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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
11 April 2024 (11.04.2024)

WIPO | PCT

(10) **International Publication Number**
WO 2024/074129 A1

(51) International Patent Classification:
H04N 19/176 (2014.01)

(21) International Application Number:
PCT/CN2023/123158

(22) International Filing Date:
07 October 2023 (07.10.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

63/378708	07 October 2022 (07.10.2022)	US
63/380968	26 October 2022 (26.10.2022)	US

(71) Applicant: MEDIATEK INC. [CN/CN]; No. 1, Dusing 1st Rd., Hsinchu Science Park, Hsinchu City, Taiwan 30078 (CN).

(72) Inventors: TSAI, Chia-Ming; No. 1, Dusing 1st Rd., Hsinchu Science Park, Hsinchu City, Taiwan 30078 (CN). **HSU, Chih-Wei**; No. 1, Dusing 1st Rd., Hsinchu Science Park, Hsinchu City, Taiwan 30078 (CN). **CHEN, Yi-Wen**; 2840 Junction Ave, San Jose, California 95134 (US). **CHEN, Ching-Yeh**; No. 1, Dusing 1st Rd., Hsinchu Science Park, Hsinchu City, Taiwan 30078 (CN). **CHUANG, Tzu-Der**; No. 1, Dusing 1st Rd., Hsinchu Science Park, Hsinchu City, Taiwan 30078 (CN). **CHUANG, Cheng-Yen**; No. 1, Dusing 1st Rd., Hsinchu Science Park, Hsinchu City, Taiwan 30078 (CN).

(74) Agent: BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.; 16th Fl., Block A, Corporate Square, No. 35 Jinrong Street, Xicheng District, Beijing 100033 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, 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,

(54) Title: METHOD AND APPARATUS OF INHERITING TEMPORAL NEIGHBOURING MODEL PARAMETERS IN VIDEO CODING SYSTEM

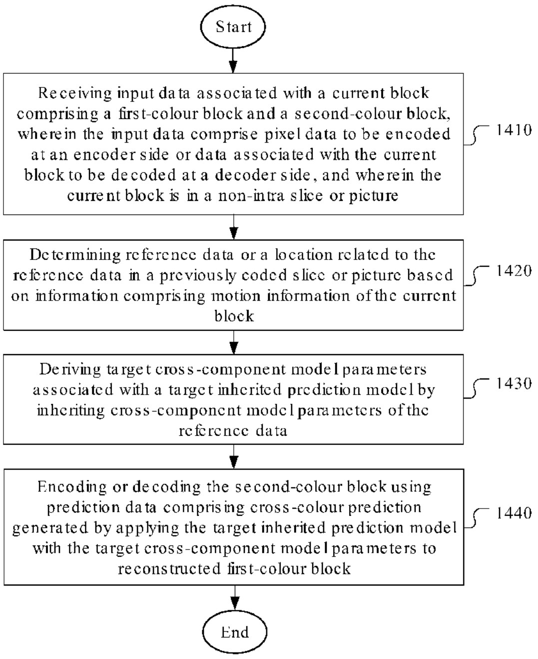

Fig. 14

(57) Abstract: A method and apparatus for coding colour pictures using cross-component models. Input data associated with a current block comprising a first-colour block and a second-colour block are received, wherein the input data comprise pixel data to be encoded at an encoder side or coded data associated with the current block to be decoded at a decoder side, and wherein the current block is in a non-intra slice/picture. Reference data or a location related to the reference data in a previously coded slice/picture is determined based on information comprising motion information of the current block. Target cross-component model parameters associated with a target inherited prediction model are derived by inheriting cross-component model parameters of the reference block. The second-colour block is encoded or decoded using prediction data comprising cross-colour prediction generated by applying the target inherited prediction model with the target cross-component model parameters to reconstructed first-colour block.

WO 2024/074129 A1

[Continued on next page]

DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

METHOD AND APPARATUS OF INHERITING TEMPORAL NEIGHBOURING MODEL PARAMETERS IN VIDEO CODING SYSTEM

CROSS REFERENCE TO RELATED APPLICATIONS

The present invention is a non-Provisional Application of and claims priority to U.S. Provisional Patent Application No. 63/378,708, filed on October 7, 2022 and U.S. Provisional Patent Application No. 63/380,968, filed on October 26, 2022. The U.S. Provisional Patent Applications are hereby incorporated by reference in their entireties.

FIELD OF THE INVENTION

The present invention relates to video coding system. In particular, the present invention relates to inheriting temporal neighbouring model parameters for cross-component prediction related modes in a video coding system.

BACKGROUND AND RELATED ART

Versatile video coding (VVC) is the latest international video coding standard developed by the Joint Video Experts Team (JVET) of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The standard has been published as an ISO standard: ISO/IEC 23090-3:2021, Information technology - Coded representation of immersive media - Part 3: Versatile video coding, published Feb. 2021. VVC is developed based on its predecessor HEVC (High Efficiency Video Coding) by adding more coding tools to improve coding efficiency and also to handle various types of video sources including 3-dimensional (3D) video signals.

Fig. 1A illustrates an exemplary adaptive Inter/Intra video encoding system incorporating loop processing. For Intra Prediction, the prediction data is derived based on previously encoded video data in the current picture. For Inter Prediction 112, Motion Estimation (ME) is performed at the encoder side and Motion Compensation (MC) is performed based on the result of ME to provide prediction data derived from other picture(s) and motion data. Switch 114 selects Intra Prediction 110 or Inter-Prediction 112 and the selected prediction data is supplied to Adder 116 to form prediction errors, also called residues. The prediction error is then processed by Transform (T) 118 followed by Quantization (Q) 120. The transformed and quantized residues are then coded by Entropy Encoder 122 to be included in a video bitstream corresponding to the compressed video data. The bitstream associated with the transform coefficients is then packed with side information such as motion and coding modes associated with Intra prediction and Inter prediction, and other information such as parameters associated with loop filters applied to underlying image area. The side information associated with Intra Prediction 110, Inter prediction 112 and in-loop filter 130, are provided to

Entropy Encoder 122 as shown in Fig. 1A. When an Inter-prediction mode is used, a reference picture or pictures have to be reconstructed at the encoder end as well. Consequently, the transformed and quantized residues are processed by Inverse Quantization (IQ) 124 and Inverse Transformation (IT) 126 to recover the residues. The residues are then added back to prediction data 136 at Reconstruction (REC) 128 to reconstruct video data. The reconstructed video data may be stored in Reference Picture Buffer 134 and used for prediction of other frames.

As shown in Fig. 1A, incoming video data undergoes a series of processing in the encoding system. The reconstructed video data from REC 128 may be subject to various impairments due to a series of processing. Accordingly, in-loop filter 130 is often applied to the reconstructed video data before the reconstructed video data are stored in the Reference Picture Buffer 134 in order to improve video quality. For example, deblocking filter (DF), Sample Adaptive Offset (SAO) and Adaptive Loop Filter (ALF) may be used. The loop filter information may need to be incorporated in the bitstream so that a decoder can properly recover the required information. Therefore, loop filter information is also provided to Entropy Encoder 122 for incorporation into the bitstream. In Fig. 1A, Loop filter 130 is applied to the reconstructed video before the reconstructed samples are stored in the reference picture buffer 134. The system in Fig. 1A is intended to illustrate an exemplary structure of a typical video encoder. It may correspond to the High Efficiency Video Coding (HEVC) system, VP8, VP9, H.264 or VVC.

The decoder, as shown in Fig. 1B, can use similar or portion of the same functional blocks as the encoder except for Transform 118 and Quantization 120 since the decoder only needs Inverse Quantization 124 and Inverse Transform 126. Instead of Entropy Encoder 122, the decoder uses an Entropy Decoder 140 to decode the video bitstream into quantized transform coefficients and needed coding information (e.g. ILPF information, Intra prediction information and Inter prediction information). The Intra prediction 150 at the decoder side does not need to perform the mode search. Instead, the decoder only needs to generate Intra prediction according to Intra prediction information received from the Entropy Decoder 140. Furthermore, for Inter prediction, the decoder only needs to perform motion compensation (MC 152) according to Inter prediction information received from the Entropy Decoder 140 without the need for motion estimation.

According to VVC, an input picture is partitioned into non-overlapped square block regions referred as CTUs (Coding Tree Units), similar to HEVC. Each CTU can be partitioned into one or multiple smaller size coding units (CUs). The resulting CU partitions can be in square or rectangular shapes. Also, VVC divides a CTU into prediction units (PUs) as a unit to apply prediction process, such as Inter prediction, Intra prediction, etc.

The VVC standard incorporates various new coding tools to further improve the coding efficiency over the HEVC standard. Among various new coding tools, some coding tools relevant to the present invention are reviewed as follows. For example, 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.

Various cross-component prediction modes are being considered for the emerging new video coding standard. Inheriting model parameters for these cross-component prediction modes is an effective way to reduce data rate related to signalling model parameters. In the present invention, schemes to improve the coding efficiency associated with inheriting model parameters for cross-component prediction modes are disclosed.

BRIEF SUMMARY OF THE INVENTION

A method and apparatus for coding colour pictures using coding tools including one or more cross component models related modes are disclosed. Input data associated with a current block comprising a first-colour block and a second-colour block are received, wherein the input data comprise pixel data to be encoded at an encoder side or data associated with the current block to be decoded at a decoder side, and wherein the current block is in a non-intra slice or picture. Reference data or a location related to the reference data in a previously coded slice or picture is determined based on information comprising motion information of the current block. Target cross-component model parameters associated with a target inherited prediction model are derived by inheriting cross-component model parameters of the reference data. The second-colour block is encoded or decoded using prediction data comprising cross-colour prediction generated by applying the target inherited prediction model with the target cross-component model parameters to reconstructed first-colour block. The first-colour block may correspond to a luma block and the second-colour block may correspond to a chroma block.

In one embodiment, the reference data in the previously coded slice/picture is located according to a corresponding location of the current block in the previously coded slice/picture and a motion vector of the current block. In one embodiment, when the current block is intra coded, the motion vector used in deriving the target cross-component model parameters of the current block is set to 0.

In one embodiment, the location of the reference data is determined based on the corresponding location of the current block shifted by the motion vector of the current block. In another embodiment, the location of the reference data is determined based on the corresponding location of the current block shifted by the motion vector of the current block with one or more additional offsets. In one embodiment, said one or more additional offsets depend on width of the current block, height of the current block, or both. In one embodiment, said one or more additional offsets correspond to a horizontal offset by the width of the current block and a vertical offset by the height of the current block. In another embodiment, said one or more additional offsets correspond to a horizontal offset by half of the width of the current block and a vertical offset by half of the height of the current block.

In one embodiment, one or more of the cross-component model parameters of the reference data are refined and then used as the target cross-component model parameters.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1A illustrates an exemplary adaptive Inter/Intra video encoding system incorporating loop processing.

Fig. 1B illustrates a corresponding decoder for the encoder in Fig. 1A.

Fig. 2 illustrates examples of a multi-type tree structure corresponding to vertical binary splitting (SPLIT_BT_VER), horizontal binary splitting (SPLIT_BT_HOR), vertical ternary splitting (SPLIT_TT_VER), and horizontal ternary splitting (SPLIT_TT_HOR).

Fig. 3 illustrates an example of the signalling mechanism of the partition splitting information in quadtree with nested multi-type tree coding tree structure.

Fig. 4 shows an example of a CTU divided into multiple CUs with a quadtree and nested multi-type tree coding block structure, where the bold block edges represent quadtree partitioning and the remaining edges represent multi-type tree partitioning.

Fig. 5 shows the intra prediction modes as adopted by the VVC video coding standard.

Fig. 6 shows an example of the location of the left and above samples and the sample of the current block involved in the LM_LA mode.

Fig. 7 shows an example of classifying the neighbouring samples into two groups according to multiple mode CCLM.

Fig. 8 illustrates an example of spatial part of the convolutional filter.

Fig. 9 illustrates an example of reference area with paddings used to derive the filter coefficients.

Fig. 10 illustrates the 16 gradient patterns for Gradient Linear Model (GLM).

Fig. 11 illustrates an example of inheriting temporal neighbouring model parameters.

Fig. 12 illustrates an example of neighbouring templates for calculating model error.

Fig. 13 illustrates an example of neighbouring templates for calculating model error.

Fig. 14 illustrates a flowchart of an exemplary video coding system that incorporates inheriting temporal cross-component model parameters according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

It will be readily understood that the components of the present invention, as generally described and illustrated in the figures herein, may be arranged and designed in a wide variety of different configurations. Thus, the following more detailed description of the embodiments of the systems and methods of the present invention, as represented in the figures, is not intended to limit the scope of the invention, as claimed, but is merely representative of selected embodiments of the invention. References throughout this specification to “one embodiment,” “an embodiment,” or similar language mean that a particular feature, structure, or characteristic described in connection with the embodiment may be included in at least one embodiment of the present invention. Thus, appearances of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment.

