



(51) International Patent Classification:

H04N 19/105 (2014.01) H04N 19/132 (2014.01)
H04N 19/176 (2014.01) H04N 19/52 (2014.01)
H04N 19/186 (2014.01) H04N 19/593 (2014.01)
H04N 19/11 (2014.01)

(21) International Application Number:

PCT/US2024/010210

(22) International Filing Date:

03 January 2024 (03.01.2024)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

63/436,800	03 January 2023 (03.01.2023)	US
63/439,003	13 January 2023 (13.01.2023)	US
63/487,202	27 February 2023 (27.02.2023)	US

(71) Applicant: **BYTEDANCE INC.** [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) Inventors: **ZHANG, Kai**; c/o Bytedance Inc., 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Li**; c/o Bytedance Inc., 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(74) Agent: **BINDSEIL, James J.**, et al.; Astute IP Law Group, P.O. Box 66358, Washington, DC 20035 (US).

(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, 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))

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

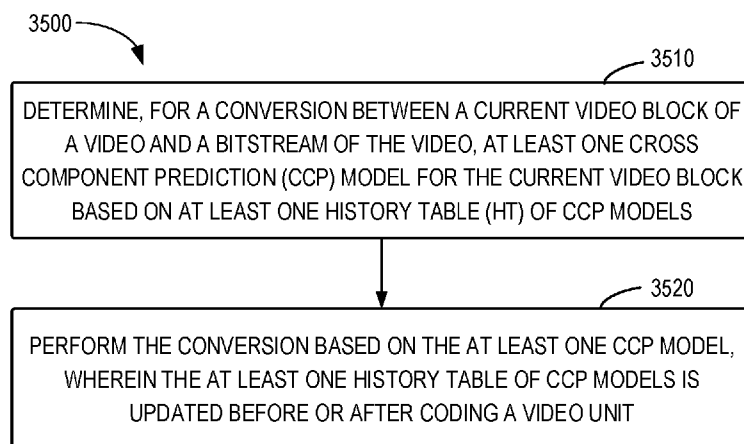


Fig. 35

(57) Abstract: Embodiments of the present disclosure provide a solution for video processing. A method for video processing is proposed. In the method, for a conversion between a current video block of a video and a bitstream of the video, at least one cross component prediction (CCP) model for the current video block is determined based on at least one history table (HT) of CCP models. The conversion is performed based on the at least one CCP model. The at least one history table of CCP models is updated before or after coding a video unit.



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 history-based cross component prediction.

BACKGROUND

[0002] In nowadays, digital video capabilities are being applied in various aspects of
peoples' lives. Multiple types of video compression technologies, such as MPEG-2,
MPEG-4, ITU-TH.263, ITU-TH.264/MPEG-4 Part 10 Advanced Video Coding (AVC),
10 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

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

15 [0004] In a first aspect, a method for video processing is proposed. The method
comprises: determining, for a conversion between a current video block of a video and a
bitstream of the video, at least one cross component prediction (CCP) model for the
current video block based on at least one history table (HT) of CCP models; and
performing the conversion based on the at least one CCP model, wherein the at least one
20 history table of CCP models is updated before or after coding a video unit. The method in
accordance with the first aspect of the present disclosure updates the history table of the
CCP models. In this way, the coding efficiency and coding effectiveness can be improved.

[0005] In a second aspect, an apparatus for video processing is proposed. The apparatus
comprises a processor and a non-transitory memory with instructions thereon. The
25 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
cause a processor to perform a method in accordance with the first aspect of the present
30 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. The method comprises: determining at least one cross component prediction (CCP) model for a current video block of the video based on at least one history table (HT) of CCP models; and generating the bitstream based on the at least one CCP model, wherein the at least one history table of CCP models is updated before or after coding a video unit.

[0008] In a fifth aspect, a method for storing a bitstream of a video is proposed. The method comprises: determining at least one cross component prediction (CCP) model for a current video block of the video based on at least one history table (HT) of CCP models; generating the bitstream based on the at least one CCP model; and storing the bitstream in a non-transitory computer-readable recording medium, wherein the at least one history table of CCP models is updated before or after coding a video unit.

[0009] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

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

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

[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 nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a picture;

- [0015] Fig. 5 illustrates an example of encoder block diagram;
- [0016] Fig. 6 illustrates 67 intra prediction modes;
- [0017] Fig. 7 illustrates reference samples for wide-angular intra prediction;
- [0018] Fig. 8 illustrates problem of discontinuity in case of directions beyond 45° ;
- 5 [0019] Fig. 9 illustrates locations of the samples used for the derivation of α and β ;
- [0020] Fig. 10 illustrates an example of classifying the neighboring samples into two groups;
- [0021] Fig. 11A is a schematic diagram illustrating definition of samples used by PDPC applied to a diagonal top-right mode;
- 10 [0022] Fig. 11B is a schematic diagram illustrating definition of samples used by PDPC applied to a diagonal bottom-left mode;
- [0023] Fig. 11C is a schematic diagram illustrating definition of samples used by PDPC applied to an adjacent diagonal top-right mode;
- [0024] Fig. 11D is a schematic diagram illustrating definition of samples used by PDPC
- 15 applied to an adjacent diagonal bottom-left mode;
- [0025] Fig. 12 is a schematic diagram illustrating gradient approach for non-vertical/non-horizontal mode;
- [0026] Fig. 13 is a schematic diagram illustrating nScale values with respect to nTbH and mode number; for all nScale<0 cases gradient approach is used;
- 20 [0027] Fig. 14 is a schematic diagram illustrating flowcharts of current PDPC and proposed PDPC;
- [0028] Fig. 15 is a schematic diagram illustrating neighbouring blocks (L, A, BL, AR, AL) used in the derivation of a general MPM list;
- [0029] Fig. 16 is a schematic diagram illustrating an example on proposed intra
- 25 reference mapping;
- [0030] Fig. 17 is a schematic diagram illustrating an example of four reference lines neighbouring to a prediction block;
- [0031] Fig. 18A is a schematic diagram illustrating examples of sub-partitions for 4×8

and 8×4 CUs;

[0032] Fig. 18B is a schematic diagram illustrating examples of sub-partitions for CUs other than 4×8 , 8×4 and 4×4 ;

5 [0033] Fig. 19 is a schematic diagram illustrating matrix weighted intra prediction process;

[0034] Fig. 20 is a schematic diagram illustrating target samples, template samples and the reference samples of template used in the DIMD;

[0035] Fig. 21 is a schematic diagram illustrating proposed intra block decoding process;

10 [0036] Fig. 22 is a schematic diagram illustrating HoG computation from a template of width 3 pixels;

[0037] Fig. 23 is a schematic diagram illustrating prediction fusion by weighted averaging of two HoG modes and planar;

[0038] Fig. 24 is a schematic diagram illustrating spatial part of the convolutional filter;

15 [0039] Fig. 25 is a schematic diagram illustrating reference area (with its paddings) used to derive the filter coefficients;

[0040] Fig. 26 is a schematic diagram illustrating four Sobel based gradient patterns for GLM;

[0041] Fig. 27 is a schematic diagram illustrating spatial samples used for GL-CCCM;

[0042] Fig. 28 is a schematic diagram illustrating non-downsampled luma samples;

20 [0043] Fig. 29 illustrates spatial GPM candidates;

[0044] Fig. 30 illustrates GPM templates;

[0045] Fig. 31 illustrates GPM blending;

[0046] Fig. 32 illustrates binarization of cross-component prediction modes in ECM. "CCLM" in Fig. 32 may be replaced by "CCCM";

25 [0047] Fig. 33 illustrates possible positions of candidate regions;

[0048] Fig. 34A to Fig. 34C illustrate possible templates;

[0049] Fig. 35 illustrates a flowchart of a method for video processing in accordance

with embodiments of the present disclosure; and

[0050] Fig. 36 illustrates a block diagram of a computing device in which various embodiments of the present disclosure can be implemented.

[0051] Throughout the drawings, the same or similar reference numerals usually refer to the same or similar elements.

DETAILED DESCRIPTION

[0052] 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.

[0053] 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.

[0054] 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.

[0055] 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.

[0056] 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

[0057] 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.

[0058] 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.

[0059] 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.

[0060] 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.

[0061] 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.

[0062] 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.

[0063] 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.

[0064] In some embodiments, the video encoder 200 may include a partition unit 201, a prediction 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.

[0065] In other examples, the video encoder 200 may include more, fewer, or different functional components. In an example, the prediction unit 202 may include an intra block copy (IBC) unit. The IBC unit may perform prediction in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0066] 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.

[0067] 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.

5 **[0068]** 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 prediction (CIIP)
10 mode in which the prediction is based on an inter prediction signal and an intra prediction 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-prediction.

[0069] 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
15 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.

[0070] The motion estimation unit 204 and the motion compensation unit 205 may
20 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
25 dependent on macroblocks in the same picture.

[0071] 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
30 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.

5 **[0072]** 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
10 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
15 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.

[0073] 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
20 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.

[0074] 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
25 decoder 300 that the current video block has the same motion information as the another video block.

[0075] 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
30 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.

[0076] 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.

5 [0077] 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
10 elements.

[0078] 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
15 components of the samples in the current video block.

[0079] 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.

[0080] The transform processing unit 208 may generate one or more transform
20 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.

[0081] 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
25 more quantization parameter (QP) values associated with the current video block.

[0082] 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
30 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.

[0083] After the reconstruction unit 212 reconstructs the video block, loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[0084] 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.

[0085] 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.

[0086] The video decoder 300 may be configured to perform any or all of the techniques of this disclosure. In the example of Fig. 3, the video decoder 300 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 300. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0087] In the example of Fig. 3, the video decoder 300 includes an entropy decoding unit 301, a motion compensation unit 302, an intra prediction unit 303, an inverse quantization unit 304, an inverse 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.

[0088] The entropy decoding unit 301 may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). The entropy decoding unit 301 may decode the entropy coded video data, and from the entropy decoded video data, the motion compensation unit 302 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. The motion compensation unit 302 may, for example, determine such information by performing the AMVP and merge mode. AMVP is used, including derivation of several most probable candidates based on data from adjacent PBs and the reference picture. Motion information typically includes the horizontal and vertical motion vector displacement values, one or two reference picture indices, and, in

the case of prediction regions in B slices, an identification of which reference picture list is associated with each index. As used herein, in some aspects, a “merge mode” may refer to deriving the motion information from spatially or temporally neighboring blocks.

[0089] 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.

[0090] 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.

[0091] 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.

[0092] 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.

[0093] The reconstruction unit 306 may obtain the decoded blocks, e.g., by summing the residual blocks with the corresponding prediction blocks generated by the motion compensation unit 302 or intra-prediction unit 303. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in the buffer 307, which provides reference blocks

for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

[0094] 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

This disclosure is related to video coding technologies. Specifically, it is related to cross-component prediction. 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. Color space and chroma subsampling

Color space, also known as the color model (or color system), is an abstract mathematical model which simply describes the range of colors as tuples of numbers, typically as 3 or 4 values or color components (e.g., RGB). Basically speaking, color space is an elaboration of the coordinate system and sub-space.

For video compression, the most frequently used color spaces are YCbCr and RGB.

YCbCr, Y'CbCr, or Y Pb/Cb Pr/Cr, also written as YCBCR or Y'CBCR, is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y' is the luma component and CB and CR are the blue-difference and red-difference chroma components. Y' (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries. Chroma subsampling is the practice of encoding images by implementing less resolution for chroma information than for luma information, taking advantage of the human visual system's lower acuity for color differences than for luminance.

2.1.1. 4:4:4

Each of the three Y'CbCr components have the same sample rate, thus there is no chroma subsampling. This scheme is sometimes used in high-end film scanners and cinematic post production.

2.1.2. 4:2:2

The two chroma components are sampled at half the sample rate of luma: the horizontal chroma resolution is halved while the vertical chroma resolution is unchanged. This reduces the bandwidth of an uncompressed video signal by one-third with little to no visual difference. An example of nominal vertical and horizontal locations of 4:2:2 color format is depicted in Fig. 4 in VVC working draft. Fig. 4 illustrates nominal vertical and horizontal locations of 4:2:2 luma and chroma samples in a picture.

2.1.3. 4:2:0

In 4:2:0, the horizontal sampling is doubled compared to 4:1:1, but as the Cb and Cr channels are only sampled on each alternate line in this scheme, the vertical resolution is halved. The data rate is thus the same. Cb and Cr are each subsampled at a factor of 2 both horizontally and vertically. There are three variants of 4:2:0 schemes, having different horizontal and vertical siting.

- In MPEG-2, Cb and Cr are cosited horizontally. Cb and Cr are sited between pixels in the vertical direction (sited interstitially).
- In JPEG/JFIF, H.261, and MPEG-1, Cb and Cr are sited interstitially, halfway between alternate luma samples.
- In 4:2:0 DV, Cb and Cr are co-sited in the horizontal direction. In the vertical direction, they are co-sited on alternating lines.

Table 1 SubWidthC and SubHeightC values derived from chroma_format_idc and separate_colour_plane_flag

chroma_format_idc	separate_colour_plane_flag	Chroma format	SubWidthC	SubHeightC
0	0	Monochrome	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

2.2. Coding flow of a typical video codec

- 5 Fig. 5 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
- 10 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.

2.3. Intra mode coding with 67 intra prediction modes

- 15 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. 6 which illustrates 67 intra prediction modes, 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.

- 20 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.

2.3.1. 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.

- Fig. 7 illustrates 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. 7.

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 2.

Table 2 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

15

Fig. 8 illustrates problem of discontinuity in case of directions beyond 45° . As shown in Fig. 8, 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_a . 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

20

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.

2.4. Intra prediction mode coding for chroma component

For the chroma component of an intra PU, the encoder selects the best chroma prediction modes among five modes including Planar, DC, Horizontal, Vertical and a direct copy of the intra prediction mode for the luma component. The mapping between intra prediction direction and intra prediction mode number for chroma is shown in Table 3.

When the intra prediction mode number for the chroma component is 4, the intra prediction direction for the luma component is used for the intra prediction sample generation for the chroma component. When the intra prediction mode number for the chroma component is not 4 and it is identical to the intra prediction mode number for the luma component, the intra prediction direction of 66 is used for the intra prediction sample generation for the chroma component.

2.5. 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.

2.6. 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.

2.7. Cross-component linear model prediction

To reduce the cross-component redundancy, a cross-component linear model (CCLM)

prediction mode is used in the VVC, for which the chroma samples are predicted based on the reconstructed luma samples of the same CU by using a linear model as follows:

$$\text{pred}_C(i, j) = \alpha \cdot \text{rec}_L'(i, j) + \beta \quad (2-1)$$

where $\text{pred}_C(i, j)$ represents the predicted chroma samples in a CU and $\text{rec}_L(i, j)$ represents the down-sampled reconstructed luma samples of the same CU.

The CCLM parameters (α and β) are derived with at most four neighbouring chroma samples and their corresponding down-sampled luma samples. Suppose the current chroma block dimensions are $W \times H$, then W' and H' are set as

- $W' = W, H' = H$ when LM mode is applied;
- $W' = W + H$ when LM_T mode is applied;
- $H' = H + W$ when LM_L mode is applied;

The above neighbouring positions are denoted as $S[0, -1] \dots S[W' - 1, -1]$ and the left neighbouring positions are denoted as $S[-1, 0] \dots S[-1, H' - 1]$. Then the four samples are selected as

- $S[W' / 4, -1], S[3 * W' / 4, -1], S[-1, H' / 4], S[-1, 3 * H' / 4]$ when LM mode is applied and both above and left neighbouring samples are available;
- $S[W' / 8, -1], S[3 * W' / 8, -1], S[5 * W' / 8, -1], S[7 * W' / 8, -1]$ when LM_T mode is applied or only the above neighbouring samples are available;
- $S[-1, H' / 8], S[-1, 3 * H' / 8], S[-1, 5 * H' / 8], S[-1, 7 * H' / 8]$ when LM_L mode is applied or only the left neighbouring samples are available;

The four neighbouring luma samples at the selected positions are down-sampled and compared four times to find two larger values: x_A^0 and x_A^1 , and two smaller values: x_B^0 and x_B^1 . Their corresponding chroma sample values are denoted as y_A^0, y_A^1, y_B^0 and y_B^1 . Then x_A, x_B, y_A and y_B are derived as:

$$X_a = (x_A^0 + x_A^1 + 1) \gg 1; X_b = (x_B^0 + x_B^1 + 1) \gg 1; Y_a = (y_A^0 + y_A^1 + 1) \gg 1; Y_b = (y_B^0 + y_B^1 + 1) \gg 1 \quad (2-2)$$

Finally, the linear model parameters α and β are obtained according to the following equations.

$$\alpha = \frac{Y_a - Y_b}{X_a - X_b} \quad (2-3)$$

$$\beta = Y_b - \alpha \cdot X_b \quad (2-4)$$

Fig. 9 shows an example of the location of the left and above samples and the sample of the current block involved in the CCLM mode. Fig. 9 illustrates locations of the samples used for the derivation of α and β .

The division operation to calculate parameter α is implemented with a look-up table. To reduce the memory required for storing the table, the *diff* value (difference between maximum and minimum values) and the parameter α are expressed by an exponential notation. For example, *diff* is approximated with a 4-bit significant part and an exponent. Consequently, the table for $1/\text{diff}$ is reduced into 16 elements for 16 values of the significand as follows:

$$\text{DivTable} [] = \{ 0, 7, 6, 5, 5, 4, 4, 3, 3, 2, 2, 1, 1, 1, 1, 0 \} \quad (2-5)$$

This would have a benefit of both reducing the complexity of the calculation as well as the memory size required for storing the needed tables.

Besides the above template and left template can be used to calculate the linear model coefficients together, they also can be used alternatively in the other 2 LM modes, called LM_T, and LM_L modes.

In LM_T mode, only the above template is used to calculate the linear model coefficients. To get more samples, the above template is extended to (W+H) samples. In LM_L mode, only left template is used to calculate the linear model coefficients. To get more samples, the left template is extended to (H+W) samples.

In LM mode, left and above templates are used to calculate the linear model coefficients.

To match the chroma sample locations for 4:2:0 video sequences, two types of down-sampling filter are applied to luma samples to achieve 2 to 1 down-sampling ratio in both horizontal and vertical directions. The selection of down-sampling filter is specified by a SPS level flag. The two down-sampling filters are as follows, which are corresponding to “type-0” and “type-2” content, respectively.

$$\text{Rec}_L'(i, j) = \left[\frac{\text{rec}_L(2i - 1, 2j - 1) + 2 \cdot \text{rec}_L(2i - 1, 2j) + \text{rec}_L(2i + 1, 2j - 1) + \text{rec}_L(2i + 1, 2j) + 4}{\text{rec}_L(2i - 1, 2j) + 2 \cdot \text{rec}_L(2i, 2j) + \text{rec}_L(2i + 1, 2j) + 4} \right] \gg 3 \quad (2-6)$$

$$\text{rec}_L'(i, j) = \left[\frac{\text{rec}_L(2i, 2j - 1) + \text{rec}_L(2i - 1, 2j) + 4 \cdot \text{rec}_L(2i, 2j) + \text{rec}_L(2i + 1, 2j) + \text{rec}_L(2i, 2j + 1) + 4}{\text{rec}_L(2i + 1, 2j) + \text{rec}_L(2i, 2j + 1) + 4} \right] \gg 3 \quad (2-7)$$

Note that only one luma line (general line buffer in intra prediction) is used to make the down-sampled luma samples when the upper reference line is at the CTU boundary.

This parameter computation is performed as part of the decoding process, and is not just as an encoder search operation. As a result, no syntax is used to convey the α and β values to the decoder.

For chroma intra mode coding, a total of 8 intra modes are allowed for chroma intra mode coding. Those modes include five conventional intra modes and three cross-component linear model modes (LM, LM_T, and LM_L). Chroma mode signalling and derivation process are shown in Table 3. Chroma mode coding directly depends on the intra prediction mode of the corresponding luma block. Since separate block partitioning structure for luma and chroma components is enabled in I slices, one chroma block may correspond to multiple luma blocks. Therefore, for Chroma DM mode, the intra prediction mode of the corresponding luma block covering the center position of the current chroma block is directly inherited.

Table 3 Derivation of chroma prediction mode from luma mode when CCLM is enabled

Chroma prediction mode	Corresponding luma intra prediction mode				
	0	50	18	1	X (0 <= X <= 66)
0	66	0	0	0	0
1	50	66	50	50	50
2	18	18	66	18	18
3	1	1	1	66	1
4	0	50	18	1	X
5	81	81	81	81	81
6	82	82	82	82	82
7	83	83	83	83	83

A single binarization table is used regardless of the value of `sps_cclm_enabled_flag` as shown in Table 4.

Table 4 Unified binarization table for chroma prediction mode

Value of <code>intra_chroma_pred_mode</code>	Bin string
4	00
0	0100
1	0101
2	0110
3	0111
5	10
6	110
7	111

In Table 4, the first bin indicates whether it is regular (0) or LM modes (1). If it is LM mode, then the next bin indicates whether it is LM_CHROMA (0) or not. If it is not LM_CHROMA, next 1 bin indicates whether it is LM_L (0) or LM_T (1). For this case, when `sps_cclm_enabled_flag` is 0, the first bin of the binarization table for the corresponding `intra_chroma_pred_mode` can be discarded prior to the entropy coding. Or, in other words, the first bin is inferred to be 0 and hence not coded. This single binarization table is used for both `sps_cclm_enabled_flag` equal to 0 and 1 cases. The first two bins in Table 4 are context coded with its own context model, and the rest bins are bypass coded.

In addition, in order to reduce luma-chroma latency in dual tree, when the 64×64 luma coding tree node is partitioned with Not Split (and ISP is not used for the 64×64 CU) or QT, the chroma CUs in 32×32 / 32×16 chroma coding tree node is allowed to use CCLM in the following way:

- If the 32×32 chroma node is not split or partitioned QT split, all chroma CUs in the 32×32 node can use CCLM.
- If the 32×32 chroma node is partitioned with Horizontal BT, and the 32×16 child node does not split or uses Vertical BT split, all chroma CUs in the 32×16 chroma node can use CCLM.

In all the other luma and chroma coding tree split conditions, CCLM is not allowed for chroma CU.

2.8. Multi-model linear model (MMLM)

With MMLM, there can be more than one linear models between the luma samples and chroma samples in a CU. In this method, neighboring luma samples and neighboring chroma samples of the current block are classified into several groups, each group is used as a training set to derive a linear model (i.e., particular α and β are derived for a particular group). Furthermore, the samples of the current luma block is also classified based on the same rule for the classification of neighboring luma samples.

The neighboring samples can be classified into M groups, where M is 2 or 3. The MMLM method with M=2 and M=3 are designed as two appended Chroma prediction modes named MMLM2 and MMLM3, besides the original LM mode. The encoder chooses the optimal mode in the RDO process and signal the mode.

When M is equal to 2, Fig. 10 shows an example of classifying the neighboring samples into two groups. Threshold is calculated as the average value of the neighboring reconstructed Luma samples. A neighboring sample with $Rec'_L[x,y] \leq Threshold$ is classified into group 1; while a neighboring sample with $Rec'_L[x,y] > Threshold$ is classified into group 2. Similar to CCLM, there are 3 modes in MMLM, namely MMLM, MMLM_T, and MMLM_L. Two models are derived as

$$\begin{cases} Pred_c[x,y] = \alpha_1 \times Rec'_L[x,y] + \beta_1 & \text{if } Rec'_L[x,y] \leq Threshold \\ Pred_c[x,y] = \alpha_2 \times Rec'_L[x,y] + \beta_2 & \text{if } Rec'_L[x,y] > Threshold \end{cases} \quad (2-8).$$

The threshold which is the average of the luma reconstructed neighboring samples. The linear model of each class is derived by using the Least-Mean-Square (LMS) method, if enabled, or min/max method of VVC.

2.9. Position dependent intra prediction combination

In VVC, the results of intra prediction of DC, planar and several angular modes are further modified by a position dependent intra prediction combination (PDPC) method. PDPC is an intra prediction method which invokes a combination of the boundary reference samples and

HEVC style intra prediction with filtered boundary reference samples. PDPC is applied to the following intra modes without signalling: planar, DC, intra angles less than or equal to horizontal, and intra angles greater than or equal to vertical and less than or equal to 80. If the current block is BDPCM mode or MRL index is larger than 0, PDPC is not applied.

- 5 The prediction sample $pred(x', y')$ is predicted using an intra prediction mode (DC, planar, angular) and a linear combination of reference samples according to the Equation 2-8 as follows:

$$pred(x', y') = \text{Clip}(0, (1 \ll \text{BitDepth}) - 1, (wL \times R_{x', -1} + wT \times R_{x', -1} + (64 - wL - wT) \times pred(x', y') + 32) \gg 6) \quad (2-9)$$

- 10 where $R_{x', -1}$, $R_{-1, y'}$ represent the reference samples located at the top and left boundaries of current sample (x, y) , respectively.

If PDPC is applied to DC, planar, horizontal, and vertical intra modes, additional boundary filters are not needed, as required in the case of HEVC DC mode boundary filter or horizontal/vertical mode edge filters. PDPC process for DC and Planar modes is identical. For
 15 angular modes, if the current angular mode is HOR_IDX or VER_IDX, left or top reference samples is not used, respectively. The PDPC weights and scale factors are dependent on prediction modes and the block sizes. PDPC is applied to the block with both width and height greater than or equal to 4.

- Fig. 11A is a schematic diagram illustrating definition of samples used by PDPC applied to a
 20 diagonal top-right mode. Fig. 11B is a schematic diagram illustrating definition of samples used by PDPC applied to a diagonal bottom-left mode. Fig. 11C is a schematic diagram illustrating definition of samples used by PDPC applied to an adjacent diagonal top-right mode. Fig. 11D is a schematic diagram illustrating definition of samples used by PDPC applied to an adjacent diagonal bottom-left mode.

- 25 Fig. 11A to Fig. 11D illustrate the definition of reference samples ($R_{x', -1}$ and $R_{-1, y'}$) for PDPC applied over various prediction modes. The prediction sample $pred(x', y')$ is located at (x', y') within the prediction block. As an example, the coordinate x of the reference sample $R_{x', -1}$ is given by: $x = x' + y' + 1$, and the coordinate y of the reference sample $R_{-1, y'}$ is similarly given by: $y = x' + y' + 1$ for the diagonal modes. For the other angular mode, the reference samples
 30 $R_{x', -1}$ and $R_{-1, y'}$ could be located in fractional sample position. In this case, the sample value of the nearest integer sample location is used.

2.10. Gradient PDPC

- The gradient based approach is extended for non-vertical/non-horizontal mode, as shown in Fig. 12. Here, the gradient is computed as $r(-1, y) - r(-1 + d, -1)$, where d is the horizontal
 35 displacement depending on the angular direction. A few points to note here:

The gradient term $r(-1, y) - r(-1 + d, -1)$ is needed to be computed once for every row, as it does

not depend on the x position.

The computation of d is already part of original intra prediction process which can be reused, so a separate computation of d is not needed. Accordingly, d is in 1/32 pixel accuracy.

Two tap (linear) filtering are used when d is at fractional position, i.e., if dPos is the displacement in 1/32 pixel accuracy, dInt is the (floored) integer part ($dPos \gg 5$), and dFrac is the fractional part in 1/32 pixel accuracy ($dPos \& 31$), then $r(-1+d)$ is computed as:

$$r(-1+d) = (32 - dFrac) * r(-1+dInt) + dFrac * r(-1+dInt+1).$$

This 2 tap filtering is performed once per row (if needed), as explained in a.

Finally, the prediction signal is computed.

$$10 \quad p(x,y) = \text{Clip}((64 - wL(x)) * p(x,y) + wL(x) * (r(-1, y) - r(-1+d,-1)) + 32) \gg 6)$$

Where $wL(x) = 32 \gg ((x < 1) \gg nScale2)$, and $nScale2 = (\log_2(nTbH) + \log_2(nTbW) - 2) \gg 2$, which are the same as vertical/horizontal mode. In a nutshell, the same process is applied compared to vertical/horizontal mode (in fact, $d = 0$ indicates vertical/horizontal mode).

15 Second, the gradient based approach is activated for non-vertical/non-horizontal mode when ($nScale < 0$) or when PDPC can't be applied due to unavailability of secondary reference sample. We have shown the values of nScale in Fig. 13, with respect to TB size and angular mode, to better visualize the cases where gradient approach is used. Additionally, in Fig. 14, we have shown the flowchart for current and proposed PDPC. Fig. 14 is a schematic diagram illustrating
20 flowcharts of current PDPC (left) and proposed PDPC (right).

2.11. Secondary MPM

The existing primary MPM (PMPM) list consists of 6 entries and the secondary MPM (SMPM) list includes 16 entries. A general MPM list with 22 entries is constructed first, and then the first 6 entries in this general MPM list are included into the PMPM list, and the rest of entries
25 form the SMPM list. The first entry in the general MPM list is the Planar mode. The remaining entries are composed of the intra modes of the left (L), above (A), below-left (BL), above-right (AR), and above-left (AL) neighbouring blocks as shown in Fig. 15, the directional modes with added offset from the first two available directional modes of neighbouring blocks, and the default modes.

30 If a CU block is vertically oriented, the order of neighbouring blocks is A, L, BL, AR, AL; otherwise, it is L, A, BL, AR, AL.

A PMPM flag is parsed first, if equal to 1 then a PMPM index is parsed to determine which entry of the PMPM list is selected, otherwise the SPMPM flag is parsed to determine whether to parse the SMPM index or the remaining modes.

2.12. 6-tap intra interpolation filter

To improve prediction accuracy, it is proposed to replace 4-tap Cubic interpolation filter with 6-tap interpolation filter, the filter coefficients are derived based on the same polynomial regression model, but with polynomial order of 6.

5 Filter coefficients are listed below,

```

    { 0, 0, 256, 0, 0, 0 }, // 0/32 position
    { 0, -4, 253, 9, -2, 0 }, // 1/32 position
    { 1, -7, 249, 17, -4, 0 }, // 2/32 position
10  { 1, -10, 245, 25, -6, 1 }, // 3/32 position
    { 1, -13, 241, 34, -8, 1 }, // 4/32 position
    { 2, -16, 235, 44, -10, 1 }, // 5/32 position
    { 2, -18, 229, 53, -12, 2 }, // 6/32 position
    { 2, -20, 223, 63, -14, 2 }, // 7/32 position
15  { 2, -22, 217, 72, -15, 2 }, // 8/32 position
    { 3, -23, 209, 82, -17, 2 }, // 9/32 position
    { 3, -24, 202, 92, -19, 2 }, // 10/32 position
    { 3, -25, 194, 101, -20, 3 }, // 11/32 position
    { 3, -25, 185, 111, -21, 3 }, // 12/32 position
20  { 3, -26, 178, 121, -23, 3 }, // 13/32 position
    { 3, -25, 168, 131, -24, 3 }, // 14/32 position
    { 3, -25, 159, 141, -25, 3 }, // 15/32 position
    { 3, -25, 150, 150, -25, 3 }, // half-pel position.
```

The reference samples used for interpolation come from reconstructed samples or padded as in
 25 HEVC, so that the conditional check on reference sample availability is not needed.

Instead of using nearest rounding operation to derive the extended Intra reference sample, it is proposed to use 4-tap Cubic interpolation filter. As shown in an example in Fig. 16, to derive the value of reference sample P, a four tap interpolation filter is used, while in JEM-3.0 or HM, P is directly set as X1.

30 2.13. Multiple reference line (MRL) intra prediction

Multiple reference line (MRL) intra prediction uses more reference lines for intra prediction. In Fig. 17, an example of 4 reference lines is depicted, where the samples of segments A and F are not fetched from reconstructed neighbouring samples but padded with the closest samples from Segment B and E, respectively. HEVC intra-picture prediction uses the nearest reference line
 35 (i.e., reference line 0). In MRL, 2 additional lines (reference line 1 and reference line 2) are used.

The index of selected reference line (mrl_idx) is signalled and used to generate intra predictor. For reference line index, which is greater than 0, only include additional reference line modes in MPM list and only signal MPM index without remaining mode. The reference line index is signalled before intra prediction modes, and Planar mode is excluded from intra prediction modes in case a nonzero reference line index is signalled.

MRL is disabled for the first line of blocks inside a CTU to prevent using extended reference samples outside the current CTU line. Also, PDPC is disabled when additional line is used. For MRL mode, the derivation of DC value in DC intra prediction mode for non-zero reference line indices are aligned with that of reference line index 0. MRL requires the storage of 3 neighbouring luma reference lines with a CTU to generate predictions. The Cross-Component Linear Model (CCLM) tool also requires 3 neighbouring luma reference lines for its down-sampling filters. The definition of MRL to use the same 3 lines is aligned as CCLM to reduce the storage requirements for decoders.

2.14. Intra sub-partitions (ISP)

The intra sub-partitions (ISP) divides luma intra-predicted blocks vertically or horizontally into 2 or 4 sub-partitions depending on the block size. For example, minimum block size for ISP is 4×8 (or 8×4). If block size is greater than 4×8 (or 8×4) then the corresponding block is divided by 4 sub-partitions. It has been noted that the $M \times 128$ (with $M \leq 64$) and $128 \times N$ (with $N \leq 64$) ISP blocks could generate a potential issue with the 64×64 VDPU. For example, an $M \times 128$ CU in the single tree case has an $M \times 128$ luma TB and two corresponding $\frac{M}{2} \times 64$ chroma TBs. If the CU uses ISP, then the luma TB will be divided into four $M \times 32$ TBs (only the horizontal split is possible), each of them smaller than a 64×64 block. However, in the current design of ISP chroma blocks are not divided. Therefore, both chroma components will have a size greater than a 32×32 block. Analogously, a similar situation could be created with a $128 \times N$ CU using ISP. Hence, these two cases are an issue for the 64×64 decoder pipeline. For this reason, the CU sizes that can use ISP is restricted to a maximum of 64×64 . Fig. 18A and Fig. 18B show examples of the two possibilities. All sub-partitions fulfill the condition of having at least 16 samples.

