.g., gradients), or position information of the reconstruction/reference/prediction samples.

b. In one example, the parameters of the equation used in IBC-LIC may be pre-defined, or signalled in the bitstream.

c. In one example, the parameters of the equation used in IBC-LIC may be derived using coding information.

- i. In one example, a current template consists of the neighbouring reconstructed (adjacent or non-adjacent) samples of the video unit and a reference template may be used to derive the parameters. An example is shown in Fig. 16.

1) In one example, the reference template may be derived using the BV that is used to obtain the prediction of the video unit.

a) In one example, when the BV is fractional BV, it may be rounded to an integer BV.

i. In one example, clipping operation may be used when rounding the fractional BV.

ii. Alternatively, the fractional BV is still used to derive the reference template.

1. In one example, the same interpolation method is used to derive the reference template as the interpolation method used to derive the prediction of the video unit.

2. Alternatively, a different interpolation method is used to derive the reference template as the interpolation method to used derive the prediction of the video unit, such as different interpolation taps.

a. In one example, 2-tap interpolation filter may be used.

2) In one example, partial or all samples of the current template and the reference template may be used to derive the parameters.

3) In one example, a least square error method may be used to derive the parameters.

4) In one example, the method solving a linear/non-linear equation (e.g., MSE minimization) used for other coding tools may be used to derive the parameters.

a) In one example, the method may refer to Cholesky decomposition used in ALF and its variants.

b) In one example, the method may refer to LDL decomposition used in CCCM and its variants.

c) In one example, the method may refer to Gaussian elimination used in CCCM and its variants.

5) In one example, MSE minimization may be used as the metric to solve the linear/non-linear equation.

a) In one example, other metrics may be used to solve the linear/non-linear equation, such as SSE or mean of SSE.

b) In one example, a subjective metric may be used to solve the linear/non-linear equation, such as SSIM.

ii. In one example, how to derive the parameters using the current template and the reference template may be same as LIC for inter prediction.

iii. In one example, more than one method may be used to derive the parameters for IBC-LIC.

- 1) In one example, which method is used may be signalled, or pre-defined, or determined using coding information.
- 2) In one example, more than one method may be used to derive more than one set of parameters, and the predictions of the current video unit modified using the more than one set of parameters may be fused.
 - a) In one example, the weights used for fusion may be pre-defined, or signalled, or determined using coding information.
- d. In one example, partial or all prediction samples of the video unit may be compensated using IBC-LIC.
3. In one example, IBC-LIC may be applied to IBC AMVP mode and/or IBC merge mode.
 - a. In one example, IBC AMVP mode may refer to normal IBC AMVP, or TM based IBC AMVP, or RR-IBC AMVP mode, or CIBCIP (IBC-CIIP), or IBC-GPM, or other IBC AMVP mode wherein a BV predictor is derived and BVD is signalled/derived.
 - b. In one example, IBC merge mode may refer to normal IBC merge mode, or IBC-TM merge mode, or IBC-MBVD mode, or CIBCIP (IBC-CIIP), or IBC-GPM.
 - i. In one example, IBC-LIC may be applied to a specific IBC merge candidate type.
 - ii. In another example, IBC-LIC may be not allowed to apply to a specific IBC merge candidate type.
 - 1) In one example, the merge candidate type may refer to RR-IBC candidate.
 - c. Alternatively, IBC-LIC may be not allowed to be applied to one or more of the above IBC coding tools.
 - i. In one example, the IBC coding tool may refer to RR-IBC, or CIBCIP (IBC-CIIP), or IBC-GPM.
 - d. In one example, whether to and/or how to apply IBC-LIC for IBC AMVP mode and/or IBC merge mode may be signalled or determined using coding information.
 - e. In one example, one or more syntax elements may be signalled to indicate whether to and/or how to apply IBC-LIC for IBC AMVP mode and/or IBC merge mode.
 - f. In one example, whether to and/or how to apply IBC-LIC for IBC merge mode may be inherited.
 - i. In one example, the inheritance of whether to and/or how to apply IBC-LIC may be associated with the merge candidate.
 - 1) In one example, IBC-LIC may be disabled when a merge candidate is a specific merge type.
 - a) In one example, the specific type may refer to RR-IBC.
 - ii. In one example, whether to apply IBC-LIC for IBC merge mode may be derived.
 - 1) In one example, template matching based method may be used.
 - a) In one example, a first cost (C_1) may be calculated between the prediction of the template of current video unit and its reconstruction when IBC-LIC is applied; a second cost (C_2) may be calculated between the prediction of the template of current video unit and its reconstruction when IBC-LIC is not applied.

- i. In one example, when $C_1 \leq C_2$, IBC-LIC may be applied; when $C_1 > C_2$, IBC-LIC may be not applied.
 - ii. In one example, when $C_1 \leq S * C_2$, IBC-LIC may be applied; when $C_1 > S * C_2$, IBC-LIC may be not applied, wherein S is a scale factor.
 - iii. In one example, when $C_1 \leq C_2 + O$, IBC-LIC may be applied; when $C_1 > C_2 + O$, IBC-LIC may be not applied, wherein O is an offset.
- 4. In one example, IBC-LIC may be used in the process of reordering a BV candidate list.
 - a. In one example, the BV candidate list may refer to IBC AMVP candidate list, and/or IBC merge candidate list.
 - i. In one example, the BV candidate list may refer to IBC regular merge list, and/or IBC TM merge list, and/or IBC-MBVD merge list.
 - b. In one example, two-pass reordering may be used to reorder the BV candidate list.
 - i. In one example, IBC-LIC may be applied to in the first pass reordering, or the second pass reordering, or both of first pass and second pass reordering.
 - c. In one example, whether to and/or how to apply IBC-LIC in the BV candidate list reordering process may be same as that applying IBC-LIC to the current video unit.
 - d. In one example, when IBC-LIC is used for a BV candidate, the prediction of the template may be refined using the IBC-LIC model.
 - i. In one example, the neighbouring reconstructed samples of the left/above template, and/or the neighbouring reconstructed samples of the reference template of the left/above template may be used to derive the IBC-LIC parameters. An example is shown in Fig. 17.
 - 1) In one example, the reference template of the left/above template may be derived using the BV associated with the BV candidate.
 - 2) In one example, the neighbouring reconstructed samples of the left/above template, and/or the neighbouring reconstructed samples of the reference template of the left/above template, and/or the left/above template may be constrained in the IBC buffer.
 - 3) In one example, when the neighbouring reconstructed samples of the left/above template, and/or the neighbouring reconstructed samples of the reference template of the left/above template, and/or the left/above template are out of the IBC buffer, IBC-LIC may be not used.
 - a) Alternatively, when the neighbouring reconstructed samples of the left/above template, and/or the neighbouring reconstructed samples of the reference template of the left/above template, and/or the left/above template are out of the IBC buffer, the samples out of the IBC buffer may be padded using the samples in the IBC buffer and IBC-LIC may be used.
 - e. In one example, IBC-LIC may be applied to one or more BV candidates in the BV candidate list.

- i. In one example, whether to apply IBC-LIC to a BV candidate may depend on the type of the BV candidate.
 - 1) In one example, the type of the BV candidate may refer to spatial BV candidate, or HMVP BV candidate, or pairwise BV candidate, or default BV candidate, or other types of BV candidate.
 - ii. In one example, IBC-LIC may be applied to partial BV candidates in the list.
 - f. In one example, when a BV candidate is RR-IBC (e.g., RR-IBC flip type is horizontal or vertical), IBC-LIC may be not used.
 - i. Alternatively, IBC-LIC may be used.
 - 1) In one example, the original BV of the BV candidate may be used to derive the IBC-LIC parameters.
 - 2) In one example, the adjusted BV of the BV candidate according to the RR-IBC flip type may be used to derive the IBC-LIC parameters.
 - g. In one example, a set of IBC-LIC parameters may be derived.
 - i. In one example, the derived set of IBC-LIC parameters may be used for left and/or above template.
 - h. In one example, multiple sets of IBC-LIC parameters may be derived.
 - i. In one example, a first set of IBC-LIC parameters may be used for the left template.
 - ii. In one example, a second set of IBC-LIC parameters may be used for the above template.
 - i. In one example, whether to and/or how to apply IBC-LIC to the BV candidate list reordering may depend on a specific coding tool using the BV candidate list.
 - i. In one example, IBC-LIC may be applied to the BV candidate list reordering when the coding tool is IBC AMVP mode, and/or IBC regular merge mode, and/or IBC TM merge mode, and/or IBC-MBVD merge mode.
 - ii. Alternatively, IBC-LIC may be not applied to the BV candidate list reordering when the coding tool is IBC AMVP mode, and/or IBC regular merge mode, and/or IBC TM merge mode, and/or IBC-MBVD merge mode.
5. In one example, when RR-IBC is used, IBC-LIC may be used.
- a. In one example, the positions/shapes/ of a template used for LIC may depend on whether RR-IBC is applied.
 - b. In one example, the original BV without adjusted according to the RR-IBC flip type may be used to derive the IBC-LIC parameters.
 - c. In one example, the adjusted BV according to the RR-IBC flip type may be used to derive the IBC-LIC parameters.
 - d. In one example, the original template of the reference block may be used to derive the IBC-LIC parameters.
 - i. In one example, the original template of the reference block may be constrained in the IBC buffer.
 - ii. In one example, when the original template is not in the IBC buffer, IBC-LIC may be not used.

- 1) Alternatively, when the original template is not in the IBC buffer, the original template may be padded using the samples in the IBC buffer and then used in IBC-LIC.
- e. In one example, the adjusted template of the reference block may be used to derive the IBC-LIC parameters.
 - i. In one example, the adjusted template of the reference block may be constrained in the IBC buffer.
 - ii. In one example, the adjusted template of the reference block may be adjusted according to the RR-IBC flip type.
 - 1) In one example, when RR-IBC flip type is horizontal, the adjusted left and above template of the reference template may be used. An example is shown in Fig. 18.
 - 2) In one example, when RR-IBC flip type is vertical, the adjusted left and above template of the reference template may be used. An example is shown in Fig. 19.
 - iii. In one example, when the adjusted template is not in the IBC buffer, IBC-LIC may be not used.
 - 1) Alternatively, when the adjusted template is not in the IBC buffer, the adjusted template may be padded using the samples in the IBC buffer and then used in IBC-LIC.
- f. In one example, IBC-LIC may be used with a specific RR-IBC flip type.
 - i. In one example, the flip type may be horizontal.
 - ii. In one example, the flip type may be vertical.
6. In one example, whether to and/or how to apply IBC-LIC may depend on the coding information including:
 - a. block dimensions and/or block size
 - i. In one example, the block is allowed to be coded with IBC-LIC when the block size ($W \times H$) is less than or equal to a threshold ($T1$), wherein W and H denotes block width and block height, respectively.
 - 1) In one example, $T1 = 256$, or 512 , or 1024 , or 2048 , or 4096 .
 - 2) In one example, $T1$ may depend on whether IBC AMVP mode or IBC merge mode is used.
 - 3) In one example, $T1$ may depend on slice/picture type.
 - ii. In one example, the block is allowed to be coded with IBC-LIC when the block size ($W \times H$) is larger than or equal to a threshold ($T2$), wherein W and H denotes block width and block height, respectively.
 - 1) In one example, $T2 = 4$, or 8 , or 16 , or 32 , or 64 , or 128 , or 256 .
 - 2) In one example, $T2$ may depend on whether IBC AMVP mode or IBC merge mode is used.
 - 3) In one example, $T2$ may depend on slice/picture type.
 - iii. In one example, the block is allowed to be coded with IBC-LIC when the block size W and/or H is less than or equal to a threshold ($T1'$), wherein W and H denotes block width and block height, respectively.
 - 1) In one example, $T1' = 4$, or 8 , or 16 , or 32 , or 64 , or 128 , or 256 .
 - 2) In one example, $T1'$ may depend on whether IBC AMVP mode or IBC merge mode is used.

- 3) In one example, $T1'$ may depend on slice/picture type.
- iv. In one example, the block is allowed to be coded with IBC-LIC when the block size W and/or H is larger than or equal to a threshold ($T2'$), wherein W and H denotes block width and block height, respectively.
- 5 1) In one example, $T2' = 4$, or 8, or 16, or 32, or 64, or 128, or 256.
- 2) In one example, $T2'$ may depend on whether IBC AMVP mode or IBC merge mode is used.
- 3) In one example, $T2'$ may depend on slice/picture type.
- 10 v. In one example, the block size may refer to luma block size.
- vi. In one example, the block size may refer to chroma block size.
- b. the coded information may refer to the depth of a block.
- c. slice/picture type and/or partition tree type (single, or dual tree, or local dual tree)
 - i. In one example, IBC-LIC may be only applied to I slice/picture.
- d. block location
- 15 e. quantization parameter
- f. colour component
- 7. In one example, more than one LIC equations may be used to compensate the prediction (reconstruction) of a video unit which is derived using IBC.
 - 20 a. In one example, the multiple LIC types may refer to different LIC equations with adjustment parameters for one or more existing parameters of LIC (e.g., α and β).
 - i. In one example, an adjustment parameter may be used to adjust α , such as $\alpha + u$ or $\alpha \times u$.
 - 25 ii. In one example, an adjustment parameter may be used to adjust β , such as $\beta + v$ or $\beta \times v$.
 - b. In one example, the parameters of more than one LIC equations may be derived using different templates.
 - i. In one example, different sample lines of the template may be used.
 - 30 ii. In one example, left, or above, or left-above, left-below, above-right templates may be used.
 - iii. In one example, the templates may be adjacent and/or non-adjacent.
 - iv. In one example, samples from different positions in the template may be used.
 - 1) In one example, the positions may refer to down-sampling positions.
 - 35 v. In one example, samples in different categories may be used.
 - 1) In one example, the different categories may be classified depending on the samples of the template.
 - 2) In one example, a mean value of the samples in the template may be used to derive the different categories.
 - 40 c. In one example, more than one method may be used to derive the parameters for IBC-LIC.
 - i. In one example, which method is used may signalled, or per-defined, or determined using coding information.
 - 45 d. In one example, more than one method and/or more than templates may be used to derive more than one set of parameters, wherein the prediction signals of the

current blocks modified by using the more than one set of parameters may be fused.

- i. In one example, the weights used for fusion may be pre-defined, or signalled, or determined using coding information.
- 5 e. In one example, whether to and how to apply one of the more than one LIC equations may be indicated using a syntax element which is signalled in the bitstream.
- f. In one example, whether to and how to apply one of the more than one LIC equations may be determined adaptively.
- 10 8. In one example, at least one parameter in an IBC-LIC equation and/or at least one IBC-LIC equation used for the video units coded/decoded before the current video unit may be reused for the current video unit.
 - a. In one example, the video units coded/decoded before current video unit may be in a different slice/tile/picture/CTU/CTU row.
 - 15 b. In one example, the video units coded/decoded before current video unit may be in the same slice/tile/picture/CTU/CTU row as the current video unit.
 - c. In one example, the video units coded/decoded before current video unit may be the neighbouring spatial (e.g., adjacent and/or non-adjacent) video units.
 - d. In one example, the reused parameters may be stored in a list/table (e.g., HMVP IBC-LIC parameter table).
 - 20 i. In one example, the list/table may be updated during the coding/decoding process.
 - ii. In one example, the maximum size of the list/table may be pre-defined, or signalled, or derived.
 - 25 iii. In one example, the list/table may be re-initialized at the beginning of a slice/tile/picture/CTU/CTU.
 - 1) In one example, the list/table may be re-initialized as a void list/table.
 - 2) In one example, the list/table may be re-initialized using one or more pre-defined/derived/signalled parameters.
 - 30 iv. In one example, how to and/or whether to use/update the list/table may depend on coding information.
 - 1) In one example, the coding information may refer to block dimensions/size/position.
 - e. In one example, when the current video unit is chroma video unit, the reused parameters may come from luma video unit and/or chroma video unit.
 - 35 9. In one example, IBC-LIC may be applied to the sub-blocks within the current video unit.
 - a. In one example, whether to and/or how to apply IBC-LIC to the sub-blocks may be pre-defined, or signalled, or determined.
 - i. In one example, a syntax element may be used to indicate whether IBC-LIC is applied with sub-block level.
 - 40 ii. In one example, whether to and/or how to apply IBC-LIC to the sub-blocks may depend on block dimensions/size.
 - iii. In one example, IBC-LIC may be applied to all sub-blocks or not applied to all sub-blocks.
 - 45 iv. In one example, IBC-LIC may be applied to partial sub-blocks and IBC-LIC may be not applied to the other sub-blocks.
 - 1) In one example, the partial sub-blocks to which IBC-LIC is applied to may be the boundary sub-blocks.

- b. In one example, the IBC-LIC parameters may be derived different for at least one sub-block.
 - c. In one example, the IBC-LIC parameters may be shared by at least one sub-block.
- 10. In one example, the positions/shape of the template may depend on coding information.
 - a. In one example, if the left neighboring samples are unavailable, the template only contains the above neighbouring samples.
 - b. In one example, if the above neighboring samples are unavailable, the template only contains the left neighbouring samples.
 - c. In one example, if the left and above neighboring samples are unavailable, IBC-LIC may not be applicable.
 - d. In one example, the template may refer to the template of the current block or the reference block.
 - e. In one example, the positions/shape of the template may depend on whether RR-IBC or normal IBC is applied.
 - f. In one example, the positions/shape of the template may consist of one or more sample lines.
 - g. In one example, the positions/shape of the template may be pre-defined, or signalled, or derived on-the-fly.
 - h. In one example, the positions/shape of the template may depend on the width and height of the video unit.
 - i. In one example, the reference template may be constrained in the IBC buffer.
 - i. Alternatively, the reference template may be not constrained in the IBC buffer.
- 11. The determination of whether a block is allowed to be coded with IBC-LIC mode may depend on coded information including:
 - a. block dimensions and/or block size.
 - i. In one example, the block is allowed to be coded with IBC-LIC mode when the block size ($W \times H$) is less than or equal to a threshold (T), wherein W and H denotes block width and block height, respectively.
 - 1) In one example, $T = 256$, or 512 , or 1024 , or 2048 , or 4096 .
 - 2) In one example, T may depend on whether IBC AMVP mode or IBC merge mode is used.
 - ii. In one example, the block is allowed to be coded with IBC-LIC when the block size ($W \times H$) is larger than or equal to a threshold (T_4), wherein W and H denotes block width and block height, respectively.
 - 1) In one example, $T_4 = 4$, or 8 , or 16 , or 32 , or 64 , or 128 , or 256 , or 512 , or 1024 , or 2048 .
 - 2) In one example, T_4 may depend on whether IBC AMVP mode or IBC merge mode is used.
 - iii. In one example, the block is allowed to be coded with IBC-LIC when the block size W and/or H is less than or equal to a threshold (T_3'), wherein W and H denotes block width and block height, respectively.
 - 1) In one example, $T_3' = 4, 8, 16, \text{ or } 32, \text{ or } 64, \text{ or } 128, \text{ or } 256$.
 - 2) In one example, T_3' may depend on whether IBC AMVP mode or IBC merge mode is used.

- iv. In one example, the block is allowed to be coded with IBC-LIC when the block size W and/or H is larger than or equal to a threshold ($T4'$), wherein W and H denotes block width and block height, respectively.
 - 1) In one example, $T4' = 4, 8, 16, \text{ or } 32, \text{ or } 64, \text{ or } 128, \text{ or } 256$.
 - 2) In one example, $T4'$ may depend on whether IBC AMVP mode or IBC merge mode is used.
 - v. In above examples, T, $T4$, $T3'$, $T4'$ may depend on slice/picture type.
 - vi. In one example, the block size may refer to luma block size.
 - vii. In one example, the block size may refer to chroma block size.
- 10 b. depth of a block.
- c. block location.
- d. slice/picture type.
- e. temporal layer (e.g., temporal layer index).
- f. colour format.
- 15 g. colour component.
12. In one example, whether to and/or how to apply IBC-LIC may depend on colour format and/or colour components.
- a. In one example, IBC-LIC may apply to all colour components.
 - b. In one example, whether to and/or how to apply IBC-LIC to a first component may depend on whether to apply IBC-LIC to a second component.
 - i. In one example, the first component may refer to chroma component (e.g., Cb and/or Cr), and the second component may refer to luma component (e.g., Y).
 - ii. In one example, the way to apply IBC-LIC to the first component may be same as the second component.
 - 1) Alternatively, the way to apply IBC-LIC to the first component may be different from the second component.
 - iii. In one example, the determination of whether to and/or how to apply IBC-LIC to a first video unit in the first component may depend on a second video unit in the second component.
 - 1) In one example, the second video unit may be the collocated luma video unit of the first video unit.
 - a) In one example, $P_L(x, y) = P_C(x * \text{SubWidthC}, y * \text{SubHeightC})$, Wherein P_C and P_L denote the positions of the first video unit and the second video unit respectively.
 - b) In one example, ($P_C(x, y)$) may refer to the center position of the chroma video unit. Denote the width and height of the chroma video unit as W and H, and x is in the range of 0 and $W - 1$, inclusive, and y is in the range of 0 and $H - 1$, inclusive.
 - i. In one example, $(P_C(x, y)) = (W / 2 - 1, H / 2 + 1)$.
 - ii. In one example, $(P_C(x, y)) = (W / 2 + 1, H / 2 - 1)$.
 - iii. In one example, $(P_C(x, y)) = (W / 2 - 1, H / 2 + 1)$.
 - iv. In one example, $(P_C(x, y)) = (W / 2 + 1, H / 2 + 1)$.