Furthermore, the described features, structures, or characteristics may be combined in any suitable manner in one or more embodiments. One skilled in the relevant art will recognize, however, that the invention can be practiced without one or more of the specific details, or with other methods, components, etc. In other instances, well-known structures, or operations are not shown or described in detail to avoid obscuring aspects of the invention. The illustrated embodiments of the invention will be best understood by reference to the drawings, wherein like parts are designated by like numerals throughout. The following description is intended only by way of example, and simply illustrates certain selected embodiments of apparatus and methods that are consistent with the invention as claimed herein.

Partitioning of the CTUs Using a Tree Structure

In HEVC, a CTU is split into CUs by using a quaternary-tree (QT) structure denoted as coding tree to adapt to various local characteristics. The decision whether to code a picture area using inter-picture (temporal) or intra-picture (spatial) prediction is made at the leaf CU level. Each leaf CU can be further split into one, two or four Pus according to the PU splitting type. Inside one PU, the same prediction process is applied and the relevant information is transmitted to the decoder on a PU basis. After obtaining the residual block by applying the prediction process based on the PU splitting type, a leaf CU can be partitioned into transform units (TUs) according to another quaternary-tree structure similar to the coding tree for the CU. One of key feature of the HEVC structure is that it has the multiple partition conceptions including CU, PU, and TU.

In VVC, a quadtree with nested multi-type tree using binary and ternary splits segmentation structure replaces the concepts of multiple partition unit types, i.e. it removes the separation of the CU, PU and TU concepts except as needed for CUs that have a size too large for the maximum transform length, and supports more flexibility for CU partition shapes. In the coding tree structure, a CU can have either a square or rectangular shape. A coding tree unit (CTU) is first partitioned by a quaternary tree (a.k.a. quadtree) structure. Then the quaternary tree leaf nodes can be further partitioned by a multi-type tree structure. As shown in Fig. 2, there are four splitting types in multi-type tree structure, vertical binary splitting (SPLIT_BT_VER 210), horizontal binary splitting (SPLIT_BT_HOR 220), vertical ternary splitting (SPLIT_TT_VER 230), and horizontal ternary splitting (SPLIT_TT_HOR 240). The multi-type tree leaf nodes are called coding units (CUs), and unless the CU is too large for the maximum transform length, this segmentation is used for prediction and transform processing without any further partitioning. This means that, in most cases, the CU, PU and TU have the same block size in the quadtree with nested multi-type tree coding block structure. The exception occurs when maximum supported transform length is smaller than the width or height of the colour component of the CU.

Fig. 3 illustrates the signalling mechanism of the partition splitting information in quadtree with nested multi-type tree coding tree structure. A coding tree unit (CTU) is treated as the root of a quaternary tree and is first partitioned by a quaternary tree structure. Each quaternary tree leaf node (when sufficiently large to allow it) is then further partitioned by a multi-type tree structure. In quadtree with nested multi-type tree coding tree structure, for each CU node, a first flag (split_cu_flag) is signalled to indicate whether the node is further partitioned. If the current CU node is a quadtree CU node, a second flag (split_qt_flag) whether it's a QT partitioning or MTT partitioning mode. When a node is partitioned with MTT partitioning mode, a third flag (mtt_split_cu_vertical_flag) is signalled to indicate the splitting direction, and then a fourth flag (mtt_split_cu_binary_flag) is signalled to indicate whether the split is a binary split or a ternary split. Based on the values of mtt_split_cu_vertical_flag and mtt_split_cu_binary_flag, the multi-type tree slitting mode (MttSplitMode) of a CU is derived as shown in Table 1.

Table 1 – MttSplitMode derivation based on multi-type tree syntax elements

MttSplitMode	mtt_split_cu_vertical_flag	mtt_split_cu_binary_flag
SPLIT_TT_HOR	0	0
SPLIT_BT_HOR	0	1
SPLIT_TT_VER	1	0
SPLIT_BT_VER	1	1

Fig. 4 shows a CTU divided into multiple CUs with a quadtree and nested multi-type tree coding block structure, where the bold block edges represent quadtree partitioning and the remaining edges represent multi-type tree partitioning. The quadtree with nested multi-type tree partition provides a content-adaptive coding tree structure comprised of CUs. The size of the CU may be as large as the CTU or as small as 4×4 in units of luma samples. For the case of the 4:2:0 chroma format, the maximum chroma CB size is 64×64 and the minimum size chroma CB consist of 16 chroma samples.

In VVC, the maximum supported luma transform size is 64×64 and the maximum supported chroma transform size is 32×32. When the width or height of the CB is larger the maximum transform width or height, the CB is automatically split in the horizontal and/or vertical direction to meet the transform size restriction in that direction.

The following parameters are defined for the quadtree with nested multi-type tree coding tree scheme. These parameters are specified by SPS (Sequence Parameter Set) syntax elements and can be further refined by picture header syntax elements.

- CTU size: the root node size of a quaternary tree

- *MinQTSize*: the minimum allowed quaternary tree leaf node size
- *MaxBtSize*: the maximum allowed binary tree root node size
- *MaxTtSize*: the maximum allowed ternary tree root node size
- *MaxMttDepth*: the maximum allowed hierarchy depth of multi-type tree splitting from a quadtree leaf
- *MinCbSize*: the minimum allowed coding block node size

In one example of the quadtree with nested multi-type tree coding tree structure, the CTU size is set as 128×128 luma samples with two corresponding 64×64 blocks of 4:2:0 chroma samples, the *MinQTSize* is set as 16×16, the *MaxBtSize* is set as 128×128 and *MaxTtSize* is set as 64×64, the *MinCbsize* (for both width and height) is set as 4×4, and the *MaxMttDepth* is set as 4. The quaternary tree partitioning is applied to the CTU first to generate quaternary tree leaf nodes. The quaternary tree leaf nodes may have a size from 16×16 (i.e., the *MinQTSize*) to 128×128 (i.e., the CTU size). If the leaf QT node is 128×128, it will not be further split by the binary tree since the size exceeds the *MaxBtSize* and *MaxTtSize* (i.e., 64×64). Otherwise, the leaf qdtree node can be further partitioned by the multi-type tree. Therefore, the quaternary tree leaf node is also the root node for the multi-type tree and it has multi-type tree depth (mttDepth) as 0. When the multi-type tree depth reaches *MaxMttDepth* (i.e., 4), no further splitting is considered. When the multi-type tree node has width equal to *MinCbsize*, no further horizontal splitting is considered. Similarly, when the multi-type tree node has height equal to *MinCbsize*, no further vertical splitting is considered.

In VVC, the coding tree scheme supports the ability for the luma and chroma to have a separate block tree structure. For P and B slices, the luma and chroma CTBs in one CTU have to share the same coding tree structure. However, for I slices, the luma and chroma can have separate block tree structures. When the separate block tree mode is applied, luma CTB is partitioned into CUs by one coding tree structure, and the chroma CTBs are partitioned into chroma CUs by another coding tree structure. This means that a CU in an I slice may consist of a coding block of the luma component or coding blocks of two chroma components, and a CU in a P or B slice always consists of coding blocks of all three colour components unless the video is monochrome.

Intra Mode Coding with 67 Intra Prediction Modes

To capture the arbitrary edge directions presented in natural video, the number of directional intra modes in VVC is extended from 33, as used in HEVC, to 65. The new directional modes not in HEVC are depicted as dotted arrows in Fig. 5, and the planar and DC modes remain the same. These denser directional intra prediction modes apply for all block sizes and for both luma and chroma intra predictions.

In VVC, several conventional angular intra prediction modes are adaptively replaced with wide-angle intra prediction modes for the non-square blocks.

In 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.

To keep the complexity of the most probable mode (MPM) list generation low, an intra mode coding method with 6 MPMs is used by considering two available neighbouring intra modes. The following three aspects are considered to construct the MPM list:

- Default intra modes
- Neighbouring intra modes
- Derived intra modes.

A unified 6-MPM list is used for intra blocks irrespective of whether MRL and ISP coding tools are applied or not. The MPM list is constructed based on intra modes of the left and above neighbouring block. Suppose the mode of the left is denoted as Left and the mode of the above block is denoted as *Above*, the unified MPM list is constructed as follows:

- When a neighbouring block is not available, its intra mode is set to Planar by default.
- If both modes Left and Above are non-angular modes:
  - MPM list → {Planar, DC, V, H, V – 4, V + 4}
- If one of modes Left and Above is angular mode, and the other is non-angular:
  - Set a mode Max as the larger mode in Left and Above
  - MPM list → {Planar, Max, Max – 1, Max + 1, Max – 2, M + 2}
- If Left and Above are both angular and they are different:
  - Set a mode Max as the larger mode in Left and Above
  - If Max – Min is equal to 1:
    - MPM list → {Planar, Left, Above, Min – 1, Max + 1, Min – 2}
  - Otherwise, if Max – Min is greater than or equal to 62:
    - MPM list → {Planar, Left, Above, Min + 1, Max – 1, Min + 2}
  - Otherwise, if Max – Min is equal to 2:
    - MPM list → {Planar, Left, Above, Min + 1, Min – 1, Max + 1}
  - Otherwise:
    - MPM list → {Planar, Left, Above, Min – 1, –Min + 1, Max – 1}
- If Left and Above are both angular and they are the same:
  - MPM list → {Planar, Left, Left – 1, Left + 1, Left – 2, Left + 2}

Besides, the first bin of the MPM index codeword is CABAC context coded. In total three contexts are used, corresponding to whether the current intra block is MRL enabled, ISP enabled, or a normal intra block.

During 6 MPM list generation process, pruning is used to remove duplicated modes so that only unique modes can be included into the MPM list. For entropy coding of the 61 non-MPM modes, a Truncated Binary Code (TBC) is used.

Cross-Component Linear Model (CCLM) 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:

$$\mathrm{pred_C}(i,j) = \alpha \cdot \mathrm{rec_L}'(i,j) + \beta \qquad (1)$$

where $\mathrm{pred_C}(i,j)$ represents the predicted chroma samples in a CU and $\mathrm{rec_L}'(i,j)$ represents the downsampled 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×H, then W' and H' are set as

- W' = W, H' = H when LM_LA mode is applied;
- W' =W + H when LM_A mode is applied;
- H' = H + W when LM_L mode is applied.

The above neighbouring positions are denoted as S[0, −1]...S[W' − 1, −1] and the left neighbouring positions are denoted as S[−1, 0]...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-A 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^0_A and x^1_A, and two smaller values: x^0_B and x^1_B. Their corresponding chroma sample values are denoted as y^0_A, y^1_A, y^0_B and y^1_B. Then x_A, x_B, y_A and y_B are derived as:

$$Xa=(x0A + x1A +1)>>1,$$

$$Xb=(x0B + x1B +1)>>1;$$

$$Ya=(y0A + y1A +1)>>1;$$

$$Yb=(y0B + y1B +1)>>1 \qquad (2)$$

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

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

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

Fig. 6 shows an example of the location of the left and above samples and the sample of the current block involved in the LM_LA mode. Fig. 6 shows the relative sample locations of $N \times N$ chroma block 610, the corresponding $2N \times 2N$ luma block 620 and their neighbouring samples (shown as filled circles).

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/diff is reduced into 16 elements for 16 values of the significand as follows:

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

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_A, and LM_L modes.

In LM_A 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 are used to calculate the linear model coefficients. To get more samples, the left template is extended to (H+W) samples.

In LM_LA 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.

$$\begin{aligned}\mathrm{Rec_L}'(i,j) = [&\mathrm{rec_L}(2i-1,2j-1) + 2\cdot\mathrm{rec_L}(2i-1,2j-1) + \mathrm{rec_L}(2i+1,2j-1) +\\ &\mathrm{rec_L}(2i-1,2j) + 2\cdot\mathrm{rec_L}(2i,2j) + \mathrm{rec_L}(2i+1,2j) + 4] \gg 3 \qquad (6)\end{aligned}$$

$$\begin{aligned}\mathrm{Rec_L}'(i,j) = &\mathrm{rec_L}(2i,2j-1) + \mathrm{rec_L}(2i-1,2j) + 4\cdot\mathrm{rec_L}(2i,2j) + \mathrm{rec_L}(2i+1,2j) +\\ &\mathrm{rec_L}(2i,2j+1) + 4] \gg 3 \qquad (7)\end{aligned}$$

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 traditional intra modes and three cross-component linear model modes (LM_LA, LM_A, and LM_L). Chroma mode signalling and derivation process are shown in Table 2. 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 centre position of the current chroma block is directly inherited.

Table -2 – Derivation of chroma prediction mode from luma mode when CCLM is allowed

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 3.

Table 3– Unified binarization table for chroma prediction mode

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

In Table 3, the first bin indicates whether it is regular (0) or CCLM modes (1). If it is LM mode, then the next bin indicates whether it is LM_LA (0) or not. If it is not LM_LA, next 1 bin indicates whether it is LM_L (0) or LM_A (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 3 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 64x64 luma coding tree node is partitioned with Not Split (and ISP is not used for the 64x64 CU) or QT, the chroma CUs in 32x32 / 32x16 chroma coding tree node are allowed to use CCLM in the following way:

- If the 32x32 chroma node is not split or partitioned QT split, all chroma CUs in the 32x32 node can use CCLM
- If the 32x32 chroma node is partitioned with Horizontal BT, and the 32x16 child node does not split or uses Vertical BT split, all chroma CUs in the 32x16 chroma node can use CCLM.

