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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 December 2020 (24.12.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/253861 A1

(51) International Patent Classification:
H04N 19/96 (2014.01) *H04N 19/00* (2014.01)

(21) International Application Number:
PCT/CN2020/097368

(22) International Filing Date:
22 June 2020 (22.06.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2019/092326
21 June 2019 (21.06.2019) CN

(71) Applicants: **BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD.** [CN/CN]; Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District, Beijing 100041 (CN). **BYTEDANCE INC.** [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) Inventors: **ZHANG, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **XU, Jizheng**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **LIU, Hongbin**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). **ZHU, Weijia**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **WANG, Yue**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN).

(74) Agent: **LIU, SHEN & ASSOCIATES**; 10th Floor, Building 1, 10 Caihefang Road, Haidian District, Beijing 100080 (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,

(54) Title: ADAPTIVE IN-LOOP COLOR-SPACE TRANSFORM FOR VIDEO CODING

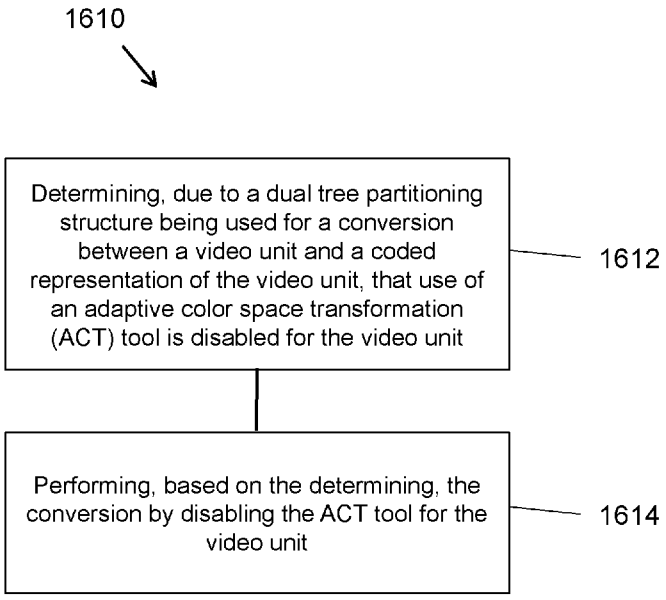


FIG. 16A

(57) Abstract: A method of video processing is provided. The method includes determining, due to a dual tree partitioning structure being used for a conversion between a video unit and a coded representation of the video unit, that use of an adaptive color space transformation (ACT) tool is disabled for the video unit; and performing, based on the determining, the conversion by disabling the ACT tool for the video unit, wherein the use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

WO 2020/253861 A1

[Continued on next page]

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

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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

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

ADAPTIVE IN-LOOP COLOR-SPACE TRANSFORM FOR VIDEO CODING

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefit of International Patent Application No. PCT/CN2019/092326, filed on June 21, 2019. For all purposes under the law, the entire disclosure of the aforementioned application is incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[0002] This patent document is directed generally to video coding and decoding technologies.

BACKGROUND

[0003] Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/ High Efficiency Video Coding (HEVC) standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM). In April 2018, the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the next generation Versatile Video Coding (VVC) standard targeting at 50% bitrate reduction compared to HEVC.

SUMMARY

[0004] Using the disclosed video coding, transcoding or decoding techniques, embodiments of video encoders or decoders can handle virtual boundaries of coding tree blocks to provide better compression efficiency and simpler implementations of coding or decoding tools.

[0005] In one example aspect, a method of video processing is disclosed. The method includes determining, due to a dual tree partitioning structure being used for a conversion between a video unit and a coded representation of the video unit, that use of an adaptive color space transformation