- c) In one example, $(P_c(x, y))$ may refer to the left-top/right-top/left-bottom/right-bottom position of the chroma video unit.
 - 2) In one example, the second video unit may be the collocated luma video unit, or the spatial neighbouring (adjacent and/or non-adjacent) luma video units of the collocated luma video unit.
 - a) In one example, the spatial neighbouring luma video units may refer to left, left-below, left-above, above, above-right neighbouring luma video units.
 - b) In one example, the spatial neighbouring (adjacent and/or non-adjacent) luma video units of the collocated luma video unit may be used when the collocated luma video unit is not coded with IBC mode or IntraTMP mode.
 - 3) In one example, IBC-LIC may be applied to the first video unit when the second video unit is coded with IBC mode or IntraTMP mode.
- c. In one example, IBC-LIC may apply to luma component, but not to chroma components.
 - i. In one example, luma component may refer to Y in YCbCr colour space or G in RGB colour space.
 - ii. In one example, chroma components may refer to Cb and/or Cr in YCbCr colour space or R and/or B in RGB colour space.
- d. In one example, the derivation methods of IBC-LIC parameters for a first component may be same as a second component.
 - i. Alternatively, the derivation methods of IBC-LIC parameters for a first component may be different from a second component.
- e. In one example, how to derive the IBC-LIC parameters may depend on colour component and/or colour format.
 - i. In one example, the number of samples used to derive IBC-LIC parameters for chroma components may depend on colour format.
 - 1) In one example, the number of samples used chroma components may be same as luma component for 4:4:4 colour format.
 - a) Alternatively, the number of samples used chroma components may be different from luma component for 4:4:4 colour format.
 - 2) In one example, the number of samples used chroma components may be different from luma component for 4:2:0/4:2:2 colour format.
 - 3) In one example, the minimum samples used to derive IBC-LIC parameters for chroma components may be same as luma component regardless of colour format.
- f. In one example, at least one parameter may be not derived for a second component (e.g., chroma component such as Cb and/or Cr) but inherited from a first component (e.g., Y).
- g. In one example, the IBC-LIC parameters may be derived individually for different components.
 - i. Alternatively, at least one parameter may be shared by different components.
 - ii. In one example, the different components may refer to luma and chroma components.

- iii. In one example, the different components may refer to chroma components (e.g., Cb and Cr).
- h. In one example, when a luma BV is used for chroma components, it may be modified depending on the colour format.
 - i. In one example, $BV_C = BV_L (x \gg (\text{SubWidthC} - 1), y \gg (\text{SubHeightC} - 1))$.
 - ii. In one example, $BV_C = BV_L ((x + 1) \gg (\text{SubWidthC} - 1), (y + 1) \gg (\text{SubHeightC} - 1))$.
 - iii. In one example, $BV_C = BV_L ((x - 1) \gg (\text{SubWidthC} - 1), (y - 1) \gg (\text{SubHeightC} - 1))$.
 - iv. In one example, $BV_C = BV_L ((x + 1) \gg (\text{SubWidthC} - 1), (y - 1) \gg (\text{SubHeightC} - 1))$.
 - v. In one example, $BV_C = BV_L ((x - 1) \gg (\text{SubWidthC} - 1), (y + 1) \gg (\text{SubHeightC} - 1))$.
- 1) In one example, BV_C may be used to derive the prediction of the chroma video unit.
- 2) In one example, BV_C may be used to derive the reference template used to derive IBC-LIC parameters for chroma video unit.
- 3) In one example, when BV_C is fractional BV, it may be rounded to the integer BV.
 - a) In one example, clipping operation may be used when rounding the fractional BV.
 - b) Alternatively, the fractional BV is still used to derive the reference template.
- i. In one example, the same interpolation method is used to derive the reference template as the interpolation method to derive the prediction of the chroma video unit.
- ii. Alternatively, a different interpolation method is used to derive the reference template as the interpolation method to derive the prediction of the video unit, such as different interpolation taps.
 - 1. In one example, 2-tap interpolation filter may be used.

35 On signalling of IBC-LIC

- 13. Indication of the IBC-LIC mode may be conditionally signalled wherein the condition may include:
 - a. whether a specific coding method is allowed, such as IBC (IBC AMVP or IBC merge), or RR-IBC, or CIBCIP (IBC-CIIP), or IBC-TM, or IBC-GPM
 - b. block dimensions and/or block size
 - i. In one example, the indication of the IBC-LIC mode may be not signalled when the block size ($W \times H$) is less than or equal to a threshold ($T3$), wherein W and H denotes block width and block height, respectively.
 - 1) In one example, $T3 = 256$, or 512 , or 1024 , or 2048 , or 4096 .
 - 2) In one example, $T3$ may depend on whether IBC AMVP mode or IBC merge mode is used.

- 3) In one example, T3 may depend on slice/picture type.
- ii. In one example, the indication of the IBC-LIC mode may be not signalled when the block size ($W \times H$) is smaller than or equal to a threshold (T4), wherein W and H denotes block width and block height, respectively.
- 5 1) In one example, T4 = 16, or 32, or 64, or 128, or 256.
- 2) In one example, T4 may depend on whether IBC AMVP mode or IBC merge mode is used.
- 3) In one example, T4 may depend on slice/picture type.
- iii. In one example, the block size may refer to luma block size.
- 10 c. block depth
- d. slice/picture type and/or partition tree type (single, or dual tree, or local dual tree)
- e. temporal layer identification
- f. block location
- g. colour component
- 15 h. In one example, the indication of the IBC-LIC mode may be not signalled but derived.
- i. In one example, if the indication of the IBC-LIC mode is not signalled, it may be inferred to be a default value.
- i. In one example, if the indication of the IBC-LIC mode is not signalled, it may be inferred to be false.
- 20 ii. In one example, if the indication of the IBC-LIC mode is not signalled, it may be inferred to be true.
14. Whether current block is coded with IBC-LIC mode may be signalled using one or more syntax elements.
- 25 a. In one example, the syntax element may be binarized with fixed length coding, or truncated unary coding, or unary coding, or EG coding, or coded a flag.
- b. In one example, the syntax element may be bypass coded or context coded.
- i. The context may depend on coded information, such as block dimensions, and/or block size, and/or slice/picture types, and/or the information of neighbouring blocks (adjacent or non-adjacent), and/or the information of other coding tools used for current block, and/or the information of temporal layer.
- 30 1) In one example, the context may depend on whether the neighbouring blocks are coded with IBC-LIC.
- 35 c. In one example, the indication of IBC-LIC mode may be signalled when current video unit is IBC coded.
- d. In one example, the indication of IBC-LIC mode may be not signalled when current video unit is IBC merge mode.
- e. In one example, the syntax element may be signalled before or after the indication of a specific coding tool.
- 40 i. In one example, the specific coding tool may refer to RR-IBC mode, or IBC-TM mode, or IBC-MBVD mode, or CIBCIP (IBC-CIIP), or IBC-GPM.
- ii. In one example, whether to signal and/or how to the syntax element may be dependent on whether IBC mode, RR-IBC mode, or IBC-TM mode, or IBC-MBVD mode, or CIBCIP (IBC-CIIP), or IBC-GPM is enabled for the video unit.
- 45

- iii. In one example, the syntax element may be signalled after the indication of RR-IBC mode.
 - 1) In one example, when RR-IBC mode is applied, the syntax element indicating IBC-LIC is not signalled and set to a default value which indicates IBC-LIC is not applied.
 - iv. In one example, the syntax element may be signalled when the video unit is IBC-AMVP mode.
 - f. In one example, the one or more syntax elements may be signalled at sequence header/picture header/SPS/VPS/DPS/DCI/PPS/APS/slice header/tile group header.
 - g. In one example, the syntax element may be coded in a predictive way.
 - h. For example, the syntax element of the current block may be predicted by that of a neighboring block.
 - i. In one example, whether a block is allowed to be coded with IBC-LIC mode may depend on one or more syntax elements.
 - i. In one example, the one or more syntax elements may be signalled as general constraints information.
 - 1) In one example, when a syntax element (e.g., `gci_no_ibc_lic_constraint_flag`) indicating general constraint on IBC-LIC is equal to X1 (e.g., $X1 = 0$ or $X1 = 1$), IBC-LIC shall be not allowed.
 - 2) In one example, when either a syntax element (e.g., `gci_no_ibc_constraint_flag`) indicating general constraint on IBC is equal to X2 (e.g., $X2 = 0$ or $X2 = 1$), IBC-LIC shall be not allowed.
 - ii. In one example, the one or more syntax elements may be signalled at sequence header/picture header/SPS/VPS/DPS/DCI/PPS/APS/slice header/tile group header.
15. Whether to and/or how to apply the disclosed methods above may depend on the video characteristics.
- a. In one example, the video characteristics may refer to screen content, camera-captured/natural content, mixed content.
 - i. In one example, one or more syntax elements may be used to indicate whether the video unit belongs to which content.
 - ii. In one example, which content the video unit belongs to may be determined adaptively, such as using the neighbouring reconstructed/original samples and/or prediction/original samples of the video unit, hash value.
 - iii. In one example, which content the video unit belongs to may be determined adaptively, such as whether one or more neighbouring (adjacent or non-adjacent) video units are coded with a specific mode (e.g., IBC, Palette, IntraTMP).
16. The disclosed methods above may apply to other coding tools.
- a. In one example, the coding tool may refer to inter prediction, such as LIC.
 - b. In one example, the coding tool may refer to intra prediction, such as IntraTMP.
 - c. In one example, the coding tool may refer to in-loop filter.

General aspects

17. In above examples, the video unit may refer to the colour component/sub-picture/slice/tile/coding tree unit (CTU)/CTU row/groups of CTU/coding unit (CU)/prediction unit (PU)/transform unit (TU)/ coding tree block (CTB)/ coding block (CB)/prediction block(PB)/transform block (TB)/a block/sub-block of a block/sub-region within a block/ any other region that contains more than one sample or pixel.

18. Whether to and/or how to apply the disclosed methods above may be signalled at sequence level/group of pictures level/picture level/slice level/tile group level, such as in sequence header/picture header/SPS/VPS/DPS/DCI/PPS/APS/slice header/tile group header.

19. Whether to and/or how to apply the disclosed methods above may be signalled at PB/TB/CB/PU/TU/CU/VPDU/CTU/CTU row/slice/tile/sub-picture/other kinds of region contains more than one sample or pixel.

20. Whether to and/or how to apply the disclosed methods above may be dependent on coded information, such as block size, colour format, single/dual tree partitioning, colour component, slice/picture type.

[0077] As used herein, the term “video unit” or “video block” may be a sequence, a picture, a slice, a tile, a brick, a subpicture, a coding tree unit (CTU)/coding tree block (CTB), a CTU/CTB row, one or multiple coding units (CUs)/coding blocks (CBs), one or multiple CTUs/CTBs, one or multiple Virtual Pipeline Data Unit (VPDU), a sub-region within a picture/slice/tile/brick. The term “reference line” may refer to a row and/or a column reconstructed samples adjacent to or non-adjacent to the current block, which is used to derive the intra prediction of current video unit via an interpolation filter along a certain direction, and the certain direction is determined by an intra prediction mode (e.g., conventional intra prediction with intra prediction modes), or derive the intra prediction of current video unit via weighting the reference samples of the reference line with a matrix or vector (e.g., MIP).

[0078] Fig. 20 illustrates a flowchart of a method 2000 for video processing in accordance with embodiments of the present disclosure. The method 2000 is implemented during a conversion between a video unit of a video and a bitstream of the video.

[0079] At block 2010, for a conversion between a video unit of a video and a bitstream of the video unit, it determines that whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to the video unit.

[0080] At block 2020, if the IBC-LIC is applied to the video unit, a compensation process is performed to a prediction/reconstruction of the video unit using a model of the IBC-LIC.

[0081] At block 2030, the conversion is performed based on the compensated

prediction/reconstruction of the video unit. In some embodiments, the conversion may include encoding the video unit into the bitstream. Alternatively, or in addition, the conversion may include decoding the video unit from the bitstream. In this way, it can improve coding efficiency and coding performance.

5 [0082] In some embodiments, one or more offsets are used to modify one or more parameters of the model. In an example, the one or more offsets are indicated. In another example, the one or more offsets are derived. In a further example, the one or more offsets are predefined.

[0083] In some embodiments, whether to and/or a way to use the one or more offsets to
10 modify the one or more parameters is indicated. Alternatively, whether to and/or a way to use the one or more offsets to modify the one or more parameters is predefined. In some other embodiments, whether to and/or a way to use the one or more offsets to modify the one or more parameters is determined based on coding information.

[0084] In some embodiments, the model may be $c_0 \times x_0 + c_1 \times x_1 + c_2 \times x_2 + \dots +$
15 $c_N \times x_N + B$, where c_i are the parameters and x_i denote the coding information, i is an integer number and in a range from 0 to N , N is a predefined number, B represents a bias parameter which is derived together with c_i . For example, the coding information comprises at least one of: reconstruction samples, reference samples, prediction samples, values calculated using the reconstruction samples, values calculated using the reference
20 samples, values calculated using the prediction samples, position information of the reconstruction samples, position information of the reference samples, or position information of the prediction samples.

[0085] In some embodiments, a current template including neighboring reconstructed samples of the video unit and a reference template are used to derive parameters of the
25 model. In this case, the reference template may be derived using a block vector (BV) that is used to obtain the prediction of the video unit.

[0086] In some embodiments, if the BV is a fractional BV, the fractional BV is rounded to an integer BV. For example, a clipping operation is used during rounding the fractional BV. Alternatively, if the BV is a fractional BV, the fractional BV is used to derive the
30 reference template.

[0087] In some embodiments, an interpolation method used to derive the reference

template is same as an interpolation method used to derive the prediction of the video unit. Alternatively, an interpolation method used to derive the reference template is different as an interpolation method used to derive the prediction of the video unit. For example, the interpolation method used to derive the reference template and the interpolation method used to derive the prediction of the video unit have different interpolation taps. In some embodiments, 2-tap interpolation filter is used to derive the reference template.

[0088] In some embodiments, a way solving a model used for another coding tool is used to derive parameters of the model. In one example, the method solving a linear/non-linear equation (e.g., MSE minimization) used for other coding tools may be used to derive the parameters.

[0089] In some embodiments, the way is Cholesky decomposition used in adaptive loop filtering (ALF) and one or more variants of Cholesky decomposition. In some other embodiments, the way is LDL decomposition used in convolutional cross-component model (CCCM) and one or more variants of LDL decomposition. Alternatively, the way is Gaussian elimination used in CCCM and one or more variants of Gaussian elimination.

[0090] In some embodiments, a mean square error (MSE) minimization is used as a metric to solve the model. For example, one or more other metrics used to solve the model. As an example, the one or more other metrics comprises at least one of: sum of squared error (SSE) or a mean of SSE. In some embodiments, a subjective metric is used to solve the model. For example, the subjective metric comprises a structure similarity index measure (SSIM).

[0091] In some embodiments, more than one way is used to derive parameters for IBC-LIC. In some embodiments, which way is used to derive the parameters is indicated. Alternatively, which way is used to derive the parameters is predefined. In some other embodiments, which way is used to derive the parameters is determined based coding information.

[0092] In some embodiments, more than one way is used to derive more than one set of parameters, and predictions of the current video unit modified using the more than one set of parameters are fused. In some embodiments, weights used for the fusion are predefined. Alternatively, the weights used for the fusion are indicated. In some other embodiments, the weights used for the fusion are determined based on coding information.

[0093] In some embodiments, the IBC-LIC is used in a process of reordering a BV candidate list. For example, two-pass reordering is used to reorder the BV candidate list. In some embodiments, the IBC-LIC is applied in at least one of: a first pass reordering, or a second pass reordering. In some embodiments, the IBC-LIC is applied to partial BV candidates in the BV candidate list.

[0094] In some embodiments, whether to and/or an approach to apply the IBC-LIC depends on coding information of the video unit. The coding information of the video unit may include at least one of: a block dimension, a block size, a depth of the video unit, a slice type, a picture type, a partition tree type, a block location, a quantization parameter, or a colour component.

[0095] In some embodiments, a block is allowed to be coded with the IBC-LIC, if a block size is larger than or equal to a first threshold, where the block size is equal to WH , and W and H denotes block width and block height of the block, respectively. In some embodiments, the second threshold is equal to one of: 4, 8, 16, 32, 64, 128, or 256.

[0096] In some other embodiments, a block is allowed to be coded with the IBC-LIC, if a block size is less than or equal to a second threshold, where the block size is equal to WH , and W and H denotes block width and block height of the block, respectively. In some embodiments, the second threshold is equal to one of: 4, 8, 16, 32, 64, 128, or 256. In some embodiments, the second threshold depends on whether IBC advanced motion vector prediction (AMVP) mode or IBC merge mode is used. In some other embodiments, the second threshold depends on slice type or picture type.

[0097] In some embodiments, a block is allowed to be coded with the IBC-LIC, if a block size is larger than or equal to a third threshold, where the block size is equal to WH , and W and H denotes block width and block height of the block, respectively. In some embodiments, the third threshold is equal to one of: 4, 8, 16, 32, 64, 128, or 256. In some other embodiments, the third threshold depends on whether IBC AMVP mode or IBC merge mode is used. In some further embodiments, the third threshold depends on slice type or picture type. In some embodiments, the block size refers to chroma block size.

[0098] In some embodiments, more than one LIC equation is used to compensate the prediction or reconstruction of the video unit, and parameters of the more than one LIC equations are derived using different templates. In some embodiments, at least one of the following templates is used to derive the parameters: left template, above template, left-

above template, left-below template, or above-right template. In some embodiments, the templates are adjacent. Alternatively, the templates are non-adjacent.

[0099] In some embodiments, more than one way is used to derive parameters for IBC-LIC. In some embodiments, which way is used to derive the parameters is indicated.

5 Alternatively, which way is used to derive the parameters is predefined. In some other embodiments, which way is used to derive the parameters is determined based on coding information.

[0100] In some embodiments, more than one way and/or more than one template is used to derive more than one set of parameters of the model, and predictions of the video unit

10 modified by the more than one set of parameters are fused. In some embodiments, weights used for the fusion are pre-defined. Alternatively, the weights used for the fusion are indicated. In some other embodiments, the weights used for the fusion are determined based on coding information.

[0101] In some embodiments, one or more of: at least one parameter in the model or at

15 least one IBC-LIC model used for video units coded before a current video unit are used for the current video unit. Alternatively, or in addition, one or more of: at least one parameter in the model or at least one IBC-LIC model used for video units decoded before a current video unit are used for the current video unit.

[0102] In some embodiments, the video units coded or decoded before the current video

20 unit are in one of: a different slice, a different tile, a different picture, a different coding tree unit (CTU), or a different CTU row. In some other embodiments, the video units coded or decoded before the current video unit are in one of: a same slice as the current video unit, a same tile as the current video unit, a same picture as the current video unit, a same CTU as the current video unit, or a same CTU row as the current video unit.

25 **[0103]** In some embodiments, the video units coded or decoded before the current video

unit are neighbouring spatial video units. In some embodiments, the neighbouring spatial video units are adjacent video units. Alternatively, or in addition, the neighbouring spatial video units are non-adjacent video units.

[0104] In some embodiments, reused parameters are stored in a list or a table. For

30 example, the table is a history-based motion vector prediction IBC-LIC parameter table. In some embodiments, the list or table is updated during a coding process. Alternatively,

or in addition, the list or table is updated during a decoding process.

[0105] In some embodiments, a maximum size of the list or table is predefined. Alternatively, the maximum size of the list or table is indicated. In some other embodiments, the maximum size of the list or table is derived.

- 5 [0106] In some embodiments, the list or table is re-initialized at a beginning of one of: a slice, a tile, a picture, or a CTU. In some embodiments, the list is re-initialized as a void list. Alternatively, the table is re-initialized as a void table. In some other embodiments, the list or table is re-initialized using one or more parameters. In some embodiments, the one or more parameters are predefined. Alternatively, the one or more parameters are
10 derived. In some other embodiments, the one or more parameters are indicated.

- [0107] In some embodiments, whether to and/or how to use the list or table depends on coding information. Alternatively, whether to and/or how to update the list or table depends on coding information. In some embodiments, the coding information comprises at least one of: block dimensions, block size, or block position. In some embodiments, if
15 the current video unit is chroma video unit, reused parameters are from at least one of luma video unit or chroma video unit.