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

Multiple Model CCLM (MMLM)

In the JEM (J. Chen, E. Alshina, G. J. Sullivan, J.-R. Ohm, and J. Boyce, Algorithm Description of Joint Exploration Test Model 7, document JVET-G1001, ITU-T/ISO/IEC Joint Video Exploration Team (JVET), Jul. 2017), multiple model CCLM mode (MMLM) is proposed for using two models for predicting the chroma samples from the luma samples for the whole CU. In MMLM, neighbouring luma samples and neighbouring chroma samples of the current block are classified into two groups, each group is used as a training set to derive a linear model (i.e., a particular α and β are derived for a particular group). Furthermore, the samples of the current luma block are also classified based on the same rule for the classification of neighbouring luma samples.

Fig. 7 shows an example of classifying the neighbouring samples into two groups. *Threshold* is calculated as the average value of the neighbouring reconstructed luma samples. A neighbouring sample with $Rec'_L[x,y]$ <= *Threshold* is classified into group 1; while a neighbouring sample with $Rec'_L[x,y]$ > *Threshold* is classified into group 2.

$$\begin{cases} Pred_C[x,y] = \alpha_1 \times Rec'_L[x,y] + \beta_1 & if\ Rec'_L[x,y] \le Threshold \\ Pred_C[x,y] = \alpha_2 \times Rec'_L[x,y] + \beta_2 & if\ Rec'_L[x,y] > Threshold \end{cases} \quad (8)$$

Convolutional cross-component model (CCCM)

In CCCM, a convolutional model is applied to improve the chroma prediction performance. The convolutional model has 7-tap filter consisting 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 centre (C) luma sample which is collocated with the chroma sample to be predicted and its above/north (N), below/south (S), left/west (W) and right/east (E) neighbours as shown in Fig. 8.

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

$$P = (C{*}C + \mathrm{midVal}) >> \mathrm{bitDepth}.$$

For example, for 10-bit contents, the nonlinear term is calculated as:

$$P = (C{*}C + 512) >> 10$$

The bias term (denoted as B) represents a scalar offset between the input and output (similarly to the offset term in CCLM) and is set to the 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 values and clipped to the range of valid chroma samples:

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

The filter coefficients c_i are calculated by minimising MSE between predicted and reconstructed chroma samples in the reference area. Fig. 9 illustrates an example of the reference area which consists of 6 lines of chroma samples above and left of the PU 910. 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 (indicated as "paddings") are needed to support the "side samples" of the plus-shaped spatial filter in Fig. 8 and are padded when in 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, however LDL decomposition was chosen instead of Cholesky decomposition to avoid using square root operations.

Gradient Linear Model (GLM)

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.

$$C = \alpha \cdot G + \beta$$

For signalling, when the CCLM mode is enabled for the current CU, two flags are signalled separately for Cb and Cr components to indicate whether GLM is enabled for each component. If the GLM is enabled for one component, one syntax element is further signalled to select one of 16 gradient filters (1010-1040 in Fig. 10) for the gradient calculation. The GLM can be combined with the existing CCLM by signalling one extra flag in bitstream. When such combination is applied, the filter coefficients that are used to derive the input luma samples of the linear model are calculated as the combination of the selected gradient filter of the GLM and the down-sampling filter of the CCLM.

In order to improve the prediction accuracy or coding performance of cross-component prediction, various schemes related to inheriting cross-component models are disclosed.

Guided parameter set for refining the cross-component model parameters

According to this method, the guided parameter set is used to refine the derived model parameters by a specified CCLM mode. For example, the guided parameter set is explicitly signalled in the bitstream, after deriving the model parameters, the guided parameter set is added to the derived model parameters as the final model parameters. The guided parameter set contain at least one of a differential scaling parameter (dA), a differential offset parameter (dB), and a differential shift parameter (dS). For example, equation (1) can be rewritten as:

$$\mathrm{pred_C(i,j)} = \left(\left(\alpha' \cdot \mathrm{rec_L}'(\mathrm{i,j})\right) \gg \mathrm{s}\right) + \beta,$$

and if dA is signalled, the final prediction is:

$$\mathrm{pred_C(i,j)} = \left(\left((\alpha' + \mathrm{dA}) \cdot \mathrm{rec_L}'(\mathrm{i,j})\right) \gg \mathrm{s}\right) + \beta.$$

Similarly, if dB is signalled, then the final prediction is:

$$\mathrm{pred_C(i,j)} = \left(\left(\alpha' \cdot \mathrm{rec_L}'(\mathrm{i,j})\right) \gg \mathrm{s}\right) + (\beta + \mathrm{dB}).$$

If dS is signalled, then the final prediction is:

$$\mathrm{pred_C(i,j)} = \left(\left(\alpha' \cdot \mathrm{rec_L}'(\mathrm{i,j})\right) \gg (\mathrm{s} + \mathrm{dS})\right) + \beta.$$

If dA and dB are signalled, then the final prediction is:

$$\mathrm{pred_C(i,j)} = \left(\left((\alpha' + \mathrm{dA}) \cdot \mathrm{rec_L}'(\mathrm{i,j})\right) \gg \mathrm{s}\right) + (\beta + \mathrm{dB}).$$

The guided parameter set can be signalled per colour component. For example, one guided parameter set is signalled for Cb component, and another guided parameter set is signalled for Cr component. Alternatively, one guided parameter set can be signalled and shared among colour components. The signalled dA and dB can be a positive or negative value. When signalling dA, one bin is signalled to indicate the sign of dA. Similarly, when signalling dB, one bin is signalled to indicate the sign of dB.

For another embodiment, if dA is signalled, dB can be implicitly derived from the average value of neighbouring (e.g. L-shape) reconstructed samples. For example, in VVC, four neighbouring luma and chroma reconstructed samples are selected to derived model parameters. Suppose the average value of neighbouring luma and chroma samples are lumaAvg and chromaAvg, then β is derived by $\beta = \mathrm{chromaAvg} - (\alpha' + \mathrm{dA}) \cdot \mathrm{lumaAvg}$. The average value of neighbouring luma samples (i.e., lumaAvg) can be calculated by all selected luma samples, the luma DC mode value of the current luma CB, or the average of the maximum and minimum luma samples (e.g., $\mathrm{lumaAvg} = (\mathrm{Max}(x_A^0, x_A^1) + \mathrm{Min}(x_B^0, x_B^1) + 1) \gg 1$, or $\mathrm{lumaAvg} = (\mathrm{Min}(x_A^0, x_A^1) + \mathrm{Max}(x_B^0, x_B^1) + 1) \gg 1$).

Similarly, average value of neighbouring chroma samples (i.e., chromaAvg) can be calculated by all selected chroma samples, the chroma DC mode value of the current chroma CB, or the average of the maximum and minimum chroma samples (e.g., $\text{chromaAvg} = (\text{Max}(y_A^0, y_A^1) + \text{Min}(y_B^0, y_B^1) + 1) \gg 1$, or $\text{chromaAvg} = (\text{Min}(y_A^0, y_A^1) + \text{Max}(y_B^0, y_B^1) + 1) \gg 1$). Note, for non-4:4:4 colour subsampling format, the selected neighbouring luma reconstructed samples can be from the output of CCLM downsampling process.

For another embodiment, the shift parameter, s, can be a constant value (e.g., s can be 3, 4, 5, 6, 7, or 8), and dS is equal to 0 and no need to be signalled.

For another embodiment, in MMLM, the guided parameter set can also be signalled per model. For example, one guided parameter set is signalled for one model and another guided parameter set is signalled for another model. Alternatively, one guided parameter set is signalled and shared among linear models. Or only one guided parameter set is signalled for one selected model, and another model is not further refined by guided parameter set.

Inherit neighbouring model parameters for refining the cross-component model parameters

The final scaling parameter of the current block is inherited from the neighbouring blocks and further refined by dA (e.g., dA derivation or signalling can be similar or the same as the method in the previous “Guided parameter set for refining the cross-component model parameters”). Once the final scaling parameter is determined, the offset parameter (e.g., β in CCLM) is derived based on the inherited scaling parameter and the average value of neighbouring luma and chroma samples of the current block. For example, if the final scaling parameter is inherited from a selected neighbouring block, and the inherited scaling parameter is α'_{nei}, then the final scaling parameter is (α'_{nei} + dA). For yet another embodiment, the final scaling parameter is inherited from a historical list and further refined by dA. For example, the historical list records the most recent j entries of final scaling parameters from previous CCLM-coded blocks. Then, the final scaling parameter is inherited from one selected entry of the historical list, α'_{list}, and the final scaling parameter is (α'_{list} + dA). For yet another embodiment, the final scaling parameter is inherited from a historical list or the neighbouring blocks, but only the MSB (Most Significant Bit) part of the inherited scaling parameter is taken, and the LSB (Least Significant Bit) of the final scaling parameter is from dA. For yet another embodiment, the final scaling parameter is inherited from a historical list or the neighbouring blocks, but does not further refine by dA.

For yet another embodiment, after inheriting model parameters, the offset can be further refined by dB. For example, if the final offset parameter is inherited from a selected neighbouring block, and the inherited offset parameter is β'_{nei}, then the final offset parameter is (β'_{nei} + dB). For still another embodiment, the final offset parameter is inherited from a historical list and further refined by dB. For example, the historical list records the most recent j entries of final offset parameters from previous CCLM-coded blocks. Then, the final offset parameter is inherited from one selected entry of the historical list, β'_{list}, and the final offset parameter is (β'_{list} + dB).

For yet another embodiment, if the inherited neighbour block is coded with CCCM, the filter coefficients (c_i) are inherited. The offset parameter (e.g., $c_6 \times B$ or c_6 in CCCM) can be re-derived based on the inherited parameter and the average value of neighbouring corresponding position luma and chroma samples of the current block. For still another embodiment, only partial filter coefficients are inherited (e.g., only n out of 6 filter coefficients are inherited, where $1 \leq n < 6$), the rest filter coefficients are further re-derived using the neighbouring luma and chroma samples of the current block.

For still another embodiment, if the inherited candidate applies GLM gradient pattern to its luma reconstructed samples, the current block shall also inherit the GLM gradient pattern of the candidate and apply to the current luma reconstructed samples.

For still another embodiment, if the inherited neighbour block is coded with multiple cross-component models (e.g., MMLM, or CCCM with multi-model), the classification threshold is also inherited to classify the neighbouring samples of the current block into multiple groups, and the inherited multiple cross-component model parameters are further assigned to each group. For yet another embodiment, the classification threshold is the average value of the neighbouring reconstructed luma samples, and the inherited multiple cross-component model parameters are further assigned to each group. Similarly, once the final scaling parameter of each group is determined, the offset parameter of each group is re-derived based on the inherited scaling parameter and the average value of neighbouring luma and chroma samples of each group of the current block. For another example, if CCCM with multi-model is used, once the final coefficient parameter of each group is determined (e.g., c_0 to c_5 except for c_6 in CCCM), the offset parameter (e.g., $c_6 \times B$ or c_6 in CCCM) of each group is re-derived based on the inherited coefficient parameter and the neighbouring luma and chroma samples of each group of the current block.

For still another embodiment, inheriting model parameters may depend on the colour component. For example, Cb and Cr components may inherit model parameters or model derivation method from the same candidate or different candidates. For yet another example, only one of colour components inherits model parameters, and the other colour component derives model parameters based on the inherited model derivation method (e.g., if the inherit candidate is coded by MMLM or CCCM, the current block also derives model parameters based on MMLM or CCCM using the current neighbouring reconstructed samples). For still another example, only one of colour components inherits model parameters, and the other colour component derives its model parameters using the current neighbouring reconstructed samples.

For yet another embodiment, after decoding a block, a cross-component model of the current block is derived and stored for later reconstruction process of neighbouring blocks using inherited neighbours model parameter. For example, even the current block is coded by inter prediction, the cross-component model parameters of the current block can be derived by using the current luma and chroma reconstruction or prediction samples. Later, if another block is predicted by using inherited neighbours model parameters, it can inherit the model parameters from the current block. For another example, the current block is coded by cross-component prediction, the cross-component model parameters of the current block are re-derived by using the current luma and chroma reconstruction or prediction samples. For another example, the stored cross-component model can be CCCM, LM_LA (i.e., single model LM using both above and left neighbouring samples to derive model), or MMLM_LA (i.e., multi-model LM using both above and left neighbouring samples to derive model).