In ISP, the dependence of $1 \times N/2 \times N$ subblock prediction on the reconstructed values of previously decoded $1 \times N/2 \times N$ subblocks of the coding block is not allowed so that the minimum width of prediction for subblocks becomes four samples. For example, an $8 \times N$ ($N > 4$) coding block that is coded using ISP with vertical split is split into two prediction regions each of size $4 \times N$ and four transforms of size $2 \times N$. Also, a $4 \times N$ coding block that is coded using ISP with vertical split is predicted using the full $4 \times N$ block; four transform each of $1 \times N$ is used. Although the transform sizes of $1 \times N$ and $2 \times N$ are allowed, it is asserted that the transform of these blocks in $4 \times N$ regions can be performed in parallel. For example, when a $4 \times N$ prediction region contains four $1 \times N$ transforms, there is no transform in the horizontal direction; the transform in

the vertical direction can be performed as a single $4 \times N$ transform in the vertical direction. Similarly, when a $4 \times N$ prediction region contains two $2 \times N$ transform blocks, the transform operation of the two $2 \times N$ blocks in each direction (horizontal and vertical) can be conducted in parallel. Thus, there is no delay added in processing these smaller blocks than processing 4×4 regular-coded intra blocks.

Fig. 18A is a schematic diagram illustrating examples of sub-partitions for 4×8 and 8×4 CUs.

Fig. 18B is a schematic diagram illustrating examples of sub-partitions for CUs other than 4×8 , 8×4 and 4×4 .

Table 5 Entropy coding coefficient group size

Block Size	Coefficient group Size
$1 \times N, N \geq 16$	1×16
$N \times 1, N \geq 16$	16×1
$2 \times N, N \geq 8$	2×8
$N \times 2, N \geq 8$	8×2
All other possible $M \times N$ cases	4×4

For each sub-partition, reconstructed samples are obtained by adding the residual signal to the prediction signal. Here, a residual signal is generated by the processes such as entropy decoding, inverse quantization and inverse transform. Therefore, the reconstructed sample values of each sub-partition are available to generate the prediction of the next sub-partition, and each sub-partition is processed repeatedly. In addition, the first sub-partition to be processed is the one containing the top-left sample of the CU and then continuing downwards (horizontal split) or rightwards (vertical split). As a result, reference samples used to generate the sub-partitions prediction signals are only located at the left and above sides of the lines. All sub-partitions share the same intra mode. The followings are summary of interaction of ISP with other coding tools.

- Multiple Reference Line (MRL): if a block has an MRL index other than 0, then the ISP coding mode will be inferred to be 0 and therefore ISP mode information will not be sent to the decoder.
- Entropy coding coefficient group size: the sizes of the entropy coding subblocks have been modified so that they have 16 samples in all possible cases, as shown in Table 5. Note that the new sizes only affect blocks produced by ISP in which one of the dimensions is less than 4 samples. In all other cases coefficient groups keep the 4×4 dimensions.

- CBF coding: it is assumed to have at least one of the sub-partitions has a non-zero CBF. Hence, if n is the number of sub-partitions and the first $n - 1$ sub-partitions have produced a zero CBF, then the CBF of the n -th sub-partition is inferred to be 1.
- Transform size restriction: all ISP transforms with a length larger than 16 points uses the DCT-II.
- MTS flag: if a CU uses the ISP coding mode, the MTS CU flag will be set to 0 and it will not be sent to the decoder. Therefore, the encoder will not perform RD tests for the different available transforms for each resulting sub-partition. The transform choice for the ISP mode will instead be fixed and selected according the intra mode, the processing order and the block size utilized. Hence, no signalling is required. For example, let t_H and t_V be the horizontal and the vertical transforms selected respectively for the $w \times h$ sub-partition, where w is the width and h is the height. Then the transform is selected according to the following rules:
 - If $w = 1$ or $h = 1$, then there is no horizontal or vertical transform respectively.
 - If $w \geq 4$ and $w \leq 16$, $t_H = \text{DST-VII}$, otherwise, $t_H = \text{DCT-II}$.
 - If $h \geq 4$ and $h \leq 16$, $t_V = \text{DST-VII}$, otherwise, $t_V = \text{DCT-II}$.

In ISP mode, all 67 intra prediction modes are allowed. PDPC is also applied if corresponding width and height is at least 4 samples long. In addition, the reference sample filtering process (reference smoothing) and the condition for intra interpolation filter selection doesn't exist anymore, and Cubic (DCT-IF) filter is always applied for fractional position interpolation in ISP mode.

2.15. Matrix weighted Intra Prediction (MIP)

Matrix weighted intra prediction (MIP) method is a newly added intra prediction technique into VVC. For predicting the samples of a rectangular block of width W and height H , matrix weighted intra prediction (MIP) takes one line of H reconstructed neighbouring boundary samples left of the block and one line of W reconstructed neighbouring boundary samples above the block as input. If the reconstructed samples are unavailable, they are generated as it is done in the conventional intra prediction. The generation of the prediction signal is based on the following three steps, which are averaging, matrix vector multiplication and linear interpolation as shown in Fig. 19.

2.15.1. Averaging neighbouring samples

Among the boundary samples, four samples or eight samples are selected by averaging based on block size and shape. Specifically, the input boundaries $bdry^{top}$ and $bdry^{left}$ are reduced

to smaller boundaries $bdry_{red}^{top}$ and $bdry_{red}^{left}$ by averaging neighbouring boundary samples according to predefined rule depends on block size. Then, the two reduced boundaries $bdry_{red}^{top}$ and $bdry_{red}^{left}$ are concatenated to a reduced boundary vector $bdry_{red}$ which is thus of size four for blocks of shape 4×4 and of size eight for blocks of all other shapes. If *mode* refers to the

5 MIP-mode, this concatenation is defined as follows:

$$bdry_{red} = \begin{cases} [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } W = H = 4 \text{ and } mode < 18 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } W = H = 4 \text{ and } mode \geq 18 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } \max(W, H) = 8 \text{ and } mode < 10 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } \max(W, H) = 8 \text{ and } mode \geq 10 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } \max(W, H) > 8 \text{ and } mode < 6 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } \max(W, H) > 8 \text{ and } mode \geq 6. \end{cases} \quad (2-10).$$

2.15.2. Matrix Multiplication

A matrix vector multiplication, followed by addition of an offset, is carried out with the averaged samples as an input. The result is a reduced prediction signal on a subsampled set of samples in the original block. Out of the reduced input vector $bdry_{red}$ a reduced prediction signal $pred_{red}$, which is a signal on the down-sampled block of width W_{red} and height H_{red} is generated. Here, W_{red} and H_{red} are defined as:

$$W_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(W, 8) & \text{for } \max(W, H) > 8 \end{cases} \quad (2-11)$$

$$H_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(H, 8) & \text{for } \max(W, H) > 8 \end{cases} \quad (2-12).$$

15 The reduced prediction signal $pred_{red}$ is computed by calculating a matrix vector product and adding an offset:

$$pred_{red} = A \cdot bdry_{red} + b. \quad (2-13).$$

Here, A is a matrix that has $W_{red} \cdot H_{red}$ rows and 4 columns if $W = H = 4$ and 8 columns in all other cases. b is a vector of size $W_{red} \cdot H_{red}$. The matrix A and the offset vector b are taken

20 from one of the sets S_0, S_1, S_2 . One defines an index $idx = idx(W, H)$ as follows:

$$idx(W, H) = \begin{cases} 0 & \text{for } W = H = 4 \\ 1 & \text{for } \max(W, H) = 8 \\ 2 & \text{for } \max(W, H) > 8. \end{cases} \quad (2-14).$$

Here, each coefficient of the matrix A is represented with 8 bit precision. The set S_0 consists of 16 matrices $A_0^i, i \in \{0, \dots, 15\}$ each of which has 16 rows and 4 columns and 16 offset vectors $b_0^i, i \in \{0, \dots, 16\}$ each of size 16. Matrices and offset vectors of that set are used for blocks of size 4×4 . The set S_1 consists of 8 matrices $A_1^i, i \in \{0, \dots, 7\}$, each of which has 16 rows and 8 columns and 8 offset vectors $b_1^i, i \in \{0, \dots, 7\}$ each of size 16. The set S_2 consists of 6 matrices $A_2^i, i \in \{0, \dots, 5\}$, each of which has 64 rows and 8 columns and of 6 offset vectors $b_2^i, i \in \{0, \dots, 5\}$ of size 64.

2.15.3. Interpolation

The prediction signal at the remaining positions is generated from the prediction signal on the subsampled set by linear interpolation which is a single step linear interpolation in each direction. The interpolation is performed firstly in the horizontal direction and then in the vertical direction regardless of block shape or block size.

2.15.4. Signalling of MIP mode and harmonization with other coding tools

For each Coding Unit (CU) in intra mode, a flag indicating whether an MIP mode is to be applied or not is sent. If an MIP mode is to be applied, MIP mode (*predModelIntra*) is signalled. For an MIP mode, a transposed flag (*isTransposed*), which determines whether the mode is transposed, and MIP mode Id (*modelId*), which determines which matrix is to be used for the given MIP mode is derived as follows

$$isTransposed = predModelIntra \& 1$$

$$modelId = predModelIntra \gg 1 \quad (2-15).$$

MIP coding mode is harmonized with other coding tools by considering following aspects:

- LFNST is enabled for MIP on large blocks. Here, the LFNST transforms of planar mode are used.
- The reference sample derivation for MIP is performed exactly as for the conventional intra prediction modes.
- For the up-sampling step used in the MIP-prediction, original reference samples are used instead of down-sampled ones.
- Clipping is performed before up-sampling and not after up-sampling.
- MIP is allowed up to 64×64 regardless of the maximum transform size.

The number of MIP modes is 32 for sizeId=0, 16 for sizeId=1 and 12 for sizeId=2.

2.16. Decoder-side intra mode derivation

In JEM-2.0 intra modes are extended to 67 from 35 modes in HEVC, and they are derived at encoder and explicitly signalled to decoder. A significant amount of overhead is spent on intra mode coding in JEM-2.0. For example, the intra mode signalling overhead may be up to 5~10% of overall bitrate in all intra coding configuration. This contribution proposes the decoder-side intra mode derivation approach to reduce the intra mode coding overhead while keeping prediction accuracy.

To reduce the overhead of intra mode signalling, this contribution presents a decoder-side intra mode derivation (DIMD) approach. In the proposed approach, instead of signalling intra mode explicitly, the information is derived at both encoder and decoder from the neighbouring reconstructed samples of current block. The intra mode derived by DIMD is used in two ways:

- 1) For 2N×2N CUs, the DIMD mode is used as the intra mode for intra prediction when the

corresponding CU-level DIMD flag is turned on;

2) For $N \times N$ CUs, the DIMD mode is used to replace one candidate of the existing MPM list to improve the efficiency of intra mode coding.

2.16.1. Templated based intra mode derivation

5 Fig. 20 is a schematic diagram illustrating target samples, template samples and the reference samples of template used in the DIMD. As illustrated in Fig. 20, the target denotes the current block (of block size N) for which intra prediction mode is to be estimated. The template (indicated by the patterned region in Fig. 20) specifies a set of already reconstructed samples, which are used to derive the intra mode. The template size is denoted as the number of samples
10 within the template that extends to the above and the left of the target block, i.e., L . In the current implementation, a template size of 2 (i.e., $L = 2$) is used for 4×4 and 8×8 blocks and a template size of 4 (i.e., $L = 4$) is used for 16×16 and larger blocks. The reference of template (indicated by the dotted region in Fig. 20) refers to a set of neighbouring samples from above and left of the template, as defined by JEM-2.0. Unlike the template samples which are always
15 from reconstructed region, the reference samples of template may not be reconstructed yet when encoding/decoding the target block. In this case, the existing reference samples substitution algorithm of JEM-2.0 is utilized to substitute the unavailable reference samples with the available reference samples.

For each intra prediction mode, the DIMD calculates the absolute difference (SAD) between
20 the reconstructed template samples and its prediction samples obtained from the reference samples of the template. The intra prediction mode that yields the minimum SAD is selected as the final intra prediction mode of the target block.

2.16.2. DIMD for intra $2N \times 2N$ CUs

For intra $2N \times 2N$ CUs, the DIMD is used as one additional intra mode, which is adaptively
25 selected by comparing the DIMD intra mode with the optimal normal intra mode (i.e., being explicitly signalled). One flag is signalled for each intra $2N \times 2N$ CU to indicate the usage of the DIMD. If the flag is one, then the CU is predicted using the intra mode derived by DIMD; otherwise, the DIMD is not applied and the CU is predicted using the intra mode explicitly signalled in the bit-stream. When the DIMD is enabled, chroma components always reuse the
30 same intra mode as that derived for luma component, i.e., DM mode.

Additionally, for each DIMD-coded CU, the blocks in the CU can adaptively select to derive their intra modes at either PU-level or TU-level. Specifically, when the DIMD flag is one, another CU-level DIMD control flag is signalled to indicate the level at which the DIMD is performed. If this flag is zero, it means that the DIMD is performed at the PU level and all the
35 TUs in the PU use the same derived intra mode for their intra prediction; otherwise (i.e., the

DIMD control flag is one), it means that the DIMD is performed at the TU level and each TU in the PU derives its own intra mode.

Further, when the DIMD is enabled, the number of angular directions increases to 129, and the DC and planar modes still remain the same. To accommodate the increased granularity of angular intra modes, the precision of intra interpolation filtering for DIMD-coded CUs increases from 1/32-pel to 1/64-pel. Additionally, in order to use the derived intra mode of a DIMD coded CU as MPM candidate for neighbouring intra blocks, those 129 directions of the DIMD-coded CUs are converted to “normal” intra modes (i.e., 65 angular intra directions) before they are used as MPM.

2.16.3. DIMD for intra N×N CUs

In the proposed method, intra modes of intra N×N CUs are always signalled. However, to improve the efficiency of intra mode coding, the intra modes derived from DIMD are used as MPM candidates for predicting the intra modes of four PUs in the CU. In order to not increase the overhead of MPM index signalling, the DIMD candidate is always placed at the first place in the MPM list and the last existing MPM candidate is removed. Also, pruning operation is performed such that the DIMD candidate will not be added to the MPM list if it is redundant.

2.16.4. Intra mode search algorithm of DIMD

In order to reduce encoding/decoding complexity, one straightforward fast intra mode search algorithm is used for DIMD. Firstly, one initial estimation process is performed to provide a good starting point for intra mode search. Specifically, an initial candidate list is created by selecting N fixed modes from the allowed intra modes. Then, the SAD is calculated for all the candidate intra modes and the one that minimizes the SAD is selected as the starting intra mode. To achieve a good complexity/performance trade-off, the initial candidate list consists of 11 intra modes, including DC, planar and every 4-th mode of the 33 angular intra directions as defined in HEVC, i.e., intra modes 0, 1, 2, 6, 10... 30, 34.

If the starting intra mode is either DC or planar, it is used as the DIMD mode. Otherwise, based on the starting intra mode, one refinement process is then applied where the optimal intra mode is identified through one iterative search. It works by comparing at each iteration the SAD values for three intra modes separated by a given search interval and maintain the intra mode that minimize the SAD. The search interval is then reduced to half, and the selected intra mode from the last iteration will serve as the center intra mode for the current iteration. For the current DIMD implementation with 129 angular intra directions, up to 4 iterations are used in the refinement process to find the optimal DIMD intra mode.

2.17. Decoder-side intra mode derivation by calculating the gradients of neighbouring samples

Three angular modes are selected from a Histogram of Gradient (HoG) computed from the neighboring pixels of current block. Once the three modes are selected, their predictors are computed normally and then their weighted average is used as the final predictor of the block. To determine the weights, corresponding amplitudes in the HoG are used for each of the three modes. The DIMD mode is used as an alternative prediction mode and is always checked in the FullRD mode.

Current version of DIMD has modified some aspects in the signaling, HoG computation and the prediction fusion. The purpose of this modification is to improve the coding performance as well as addressing the complexity concerns raised during the last meeting (i.e., throughput of 4x4 blocks). The following sections describe the modifications for each aspect.

2.17.1. Signalling

Fig. 21 is a schematic diagram illustrating proposed intra block decoding process. Fig. 21 shows the order of parsing flags/indices in VTM5, integrated with the proposed DIMD.

As can be seen, the DIMD flag of the block is parsed first using a single CABAC context, which is initialized to the default value of 154.

If flag = 0, then the parsing continues normally.

Else (if flag = 1), only the ISP index is parsed and the following flags/indices are inferred to be zero: BDPCM flag, MIP flag, MRL index. In this case, the entire IPM parsing is also skipped. During the parsing phase, when a regular non-DIMD block inquires the IPM of its DIMD neighbor, the mode PLANAR_IDX is used as the virtual IPM of the DIMD block.

2.17.2. Texture analysis

Fig. 22 is a schematic diagram illustrating HoG computation from a template of width 3 pixels. The texture analysis of DIMD includes a Histogram of Gradient (HoG) computation (Fig. 22). The HoG computation is carried out by applying horizontal and vertical Sobel filters on pixels in a template of width 3 around the block. Except, if above template pixels fall into a different CTU, then they will not be used in the texture analysis.

Once computed, the IPMs corresponding to two tallest histogram bars are selected for the block.

In previous versions, all pixels in the middle line of the template were involved in the HoG computation. However, the current version improves the throughput of this process by applying the Sobel filter more sparsely on 4x4 blocks. To this aim, only one pixel from left and one pixel from above are used. This is shown in Fig. 22.

In addition to reduction in the number of operations for gradient computation, this property also

simplifies the selection of best 2 modes from the HoG, as the resulting HoG cannot have more than two non-zero amplitudes.

2.17.3. Prediction fusion

The current method uses a fusion of three predictors for each block. However, the choice of prediction modes is different and makes use of the combined hypothesis intra-prediction method proposed, where the Planar mode is considered to be used in combination with other modes when computing an intra-predicted candidate. In the current version, the two IPMs corresponding to two tallest HoG bars are combined with the Planar mode.

The prediction fusion is applied as a weighted average of the above three predictors. To this aim, the weight of planar is fixed to $21/64$ ($\sim 1/3$). The remaining weight of $43/64$ ($\sim 2/3$) is then shared between the two HoG IPMs, proportionally to the amplitude of their HoG bars. Fig. 23 visualises this process. Fig. 23 is a schematic diagram illustrating prediction fusion by weighted averaging of two HoG modes and planar.

2.18. Template-based intra mode derivation (TIMD)

This contribution proposes a template-based intra mode derivation (TIMD) method using MPMs, in which a TIMD mode is derived from MPMs using the neighbouring template. The TIMD mode is used as an additional intra prediction method for a CU.

2.18.1. TIMD mode derivation

For each intra prediction mode in MPMs, The SATD between the prediction and reconstruction samples of the template is calculated. The intra prediction mode with the minimum SATD is selected as the TIMD mode and used for intra prediction of current CU. Position dependent intra prediction combination (PDPC) is included in the derivation of the TIMD mode.

2.18.2. TIMD signalling

A flag is signalled in sequence parameter set (SPS) to enable/disable the proposed method. When the flag is true, a CU level flag is signalled to indicate whether the proposed TIMD method is used. The TIMD flag is signalled right after the MIP flag. If the TIMD flag is equal to true, the remaining syntax elements related to luma intra prediction mode, including MRL, ISP, and normal parsing stage for luma intra prediction modes, are all skipped.

2.18.3. Interaction with new coding tools

A DIMD method with prediction fusion using Planar was integrated in EE2. When EE2 DIMD

flag is equal to true, the proposed TIMD flag is not signalled and set equal to false.

Similar to PDPC, Gradient PDPC is also included in the derivation of the TIMD mode.

When secondary MPM is enabled, both the primary MPMs and the secondary MPMs are used to derive the TIMD mode.

- 5 6-tap interpolation filter is not used in the derivation of the TIMD mode.

2.18.4. Modification of MPM list construction in the derivation of TIMD mode

- During the construction of MPM list, intra prediction mode of a neighbouring block is derived as Planar when it is inter-coded. To improve the accuracy of MPM list, when a neighbouring block is inter-coded, a propagated intra prediction mode is derived using the motion vector and reference picture and used in the construction of MPM list. This modification is only applied to the derivation of the TIMD mode.
- 10

2.18.5. TIMD with fusion

- Instead of selecting the only one mode with the smallest SATD cost, this contribution proposes to choose the first two modes with the smallest SATD costs for the intra modes derived using TIMD method and then fuse them with the weights, and such weighted intra prediction is used to code the current CU.
- 15

The costs of the two selected modes are compared with a threshold, in the test the cost factor of 2 is applied as follows:

$$\text{costMode2} < 2 \times \text{costMode1}.$$

- 20 If this condition is true, the fusion is applied, otherwise the only model1 is used.

Weights of the modes are computed from their SATD costs as follows:

$$\begin{aligned} \text{weight1} &= \text{costMode2} / (\text{costMode1} + \text{costMode2}), \\ \text{weight2} &= 1 - \text{weight1}. \end{aligned}$$

2.19. Convolutional cross-component model (CCCM) for intra prediction

- 25 It is proposed to apply convolutional cross-component model (CCCM) to predict chroma samples from reconstructed luma samples in a similar spirit as done by the current CCLM modes. As with CCLM, the reconstructed luma samples are down-sampled to match the lower resolution chroma grid when chroma sub-sampling is used.

- Also, similarly to CCLM, there is an option of using a single model or multi-model variant of CCCM. The multi-model variant uses two models, one model derived for samples above the average luma reference value and another model for the rest of the samples (following the spirit of the CCLM design). Multi-model CCCM mode can be selected for PUs which have at least 128 reference samples available.
- 30

2.19.1. Convolutional filter

The proposed convolutional 7-tap filter consist of a 5-tap plus sign shape spatial component, a nonlinear term and a bias term. The input to the spatial 5-tap component of the filter consists of a center (C) luma sample which is collocated with the chroma sample to be predicted and its
 5 above/north (N), below/south (S), left/west (W) and right/east (E) neighbors as illustrated below in Fig. 24, which illustrates spatial part of the convolutional filter.

The nonlinear term P is represented as power of two of the center luma sample C and scaled to the sample value range of the content:

$$P = (C * C + \text{midVal}) \gg \text{bitDepth}.$$

10 That is, for 10-bit content it is calculated as:

$$P = (C * C + 512) \gg 10.$$

The bias term B represents a scalar offset between the input and output (similarly to the offset term in CCLM) and is set to middle chroma value (512 for 10-bit content).

Output of the filter is calculated as a convolution between the filter coefficients c_i and the input
 15 values and clipped to the range of valid chroma samples:

$$\text{predChromaVal} = c_0C + c_1N + c_2S + c_3E + c_4W + c_5P + c_6B.$$

2.19.2. Calculation of filter coefficients

The filter coefficients c_i are calculated by minimising MSE between predicted and reconstructed chroma samples in the reference area. Fig. 25 is a schematic diagram illustrating
 20 reference area (with its paddings) used to derive the filter coefficients. Fig. 25 illustrates the reference area which consists of 6 lines of chroma samples above and left of the PU. Reference area extends one PU width to the right and one PU height below the PU boundaries. Area is adjusted to include only available samples. The extensions to the area shown in blue are needed to support the “side samples” of the plus shaped spatial filter and are padded when in
 25 unavailable areas.

The MSE minimization is performed by calculating autocorrelation matrix for the luma input and a cross-correlation vector between the luma input and chroma output. Autocorrelation matrix is LDL decomposed and the final filter coefficients are calculated using back-substitution. The process follows roughly the calculation of the ALF filter coefficients in ECM,
 30 however LDL decomposition was chosen instead of Cholesky decomposition to avoid using square root operations. The proposed approach uses only integer arithmetic.

2.19.3. Bitstream signalling

Usage of the mode is signalled with a CABAC coded PU level flag. One new CABAC context was included to support this. When it comes to signalling, CCCM is considered a sub-mode of

CCLM. That is, the CCCM flag is only signalled if intra prediction mode is LM_CHROMA_IDX (to enable single mode CCCM) or MMLM_CHROMA_IDX (to enable multi-model CCCM).

2.20. Gradient Linear Model (GLM)

- 5 Compared with the CCLM, instead of down-sampled luma values, the GLM utilizes luma sample gradients to derive the linear model. Specifically, when the GLM is applied, the input to the CCLM process, i.e., the down-sampled luma samples L , are replaced by luma sample gradients G . The other parts of the CCLM (e.g., parameter derivation, prediction sample linear transform) are kept unchanged.

$$10 \quad C = \alpha \cdot G + \beta.$$

For signaling, when the CCLM mode is enabled to the current CU, two flags are signaled separately for Cb and Cr components to indicate whether GLM is enabled to each component; if the GLM is enabled for one component, one syntax element is further signaled to select one of 4 gradient filters for the gradient calculation.

- 15 Four gradient filters are enabled for the GLM, as illustrated in Fig. 26.

GLM with Luma

In ECM-6.0, GLM utilizes the gradient of luma samples to predict a chroma sample as:

$$pred_c(i, j) = \alpha \cdot G(i, j) + \beta$$

where $pred_c(i, j)$ represents the predicted value of a chroma sample, $G(i, j)$ represents the gradient of the corresponding reconstructed luma samples, and the linear model parameters α and β are derived by adjacent reconstructed samples based on the linear minimum mean square error (LMMSE) method as CCLM.

- 20 and β are derived by adjacent reconstructed samples based on the linear minimum mean square error (LMMSE) method as CCLM.

A new GLM mode is proposed that a chroma sample is predicted based on both the gradient $G(i, j)$ of luma samples and the reconstructed value $rec'_L(i, j)$ of the down-sampled luma sample with different parameters:

$$25 \quad pred_c(i, j) = \alpha_0 \cdot G(i, j) + \alpha_1 \cdot rec'_L(i, j) + \alpha_2 \cdot midValue$$

where the model parameters α_0 , α_1 and α_2 are derived from six rows and columns adjacent samples based on the LDL decomposition method as the CCCM mode in ECM-6.0.

2.21. Gradient and location based convolutional cross-component model (GL-CCCM) for intra prediction

- 30 The proposed GL-CCCM method uses gradient and location information instead of the 4 spatial neighbor samples in the CCCM filter. The GL-CCCM filter for the prediction is:

$$predChromaVal = c_0C + c_1G_y + c_2G_x + c_3Y + c_4X + c_5P + c_6B.$$

Where G_y and G_x are the vertical and horizontal gradients, respectively, and are calculated as:

$$G_y = (2N + NW + NE) - (2S + SW + SE),$$

$$G_x = (2W + NW + SW) - (2E + NE + SE).$$

Moreover, the Y and X parameters are the vertical and horizontal locations of the center luma sample and they are calculated with respect to the top-left coordinates of the block.

The rest of the parameters are the same as CCCM tool. The reference area for the parameter calculation is the same as CCCM method.

Fig. 27 is a schematic diagram illustrating spatial samples used for GL-CCCM.

Bitstream signalling

Usage of the mode is signalled with a CABAC coded PU level flag. One new CABAC context was included to support this. When it comes to signalling, GL-CCCM is considered a sub-mode of CCCM. That is, the GL-CCCM flag is only signalled if original CCCM flag is true.

Encoder operation

The encoder performs two new RD checks in the chroma prediction mode loop, one for checking single model GL-CCCM mode and one for checking multi-model GL-CCCM mode.

2.22. CCCM using non-downsampled luma samples

2.22.1. Block level

In this contribution, the CCCM using non-downsampled luma samples is proposed where the chroma samples are directly predicted from the original reconstructed luma samples, i.e., without downsampling. Fig. 28 is a schematic diagram illustrating non-downsampled luma samples. As shown in Fig. 28, the proposed CCCM filter consists of 6-tap spatial terms, two nonlinear terms and a bias term. The 6-tap spatial terms correspond to 6 neighboring luma samples (i.e., $L_0, L_1, \dots, L_5$) to the chroma sample (i.e., C) to be predicted.

$$C = \sum_{i=0}^5 \alpha_i \cdot L_i + \sum_{i=6}^7 \alpha_i \cdot (((L_{i-4})^2 + \beta) \gg \text{bitDepth}) + \alpha_8 \cdot \beta$$

where α_i is the coefficient associated with L_i and β is the offset. Same to the existing CCCM design, up to 6 lines/columns of chroma samples above and left to the current CU are applied to derive the filter coefficients. The filter coefficients are derived based on the same LDL decomposition method used in CCCM. In the contribution, the proposed method is signaled as one extra CCCM model besides the existing CCCM model. For signaling, when the CCCM is selected, one single flag is signaled and used for both two chroma components to indicate whether the default CCCM model or the proposed CCCM model is applied.

2.22.2. High level control

Subsampling of luma component may not be optimal for CCCM model derivation for the content which has sharp details, such as SCC content. In this contribution it is proposed to

disable luma subsampling, derive and apply model on nonsubsampling luma samples directly. CCCM model shape is diamond 5×5 if subsampling is not applied. SPS flag is signalled to indicate whether luma subsampling is applied for CCCM.

2.23. Spatial GPM (SGPM)

- 5 In spatial GPM, a candidate list is built which includes partition split and two intra prediction modes. Up to 11 MPMs of intra prediction modes are used to form the combinations, the length of the candidate list is set equal to 16. The selected candidate index is signalled.

Fig. 29 illustrates spatial GPM candidates. The list is reordered using template shown in Fig. 29. GPM blending process is not used in the template, and SAD between the prediction and
10 reconstruction of the template is used for ordering.

The SGPM mode is applied to blocks whose width and height meet the same restrictions as in inter GPM.

Fig. 30 illustrates GPM templates.

The following items are considered:

- 15 ● Spatial GPM partition modes:
26 predefined modes.
Adaptive derivation algorithm based on the horizontal and vertical gradients ratio.
- Intra prediction mode selection:
IPM list with and without TIMD:
20 For each partition mode, an IPM list is derived for each part using intra-inter GPM list derivation. The IPM list size is 3. In the list, TIMD derived mode is replaced by 2 derived modes with horizontal and vertical orientations (using top or left templates) or TIMD derived mode is excluded.
- MPM list:
25 A uniform MPM list, up to 11 elements, is used for all partition modes.
- Template size (left and above): 1 or 4.
- Extended block size:
Spatial GPM is extended to be further applied to 4×8 , 8×4 , 4×16 and 16×4 blocks, which
30 can be described as $4 \leq \text{width} \leq 64$, $4 \leq \text{height} \leq 64$, $\text{width} < \text{height} * 8$, $\text{height} < \text{width} * 8$, $\text{width} * \text{height} \geq 32$.
- Adaptive blending:
Adaptive blending is tested for spatial GPM, where blending depth τ is derived as follows:
35 ■ If $\min(\text{width}, \text{height}) = 4$, $1/2 \tau$ is selected.

- else if $\min(\text{width}, \text{height}) == 8$, τ is selected.
- else if $\min(\text{width}, \text{height}) == 16$, 2τ is selected.
- else if $\min(\text{width}, \text{height}) == 32$, 4τ is selected.
- else, 8τ is selected.

5 Fig. 31 illustrates GPM blending.

2.24. Signaling of cross-component prediction modes in ECM

Fig. 32 illustrates binarization of cross-component prediction modes in ECM. “CCLM” in Fig. 32 may be replaced by “CCCM”.

10 In ECM-7, cross-components modes include CCLM, CCLM-L, CCLM-T, MM-CCLM, MM-CCLM-L, MM-CCLM-T, and CCCM, CCCM-L, CCCM-T, MM-CCCM, MM-CCCM-L, MM-CCCM-T. One flag is signaled to determine whether it is a kind of CCCM mode or a kind of CCLM mode. A truncated unary code is applied to indicate the CCLM mode or CCCM mode shown in Fig. 32.

CCLM or CCCM: 0.

15 MM-CCLM or MM-CCCM: 10.

CCLM-L or CCCM-L: 110.

CCLM-T or CCCM-T: 1110.

MM-CCLM-L or MM-CCCM-L: 11110.

MM-CCLM-T or MM-CCCM-T: 11110.

20 2.25. Slope adjustment for CCLM

CCLM uses a model with 2 parameters to map luma values to chroma values. The slope parameter “a” and the bias parameter “b” define the mapping as follows:

$$\text{chromaVal} = a * \text{lumaVal} + b.$$

25 It is proposed signal an adjustment “u” to the slope parameter to update the model to the following form:

$$\text{chromaVal} = a' * \text{lumaVal} + b'$$

where

$$a' = a + u,$$

$$b' = b - u * y_r.$$

30 With this selection the mapping function is tilted or rotated around the point with luminance value y_r . It is proposed to use the average of the reference luma samples used in the model creation as y_r in order to provide a meaningful modification to the model. Picture below illustrates the process.

2.26. Non-Adjacent cross-component prediction

Non-adjacent cross-component prediction (NA-CCP) mode is proposed. With NA-CCP mode, Samples in regions non-adjacent to the current block can be used to derive a CCCM model for the current block. A candidate region list with 6 candidates is constructed by checking potential 8×8 regions in order. If a checked region is available, it is put into the candidate region list. The top-left positions of the potential 8×8 regions are predetermined as $\{(-xStep, 0), (0, -yStep), (xStep, -yStep), (-xStep, yStep), (-xStep, -yStep), (-2 * xStep, 0), (0, -2 * yStep), (-2 * xStep, 2 * yStep), (2 * xStep, -2 * yStep), (-2 * xStep, yStep), (xStep, -2 * yStep), (-2 * xStep, -yStep), (-xStep, -2 * yStep), (-2 * xStep, -2 * yStep), (-xStep/2, 0), (0, -yStep/2), (xStep/2, -yStep/2), (-xStep/2, yStep/2), (-xStep/2, -yStep/2)\}$, where $xStep = \text{Max}(\text{width}, 16)$, $yStep = \text{Max}(\text{height}, 16)$. Fig. 33 shows some possible positions of candidate regions.