(ACT) tool is disabled for the video unit; and performing, based on the determining, the conversion by disabling the ACT tool for the video unit, wherein the use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[0006] In another example aspect, another method of video processing is disclosed. The method includes determining that a dual tree partitioning structure and an adaptive color space transformation (ACT) tool are used for a conversion between a video unit and a coded representation of the video unit; and performing, based on the determining, the conversion by enabling the ACT tool for the video unit, wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[0007] In another example aspect, another method of video processing is disclosed. The method includes determining, for a conversion between a current video block of a video and a coded representation of the video, that applicability of a first coding tool and a second coding tool is mutually exclusive; and performing the conversion based on the determining, wherein the first coding tool corresponds to an adaptive color space transformation (ACT) tool; wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[0008] In another example aspect, another method of video processing is disclosed. The method includes determining that both a coding tool and an adaptive color space transformation (ACT) tool are used for a conversion between a current video block of a video and a coded representation of the video; and performing, based on the determining, the conversion by enabling the ACT tool for the current video block, wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[0009] In another example aspect, another method of video processing is disclosed. The method includes determining, for a conversion between a current video block of a video unit of a video

and a coded representation of the video, that an adaptive color space transformation (ACT) tool is disabled for the conversion due to an in-loop reshaping (ILR) tool being enabled for the video unit; and performing, based on the determining, the conversion, and wherein the use of the ILR tool includes constructing the video unit based on a luma reshaping between a first domain and a second domain and/or a chroma residue scaling in a luma-dependent manner, and wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[0010] In another example aspect, another method of video processing is disclosed. The method includes determining that both an in-loop reshaping (ILR) tool and an adaptive color space transformation (ACT) tool are enabled for a conversion between a video unit and a coded representation of the video unit; and performing, based on the determining, the conversion, and wherein use of the ILR tool includes constructing the current video unit based on a first domain and a second domain and/or scaling chroma residue in a luma-dependent manner, and wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[0011] In another example aspect, another method of video processing is disclosed. The method includes determining that both a sub-block transform (SBT) tool and an adaptive color space transformation (ACT) coding tool are enabled for a conversion between a current video block and a coded representation of the current video block; and performing, based on the determining, the conversion, wherein use of the SBT tool comprises applying a transform process or an inverse transform process on a sub-part of a prediction residual block, and wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[0012] In another example aspect, another method of video processing is disclosed. The method includes performing a conversion between a video unit of a video and a coded representation of the video, where the video unit comprises one or more partitions at a first level comprising one or more partitions at a second level, wherein the coded representation conforms to a formatting rule, wherein the formatting rule specifies whether to include, or a partition level at which a syntax

element indicative of use of an adaptive color space transformation (ACT) tool for representing the one or more second level partitions in the coded representation is included in the coded representation, wherein the partition level is one of the first level, the second level or the video unit.

[0013] In yet another example aspect, a video encoding apparatus configured to perform an above-described method is disclosed.

[0014] In yet another example aspect, a video decoder that is configured to perform an above-described method is disclosed.

[0015] In yet another example aspect, a machine-readable medium is disclosed. The medium stores code which, upon execution, causes a processor to implement one or more of the above-described methods.

[0016] The above and other aspects and features of the disclosed technology are described in greater detail in the drawings, the description and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 shows an example of encoding flow with adaptive color-space transform (ACT).

[0018] FIG. 2 shows an example of a decoding flow with ACT.

[0019] FIG. 3 shows an example of neighbouring samples used for deriving IC parameters.

[0020] FIG. 4 shows an example flowchart of decoding flow with reshaping.

[0021] FIG. 5 is a reference line example.

[0022] FIG. 6 shows example of division of 4×8 and 8×4 blocks.

[0023] FIG. 7 shows an example of division of all blocks except 4×8, 8×4 and 4×4.

[0024] FIG. 8 is an illustration of ALWIP for 4×4 blocks.

[0025] FIG. 9 is an illustration of ALWIP for 8x8 blocks.

[0026] FIG. 10 is an illustration of ALWIP for 8×4 blocks.

[0027] FIG. 11 is an illustration of ALWIP for 16x16 blocks.

[0028] FIG. 12 is an illustration of sub-block transform modes SBT-V and SBT-H (The grey area is a TU which may have non-zero coefficients; the white area is a zero-out TU with all zero coefficients).

[0029] FIG. 13 is an illustration of sub-block transform modes SBT-Q.

[0030] FIGS. 14 and 15 are block diagrams of an example apparatus for video processing.

[0031] FIGS. 16A and 16B are flowcharts for example methods of video processing based on some implementations of the disclosed technology.