Inheriting temporal neighbouring model parameters

For still another embodiment, if the current slice/picture is a non-intra slice/picture, the inherited model parameters can be from the block in the previous coded slices/pictures. For example, as shown in the Fig. 11, the current block position is at (x, y) and the block size is $w \times h$. The inherited model parameters can be from the block at position (x', y') (x', y' + h/2) , (x' + w/2, y'), (x' + w/2, y' + h/2), (x' + w, y'), (x', y' + h), or (x' + w, y' + h) of the previous coded slices/picture, where x' = x + Δx and y' = y + Δy. In one embodiment, if the prediction mode of the current block is intra, Δx and Δy are set to 0. If the prediction mode of the current block is inter, Δx and Δy are set to the horizontal and vertical motion vector of the current block. In another embodiment, if the current block is inter bi-prediction, Δx and Δy are set to the horizontal and vertical motion vectors in reference picture list 0. In still another embodiment, if the current block is inter bi-prediction, Δx and Δy are set to the horizontal and vertical motion vectors in reference picture list 1.

Removing or modifying similar neighbouring model parameters

When inheriting cross-component model parameters from other blocks, it can further check the similarity between the inherited model and the existing models in the candidate list or those model candidates derived by the neighbouring reconstructed samples of the current block (e.g., models derived by CCLM, MMLM, or CCCM using the neighbouring reconstructed samples of the current block). If the model of a candidate parameter is similar to the existing models, the model will not be included in the candidate list. In one embodiment, it can compare the similarity of $(\alpha \times \text{lumaAvg} + \beta)$ or α among existing candidates to decide whether to include the model of a candidate or not. For example, if the $(\alpha \times \text{lumaAvg} + \beta)$ or α of the candidate is the same as one of the existing candidates, the model of the candidate is not included. For another example, if the difference of $(\alpha \times \text{lumaAvg} + \beta)$ or α between the candidate and one of existing candidates is less than a threshold, the model of the candidate is not included. Besides, the threshold can be adaptive based on coding information (e.g., the current block size or area). For another example, when comparing the similarity, if a model from a candidate and the existing model both use CCCM, it can compare similarity by checking the value of $(c_0C + c_1N + c_2S + c_3E + c_4W + c_5P + c_6B)$ to decide whether to include the model of a candidate or not. In another embodiment, if a candidate position point to a CU which is the same one of the existing candidates, the model of the candidate parameter is not included. In still another embodiment, if the model of a candidate is similar to one of existing candidate models, it can adjust the inherited model parameters so that the inherited model is different from the existing candidate models. For example, if the inherited scaling parameter is similar to one of existing candidate models, the inherited scaling parameter can add a predefined offset (e.g., $1 \gg S$ or $-(1 \gg S)$, where S is the shift parameter) so that the inherited parameter is different from the existing candidate models.

Reordering the candidates in the list

The candidates in the list can be reordered to reduce the syntax overhead when signalling the selected candidate index. The reordering rules can depend on the coding information of neighbouring blocks or the model error. For example, if neighbouring above or left blocks are coded by MMLM, the MMLM candidates in the list can be moved to the head of the current list. Similarly, if neighbouring above or left blocks are coded by single model LM or CCCM, the single model LM or CCCM candidates in the list can be moved to the head of the current list. Similarly, if GLM is used by neighbouring above or left blocks, the GLM related candidates in the list can be moved to the head of the current list.

In still another embodiment, the reordering rule is based on the model error by applying the candidate model to the neighbouring templates of the current block, and then compare the error with the reconstructed samples of the neighbouring template. For example, as shown in Fig. 12, the size of above neighbouring template of the current block is $w_a \times h_a$, and the size of left neighbouring template of the current block is $w_b \times h_b$. Suppose *K* models are in the current candidate list, and α_k and β_k are the final scale and offset parameters after inheriting the candidate *k*. The model error of candidate *k* corresponding to the above neighbouring template is:

$$e_a^k = \sum_{i,j} \left| \left(\alpha_k \times recL'^{(i,j)}_a + \beta_k \right) - recC^{(i,j)}_a \right|$$

where, $recL'^{(i,j)}_a$ and $recC^{(i,j)}_a$ are the reconstructed samples of luma (e.g., after downsampling process or after applying GLM pattern) and reconstructed samples of chroma at position (*i*, *j*) in the above template, and $0 \le i < w_a$ and $0 \le j < h_a$.

Similarly, the model error of candidate *k* by the left neighbouring template is:

$$e_b^k = \sum_{m,n} \left| \left(\alpha_k \times recL'^{(m,n)}_b + \beta_k \right) - recC^{(m,n)}_b \right|$$

where $recL'^{(m,n)}_b$ and $recC^{(m,n)}_b$ are the reconstructed samples of luma (e.g., after applying downsampling process or GLM pattern) and reconstructed samples of chroma at position (*m*, *n*) in the left template, and $0 \le m < w_b$ and $0 \le n < h_b$.

Then the model error of candidate *k* is:

$$e^k = e_a^k + e_b^k$$

After calculating the model error among all candidates, it can get a model error list $E = \{e^0, e^1, e^2, \ldots, e^k, \ldots, e^K\}$. Then, it can reorder the candidate index in the inherited candidate list by sorting the model error list in ascending order.

In still another embodiment, if the candidate *k* uses CCCM prediction, the e_a^k and e_b^k are defined as:

$$\begin{aligned} e_a^k = \sum_{i,j} \Big| \Big(& c0_k \times recL'^{(i,j)}_a + c1_k \times recL'^{(i,j-1)}_a + c2_k \times recL'^{(i,j+1)}_a + c3_k \\ & \times recL'^{(i-1,j)}_a + c4_k \times recL'^{(i+1,j)}_a + c5_k \times P + c6_k \times B \Big) \\ & - recC_a^{(i,j)} \Big| \end{aligned}$$

$$\begin{aligned} e_b^k = \sum_{i,j} \Big| \Big(& c0_k \times recL'^{(i,j)}_b + c1_k \times recL'^{(i,j-1)}_b + c2_k \times recL'^{(i,j+1)}_b + c3_k \\ & \times recL'^{(i-1,j)}_b + c4_k \times recL'^{(i+1,j)}_b + c5_k \times P + c6_k \times B \Big) \\ & - recC_b^{(i,j)} \Big| \end{aligned}$$

where $c0_k, c1_k, c2_k, c3_k, c4_k, c5_k$, and $c6_k$ are the final filtering coefficients after inheriting the candidate *k*. P and B are the nonlinear term and bias term.

In still another embodiment, if the above neighbouring template is not available, then $e^k = e_b^k$. Similarly, if the left neighbouring template is not available, then $e^k = e_a^k$. If both templates are not available, the candidate index reordering method using model error is not applied.

In still another embodiment, not all positions inside the above and left neighbouring template are used in calculating model error. It can choose partial positions inside the above and left neighbouring template to calculate model error. For example, it can define a first start position and a first subsampling interval depends on the width of the current block to partially select positions inside the above neighbouring template. Similarly, it can define a second start position and a second subsampling interval depends on the height of the current block to partially select positions inside the left neighbouring template. For another example, h_a or w_b can be a constant value (e.g., h_a or w_b can be 1, 2, 3, 4, 5, or 6). For another example, h_a or w_b can be dependent on the block size. If the current block size is greater than or equal to a threshold, h_a or w_b is equal to a first value. Otherwise, h_a or w_b is equal to a second value.

Inheriting candidates from the candidates in the candidate list of neighbours

The candidates in the current inherited candidate list can be from neighbouring blocks. For example, it can inherit the first *k* candidates in the inherited candidate list of the neighbouring blocks. As shown in the Fig. 13, the current block can inherit the first two candidates in the inherited candidate list of the above neighbouring block and the first two candidates in the inherited candidate list of the left neighbouring block. For an embodiment, after adding the neighbouring spatial candidates and non-adjacent spatial candidates, if the current inherited candidate list is not full, the candidates in the candidate list of neighbouring blocks are included into the current inherited candidate list. For another embodiment, when including the candidates in the candidate list of neighbouring blocks, the candidates in the candidate list of left neighbouring blocks are included before the candidates in the candidate list of above neighbouring blocks. For still another embodiment, when including the candidates in the candidate list of neighbouring blocks, the candidates in the candidate list of above neighbouring blocks are included before the candidates in the candidate list of left neighbouring blocks.

Signalling the inherit candidate index in the list

An on/off flag can be signalled to indicate if the current block inherits the cross-component model parameters from neighbouring blocks or not. The flag can be signalled per CU/CB, per PU, per TU/TB, or per colour component, or per chroma colour component. A high level syntax can be signalled in SPS, PPS (Picture Parameter Set), PH (Picture header) or SH (Slice Header) to indicate if the proposed method is allowed for the current sequence, picture, or slice.

If the current block inherits the cross-component model parameters from neighbouring blocks, the inherit candidate index is signalled. The index can be signalled (e.g., signalled using truncate unary code, Exp-Golomb code, or fix length code) and shared among both the current Cb and Cr blocks. For another example, the index can be signalled per colour component. For example, one inherited index is signalled for Cb component, and another inherited index is signalled for Cr component. For another example, it can use chroma intra prediction syntax (e.g., IntraPredModeC[xCb][yCb]) to store the inherited index.

If the current block inherits the cross-component model parameters from neighbouring blocks, the current chroma intra prediction mode (e.g., IntraPredModeC[xCb][yCb] as defined in VVC standard) is temporally set to a cross-component mode (e.g., CCLM_LT) at the bitstream syntax parsing stage. Later, at the prediction stage or reconstruction stage, the candidate list is derived, and the inherited candidate model is then determined by the inherited candidate index. After obtaining the inherited model, the coding information of the current block is then updated according to the inherited candidate model. The coding information of the current block includes, but not limited to, the prediction mode (e.g., CCLM_LT or MMLM_LT), related sub-mode flags (e.g., CCCM mode flag), prediction pattern (e.g., GLM pattern index), and the current model parameters. Then, the prediction of the current block is generated according to the updated coding information.

Inheriting multiple cross-component models

The final prediction of the current block can be the combination of multiple cross-component models, or fusion the selected cross-component models with the prediction by non-cross-component coding tools (e.g., intra angular prediction modes, intra planar/DC modes, or inter prediction modes). In one embodiment, if the current candidate list size is N, it can select k candidates from the total N

candidates (where $k \leq N$). Then, k predictions are respectively generated by applying the cross-component model of the selected k candidates using the corresponding luma reconstruction samples. The final prediction of the current block is the combination results of these k predictions. For example, if two candidate predictions (denoted as p_{cand1} and p_{cand2}) are combined, the final prediction at (x, y) position of the current block is $p_{final}(x,y) = (1-\alpha) \times p_{cand1}(x,y) + \alpha \times p_{cand2}(x,y)$, where α is a weighting factor. Besides, the weighting factor α can be predefined or implicitly derived by neighbouring template cost. For example, by using the template cost defined in the section entitled "**Reordering the candidates in the list**", the corresponding template cost of two candidates are e^{cand1} and e^{cand2}, then α is $e^{cand1}/(e^{cand1} + e^{cand2})$. In another embodiment, if two candidate models are combined, the selected models are from the first two candidates in the list. In still another embodiment, if i candidate models are combined, the selected models are from the first i candidates in the list.

In another embodiment, if the current candidate list size is N, it can select k candidates from the total N candidates (where $k \leq N$). The k cross-component models can be combined into one final cross-component model by weighted-averaging the corresponding model parameters. For example, if a cross-component model has M parameters, the j-th parameter of the final cross-component model are the weighted-averaging of the j-th parameter of the k selected candidates where j is 1 … M. Then, the final prediction is by applying the final cross-component model to the corresponding luma reconstruction samples. For example, if two candidate models are $\{c_0^0, c_1^0, \dots, c_{M-1}^0\}$ and $\{c_0^1, c_1^1, \dots, c_{M-1}^1\}$. The final cross-component model is $\{(1-\alpha) \times c_0^0 + \alpha \times c_0^1, (1-\alpha) \times c_1^0 + \alpha \times c_1^1, \dots, (1-\alpha) \times c_{M-1}^0 + \alpha \times c_{M-1}^1\}$, where α is a weighting factor which can be predefined or implicitly derived by neighbouring template cost, and c_x^y is the x-th model parameter of the y-th candidate. For example, by using the template cost defined in above Section 2.2.3, the corresponding template cost of two candidates are e^{cand1} and e^{cand2}, then α is $e^{cand1}/(e^{cand1} + e^{cand2})$. For still an example, the two candidate models are one from spatial adjacent neighbouring candidate, and another one from non-adjacent spatial candidate or history candidate. If the spatial adjacent neighbouring candidate is not available, then the two candidate models are all from the non-adjacent spatial candidates or history candidates. In another embodiment, if two candidate models are combined, the selected models are from the first two candidates in the list. In still another embodiment, if i candidate models are combined, the selected models are from the first i candidates in the list.