A flag is signaled to indicate whether NA-CCP is applied to a chroma block. If NA-CCP is applied, an index is signaled to indicate which candidate in the candidate region list is used to derive the CCCM model.

2.27. Fusion of chroma intra prediction modes

In the Test 1.2b, it is proposed that the DM mode and the four default modes can be fused with the MMLM_LT mode as follows:

$$pred = (w0 * pred0 + w1 * pred1 + (1 \ll (shift - 1))) \gg shift$$

where $pred0$ is the predictor obtained by applying the non-LM mode, $pred1$ is the predictor obtained by applying the MMLM_LT mode and $pred$ is the final predictor of the current chroma block. The two weights, $w0$ and $w1$ are determined by the intra prediction mode of adjacent chroma blocks and $shift$ is set equal to 2. Specifically, when the above and left adjacent blocks are both coded with LM modes, $\{w0, w1\} = \{1, 3\}$; when the above and left adjacent blocks are both coded with non-LM modes, $\{w0, w1\} = \{3, 1\}$; otherwise, $\{w0, w1\} = \{2, 2\}$.

For the syntax design, if a non-LM mode is selected, one flag is signaled to indicate whether the fusion is applied. And the proposed fusion is only applied to I slices.

3. Problems

1. The models of cross-component prediction are trained with adjacent neighbouring samples, which may not be efficient.

4. Detailed solutions

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

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

In the following discussion, CCCM may refer to the original CCCM mode, or it may refer to a variance of CCCM, such as CCCM-L, CCCM-T, MM-CCCM, MM-CCCM-L, MM-CCCM-T.

5 In the following discussion, CCLM may refer to the original CCLM mode, or it may refer to a variance of CCLM, such as CCLM-L, CCLM-T, MM-CCLM, MM-CCLM-L, MM-CCLM-T, etc.

In one following discussion, cross-component prediction (CCP) may refer to any cross-component prediction such as CCLM or CCCM or GLM or CCLM with slope adjustment.

10 Definition of History-based HT

1. It is proposed that the model(s) of cross-component prediction (CCP), such as CCLM or CCCM, in a block may be stored into a history table (HT).

a. A HT is a list with ordered entries.

i. Each entry has an index. For example, the index of the first entry is 0, and
15 indices of following entries are 1, 2, 3,...

b. Model parameters of CCLM and its variants may comprise a , b and a *shift* which controls the calculation precision.

c. Model parameters of CCLM and its variants may comprise linear parts such as $c_0 \sim c_4$ and nonlinear part such as c_5 .

20 d. Models may include models for different color components such as C_b and C_r .

i. For example, models for C_b and C_r may be coupled in an entry.

e. In one example, different CCPs like CCLM and CCCM may share the same HT.

i. In one example, a segment in an entry of the HT may reflect the type of CCP model(s) stored in the entry.

25 f. In one example, different CCPs like CCLM and CCCM may have different HTs.

i. In one example, one CCLM_HT may store models of CCLM and its variants like CCLM-L or CCLM-T.

ii. In one example, one CCCM_HT may store models of CCCM and its variants like CCCM-T or CCCM-T.

30 g. In one example, CCP with a single model (like CCLM or CCCM) and CCP with multiple models (like MM-CCLM or MM-CCCM) may have different HTs.

h. In one example, CCP with a single model (like CCLM or CCCM) and CCP with multiple models (like MM-CCLM or MM-CCCM) may share the same HT.

i. In one example, a segment in an entry of the HT may reflect the number
35 of models stored in the entry.

- ii. In one example, a segment in an entry of the HT may reflect at least one threshold used to classify samples into different groups of models.
- i. In one example, a first HT is used to store models of CCLM and its variants.
 - i. In one example, CCLM variants may comprise CCLM-L, CCLM-T, MM-CCLM, MM-CCLM-L, MM-CCLM-T, GLM and CCLM with slope adjustments.
 - 1) A segment in an entry of the HT may reflect the number of models stored in the entry.
 - 2) A segment in an entry of the HT may reflect at least one threshold used to classify samples into different groups of models.
 - 3) A segment in an entry of the HT may reflect whether GLM is applied.
 - 4) A segment in an entry of the HT may reflect the down-sampling filter of GLM.
- j. In one example, a second HT is used to store models of CCCM and its variants.
 - i. In one example, CCCM variants may comprise CCCM-L, CCCM-T, MM-CCCM, MM-CCCM-L, MM-CCCM-T.
 - 1) A segment in an entry of the HT may reflect the number of models stored in the entry.
 - 2) A segment in an entry of the HT may reflect at least one threshold used to classify samples into different groups of models.
- 2. It is proposed that a block can be coded with history-based CCP (H-CCP) mode, in which mode at least one CCP model used by the current block is fetched or derived from a HT.
 - a. In one example, at least one syntax element (SE) may be signaled to indicate whether H-CCP is applied.
 - i. In one example, the SE may be signaled conditionally. E.g. the SE is signaled only if a specific mode is used, such as CCCM or CCLM.
 - 1) For example, the SE is signaled only if the current mode is CCCM or CCLM.
 - b. In one example, at least one syntax element (SE) may be signaled to indicate which entry in a HT is fetched to derive the model(s) of cross-component prediction.
 - i. The SE may reflect an index in the HT.
 - 1) In one example, the SE may be set equal to $f(k)$ where k is an index and f is a function.

- 2) In one example, the SE may be set equal to $f(k, M)$ where k is an index, M is the number of valid entries in the HT and f is a function.
- a) In another example, M is the size of HT.
- 3) In one example, the SE may be set equal to k where k is an index.
- 4) In one example, the SE may be set equal to $M-1-k$ where k is an index and M is the number of valid entries in the HT.
- a) In another example, M is the size of HT.
- ii. The SE may reflect an index of a list and the list may be constructed based on the HT.
- 1) In one example, the list L is constructed by reversing the HT. For example, $L[i]=HT[M-1-i]$, wherein M is the number of valid entries in the HT.
- a) In another example, M is the size of HT.
- b) In one example, L may have a fixed size.
- c) In one example, if L is not full, the vacant entries are filled with default entries.
- iii. In one example, the SE may be signaled conditionally. E.g. the SE is signaled only if H-CCP is applicable.
- iv. The SE may be signaled only if more than one entry in the HT can be selected.
- v. The maximum value (denoted as V) of the SE is determined by the number of entries to be selected.
- 1) For example, $V = K$, or $V = K-1$, or $V=K+1$, or $V=K-2$, or $V=K+2$.
- c. In one example, at least one syntax element (SE) may be signaled to indicate which HT is used.
- i. In one example, the SE may be signaled conditionally. E.g. the SE is signaled only if H-CCP is applicable.
- ii. The SE may be signaled only if more than one HTs can be selected.
- d. In one example, it may be derived at encoder/decoder which HT is used.
- i. In one example, if the current mode is CCLM, a first HT storing models of CCLM and its variants is used.
- ii. In one example, if the current mode is CCCM, a second HT storing models of CCCM and its variants is used.

- e. In one example, the current block may be predicted with the CCP model fetched from the determined entry of the determined HT.
- f. In one example, the current block may be predicted with either CCCM or CCLM based on whether the first HT or the second HT is applied.
- 5 g. In one example, the current block may be predicted with multiple models.
 - i. Whether single model or multiple models are applied may be derived/fetched from the determined entry of the determined HT.
 - ii. At least one threshold used to classify samples into different groups of models may be fetched/derived from the determined entry of the
 - 10 determined HT.

Maintenance of the HT

- 3. The maximum size of a HT may be predetermined, such as to be 5 or 6.
 - a. Alternatively, the maximum size of a HT may be signaled as a SE at block level/sequence level/group of pictures level/picture level/slice level/tile group level,
 - 15 such as in coding structures of CTU/CU/TU/PU/CTB/CB/TB/PB, or sequence header/picture header/SPS/VPS/DPS/DCI/PPS/APS/slice header/tile group header.
 - b. Alternatively, the maximum size of a HT may be derived using coding/decoding information such as
 - 20 i. The mode of the current block;
 - ii. The mode of a neighbouring block;
 - iii. The mode of a luma block in the collocated region of the current block;
 - iv. The mode of a luma block in the collocated region of a neighbouring block;
 - 25 v. QP;
 - vi. Slice/picture type;
 - vii. Picture width/height;
 - viii. Block width/height;
 - ix. Reconstructed samples.
- 30 4. A HT may be refreshed at the beginning of encoding/decoding a sequence/picture/slice/tile/sub-picture/CTU row/CTU.
 - a. For example, a HT may be refreshed by emptying the table.
 - b. For example, a HT may be refreshed by fulfilling the table with default entries.

5. After encoding/decoding a block (such as a CU), a HT may be updated.
- a. For example, the CU must be a chroma CU when dual-tree coding is applied.
 - b. For example, the CU must be a CU with CCP modes.
 - c. For example, which HT to be updated may depend on the coding mode of the
5 CU.
 - i. For example, if the CU is coded with a CCLM mode (such as CCLM, CCLM-L, CCLM-T, MM-CCLM, MM-CCLM-L, MM-CCLM-T, GLM and CCLM with slope adjustments), the model(s) and related information (such as threshold(s) used to classify samples into different groups of
10 models) are stored in the first HT.
 - ii. For example, if the CU is coded with a CCCM mode (such as CCCM, CCCM-L, CCCM-T, MM-CCCM, MM-CCCM-L, MM-CCCM-T), the model(s) and related information (such as threshold(s) used to classify samples into different groups of models) are stored in the first HT.
 - d. For example, a set of information related to the CCP model(s) used by the current
15 block may be put into the HT.
 - i. The set may comprise one or multiple CCP models.
 - ii. The set may comprise the number of models.
 - iii. The set may comprise threshold(s) used to classify samples into different
20 groups of models).
 - iv. The set may comprise slope adjustments.
 - e. In one example, the CCP model may be adjusted before being used to update the HT, if the current block is coded with CCLM with slope adjustments.
6. How to put a new set of information related to the CCP model(s) into a HT may depend
25 on whether the HT is full.
- a. For example, if the HT is not full, the new set may be put to the first vacant entry of the HT.
 - i. For example, the first vacant entry is the vacant entry with the smallest
30 index.
 - ii. For example, the first vacant entry is the vacant entry with the largest index.
 - iii. After being put into the HT, the new set may be put as the last occupied entry in the HT.

- 1) The last occupied entry may be the occupied entry with the largest index.
 - 2) The last occupied entry may be the occupied entry with the smallest index.
- 5 b. For example, if the HT is full, one existing entry in the HT may be removed.
- i. In one example, the HT may be managed in a First In First Out way.
 - ii. The existing entry with the smallest index may be removed.
 - 1) The updated HT' may be set as: $HT'[i]=HT[i+1]$, for $0 \leq i \leq N-2$, and $HT'[N-1]=$ new set, wherein N is the size of HT.
 - 10 iii. The existing entry with the largest index may be removed.
 - 1) The updated HT' may be set as: $HT'[i]=HT[i-1]$, for $1 \leq i \leq N-1$, and $HT'[0]=$ new set, wherein N is the size of HT.
7. In one example, the new set may be compared with at least one of the existing entries in the HT to determine whether to put into the new set and/or how to update the HT.
- 15 8. In one example, if the new set is the same or similar to one of the existing entries in the HT, the new set is not put into the HT. Suppose the new set is the same or similar to a special entry of HT.
- a. For example, in such a case, the special entry may be put to the first of the HT, and the entries originally before the special entry are pushed one position

20 backward.

 - i. For example, suppose entries are $HT[i]$ with $i = 0, 1, \dots$, and the special entry is $HT[k]$, then the updated HT' will be as: $HT'[0] = HT[k]$; $HT'[i]=HT[i-1]$ for $1 \leq i \leq k$; $HT'[i]=HT[i]$ for $i > k$.
 - b. For example, in such a case, the special entry may be put to the end of the HT,

25 and the entries originally before the special entry are pushed one position forward.

 - i. For example, suppose entries are $HT[i]$ with $i = 0, 1, \dots$, and the special entry is $HT[k]$, then the updated HT' will be as: $HT'[N-1] = HT[k]$; $HT'[i]=HT[i+1]$ for $k \leq i \leq N-2$; $HT'[i]=HT[i]$ for $i < k$.
9. In one example, whether to put into the new set and/or how to update the HT may depend

30 on the coding information of the CU with the new set.

10. In one example, if the new set is of a CU coded with H-CCP mode, the new set is not put into the HT. Suppose a special entry in HT is used by the CU coded with H-CCP.

- a. For example, in such a case, the special entry may be put to the first of the HT, and the entries originally before the special entry are pushed one position backward.

- i. For example, suppose entries are HT[i] with $i = 0, 1, \dots$, and the special entry is HT[k], then the updated HT' will be as: HT'[0] = HT[k]; HT'[i]=HT[i-1] for $1 \leq i \leq k$; HT'[i]=HT[i] for $i > k$.

- b. For example, in such a case, the special entry may be put to the end of the HT, and the entries originally before the special entry are pushed one position forward.

- i. For example, suppose entries are HT[i] with $i = 0, 1, \dots$, and the special entry is HT[k], then the updated HT' will be as: HT'[N-1] = HT[k]; HT'[i]=HT[i+1] for $k \leq i \leq N-2$; HT'[i]=HT[i] for $i < k$.

- 11. It is proposed that an entry of HT may include models for more than one chroma components, such as Cb and Cr.

- a. If an entry is selected, then the models for component Cb and Cr are applied on the two components respectively.

- 12. It is proposed that an entry of HT may include models for only one component, such as Cb or Cr.

- a. If an entry is selected, then the model for the specific component such as Cb or Cr is applied on the specific component.

- b. In one example, different HT may be built for different components.

List mode

- 13. It is proposed that at least one list with CCP models may be constructed.

- a. In one example, a chroma block may be predicted with a CCP model in the list, with a "list mode".

- b. In one example, the list L may be filled with one type of CCP models, such as CCCM.

- c. In one example, the list may be filled with multiple types of CCP models, such as both CCCM and CCLM.

- i. In one example, the type of the CCP model will be stored in the list together with the CCP model.

- d. In one example, at least one syntax element (SE) may be signaled to indicate whether a CCP model in the list is used.

- i. In one example, the SE may be signaled conditionally. E.g. the SE is signaled only if a specific mode is used, such as CCCM or CCLM.
 - 1) For example, the SE is signaled only if the current mode is CCCM or CCLM.
 - 2) For example, the SE is signaled only if the “list mode” is applicable.
 - e. In one example, at least one syntax element (SE) may be signaled to indicate which entry in the list is used to derive the model(s) of cross-component prediction.
 - i. The SE may reflect an index in the list.
 - 1) In one example, the SE may be set equal to $f(k)$ where k is an index and f is a function.
 - 2) In one example, the SE may be set equal to $f(k, M)$ where k is an index, M is the number of valid entries in the list and f is a function.
 - a) In another example, M is the size of list.
 - 3) In one example, the SE may be set equal to k where k is an index.
 - 4) In one example, the SE may be set equal to $M-1-k$ where k is an index and M is the number of valid entries in the list.
 - a) In another example, M is the size of list.
 - f. In one example, L may have a fixed size.
 - g. In one example, multiple lists may be constructed.
 - i. For example, at least one syntax element (SE) may be signaled to indicate which list is used.
 - ii. In one example, the SE may be signaled conditionally. E.g. the SE is signaled only if “list mode” is applicable.
 - iii. The SE may be signaled only if more than one list can be selected.
 - h. In one example, it may be derived at encoder/decoder which list is used.
 - i. In one example, if the current mode is CCLM, a first list storing models of CCLM and its variants is used.
 - ii. In one example, if the current mode is CCCM, a second list storing models of CCCM and its variants is used.
14. It is proposed that an entry of list may include models for more than one chroma components, such as Cb and Cr.
- a. If an entry is selected, then the models for component Cb and Cr are applied on the two components respectively.

15. It is proposed that an entry of list may include models for only one component, such as Cb or Cr.

- a. If an entry is selected, then the model for the specific component such as Cb or Cr is applied on the specific component.

5 16. Multiple candidates may be put into the list, including.

- a. A CCP model of an adjacent neighbouring block.
- b. A CCP model of a non-adjacent neighbouring block.
- c. A CCP model of a collocated block in a reference picture.
- d. A CCP model of a reference block in a reference picture.
- 10 e. A CCP model in a history table.
- f. A CCP model derived from non-adjacent samples.
- g. A default CCP mode.

17. In one example, a list may be constructed by checking possible candidates in an order.

- a. For example, the order may be adjacent neighbouring blocks, non-adjacent
15 neighbouring blocks, models in a history table, models derived from non-adjacent samples.
- b. For example, the list construction is finished if the number of candidates in the list achieves the maximum allowed size of the list (such as 5 or 6).
- c. For example, the list construction is finished if the number of candidates in the
20 list achieves $f(d)$, where d is the index of the selected candidate and f is a function. For example, $f(d)=d+1$.
- d. For example, default models may be put into the list if all possible candidates have been checked the construction is not finished.

18. In one example, if a potential candidate is put into the list, it may be compared with at
25 least one existing candidate in the list.

- a. For example, the potential candidate is not put into the list, if it is the same or similar to the existing candidate.
- b. In one example, if a potential entry of CCP information is put into the history-based table, it may be compared with at least one existing entries in the list.
30
 - i. For example, the potential entry is not put into the list, if it is the same or similar to an existing entry.
- c. In one example, two CCP candidates or entries are determined NOT to be the same if:
 - i. The CCP types are different.

- ii. The numbers of models are different.
- iii. The thresholds are different if the CCP has multiple models.
- iv. At least one model is different.
- v. The luma sample offset is different. (maybe only applicable if the type is CCCM or GL-CCCM or GLM or CCCM with using non-downsampled luma samples).
- vi. The sample location shifts are different. (maybe only applicable if the type is GL-CCCM).

19. For example, CCP information of an entry in the history-based table or of a candidate in a CCP candidate list may comprise:

- a. The type of the CCP method, such as CCLM or CCCM or GLM or GLM with luma or GL-CCCM or CCCM using non-downsampled luma samples.
 - i. In one example, GLM method using different down-sampling filters may be considered as different types.
 - ii. In one example, GLM with luma method using different down-sampling filters may be considered as different types.
 - iii. In one example, the types may be CCCM, CCLM, 4 types of GLM using different down-sampling filters, 4 types of GLM with luma using different down-sampling filters, GL-CCCM and CCCM using non-downsampled luma samples.
 - iv. "Not coded with CCP" (denoted as NonCCP) may also be treated as a type.
- b. The position (x, y).
- c. The number of models.
 - i. For example, the number of models may be 1 or 2.
 - ii. In one example, the number of models may be considered as a part of the CCP type. For example, CCLM and MM-CCLM may be considered as two types.
- d. At least one threshold to classify samples for different models.
 - i. The threshold may be used only if the number of models is at least 2.
- e. At least one luma sample value offset.
 - i. The luma sample value offset may be added to or subtracted from a luma sample (which may be down sampled) when it is used to derive a chroma prediction value.

- ii. The luma sample value offset may be used only for specific types such as CCCM, GLM with luma, GL-CCCM and CCCM using non-down-sample luma samples.
 - f. At least one chroma sample value offset.
 - 5 i. The chroma sample value offset may be added to or subtracted from a chroma prediction value derived by a CCP model to generate the final prediction.
 - g. At least one models for at least one chroma component.
 - i. For example, it may include different models for Cb and Cr components.
 - 10 ii. For example, the number of models for each component may be included as a part of the information.
 - iii. The model may be represented by the model form of CCLM or CCCM or GLM or GLM with luma or GL-CCCM or CCCM using non-downsampled luma samples.
 - 15 h. At least one sample location shift denoted as (dX, dY).
 - i. The chroma sample location shift may be added to or subtracted from the sample location (x, y) when it is used to derive the chroma prediction value.
 - ii. The chroma sample location shift may be used only for specific types
 - 20 such as GL-CCCM.
- 20. For example, the CCP coding information of a chroma block after being coded/decoded may be stored in the history-based table or in the CCP candidate list.
 - a. In one example, the CCP coding information may be stored only if the chroma block is coded with a CCP mode.
 - 25 i. In one example, the CCP coding information may be stored if the chroma block is coded with at least one CCP mode, such as with the fusion of chroma intra prediction mode.
 - 1) The stored type may be set to be the CCP type used in the fusion of chroma intra prediction mode.
 - 30 b. In one example, the CCP coding information may be stored for any chroma block.
 - i. If the chroma block is not coded with a CCP mode, the type is stored as “NonCCP”.
 - c. If the chroma block is coded with a CCP mode, the type of information may be stored as depending on the coding mode.

- i. The type is set to be “CCCM” if the mode is CCCM, or CCCM-T, or CCCM-L, or MM-CCCM, or MM-CCCM-T, or MM-CCCM-L.
- ii. The type is set to be “CCLM” if the mode is CCLM, or CCLM-T, or CCLM-L, or MM-CCLM, or MM-CCLM-T, or MM-CCLM-L.
- 5 iii. The type is set to be “CCLM” if the mode is CCLM, or CCLM-T, or CCLM-L, or MM-CCLM, or MM-CCLM-T, or MM-CCLM-L, with slope adjustments.
- iv. The type is set to be “GLM using filter X” if the mode is GLM using filter X.
- 10 v. The type is set to be “GLM with luma using filter X” if the mode is GLM with luma using filter X.
- vi. The type is set to be “GL-CCCM” if the mode is GL-CCCM.
- vii. The type is set to be “CCCM using non-down-sample” if the mode is CCCM using non-down-sample.
- 15 viii. The type is set to be “CCLM” if the mode is the fusion of chroma intra prediction mode.
- d. The number of models may be stored as the number of models of the chroma block.
 - i. For example, the number of models is set to be 2 if the mode is MM-CCLM, or MM-CCLM-T, or MM-CCLM-L, or MM-CCLM, or MM-CCLM-T, or MM-CCLM-L or any other multi-model CCP modes (such as GLM or GL-CCCM or CCCM using non-downsampled luma samples with multi-models).
- 20 e. Information such as the threshold, the luma/chroma sample value offset, sample location shift may be stored as the information used by the chroma block.
- 25 f. The CCP model of one component may be stored as the model used by the chroma block.
 - i. The model may be derived by any CCP method such as CCLM, or CCLM-T, or CCLM-L, or MM-CCLM, or MM-CCLM-T, or MM-CCLM-L or CCCM, or CCCM-T, or CCCM-L, or MM-CCCM, or MM-CCCM-T, or MM-CCCM-L or GLM using different down-sampling filters, or GLM with luma using different down-sampling filters, or GL-CCCM or CCCM using non-downsampled luma samples.
- 30

- ii. The stored model may be the final applied one, such as the one after been modified by the slope adjustment.

21. In one example, when CCP information of a CCP candidate in the candidate list or of an entry in the history-based table is used to generate prediction for the current block, the CCP will be performed following the CCP information.

- a. CCCM, CCLM, 4 types of GLM using different down-sampling filters, 4 types of GLM with luma using different down-sampling filters, GL-CCCM and CCCM using non-downsampled luma samples may be applied to the current block, based on the CCP type of the candidate.

- b. One model or multiple models with at least one threshold may be used, based on the model number and thresholds of the candidate.

- c. The luma sample value offset of the candidate may be added to or subtracted from the luma samples (which may be down-sampled) to be put into the CCP model.

- i. The process may be only applicable if the type is CCCM or GL-CCCM or GLM or CCCM with using non-downsampled luma samples.

- d. The sample location shifts may be added to or subtracted from the location coordinator to be put into the CCP model.

- i. The process may be only applicable if the type is GL-CCCM.

- e. How to get down-sampled luma samples may be based on the CCP type.

- i. The down-sampled luma samples may be obtained following the down-sampling method required by the CCP mode corresponding to the type.

22. In example, the prediction value generated by a CCP candidate list or an entry in the history-based table may be modified before being used to obtain the reconstruction sample value.

- a. In one example, an offset D may be added to or subtracted from the prediction value.

- b. In one example, the offset may be derived based on luma/chroma samples of a template, which is calculated using reconstructed samples neighbouring to the current block, known as a “template”. Fig. 34A to Fig. 34 C show examples of a template.

- i. In one example, the template may consist of reconstructed samples left to the current block, if reconstructed samples left to the current block are available.

- ii. In one example, the template may consist of reconstructed samples above to the current block, if reconstructed samples above to the current block are available.
- iii. In one example, the template may consist of reconstructed samples above or left to the current block, if reconstructed samples above/left to the current block are available.
- iv. Corresponding luma samples of the template may be down-sampled with the same manner as luma samples inside the current block.
- c. In one example, if there are N models (such as two models) required by the CCP type, N offsets denoted as $\{D^0, \dots, D^{N-1}\}$ may be derived for the N models.
- i. Offset D^i may be added to or subtracted from the prediction value generated by model i.
- d. In one example, the CCP method indicated by the type of the CCP candidate may be applied on the template.
- i. For example, for the k-th sample of the template, $S_k = R_k - P_k$ is calculated, where R_k and P_k represent the reconstructed sample value and the prediction value with CCP of the k-th sample, respectively, is calculated.
- 1) For example, D is calculated as the average value of $\{S_k\}$.
- 2) For example, suppose the number of S_k is M, D is calculated as $D = \text{sign}(\text{sum}) \times ((|\text{sum}| + \text{off}) \gg W)$, where $\text{sum} = \sum_{k=0}^{M-1} S_k$ and $W = \lfloor \log_2 |M| \rfloor$.
- ii. For example, for the k-th sample using model i of the template, $S_k^i = R_k^i - P_k^i$ is calculated, where R_k^i and P_k^i represent the reconstructed sample value and the prediction value with CCP of the k-th sample using model i, respectively, is calculated.
- 1) For example, D^i is calculated as the average value of $\{S_k^i\}$.
- 2) For example, suppose the number of S_k^i is M, D is calculated as $D^i = \text{sign}(\text{sum}) \times ((|\text{sum}| + \text{off}) \gg W)$, where $\text{sum} = \sum_{k=0}^{M-1} S_k^i$ and $W = \lfloor \log_2 |M| \rfloor$.
- iii. In one example, no division operation is used to calculate D or D^i .
- 1) For example, a lookup table may be used to calculate D or D^i .
- e. For example, only specific types of CCP may apply the modifications, such as CCLM and CMMM with multiple models.

23. In one example, a history table of CCP information after coding/decoding a region (such as a CU/CTU/CTU line) may be stored, known as a stored table.

a. The history table of CCP information maintained for the current block (known as an online table) may be used together with the stored history table of CCP information.

b. In one example, entries in a stored table and in an on-line table may be checked in an order to generate new candidates.

i. In one example, entries in the on-line table may be checked before all entries in the stored table.

ii. In one example, entries in the stored table may be checked before all entries in the on-line table.

iii. For example, k-th entry in the stored table may be checked after the k-th entry in the on-line table.

iv. For example, k-th entry in the on-line table may be checked after the k-th entry in the stored table.

v. For example, k-th entry in the on-line table may be checked after all the m-th entries, in the stored table, for $m = 0 \dots S$ where S is an integer.

vi. For example, k-th entry in the stored table may be checked after all the m-th entries, in the on-line table, for $m = 0 \dots S$ where S is an integer.

vii. For example, k-th entry in the on-line table may be checked after all the m-th entries, in the stored table, for $m = S \dots \text{maxT}$, where S is an integer and maxT is the last entry.

viii. For example, k-th entry in the stored table may be checked after all the m-th entries, in the on-line table, for $m = S \dots \text{maxT}$, where S is an integer and maxT is the last entry.

c. In one example, which stored table(s) to be used may depend on the dimension and/or location of the current block.

i. For example, the table stored in the CTU above the current CTU may be used.

ii. For example, the table stored in the CTU left-above to the current CTU may be used.

iii. For example, the table stored in the CTU right-above to the current CTU may be used.

d. In one example, whether to and/or how to use a stored table may depend on the dimension and/or location of the current block.

i. In one example, whether to and/or how to use a stored table may depend on whether the current CU is at the top boundary of a CTU and the above neighbouring CTU is available.

1) For example, a stored table may be used only if the current CU is at the top boundary of a CTU and the above neighbouring CTU is available.

2) For example, at least one entry in a stored table may be put to a more forward position if the current CU is at the top boundary of a CTU and the above neighbouring CTU is available.

e. In one example, entries in two stored tables may be checked in an order to generate new candidates.

i. For example, a first (or a second) stored table may be stored in the CTU above the current CTU may be used.

ii. For example, a first (or a second) stored table may be stored in the CTU left-above to the current CTU may be used.

iii. For example, a first (or a second) stored table may be stored in the CTU right-above to the current CTU may be used.

20 General claims

24. A syntax element disclosed above may be binarized as a flag, a fixed length code, an EG(x) code, a unary code, a truncated unary code, a truncated binary code, etc. It can be signed or unsigned.

25. A syntax element disclosed above may be coded with at least one context model. Or it may be bypass coded.

26. A syntax element disclosed above may be signaled in a conditional way.

a. The SE is signaled only if the corresponding function is applicable.

b. The SE is signaled only if the dimensions (width and/or height) of the block satisfy a condition.

27. A syntax element disclosed above may be signaled at block level/ sequence level/group of pictures level/picture level/slice level/tile group level, such as in coding structures of CTU/CU/TU/PU/CTB/CB/TB/PB, or sequence header/picture header/SPS/VPS/DPS/DCI/PPS/APS/slice header/tile group header.

28. Whether to and/or how to apply the disclosed methods above may be signalled at block level/ sequence level/group of pictures level/picture level/slice level/tile group level, such as in coding structures of CTU/CU/TU/PU/CTB/CB/TB/PB, or sequence header/picture header/SPS/VPS/DPS/DCI/PPS/APS/slice header/tile group header.

5 29. 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.

30. The proposed methods disclosed in this document may be used in other coding tools which require chroma fusion.

10 **[0095]** Fig. 35 illustrates a flowchart of a method 3500 for video processing in accordance with embodiments of the present disclosure. The method 3500 is implemented during a conversion between a current video block of a video and a bitstream of the video. In some embodiments, the conversion may include encoding the current video block into the bitstream. Alternatively, or in addition, the conversion may include decoding the
15 current video block from the bitstream.

[0096] At block 3510, at least one cross component prediction (CCP) model for the current video block is determined based on at least one history table (HT) of CCP models.

[0097] At block 3520, the conversion is performed based on the at least one CCP model. The at least one history table of CCP models is updated before or after coding a video
20 unit.

[0098] The method 3500 enables updating the history table of CCP models. In this way, the CCP model used for coding a video block can be updated without training based on neighboring samples. The coding efficiency and coding effectiveness can thus be improved.

25 **[0099]** In some embodiments, the at least one CCP model comprises at least one of: a cross-component linear model (CCLM), a variant of CCLM, a convolutional cross-component model (CCCM), or a variant of CCCM.

[0100] In some embodiments, the at least one history table is updated at a beginning of coding the video unit comprising the current video block, the video unit comprising at
30 least one of: a sequence, a picture, a slice, a tile, a sub-picture, a coding tree unit (CTU), or a CTU row. For example, a HT may be refreshed at the beginning of encoding/decoding

a sequence/picture/slice/tile/sub-picture/CTU row/CTU.

[0101] In some embodiments, updating the at least one history table comprises one of: emptying the at least one history table; or filling the at least one history table with default entries.

- 5 **[0102]** In some embodiments, the at least one history table is updated after coding the video unit, the video unit comprising a coding unit (CU). That is, after encoding/decoding a block (such as a CU), a HT may be updated.

[0103] In some embodiments, a dual-tree coding is applied, and the coding unit comprises a chroma coding unit.

- 10 **[0104]** In some embodiments, the coding unit is coded with a CCP mode.

[0105] In some embodiments, a target history table to be updated of the at least one history table is based on a coding mode of the coding unit. For example, which HT to be updated may depend on the coding mode of the CU.

- [0106]** In some embodiments, the coding mode of the coding unit comprises at least one of: a cross-component linear model (CCLM), a CCLM based on top neighboring samples of the coding unit (CCLM-T), a CCLM based on left neighboring samples of the coding unit (CCLM-L), a multi-model based CCLM (MM-CCLM), a multi-model based CCLM-T (MM-CCCM-T), multi-model based CCLM-L (MM-CCCM-L), a gradient linear model (GLM), or a CCLM with slope adjustment, and wherein a set of CCP models and related information for classifying samples into a plurality of groups of CCP models are stored in a first history table of the at least one history table, the first history table being associated with CCLM. For example, if the CU is coded with a CCLM mode (such as CCLM, CCLM-L, CCLM-T, MM-CCLM, MM-CCLM-L, MM-CCLM-T, GLM and CCLM with slope adjustments), the model(s) and related information (such as threshold(s) used to classify samples into different groups of models) are stored in the first HT.
- 15
20
25

- [0107]** In some embodiments, the coding mode of the coding unit comprises at least one of: a convolutional cross-component model (CCCM), a CCCM based on top neighboring samples of the current video block (CCCM-T), a CCCM based on left neighboring samples of the current video block (CCCM-L), a multi-model based CCCM (MM-CCCM), a multi-model based CCCM-T (MM-CCCM-T), or a multi-model based CCCM-L (MM-CCCM-L), and wherein a set of CCP models and related information for classifying samples into
- 30

a plurality of groups of CCP models are stored in a second history table of the at least one history table, the second history table being associated with CCCM. For example, if the CU is coded with a CCCM mode (such as CCCM, CCCM-L, CCCM-T, MM-CCCM, MM-CCCM-L, MM-CCCM-T), the model(s) and related information (such as threshold(s) used

5 to classify samples into different groups of models) are stored in the second HT.

[0108] In some embodiments, the coding unit is coded with a cross-component linear model (CCLM) with slope adjustment, and a corresponding CCP model is adjusted by the slope adjustment before being used to update the at least one history table.