[0032] FIGS. 17A to 17E are flowcharts for example methods of video processing based on some implementations of the disclosed technology.

[0033] FIG. 18 is a flowchart for an example method of video processing based on some implementations of the disclosed technology.

DETAILED DESCRIPTION

[0034] Section headings are used in the present document to facilitate ease of understanding and do not limit the embodiments disclosed in a section to only that section. Furthermore, while certain embodiments are described with reference to Versatile Video Coding or other specific video codecs, the disclosed techniques are applicable to other video coding technologies also. Furthermore, while some embodiments describe video coding steps in detail, it will be understood that corresponding steps decoding that undo the coding will be implemented by a decoder. Furthermore, the term video processing encompasses video coding or compression, video decoding or decompression and video transcoding in which video pixels are represented from one compressed format into another compressed format or at a different compressed bitrate.

1. Summary

[0035] This document is related to video coding technologies. Specifically, it is related to interactions of adaptive color-space transform with other tools in video coding. It may be applied to the existing video coding standard like HEVC, or the standard (Versatile Video Coding) to be finalized. It may be also applicable to future video coding standards or video codec.

2. Background

[0036] Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal

prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM). In April 2018, the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC.

[0037] The latest version of VVC draft, i.e., Versatile Video Coding (Draft 5) could be found at:

[0038] http://phenix.it-sudparis.eu/jvet/doc_end_user/documents/14_Geneva/wg11/JVET-N1001-v2.zip

[0039] The latest reference software of VVC, named VTM, could be found at:

[0040] https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-5.0

2.1. Adaptive color-space transform (ACT) in HEVC Screen Content Coding Extensions

[0041] In the HEVC SCC Extensions, several tools have been proposed and employed to improve the SCC efficiency under the HEVC framework. For example, to exploit the repeated patterns in SC, an intra block copying (IBC) scheme was adopted. Similar to the motion compensation scheme used for inter pictures, the IBC mode searches for the repeated patterns in the already reconstructed region of the current picture. Another direction to improve SCC is to reduce the inter-color-component redundancy for the RGB/YCbCr sequences in the 4:4:4 chroma format.

[0042] The cross-component prediction (CCP) technology signals a weighting parameter index for each chroma color component of a transform unit. CCP provides good coding efficiency improvements with limited added complexity and thus, it was adopted to the HEVC Range Extensions and is part of HEVC Ver. 2 which specifies descriptions of Range Extensions, and other Extensions.

[0043] In order to further exploit inter-color-components correlation for SCC, an in-loop adaptive color-space transform (ACT) for HEVC SCC Extensions was employed. The basic idea of ACT is to adaptively convert the prediction residual into a color space with reduced redundancy among the three-color components. Before and after that, the signal follows the existing coding path in

HEVC Range Extensions. To keep the complexity as low as possible, only one additional color-space (i.e., RGB to YCgCo-R inverse transform) is considered, which can be easily implemented with shift and add operations.

2.1.1. Transforms used in ACT

[0044] For lossy coding, the YCgCo transform is used while its reversible variant, i.e., YCgCo-R, is used for lossless coding.

[0045] The forward and inverse YCgCo transform process is listed as below: taking a pixel in (R, G, B) color format as an example:

$$\text{Forward:} \quad \begin{bmatrix} Y \\ Cg \\ Co \end{bmatrix} = \frac{1}{4} * \begin{bmatrix} 1 & 2 & 1 \\ -1 & 2 & -1 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\text{Inverse:} \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Cg \\ Co \end{bmatrix}$$

[0046] Different from YCgCo transform which could be implemented by a matrix multiplication, the reversible color-space transform, i.e., YCgCo-R, used in ACT can only be performed in lifting-based operation as follows:

$$\text{Forward:} \quad \begin{aligned} Co &= R - B \\ t &= B + (Co \gg 1) \\ Cg &= G - t \\ Y &= t + (Cg \gg 1) \end{aligned}$$

$$\text{Inverse:} \quad \begin{aligned} t &= Y - (Cg \gg 1) \\ G &= Cg + t \\ B &= t - (Co \gg 1) \\ R &= Co + b \end{aligned}$$

2.1.2. Usage of ACT

[0047] For each TU, a flag may be signaled to indicate the usage of color-space transform. In addition, for intra coded CUs, ACT is enabled only when the chroma and luma intra prediction modes are the same, i.e., the chroma block is coded with DM mode.