- [0108] In some embodiments, the IBC-LIC is applied to sub-blocks within the video unit which is a current video unit. In some embodiments, whether to and/or how to apply the IBC-LIC to the sub-blocks is predefined. Alternatively, whether to and/or how to apply
20 the IBC-LIC to the sub-blocks is indicated. For example, a syntax element is used to indicate whether the IBC-LIC is applied with sub-block level. In some other embodiments, whether to and/or how to apply the IBC-LIC to the sub-blocks is determined. In some embodiments, whether to and/or how to apply the IBC-LIC to the sub-blocks depends on block dimensions or block size.

- 25 [0109] In some embodiments, the IBC-LIC is applied to all sub-blocks. Alternatively, the IBC-LIC is not applied to all sub-blocks.

[0110] In some embodiments, the IBC-LIC is applied to partial sub-blocks, and the IBC-LIC is not applied to other sub-blocks. In some embodiments, the partial sub-blocks to which the IBC-LIC is applied are boundary sub-blocks.

- 30 [0111] In some embodiments, IBC-LIC parameters are different for at least one sub-block. In some other embodiments, IBC-LIC parameters are shared by at least one sub-

block.

[0112] In some embodiments, the method 2000 may also include determining whether the video unit is allowed to be coded with an IBC with LIC mode based on coded information of the video unit. In some embodiments, the coded information comprises at least one of: a block dimension, a block size, a depth of a block, a block location, a slice type, a picture type, a temporal layer, a colour format, or a colour component.

[0113] In some embodiments, if the block size that is represented as WH is less than or equal to a fourth threshold, the video unit is allowed to be coded with the IBC with LIC mode, where W represents a block width of the video unit and H represents a block height of the video unit. In some embodiments, the fourth threshold depends on whether IBC AMVP mode or IBC merge mode is used. Alternatively, the fourth threshold depends on slice type or picture.

[0114] In some embodiments, if the block size that is represented as WH is larger than or equal to a fifth threshold, the video unit is allowed to be coded with the IBC with LIC mode, where W represents a block width of the video unit and H represents a block height of the video unit. In some embodiments, the fifth threshold is one of: 4, 8, 16, 32, 64, 128, 256, 512, 1024, or 2048. In some other embodiments, the fifth threshold depends on whether IBC AMVP mode or IBC merge mode is used. Alternatively, the fifth threshold depends on slice type or picture.

[0115] In some embodiments, if the block size that is represented as WH is less than or equal to a sixth threshold, the video unit is allowed to be coded with the IBC with LIC mode, where W represents a block width of the video unit and H represents a block height of the video unit. In some embodiments, the sixth threshold is one of: 4, 8, 16, 32, 64, 128, or 256. In some other embodiments, the sixth threshold depends on whether IBC AMVP mode or IBC merge mode is used. Alternatively, the sixth threshold depends on slice type or picture.

[0116] In some embodiments, if the block size that is represented as WH is larger than or equal to a seventh threshold, the video unit is allowed to be coded with the IBC with LIC mode, where W represents a block width of the video unit and H represents a block height of the video unit. In some embodiments, the seventh threshold is one of: 4, 8, 16, 32, 64, 128, or 256. In some other embodiments, the seventh threshold depends on whether IBC AMVP mode or IBC merge mode is used. Alternatively, the seventh threshold

depends on slice type or picture.

[0117] In some embodiments, the block size is luma block size. In some other embodiments, the block size is chroma block size.

[0118] In some embodiments, a determination of whether to and/or how to apply the IBC-LIC to a first video unit in a first component depends on a second video unit in a second component. In some embodiments, the second video unit is a collocated luma video unit of the first video unit. In some embodiments, $P_L(x, y) = P_C(x * \text{SubWidthC}, y * \text{SubHeightC})$, and where P_C represents a position of the first video unit, P_L represents a position of the second video unit, SubWidthC represents a subblock width, SubHeightC represents a subblock height, x represents a coordinate in horizontal direction, y represents a coordinate in a vertical direction.

[0119] In one example, $(P_C(x, y))$ may refer to the center position of the chroma video unit. The width and height of the chroma video unit may be represented as W and H , and x is in the range of 0 and $W - 1$, inclusive, and y is in the range of 0 and $H - 1$, inclusive. In one example, $(P_C(x, y)) = (W / 2 - 1, H / 2 + 1)$. In one example, $(P_C(x, y)) = (W / 2 + 1, H / 2 - 1)$. In one example, $(P_C(x, y)) = (W / 2 - 1, H / 2 + 1)$. In one example, $(P_C(x, y)) = (W / 2 + 1, H / 2 + 1)$. In some embodiments, $(P_C(x, y))$ refers to one of: a left-top position of the chroma video unit, a right-top position of the chroma video unit, a left-bottom position of the chroma video unit, or a right-bottom position of the chroma video unit.

[0120] In some embodiments, the second video unit is a collocated luma video unit of the first video unit. In some embodiments, the second video unit is a spatial neighbouring luma video unit of the collocated luma video unit. For example, the spatial neighbouring luma video unit is an adjacent neighbouring luma video unit. Alternatively, the spatial neighbouring luma video unit may be a non-adjacent neighbouring luma video unit.

[0121] In some embodiments, the spatial neighbouring luma video unit comprises one of: a left neighbouring luma video unit, a left-below neighbouring luma video unit, a left-above neighbouring luma video unit, an above neighbouring luma video unit, or an above-right neighbouring luma video unit. In some other embodiments, the spatial neighbouring luma video unit of the collocated luma video unit is used, if the collocated luma video unit is not coded with IBC mode or intra template matching prediction (IntraTMP) mode.

In some embodiments, the IBC-LIC is applied to the first video unit, if the second video unit is coded with IBC mode or IntraTMP mode.

[0122] In some embodiments, a derivation way of IBC-LIC parameters for a first component is same as a derivation way of IBC-LIC parameters for a second component.

5 In some other embodiments, the derivation way of IBC-LIC parameters for the first component is different from the derivation way of IBC-LIC parameters for the second component.

[0123] In some embodiments, a way to derive IBC-LIC parameters depends on at least one of colour component or colour format. In some embodiments, the number of samples used to derive the IBC-LIC parameters for chroma components depends on colour format. For example, the number of samples used chroma components is same as luma component for 4:4:4 colour format. Alternatively, the number of samples used chroma components is different from luma component for 4:4:4 colour format. In some embodiments, the number of samples used chroma components is different from luma component for 4:2:0/4:2:2 colour format. In some other embodiments, a minimum number of samples used to derive the IBC-LIC parameters for chroma components is same as luma component regardless of colour format.

[0124] In some embodiments, at least one parameter is not derived for a second component but inherited from a first component. In some embodiments, the first component is Y component, and the second component is a chroma component (such as Cb and/or Cr).

[0125] In some embodiments, IBC-LIC parameters are derived individually for different components. Alternatively, at least one IBC-LIC parameter is shared by different components. In some embodiments, the components comprise luma and chroma components. In some other embodiments, the components comprise chroma components (e.g., Cb and Cr).

[0126] In some embodiments, if a luma BV is used for chroma components, the luma BV is modified depending on the colour format. In one example, $BV_C = BV_L \left((x + 1) \gg (\text{SubWidthC} - 1), (y + 1) \gg (\text{SubHeightC} - 1) \right)$. In one example, $BV_C = BV_L \left((x - 1) \gg (\text{SubWidthC} - 1), (y - 1) \gg (\text{SubHeightC} - 1) \right)$. In one example, $BV_C = BV_L \left((x + 1) \gg (\text{SubWidthC} - 1), (y - 1) \gg (\text{SubHeightC} - 1) \right)$. In one

example, $BV_C = BV_L \left((x - 1) \gg (\text{SubWidthC} - 1), (y + 1) \gg (\text{SubHeightC} - 1) \right)$. In above cases, BV_L represents the luma BV, BV_C represents a chroma BV, SubWidthC represents a subblock width, SubHeightC represents a subblock height, x represents a coordinate in horizontal direction, y represents a coordinate in a vertical direction.

- 5 [0127] In some embodiments, BV_C is used to derive a prediction of a chroma video unit. In some other embodiments, BV_C is used to derive a reference template used to derive IBC-LIC parameters for a chroma video unit.

[0128] In some embodiments, if BV_C is a fractional BV, BV_C is rounded to an integer BV. For example, a clipping operation is used during rounding the fractional BV.

- 10 Alternatively, the fractional BV is used to derive the reference template.

[0129] In some embodiments, an interpolation method used to derive the reference template is same as an interpolation method used to derive the prediction of the chroma video unit. In some other embodiments, an interpolation method used to derive the reference template is different as an interpolation method used to derive the prediction of the chroma video unit.

- 15

[0130] In some embodiments, the interpolation method used to derive the reference template and the interpolation method used to derive the prediction of the video unit have different interpolation taps. For example, 2-tap interpolation filter is used to derive the reference template.

- 20 [0131] In some embodiments, whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using a model of the IBC-LIC depends on video characteristics. In some embodiments, the video characteristics comprises at least one of: screen content, camera-captured content, natural content, or mixed content.

- 25 [0132] In some embodiments, one or more syntax elements are used to indicate whether the video unit belongs to which content. In some embodiments, which content the video unit belongs to is determined adaptively. For example, which content the video unit belongs to is determined using at least one of: neighboring reconstructed samples of the video unit, neighboring original samples, prediction samples of the video unit, original samples of the video unit, hash value, or one or more neighboring video units are coded with a mode.
- 30

[0133] In some embodiments, the video unit is applied with other coding tools. In some embodiments, the other coding tool comprises an inter prediction, or wherein the other coding tool comprises an intra prediction, or wherein the other coding tool comprises an in-loop filter. In some embodiments, the inter prediction is a local illumination compensation (LIC). Alternatively, the intra prediction comprises IntraTMP.

[0134] In some embodiments, the video unit comprises at least one of: a color component, a prediction block (PB), a transform block (TB), a coding block (CB), a prediction unit (PU), a transform unit (TU), a coding tree block (CTB), a coding unit (CU), a coding tree unit (CTU), a CTU row, groups of CTU, a slice, a tile, a sub-picture, a block, a sub-region within a block, or a region containing more than one sample or pixel.

[0135] In some embodiments, an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated at one of the followings: sequence level, group of pictures level, picture level, slice level, or tile group level. In some embodiments, an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated in one of the following: a sequence header, a picture header, a sequence parameter set (SPS), a video parameter set (VPS), a dependency parameter set (DPS), a decoding capability information (DCI), a picture parameter set (PPS), an adaptation parameter sets (APS), a slice header, or a tile group header. In some embodiments, an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated at one of the following: a PB, a TB, a CB, a PU, a TU, a CU, a VPDU, a CTU, a CTU row, a slice, a tile, a sub-picture, or a region contains more than one sample or pixel.

[0136] In some embodiments, whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is coded information of the video unit. The coded information may include at least one of a block size, a colour format, a single tree partitioning, a dual tree partitioning, a colour component, a slice type, or a picture type.

[0137] According to further embodiments of the present disclosure, a non-transitory computer-readable recording medium is provided. The non-transitory computer-readable recording medium stores a bitstream of a video which is generated by a method performed

by an apparatus for video processing. The method comprises: determining whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to a video unit of the video; in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; and generating the bitstream based on the compensated prediction/reconstruction of the video unit.

[0138] According to still further embodiments of the present disclosure, a method for storing bitstream of a video is provided. The method comprises: determining whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to a video unit of the video; in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; generating the bitstream based on the compensated prediction/reconstruction of the video unit; and storing the bitstream in a non-transitory computer-readable recording medium.

[0139] 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.

[0140] Clause 1. A method of video processing, comprising: determining, for a conversion between a video unit of a video and a bitstream of the video unit, whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to the video unit based on a model used for the IBC-LIC; in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; and performing the conversion based on the compensated prediction/reconstruction of the video unit.

[0141] Clause 2. The method of clause 1, wherein one or more offsets are used to modify one or more parameters of the model.

[0142] Clause 3. The method of clause of clause 2, wherein the one or more offsets are indicated, or wherein the one or more offsets are derived, or wherein the one or more offsets are predefined.

[0143] Clause 4. The method of clause 2, wherein whether to and/or a way to use the one or more offsets to modify the one or more parameters is indicated, or wherein whether

to and/or a way to use the one or more offsets to modify the one or more parameters is predefined, or wherein whether to and/or a way to use the one or more offsets to modify the one or more parameters is determined based on coding information.

[0144] Clause 5. The method of clause 1, wherein the model is presented as: $c_0 \times x_0 + c_1 \times x_1 + c_2 \times x_2 + \dots + c_N \times x_N + B$, and wherein c_i represents a parameter and x_i represents coding information, i is an integer number and in a range from 0 to N , N is a predefined number, B represents a bias parameter which is derived together with c_i .

[0145] Clause 6. The method of clause 5, wherein the coding information comprises at least one of: reconstruction samples, reference samples, prediction samples, values calculated using the reconstruction samples, values calculated using the reference samples, values calculated using the prediction samples, position information of the reconstruction samples, position information of the reference samples, or position information of the prediction samples.

[0146] Clause 7. The method of clause 1, wherein a current template comprising neighboring reconstructed samples of the video unit and a reference template are used to derive parameters of the model, and the reference template is derived using a block vector (BV) that is used to obtain the prediction of the video unit.

[0147] Clause 8. The method of clause 7, wherein if the BV is a fractional BV, the fractional BV is rounded to an integer BV.

[0148] Clause 9. The method of clause 8, wherein a clipping operation is used during rounding the fractional BV.

[0149] Clause 10. The method of clause 7, wherein if the BV is a fractional BV, the fractional BV is used to derive the reference template.

[0150] Clause 11. The method of clause 10, wherein an interpolation method used to derive the reference template is same as an interpolation method used to derive the prediction of the video unit.

[0151] Clause 12. The method of clause 10, wherein an interpolation method used to derive the reference template is different as an interpolation method used to derive the prediction of the video unit.

[0152] Clause 13. The method of clause 12, wherein the interpolation method used to

derive the reference template and the interpolation method used to derive the prediction of the video unit have different interpolation taps.

[0153] Clause 14. The method of clause 12, wherein 2-tap interpolation filter is used to derive the reference template.

5 [0154] Clause 15. The method of clause 1, wherein a way solving a model used for another coding tool is used to derive parameters of the model.

[0155] Clause 16. The method of clause 15, wherein the way is Cholesky decomposition used in adaptive loop filtering (ALF) and a variant of Cholesky decomposition.

10 [0156] Clause 17. The method of clause 15, wherein the way is LDL decomposition used in convolutional cross-component model (CCCM) and one or more variants of LDL decomposition.

[0157] Clause 18. The method of clause 15, wherein the way is Gaussian elimination used in CCCM and one or more variants of Gaussian elimination.

15 [0158] Clause 19. The method of clause 1, wherein a mean square error (MSE) minimization is used as a metric to solve the model.

[0159] Clause 20. The method of clause 19, wherein one or more other metrics used to solve the model.

[0160] Clause 21. The method of clause 20, wherein the one or more other metrics comprises at least one of: sum of squared error (SSE) or a mean of SSE.

20 [0161] Clause 22. The method of clause 19, wherein a subjective metric is used to solve the model.

[0162] Clause 23. The method of clause 22, wherein the subjective metric comprises a structure similarity index measure (SSIM).

25 [0163] Clause 24. The method of clause 1, wherein more than one way is used to derive parameters for IBC-LIC.

[0164] Clause 25. The method of clause 24, wherein which way is used to derive the parameters is indicated, or wherein which way is used to derive the parameters is predefined, or wherein which way is used to derive the parameters is determined based coding information.

[0165] Clause 26. The method of clause 1, wherein more than one way is used to derive more than one set of parameters, and predictions of the current video unit modified using the more than one set of parameters are fused.

[0166] Clause 27. The method of clause 26, wherein weights used for the fusion are predefined, or wherein the weights used for the fusion are indicated, or wherein the weights used for the fusion are determined based on coding information.

[0167] Clause 28. The method of clause 1, wherein the IBC-LIC is used in a process of reordering a BV candidate list.

[0168] Clause 29. The method of clause 28, wherein two-pass reordering is used to reorder the BV candidate list.

[0169] Clause 30. The method of clause 29, wherein the IBC-LIC is applied in at least one of: a first pass reordering, or a second pass reordering.

[0170] Clause 31. The method of clause 28, wherein the IBC-LIC is applied to partial BV candidates in the BV candidate list.

[0171] Clause 32. The method of any of clauses 1-31, wherein whether to and/or an approach to apply the IBC-LIC depends on coding information of the video unit, and wherein the coding information of the video unit comprises at least one of: a block dimension, a block size, a depth of the video unit, a slice type, a picture type, a partition tree type, a block location, a quantization parameter, or a colour component.

[0172] Clause 33. The method of clause 32, wherein a block is allowed to be coded with the IBC-LIC, if a block size is larger than or equal to a first threshold, wherein the block size is equal to WH , and W and H denotes block width and block height of the block, respectively.

[0173] Clause 34. The method of clause 33, wherein the second threshold is equal to one of: 4, 8, 16, 32, 64, 128, or 256.

[0174] Clause 35. The method of clause 32, wherein a block is allowed to be coded with the IBC-LIC, if a block size is less than or equal to a second threshold, wherein the block size is equal to WH , and W and H denotes block width and block height of the block, respectively.

[0175] Clause 36. The method of clause 35, wherein the second threshold is equal to

one of: 4, 8, 16, 32, 64, 128, or 256.

[0176] Clause 37. The method of clause 35, wherein the second threshold depends on whether IBC advanced motion vector prediction (AMVP) mode or IBC merge mode is used.

5 [0177] Clause 38. The method of clause 35, wherein the second threshold depends on slice type or picture type.

[0178] Clause 39. The method of clause 32, wherein a block is allowed to be coded with the IBC-LIC, if a block size is larger than or equal to a third threshold, wherein the block size is equal to WH , and W and H denotes block width and block height of the block,
10 respectively.

[0179] Clause 40. The method of clause 39, wherein the third threshold is equal to one of: 4, 8, 16, 32, 64, 128, or 256.

[0180] Clause 41. The method of clause 39, wherein the third threshold depends on whether IBC AMVP mode or IBC merge mode is used.

15 [0181] Clause 42. The method of clause 39, wherein the third threshold depends on slice type or picture type.

[0182] Clause 43. The method of any of clauses 32-42, wherein the block size refers to chroma block size.

[0183] Clause 44. The method of clause 1, wherein more than one LIC equation is used
20 to compensate the prediction or reconstruction of the video unit, and parameters of the more than one LIC equations are derived using different templates.

[0184] Clause 45. The method of clause 44, wherein at least one of the following templates is used to derive the parameters: left template, above template, left-above template, left-below template, or above-right template.

25 [0185] Clause 46. The method of clause 45, wherein the templates are adjacent, and/or wherein the templates are non-adjacent.

[0186] Clause 47. The method of clause 1, wherein more than one way is used to derive parameters for IBC-LIC.

[0187] Clause 48. The method of clause 47, wherein which way is used to derive the

parameters is indicted, or wherein which way is used to derive the parameters is predefined, or wherein which way is used to derive the parameters is determined based on coding information.

[0188] Clause 49. The method of clause 1, wherein more than one way and/or more than one template is used to derive more than one set of parameters of the model, and predictions of the video unit modified by the more than one set of parameters are fused.

[0189] Clause 50. The method of clause 49, wherein weights used for the fusion are pre-defined, or wherein the weights used for the fusion are indicated, or wherein the weights used for the fusion are determined based on coding information.

[0190] Clause 51. The method of clause 1, wherein one or more of: at least one parameter in the model or at least one IBC-LIC model used for video units coded before a current video unit are used for the current video unit, and/or wherein one or more of: at least one parameter in the model or at least one IBC-LIC model used for video units decoded before a current video unit are used for the current video unit.

[0191] Clause 52. The method of clause 51, wherein the video units coded or decoded before the current video unit are in one of: a different slice, a different tile, a different picture, a different coding tree unit (CTU), or a different CTU row.

[0192] Clause 53. The method of clause 51, wherein the video units coded or decoded before the current video unit are in one of: a same slice as the current video unit, a same tile as the current video unit, a same picture as the current video unit, a same CTU as the current video unit, or a same CTU row as the current video unit.

[0193] Clause 54. The method of clause 51, wherein the video units coded or decoded before the current video unit are neighbouring spatial video units.

[0194] Clause 55. The method of clause 54, wherein the neighbouring spatial video units are adjacent video units, and/or wherein the neighbouring spatial video units are non-adjacent video units.

[0195] Clause 56. The method of clause 51, wherein reused parameters are stored in a list or a table.

[0196] Clause 57. The method of clause 56, wherein the table is a history-based motion vector prediction IBC-LIC parameter table.

[0197] Clause 58. The method of clause 56, wherein the list or table is updated during a coding process, and/or wherein the list or table is updated during a decoding process.

[0198] Clause 59. The method of clause 56, wherein a maximum size of the list or table is predefined, or wherein the maximum size of the list or table is indicated, or wherein the maximum size of the list or table is derived.