[0109] In some embodiments, the at least one history table is updated by: adding a set
10 of information related to the at least one model used by the coding unit into the at least one history table. For example, a set of information related to the CCP model(s) used by the current block may be put into the HT.

[0110] In some embodiments, the set of information comprises at least one of: the at least one CCP model, the number of the at least one CCP model, a threshold for classifying
15 samples into a plurality of groups of CCP models, or slope adjustment information.

[0111] In some embodiments, an approach for adding the set of information related to the at least one CCP model into the history table is based on whether the history table is full.

[0112] In some embodiments, if the history table is not full, the set of information is
20 added to a first vacant entry of the history table.

[0113] In some embodiments, the first vacant entry comprises one of: a vacant entry with a smallest index in the history table, or a vacant entry with a largest index in the history table.

[0114] In some embodiments, the first vacant entry with the set of information is put as
25 a last occupied entry in the history table.

[0115] In some embodiments, the last occupied entry comprises one of: an occupied entry with a largest index in the history table, or an occupied entry with a smallest index in the history table.

[0116] In some embodiments, if the history table is full, an entry in the history table is
30 removed, and the history table is updated by adding the set of information to a further

vacant entry of the history table.

[0117] In some embodiments, the entry is with a smallest index in the history table.

[0118] In some embodiments, the updated history table is determined by $HT'[i]=HT[i+1]$, for $0 \leq i \leq N-2$, where HT denotes the history table, HT' denotes the
5 updated history table, $HT'[N-1]$ stores the set of information, and N denotes a size of the HT.

[0119] In some embodiments, the entry is with a largest index in the history table.

[0120] In some embodiments, the updated history table is determined by $HT'[i]=HT[i-1]$, for $1 \leq i \leq N-1$, where HT denotes the history table, HT' denotes the updated history
10 table, $HT'[0]$ stores the set of information, and N denotes a size of the HT.

[0121] In some embodiments, updating of the history table is based on a first in first out approach.

[0122] In some embodiments, at least one of whether to update the history table based on the set of information or how to update the history table is determined based on at least
15 one similarity between the set of information and at least one entry in the history table.

[0123] In some embodiments, if a similarity between the set of information and a target entry is less than or equal to a threshold, the set of information is not added into the history table.

[0124] In some embodiments, the target entry is moved to a beginning position of the
20 history table, and an entry in the history table before the target entry is moved one position backward.

[0125] In some embodiments, the history table is updated by: $HT'[0] = HT[k]$; $HT'[i]=HT[i-1]$ for $1 \leq i \leq k$; $HT'[i]=HT[i]$ for $i > k$, where HT denotes the history table, HT' denotes the updated history table, $HT[i]$ with $i = 0, 1 \dots$ denotes an entry in the history
25 table, $HT[k]$ denotes the target entry.

[0126] In some embodiments, the target entry is moved to an end position of the history table, and an entry in the history table before the target entry is moved one position forward.

[0127] In some embodiments, the history table is updated by $HT'[N-1] = HT[k]$; $HT'[i]=HT[i+1]$ for $k \leq i \leq N-2$; $HT'[i]=HT[i]$ for $i < k$, where HT denotes the history
30

table, HT' denotes the updated history table, HT[i] with $i = 0, 1 \dots$ denotes an entry in the history table, HT[k] denotes the target entry, and N denotes a size of the history table.

[0128] In some embodiments, at least one of whether to update the history table based on the set of information or how to update the history table is determined based on coding information of the coding unit with the set of information.

[0129] In some embodiments, the set of information is of the coding unit coded with history-based CCP (H-CCP) mode, the set of information is not added into the history table.

[0130] In some embodiments, a target entry in the history table is used by the coding unit, the target entry is moved to a beginning position of the history table, and an entry in the history table before the target entry is moved one position backward.

[0131] In some embodiments, the history table is updated by: $HT'[0] = HT[k]$; $HT'[i] = HT[i-1]$ for $1 \leq i \leq k$; $HT'[i] = HT[i]$ for $i > k$, where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1 \dots$ denotes an entry in the history table, HT[k] denotes the target entry.

[0132] In some embodiments, a target entry in the history table is used by the coding unit, the target entry is moved to an end position of the history table, and an entry in the history table before the target entry is moved one position forward.

[0133] In some embodiments, the history table is updated by $HT'[N-1] = HT[k]$; $HT'[i] = HT[i+1]$ for $k \leq i \leq N-2$; $HT'[i] = HT[i]$ for $i < k$, where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1 \dots$ denotes an entry in the history table, HT[k] denotes the target entry, and N denotes a size of the history table.

[0134] In some embodiments, a maximum size of a history table of the at least one history table is predetermined. For example, the maximum size may be 5 or 6.

[0135] In some embodiments, a syntax element (SE) in the bitstream indicates a maximum size of a history table of the at least one history table.

[0136] In some embodiments, the syntax element is at one of: a block level, a sequence level, a group of pictures level, a picture level, a slice level, or a tile group level.

[0137] In some embodiments, the syntax element is in at least one of the following coding structures: a coding tree unit (CTU), a coding unit (CU), a transform unit (TU), a

prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform block (TB), a prediction block (PB), a sequence header, a picture header, a sequence parameter set (SPS), a Video Parameter Set (VPS), a decoded parameter set (DPS), Decoding Capability Information (DCI), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), a slice header or a tile group header.

[0138] In some embodiments, a maximum size of a history table of the at least one history table is determined based on coding information.

[0139] In some embodiments, the coding information comprises at least one of: a mode of the current video block, a mode of a neighbouring block of the current video block, a mode of a luma block in a collocated region of the current video block, a mode of a luma block in the collocated region of a neighbouring block of the current video block, a quantization parameter (QP), a slice type, a picture type, a width or a height of a picture, a width or a height of a block, or reconstructed samples of the current video block.

[0140] In some embodiments, an entry of the at least one history table includes CCP models for a plurality of chroma components.

[0141] In some embodiments, the plurality of chroma components comprises a first chroma component and a second chroma component. For example, an entry of HT may include models for more than one chroma components, such as Cb and Cr.

[0142] In some embodiments, the entry is selected for the current video block, and the first and second chroma components of the current video block is coded by corresponding CCP models in the entry. If an entry is selected, then the models for component Cb and Cr are applied on the two components respectively.

[0143] In some embodiments, an entry of the at least one history table includes a CCP model for a single color component.

[0144] In some embodiments, the single color component comprises a first chroma component or a second chroma component. If an entry is selected, then the model for the specific component such as Cb or Cr is applied on the specific component.

[0145] In some embodiments, the entry is selected for the current video block, and a corresponding color component of the current video block is coded by the CCP model in the entry.

[0146] In some embodiments, a plurality of history tables of CCP models are determined for a plurality of color components. b. In one example, different HT may be built for different components.

[0147] In some embodiments, a syntax element in the bitstream is binarized as at least one of: a flag, a fixed length code, an exponential Golomb(x) (EG(x)) code, a unary code, a truncated unary code, or a truncated binary code, the syntax element comprising an indication or a flag.

[0148] In some embodiments, the syntax element is signed or unsigned.

[0149] In some embodiments, a syntax element in the bitstream is coded with at least one context model, or bypass coded, the syntax element comprising an indication or a flag.

[0150] In some embodiments, the syntax element is included in the bitstream based on a condition that a function associated with the syntax element is applicable.

[0151] In some embodiments, the syntax element is at at least one of: a block level, a sequence level, a group of pictures level, a picture level, a slice level, or a tile group level.

[0152] In some embodiments, the syntax element is in at least one of the following coding structures: a coding tree unit (CTU), a coding unit (CU), a transform unit (TU), a prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform block (TB), a prediction block (PB), a sequence header, a picture header, a sequence parameter set (SPS), a Video Parameter Set (VPS), a decoded parameter set (DPS), Decoding Capability Information (DCI), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), a slice header or a tile group header.

[0153] In some embodiments, information regarding whether to and/or how to apply the method is included in the bitstream.

[0154] In some embodiments, the information is indicated at one of: a sequence level, a group of pictures level, a picture level, a slice level or a tile group level.

[0155] In some embodiments, the information is indicated in at least one of the following coding structures: a coding tree unit (CTU), a coding unit (CU), a transform unit (TU), a prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform block (TB), a prediction block (PB), a sequence header, a picture header, a sequence parameter set (SPS), a Video Parameter Set (VPS), a decoded parameter set

(DPS), Decoding Capability Information (DCI), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), a slice header or a tile group header.

[0156] In some embodiments, the information is based on coded information.

[0157] In some embodiments, the coded information comprises at least one of: a block size, a colour format, a single or dual tree partitioning, a colour component, a slice type, or a picture type.

[0158] In some embodiments, the method is used in a coding tool requiring a chroma fusion.

[0159] 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. In the method, at least one cross component prediction (CCP) model for a current video block of the video is determined based on at least one history table (HT) of CCP models. The bitstream is generated based on the at least one CCP model. At least one history table of CCP models is updated before or after coding a video unit.

[0160] According to still further embodiments of the present disclosure, a method for storing bitstream of a video is provided. The method comprises: In the method, at least one cross component prediction (CCP) model for a current video block of the video is determined based on at least one history table (HT) of CCP models. The bitstream is generated based on the at least one CCP model. The bitstream is stored in a non-transitory computer-readable recording medium. At least one history table of CCP models is updated before or after coding a video unit.

[0161] 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.

[0162] Clause 1. A method for video processing, comprising: determining, for a conversion between a current video block of a video and a bitstream of the video, at least one cross component prediction (CCP) model for the current video block based on at least one history table (HT) of CCP models; and performing the conversion based on the at least one CCP model, wherein the at least one history table of CCP models is updated before or after coding a video unit.

[0163] Clause 2. The method of clause 1, wherein the at least one CCP model comprises at least one of: a cross-component linear model (CCLM), a variant of CCLM, a convolutional cross-component model (CCCM), or a variant of CCCM.

[0164] Clause 3. The method of clause 1 or 2, wherein the at least one history table is updated at a beginning of coding the video unit comprising the current video block, the video unit comprising at least one of: a sequence, a picture, a slice, a tile, a sub-picture, a coding tree unit (CTU), or a CTU row.

[0165] Clause 4. The method of clause 3, wherein updating the at least one history table comprises one of: emptying the at least one history table; or filling the at least one history table with default entries.

[0166] Clause 5. The method of clause 1 or 2, wherein the at least one history table is updated after coding the video unit, the video unit comprising a coding unit (CU).

[0167] Clause 6. The method of clause 5, wherein a dual-tree coding is applied, and the coding unit comprises a chroma coding unit.

[0168] Clause 7. The method of clause 5 or 6, wherein the coding unit is coded with a CCP mode.

[0169] Clause 8. The method of any of clauses 5-7, wherein a target history table to be updated of the at least one history table is based on a coding mode of the coding unit.

[0170] Clause 9. The method of clause 8, wherein the coding mode of the coding unit comprises at least one of: a cross-component linear model (CCLM), a CCLM based on top neighboring samples of the coding unit (CCLM-T), a CCLM based on left neighboring samples of the coding unit (CCLM-L), a multi-model based CCLM (MM-CCLM), a multi-model based CCLM-T (MM-CCCM-T), multi-model based CCLM-L (MM-CCCM-L), a gradient linear model (GLM), or a CCLM with slope adjustment, and wherein a set of CCP models and related information for classifying samples into a plurality of groups of CCP models are stored in a first history table of the at least one history table, the first history table being associated with CCLM.

[0171] Clause 10. The method of clause 8, wherein the coding mode of the coding unit comprises at least one of: a convolutional cross-component model (CCCM), a CCCM based on top neighboring samples of the current video block (CCCM-T), a CCCM based on left neighboring samples of the current video block (CCCM-L), a multi-model based

CCCM (MM-CCCM), a multi-model based CCCM-T (MM-CCCM-T), or a multi-model based CCCM-L (MM-CCCM-L), and wherein a set of CCP models and related information for classifying samples into a plurality of groups of CCP models are stored in a second history table of the at least one history table, the second history table being
5 associated with CCCM.

[0172] Clause 11. The method of any of clauses 5-10, wherein the coding unit is coded with a cross-component linear model (CCLM) with slope adjustment, and a corresponding CCP model is adjusted by the slope adjustment before being used to update the at least one history table.

10 **[0173]** Clause 12. The method of any of clauses 5-11, wherein the at least one history table is updated by: adding a set of information related to the at least one model used by the coding unit into the at least one history table.

[0174] Clause 13. The method of clause 12, wherein the set of information comprises at least one of: the at least one CCP model, the number of the at least one CCP model, a
15 threshold for classifying samples into a plurality of groups of CCP models, or slope adjustment information.

[0175] Clause 14. The method of clause 12 or 13, wherein an approach for adding the set of information related to the at least one CCP model into the history table is based on whether the history table is full.

20 **[0176]** Clause 15. The method of clause 14, wherein if the history table is not full, the set of information is added to a first vacant entry of the history table.

[0177] Clause 16. The method of clause 15, wherein the first vacant entry comprises one of: a vacant entry with a smallest index in the history table, or a vacant entry with a largest index in the history table.

25 **[0178]** Clause 17. The method of clause 15 or 16, wherein the first vacant entry with the set of information is put as a last occupied entry in the history table.

[0179] Clause 18. The method of clause 17, wherein the last occupied entry comprises one of: an occupied entry with a largest index in the history table, or an occupied entry with a smallest index in the history table.

30 **[0180]** Clause 19. The method of clause 14, wherein if the history table is full, an entry

in the history table is removed, and the history table is updated by adding the set of information to a further vacant entry of the history table.

[0181] Clause 20. The method of clause 19, wherein the entry is with a smallest index in the history table.

- 5 **[0182]** Clause 21. The method of clause 20, wherein the updated history table is determined by $HT'[i]=HT[i+1]$, for $0 \leq i \leq N-2$, where HT denotes the history table, HT' denotes the updated history table, HT'[N-1] stores the set of information, and N denotes a size of the HT.

- 10 **[0183]** Clause 22. The method of clause 19, wherein the entry is with a largest index in the history table.

[0184] Clause 23. The method of clause 29, wherein the updated history table is determined by $HT'[i]=HT[i-1]$, for $1 \leq i \leq N-1$, where HT denotes the history table, HT' denotes the updated history table, HT'[0] stores the set of information, and N denotes a size of the HT.

- 15 **[0185]** Clause 24. The method of clause 19, wherein updating of the history table is based on a first in first out approach.

- [0186]** Clause 25. The method of any of clauses 12-24, wherein at least one of whether to update the history table based on the set of information or how to update the history table is determined based on at least one similarity between the set of information and at
20 least one entry in the history table.

[0187] Clause 26. The method of clause 25, wherein if a similarity between the set of information and a target entry is less than or equal to a threshold, the set of information is not added into the history table.

- 25 **[0188]** Clause 27. The method of clause 26, wherein the target entry is moved to a beginning position of the history table, and an entry in the history table before the target entry is moved one position backward.

- [0189]** Clause 28. The method of clause 26 or 27, wherein the history table is updated by: $HT'[0] = HT[k]$; $HT'[i]=HT[i-1]$ for $1 \leq i \leq k$; $HT'[i]=HT[i]$ for $i > k$, where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1, \dots$
30 denotes an entry in the history table, HT[k] denotes the target entry.

[0190] Clause 29. The method of clause 26, wherein the target entry is moved to an end position of the history table, and an entry in the history table before the target entry is moved one position forward.

[0191] Clause 30. The method of clause 29, wherein the history table is updated by
 5 $HT'[N-1] = HT[k]$; $HT'[i] = HT[i+1]$ for $k \leq i \leq N-2$; $HT'[i] = HT[i]$ for $i < k$, where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1 \dots$ denotes an entry in the history table, HT[k] denotes the target entry, and N denotes a size of the history table.

[0192] Clause 31. The method of any of clauses 12-24, wherein at least one of whether
 10 to update the history table based on the set of information or how to update the history table is determined based on coding information of the coding unit with the set of information.

[0193] Clause 32. The method of clause 31, wherein the set of information is of the coding unit coded with history-based CCP (H-CCP) mode, the set of information is not
 15 added into the history table.

[0194] Clause 33. The method of clause 32, wherein a target entry in the history table is used by the coding unit, the target entry is moved to a beginning position of the history table, and an entry in the history table before the target entry is moved one position backward.

[0195] Clause 34. The method of clause 33, wherein the history table is updated by:
 20 $HT'[0] = HT[k]$; $HT'[i] = HT[i-1]$ for $1 \leq i \leq k$; $HT'[i] = HT[i]$ for $i > k$, where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1 \dots$ denotes an entry in the history table, HT[k] denotes the target entry.

[0196] Clause 35. The method of clause 32, wherein a target entry in the history table
 25 is used by the coding unit, the target entry is moved to an end position of the history table, and an entry in the history table before the target entry is moved one position forward.

[0197] Clause 36. The method of clause 35, wherein the history table is updated by
 30 $HT'[N-1] = HT[k]$; $HT'[i] = HT[i+1]$ for $k \leq i \leq N-2$; $HT'[i] = HT[i]$ for $i < k$, where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1 \dots$ denotes an entry in the history table, HT[k] denotes the target entry, and N denotes a size of the history table.

[0198] Clause 37. The method of any of clauses 1-36, wherein a maximum size of a history table of the at least one history table is predetermined.

[0199] Clause 38. The method of clause 37, wherein the maximum size is 5 or 6.

[0200] Clause 39. The method of any of clauses 1-36, wherein a syntax element (SE) in
5 the bitstream indicates a maximum size of a history table of the at least one history table.

[0201] Clause 40. The method of clause 39, wherein the syntax element is at one of: a block level, a sequence level, a group of pictures level, a picture level, a slice level, or a tile group level.

[0202] Clause 41. The method of clause 39 or 40, wherein the syntax element is in at
10 least one of the following coding structures: a coding tree unit (CTU), a coding unit (CU), a transform unit (TU), a prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform block (TB), a prediction block (PB), a sequence header, a picture header, a sequence parameter set (SPS), a Video Parameter Set (VPS), a decoded parameter set (DPS), Decoding Capability Information (DCI), a Picture Parameter Set (PPS), an
15 Adaptation Parameter Set (APS), a slice header or a tile group header.

[0203] Clause 42. The method of any of clauses 1-36, wherein a maximum size of a history table of the at least one history table is determined based on coding information.

[0204] Clause 43. The method of clause 42, wherein the coding information comprises at least one of: a mode of the current video block, a mode of a neighbouring block of the
20 current video block, a mode of a luma block in a collocated region of the current video block, a mode of a luma block in the collocated region of a neighbouring block of the current video block, a quantization parameter (QP), a slice type, a picture type, a width or a height of a picture, a width or a height of a block, or reconstructed samples of the current video block.

[0205] Clause 44. The method of any of clauses 1-43, wherein an entry of the at least
25 one history table includes CCP models for a plurality of chroma components.

[0206] Clause 45. The method of clause 44, wherein the plurality of chroma components comprises a first chroma component and a second chroma component.

[0207] Clause 46. The method of clause 45, wherein the entry is selected for the current
30 video block, and the first and second chroma components of the current video block is

coded by corresponding CCP models in the entry.

[0208] Clause 47. The method of any of clauses 1-43, wherein an entry of the at least one history table includes a CCP model for a single color component.

[0209] Clause 48. The method of clause 47, wherein the single color component
5 comprises a first chroma component or a second chroma component.

[0210] Clause 49. The method of clause 47, wherein the entry is selected for the current video block, and a corresponding color component of the current video block is coded by the CCP model in the entry.

[0211] Clause 50. The method of any of clauses 1-43, wherein a plurality of history
10 tables of CCP models are determined for a plurality of color components.

[0212] Clause 51. The method of any of clauses 1-50, wherein a syntax element in the bitstream is binarized as at least one of: a flag, a fixed length code, an exponential Golomb(x) (EG(x)) code, a unary code, a truncated unary code, or a truncated binary code, the syntax element comprising an indication or a flag.

[0213] Clause 52. The method of clause 51, wherein the syntax element is signed or
15 unsigned.

[0214] Clause 53. The method of any of clauses 1-52, wherein a syntax element in the bitstream is coded with at least one context model, or bypass coded, the syntax element comprising an indication or a flag.

[0215] Clause 54. The method of any of clauses 51-53, wherein the syntax element is
20 included in the bitstream based on a condition that a function associated with the syntax element is applicable.

[0216] Clause 55. The method of any of clauses 51-54, wherein the syntax element is
25 at at least one of: a block level, a sequence level, a group of pictures level, a picture level, a slice level, or a tile group level.

[0217] Clause 56. The method of any of clauses 51-55, wherein the syntax element is
in at least one of the following coding structures: a coding tree unit (CTU), a coding unit (CU), a transform unit (TU), a prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform block (TB), a prediction block (PB), a sequence header, a picture
30 header, a sequence parameter set (SPS), a Video Parameter Set (VPS), a decoded

parameter set (DPS), Decoding Capability Information (DCI), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), a slice header or a tile group header.

[0218] Clause 57. The method of any of clauses 1-56, wherein information regarding whether to and/or how to apply the method is included in the bitstream.

5 **[0219]** Clause 58. The method of clause 57, wherein the information is indicated at one of: a sequence level, a group of pictures level, a picture level, a slice level or a tile group level.

[0220] Clause 59. The method of clause 57 or 58, wherein the information is indicated in at least one of the following coding structures: a coding tree unit (CTU), a coding unit
10 (CU), a transform unit (TU), a prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform block (TB), a prediction block (PB), a sequence header, a picture header, a sequence parameter set (SPS), a Video Parameter Set (VPS), a decoded parameter set (DPS), Decoding Capability Information (DCI), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), a slice header or a tile group header.

15 **[0221]** Clause 60. The method of any of clauses 57-59, wherein the information is based on coded information.

[0222] Clause 61. The method of clause 60, wherein the coded information comprises at least one of: a block size, a colour format, a single or dual tree partitioning, a colour component, a slice type, or a picture type.

20 **[0223]** Clause 62. The method of any of clauses 1-61, wherein the method is used in a coding tool requiring a chroma fusion.

[0224] Clause 63. The method of any of clauses 1-62, wherein the conversion includes encoding the current video block into the bitstream.

[0225] Clause 64. The method of any of clauses 1-62, wherein the conversion includes
25 decoding the current video block from the bitstream.

[0226] Clause 65. 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 1-64.

30 **[0227]** Clause 66. A non-transitory computer-readable storage medium storing

instructions that cause a processor to perform a method in accordance with any of clauses 1-64.

[0228] Clause 67. 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 at least one cross component prediction (CCP) model for a current video block of the video based on at least one history table (HT) of CCP models; and generating the bitstream based on the at least one CCP model, wherein the at least one history table of CCP models is updated before or after coding a video unit.

[0229] Clause 68. A method for storing a bitstream of a video, comprising: determining at least one cross component prediction (CCP) model for a current video block of the video based on at least one history table (HT) of CCP models; generating the bitstream based on the at least one CCP model; and storing the bitstream in a non-transitory computer-readable recording medium, wherein the at least one history table of CCP models is updated before or after coding a video unit.

Example Device

[0230] Fig. 36 illustrates a block diagram of a computing device 3600 in which various embodiments of the present disclosure can be implemented. The computing device 3600 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).

[0231] It would be appreciated that the computing device 3600 shown in Fig. 36 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.

[0232] As shown in Fig. 36, the computing device 3600 includes a general-purpose computing device 3600. The computing device 3600 may at least comprise one or more processors or processing units 3610, a memory 3620, a storage unit 3630, one or more communication units 3640, one or more input devices 3650, and one or more output devices 3660.

[0233] In some embodiments, the computing device 3600 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 3600 can support any type of interface to a user (such as “wearable” circuitry and the like).

[0234] The processing unit 3610 may be a physical or virtual processor and can implement various processes based on programs stored in the memory 3620. 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 3600.

15 The processing unit 3610 may also be referred to as a central processing unit (CPU), a microprocessor, a controller or a microcontroller.

[0235] The computing device 3600 typically includes various computer storage medium. Such medium can be any medium accessible by the computing device 3600, including, but not limited to, volatile and non-volatile medium, or detachable and non-detachable medium. The memory 3620 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 3630 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 3600.

[0236] The computing device 3600 may further include additional detachable/non-detachable, volatile/non-volatile memory medium. Although not shown in Fig. 36, 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.

[0237] The communication unit 3640 communicates with a further computing device via the communication medium. In addition, the functions of the components in the computing device 3600 can be implemented by a single computing cluster or multiple computing machines that can communicate via communication connections. Therefore, the computing device 3600 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.

[0238] The input device 3650 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 3660 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 3640, the computing device 3600 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 3600, or any devices (such as a network card, a modem and the like) enabling the computing device 3600 to communicate with one or more other computing devices, if required. Such communication can be performed via input/output (I/O) interfaces (not shown).

[0239] In some embodiments, instead of being integrated in a single device, some or all components of the computing device 3600 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.

[0240] The computing device 3600 may be used to implement video encoding/decoding in embodiments of the present disclosure. The memory 3620 may include one or more video coding modules 3625 having one or more program instructions. These modules are accessible and executable by the processing unit 3610 to perform the functionalities of the various embodiments described herein.

[0241] In the example embodiments of performing video encoding, the input device 3650 may receive video data as an input 3670 to be encoded. The video data may be processed, for example, by the video coding module 3625, to generate an encoded bitstream. The encoded bitstream may be provided via the output device 3660 as an output 3680.

[0242] In the example embodiments of performing video decoding, the input device 3650 may receive an encoded bitstream as the input 3670. The encoded bitstream may be processed, for example, by the video coding module 3625, to generate decoded video data. The decoded video data may be provided via the output device 3660 as the output 3680.

[0243] While this disclosure has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present application as defined by the appended claims. Such variations are intended to be covered by the scope of this present application. As such, the foregoing description of embodiments of the present application is not intended to be limiting.

I/We Claim:

1. A method for video processing, comprising:

5 determining, for a conversion between a current video block of a video and a bitstream of the video, at least one cross component prediction (CCP) model for the current video block based on at least one history table (HT) of CCP models; and

performing the conversion based on the at least one CCP model,

10 wherein the at least one history table of CCP models is updated before or after coding a video unit.

2. The method of claim 1, wherein the at least one CCP model comprises at least one of:
a cross-component linear model (CCLM),

a variant of CCLM,

15 a convolutional cross-component model (CCCM), or

a variant of CCCM.

3. The method of claim 1 or 2, wherein the at least one history table is updated at a beginning of coding the video unit comprising the current video block, the video unit
20 comprising at least one of: a sequence, a picture, a slice, a tile, a sub-picture, a coding tree unit (CTU), or a CTU row.

4. The method of claim 3, wherein updating the at least one history table comprises one
25 of:

emptying the at least one history table; or

filling the at least one history table with default entries.

5. The method of claim 1 or 2, wherein the at least one history table is updated after
30 coding the video unit, the video unit comprising a coding unit (CU).

6. The method of claim 5, wherein a dual-tree coding is applied, and the coding unit
comprises a chroma coding unit.

7. The method of claim 5 or 6, wherein the coding unit is coded with a CCP mode.

8. The method of any of claims 5-7, wherein a target history table to be updated of the at least one history table is based on a coding mode of the coding unit.

5 9. The method of claim 8, wherein the coding mode of the coding unit comprises at least one of:

- a cross-component linear model (CCLM),
- a CCLM based on top neighboring samples of the coding unit (CCLM-T),
- a CCLM based on left neighboring samples of the coding unit (CCLM-L),
- 10 a multi-model based CCLM (MM-CCLM),
- a multi-model based CCLM-T (MM-CCCM-T),
- multi-model based CCLM-L (MM-CCCM-L),
- a gradient linear model (GLM), or
- a CCLM with slope adjustment, and

15 wherein a set of CCP models and related information for classifying samples into a plurality of groups of CCP models are stored in a first history table of the at least one history table, the first history table being associated with CCLM.

20 10. The method of claim 8, wherein the coding mode of the coding unit comprises at least one of:

- a convolutional cross-component model (CCCM),
- a CCCM based on top neighboring samples of the current video block (CCCM-T),
- a CCCM based on left neighboring samples of the current video block (CCCM-L),
- a multi-model based CCCM (MM-CCCM),
- 25 a multi-model based CCCM-T (MM-CCCM-T), or
- a multi-model based CCCM-L (MM-CCCM-L), and

wherein a set of CCP models and related information for classifying samples into a plurality of groups of CCP models are stored in a second history table of the at least one history table, the second history table being associated with CCCM.

30 11. The method of any of claims 5-10, wherein the coding unit is coded with a cross-component linear model (CCLM) with slope adjustment, and a corresponding CCP model is adjusted by the slope adjustment before being used to update the at least one history table.

12. The method of any of claims 5-11, wherein the at least one history table is updated by:

adding a set of information related to the at least one model used by the coding unit into the at least one history table.

5

13. The method of claim 12, wherein the set of information comprises at least one of:

the at least one CCP model,

the number of the at least one CCP model,

a threshold for classifying samples into a plurality of groups of CCP models, or

10

slope adjustment information.

14. The method of claim 12 or 13, wherein an approach for adding the set of information related to the at least one CCP model into the history table is based on whether the history table is full.

15

15. The method of claim 14, wherein if the history table is not full, the set of information is added to a first vacant entry of the history table.

16. The method of claim 15, wherein the first vacant entry comprises one of:

20

a vacant entry with a smallest index in the history table, or

a vacant entry with a largest index in the history table.

17. The method of claim 15 or 16, wherein the first vacant entry with the set of information is put as a last occupied entry in the history table.

25

18. The method of claim 17, wherein the last occupied entry comprises one of:

an occupied entry with a largest index in the history table, or

an occupied entry with a smallest index in the history table.

30

19. The method of claim 14, wherein if the history table is full, an entry in the history table is removed, and the history table is updated by adding the set of information to a further vacant entry of the history table.

20. The method of claim 19, wherein the entry is with a smallest index in the history

table.

21. The method of claim 20, wherein the updated history table is determined by $HT'[i]=HT[i+1]$, for $0 \leq i \leq N-2$, where HT denotes the history table, HT' denotes the updated history table, HT'[N-1] stores the set of information, and N denotes a size of the HT.

22. The method of claim 19, wherein the entry is with a largest index in the history table.

23. The method of claim 29, wherein the updated history table is determined by $HT'[i]=HT[i-1]$, for $1 \leq i \leq N-1$, where HT denotes the history table, HT' denotes the updated history table, HT'[0] stores the set of information, and N denotes a size of the HT.

24. The method of claim 19, wherein updating of the history table is based on a first in first out approach.

25. The method of any of claims 12-24, wherein at least one of whether to update the history table based on the set of information or how to update the history table is determined based on at least one similarity between the set of information and at least one entry in the history table.

26. The method of claim 25, wherein if a similarity between the set of information and a target entry is less than or equal to a threshold, the set of information is not added into the history table.

27. The method of claim 26, wherein the target entry is moved to a beginning position of the history table, and an entry in the history table before the target entry is moved one position backward.

28. The method of claim 26 or 27, wherein the history table is updated by: $HT'[0] = HT[k]$; $HT'[i]=HT[i-1]$ for $1 \leq i \leq k$; $HT'[i]=HT[i]$ for $i > k$, where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1, \dots$ denotes an entry in the history table, HT[k] denotes the target entry.

29. The method of claim 26, wherein the target entry is moved to an end position of the

history table, and an entry in the history table before the target entry is moved one position forward.

30. The method of claim 29, wherein the history table is updated by $HT'[N-1] = HT[k]$;
 5 $HT'[i] = HT[i+1]$ for $k \leq i \leq N-2$; $HT'[i] = HT[i]$ for $i < k$,

where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1, \dots$ denotes an entry in the history table, HT[k] denotes the target entry, and N denotes a size of the history table.

10 31. The method of any of claims 12-24, wherein at least one of whether to update the history table based on the set of information or how to update the history table is determined based on coding information of the coding unit with the set of information.

15 32. The method of claim 31, wherein the set of information is of the coding unit coded with history-based CCP (H-CCP) mode, the set of information is not added into the history table.

20 33. The method of claim 32, wherein a target entry in the history table is used by the coding unit, the target entry is moved to a beginning position of the history table, and an entry in the history table before the target entry is moved one position backward.

34. The method of claim 33, wherein the history table is updated by: $HT'[0] = HT[k]$;
 $HT'[i] = HT[i-1]$ for $1 \leq i \leq k$; $HT'[i] = HT[i]$ for $i > k$,

25 where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1, \dots$ denotes an entry in the history table, HT[k] denotes the target entry.

35. The method of claim 32, wherein a target entry in the history table is used by the coding unit, the target entry is moved to an end position of the history table, and an entry in the history table before the target entry is moved one position forward.

30 36. The method of claim 35, wherein the history table is updated by $HT'[N-1] = HT[k]$;
 $HT'[i] = HT[i+1]$ for $k \leq i \leq N-2$; $HT'[i] = HT[i]$ for $i < k$,

where HT denotes the history table, HT' denotes the updated history table, HT[i] with $i = 0, 1, \dots$ denotes an entry in the history table, HT[k] denotes the target entry, and N denotes a

size of the history table.

37. The method of any of claims 1-36, wherein a maximum size of a history table of the at least one history table is predetermined.

5

38. The method of claim 37, wherein the maximum size is 5 or 6.

39. The method of any of claims 1-36, wherein a syntax element (SE) in the bitstream indicates a maximum size of a history table of the at least one history table.

10

40. The method of claim 39, wherein the syntax element is at one of: a block level, a sequence level, a group of pictures level, a picture level, a slice level, or a tile group level.

41. The method of claim 39 or 40, wherein the syntax element is in at least one of the following coding structures: a coding tree unit (CTU), a coding unit (CU), a transform unit (TU), a prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform block (TB), a prediction block (PB), a sequence header, a picture header, a sequence parameter set (SPS), a Video Parameter Set (VPS), a decoded parameter set (DPS), Decoding Capability Information (DCI), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), a slice header or a tile group header.