[0048] FIG. 1 shows the block diagram of the proposed method at the encoder with the residual signal derived from intra/inter prediction as the input. The proposed function blocks, including forward and reverse color-space transforms, are located in the coding loop and highlighted. As

shown in FIG. 1, after the intra- or inter-prediction process (including the prediction process for IBC mode), it is determined whether to perform the forward color-space transform. With the introduced color-space transform, the color space of the input signal may be converted to YCgCo with less correlation among the three-color components. After that, the original coding flow, such as CCP, integer transform (i.e., T in FIG. 1), if applicable, quantization (i.e., Q in FIG. 1) and entropy coding processes, is further invoked in order. Meanwhile, during the reconstruction or decoding process as depicted in FIG. 2, after the conventional inverse quantization (i.e., IQ in FIG. 2), inverse transform (i.e., IT in FIG. 2) and inverse CCP, if applicable, the inverse color transform is invoked to convert the coded residual back to the original color space. It should be noted that, the color-space conversion process is applied to the residual signal instead of the reconstruction signal. With such a method, the decoder only needs to perform the inverse color space transform process which could keep the complexity increase as low as possible. Furthermore, in ACT, fixed color space transforms, i.e., YCgCo and YCgCo-R, are utilized regardless of input color spaces.

2.2. Local illumination compensation in JEM

[0049] Local Illumination Compensation (LIC) is based on a linear model for illumination changes, using a scaling factor *a* and an offset *b*. And it is enabled or disabled adaptively for each inter-mode coded coding unit (CU).

[0050] When LIC applies for a CU, a least square error method is employed to derive the parameters *a* and *b* by using the neighbouring samples of the current CU and their corresponding reference samples. More specifically, as illustrated in FIG. 3, the subsampled (2:1 subsampling) neighbouring samples of the CU and the corresponding samples (identified by motion information of the current CU or sub-CU) in the reference picture are used.

2.2.1. Derivation of prediction blocks

[0051] The IC parameters are derived and applied for each prediction direction separately. For each prediction direction, a first prediction block is generated with the decoded motion information, then a temporary prediction block is obtained via applying the LIC model. Afterwards, the two temporary prediction blocks are utilized to derive the final prediction block.

[0052] When a CU is coded with merge mode, the LIC flag is copied from neighbouring blocks, in a way similar to motion information copy in merge mode; otherwise, an LIC flag is signalled

for the CU to indicate whether LIC applies or not.

[0053] When LIC is enabled for a picture, additional CU level RD check is needed to determine whether LIC is applied or not for a CU. When LIC is enabled for a CU, mean-removed sum of absolute difference (MR-SAD) and mean-removed sum of absolute Hadamard-transformed difference (MR-SATD) are used, instead of SAD and SATD, for integer pel motion search and fractional pel motion search, respectively.

[0054] To reduce the encoding complexity, the following encoding scheme is applied in the JEM.

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

2.3. Inter prediction methods in VVC

[0055] There are several new coding tools for inter prediction improvement, such as Adaptive motion vector difference resolution (AMVR) for signaling MVD, affine prediction mode, Triangular prediction mode (TPM), ATMVP, Generalized Bi-Prediction (GBI), Bi-directional Optical flow (BIO).

2.3.1. Coding block structure in VVC

[0056] In VVC, a Quad-Tree/Binary-Tree/Ternary Tree (QT/BT/TT) structure is adopted to divide a picture into square or rectangle blocks.

[0057] Besides QT/BT/TT, separate tree (a.k.a. Dual coding tree) is also adopted in VVC for I-frames. With separate tree, the coding block structure are signaled separately for the luma and chroma components.

2.4. In-loop reshaping (ILR) in JVET-M0427

[0058] The basic idea of in-loop reshaping (ILR) is to convert the original (in the first domain) signal (prediction/reconstruction signal) to a second domain (reshaped domain).