In another embodiment, two cross-component models are combined into one final model by weighted-averaging the corresponding model parameters, where the two cross-component models are one from above spatial neighbouring candidate and another one from left spatial neighbouring candidate. The above spatial neighbouring candidate is the neighbouring candidate that has the vertical position less than or equal to the top block boundary position of the current block. The left spatial neighbouring candidate is the neighbouring candidate that has the horizontal position less than or equal to the left block boundary position of the current block. The weighting factor α is determined according to the horizontal and vertical spatial positions inside the current block. For example, if two candidate predictions (denoted as p_{above} and p_{left}) are combined, the final prediction at (x, y) position of the current block is $p_{final}(x, y) = (1-\alpha) \times p_{above}(x, y) + \alpha \times p_{left}(x, y)$, where $\alpha = y/(x+y)$. In another embodiment, the above spatial neighbouring candidate is the first candidate in the list that has the vertical position less than or equal to the top block boundary position of the current block. The left spatial neighbouring candidate is the first candidate in the list that has the horizontal position less than or equal to the left block boundary position of the current block.

In another embodiment, it can combine cross-component model candidates with the prediction by non-cross-component coding tools. For example, one cross-component model candidate is selected from list, and its prediction is denoted as p_{ccm}. Another prediction can be from chroma DM, chroma DIMD, or intra angular mode, and denoted as $p_{non-ccm}$. The final prediction at (x, y) position of the current block is $p_{final}(x, y) = (1-\alpha) \times p_{ccm}(x, y) + \alpha \times p_{non-ccm}(x, y)$, where α is the weighting factor which can be predefined or implicitly derived based on the neighbouring template cost. Furthermore, the prediction by non-cross-component coding tool can be predefined or signalled. The prediction by non-cross-component coding tool is chroma DM or chroma DIMD. For another example, prediction by non-cross-component coding tool is signalled, but the index of cross-component model candidate is predefined or determined according to the coding modes of neighbouring blocks. For still the same example, if at least one of neighbouring spatial blocks is coded with CCCM mode, the first candidate has CCCM model parameters is selected. If at least one of neighbouring spatial blocks is coded with GLM mode, the first candidate has GLM pattern parameters is selected. Similarly, if at least one of neighbouring spatial blocks is coded with MMLM mode, the first candidate has MMLM parameters is selected.

In another embodiment, it can combine cross-component model candidates with the prediction by the current cross-component model. For example, one cross-component model candidate is selected from the list, and its prediction is denoted as p_{ccm}. Another prediction can be from the cross-component prediction mode by the current neighbouring reconstruction samples and denoted as $p_{curr-ccm}$. The final prediction at (x, y) position of the current block is $p_{final}(x, y) = (1-\alpha) \times p_{ccm}(x, y) + \alpha \times p_{curr-ccm}(x, y)$, where α is the weighting factor which can be predefined or implicitly derived by neighbouring template cost. For still the same example, the prediction by the current cross-component model can be predefined or signalled. The prediction by non-cross-component coding tool is CCCM_LT, LM_LT (single model LM using both top and left neighbouring samples to derive model), or MMLM_LT (multi-model LM using both top and left neighbouring samples to derive model). In one embodiment, the selected cross-component model candidate is the first candidate in the list.

In another embodiment, it can combine multiple cross-component models into one final cross-component model. For example, it can choose a first model from one candidate in the list, and choose a second model from another candidate to be a different model. The selected candidates can be CCLM/MMLM/GLM/CCCM coded candidates. The multi-model classification threshold can be the average of the offset parameters (e.g., offset/β in CCLM, or $c_6 \times B$ or c_6 in CCCM) of the two selected models. In one embodiment, if two candidate models are combined, the selected models are the first two candidates in the list.

Refining the inherited candidate positions

The final inherit model of the current block is from the cross-component model at the indicated candidate position with a delta position. For example, if the current selected candidate position is $(x_{nei}^{i}, y_{nei}^{i})$, it can further signal a delta position, $(dx_{nei}^{i}, dy_{nei}^{i})$, to indicate the position of the final inherited model. That is, the final inherited model of the current block is from the cross-component model at $(x_{nei}^{i} + dx_{nei}^{i}, y_{nei}^{i} + dy_{nei}^{i})$. In one embodiment, the signalled delta position can only have a horizontal delta position or a vertical delta position, that is, $(dx_{nei}^{i}, 0)$ or $(0, dy_{nei}^{i})$. Besides, the signalled delta position can be shared among more than one colour component or signalled per colour component. For example, the signalled delta position is share for the current Cb and Cr blocks, or the signalled delta position is only used for the current Cb block or the current Cr block. Furthermore, the signalled dx_{nei}^{i} or dy_{nei}^{i} can have a sign bit to indicate positive delta position or negative delta position. When indicating the magnitude of dx_{nei}^{i} or dy_{nei}^{i}, it can be signalled by a look-up table index. For example, a look-up table is $\{1, 2, 4, 8, 16, \ldots\}$, if $|dx_{nei}^{i}|$ is equal to 8, then the table index 3 is signalled (the first table index is 0).

Inheriting from shared cross-component models

Current picture is segmented into many non-overlapped regions, and each region size is $M \times N$. A shared cross-component model is derived for each region, respectively. The neighbouring available luma/chroma reconstruction samples of the current region are used to derive the shared cross-component model of the current region. Then, for a block inside the current region, it can determine whether to inherit the shared cross-component model or derive the cross-component model by the neighbouring available luma/chroma reconstruction samples of the block. In one embodiment, the $M \times N$ can be a predefined value (e.g. 32x32 for the chroma format), a signalled value (e.g. signalled in sequence/picture/slice/tile-level), a derived value (e.g. depending on the CTU size), or the maximum allowed transform block size.

In another embodiment, each region can have more than one shared cross-component model. For example, it can use various neighbouring templates (e.g., top and left neighbouring samples, top-only neighbouring samples, left-only neighbouring samples) to derive more than one shared cross-component model. Besides, the shared cross-component models of the current region can be inherited from previously used cross-component models. For example, the shared model can be inherited from the models of adjacent spatial neighbours, non-adjacent spatial neighbours, temporal neighbours, or from a historical list.

When doing signalling, a first flag is used to determine if the current cross-component model is inherited from the shared cross-component models or not. If the current cross-component model is inherited from the shared cross-component models, the second syntax indicate the inherited index of the shared cross-component models (e.g., signalled using truncate unary code, Exp-Golomb code, or fix length code).

The cross component prediction with inherited model parameters as described above can be implemented in an encoder side or a decoder side. For example, any of the proposed cross component prediction methods can be implemented in an Intra/Inter coding module (e.g. Intra Pred. 150/MC 152 in Fig. 1B) in a decoder or an Intra/Inter coding module is an encoder (e.g. Intra Pred. 110/Inter Pred. 112 in Fig. 1A). Any of the proposed CCLM methods can also be implemented as a circuit coupled to the intra/inter coding module at the decoder or the encoder. However, the decoder or encoder may also use additional processing unit to implement the required CCLM processing. While the Intra Pred. units (e.g. unit 110/112 in Fig. 1A and unit 150/152 in Fig. 1B) are shown as individual processing units, they may correspond to executable software or firmware codes stored on a media, such as hard disk or flash memory, for a CPU (Central Processing Unit) or programmable devices (e.g. DSP (Digital Signal Processor) or FPGA (Field Programmable Gate Array)).

Fig. 14 illustrates a flowchart of an exemplary video coding system that incorporates inheriting temporal cross-component model parameters according to an embodiment of the present invention. The steps shown in the flowchart may be implemented as program codes executable on one or more processors (e.g., one or more CPUs) at the encoder side. The steps shown in the flowchart may also be implemented based hardware such as one or more electronic devices or processors arranged to perform the steps in the flowchart. According to this method, input data associated with a current block comprising a first-colour block and a second-colour block are received in step 1410, wherein the input data comprise pixel data to be encoded at an encoder side or data associated with the current block to be decoded at a decoder side, and wherein the current block is in a non-intra slice or picture. Reference data or a location related to the reference data in a previously coded slice or picture is determined based on information comprising motion information of the current block in step 1420. Target cross-component model parameters associated with a target inherited prediction model are derived by inheriting cross-component model parameters of the reference data in step 1430. The second-colour block is encoded or decoded using prediction data comprising cross-colour prediction generated by applying the target inherited prediction model with the target cross-component model parameters to reconstructed first-colour block in step 1440.

The flowchart shown is intended to illustrate an example of video coding according to the present invention. A person skilled in the art may modify each step, re-arranges the steps, split a step, or combine steps to practice the present invention without departing from the spirit of the present invention. In the disclosure, specific syntax and semantics have been used to illustrate examples to implement embodiments of the present invention. A skilled person may practice the present invention by substituting the syntax and semantics with equivalent syntax and semantics without departing from the spirit of the present invention.

The above description is presented to enable a person of ordinary skill in the art to practice the present invention as provided in the context of a particular application and its requirement. Various modifications to the described embodiments will be apparent to those with skill in the art, and the general principles defined herein may be applied to other embodiments. Therefore, the present invention is not intended to be limited to the particular embodiments shown and described, but is to be accorded the widest scope consistent with the principles and novel features herein disclosed. In the above detailed description, various specific details are illustrated in order to provide a thorough understanding of the present invention. Nevertheless, it will be understood by those skilled in the art that the present invention may be practiced.

Embodiment of the present invention as described above may be implemented in various hardware, software codes, or a combination of both. For example, an embodiment of the present invention can be one or more circuit circuits integrated into a video compression chip or program code integrated into video compression software to perform the processing described herein. An embodiment of the present invention may also be program code to be executed on a Digital Signal Processor (DSP) to perform the processing described herein. The invention may also involve a number of functions to be performed by a computer processor, a digital signal processor, a microprocessor, or field programmable gate array (FPGA). These processors can be configured to perform particular tasks according to the invention, by executing machine-readable software code or firmware code that defines the particular methods embodied by the invention. The software code or firmware code may be developed in different programming languages and different formats or styles. The software code may also be compiled for different target platforms. However, different code formats, styles and languages of software codes and other means of configuring code to perform the tasks in accordance with the invention will not depart from the spirit and scope of the invention.

The invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described examples are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS:

1. A method of coding colour pictures using coding tools including one or more cross component models related modes, the method comprising:

receiving input data associated with a current block comprising a first-colour block and a second-colour block, wherein the input data comprise pixel data to be encoded at an encoder side or data associated with the current block to be decoded at a decoder side, and wherein the current block is in a non-intra slice or picture;

determining reference data or a location related to the reference data in a previously coded slice or picture based on information comprising motion information of the current block;

deriving target cross-component model parameters associated with a target inherited prediction model by inheriting cross-component model parameters of the reference data; and

encoding or decoding the second-colour block using prediction data comprising cross-colour prediction generated by applying the target inherited prediction model with the target cross-component model parameters to reconstructed first-colour block.

2. The method of Claim 1, wherein the reference data in the previously coded slice or picture is located according to a corresponding location of the current block in the previously coded slice or picture and a motion vector of the current block.

3. The method of Claim 2, wherein when the current block is intra coded, the motion vector used in deriving the target cross-component model parameters of the current block is set to 0.

4. The method of Claim 2, wherein the location of the reference data is determined based on the corresponding location of the current block shifted by the motion vector of the current block.

5. The method of Claim 2, wherein the location of the reference data is determined based on the corresponding location of the current block shifted by the motion vector of the current block with one or more additional offsets.

6. The method of Claim 5, wherein said one or more additional offsets depend on width of the current block, height of the current block, or both.

7. The method of Claim 6, wherein said one or more additional offsets correspond to a horizontal offset by the width of the current block and a vertical offset by the height of the current block.

8. The method of Claim 6, wherein said one or more additional offsets correspond to a horizontal offset by half of the width of the current block and a vertical offset by half of the height of the current block.

9. The method of Claim 1, wherein one or more of the cross-component model parameters of the reference data are refined and then used as the target cross-component model parameters.

10. The method of Claim 1, wherein the first-colour block corresponds to a luma block and the second-colour block corresponds to a chroma block.

11. An apparatus for coding colour pictures using coding tools including one or more cross component models related modes, the apparatus comprising one or more electronic circuits or processors arranged to:

receive input data associated with a current block comprising a first-colour block and a second-colour block, wherein the input data comprise pixel data to be encoded at an encoder side or data associated with the current block to be decoded at a decoder side, and wherein the current block is in a non-intra slice or picture;

determine reference data or a location related to the reference data in a previously coded slice or picture based on information comprising motion information of the current block;

derive target cross-component model parameters associated with a target inherited prediction model by inheriting cross-component model parameters of the reference data; and

encode or decode the second-colour block using prediction data comprising cross-colour prediction generated by applying the target inherited prediction model with the target cross-component model parameters to reconstructed first-colour block.