[0199] Clause 60. The method of clause 56, wherein the list or table is re-initialized at a beginning of one of: a slice, a tile, a picture, or a CTU.

[0200] Clause 61. The method of clause 60, wherein the list is re-initialized as a void list, or wherein the table is re-initialized as a void table.

[0201] Clause 62. The method of clause 60, wherein the list or table is re-initialized using one or more parameters.

[0202] Clause 63. The method of clause 62, wherein the one or more parameters are predefined, or wherein the one or more parameters are derived, or wherein the one or more parameters are indicated.

[0203] Clause 64. The method of clause 56, wherein whether to and/or how to use the list or table depends on coding information, or wherein whether to and/or how to update the list or table depends on coding information.

[0204] Clause 65. The method of clause 64, wherein the coding information comprises at least one of: block dimensions, block size, or block position.

[0205] Clause 66. The method of clause 51, wherein if the current video unit is chroma video unit, reused parameters are from at least one of luma video unit or chroma video unit.

[0206] Clause 67. The method of clause 1, wherein the IBC-LIC is applied to sub-blocks within the video unit which is a current video unit.

[0207] Clause 68. The method of clause 67, wherein whether to and/or how to apply the IBC-LIC to the sub-blocks is predefined, or wherein whether to and/or how to apply the IBC-LIC to the sub-blocks is indicated, or wherein whether to and/or how to apply the IBC-LIC to the sub-blocks is determined.

[0208] Clause 69. The method of clause 68, wherein a syntax element is used to indicate whether the IBC-LIC is applied with sub-block level.

[0209] Clause 70. The method of clause 68, wherein whether to and/or how to apply the IBC-LIC to the sub-blocks depends on block dimensions or block size.

[0210] Clause 71. The method of clause 68, wherein the IBC-LIC is applied to all sub-blocks, or wherein the IBC-LIC is not applied to all sub-blocks.

5 [0211] Clause 72. The method of clause 68, wherein the IBC-LIC is applied to partial sub-blocks, and the IBC-LIC is not applied to other sub-blocks.

[0212] Clause 73. The method of clause 72, wherein the partial sub-blocks to which the IBC-LIC is applied are boundary sub-blocks.

10 [0213] Clause 74. The method of clause 67, wherein IBC-LIC parameters are different for at least one sub-block.

[0214] Clause 75. The method of clause 67, wherein IBC-LIC parameters are shared by at least one sub-block.

15 [0215] Clause 76. The method of clause 1, further comprising determining whether the video unit is allowed to be coded with an IBC with LIC mode based on coded information of the video unit.

[0216] Clause 77. The method of clause 76, wherein the coded information comprises at least one of: a block dimension, a block size, a depth of a block, a block location, a slice type, a picture type, a temporal layer, a colour format, or a colour component.

20 [0217] Clause 78. The method of clause 77, wherein if the block size that is represented as WH is less than or equal to a fourth threshold, the video unit is allowed to be coded with the IBC with LIC mode, wherein W represents a block width of the video unit and H represents a block height of the video unit.

25 [0218] Clause 79. The method of clause 78, wherein the fourth threshold depends on whether IBC AMVP mode or IBC merge mode is used, or wherein the fourth threshold depends on slice type or picture.

[0219] Clause 80. The method of clause 77, wherein if the block size that is represented as WH is larger than or equal to a fifth threshold, the video unit is allowed to be coded with the IBC with LIC mode, wherein W represents a block width of the video unit and H represents a block height of the video unit.

30 [0220] Clause 81. The method of clause 80, wherein the fifth threshold is one of: 4, 8,

16, 32, 64, 128, 256, 512, 1024, or 2048.

[0221] Clause 82. The method of clause 80, wherein the fifth threshold depends on whether IBC AMVP mode or IBC merge mode is used, or wherein the fifth threshold depends on slice type or picture.

- 5 [0222] Clause 83. The method of clause 77, wherein if the block size that is represented as WH is less than or equal to a sixth threshold, the video unit is allowed to be coded with the IBC with LIC mode, wherein W represents a block width of the video unit and H represents a block height of the video unit.

[0223] Clause 84. The method of clause 83, wherein the sixth threshold is one of: 4, 8,
10 16, 32, 64, 128, or 256.

[0224] Clause 85. The method of clause 83, wherein the sixth threshold depends on whether IBC AMVP mode or IBC merge mode is used, or wherein the sixth threshold depends on slice type or picture.

[0225] Clause 86. The method of clause 77, wherein if the block size that is represented
15 as WH is larger than or equal to a seventh threshold, the video unit is allowed to be coded with the IBC with LIC mode, wherein W represents a block width of the video unit and H represents a block height of the video unit.

[0226] Clause 87. The method of clause 86, wherein the seventh threshold is one of: 4, 8, 16, 32, 64, 128, or 256.

20 [0227] Clause 88. The method of clause 86, wherein the seventh threshold depends on whether IBC AMVP mode or IBC merge mode is used, or wherein the seventh threshold depends on slice type or picture.

[0228] Clause 89. The method of clause 77, wherein the block size is luma block size.

[0229] Clause 90. The method of clause 77, wherein the block size is chroma block size.

25 [0230] Clause 91. The method of clause 1, wherein a determination of whether to and/or how to apply the IBC-LIC to a first video unit in a first component depends on a second video unit in a second component.

[0231] Clause 92. The method of clause 91, wherein the second video unit is a collocated luma video unit of the first video unit.

[0232] Clause 93. The method of clause 92, wherein $P_L(x, y) = P_C(x * \text{SubWidthC}, y * \text{SubHeightC})$, and wherein P_C represents a position of the first video unit, P_L represents a position of the second video unit, SubWidthC represents a subblock width, SubHeightC represents a subblock height, x represents a coordinate in horizontal direction, y represents a coordinate in a vertical direction.

[0233] Clause 94. The method of clause 92, wherein $(P_C(x, y)) = (W/2 - 1, H/2 + 1)$, or wherein $(P_C(x, y)) = (W/2 + 1, H/2 - 1)$, or wherein $(P_C(x, y)) = (W/2 - 1, H/2 + 1)$, or wherein $(P_C(x, y)) = (W/2 + 1, H/2 + 1)$, wherein P_C represents a center position of a chroma video unit, W represents a width of the chroma video unit, H represents a height of the chroma video unit, x is in a range from 0 to $W - 1$, inclusive, and y is in a range of 0 and $H - 1$ inclusive.

[0234] Clause 95. The method of clause 92, wherein $(P_C(x, y))$ refers to one of: a left-top position of the chroma video unit, a right-top position of the chroma video unit, a left-bottom position of the chroma video unit, or a right-bottom position of the chroma video unit.

[0235] Clause 96. The method of clause 91, wherein the second video unit is a collocated luma video unit of the first video unit.

[0236] Clause 97. The method of clause 91, wherein the second video unit is a spatial neighbouring luma video unit of the collocated luma video unit.

[0237] Clause 98. The method of clause 96, wherein the spatial neighbouring luma video unit is an adjacent neighbouring luma video unit, or wherein the spatial neighbouring luma video unit is a non-adjacent neighbouring luma video unit.

[0238] Clause 99. The method of clause 96, wherein the spatial neighbouring luma video unit comprises one of: a left neighbouring luma video unit, a left-below neighbouring luma video unit, a left-above neighbouring luma video unit, an above neighbouring luma video unit, or an above-right neighbouring luma video unit.

[0239] Clause 100. The method of clause 96, wherein the spatial neighbouring luma video unit of the collocated luma video unit is used, if the collocated luma video unit is

not coded with IBC mode or intra template matching prediction (IntraTMP) mode.

[0240] Clause 101. The method of clause 91, wherein the IBC-LIC is applied to the first video unit, if the second video unit is coded with IBC mode or IntraTMP mode.

[0241] Clause 102. The method of clause 1, wherein a derivation way of IBC-LIC parameters for a first component is same as a derivation way of IBC-LIC parameters for a second component.

[0242] Clause 103. The method of clause 1, wherein the derivation way of IBC-LIC parameters for the first component is different from the derivation way of IBC-LIC parameters for the second component.

10 [0243] Clause 104. The method of clause 1, wherein a way to derive IBC-LIC parameters depends on at least one of colour component or colour format.

[0244] Clause 105. The method of clause 104, wherein the number of samples used to derive the IBC-LIC parameters for chroma components depends on colour format.

15 [0245] Clause 106. The method of clause 105, wherein the number of samples used chroma components is same as luma component for 4:4:4 colour format.

[0246] Clause 107. The method of clause 105, wherein the number of samples used chroma components is different from luma component for 4:4:4 colour format.

[0247] Clause 108. The method of clause 105, wherein the number of samples used chroma components is different from luma component for 4:2:0/4:2:2 colour format.

20 [0248] Clause 109. The method of clause 105, wherein a minimum number of samples used to derive the IBC-LIC parameters for chroma components is same as luma component regardless of colour format.

[0249] Clause 110. The method of clause 104, wherein at least one parameter is not derived for a second component but inherited from a first component.

25 [0250] Clause 111. The method of clause 110, wherein the first component is Y component, and the second component is a chroma component.

[0251] Clause 112. The method of clause 104, wherein IBC-LIC parameters are derived individually for different components, or wherein at least one IBC-LIC parameter is shared by different components.

[0252] Clause 113. The method of clause 112, wherein the components comprise luma and chroma components.

[0253] Clause 114. The method of clause 112, wherein the components comprise chroma components.

5 [0254] Clause 115. The method of clause 104, wherein if a luma BV is used for chroma components, the luma BV is modified depending on the colour format.

[0255] Clause 116. The method of clause 115, wherein $BV_C = BV_L (x \gg (\text{SubWidthC} - 1), y \gg (\text{SubHeightC} - 1))$, or wherein $BV_C = BV_L ((x + 1) \gg (\text{SubWidthC} - 1), (y + 1) \gg (\text{SubHeightC} - 1))$, or wherein
 10 $BV_C = BV_L ((x - 1) \gg (\text{SubWidthC} - 1), (y - 1) \gg (\text{SubHeightC} - 1))$, or wherein $BV_C = BV_L ((x + 1) \gg (\text{SubWidthC} - 1), (y - 1) \gg (\text{SubHeightC} - 1))$, or wherein $BV_C = BV_L ((x - 1) \gg (\text{SubWidthC} - 1), (y + 1) \gg (\text{SubHeightC} - 1))$, and wherein BV_L represents the luma BV, BV_C represents a chroma BV, SubWidthC represents a subblock width, SubHeightC represents a subblock height, x represents a coordinate in
 15 horizontal direction, y represents a coordinate in a vertical direction.

[0256] Clause 117. The method of clause 116, wherein BVC is used to derive a prediction of a chroma video unit.

[0257] Clause 118. The method of clause 116, wherein BVC is used to derive a reference template used to derive IBC-LIC parameters for a chroma video unit.

20 [0258] Clause 119. The method of clause 118, wherein if BVC is a fractional BV, BVC is rounded to an integer BV.

[0259] Clause 120. The method of clause 119, wherein a clipping operation is used during rounding the fractional BV.

[0260] Clause 121. The method of clause 119, wherein the fractional BV is used to
 25 derive the reference template.

[0261] Clause 122. The method of clause 121, wherein an interpolation method used to derive the reference template is same as an interpolation method used to derive the prediction of the chroma video unit.

[0262] Clause 123. The method of clause 121, wherein an interpolation method used to
 30 derive the reference template is different as an interpolation method used to derive the

prediction of the chroma video unit.

[0263] Clause 124. The method of clause 123, wherein the interpolation method used to derive the reference template and the interpolation method used to derive the prediction of the video unit have different interpolation taps.

- 5 [0264] Clause 125. The method of clause 124, wherein 2-tap interpolation filter is used to derive the reference template.

[0265] Clause 126. The method of any of clauses 1-125, wherein whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using a model of the IBC-LIC depends on video characteristics.

- 10 [0266] Clause 127. The method of clause 126, wherein the video characteristics comprises at least one of: screen content, camera-captured content, natural content, or mixed content.

[0267] Clause 128. The method of clause 127, wherein one or more syntax elements are used to indicate whether the video unit belongs to which content.

- 15 [0268] Clause 129. The method of clause 127, wherein which content the video unit belongs to is determined adaptively.

- [0269] Clause 130. The method of clause 129, wherein which content the video unit belongs to is determined using at least one of: neighboring reconstructed samples of the video unit, neighboring original samples, prediction samples of the video unit, original
20 samples of the video unit, hash value, or one or more neighboring video units are coded with a mode.

[0270] Clause 131. The method of any of clauses 1-130, wherein the video unit is applied with other coding tools.

- [0271] Clause 132. The method of clause 131, wherein the other coding tool comprises
25 an inter prediction, or wherein the other coding tool comprises an intra prediction, or wherein the other coding tool comprises an in-loop filter.

[0272] Clause 133. The method of clause 132, wherein the inter prediction is A local illumination compensation (LIC), OR wherein the intra prediction comprises IntraTMP.

- [0273] Clause 134. The method of any of clauses 1-133, wherein the video unit
30 comprises at least one of: a color component, a prediction block (PB), a transform block

(TB), a coding block (CB), a prediction unit (PU), a transform unit (TU), a coding tree block (CTB), a coding unit (CU), a coding tree unit (CTU), a CTU row, groups of CTU, a slice, a tile, a sub-picture, a block, a sub-region within a block, or a region containing more than one sample or pixel.

5 **[0274]** Clause 135. The method of any of clauses 1-134, wherein an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated at one of the followings: sequence level, group of pictures level, picture level, slice level, or tile group level.

10 **[0275]** Clause 136. The method of any of clauses 1-134, wherein an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated in one of the following: a sequence header, a picture header, a sequence parameter set (SPS), a video parameter set (VPS), a dependency parameter set (DPS), a decoding capability
15 information (DCI), a picture parameter set (PPS), an adaptation parameter sets (APS), a slice header, or a tile group header.

20 **[0276]** Clause 137. The method of any of clauses 1-134, wherein an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated at one of the following: a PB, a TB, a CB, a PU, a TU, a CU, a VPDU, a CTU, a CTU row, a slice, a tile, a sub-picture, or a region contains more than one sample or pixel.

25 **[0277]** Clause 138. The method of any of clauses 1-134, wherein whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is coded information of the video unit, and wherein the coded information comprises at least one of a block size, a colour format, a single tree partitioning, a dual tree partitioning, a colour component, a slice type, or a picture type.

30 **[0278]** Clause 139. The method of any of clauses 1-138, wherein the conversion includes encoding the video unit into the bitstream.

30 **[0279]** Clause 140. The method of any of clauses 1-138, wherein the conversion includes decoding the video unit from the bitstream.

30 **[0280]** Clause 141. An apparatus for video processing comprising a processor and a

non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform a method in accordance with any of clauses 1-140.

[0281] Clause 142. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of clauses 1-140.

[0282] Clause 143. 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: determining whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to a video unit of the video; in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; and generating the bitstream based on the compensated prediction/reconstruction of the video unit.

[0283] Clause 144. A method for storing a bitstream of a video, comprising: determining whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to a video unit of the video; in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; generating the bitstream based on the compensated prediction/reconstruction of the video unit; and storing the bitstream in a non-transitory computer-readable recording medium.

Example Device

[0284] Fig. 21 illustrates a block diagram of a computing device 2100 in which various embodiments of the present disclosure can be implemented. The computing device 2100 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).

[0285] It would be appreciated that the computing device 2100 shown in Fig. 21 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.

[0286] As shown in Fig. 21, the computing device 2100 includes a general-purpose computing device 2100. The computing device 2100 may at least comprise one or more

processors or processing units 2110, a memory 2120, a storage unit 2130, one or more communication units 2140, one or more input devices 2150, and one or more output devices 2160.

[0287] In some embodiments, the computing device 2100 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 2100 can support any type of interface to a user (such as “wearable” circuitry and the like).

[0288] The processing unit 2110 may be a physical or virtual processor and can implement various processes based on programs stored in the memory 2120. 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 2100. The processing unit 2110 may also be referred to as a central processing unit (CPU), a microprocessor, a controller or a microcontroller.

[0289] The computing device 2100 typically includes various computer storage medium. Such medium can be any medium accessible by the computing device 2100, including, but not limited to, volatile and non-volatile medium, or detachable and non-detachable medium. The memory 2120 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 2130 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 2100.

[0290] The computing device 2100 may further include additional detachable/non-detachable, volatile/non-volatile memory medium. Although not shown in Fig. 21, 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.

[0291] The communication unit 2140 communicates with a further computing device via the communication medium. In addition, the functions of the components in the computing device 2100 can be implemented by a single computing cluster or multiple computing machines that can communicate via communication connections. Therefore, the computing device 2100 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.

[0292] The input device 2150 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 2160 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 2140, the computing device 2100 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 2100, or any devices (such as a network card, a modem and the like) enabling the computing device 2100 to communicate with one or more other computing devices, if required. Such communication can be performed via input/output (I/O) interfaces (not shown).

[0293] In some embodiments, instead of being integrated in a single device, some or all components of the computing device 2100 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.

10 [0294] The computing device 2100 may be used to implement video encoding/decoding in embodiments of the present disclosure. The memory 2120 may include one or more video coding modules 2125 having one or more program instructions. These modules are accessible and executable by the processing unit 2110 to perform the functionalities of the various embodiments described herein.

15 [0295] In the example embodiments of performing video encoding, the input device 2150 may receive video data as an input 2170 to be encoded. The video data may be processed, for example, by the video coding module 2125, to generate an encoded bitstream. The encoded bitstream may be provided via the output device 2160 as an output 2180.

20 [0296] In the example embodiments of performing video decoding, the input device 2150 may receive an encoded bitstream as the input 2170. The encoded bitstream may be processed, for example, by the video coding module 2125, to generate decoded video data. The decoded video data may be provided via the output device 2160 as the output 2180.

25 [0297] 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 of video processing, comprising:

determining, for a conversion between a video unit of a video and a bitstream of the video unit, whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to the video unit;

in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; and

performing the conversion based on the compensated prediction/reconstruction of the video unit.

2. The method of claim 1, wherein one or more offsets are used to modify one or more parameters of the model.

3. The method of claim of claim 2, wherein the one or more offsets are indicated, or wherein the one or more offsets are derived, or wherein the one or more offsets are predefined.

4. The method of claim 2, wherein whether to and/or a way to use the one or more offsets to modify the one or more parameters is indicated, or wherein whether to and/or a way to use the one or more offsets to modify the one or more parameters is predefined, or wherein whether to and/or a way to use the one or more offsets to modify the one or more parameters is determined based on coding information.

5. The method of claim 1, wherein the model is presented as:

$$c_0 \times x_0 + c_1 \times x_1 + c_2 \times x_2 + \dots + c_N \times x_N + B, \text{ and}$$

wherein c_i represents a parameter and x_i represents coding information, i is an integer number and in a range from 0 to N , N is a predefined number, B represents a bias parameter which is derived together with c_i .

6. The method of claim 5, wherein the coding information comprises at least one of: reconstruction samples,

reference samples,
prediction samples,
values calculated using the reconstruction samples,
values calculated using the reference samples,
5 values calculated using the prediction samples,
position information of the reconstruction samples,
position information of the reference samples, or
position information of the prediction samples.

10 7. The method of claim 1, wherein a current template comprising neighboring reconstructed samples of the video unit and a reference template are used to derive parameters of the model, and the reference template is derived using a block vector (BV) that is used to obtain the prediction of the video unit.

15 8. The method of claim 7, wherein if the BV is a fractional BV, the fractional BV is rounded to an integer BV.

9. The method of claim 8, wherein a clipping operation is used during rounding the fractional BV.

20

10. The method of claim 7, wherein if the BV is a fractional BV, the fractional BV is used to derive the reference template.

25 11. The method of claim 10, wherein an interpolation method used to derive the reference template is same as an interpolation method used to derive the prediction of the video unit.

30 12. The method of claim 10, wherein an interpolation method used to derive the reference template is different as an interpolation method used to derive the prediction of the video unit.

13. The method of claim 12, wherein the interpolation method used to derive the reference template and the interpolation method used to derive the prediction of the video unit have different interpolation taps.

14. The method of claim 12, wherein 2-tap interpolation filter is used to derive the reference template.

5 15. The method of claim 1, wherein a way solving a model used for another coding tool is used to derive parameters of the model.

10 16. The method of claim 15, wherein the way is Cholesky decomposition used in adaptive loop filtering (ALF) and a variant of Cholesky decomposition.

17. The method of claim 15, wherein the way is LDL decomposition used in convolutional cross-component model (CCCM) and a variant of LDL decomposition.

15 18. The method of claim 15, wherein the way is Gaussian elimination used in CCCM and a variant of Gaussian elimination.

19. The method of claim 1, wherein a mean square error (MSE) minimization is used as a metric to solve the model.

20 20. The method of claim 19, wherein one or more other metrics used to solve the model.

21. The method of claim 20, wherein the one or more other metrics comprises at least one of: sum of squared error (SSE) or a mean of SSE.

25 22. The method of claim 19, wherein a subjective metric is used to solve the model.