[0059] The in-loop luma reshaper is implemented as a pair of look-up tables (LUTs), but only one of the two LUTs need to be signaled as the other one can be computed from the signaled LUT. Each LUT is a one-dimensional, 10-bit, 1024-entry mapping table (1D-LUT). One LUT is a forward LUT, *FwdLUT*, that maps input luma code values Y_i to altered values Y_r : $Y_r = FwdLUT[Y_i]$. The other LUT is an inverse LUT, *InvLUT*, that maps altered code values Y_r to $\hat{Y}_i$: $\hat{Y}_i = InvLUT[Y_r]$. ($\hat{Y}_i$ represents the reconstruction values of Y_i.).

2.4.1. PWL model

[0060] Conceptually, piece-wise linear (PWL) is implemented in the following way:

[0061] Let x1, x2 be two input pivot points, and y1, y2 be their corresponding output pivot points for one piece. The output value y for any input value x between x1 and x2 can be interpolated by the following equation:

$$y = ((y2-y1)/(x2-x1)) * (x-x1) + y1$$

[0062] In fixed point implementation, the equation can be rewritten as:

$$y = ((m * x + 2^{FP_PREC-1}) >> FP_PREC) + c$$

[0063] where m is scalar, c is an offset, and FP_PREC is a constant value to specify the precision.

[0064] Note that in CE-12 software, the PWL model is used to precompute the 1024-entry FwdLUT and InvLUT mapping tables; but the PWL model also allows implementations to calculate identical mapping values on-the-fly without pre-computing the LUTs.

2.4.2. Luma reshaping

[0065] Test 2 of the in-loop luma reshaping (i.e., CE12-2 in the proposal) provides a lower complexity pipeline that also eliminates decoding latency for block-wise intra prediction in inter slice reconstruction. Intra prediction is performed in reshaped domain for both inter and intra slices.

[0066] Intra prediction is always performed in reshaped domain regardless of slice type. With such arrangement, intra prediction can start immediately after previous TU reconstruction is done. Such arrangement can also provide a unified process for intra mode instead of being slice dependent. FIG. 4 shows the block diagram of the CE12-2 decoding process based on mode.

[0067] CE12-2 also tests 16-piece piece-wise linear (PWL) models for luma and chroma residue scaling instead of the 32-piece PWL models of CE12-1.

[0068] FIG. 4 shows a flowchart of decoding flow with reshaping.

[0069] Inter slice reconstruction with in-loop luma reshaper in CE12-2 (light-green shaded blocks indicate signal in reshaped domain: luma residue; intra luma predicted; and intra luma reconstructed)

2.4.3. Luma-dependent chroma residue scaling

[0070] **Luma-dependent chroma residue scaling** is a multiplicative process implemented with fixed-point integer operation. Chroma residue scaling compensates for luma signal interaction with the chroma signal. Chroma residue scaling is applied at the TU level. More specifically, the following applies:

- For intra, the reconstructed luma is averaged.
- For inter, the prediction luma is averaged.

[0071] The average is used to identify an index in a PWL model. The index identifies a scaling factor cScaleInv. The chroma residual is multiplied by that number.

[0072] It is noted that the chroma scaling factor is calculated from forward-mapped predicted luma values rather than reconstructed luma values

2.4.3.1. Signalling of ILR side information

[0073] The parameters are (currently) sent in the tile group header (similar to ALF). These reportedly take 40-100 bits.

2.4.3.2. Usage of ILR

[0074] At the encoder side, each picture (or tile group) is firstly converted to the reshaped domain. And all the coding process is performed in the reshaped domain. For intra prediction, the neighboring block is in the reshaped domain; for inter prediction, the reference blocks (generated from the original domain from decoded picture buffer) are firstly converted to the reshaped domain. Then the residual is generated and coded to the bitstream.

[0075] After the whole picture (or tile group) finishes encoding/decoding, samples in the reshaped

domain are converted to the original domain, then deblocking filter and other filters are applied.