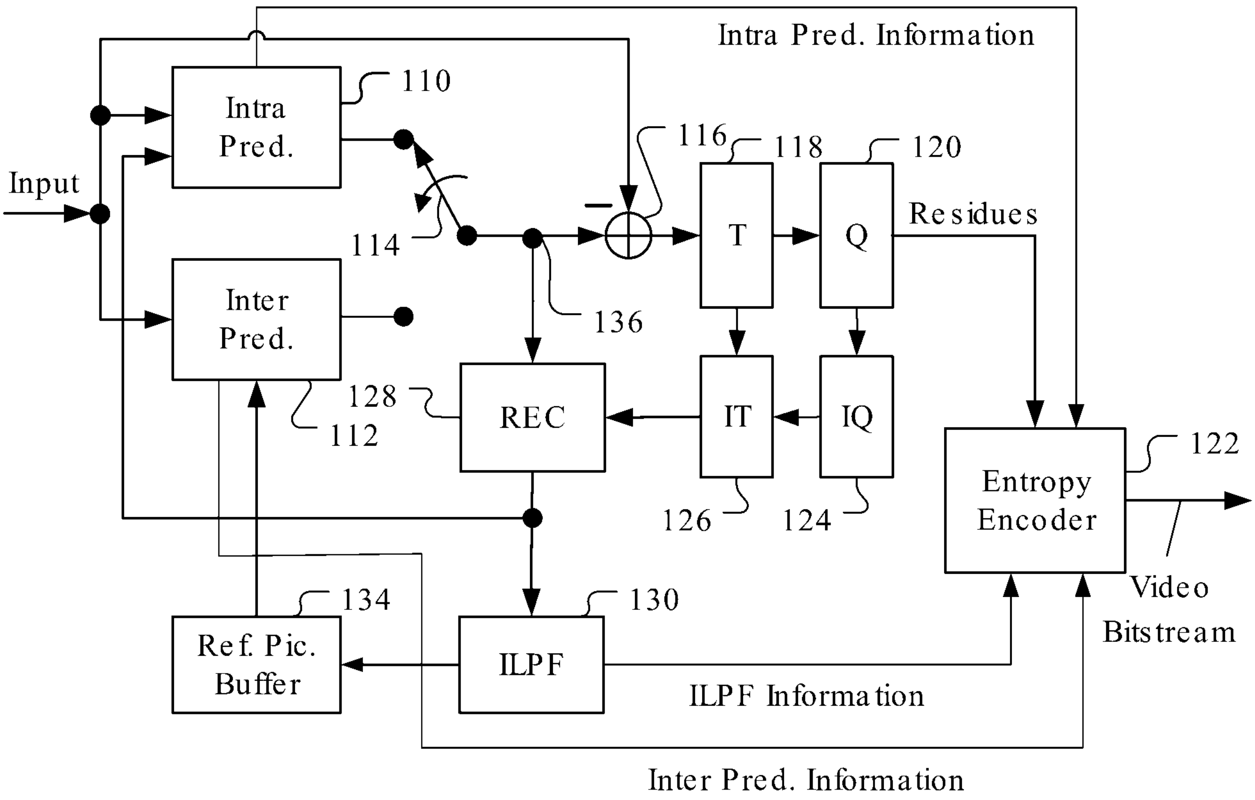
Intra Pred. Information
Intra Pred.
110
Input
116
118
120
T
Q
Residues
114
136
Inter Pred.
128
REC
IT
IQ
112
122
Entropy Encoder
126
124
Video Bitstream
134
130
Ref. Pic. Buffer
ILPF
ILPF Information
Inter Pred. Information

Fig. 1A

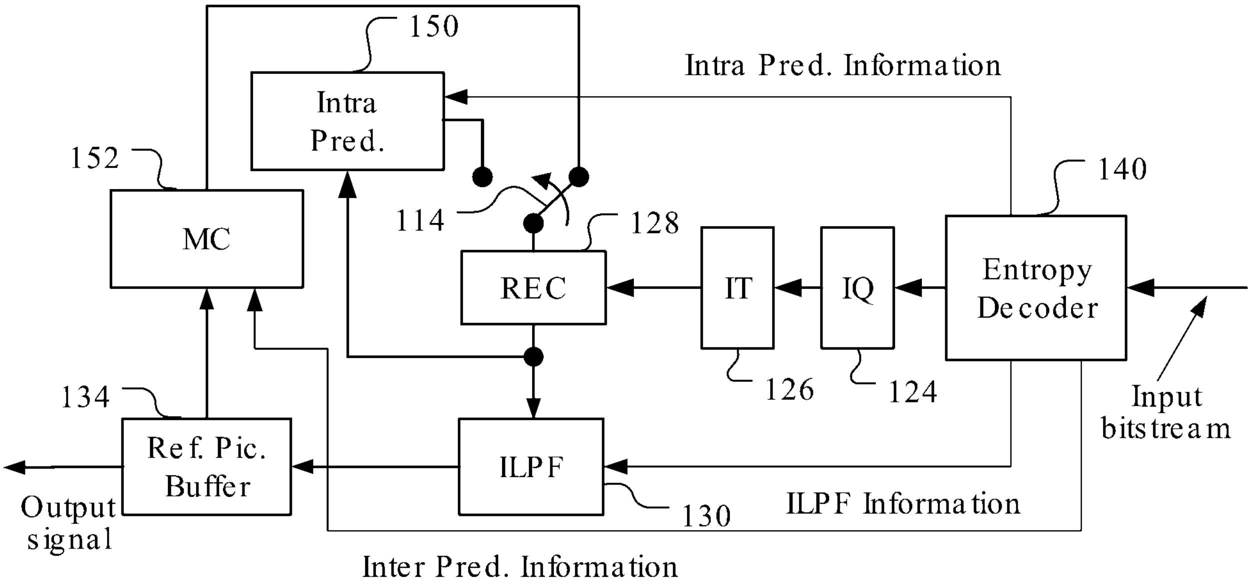
150
Intra Pred.
Intra Pred. Information
152
140
MC
114
128
REC
IT
IQ
Entropy Decoder
126
124
Input bitstream
134
Ref. Pic. Buffer
ILPF
Output signal
130
ILPF Information
Inter Pred. Information

Fig. 1B

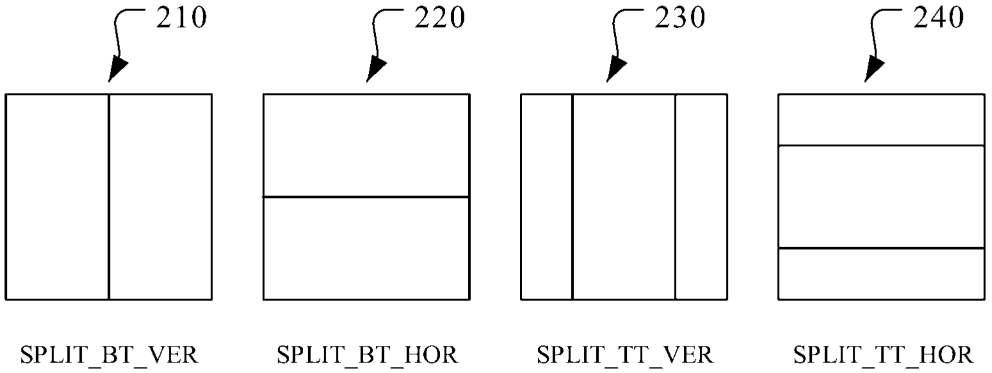
210
220
230
240
SPLIT_BT_VER
SPLIT_BT_HOR
SPLIT_TT_VER
SPLIT_TT_HOR

Fig. 2

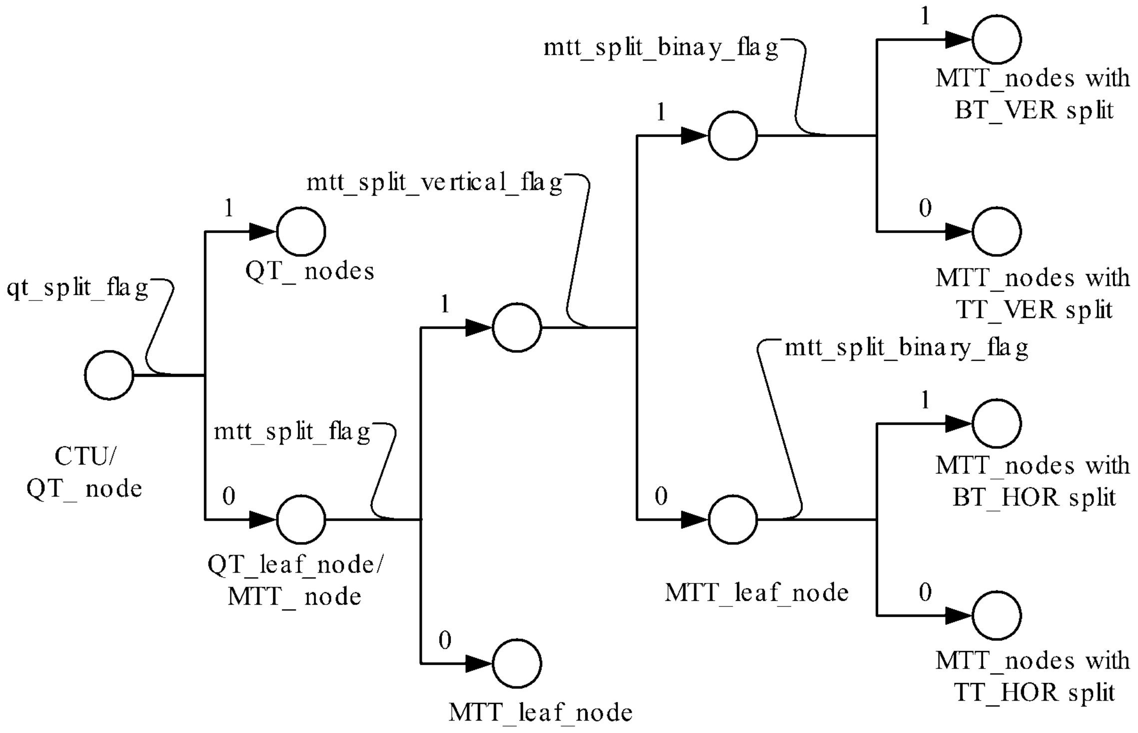
qt_split_flag
CTU/
QT_ node
1
QT_ nodes
0
mtt_split_flag
QT_leaf_node/
MTT_ node
1
0
MTT_leaf_node
mtt_split_vertical_flag
1
0
MTT_leaf_node
mtt_split_binay_flag
mtt_split_binary_flag
1
MTT_nodes with
BT_VER split
0
MTT_nodes with
TT_VER split
1
MTT_nodes with
BT_HOR split
0
MTT_nodes with
TT_HOR split

Fig. 3

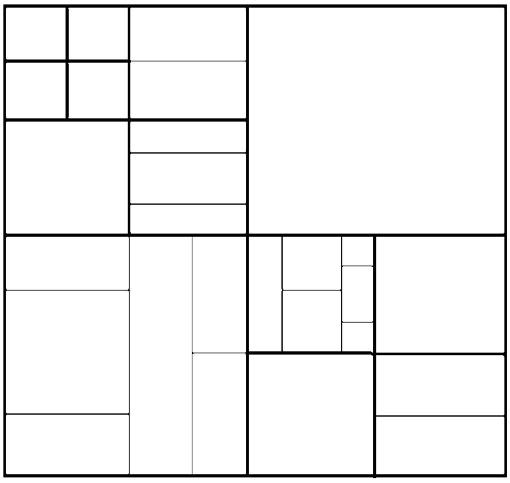

Fig. 4

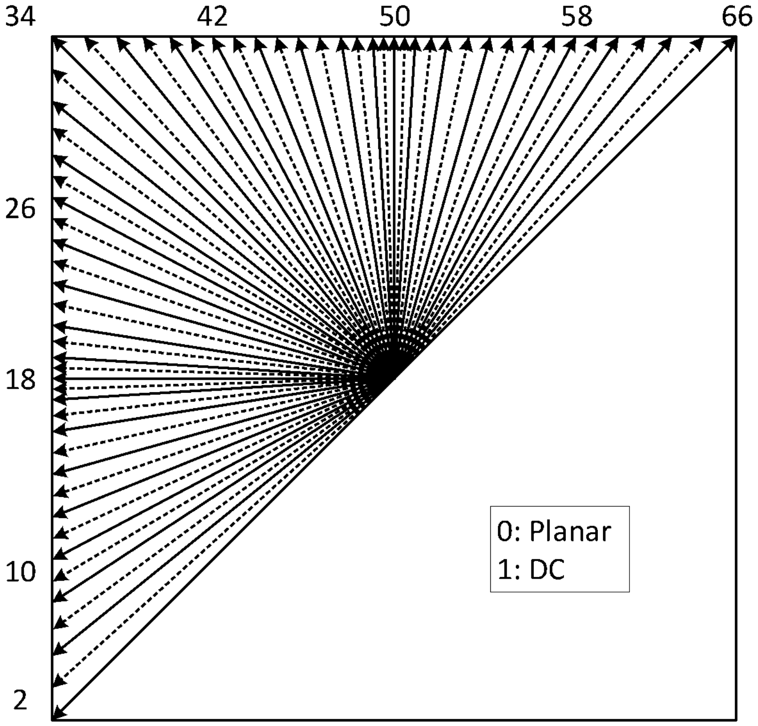

Fig. 5

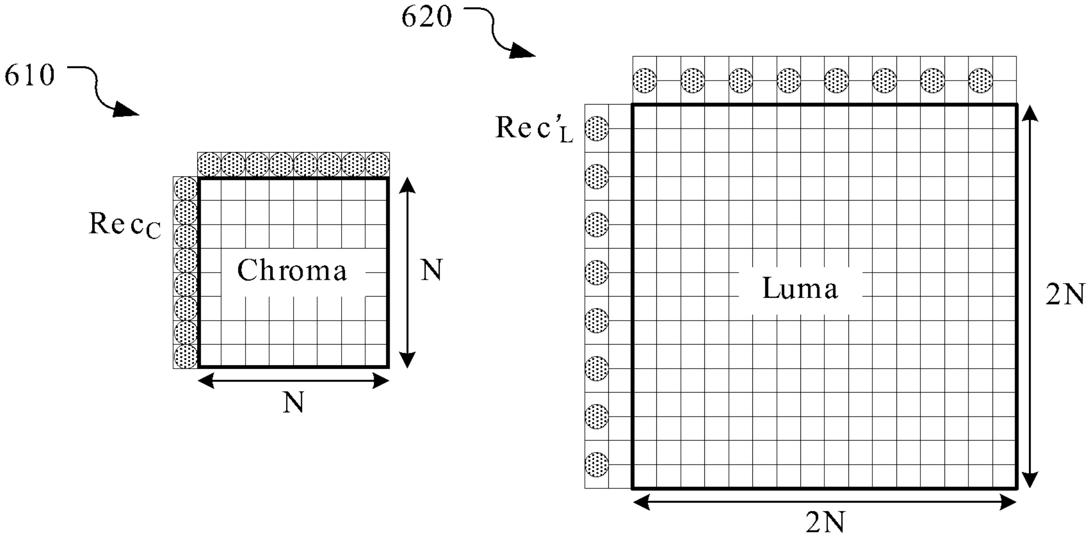
620
610
Rec'L
RecC
Chroma
N
N
Luma
2N
2N