23. The method of claim 22, wherein the subjective metric comprises a structure similarity index measure (SSIM).

30 24. The method of claim 1, wherein more than one way is used to derive parameters for IBC-LIC.

25. The method of claim 24, wherein which way is used to derive the parameters is indicated, or

wherein which way is used to derive the parameters is predefined, or
wherein which way is used to derive the parameters is determined based coding information.

5 26. The method of claim 1, wherein more than one way is used to derive more than one set of parameters, and predictions of the current video unit modified using the more than one set of parameters are fused.

10 27. The method of claim 26, wherein weights used for the fusion are predefined, or wherein the weights used for the fusion are indicated, or wherein the weights used for the fusion are determined based on coding information.

15 28. The method of claim 1, wherein the IBC-LIC is used in a process of reordering a BV candidate list.

 29. The method of claim 28, wherein two-pass reordering is used to reorder the BV candidate list.

20 30. The method of claim 29, wherein the IBC-LIC is applied in at least one of: a first pass reordering, or a second pass reordering.

 31. The method of claim 28, wherein the IBC-LIC is applied to partial BV candidates in the BV candidate list.

25 32. The method of any of claims 1-31, wherein whether to and/or an approach to apply the IBC-LIC depends on coding information of the video unit, and wherein the coding information of the video unit comprises at least one of:

30 a block dimension,
 a block size,
 a depth of the video unit,
 a slice type,
 a picture type,
 a partition tree type,
 a block location,

a quantization parameter, or
a colour component.

33. The method of claim 32, wherein a block is allowed to be coded with the IBC-LIC,
5 if a block size is larger than or equal to a first threshold, wherein the block size is equal to $W \times H$,
and W and H denotes block width and block height of the block, respectively.

34. The method of claim 33, wherein the second threshold is equal to one of: 4, 8, 16,
32, 64, 128, or 256.

35. The method of claim 32, wherein a block is allowed to be coded with the IBC-LIC,
if a block size is less than or equal to a second threshold, wherein the block size is equal to
 $W \times H$, and W and H denotes block width and block height of the block, respectively.

36. The method of claim 35, wherein the second threshold is equal to one of: 4, 8, 16,
32, 64, 128, or 256.

37. The method of claim 35, wherein the second threshold depends on whether IBC
advanced motion vector prediction (AMVP) mode or IBC merge mode is used.

38. The method of claim 35, wherein the second threshold depends on slice type or
picture type.

39. The method of claim 32, wherein a block is allowed to be coded with the IBC-LIC,
25 if a block size is larger than or equal to a third threshold, wherein the block size is equal to
 $W \times H$, and W and H denotes block width and block height of the block, respectively.

40. The method of claim 39, wherein the third threshold is equal to one of: 4, 8, 16, 32,
64, 128, or 256.

41. The method of claim 39, wherein the third threshold depends on whether IBC AMVP
mode or IBC merge mode is used.

42. The method of claim 39, wherein the third threshold depends on slice type or picture type.

43. The method of any of claims 32-42, wherein the block size refers to chroma block size.

44. The method of claim 1, wherein more than one LIC equation is used to compensate the prediction or reconstruction of the video unit, and parameters of the more than one LIC equations are derived using different templates.

45. The method of claim 44, wherein at least one of the following templates is used to derive the parameters:

left template,
above template,
left-above template,
left-below template, or
above-right template.

46. The method of claim 45, wherein the templates are adjacent, and/or wherein the templates are non-adjacent.

47. The method of claim 1, wherein more than one way is used to derive parameters for IBC-LIC.

48. The method of claim 47, wherein which way is used to derive the parameters is indicated, or wherein which way is used to derive the parameters is predefined, or wherein which way is used to derive the parameters is determined based on coding information.

49. The method of claim 1, wherein more than one way and/or more than one template is used to derive more than one set of parameters of the model, and predictions of the video unit modified by the more than one set of parameters are fused.

50. The method of claim 49, wherein weights used for the fusion are pre-defined, or wherein the weights used for the fusion are indicated, or wherein the weights used for the fusion are determined based on coding information.

5 51. The method of claim 1, wherein one or more of: at least one parameter in the model or at least one IBC-LIC model used for video units coded before a current video unit are used for the current video unit, and/or

wherein one or more of: at least one parameter in the model or at least one IBC-LIC model used for video units decoded before a current video unit are used for the current video
10 unit.

52. The method of claim 51, wherein the video units coded or decoded before the current video unit are in one of:

15 a different slice,
a different tile,
a different picture,
a different coding tree unit (CTU), or
a different CTU row.

20 53. The method of claim 51, wherein the video units coded or decoded before the current video unit are in one of:

a same slice as the current video unit,
a same tile as the current video unit,
a same picture as the current video unit,
25 a same CTU as the current video unit, or
a same CTU row as the current video unit.

54. The method of claim 51, wherein the video units coded or decoded before the current video unit are neighbouring spatial video units.

30 55. The method of claim 54, wherein the neighbouring spatial video units are adjacent video units, and/or

wherein the neighbouring spatial video units are non-adjacent video units.

56. The method of claim 51, wherein reused parameters are stored in a list or a table.

57. The method of claim 56, wherein the table is a history-based motion vector prediction IBC-LIC parameter table.

5

58. The method of claim 56, wherein the list or table is updated during a coding process, and/or

wherein the list or table is updated during a decoding process.

10

59. The method of claim 56, wherein a maximum size of the list or table is predefined,

or

wherein the maximum size of the list or table is indicated, or

wherein the maximum size of the list or table is derived.

15

60. The method of claim 56, wherein the list or table is re-initialized at a beginning of one of: a slice, a tile, a picture, or a CTU.

61. The method of claim 60, wherein the list is re-initialized as a void list, or wherein the table is re-initialized as a void table.

20

62. The method of claim 60, wherein the list or table is re-initialized using one or more parameters.

63. The method of claim 62, wherein the one or more parameters are predefined, or wherein the one or more parameters are derived, or wherein the one or more parameters are indicated.

25

64. The method of claim 56, wherein whether to and/or how to use the list or table depends on coding information, or

30

wherein whether to and/or how to update the list or table depends on coding information.

65. The method of claim 64, wherein the coding information comprises at least one of: block dimensions, block size, or

block position.

66. The method of claim 51, wherein if the current video unit is chroma video unit, reused parameters are from at least one of luma video unit or chroma video unit.

5

67. The method of claim 1, wherein the IBC-LIC is applied to sub-blocks within the video unit which is a current video unit.

68. The method of claim 67, wherein whether to and/or how to apply the IBC-LIC to the sub-blocks is predefined, or

10

wherein whether to and/or how to apply the IBC-LIC to the sub-blocks is indicated, or wherein whether to and/or how to apply the IBC-LIC to the sub-blocks is determined.

69. The method of claim 68, wherein a syntax element is used to indicate whether the IBC-LIC is applied with sub-block level.

15

70. The method of claim 68, wherein whether to and/or how to apply the IBC-LIC to the sub-blocks depends on block dimensions or block size.

20

71. The method of claim 68, wherein the IBC-LIC is applied to all sub-blocks, or wherein the IBC-LIC is not applied to all sub-blocks.

72. The method of claim 68, wherein the IBC-LIC is applied to partial sub-blocks, and the IBC-LIC is not applied to other sub-blocks.

25

73. The method of claim 72, wherein the partial sub-blocks to which the IBC-LIC is applied are boundary sub-blocks.

74. The method of claim 67, wherein IBC-LIC parameters are different for at least one sub-block.

30

75. The method of claim 67, wherein IBC-LIC parameters are shared by at least one sub-block.

76. The method of claim 1, further comprising
determining whether the video unit is allowed to be coded with an IBC with LIC mode
based on coded information of the video unit.

5 77. The method of claim 76, wherein the coded information comprises at least one of:
a block dimension,
a block size,
a depth of a block,
a block location,
10 a slice type,
a picture type,
a temporal layer,
a colour format, or
a colour component.

15 78. The method of claim 77, wherein if the block size that is represented as $W \times H$ is less
than or equal to a fourth threshold, the video unit is allowed to be coded with the IBC with LIC
mode, wherein W represents a block width of the video unit and H represents a block height of
the video unit.

20 79. The method of claim 78, wherein the fourth threshold depends on whether IBC
AMVP mode or IBC merge mode is used, or
wherein the fourth threshold depends on slice type or picture.

25 80. The method of claim 77, wherein if the block size that is represented as $W \times H$ is
larger than or equal to a fifth threshold, the video unit is allowed to be coded with the IBC with
LIC mode, wherein W represents a block width of the video unit and H represents a block height
of the video unit.

30 81. The method of claim 80, wherein the fifth threshold is one of: 4, 8, 16, 32, 64, 128,
256, 512, 1024, or 2048.

82. The method of claim 80, wherein the fifth threshold depends on whether IBC AMVP
mode or IBC merge mode is used, or

wherein the fifth threshold depends on slice type or picture.

83. The method of claim 77, wherein if the block size that is represented as $W \times H$ is less than or equal to a sixth threshold, the video unit is allowed to be coded with the IBC with LIC mode, wherein W represents a block width of the video unit and H represents a block height of the video unit.

84. The method of claim 83, wherein the sixth threshold is one of: 4, 8, 16, 32, 64, 128, or 256.

85. The method of claim 83, wherein the sixth threshold depends on whether IBC AMVP mode or IBC merge mode is used, or wherein the sixth threshold depends on slice type or picture.

86. The method of claim 77, wherein if the block size that is represented as $W \times H$ is larger than or equal to a seventh threshold, the video unit is allowed to be coded with the IBC with LIC mode, wherein W represents a block width of the video unit and H represents a block height of the video unit.

87. The method of claim 86, wherein the seventh threshold is one of: 4, 8, 16, 32, 64, 128, or 256.

88. The method of claim 86, wherein the seventh threshold depends on whether IBC AMVP mode or IBC merge mode is used, or wherein the seventh threshold depends on slice type or picture.

89. The method of claim 77, wherein the block size is luma block size.

90. The method of claim 77, wherein the block size is chroma block size.

91. The method of claim 1, wherein a determination of whether to and/or how to apply the IBC-LIC to a first video unit in a first component depends on a second video unit in a second component.

92. The method of claim 91, wherein the second video unit is a collocated luma video unit of the first video unit.

93. The method of claim 92, wherein
 $P_L(x, y) = P_C(x * \text{SubWidthC}, y * \text{SubHeightC})$, and
 wherein P_C represents a position of the first video unit, P_L represents a position of the second video unit, SubWidthC represents a subblock width, SubHeightC represents a subblock height, x represents a coordinate in horizontal direction, y represents a coordinate in a vertical direction.

94. The method of claim 92, wherein $(P_C(x, y)) = (W/2 - 1, H/2 + 1)$, or
 wherein $(P_C(x, y)) = (W/2 + 1, H/2 - 1)$, or
 wherein $(P_C(x, y)) = (W/2 - 1, H/2 + 1)$, or
 wherein $(P_C(x, y)) = (W/2 + 1, H/2 + 1)$,
 wherein P_C represents a center position of a chroma video unit, W represents a width of the chroma video unit, H represents a height of the chroma video unit, x is in a range from 0 to $W - 1$, inclusive, and y is in a range of 0 and $H - 1$ inclusive.

95. The method of claim 92, wherein $(P_C(x, y))$ refers to one of:
 a left-top position of the chroma video unit,
 a right-top position of the chroma video unit,
 a left-bottom position of the chroma video unit, or
 a right-bottom position of the chroma video unit.

96. The method of claim 91, wherein the second video unit is a collocated luma video unit of the first video unit.

97. The method of claim 91, wherein the second video unit is a spatial neighbouring luma video unit of the collocated luma video unit.

98. The method of claim 96, wherein the spatial neighbouring luma video unit is an adjacent neighbouring luma video unit, or
 wherein the spatial neighbouring luma video unit is a non-adjacent neighbouring luma video unit.

99. The method of claim 96, wherein the spatial neighbouring luma video unit comprises one of:

5 a left neighbouring luma video unit,
a left-below neighbouring luma video unit,
a left-above neighbouring luma video unit,
an above neighbouring luma video unit, or
an above-right neighbouring luma video unit.

10 100. The method of claim 96, wherein the spatial neighbouring luma video unit of the collocated luma video unit is used, if the collocated luma video unit is not coded with IBC mode or intra template matching prediction (IntraTMP) mode.

15 101. The method of claim 91, wherein the IBC-LIC is applied to the first video unit, if the second video unit is coded with IBC mode or IntraTMP mode.

102. The method of claim 1, wherein a derivation way of IBC-LIC parameters for a first component is same as a derivation way of IBC-LIC parameters for a second component.

20 103. The method of claim 1, wherein the derivation way of IBC-LIC parameters for the first component is different from the derivation way of IBC-LIC parameters for the second component.

25 104. The method of claim 1, wherein a way to derive IBC-LIC parameters depends on at least one of colour component or colour format.

105. The method of claim 104, wherein the number of samples used to derive the IBC-LIC parameters for chroma components depends on colour format.

30 106. The method of claim 105, wherein the number of samples used chroma components is same as luma component for 4:4:4 colour format.

107. The method of claim 105, wherein the number of samples used chroma components is different from luma component for 4:4:4 colour format.

108. The method of claim 105, wherein the number of samples used chroma components is different from luma component for 4:2:0/4:2:2 colour format.

109. The method of claim 105, wherein a minimum number of samples used to derive the IBC-LIC parameters for chroma components is same as luma component regardless of colour format.

110. The method of claim 104, wherein at least one parameter is not derived for a second component but inherited from a first component.

111. The method of claim 110, wherein the first component is Y component, and the second component is a chroma component.

112. The method of claim 104, wherein IBC-LIC parameters are derived individually for different components, or
wherein at least one IBC-LIC parameter is shared by different components.

113. The method of claim 112, wherein the components comprise luma and chroma components.

114. The method of claim 112, wherein the components comprise chroma components.

115. The method of claim 104, wherein if a luma BV is used for chroma components, the luma BV is modified depending on the colour format.

116. The method of claim 115, wherein $BV_C = BV_L (x \gg (\text{SubWidthC} - 1), y \gg (\text{SubHeightC} - 1))$, or

wherein $BV_C = BV_L ((x + 1) \gg (\text{SubWidthC} - 1), (y + 1) \gg (\text{SubHeightC} - 1))$,

or

wherein $BV_C = BV_L ((x - 1) \gg (\text{SubWidthC} - 1), (y - 1) \gg (\text{SubHeightC} - 1))$,

or

wherein $BV_C = BV_L ((x + 1) \gg (\text{SubWidthC} - 1), (y - 1) \gg (\text{SubHeightC} - 1))$,

or

wherein $BV_C = BV_L \left((x - 1) \gg (\text{SubWidthC} - 1), (y + 1) \gg (\text{SubHeightC} - 1) \right)$,
and

wherein BV_L represents the luma BV, BV_C represents a chroma BV, SubWidthC represents a subblock width, SubHeightC represents a subblock height, x represents a
5 coordinate in horizontal direction, y represents a coordinate in a vertical direction.

117. The method of claim 116, wherein BV_C is used to derive a prediction of a chroma video unit.

10 118. The method of claim 116, wherein BV_C is used to derive a reference template used to derive IBC-LIC parameters for a chroma video unit.

119. The method of claim 118, wherein if BV_C is a fractional BV, BV_C is rounded to an integer BV.
15

120. The method of claim 119, wherein a clipping operation is used during rounding the fractional BV.

121. The method of claim 119, wherein the fractional BV is used to derive the reference
20 template.

122. The method of claim 121, wherein an interpolation method used to derive the reference template is same as an interpolation method used to derive the prediction of the chroma video unit.
25

123. The method of claim 121, wherein an interpolation method used to derive the reference template is different as an interpolation method used to derive the prediction of the chroma video unit.

30 124. The method of claim 123, wherein the interpolation method used to derive the reference template and the interpolation method used to derive the prediction of the video unit have different interpolation taps.

125. The method of claim 124, wherein 2-tap interpolation filter is used to derive the reference template.

126. The method of any of claims 1-125, wherein whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using a model of the IBC-LIC depends on video characteristics.

127. The method of claim 126, wherein the video characteristics comprises at least one of: screen content, camera-captured content, natural content, or mixed content.

128. The method of claim 127, wherein one or more syntax elements are used to indicate whether the video unit belongs to which content.

129. The method of claim 127, wherein which content the video unit belongs to is determined adaptively.

130. The method of claim 129, wherein which content the video unit belongs to is determined using at least one of:

neighboring reconstructed samples of the video unit,
neighboring original samples,
prediction samples of the video unit,
original samples of the video unit,
hash value, or
one or more neighboring video units are coded with a mode.

131. The method of any of claims 1-130, wherein the video unit is applied with other coding tools.

132. The method of claim 131, wherein the other coding tool comprises an inter prediction, or
wherein the other coding tool comprises an intra prediction, or
wherein the other coding tool comprises an in-loop filter.

133. The method of claim 132, wherein the inter prediction is A local illumination compensation (LIC), OR
wherein the intra prediction comprises IntraTMP.

134. The method of any of claims 1-133, wherein the video unit comprises at least one of:

a color component,
a prediction block (PB),
a transform block (TB),
a coding block (CB),
a prediction unit (PU),
a transform unit (TU),
a coding tree block (CTB),
a coding unit (CU),
a coding tree unit (CTU),
a CTU row,
groups of CTU,
a slice,
a tile,
a sub-picture,
a block,
a sub-region within a block, or
a region containing more than one sample or pixel.

135. The method of any of claims 1-134, wherein an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated at one of the followings:

sequence level,
group of pictures level,
picture level,
slice level, or
tile group level.

136. The method of any of claims 1-134, wherein an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated in one of the following:

- a sequence header,
- a picture header,
- a sequence parameter set (SPS),
- a video parameter set (VPS),
- a dependency parameter set (DPS),
- a decoding capability information (DCI),
- a picture parameter set (PPS),
- an adaptation parameter sets (APS),
- a slice header, or
- a tile group header.

137. The method of any of claims 1-134, wherein an indication of whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is indicated at one of the following:

- a PB,
- a TB,
- a CB,
- a PU,
- a TU,
- a CU,
- a VPDU,
- a CTU,
- a CTU row,
- a slice,
- a tile,
- a sub-picture, or
- a region contains more than one sample or pixel.

138. The method of any of claims 1-134, wherein whether to and/or how to perform the compensation process to the prediction/reconstruction of the video unit using the model of the IBC-LIC is coded information of the video unit, and

wherein the coded information comprises at least one of
a block size,
a colour format,
a single tree partitioning,
5 a dual tree partitioning,
a colour component,
a slice type, or
a picture type.

10 139. The method of any of claims 1-138, wherein the conversion includes encoding the video unit into the bitstream.

140. The method of any of claims 1-138, wherein the conversion includes decoding the video unit from the bitstream.

15 141. An apparatus for video processing comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform a method in accordance with any of claims 1-140.

20 142. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of claims 1-140.

25 143. 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:

determining whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to a video unit of the video;

30 in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC; and

generating the bitstream based on the compensated prediction/reconstruction of the video unit.