[0076] Forward reshaping to the prediction signal is disabled for the following cases:

- Current block is intra-coded
- Current block is coded as CPR (current picture referencing, aka intra block copy, IBC)
- Current block is coded as combined inter-intra mode (CIIP) and the forward reshaping is disabled for the intra prediction block

2.5. Virtual Pipelining Data Units (VPDU)

[0077] Virtual pipeline data units (VPDUs) are defined as non-overlapping MxM-luma (L)/ NxN-chroma(C) units in a picture. In hardware decoders, successive VPDUs are processed by multiple pipeline stages at the same time; different stages process different VPDUs simultaneously. The VPDU size is roughly proportional to the buffer size in most pipeline stages, so it is said to be very important to keep the VPDU size small. In HEVC hardware decoders, the VPDU size is set to the maximum transform block (TB) size. Enlarging the maximum TB size from 32x32-L/16x16-C (as in HEVC) to 64x64-L/32x32-C (as in the current VVC) can bring coding gains, which results in 4X of VPDU size (64x64-L/32x32-C) expectedly in comparison with HEVC. However, in addition to quadtree (QT) coding unit (CU) partitioning, ternary tree (TT) and binary tree (BT) are adopted in VVC for achieving additional coding gains, and TT and BT splits can be applied to 128x128-L/64x64-C coding tree blocks (CTUs) recursively, which is said to lead to 16X of VPDU size (128x128-L/64x64-C) in comparison with HEVC.

[0078] In current design of VVC, the VPDU size is defined as 64x64-L/32x32-C.

2.6. Multiple reference line (MRL)

[0079] Multiple reference line (MRL) intra prediction uses more reference lines for intra prediction. In FIG. 5, an example of 4 reference lines is depicted, where the samples of segments A and F are not fetched from reconstructed neighbouring samples but padded with the closest samples from Segment B and E, respectively. HEVC intra-picture prediction uses the nearest reference line (i.e., reference line 0). In MRL, 2 additional lines (reference line 1 and reference line 3) are used.

[0080] The index of selected reference line (mrl_idx) is signaled and used to generate intra predictor. For reference line index, which is greater than 0, only include additional reference line modes in MPM list and only signal MPM index without remaining mode. The reference line index

is signaled before intra prediction modes, and Planar and DC modes are excluded from intra prediction modes in case a nonzero reference line index is signaled.

[0081] FIG. 5 is an example of four reference lines neighboring to a prediction block.

[0082] MRL is disabled for the first line of blocks inside a CTU to prevent using extended reference samples outside the current CTU line. Also, PDPC is disabled when additional line is used.

2.7. Intra subblock partitioning (ISP)

[0083] In JVET-M0102, ISP is proposed, which divides luma intra-predicted blocks vertically or horizontally into 2 or 4 sub-partitions depending on the block size dimensions, as shown in Table 1. FIG. 6 and FIG. 7 show examples of the two possibilities. All sub-partitions fulfill the condition of having at least 16 samples. For block sizes, 4×N or N×4 (with N>8), if allowed, the 1×N or N×1 sub-partition may exist.

Table 1: Number of sub-partitions depending on the block size (denoted maximum transform size by maxTBSize)

Splitting direction	Block Size	Number of Sub-Partitions
N/A	minimum transform size	Not divided
4x8: horizontal 8x4: vertical	4 × 8 and 8 × 4	2
Signaled	If neither 4x8 nor 8x4, and W<= maxTBSize and H<= maxTBSize	4
Horizontal	If not above cases and H> maxTBSize	4
Vertical	If not above cases and H> maxTBSize	4

[0084] For each of these sub-partitions, a residual signal is generated by entropy decoding the coefficients sent by the encoder and then invert quantizing and invert transforming them. Then, the sub-partition is intra predicted and finally the corresponding reconstructed samples are obtained by adding the residual signal to the prediction signal. Therefore, the reconstructed values of each sub-partition will be available to generate the prediction of the next one, which will repeat the process and so on. All sub-partitions share the same intra mode.

[0085] Hereinafter, inner sub-partition is used to represent sub-partitions except the first sub-partition. If an ISP block is split in horizonal (vertical) direction, the first sub-partition means the above (left) sub-partition.