15

20

42. The method of any of claims 1-36, wherein a maximum size of a history table of the at least one history table is determined based on coding information.

25

43. The method of claim 42, wherein the coding information comprises at least one of:
a mode of the current video block,
a mode of a neighbouring block of the current video block,
a mode of a luma block in a collocated region of the current video block,
a mode of a luma block in the collocated region of a neighbouring block of the current
video block,
a quantization parameter (QP),
a slice type,
a picture type,
a width or a height of a picture,

30

a width or a height of a block, or
reconstructed samples of the current video block.

5 44. The method of any of claims 1-43, wherein an entry of the at least one history table includes CCP models for a plurality of chroma components.

45. The method of claim 44, wherein the plurality of chroma components comprises a first chroma component and a second chroma component.

10 46. The method of claim 45, wherein the entry is selected for the current video block, and the first and second chroma components of the current video block is coded by corresponding CCP models in the entry.

15 47. The method of any of claims 1-43, wherein an entry of the at least one history table includes a CCP model for a single color component.

48. The method of claim 47, wherein the single color component comprises a first chroma component or a second chroma component.

20 49. The method of claim 47, wherein the entry is selected for the current video block, and a corresponding color component of the current video block is coded by the CCP model in the entry.

25 50. The method of any of claims 1-43, wherein a plurality of history tables of CCP models are determined for a plurality of color components.

30 51. The method of any of claims 1-50, wherein a syntax element in the bitstream is binarized as at least one of: a flag, a fixed length code, an exponential Golomb(x) (EG(x)) code, a unary code, a truncated unary code, or a truncated binary code, the syntax element comprising an indication or a flag.

52. The method of claim 51, wherein the syntax element is signed or unsigned.

53. The method of any of claims 1-52, wherein a syntax element in the bitstream is

coded with at least one context model, or bypass coded, the syntax element comprising an indication or a flag.

54. The method of any of claims 51-53, wherein the syntax element is included in the
5 bitstream based on a condition that a function associated with the syntax element is applicable.

55. The method of any of claims 51-54, wherein the syntax element is at at least one of:
a block level, a sequence level, a group of pictures level, a picture level, a slice level, or a tile
group level.

56. The method of any of claims 51-55, wherein the syntax element is in at least one of
the following coding structures: a coding tree unit (CTU), a coding unit (CU), a transform unit
(TU), a prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform block
(TB), a prediction block (PB), a sequence header, a picture header, a sequence parameter set
15 (SPS), a Video Parameter Set (VPS), a decoded parameter set (DPS), Decoding Capability
Information (DCI), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), a slice
header or a tile group header.

57. The method of any of claims 1-56, wherein information regarding whether to and/or
20 how to apply the method is included in the bitstream.

58. The method of claim 57, wherein the information is indicated at one of: a sequence
level, a group of pictures level, a picture level, a slice level or a tile group level.

59. The method of claim 57 or 58, wherein the information is indicated in at least one
of the following coding structures: a coding tree unit (CTU), a coding unit (CU), a transform
unit (TU), a prediction unit (PU), a coding tree block (CTB), a coding block (CB), a transform
block (TB), a prediction block (PB), a sequence header, a picture header, a sequence parameter
set (SPS), a Video Parameter Set (VPS), a decoded parameter set (DPS), Decoding Capability
30 Information (DCI), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), a slice
header or a tile group header.

60. The method of any of claims 57-59, wherein the information is based on coded
information.

61. The method of claim 60, wherein the coded information comprises at least one of: a block size, a colour format, a single or dual tree partitioning, a colour component, a slice type, or a picture type.

5

62. The method of any of claims 1-61, wherein the method is used in a coding tool requiring a chroma fusion.

63. The method of any of claims 1-62, wherein the conversion includes encoding the current video block into the bitstream.

10

64. The method of any of claims 1-62, wherein the conversion includes decoding the current video block from the bitstream.

15

65. 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 1-64.

20

66. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of claims 1-64.

67. 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:

25

determining at least one cross component prediction (CCP) model for a current video block of the video based on at least one history table (HT) of CCP models; and

generating the bitstream based on the at least one CCP model,

wherein the at least one history table of CCP models is updated before or after coding a video unit.

30

68. A method for storing a bitstream of a video, comprising:
determining at least one cross component prediction (CCP) model for a current video block of the video based on at least one history table (HT) of CCP models;
generating the bitstream based on the at least one CCP model; and

storing the bitstream in a non-transitory computer-readable recording medium,
wherein the at least one history table of CCP models is updated before or after coding a
video unit.