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

determining whether an intra block copy (IBC) and local illumination compensation (LIC) (IBC-LIC) is applied to a video unit of the video;

5 in accordance with a determination that the IBC-LIC is applied to the video unit, performing a compensation process to a prediction/reconstruction of the video unit using a model of the IBC-LIC;

generating the bitstream based on the compensated prediction/reconstruction of the video unit; and

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

DRAWINGS

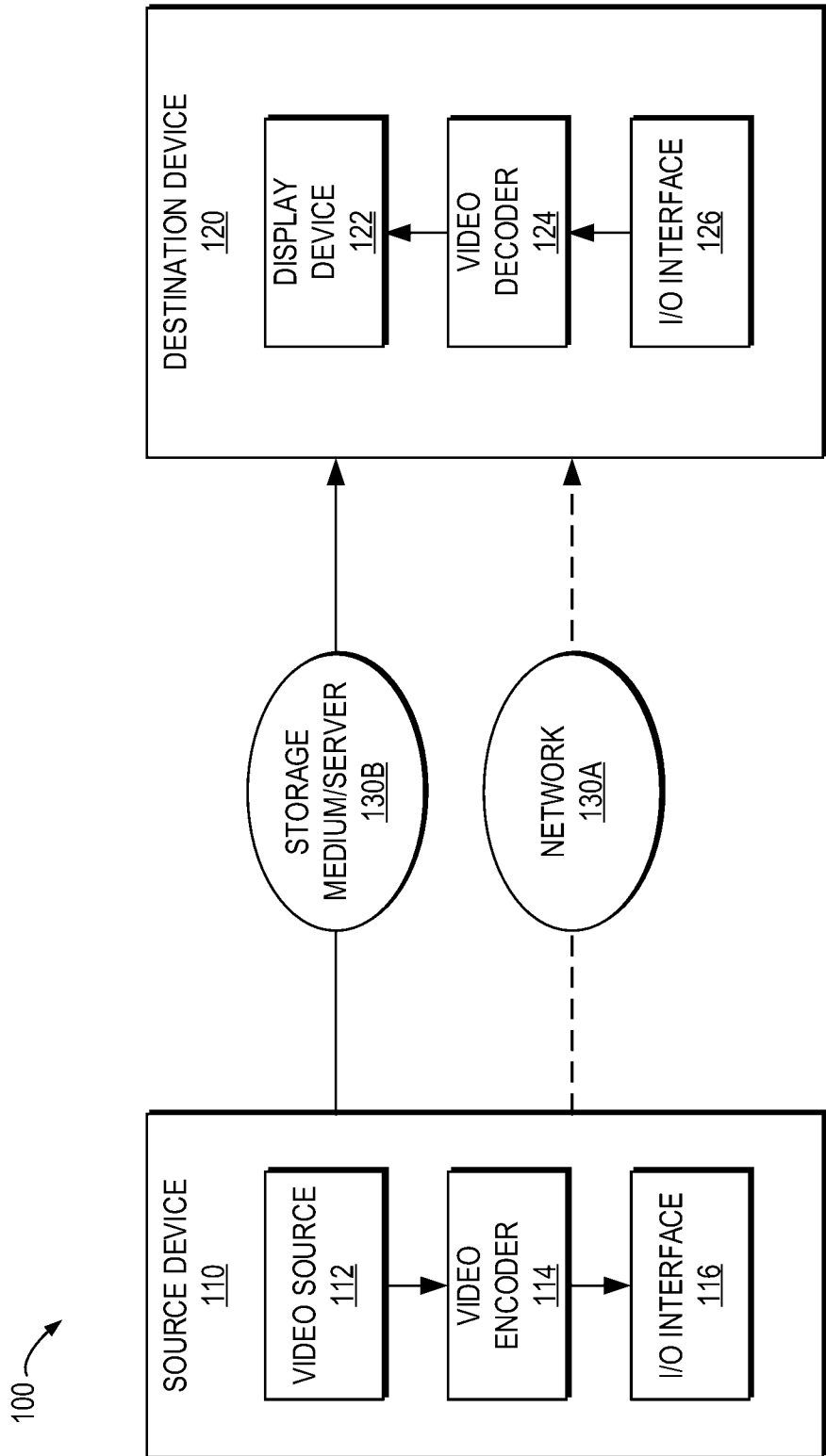


Fig. 1

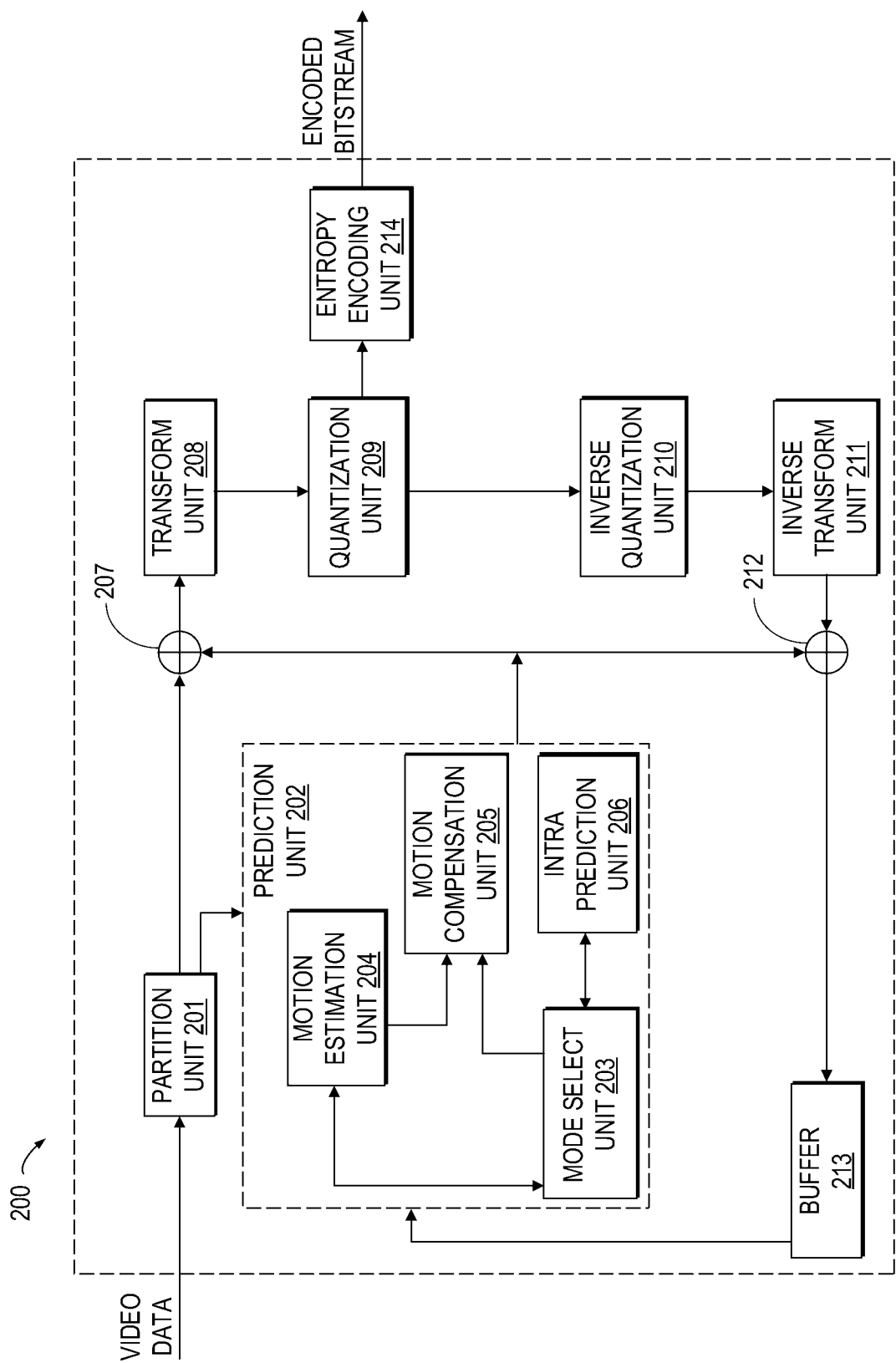


Fig. 2

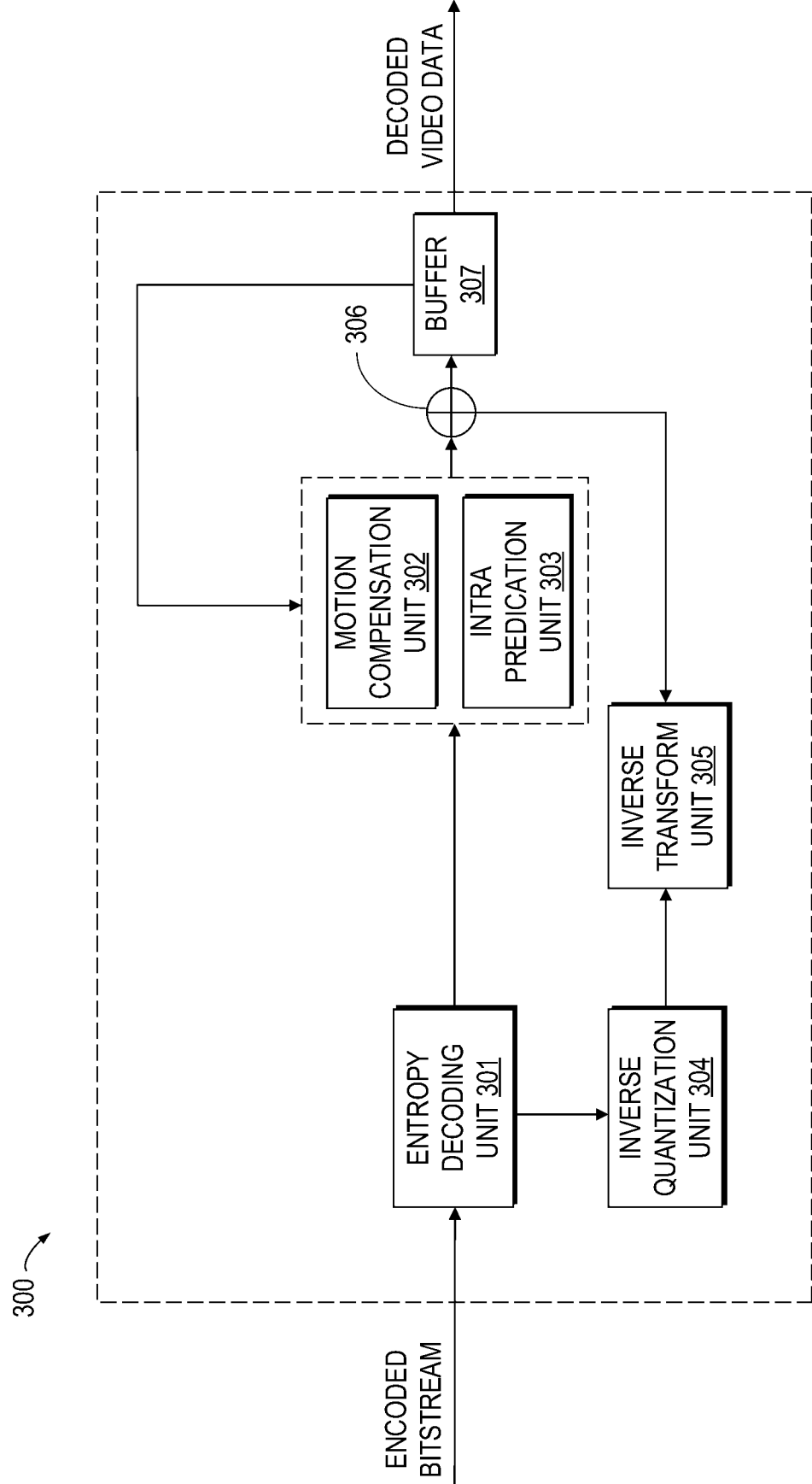


Fig. 3

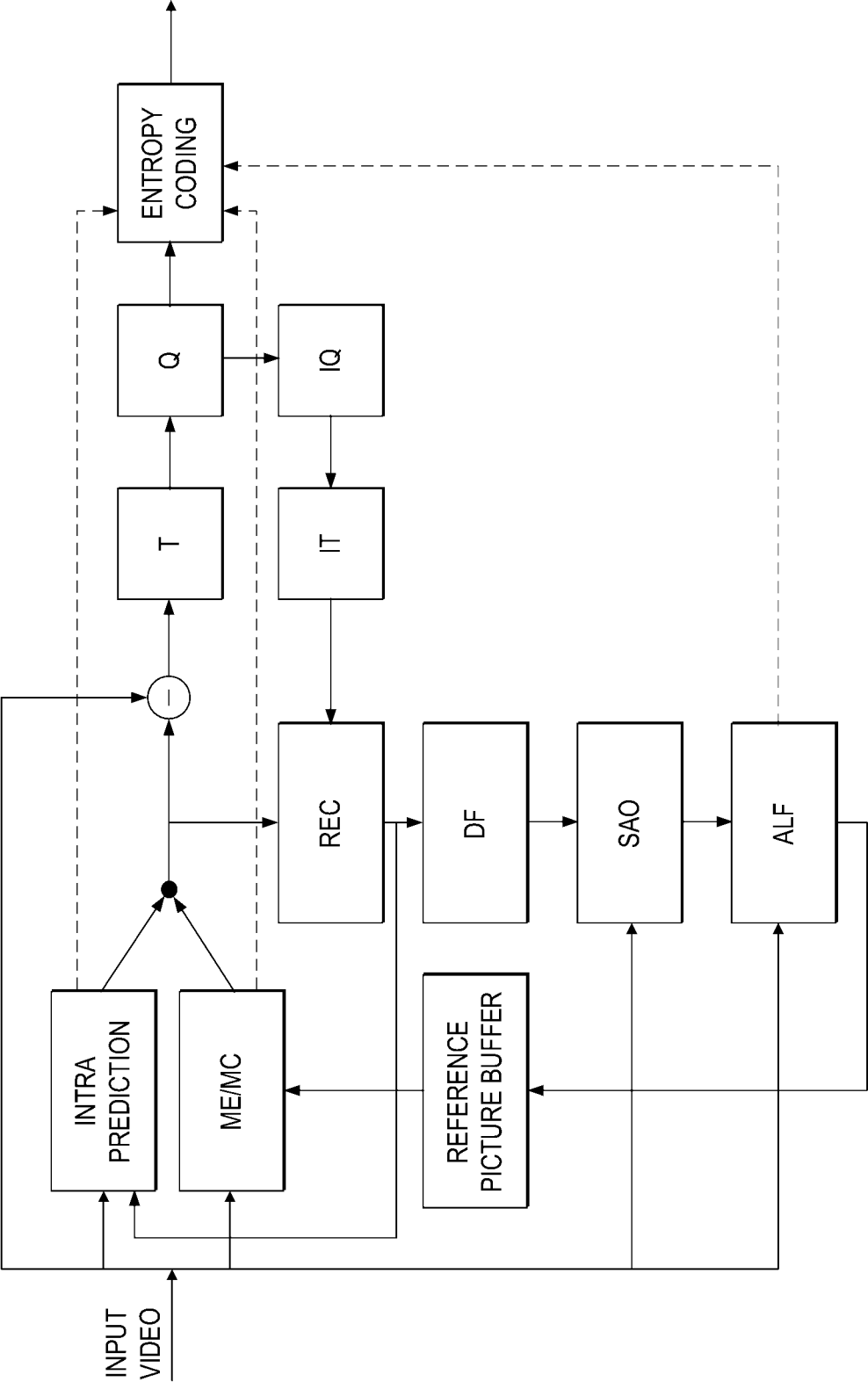


Fig. 4

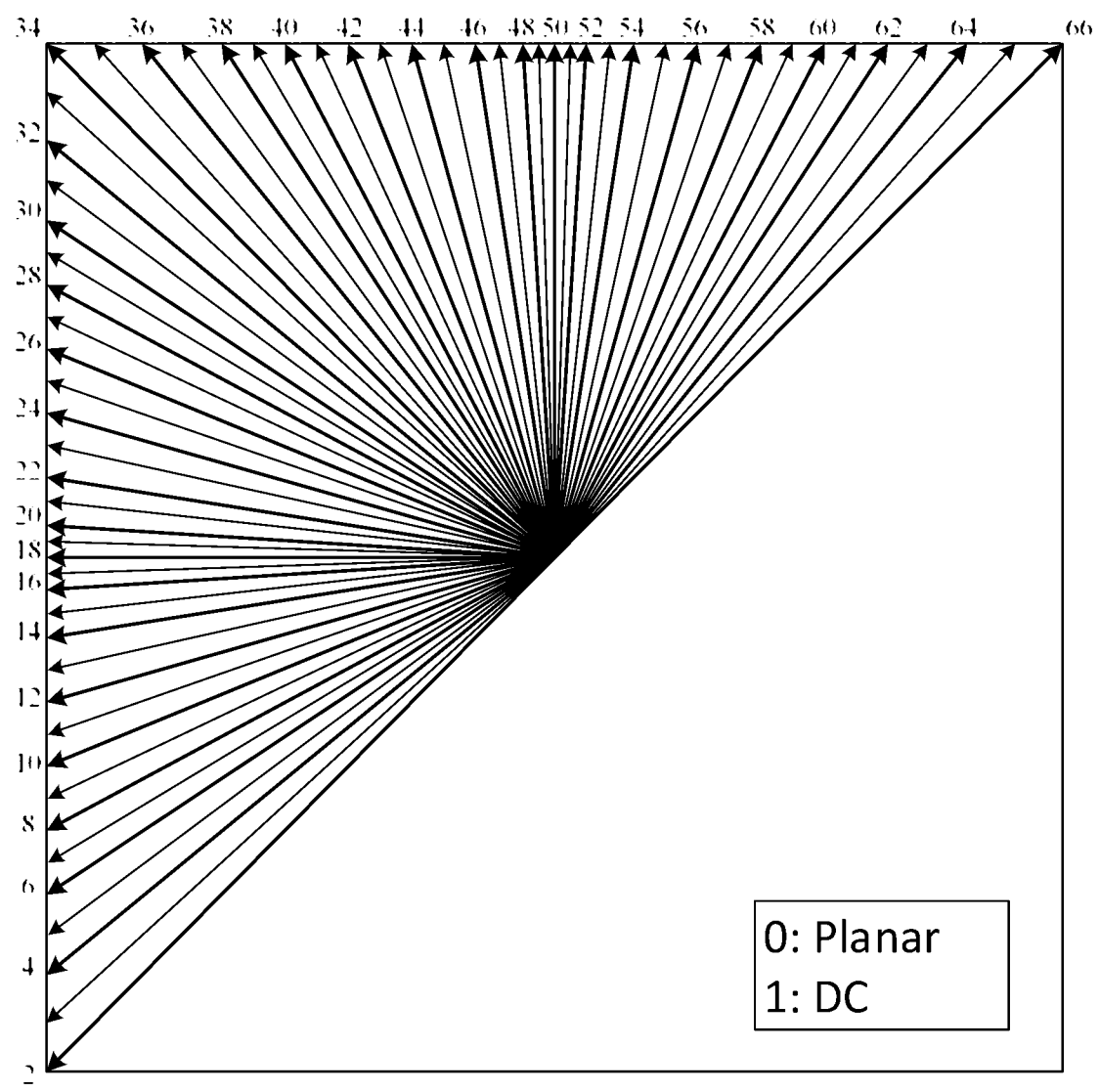


Fig. 5

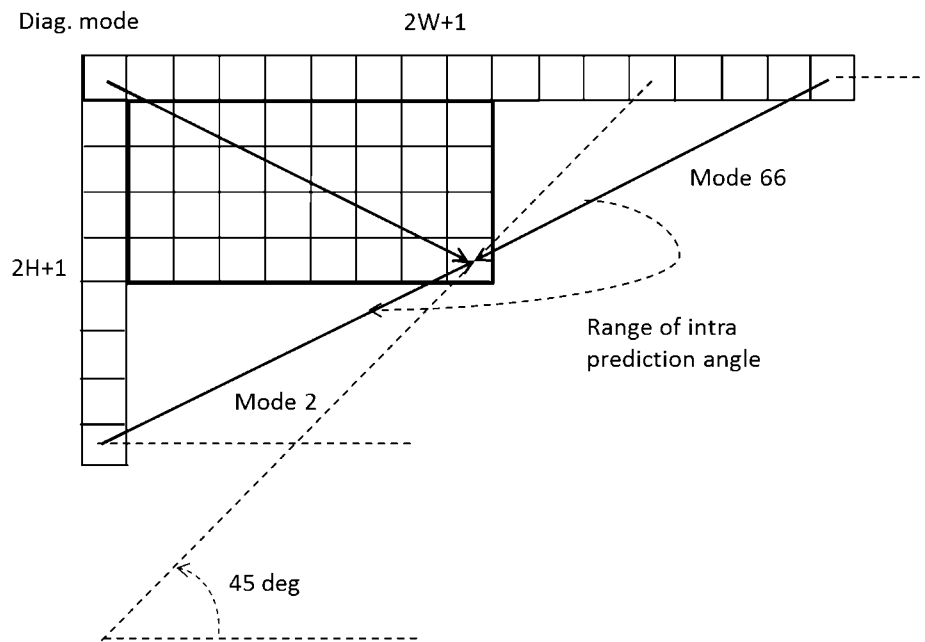


Fig. 6A

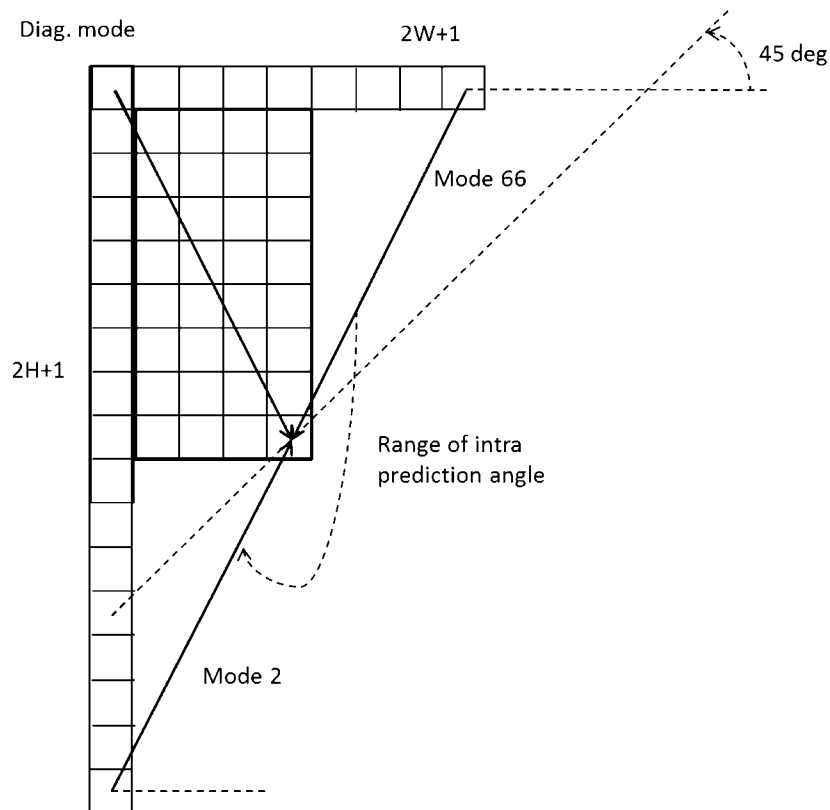


Fig. 6B

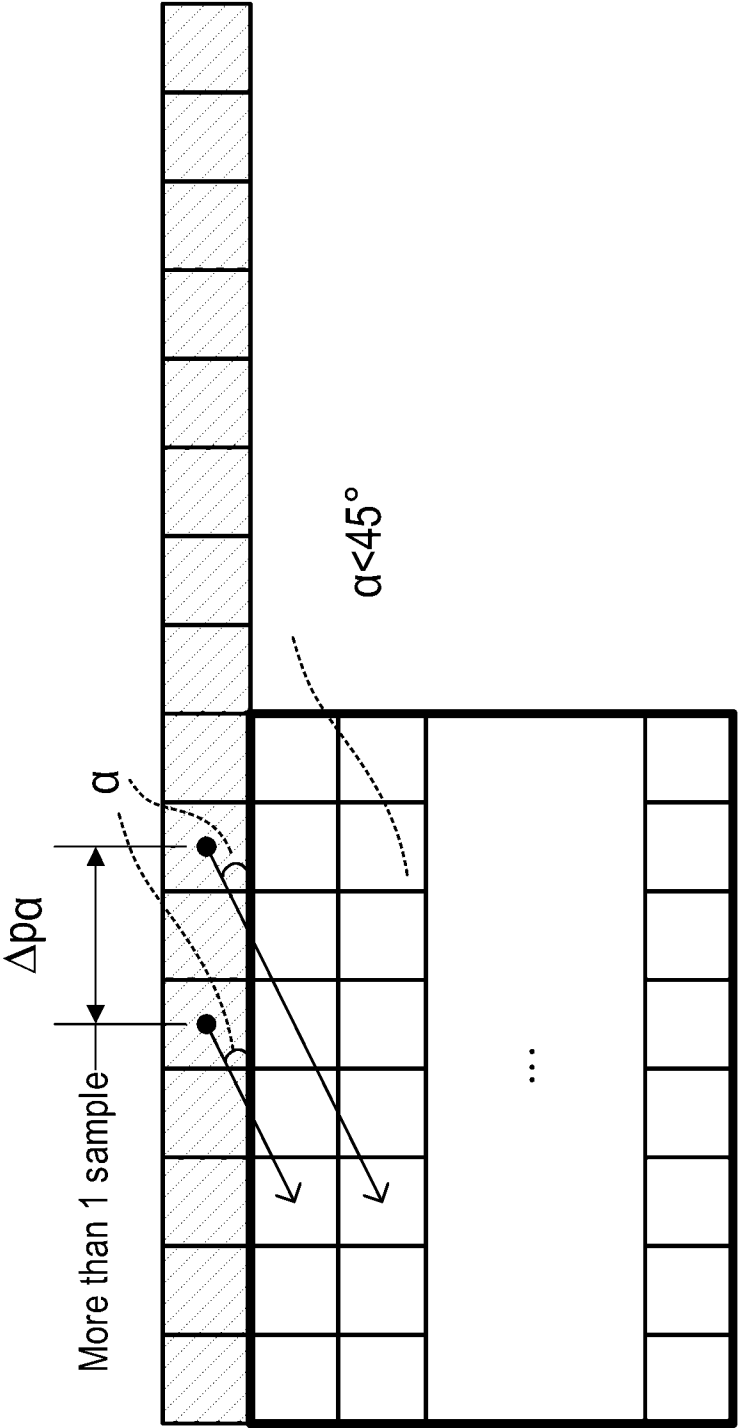


Fig. 7

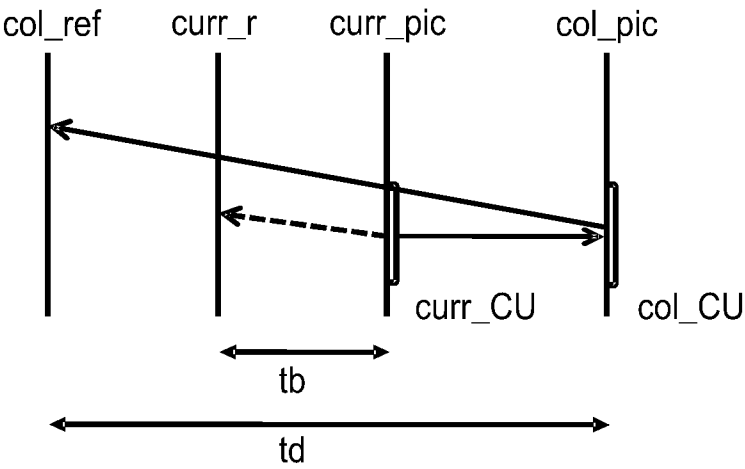


Fig. 8

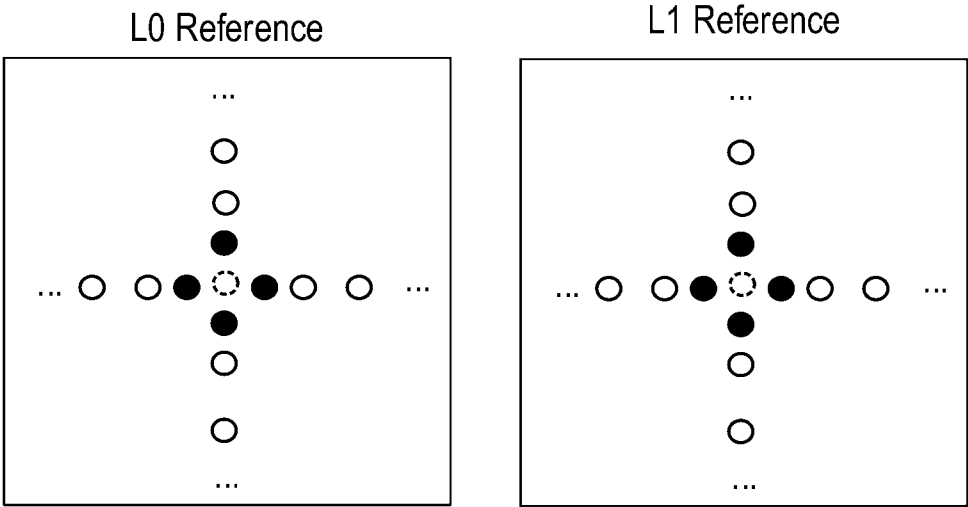


Fig. 9A

Fig. 9B

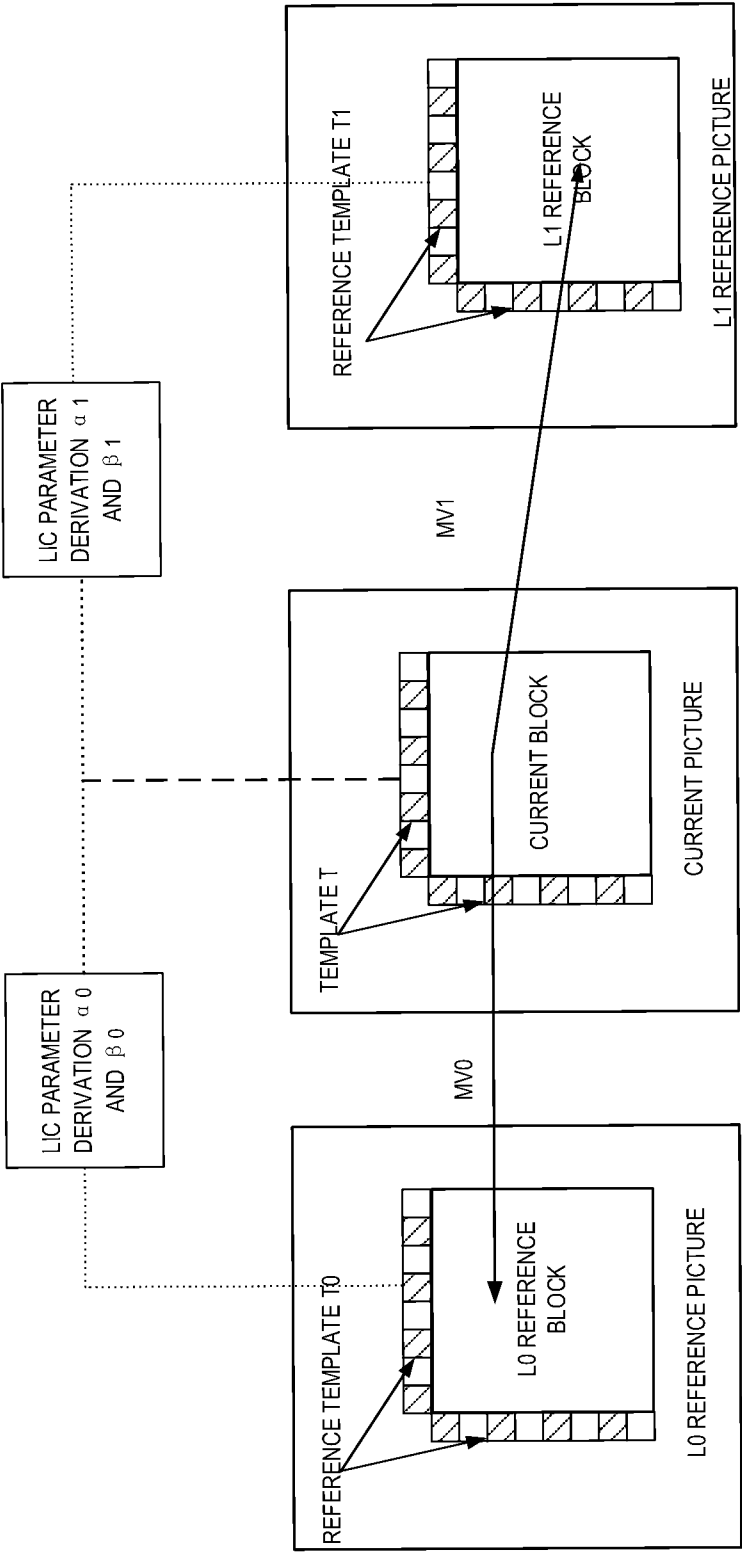


Fig. 10

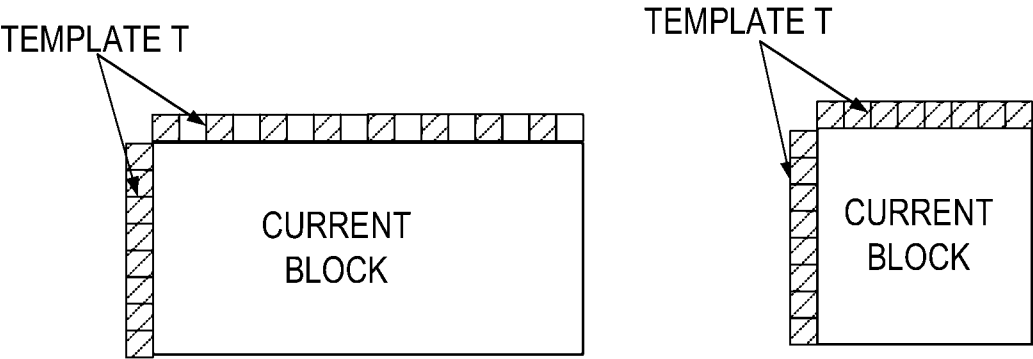


Fig. 11

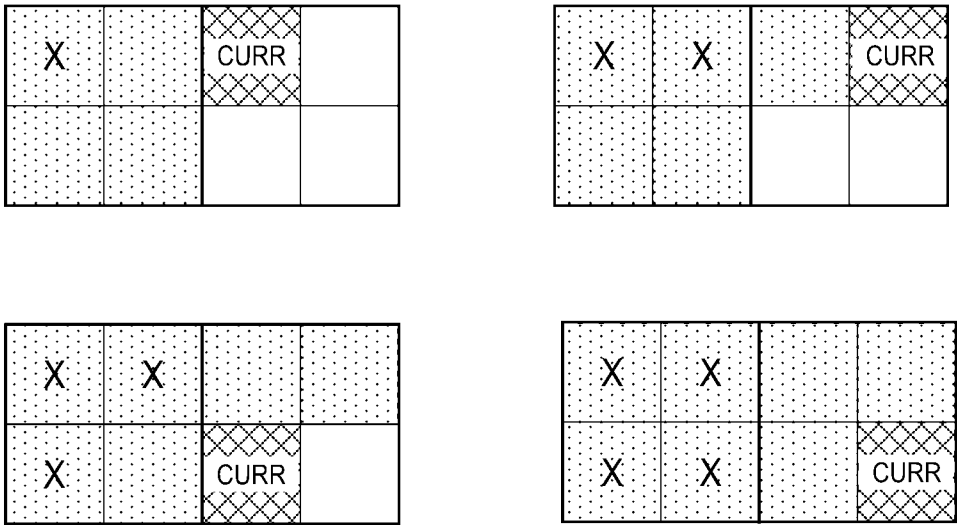


Fig. 12

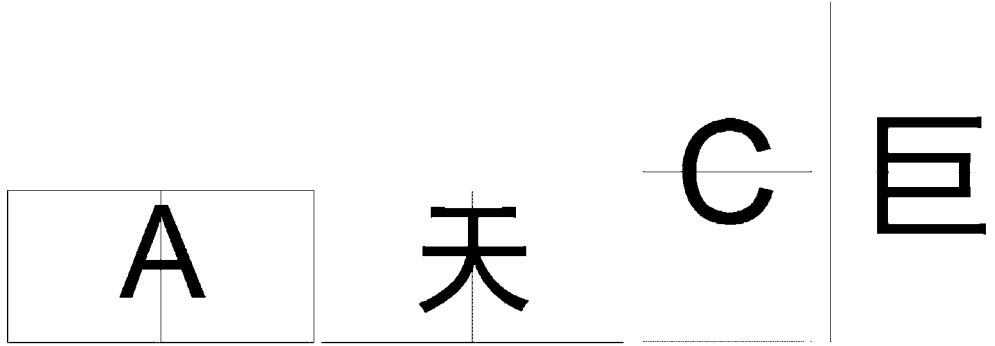


Fig. 13

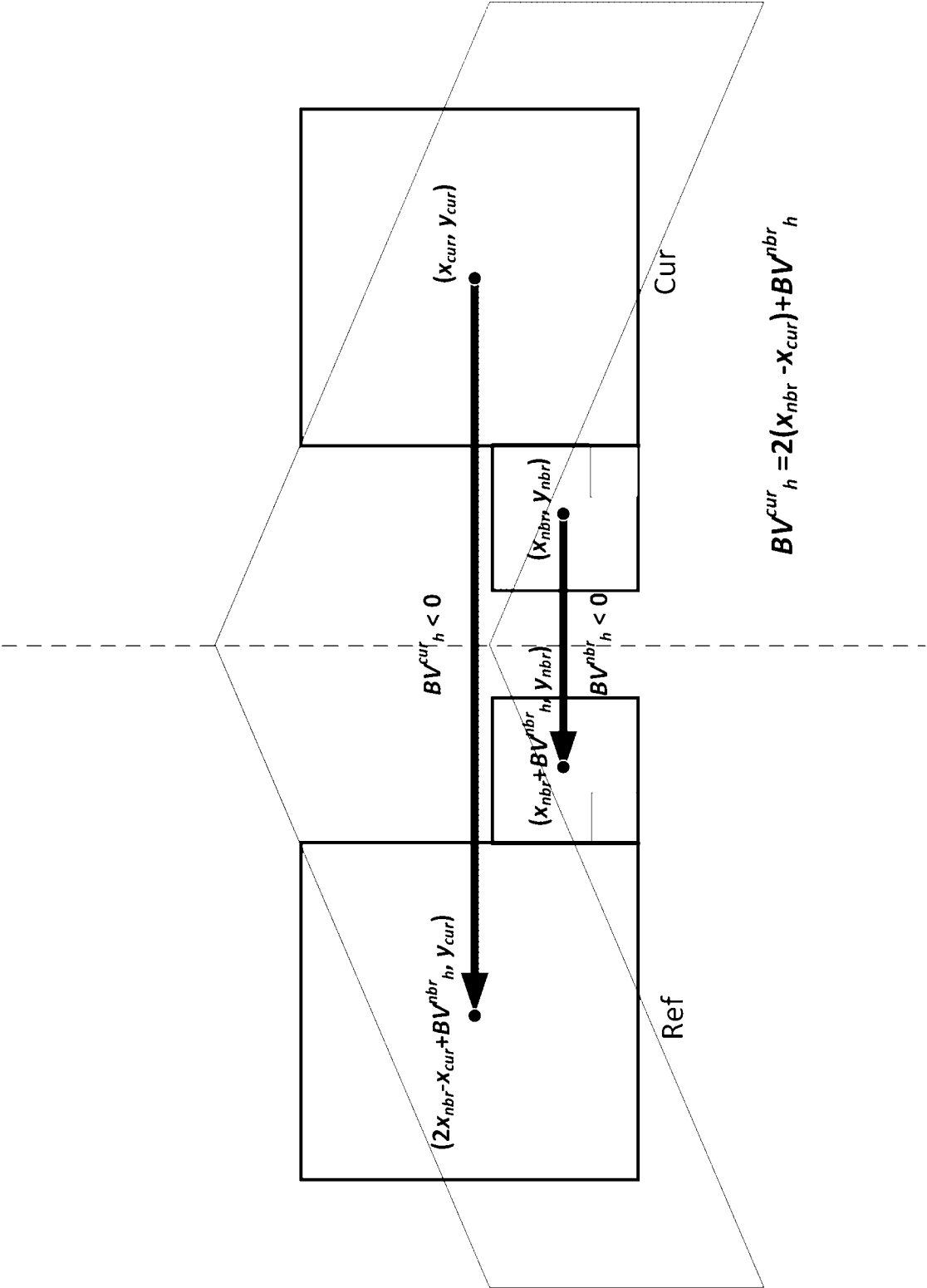


Fig. 14A

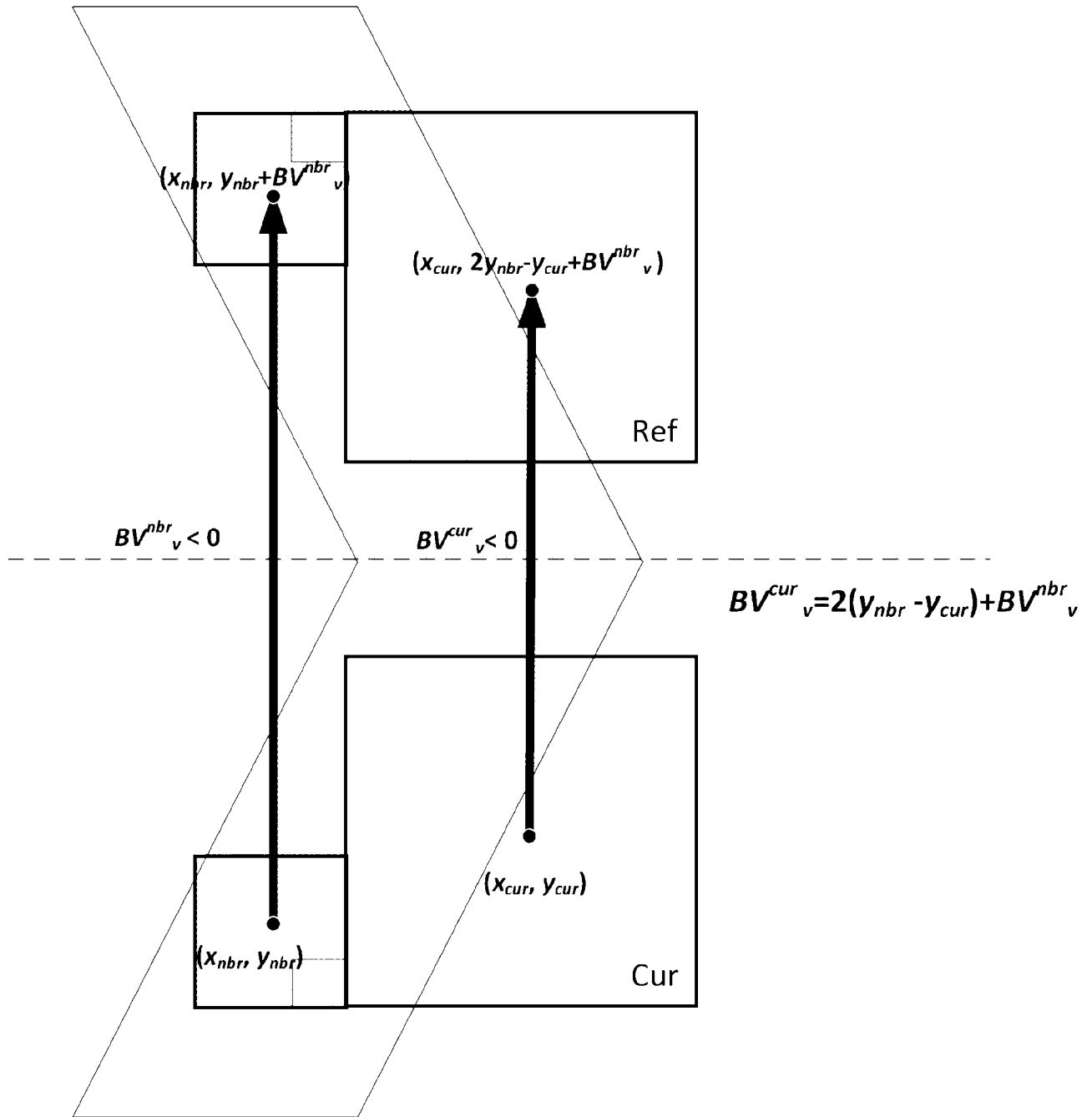


Fig. 14B

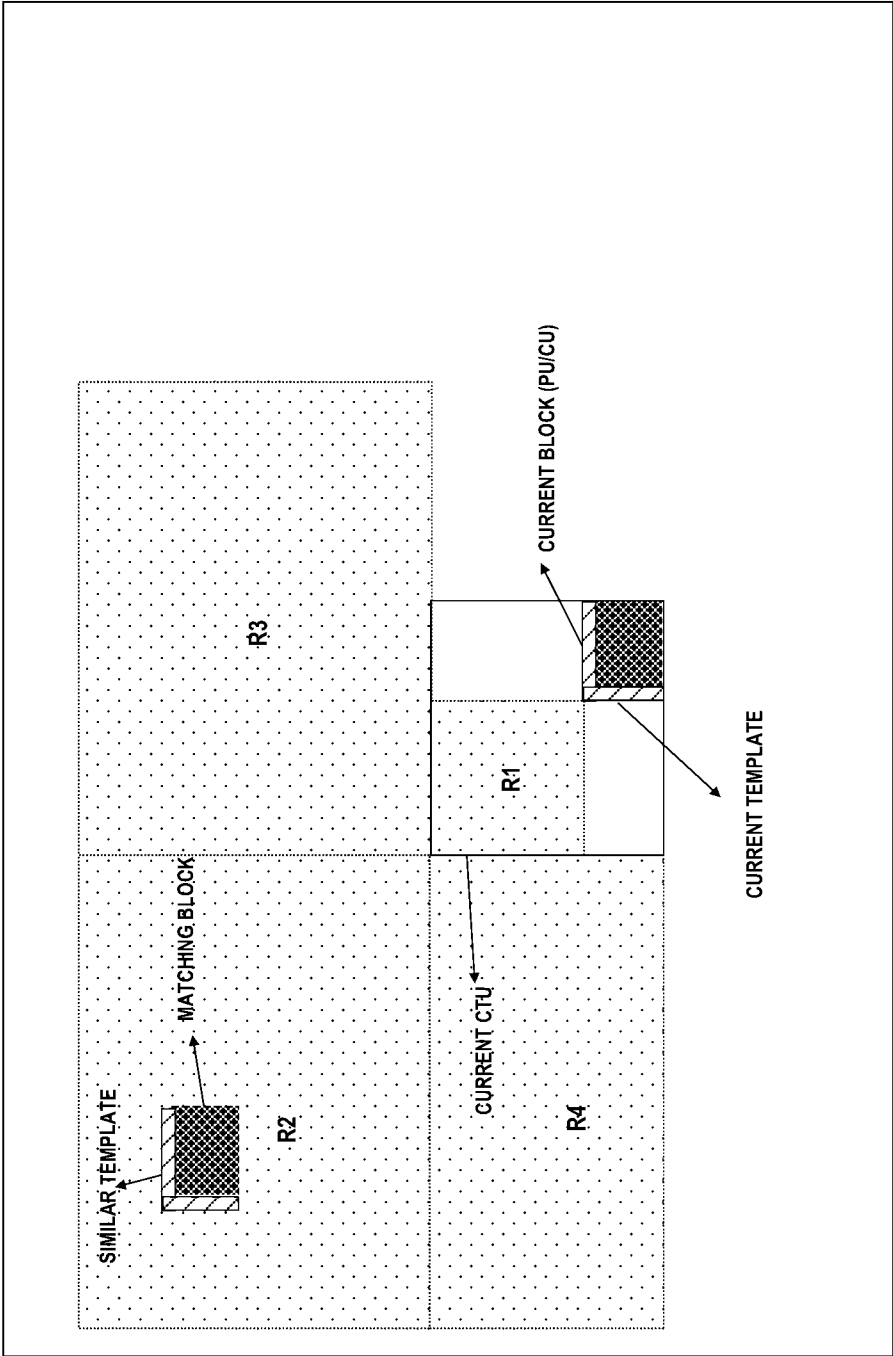


Fig. 15

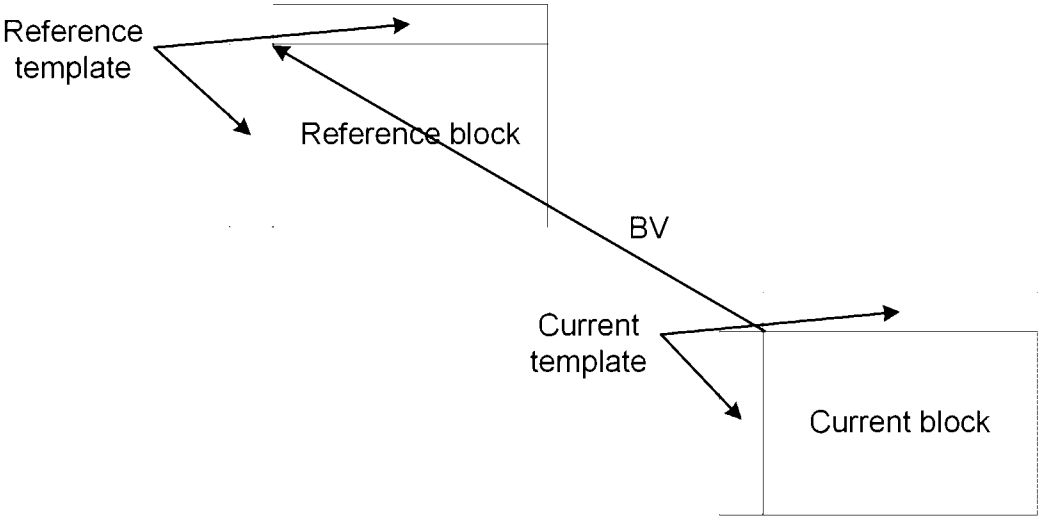


Fig. 16

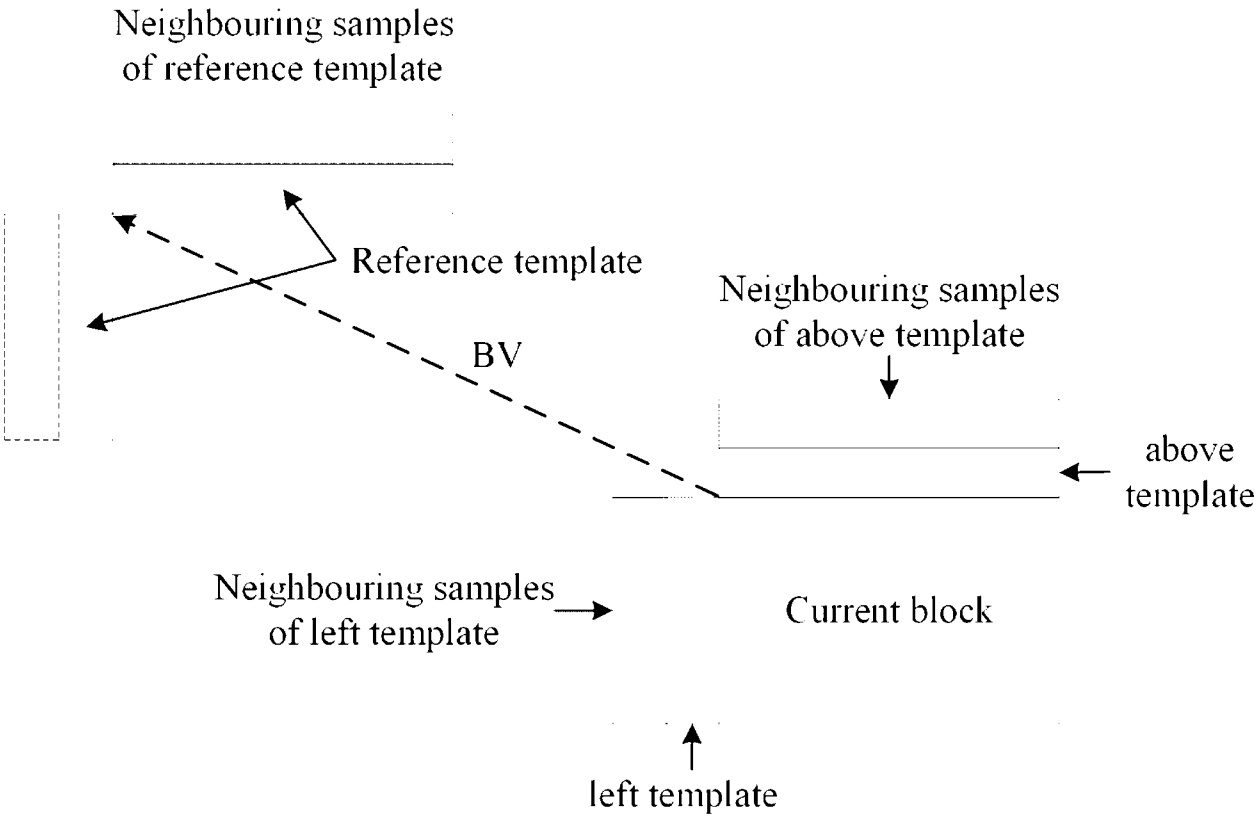