Fig. 6

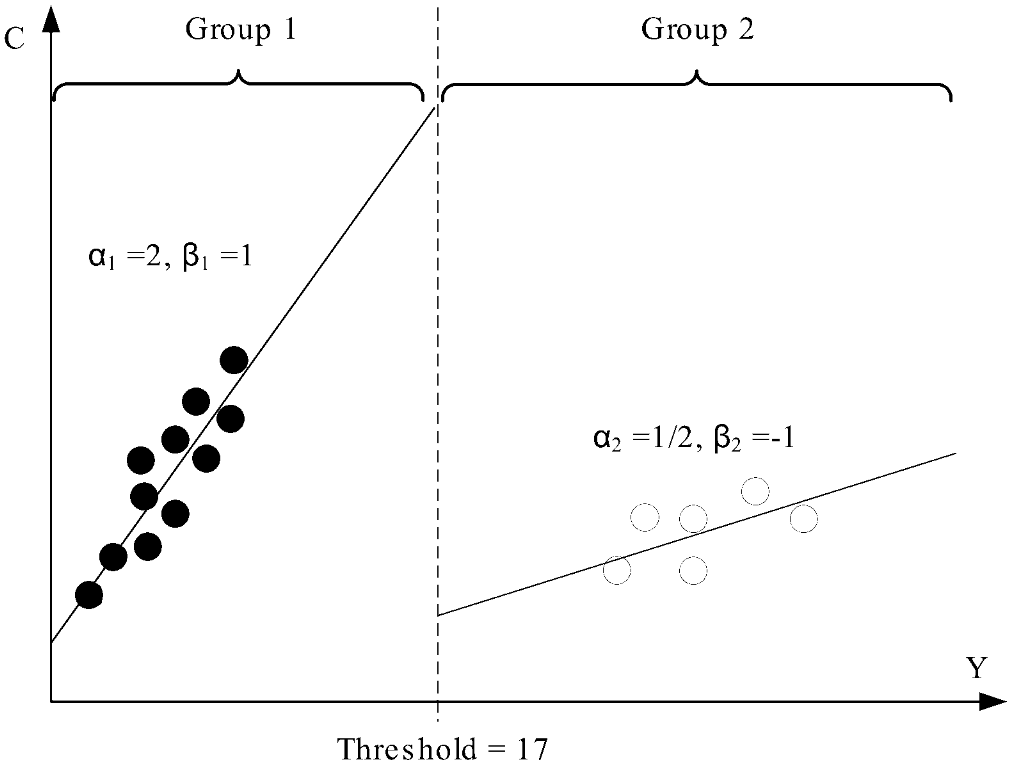
C
Group 1
Group 2
α_1 =2, β_1 =1
α_2 =1/2, β_2 =-1
Y
Threshold = 17

Fig. 7

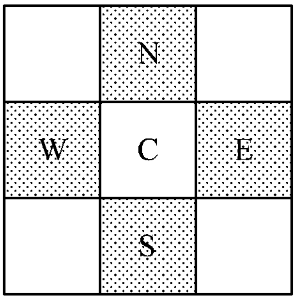
N
W
C
E
S

Fig. 8

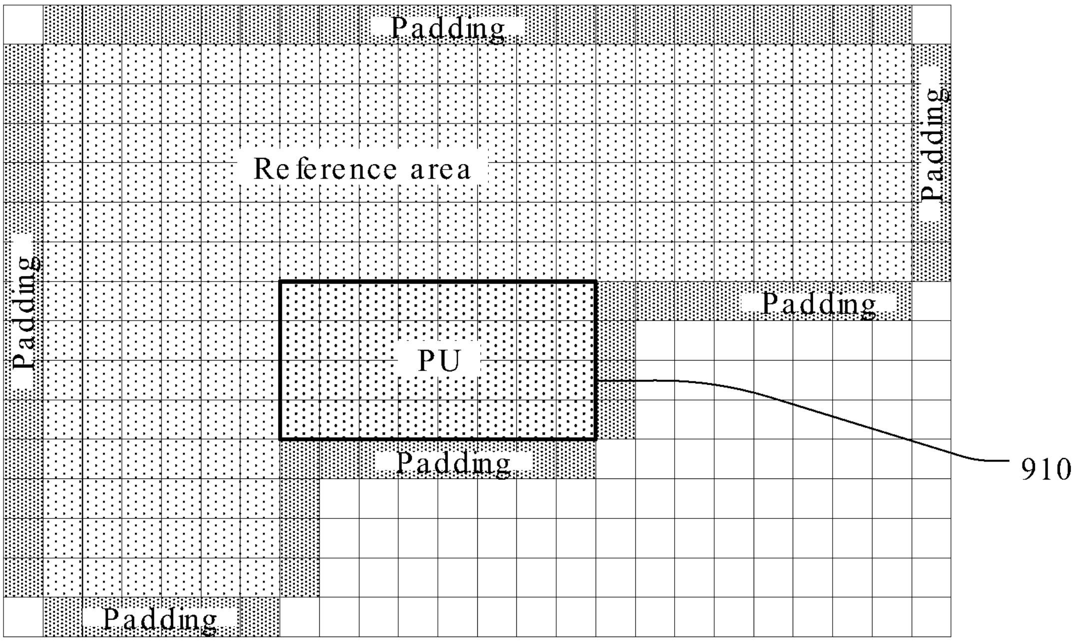
Padding
Padding
Reference area
Padding
Padding
PU
910
Padding
Padding

Fig. 9

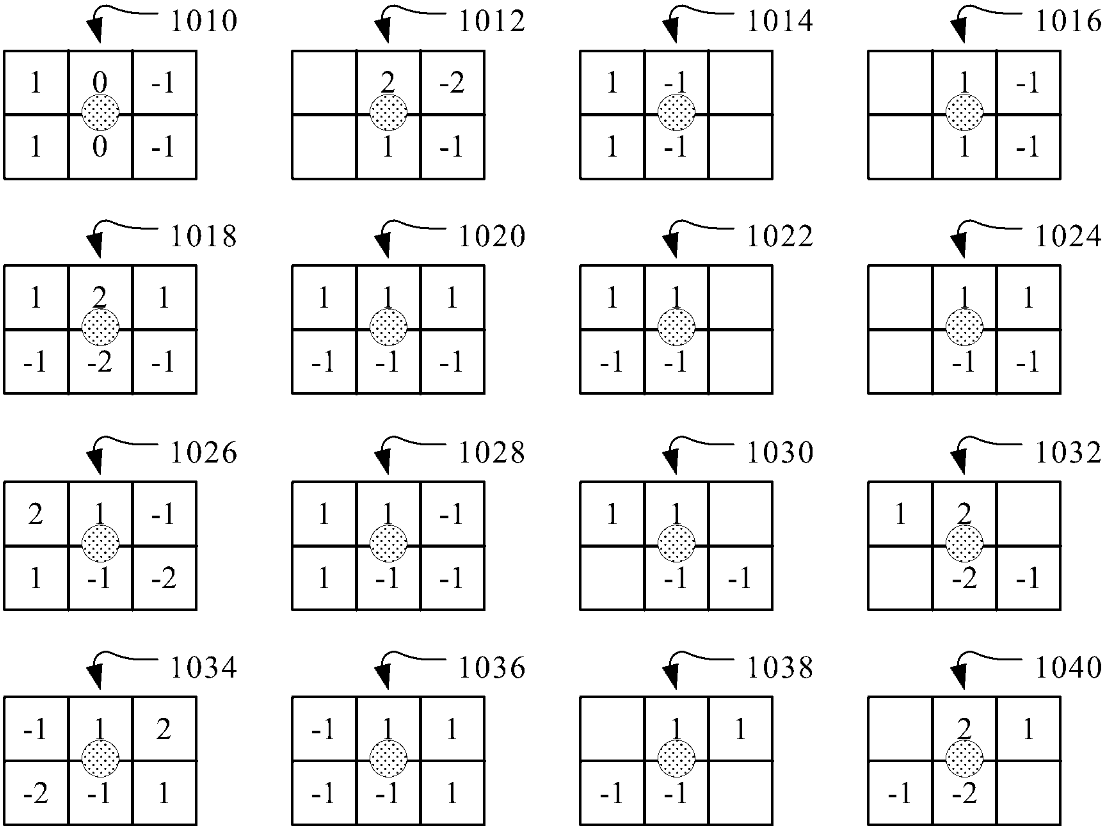
1010
1 0 -1
1 0 -1
1012
2 -2
1 -1
1014
1 -1
1 -1
1016
1 -1
1 -1
1018
1 2 1
-1 -2 -1
1020
1 1 1
-1 -1 -1
1022
1 1
-1 -1
1024
1 1
-1 -1
1026
2 1 -1
1 -1 -2
1028
1 1 -1
1 -1 -1
1030
1 1
-1 -1
1032
1 2
-2 -1
1034
-1 1 2
-2 -1 1
1036
-1 1 1
-1 -1 1
1038
1 1
-1 -1
1040
2 1
-1 -2

Fig. 10

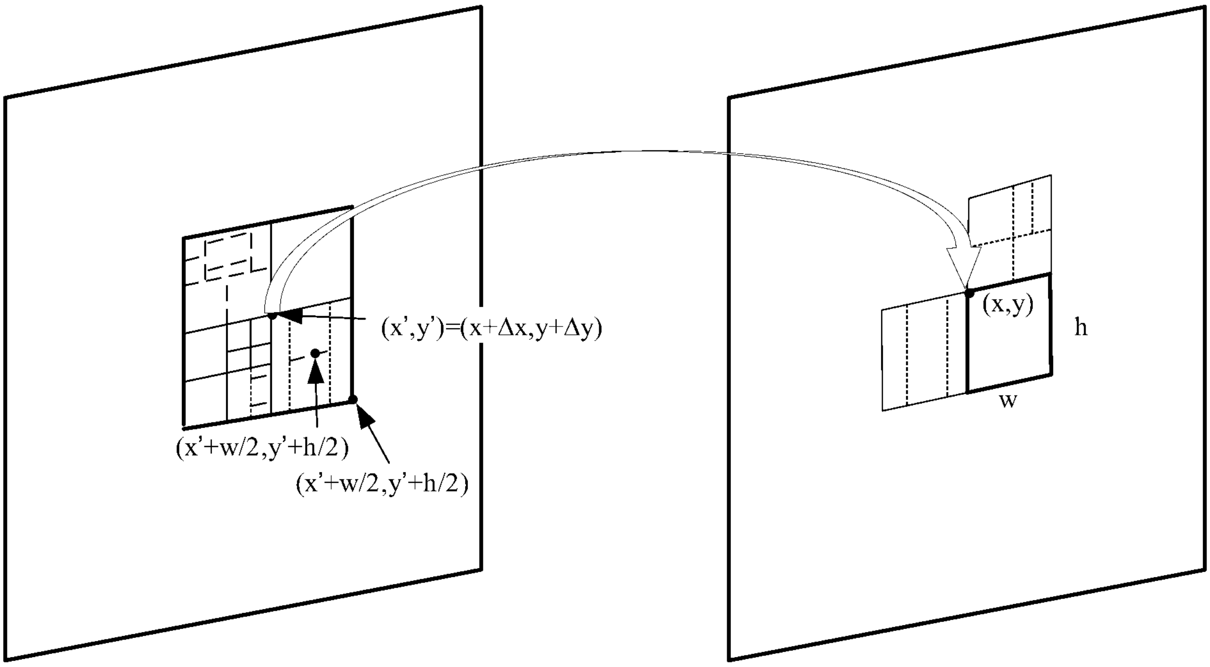
(x',y')=(x+Δx,y+Δy)
(x'+w/2,y'+h/2)
(x'+w/2,y'+h/2)
(x,y)
h
w