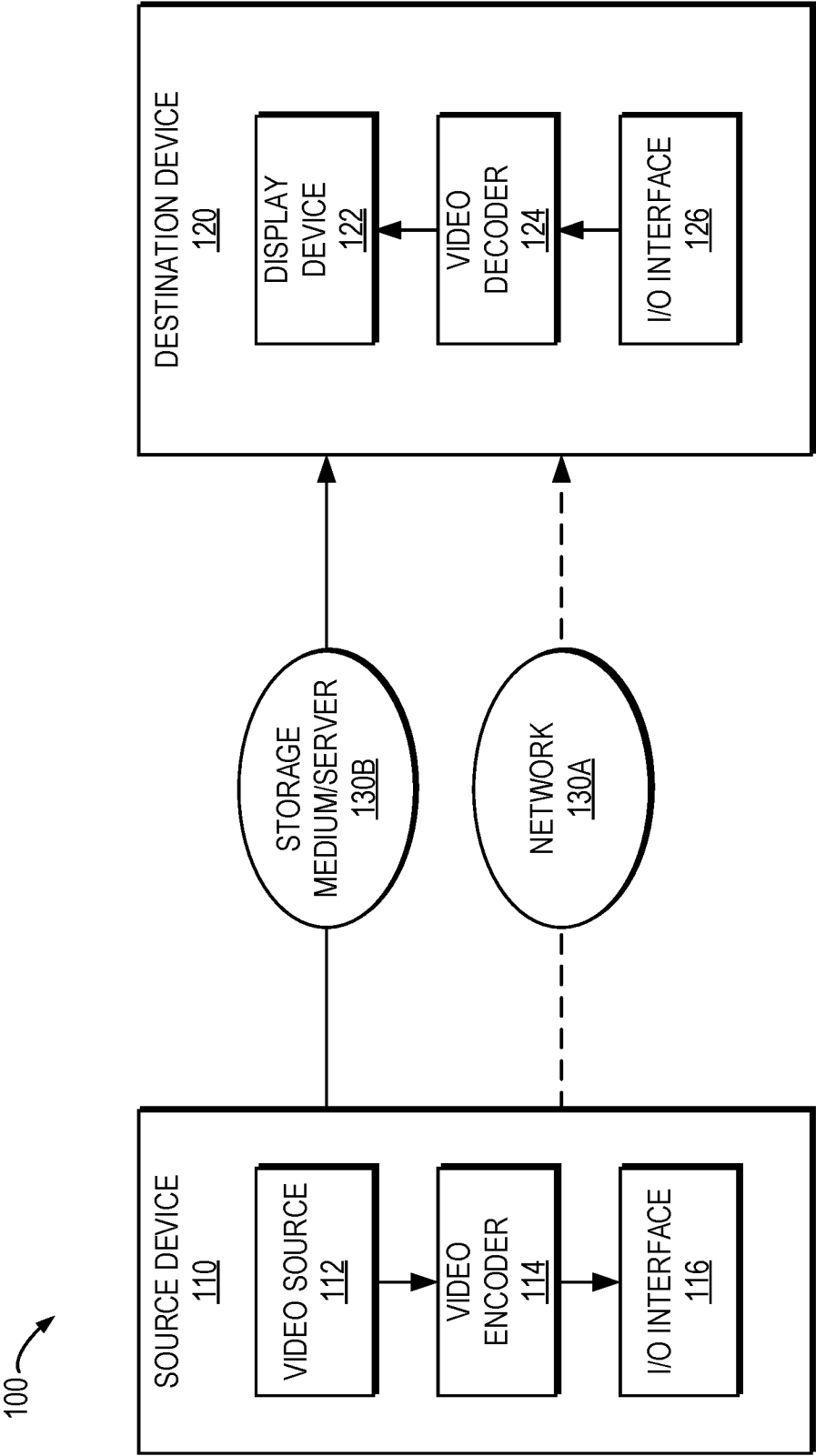


Fig. 1

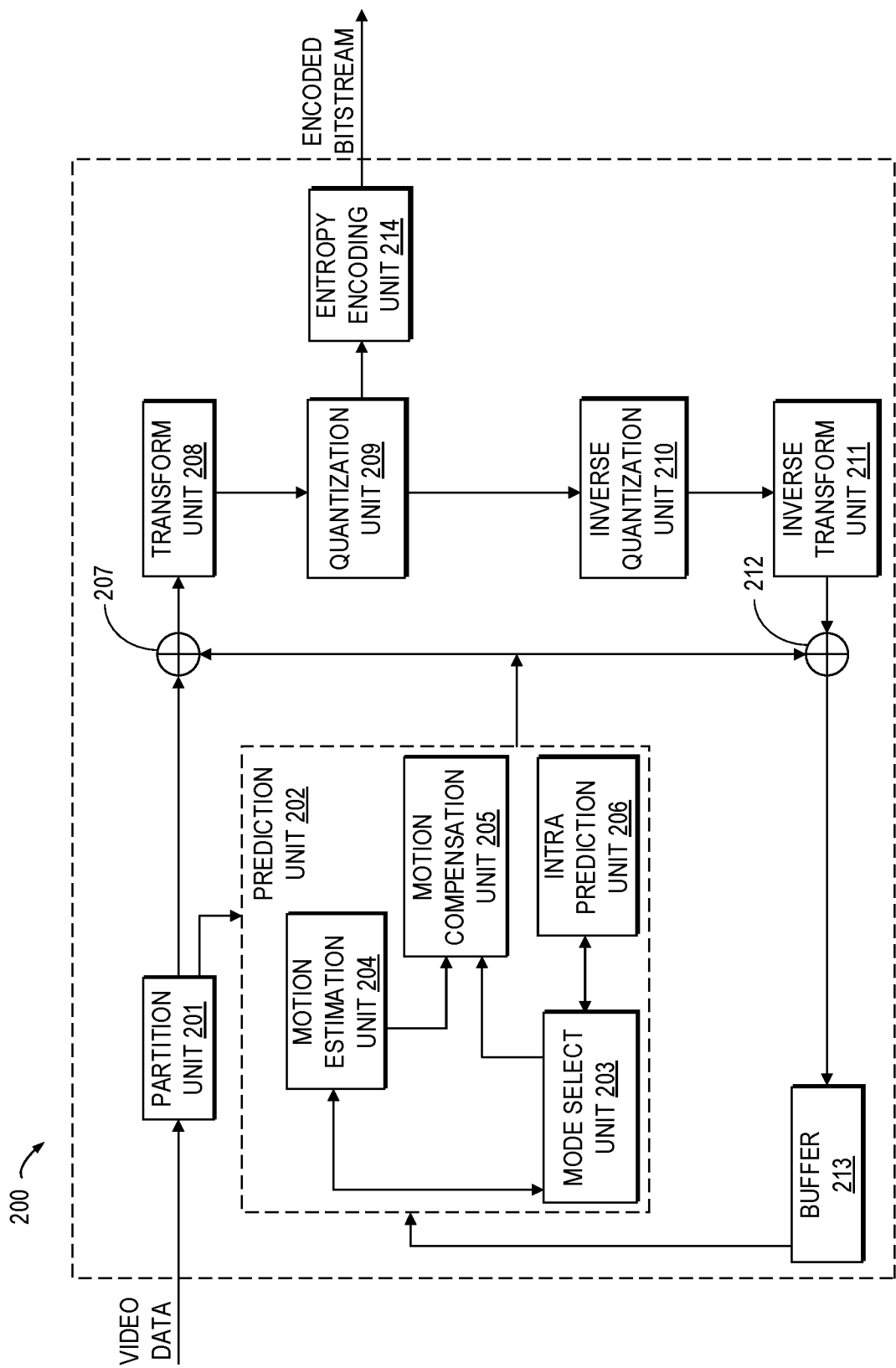


Fig. 2

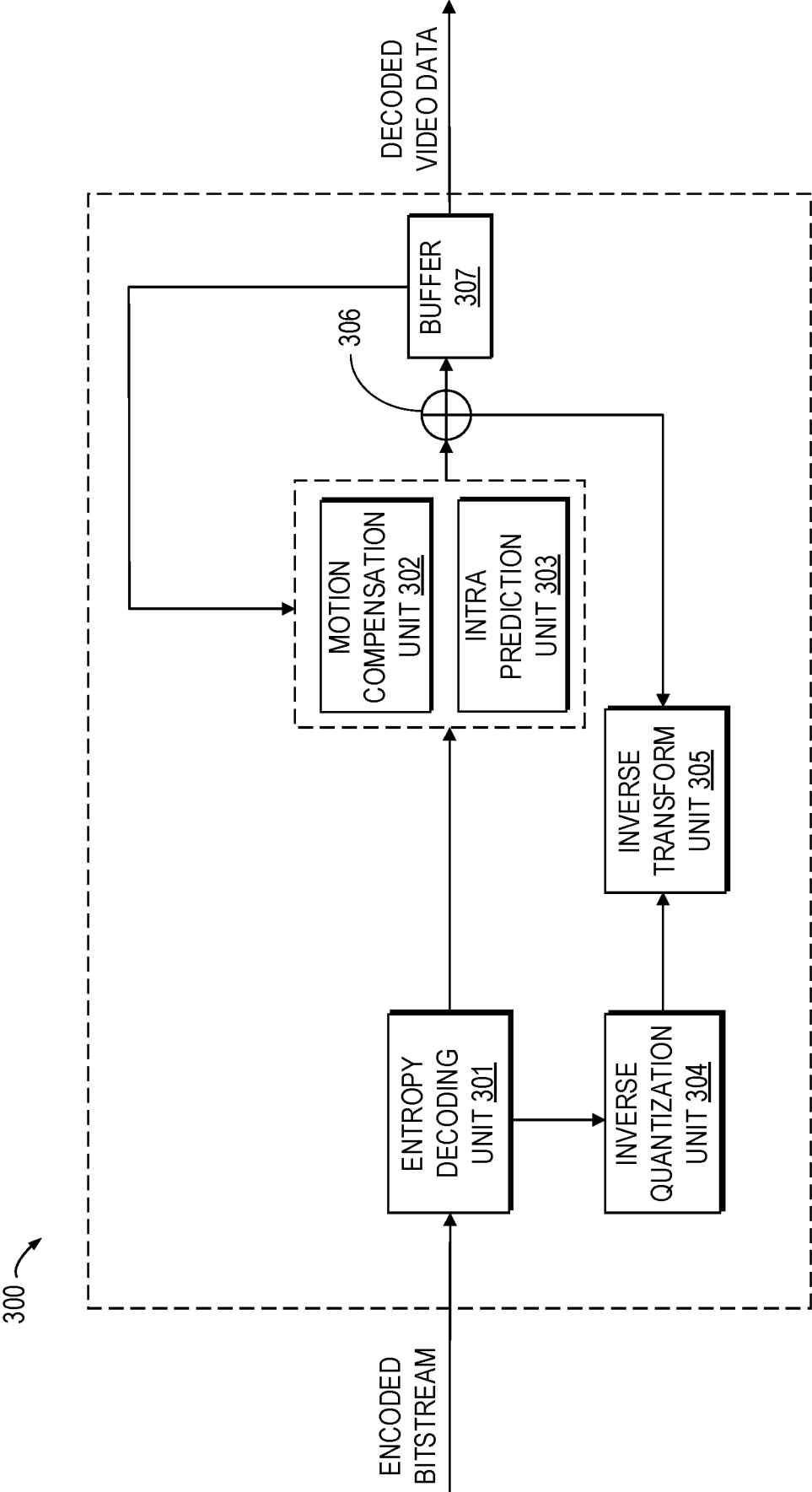


Fig. 3

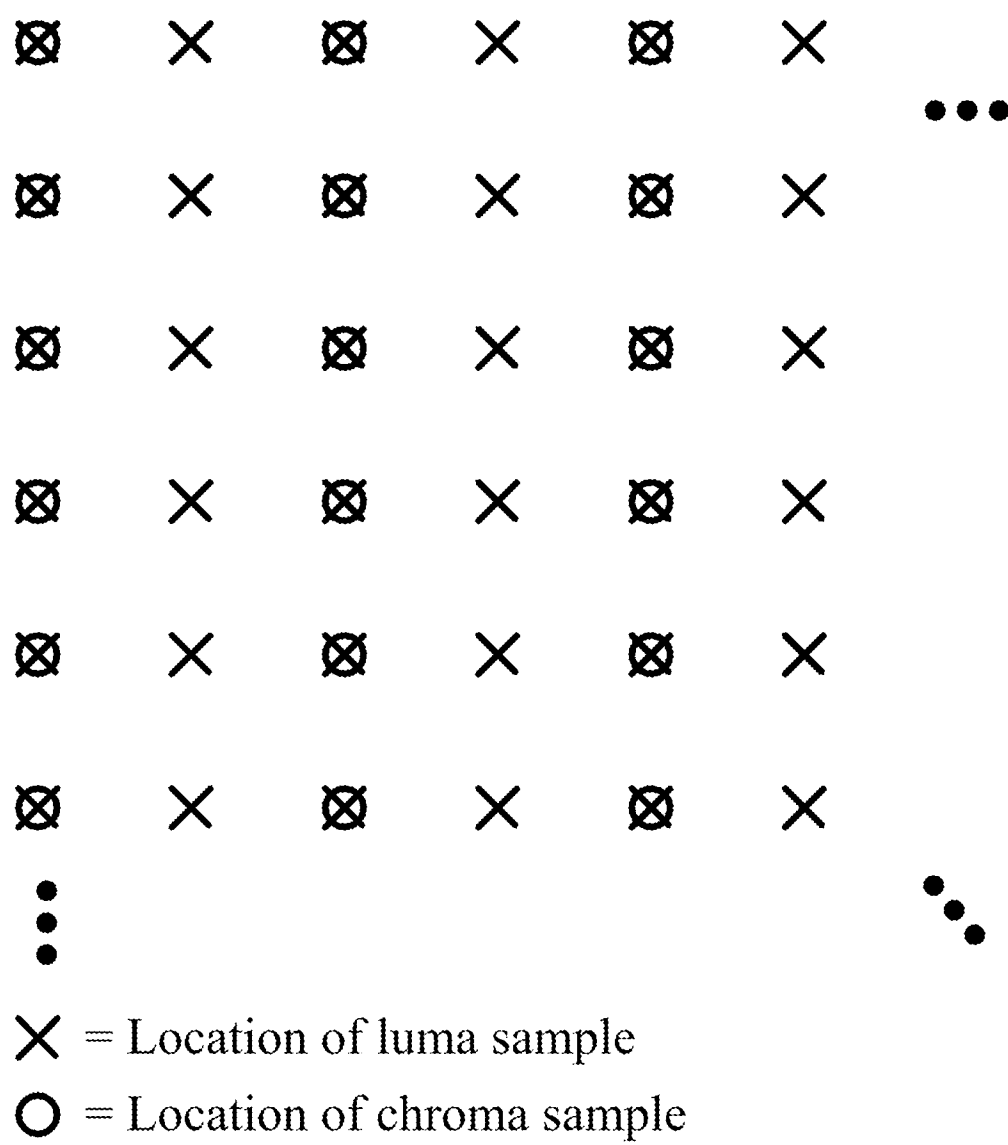


Fig. 4

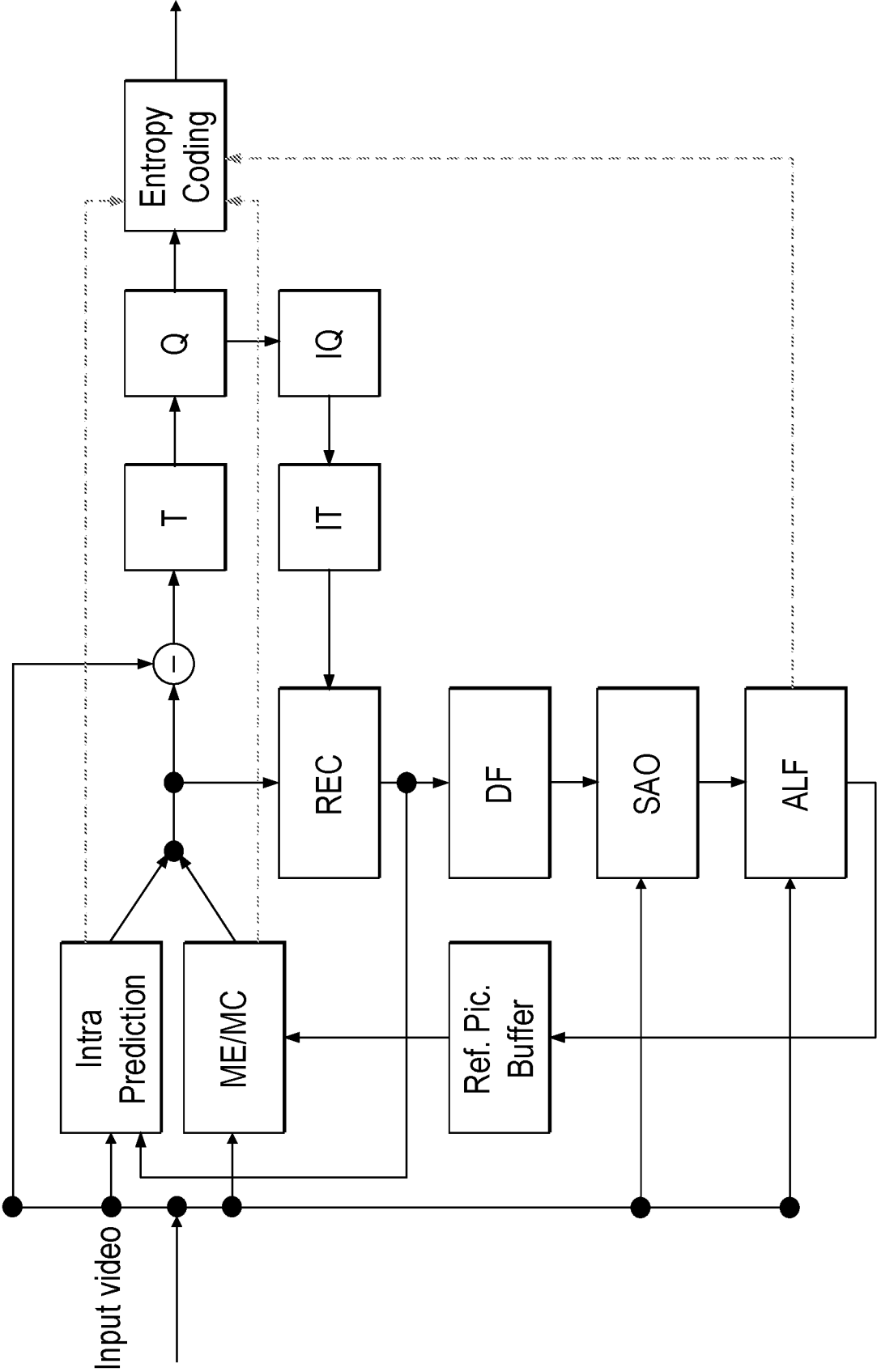


Fig. 5

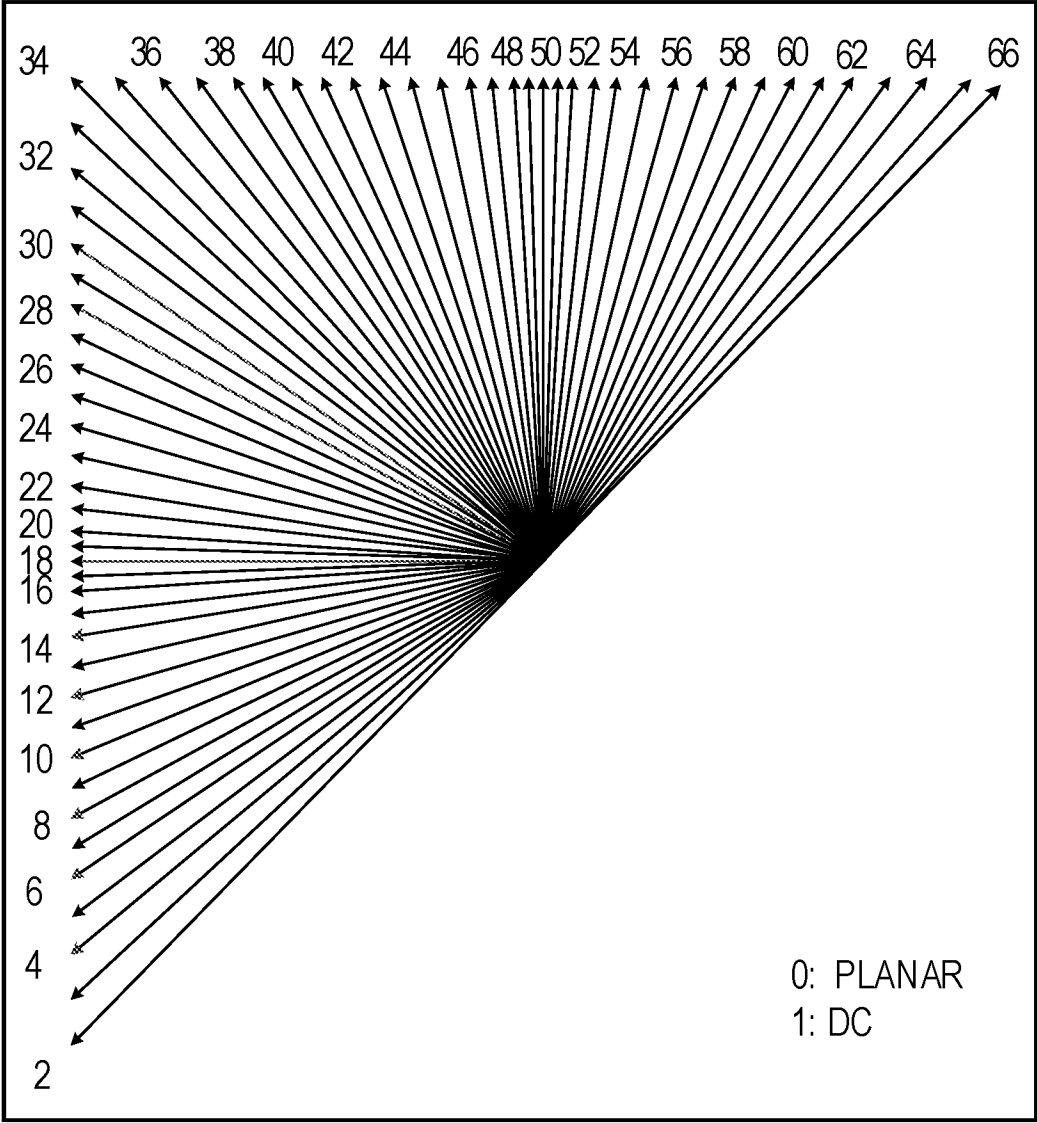


Fig. 6

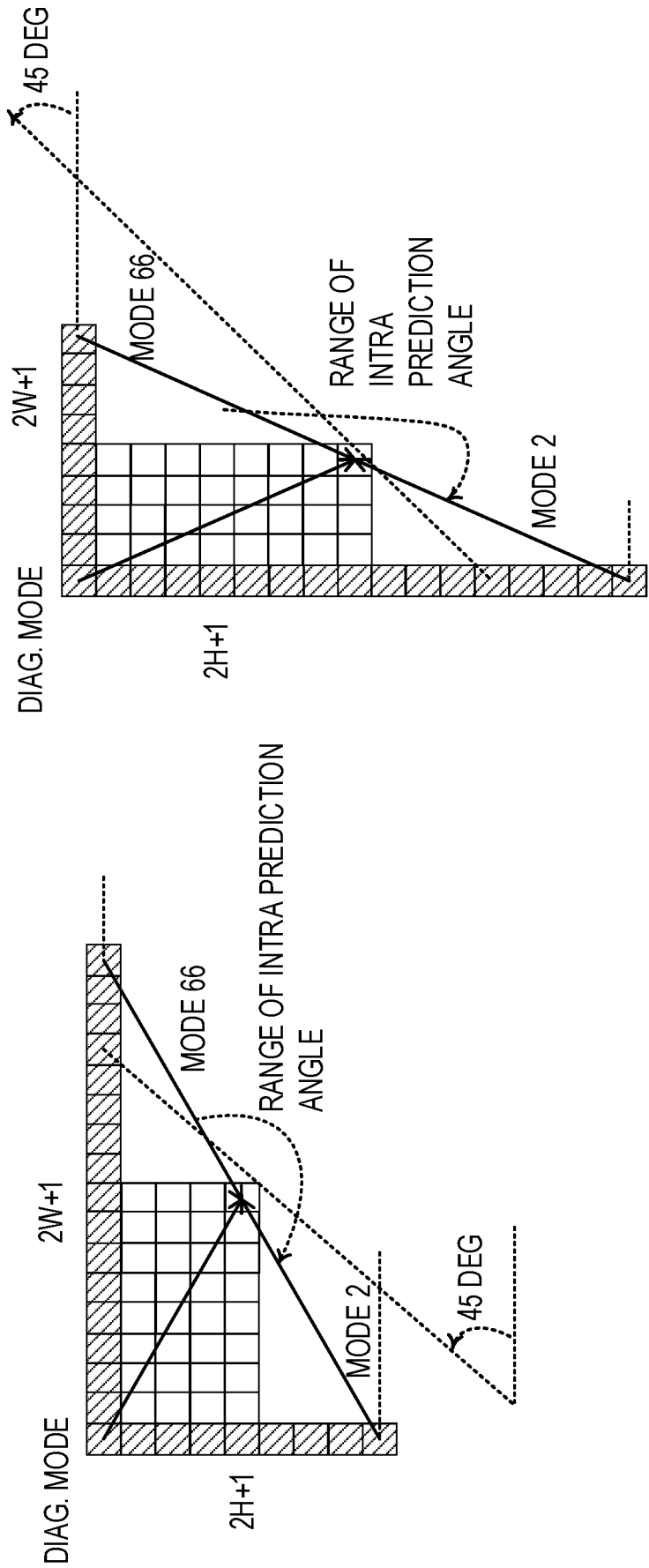


Fig. 7

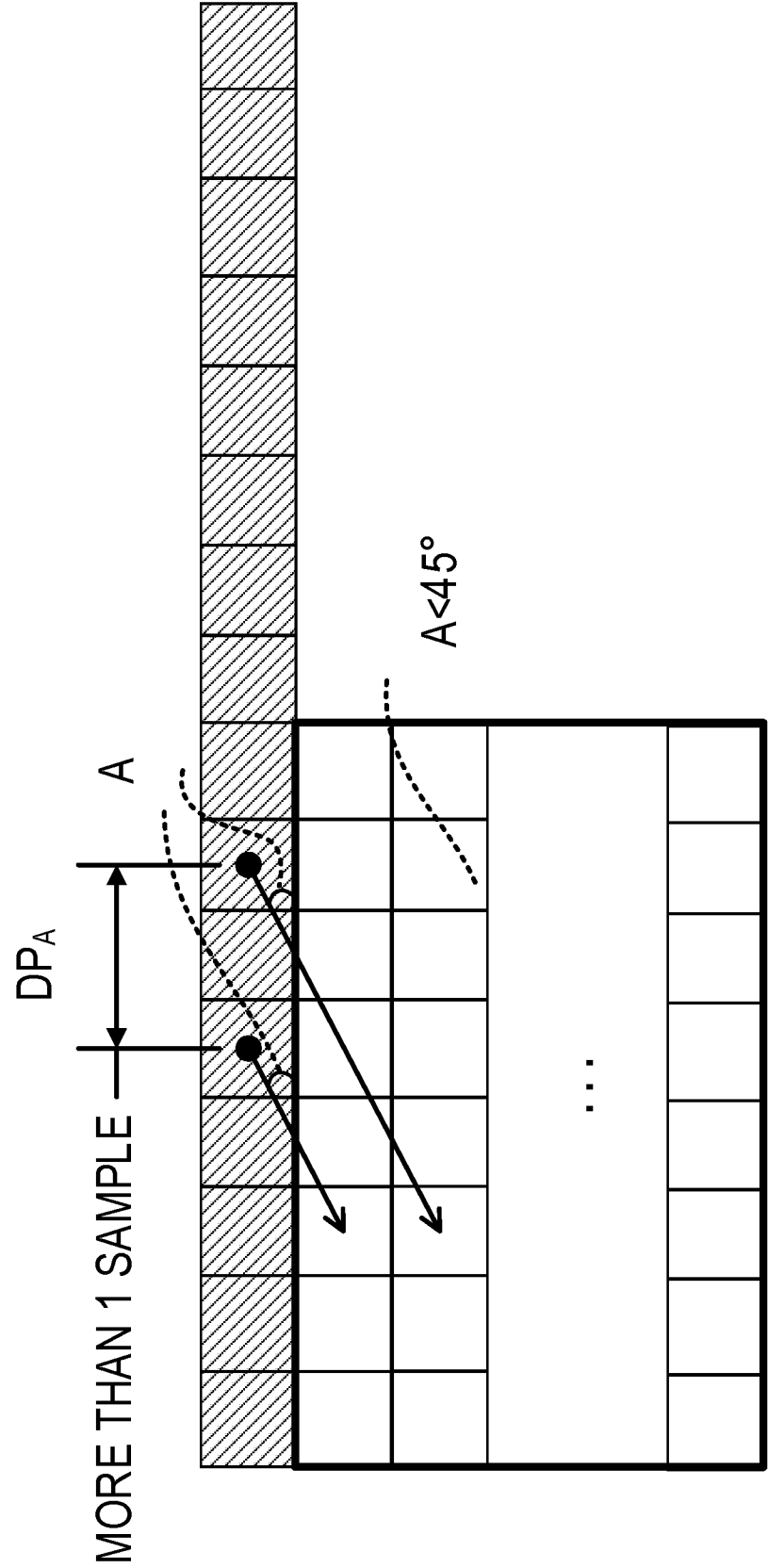


Fig. 8

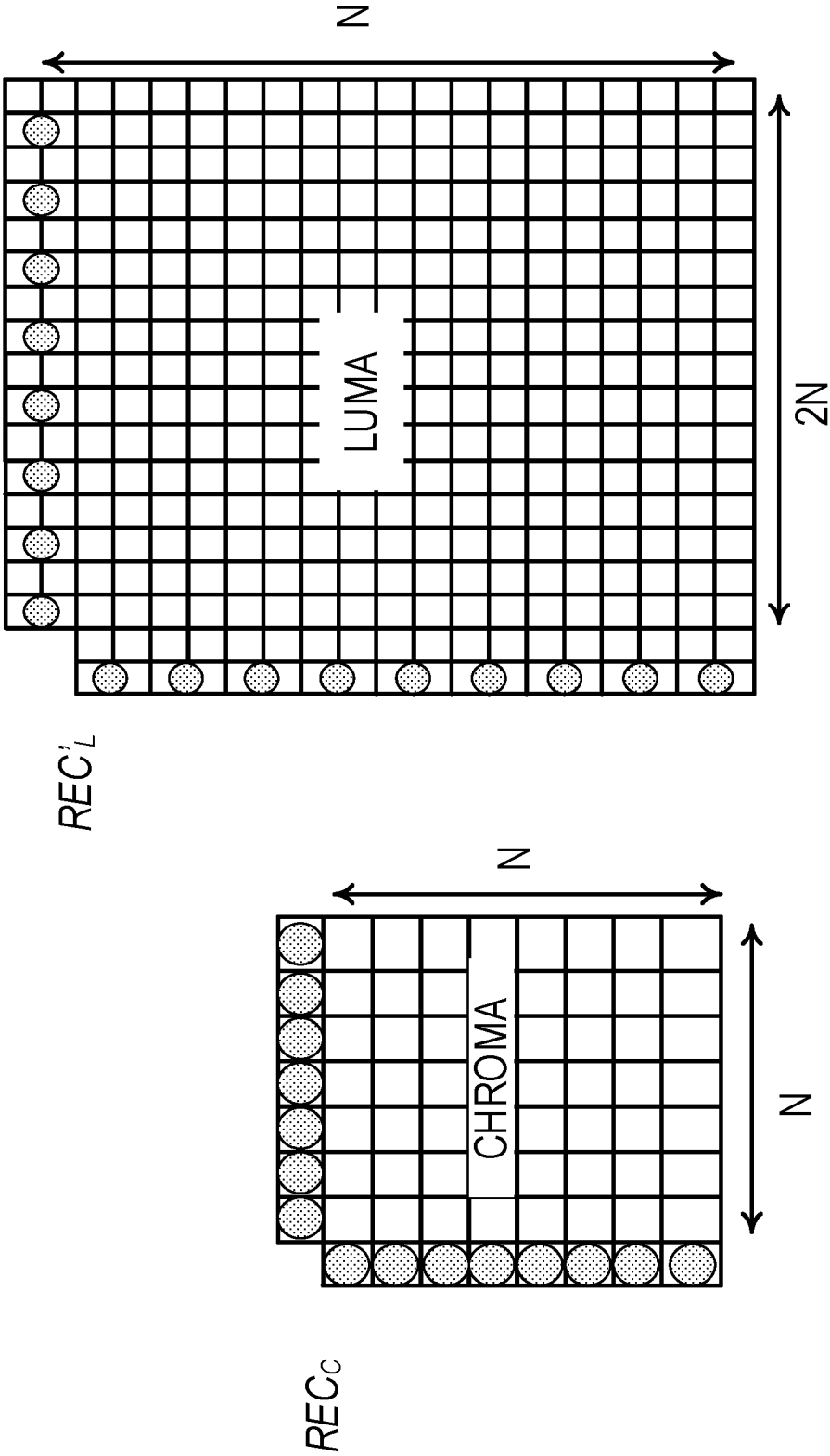


Fig. 9

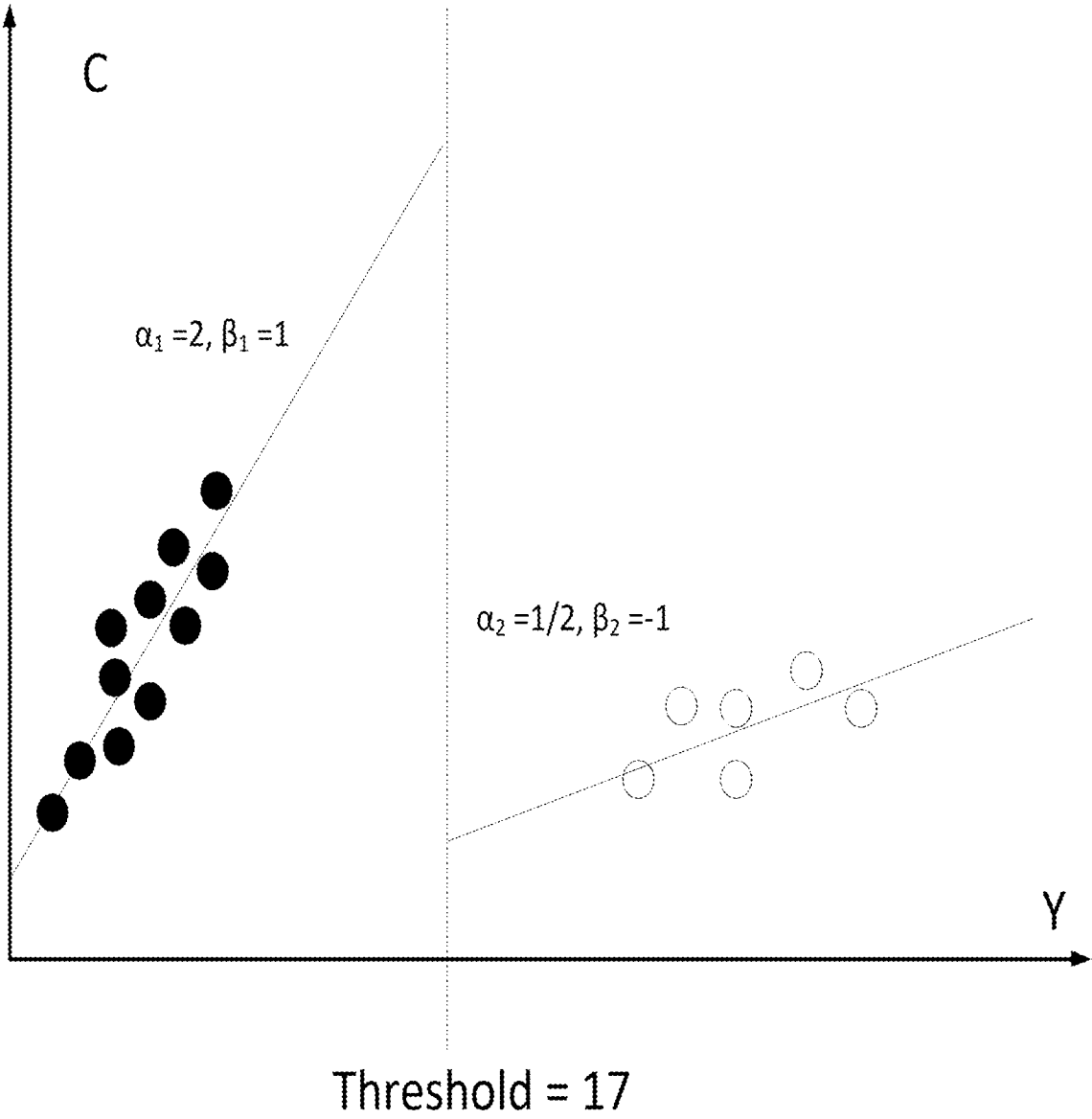


Fig. 10

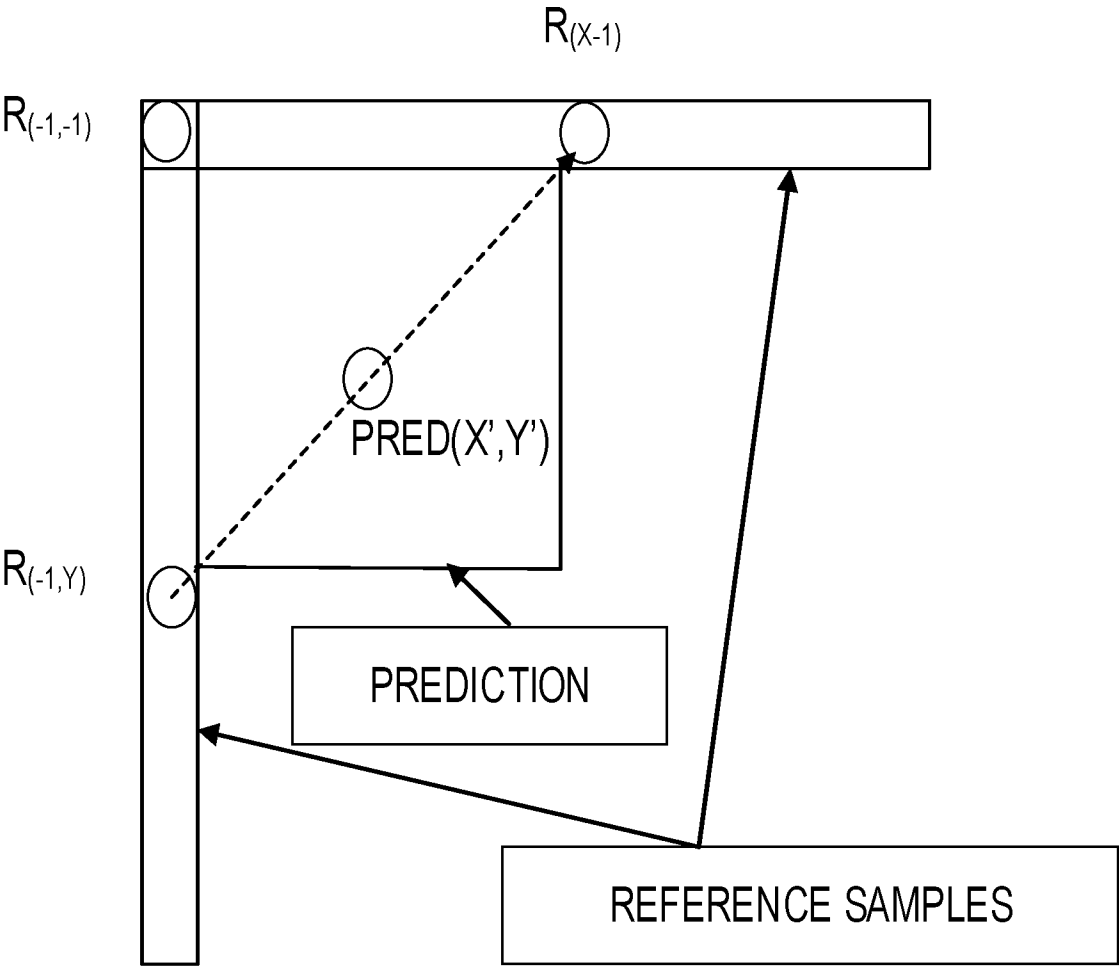


Fig. 11A

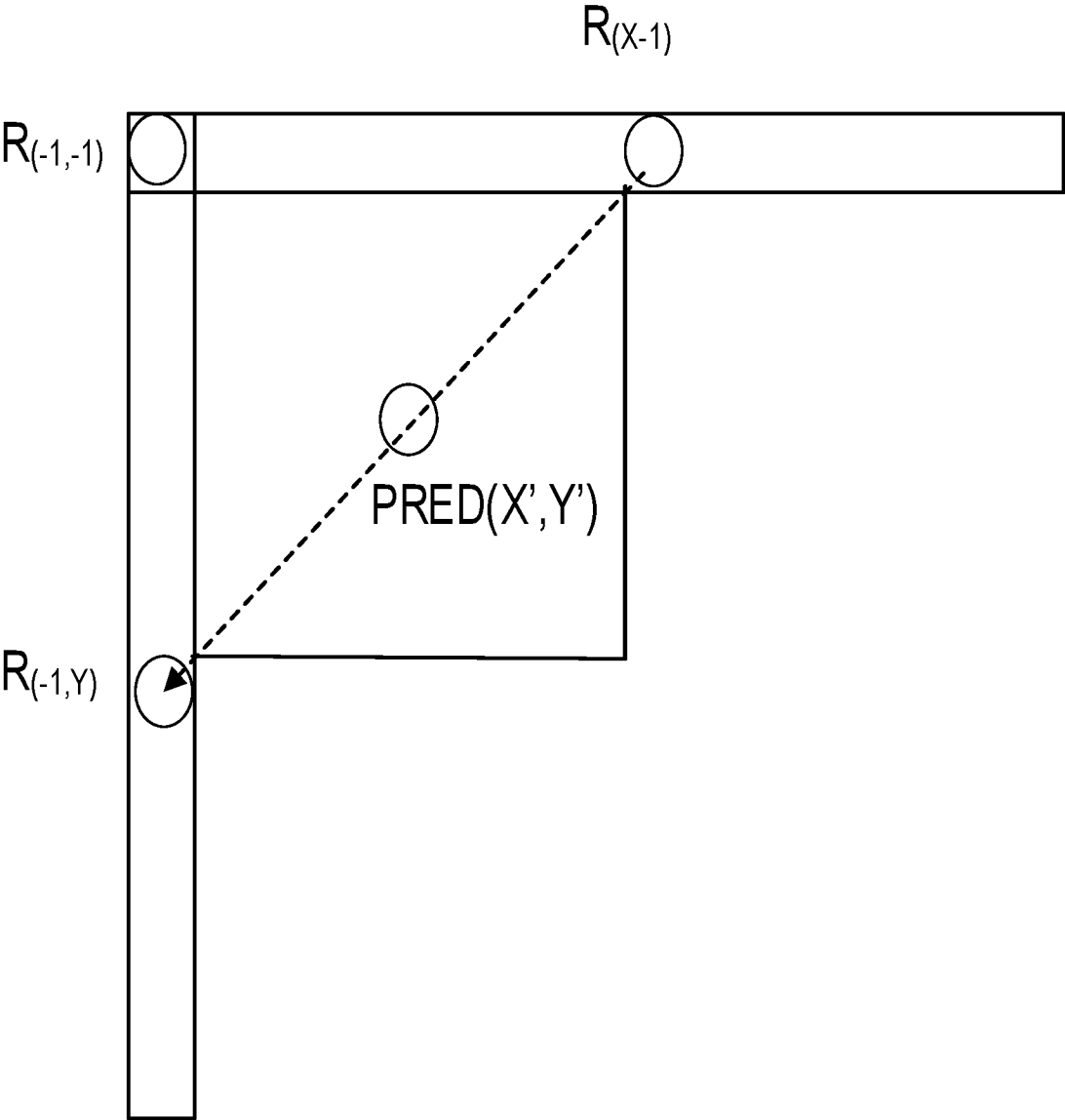


Fig. 11B

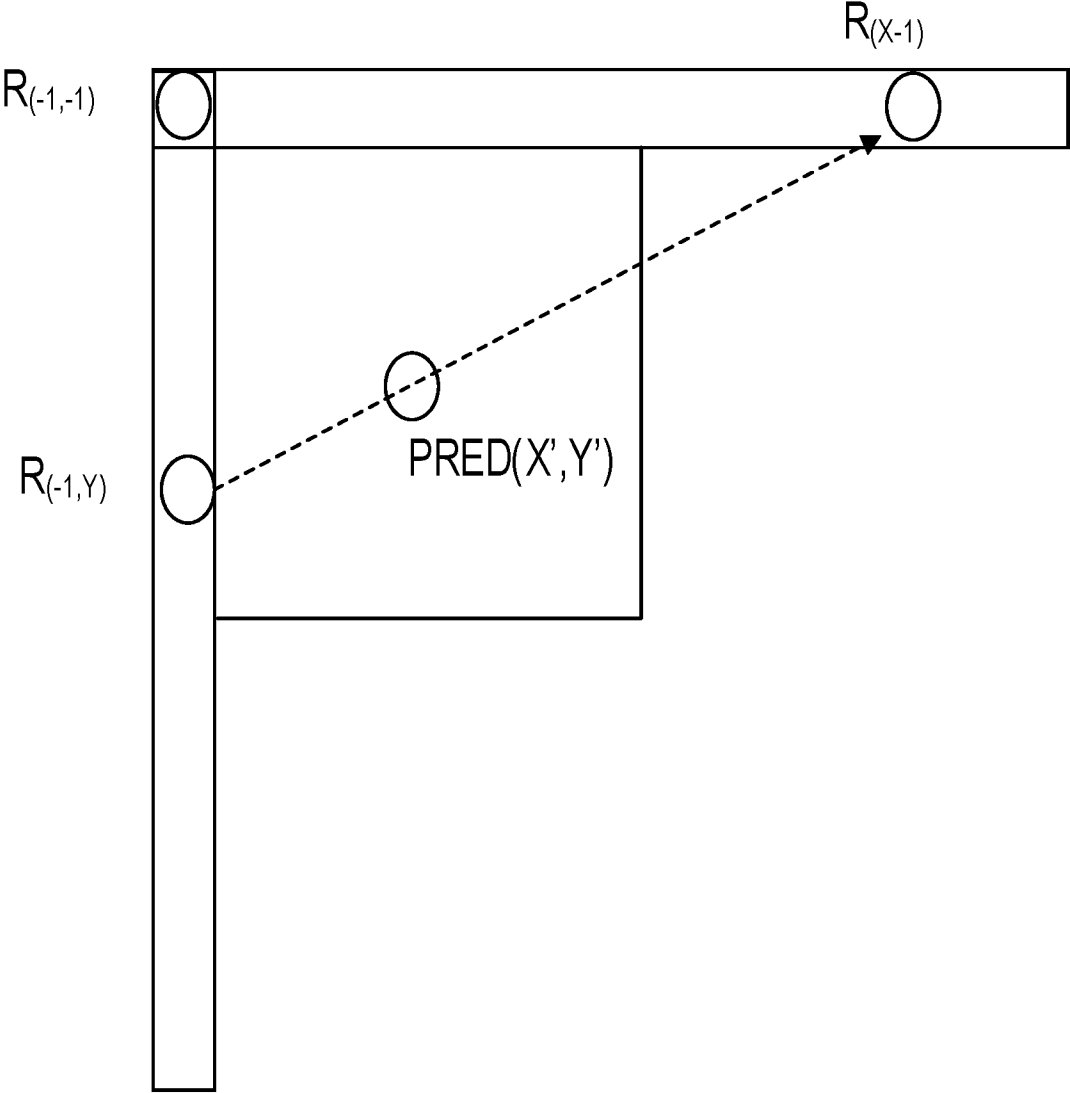


Fig. 11C

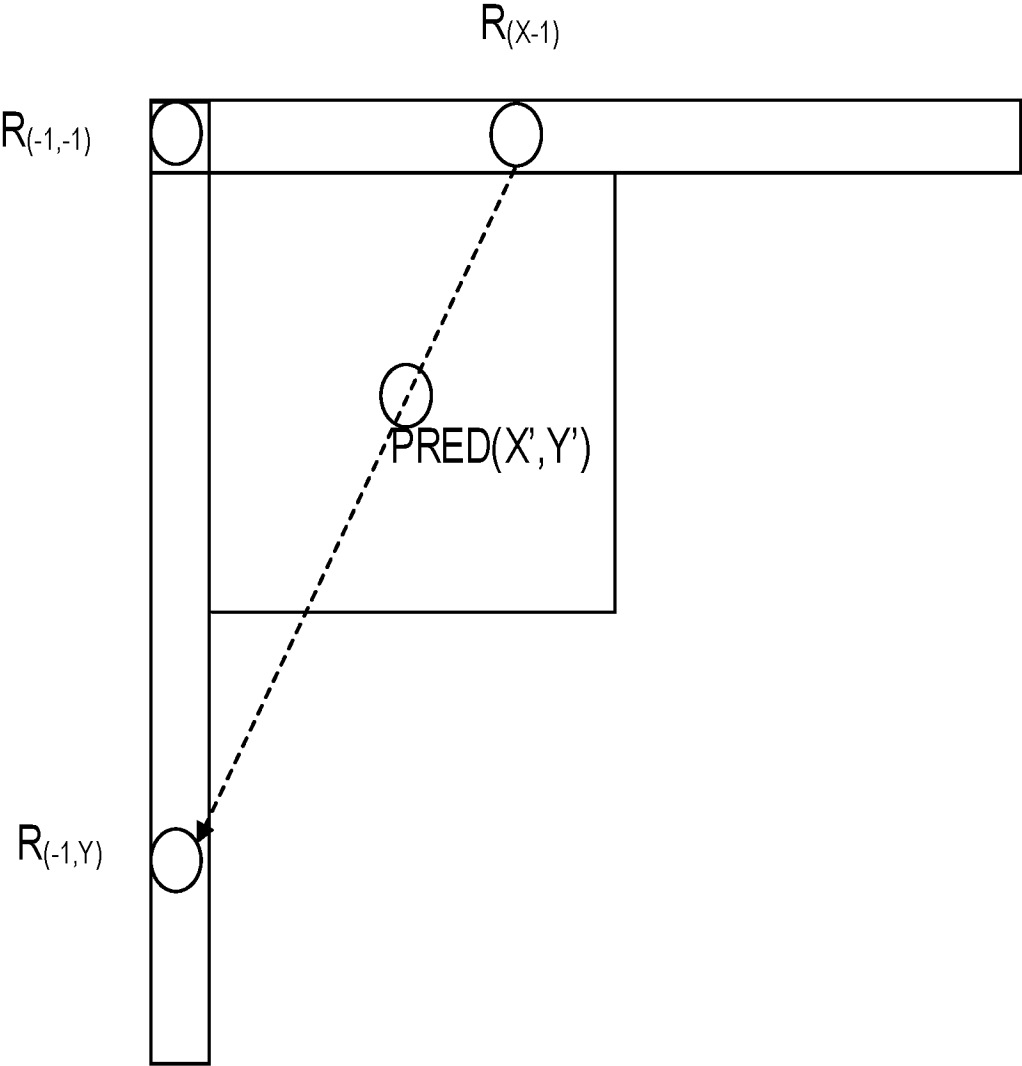


Fig. 11D

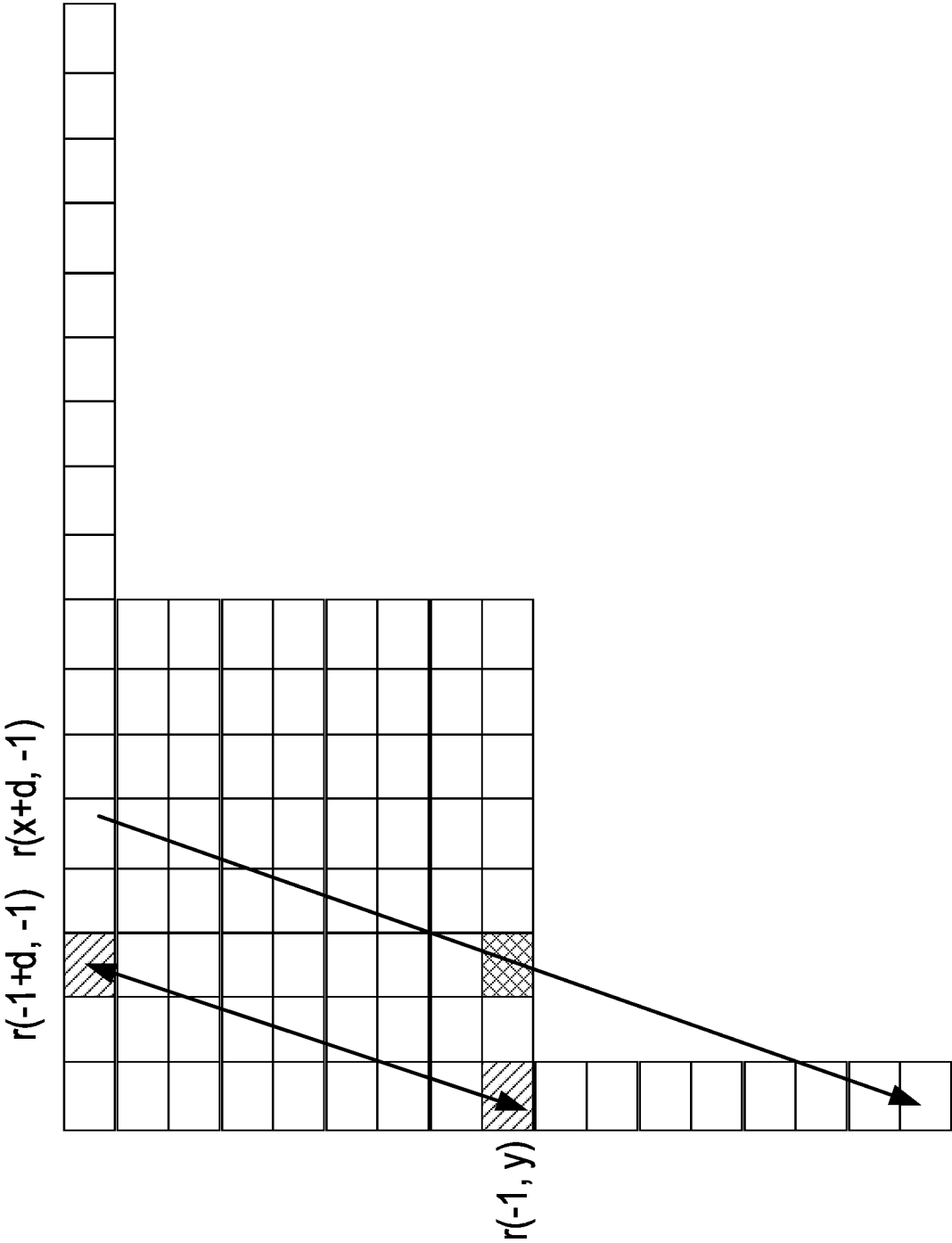