Fig. 17

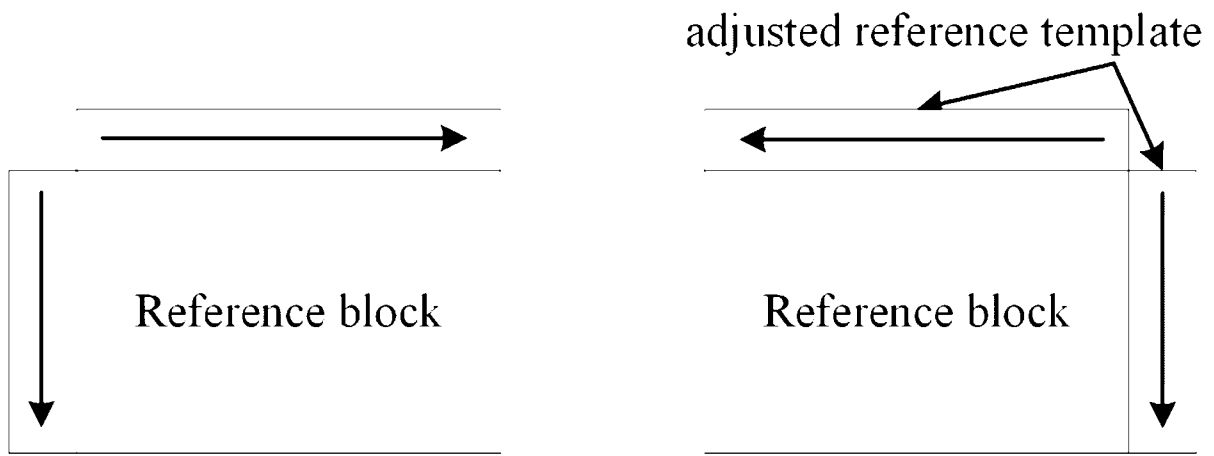


Fig. 18

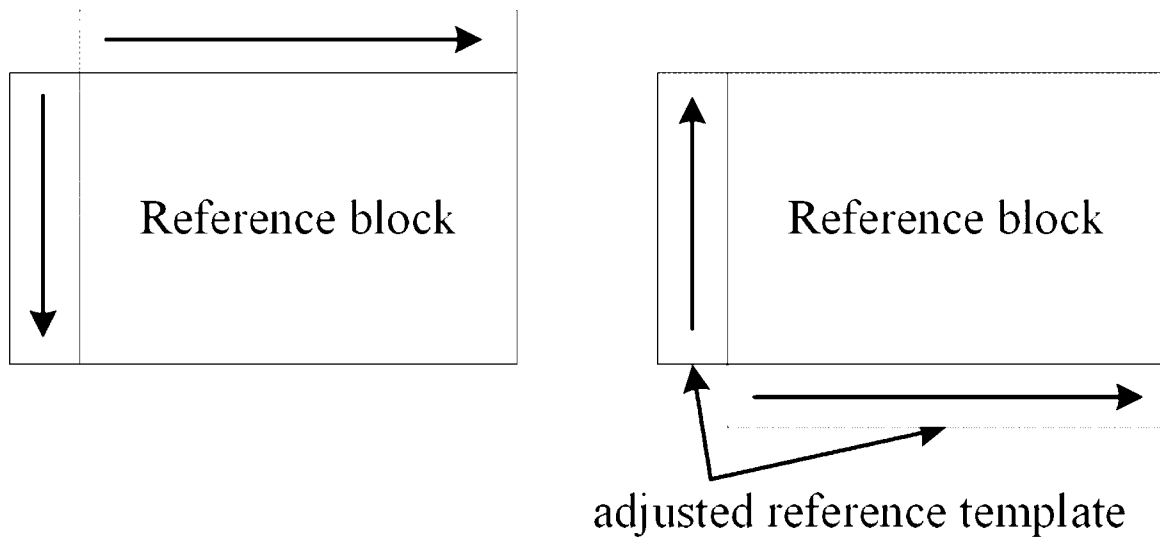
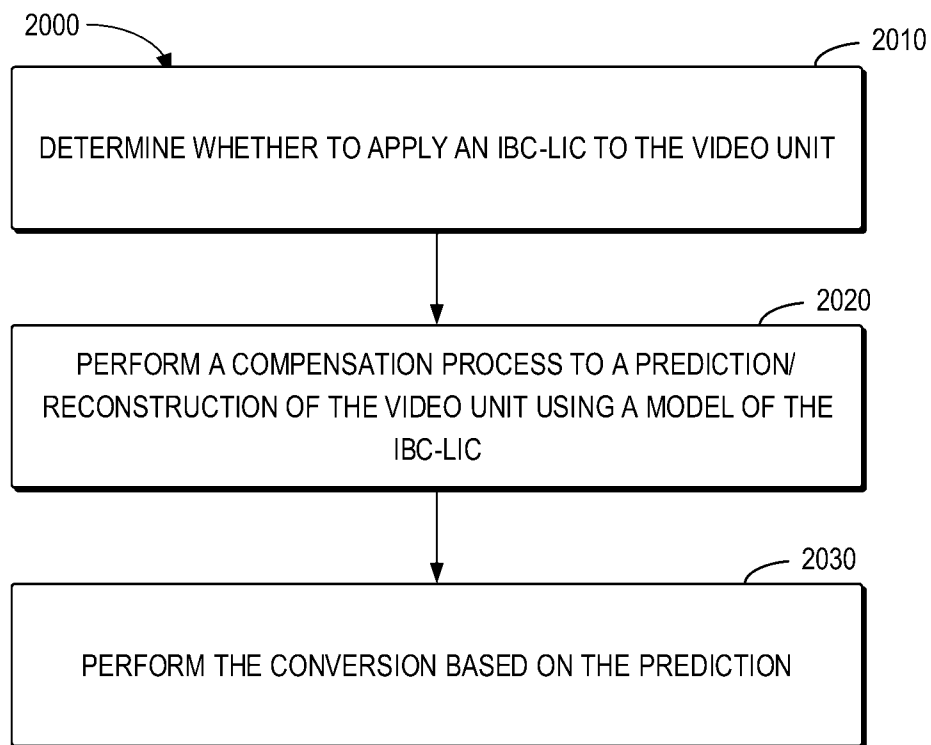
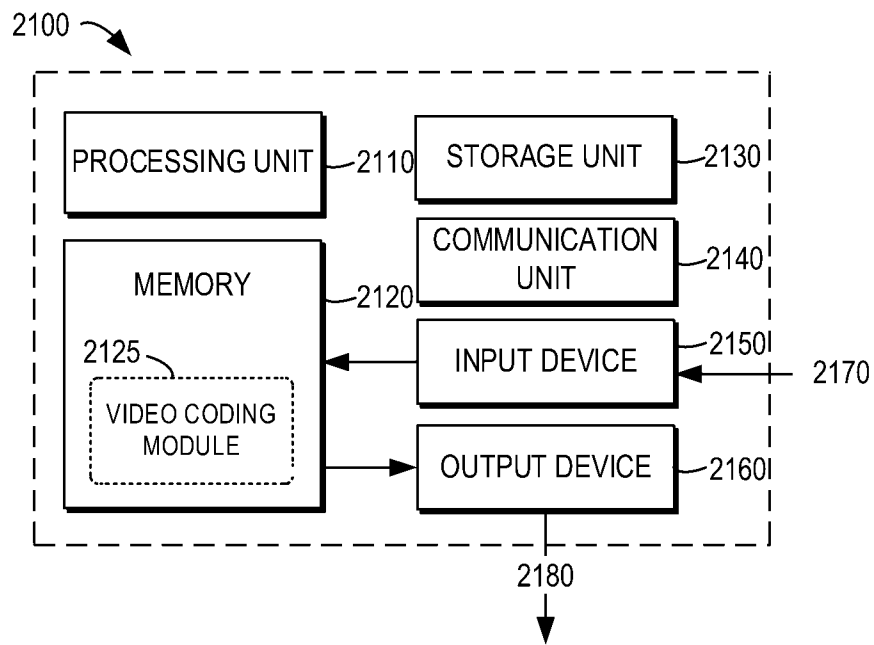


Fig. 19

**Fig. 20**

**Fig. 21**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2024/072864

A. CLASSIFICATION OF SUBJECT MATTER

H04N19/52(2014.01)i; H04N19/593(2014.01)i; H04N19/105(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC:H04N19/-

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)

CNTXT,ENTXT,ENTXTC,DWPI,CNKI,JVET:IBC,LIC,intra,block,copy,illumination,compensation,luma,chroma,interpolation filter,decomposition,colour component,threshold,AMVP,intra TMP

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2022167322 A1 (INTERDIGITAL VC HOLDINGS FRANCE SAS) 11 August 2022 (2022-08-11) description, paragraphs 114-160	1-10,15,19,20,24-26,32,35,36,38,44-50,54-56,58-65,76-78,83,84,102,103,126,127,131-144
Y	WO 2022167322 A1 (INTERDIGITAL VC HOLDINGS FRANCE SAS) 11 August 2022 (2022-08-11) description, paragraphs 114-160	27,28,35,36,38,51-53,57,67-69
Y	CN 112868238 A (BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD.et.al.) 28 May 2021 (2021-05-28) description, paragraphs 543-555,664	51-53,57,67-69
Y	CN 116939198 A (BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. et al.) 24 October 2023 (2023-10-24) description, paragraph 517,563-614	27,28,35,36,38

☒ 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

“D” document cited by the applicant in the international application