Table 2: Specification of trTypeHor and trTypeVer depending on predModeIntra

predModeIntra	trTypeHor	trTypeVer
INTRA_PLANAR, INTRA_ANGULAR31, INTRA_ANGULAR32, INTRA_ANGULAR34, INTRA_ANGULAR36, INTRA_ANGULAR37	(nTbW >= 4 && nTbW <= 16) ? DST-VII : DCT-II	(nTbH >= 4 && nTbH <= 16) ? DST-VII : DCT-II
INTRA_ANGULAR33, INTRA_ANGULAR35	DCT-II	DCT-II
INTRA_ANGULAR2, INTRA_ANGULAR4,…,INTRA_ANGULAR28, INTRA_ANGULAR30, INTRA_ANGULAR39, INTRA_ANGULAR41,…,INTRA_ANGULAR63, INTRA_ANGULAR65	(nTbW >= 4 && nTbW <= 16) ? DST-VII : DCT-II	DCT-II
INTRA_ANGULAR3, INTRA_ANGULAR5,…, INTRA_ANGULAR27, INTRA_ANGULAR29, INTRA_ANGULAR38, INTRA_ANGULAR40,…,INTRA_ANGULAR64, INTRA_ANGULAR66	DCT-II	(nTbH >= 4 && nTbH <= 16) ? DST-VII : DCT-II

2.8. Affine linear weighted intra prediction (ALWIP, a.k.a. Matrix based intra prediction)

[0086] Affine linear weighted intra prediction (ALWIP, a.k.a. Matrix based intra prediction (MIP)) is proposed in JVET-N0217.

2.8.1. Generation of the reduced prediction signal by matrix vector multiplication

[0087] The neighboring reference samples are firstly down-sampled via averaging to generate the reduced reference signal $bdry_{red}$. Then, the reduced prediction signal $pred_{red}$ is computed by calculating a matrix vector product and adding an offset:

$$pred_{red} = A \cdot bdry_{red} + b.$$

[0088] Here, A is a matrix that has $W_{red} \cdot H_{red}$ rows and 4 columns if $W = H = 4$ and 8 columns in all other cases. b is a vector of size $W_{red} \cdot H_{red}$.

2.8.2. Illustration of the entire ALWIP process

[0089] The entire process of averaging, matrix vector multiplication and linear interpolation is

illustrated for different shapes in FIG. 8, FIG. 9, FIG. 10 and FIG. 11. Note, that the remaining shapes are treated as in one of the depicted cases.

1. Given a 4×4 block, ALWIP takes two averages along each axis of the boundary. The resulting four input samples enter the matrix vector multiplication. The matrices are taken from the set S_0. After adding an offset, this yields the 16 final prediction samples. Linear interpolation is not necessary for generating the prediction signal. Thus, a total of $(4 \cdot 16)/(4 \cdot 4) = 4$ multiplications per sample are performed.
2. Given an 8×8 block, ALWIP takes four averages along each axis of the boundary. The resulting eight input samples enter the matrix vector multiplication. The matrices are taken from the set S_1. This yields 16 samples on the odd positions of the prediction block. Thus, a total of $(8 \cdot 16)/(8 \cdot 8) = 2$ multiplications per sample are performed. After adding an offset, these samples are interpolated vertically by using the reduced top boundary. Horizontal interpolation follows by using the original left boundary.
3. Given an 8×4 block, ALWIP takes four averages along the horizontal axis of the boundary and the four original boundary values on the left boundary. The resulting eight input samples enter the matrix vector multiplication. The matrices are taken from the set S_1. This yields 16 samples on the odd horizontal and each vertical positions of the prediction block. Thus, a total of $(8 \cdot 16)/(8 \cdot 4) = 4$ multiplications per sample are performed. After adding an offset, these samples are interpolated horizontally by using the original left boundary.