Fig. 12

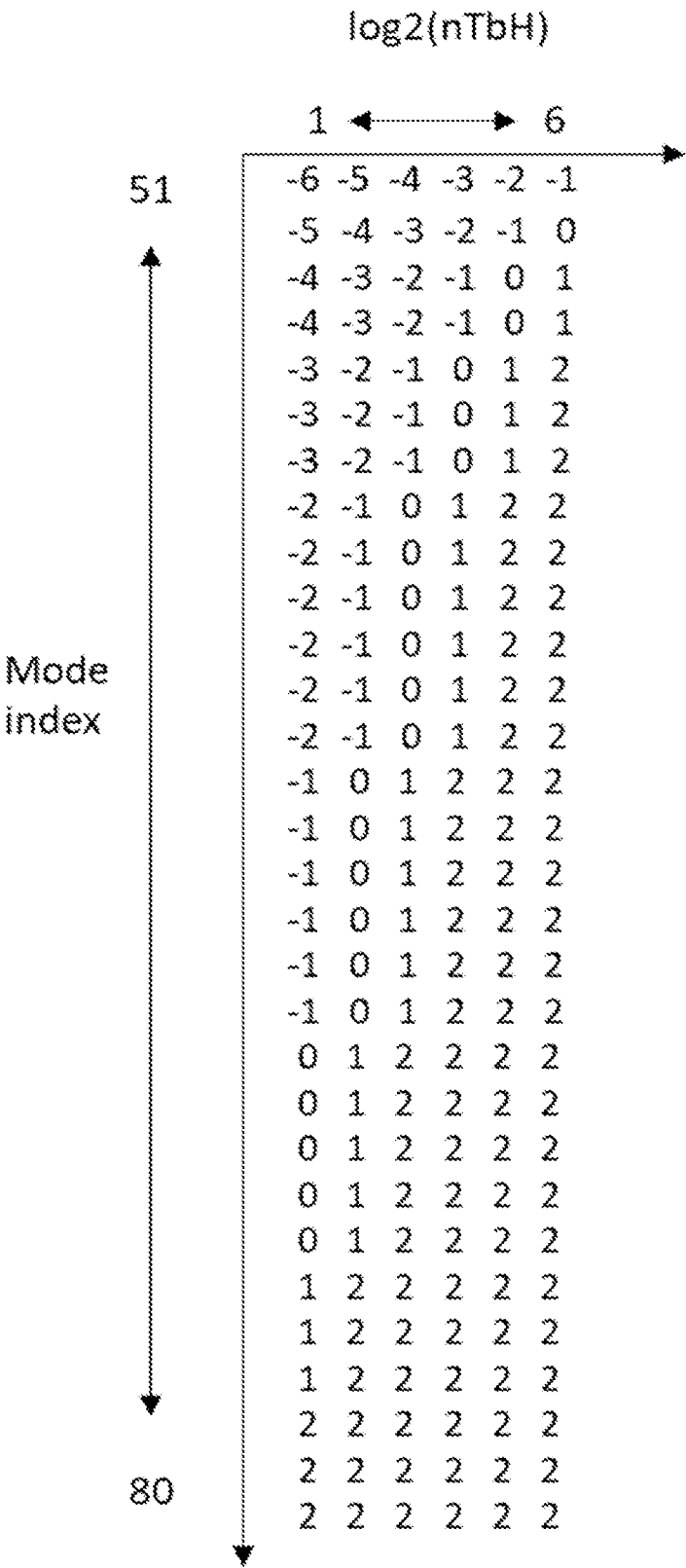


Fig. 13

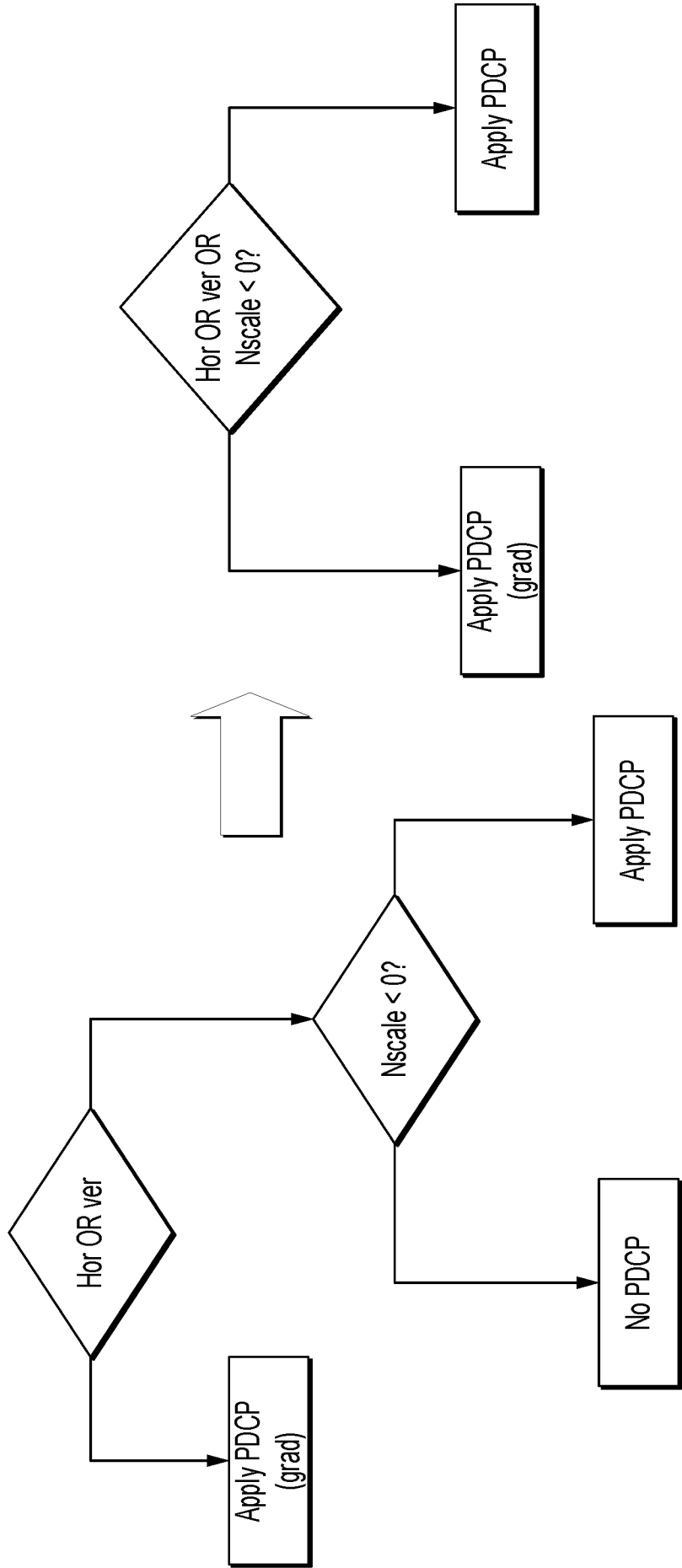


Fig. 14

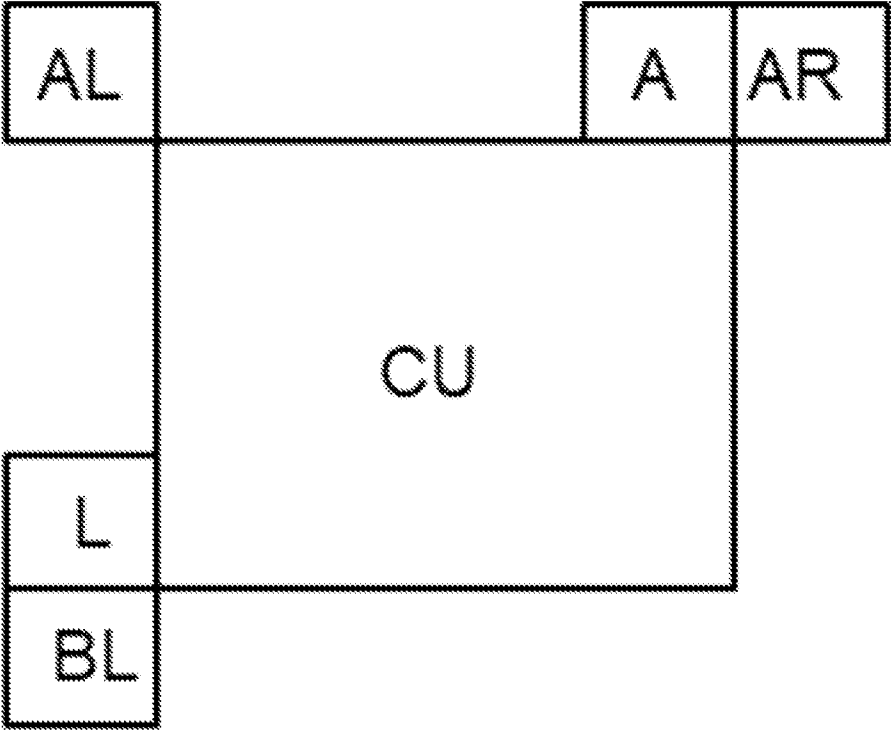


Fig. 15

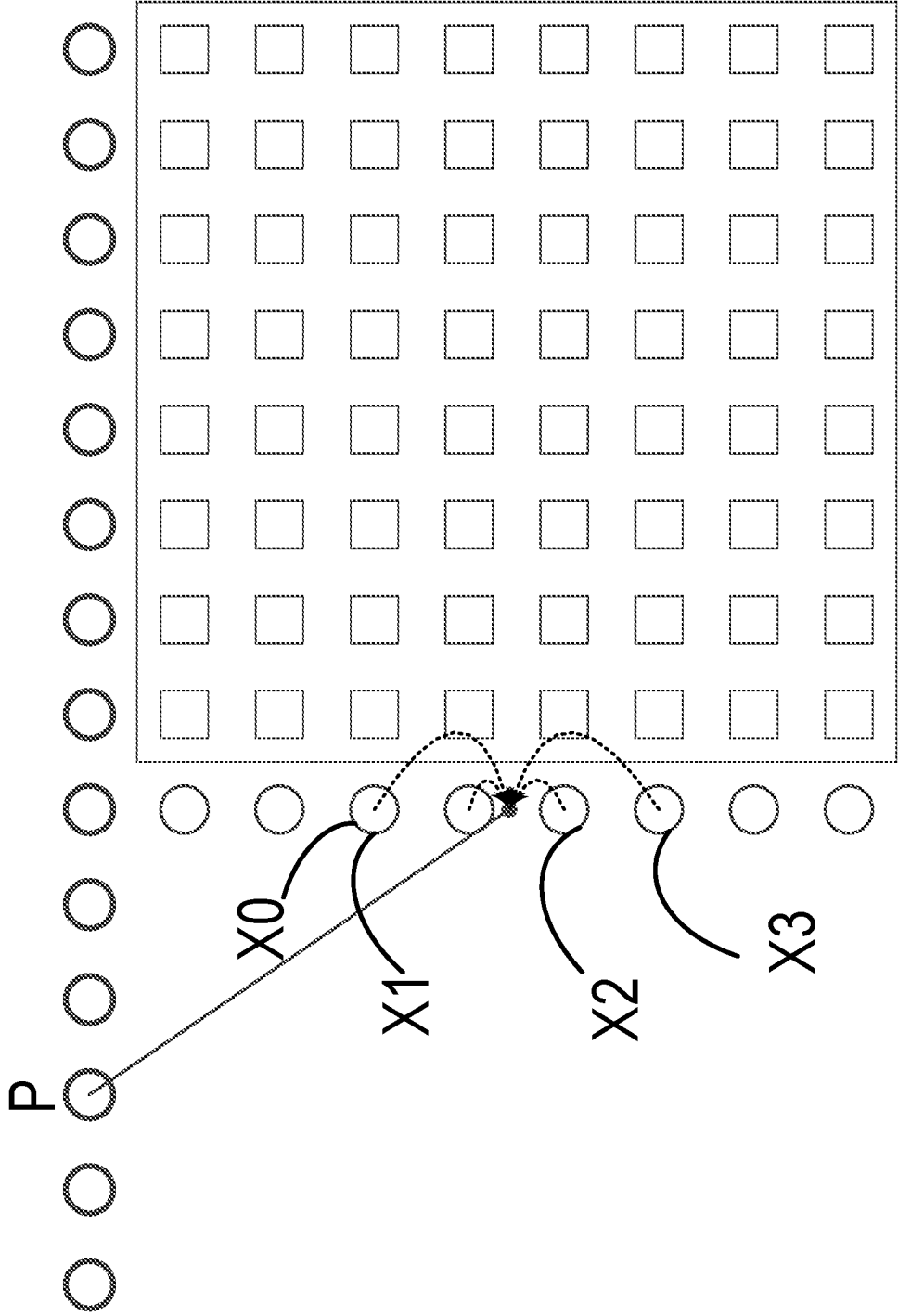


Fig. 16

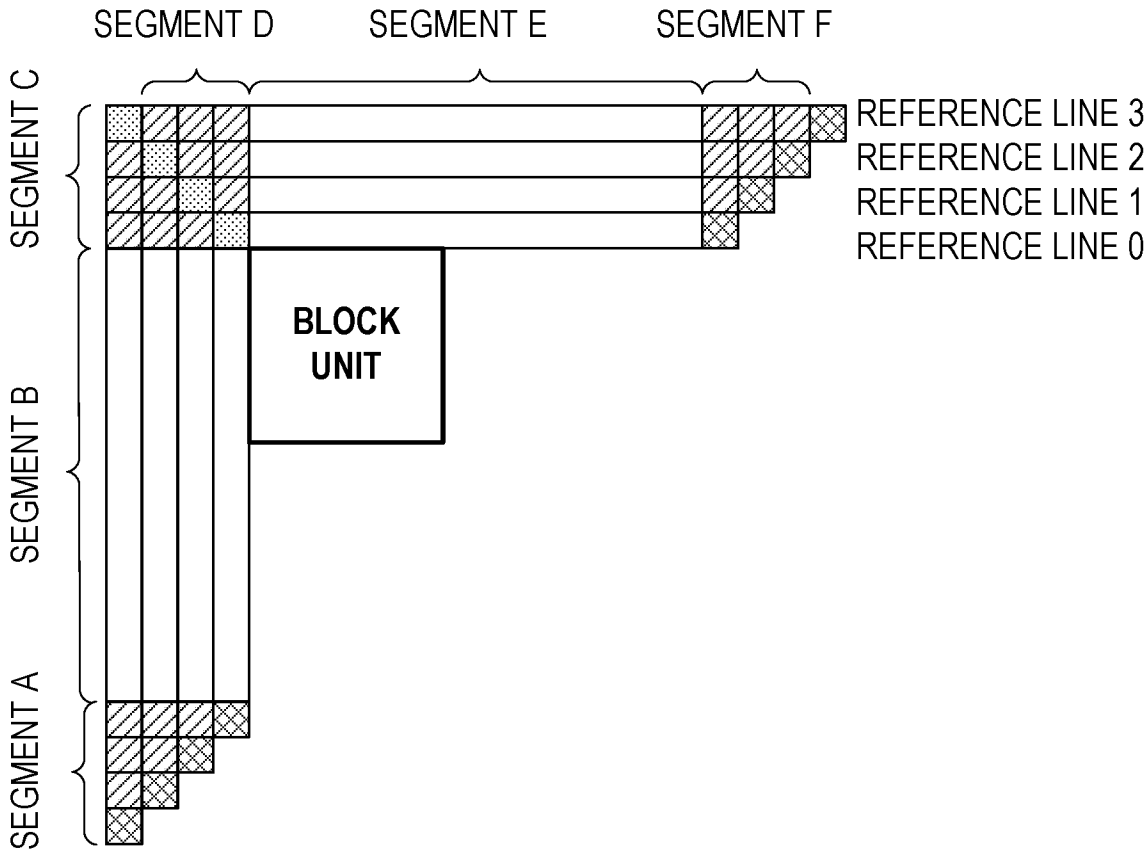


Fig. 17

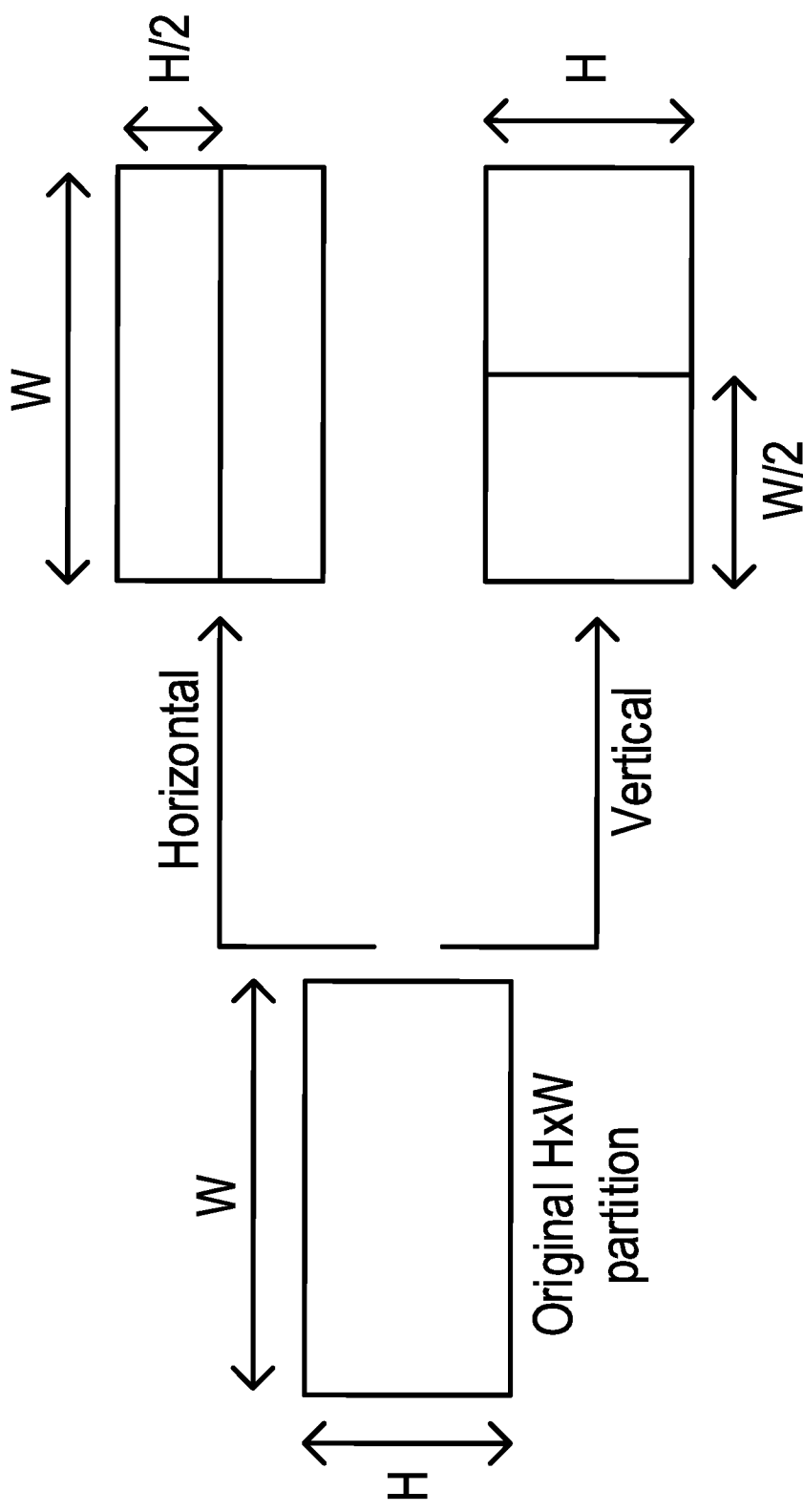


Fig. 18A

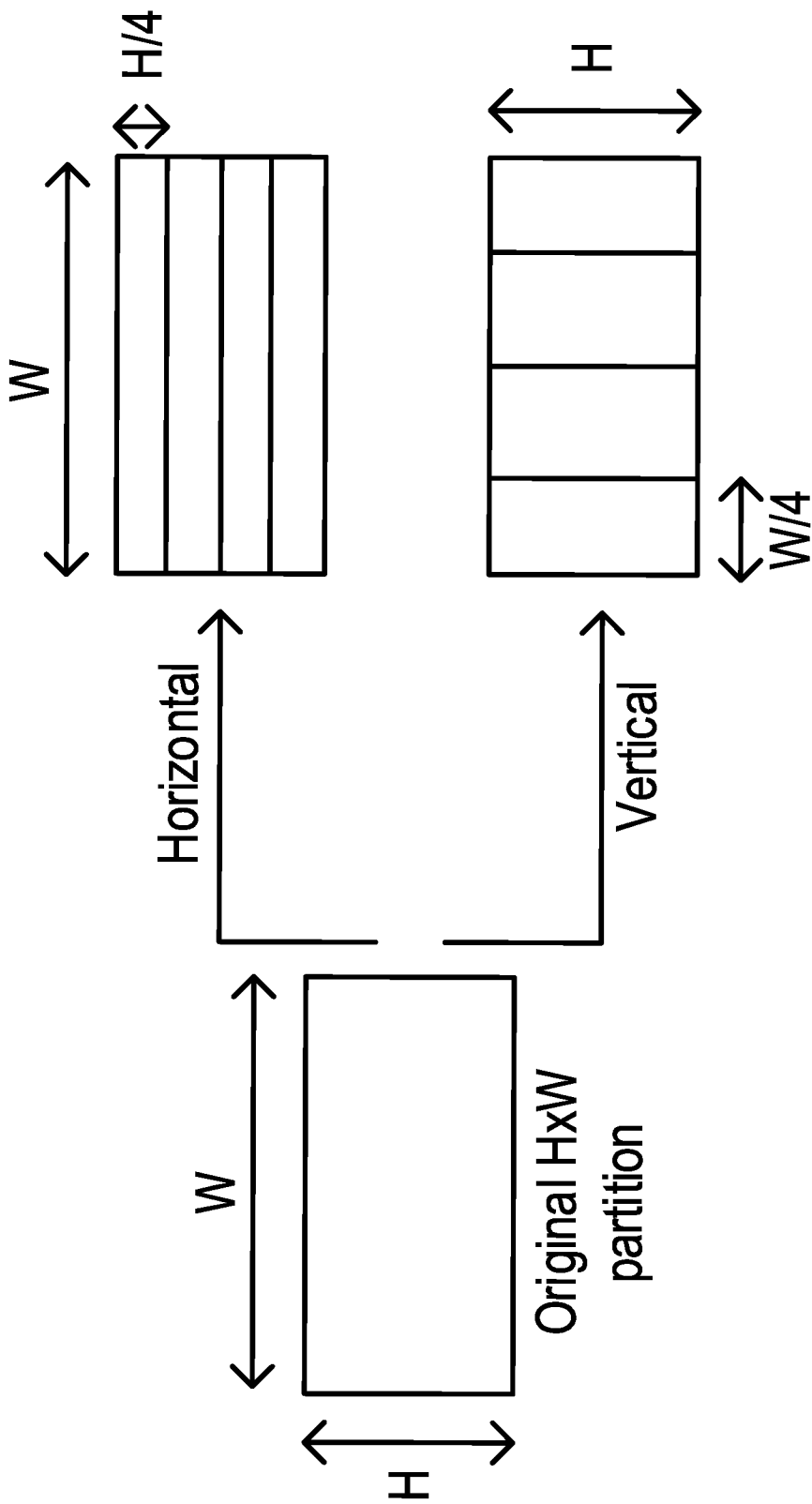


Fig. 18B

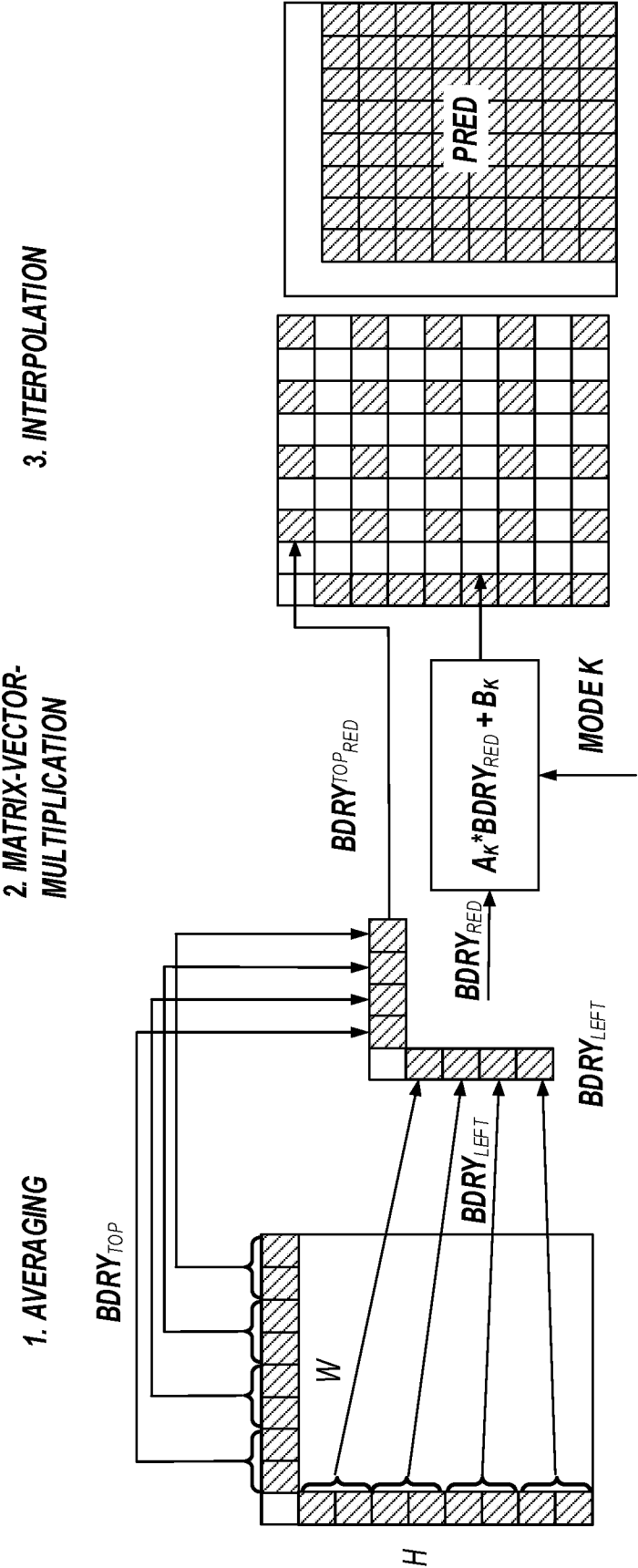
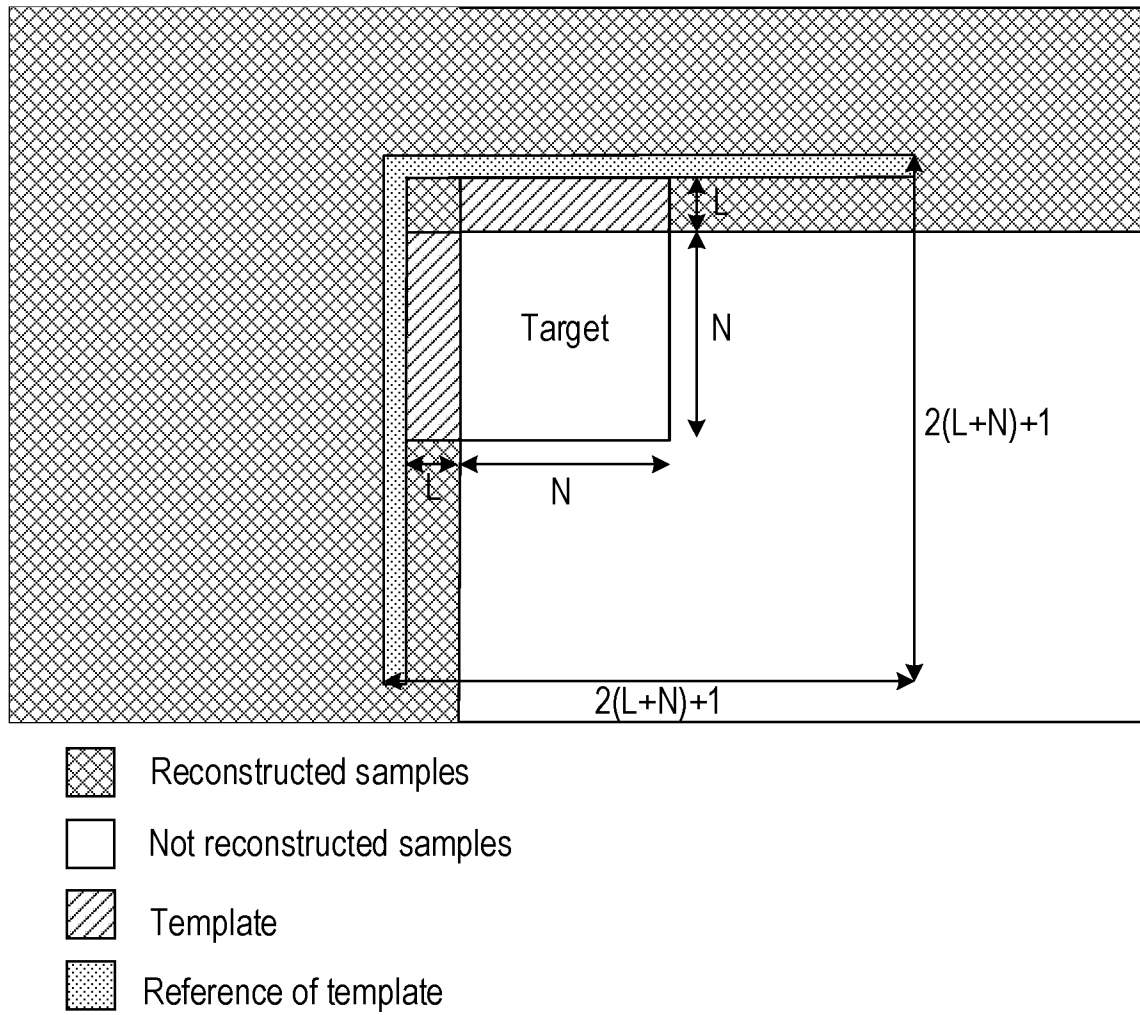


Fig. 19

**Fig. 20**

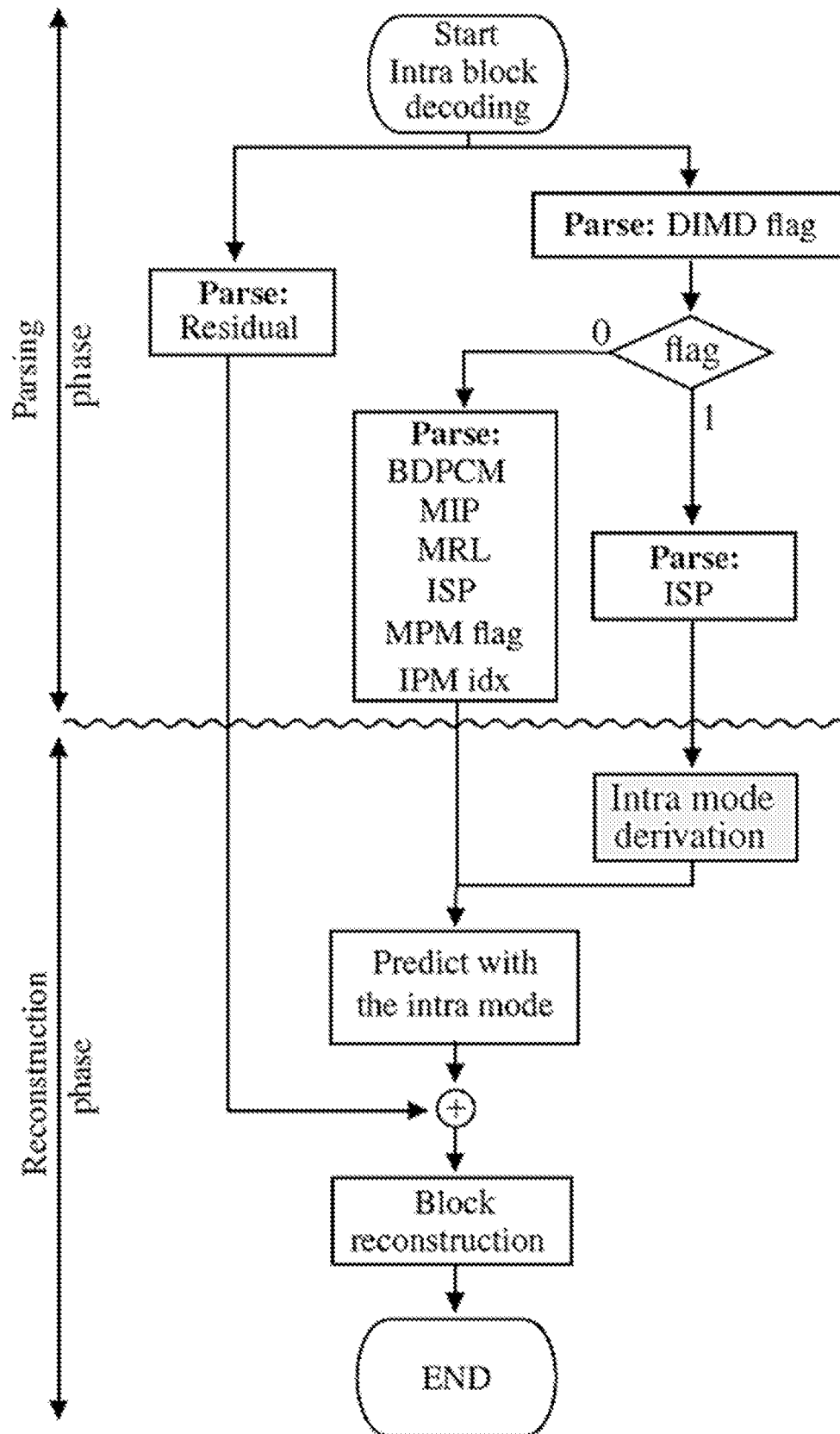


Fig. 21

- Current block
- Reconstructed area
- Unavailable area
- gray
- red
- yellow

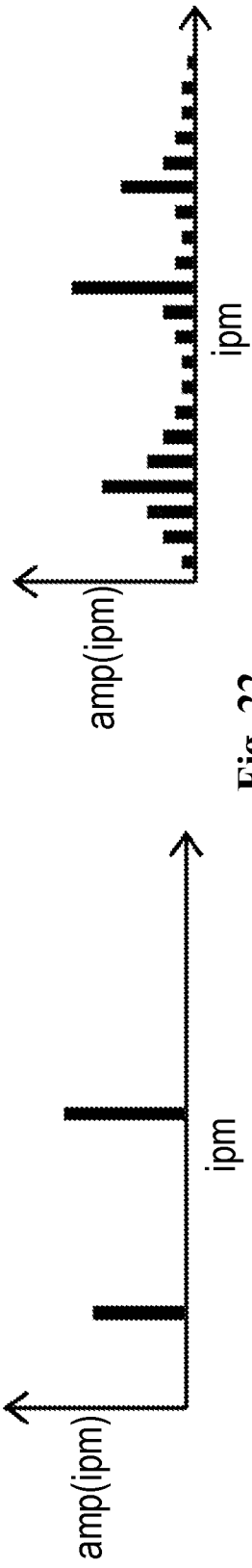
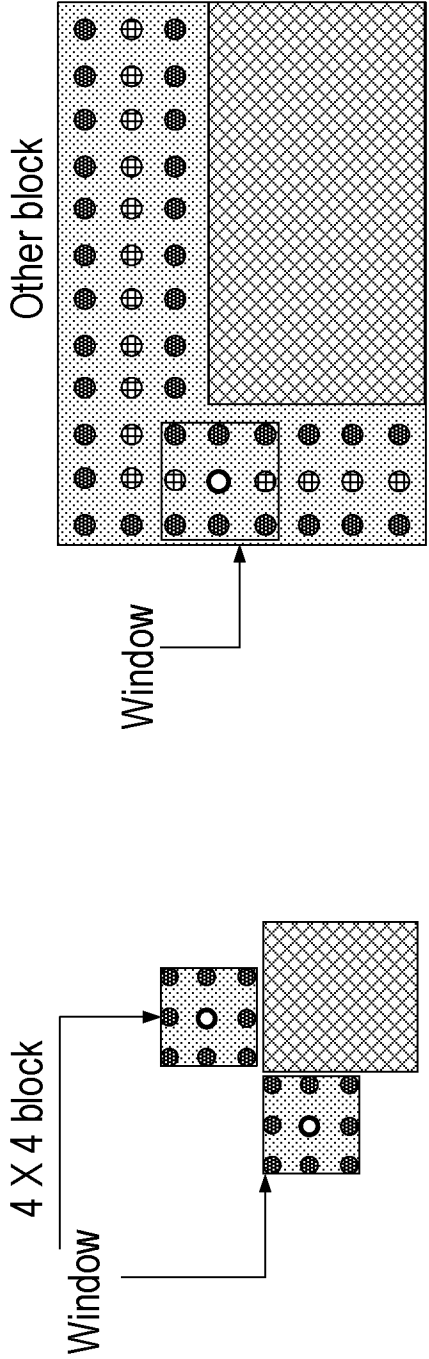
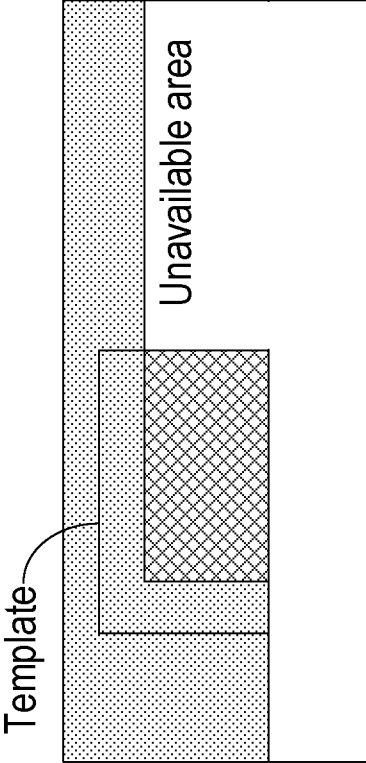