“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

22 April 2024

Date of mailing of the international search report

23 April 2024

Name and mailing address of the ISA/CN

CHINA NATIONAL INTELLECTUAL PROPERTY
ADMINISTRATION
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088, China

Authorized officer

MA,YingYing

Telephone No. (+86) 010-62411362

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2024/072864

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2021235074 A1 (BEIJING BYTEDANCE NETWORK TECH CO LTD;BYTEDANCE INC;) 29 July 2021 (2021-07-29) the whole document	11-14,16-18,21-23, 29-31,33,34,37,39-43, 66,70-75,79-82,85- 101,104-125,128-130

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2024/072864

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
WO	2022167322	A1	11 August 2022	None	
CN	112868238	A	28 May 2021	WO 2020084509 A1	30 April 2020
				US 2021235110 A1	29 July 2021
				US 11405607 B2	02 August 2022
				US 2021235092 A1	29 July 2021
				US 11758124 B2	12 September 2023
				US 2023388488 A1	30 November 2023
				WO 2020084506 A1	30 April 2020
				WO 2020084508 A1	30 April 2020
				US 2021235074 A1	29 July 2021
				US 11470307 B2	11 October 2022
CN	116939198	A	24 October 2023	US 2021227209 A1	22 July 2021
				US 11284069 B2	22 March 2022
				WO 2020084502 A1	30 April 2020
				US 2023254479 A1	10 August 2023
				US 2021235109 A1	29 July 2021
				US 11659162 B2	23 May 2023
				WO 2020084507 A1	30 April 2020
US	2021235074	A1	29 July 2021	None	