Fig. 11

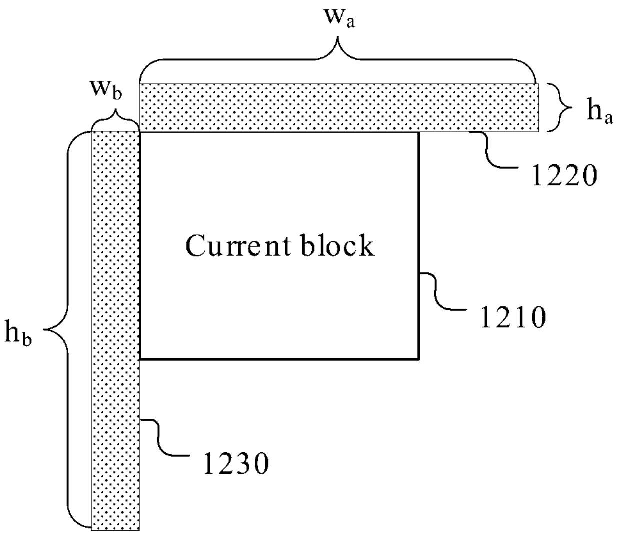
w_a
w_b
h_a
1220
Current block
1210
h_b
1230

Fig. 12

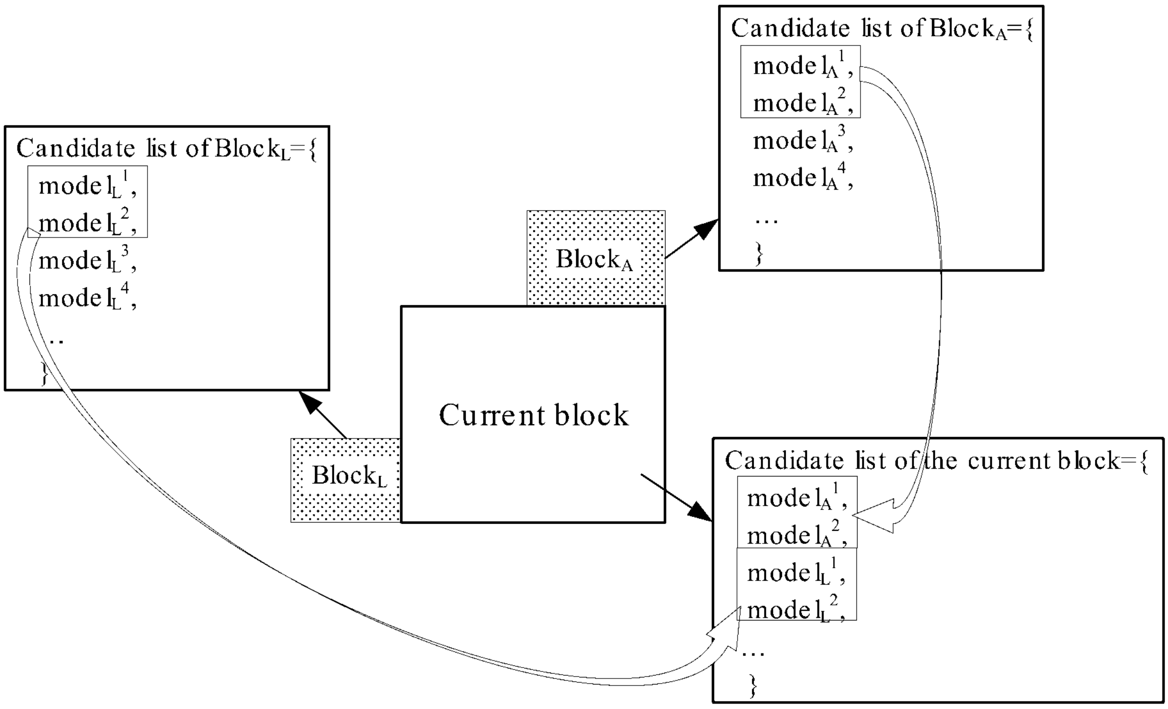
Candidate list of Block_A={
mode l_A^1,
mode l_A^2,
mode l_A^3,
mode l_A^4,
...
}
Candidate list of Block_L={
mode l_L^1,
mode l_L^2,
mode l_L^3,
mode l_L^4,
..
}
Block_A
Current block
Block_L
Candidate list of the current block={
mode l_A^1,
mode l_A^2,
mode l_L^1,
mode l_L^2,
...
}

Fig. 13

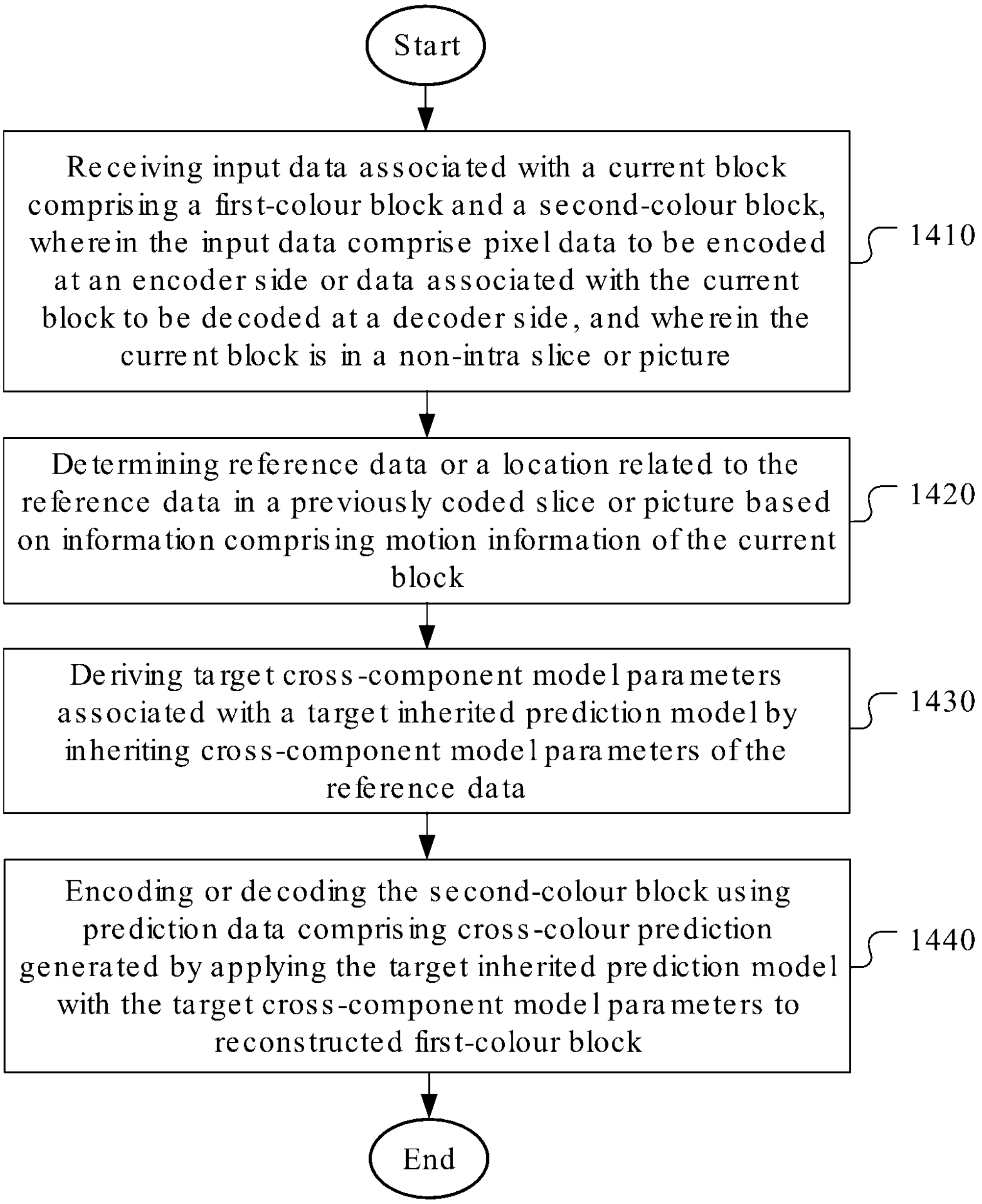
Start
Receiving input data associated with a current block comprising a first-colour block and a second-colour block, wherein the input data comprise pixel data to be encoded at an encoder side or data associated with the current block to be decoded at a decoder side, and wherein the current block is in a non-intra slice or picture
1410
Determining reference data or a location related to the reference data in a previously coded slice or picture based on information comprising motion information of the current block
1420
Deriving target cross-component model parameters associated with a target inherited prediction model by inheriting cross-component model parameters of the reference data
1430
Encoding or decoding the second-colour block using prediction data comprising cross-colour prediction generated by applying the target inherited prediction model with the target cross-component model parameters to reconstructed first-colour block
1440
End

Fig. 14

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2023/123158

A. CLASSIFICATION OF SUBJECT MATTER

H04N19/176(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04N

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

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

CNABS,CNKI,CNTXT,VEN,DWPI,WPABS,WPABSC,ENTXT,ENTXTC,CJFD,3GPP: coding, cross component, current block, encode, decode, non intra, reference, motion information, model parameter, inherit, inherited prediction model, luma, chroma, CCLM, MMLM, CCCM, neighbouring

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2018077426 A1 (QUALCOMM INC) 15 March 2018 (2018-03-15) the whole document	1-11
A	US 2020154115 A1 (QUALCOMM INC) 14 May 2020 (2020-05-14) the whole document	1-11
A	CN 106664425 A (QUALCOMM INC) 10 May 2017 (2017-05-10) the whole document	1-11
A	US 2021160515 A1 (GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP LTD) 27 May 2021 (2021-05-27) the whole document	1-11
A	WO 2020096877 A1 (INTERDIGITAL VC HOLDINGS INC) 14 May 2020 (2020-05-14) the whole document	1-11
A	WO 2021244935 A1 (NOKIA TECHNOLOGIES OY) 09 December 2021 (2021-12-09) the whole document	1-11

☐ 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 **10 January 2024**	Date of mailing of the international search report **10 January 2024**
Name and mailing address of the ISA/CN **CHINA NATIONAL INTELLECTUAL PROPERTY ADMINISTRATION** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088, China**	Authorized officer **LIU,YuanYuan** Telephone No. **(+86) 010-62411513**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2023/123158

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2018077426 A1	15 March 2018	US 10652575 B2	12 May 2020
		ES 2884375 T3	10 December 2021
		EP 3513559 A1	24 July 2019
		EP 3513559 B1	04 August 2021
		KR 20190046852 A	07 May 2019
		KR 102534901 B1	19 May 2023
		BR 112019004544 A2	28 May 2019
		WO 2018053293 A1	22 March 2018
		SG 11201900967 XA	29 April 2019
		TW 201817236 A	01 May 2018
		TWI 776818 B	11 September 2022
		JP 2019530330 A	17 October 2019
		JP 7044765 B2	30 March 2022
US 2020154115 A1	14 May 2020	US 11197005 B2	07 December 2021
CN 106664425 A	10 May 2017	WO 2015196119 A1	23 December 2015
		US 2015373349 A1	24 December 2015
		US 10200700 B2	05 February 2019
		JP 2017523672 A	17 August 2017
		EP 3158743 A1	26 April 2017
US 2021160515 A1	27 May 2021	KR 20210042355 A	19 April 2021
		BR 112021002191 A2	04 May 2021
		EP 3823278 A1	19 May 2021
		MX 2021001569 A	28 April 2021
		US 11503312 B2	15 November 2022
		AU 2018435559 A1	11 March 2021
		SG 11202101331 RA	30 March 2021
		IL 280692 A	25 March 2021
		WO 2020029187 A1	13 February 2020
		CA 3109008 A1	13 February 2020
		JP 2022500890 A	04 January 2022
		PH 12021550290 A1	11 October 2021
WO 2020096877 A1	14 May 2020	EP 3878183 A1	15 September 2021
		KR 20210083353 A	06 July 2021
		US 2022078405 A1	10 March 2022
		IL 282804 A	30 June 2021
		JP 2022506628 A	17 January 2022
WO 2021244935 A1	09 December 2021	JP 2023527920 A	30 June 2023
		CA 3177794 A1	09 December 2021
		US 2023262223 A1	17 August 2023
		EP 4162688 A1	12 April 2023