Fig. 22

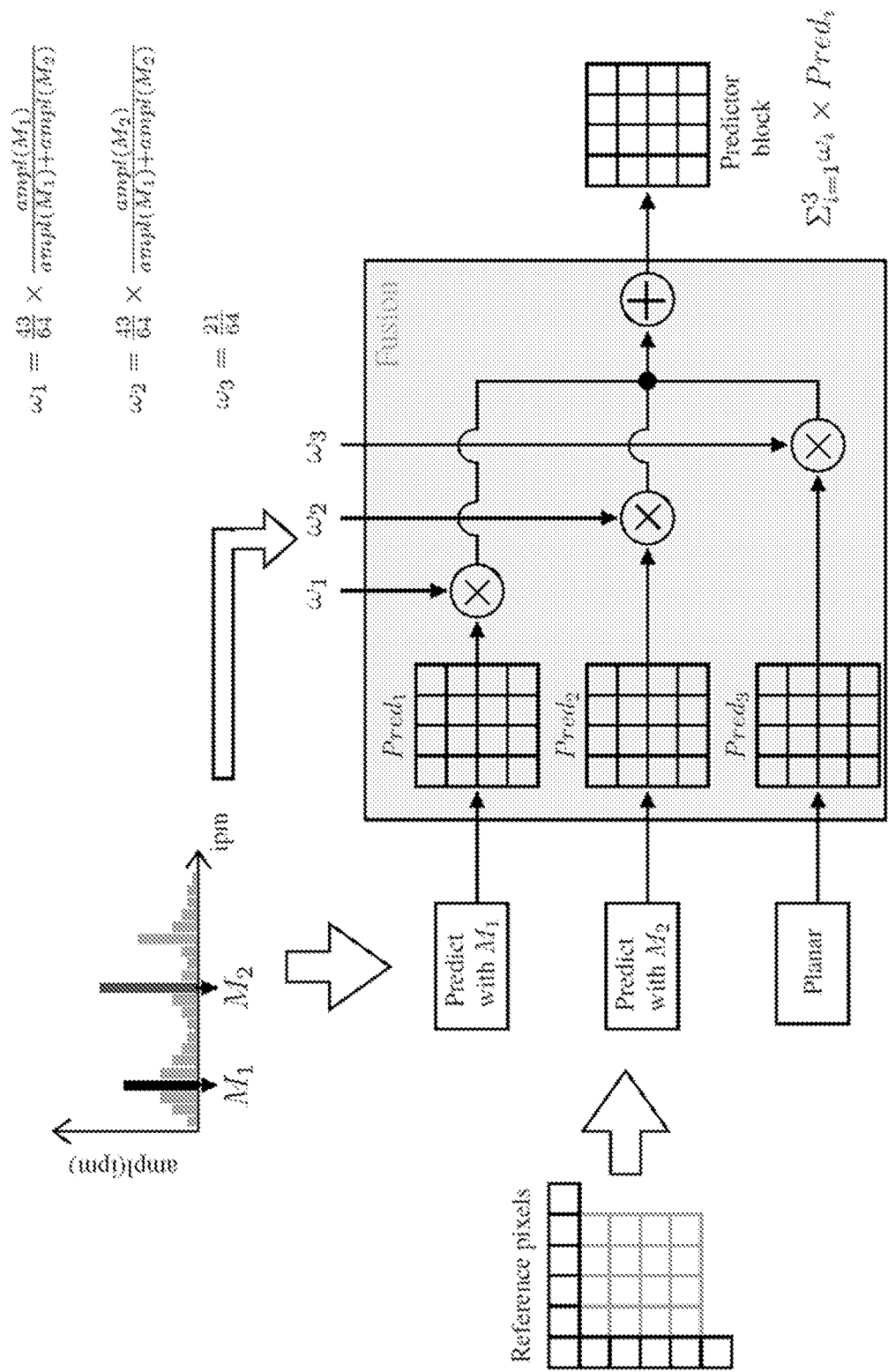


Fig. 23

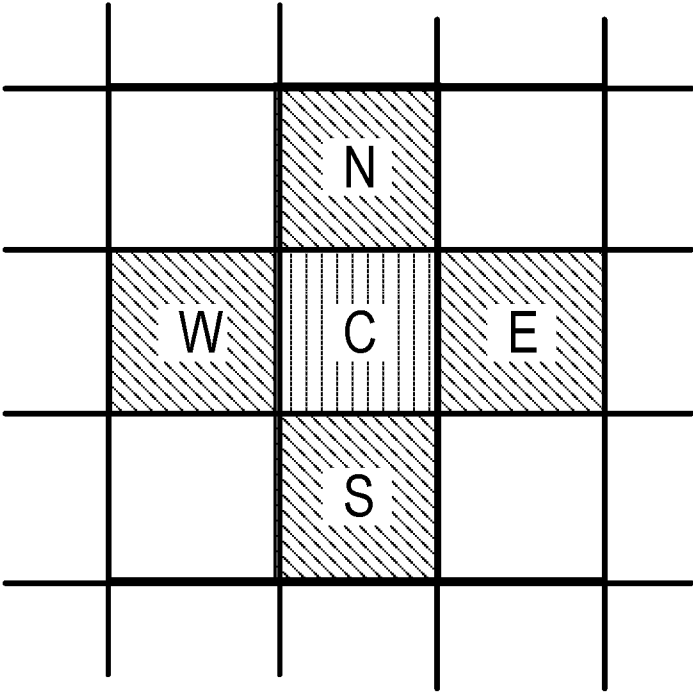


Fig. 24

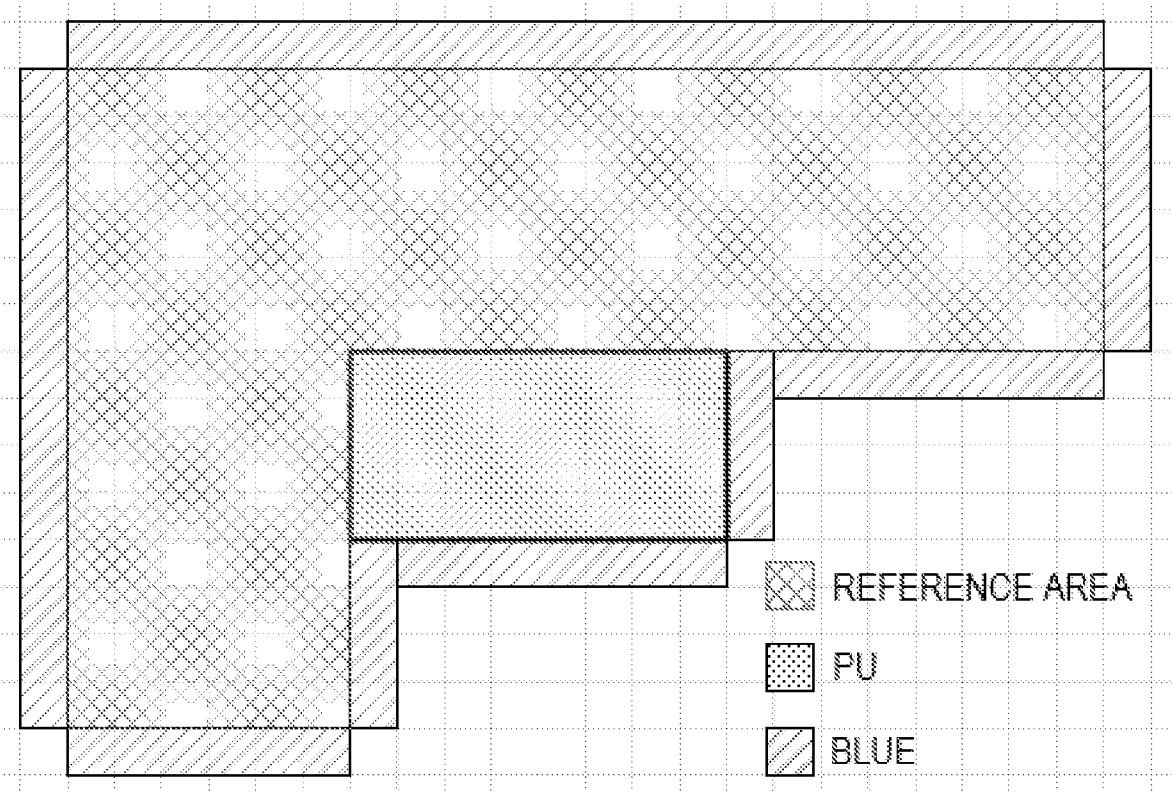


Fig. 25

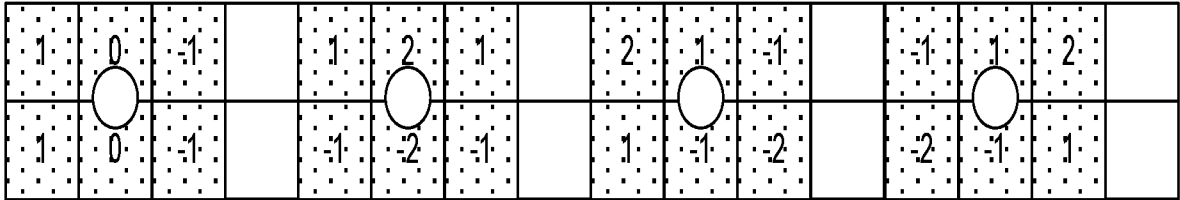


Fig. 26

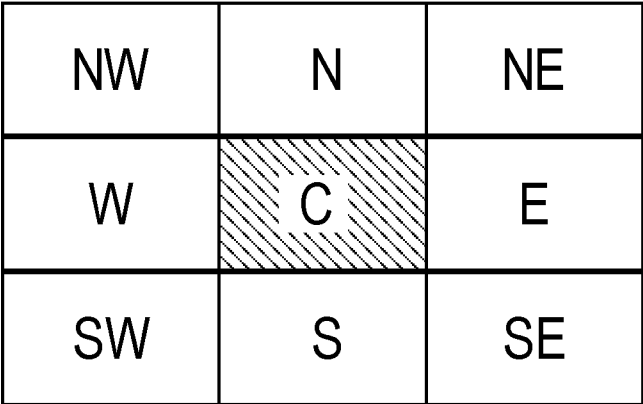


Fig. 27

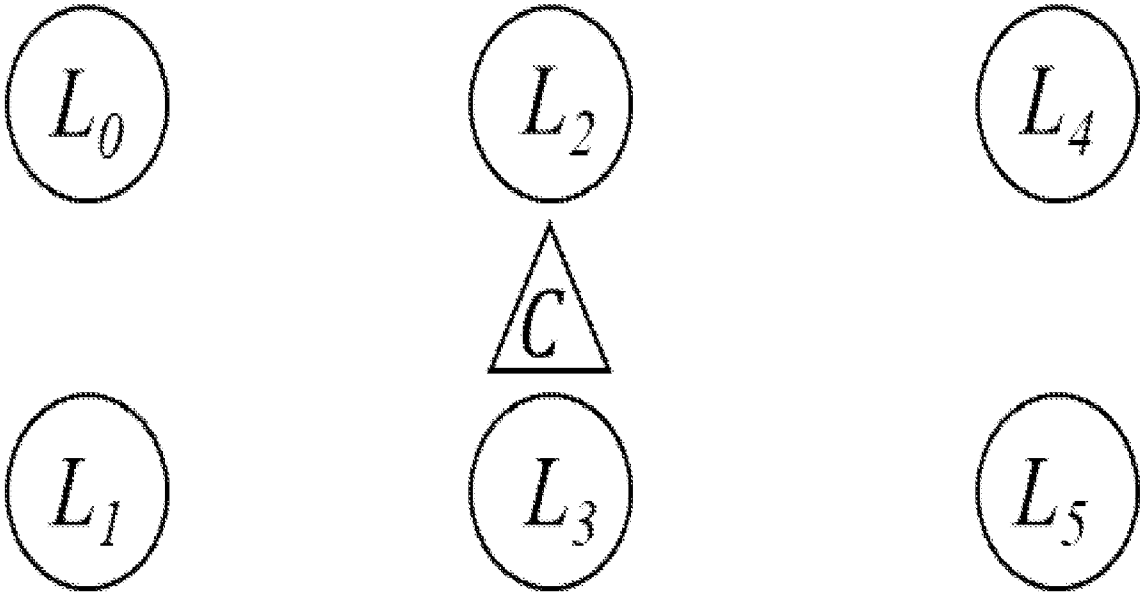


Fig. 28

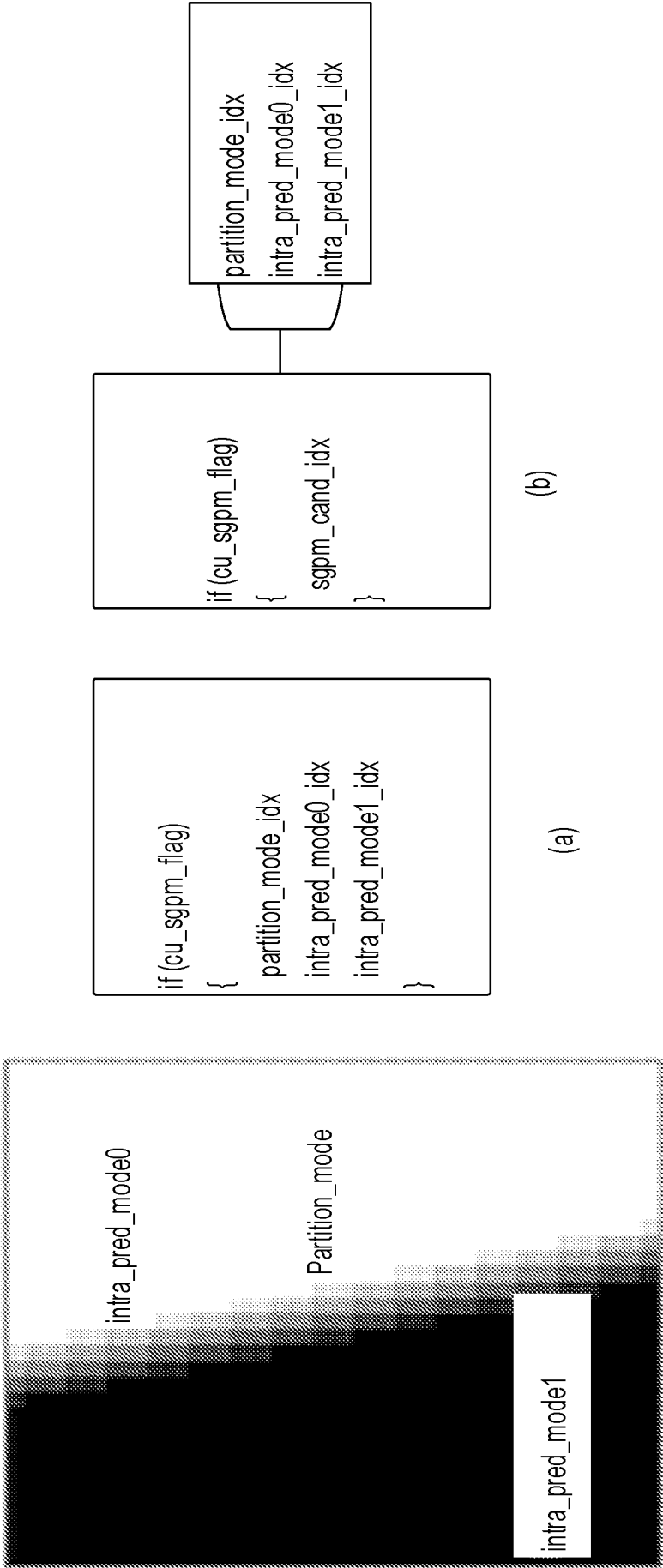


Fig. 29

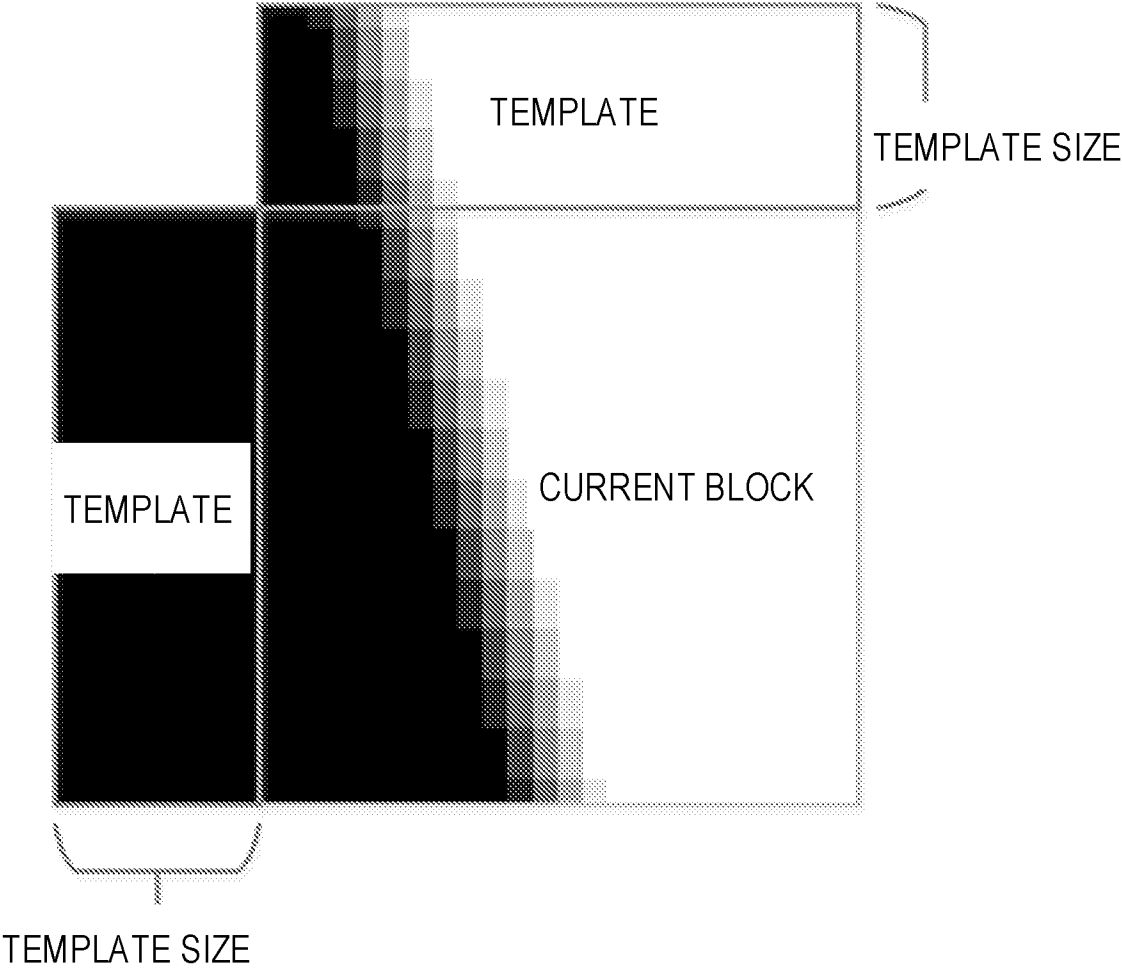


Fig. 30

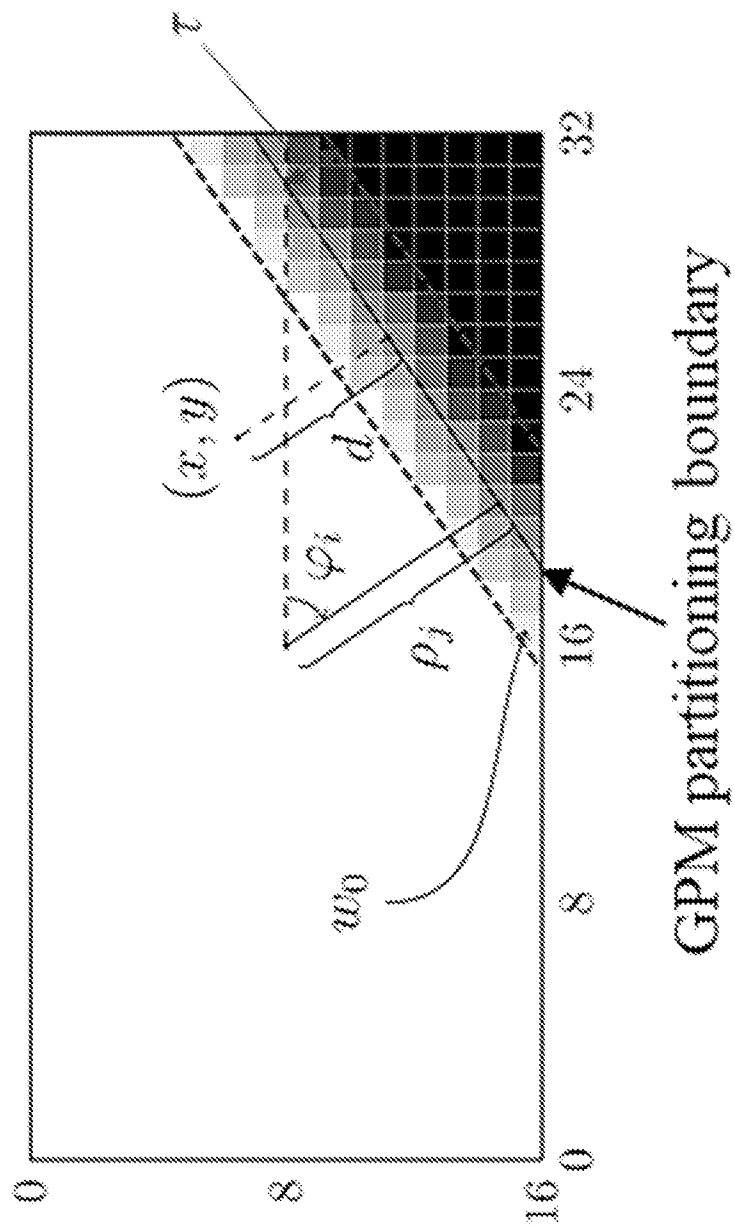
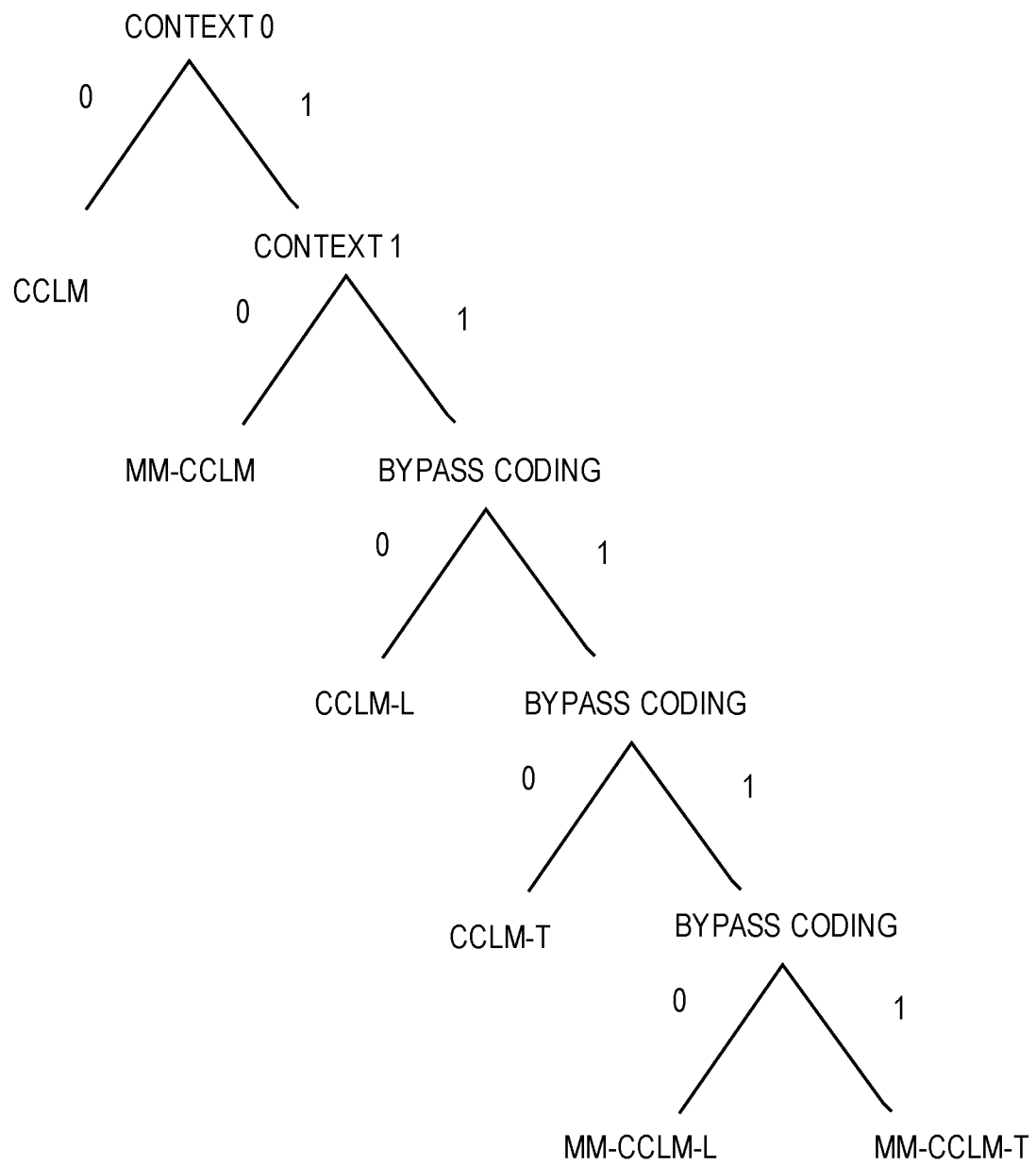


Fig. 31

**Fig. 32**

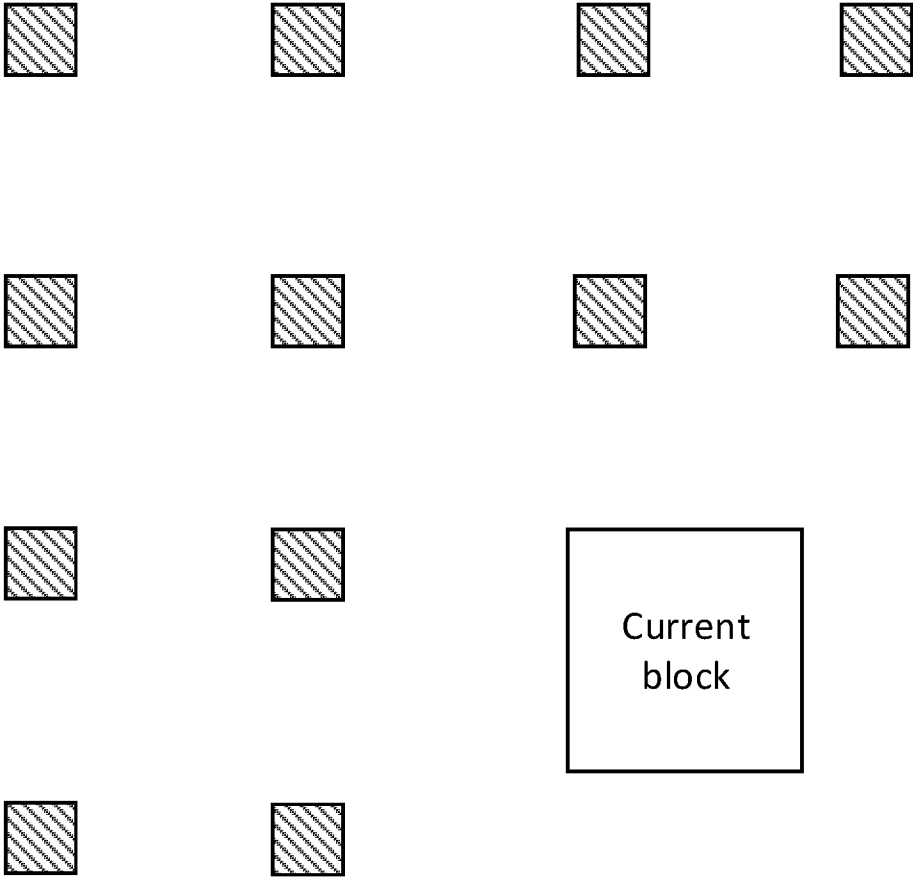


Fig. 33

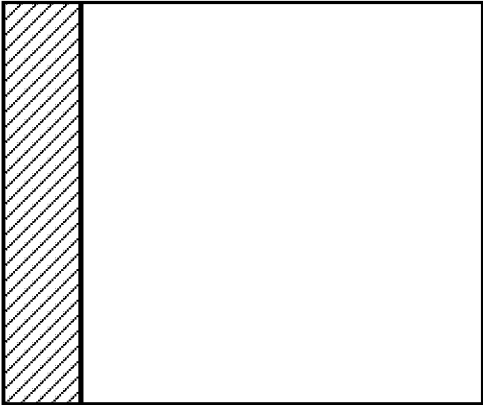


Fig. 34A

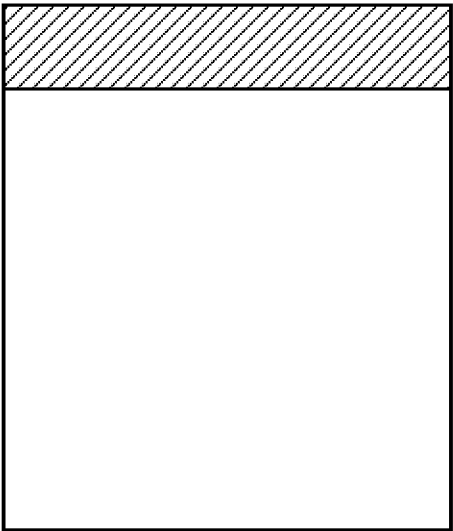


Fig. 34B

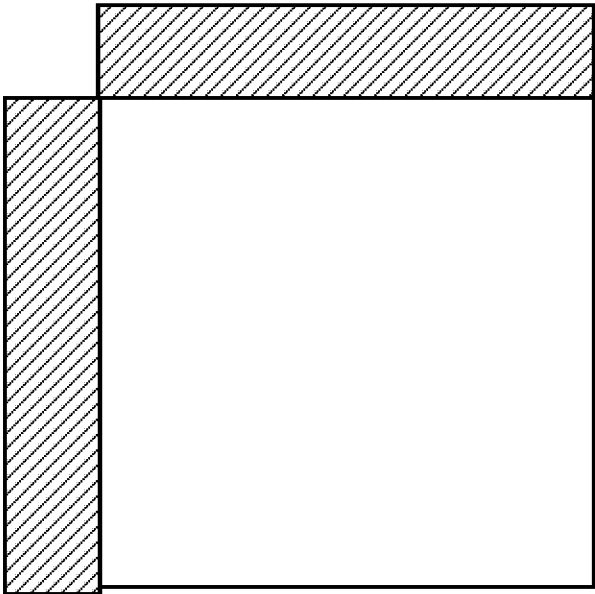


Fig. 34C

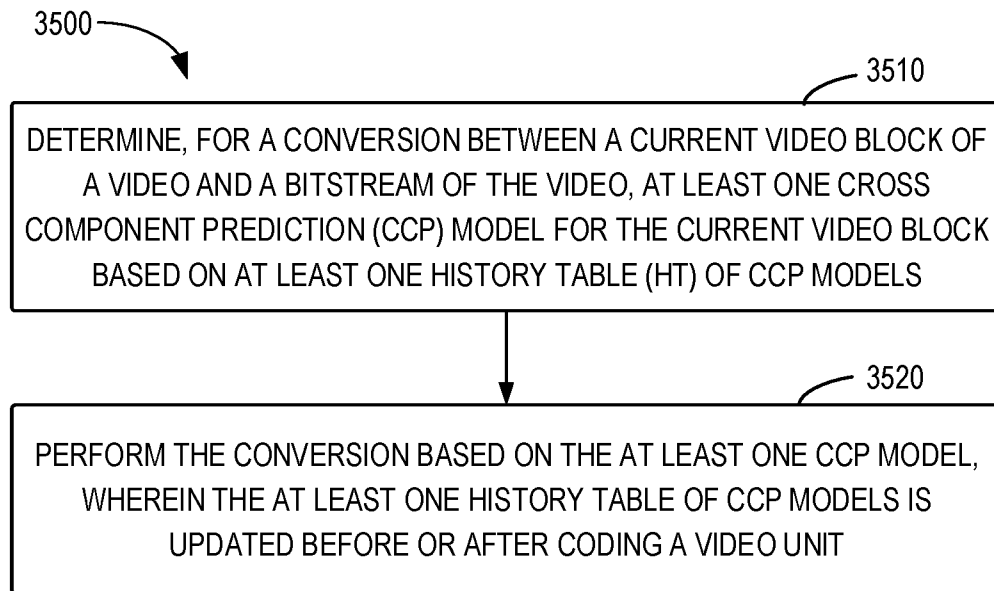


Fig. 35

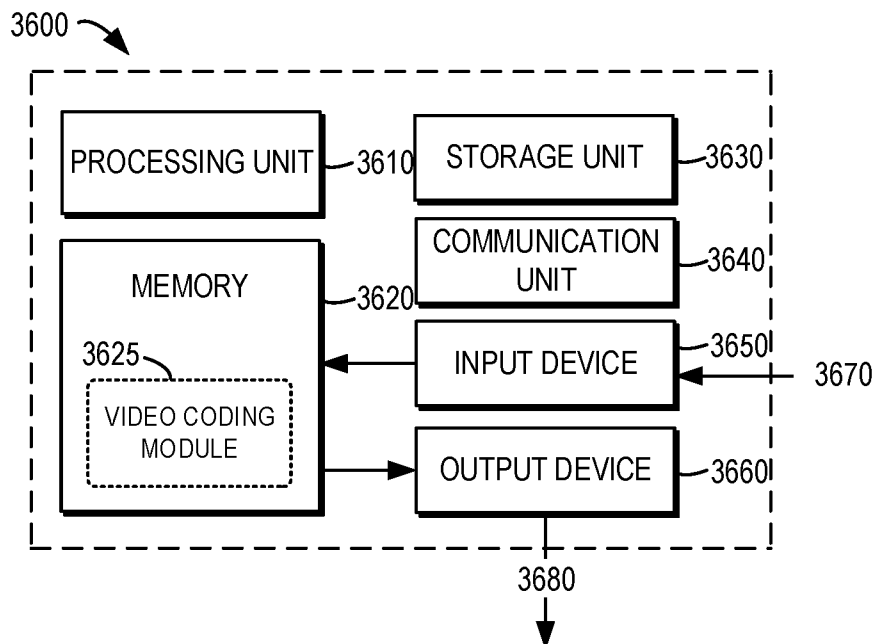


Fig. 36

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2024/010210

A. CLASSIFICATION OF SUBJECT MATTERIPC: **H04N 19/105** (2024.01); **H04N 19/176** (2024.01); **H04N 19/186** (2024.01); **H04N 19/11** (2024.01); **H04N 19/132** (2024.01); **H04N 19/52** (2024.01); **H04N 19/593** (2024.01)CPC: **H04N 19/105**; **H04N 19/176**; **H04N 19/186**; **H04N 19/11**; **H04N 19/132**; **H04N 19/52**; **H04N 19/593**

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)

See Search History Document

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

See Search History Document

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

See Search History Document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2021/0211654 A1 (BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD. et al.) 08 July 2021 (08.07.2021) entire document	1-3, 5, 6, 67, 68
Y	entire document	4
Y	US 2020/0169745 A1 (QUALCOMM INCORPORATED) 28 May 2020 (28.05.2020) entire document	4
A	BROSS et al. "Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC)." Proceedings of the IEEE 109.9 (2021): 1463-1493. Retrieved on 08 March 2024 (08.03.2024) from < https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9328514 > entire document	1-6, 67, 68
A	US 2022/0217368 A1 (BEIJING DAJIA INTERNET INFORMATION TECHNOLOGY CO., LTD.) 07 July 2022 (07.07.2022) entire document	1-6, 67, 68

☒ 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

08 March 2024 (08.03.2024)

Date of mailing of the international search report

19 March 2024 (19.03.2024)

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US

Commissioner for Patents

P.O. Box 1450, Alexandria, VA 22313-1450

Authorized officer

MATOS

TAINA

Facsimile No. 571-273-8300

Telephone No. 571-272-4300

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2024/010210

C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2021/0136405 A1 (BEIJING DAJIA INTERNET INFORMATION TECHNOLOGY CO., LTD.) 06 May 2021 (06.05.2021) entire document	1-6, 67, 68
A	US 2021/0152841 A1 (QUALCOMM INCORPORATED) 20 May 2021 (20.05.2021) entire document	1-6, 67, 68

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2024/010210

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☒ Claims Nos.: **7-66**
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).