   The transposed case is treated accordingly.
4. Given a 16×16 block, ALWIP takes four averages along each axis of the boundary. The resulting eight input samples enter the matrix vector multiplication. The matrices are taken from the set S_2. This yields 64 samples on the odd positions of the prediction block. Thus, a total of $(8 \cdot 64)/(16 \cdot 16) = 2$ multiplications per sample are performed. After adding an offset, these samples are interpolated vertically by using eight averages of the top boundary. Horizontal interpolation follows by using the original left boundary. The interpolation process, in this case, does not add any multiplications. Therefore, totally, two multiplications per sample are required to calculate ALWIP prediction.

[0090] For larger shapes, the procedure is essentially the same and it is easy to check that the

number of multiplications per sample is less than four.

[0091] For $W \times 8$ blocks with $W > 8$, only horizontal interpolation is necessary as the samples are given at the odd horizontal and each vertical position.

[0092] Finally, for $W \times 4$ blocks with $W > 8$, let A_k be the matrix that arises by leaving out every row that corresponds to an odd entry along the horizontal axis of the down-sampled block. Thus, the output size is 32 and again, only horizontal interpolation remains to be performed.

The transposed cases are treated accordingly.

2.8.3. Adapted MPM-list derivation for conventional luma and chroma intra-prediction modes

[0093] The proposed ALWIP-modes are harmonized with the MPM-based coding of the conventional intra-prediction modes as follows. The luma and chroma MPM-list derivation processes for the conventional intra-prediction modes uses fixed tables $map_alwip_to_angular_{idx}$, $idx \in \{0,1,2\}$, mapping an ALWIP-mode $predmode_{ALWIP}$ on a given PU to one of the conventional intra-prediction modes

$$predmode_{Angular} = map_alwip_to_angular_{idx(PU)}[predmode_{ALWIP}].$$

[0094] For the luma MPM-list derivation, whenever a neighboring luma block is encountered which uses an ALWIP-mode $predmode_{ALWIP}$, this block is treated as if it was using the conventional intra-prediction mode $predmode_{Angular}$. For the chroma MPM-list derivation, whenever the current luma block uses an LWIP-mode, the same mapping is used to translate the ALWIP-mode to a conventional intra prediction mode.

2.9. Quantized residual Block Differential Pulse-code Modulation(QR-BDPCM)

[0095] In JVET-M0413, a quantized residual block differential pulse-code modulation (QR-BDPCM) is proposed to code screen contents efficiently.

[0096] The prediction directions used in QR-BDPCM can be vertical and horizontal prediction modes. The intra prediction is done on the entire block by sample copying in prediction direction (horizontal or vertical prediction) similar to intra prediction. The residual is quantized and the delta between the quantized residual and its predictor (horizontal or vertical) quantized value is coded.

This can be described by the following: For a block of size M (rows) × N (cols), let $r_{i,j},\ 0 \le i \le M-1,\ 0 \le j \le N-1$ be the prediction residual after performing intra prediction horizontally (copying left neighbor pixel value across the the predicted block line by line) or vertically (copying top neighbor line to each line in the predicted block) using unfiltered samples from above or left block boundary samples. Let $Q(r_{i,j}),\ 0 \le i \le M-1,\ 0 \le j \le N-1$ denote the quantized version of the residual $r_{i,j}$, where residual is difference between original block and the predicted block values. Then the block DPCM is applied to the quantized residual samples, resulting in modified M × N array $\tilde{R}$ with elements $\tilde{r}_{i,j}$. When vertical BDPCM is signalled:

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0,\ \ 0 \le j \le (N-1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \le i \le (M-1),\ \ 0 \le j \le (N-1) \end{cases} \quad (2\text{-}7\text{-}1)$$

[0097] For horizontal prediction, similar rules apply, and the residual quantized samples are obtained by

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \le i \le (M-1),\ j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \le i \le (M-1),\ \ 1 \le j \le (N-1) \end{cases} \quad (2\text{-}7\text{-}2)$$

[0098] The residual quantized samples $\tilde{r}_{i,j}$ are sent to the decoder.

[0099] On the decoder side, the above calculations are reversed to produce $Q(r_{i,j}),\ 0 \le i \le M-1,\ 0 \le j \le N-1$. For vertical prediction case,

$$Q(r_{i,j}) = \sum_{k=0}^{i} \tilde{r}_{k,j},\quad 0 \le i \le (M-1),\ \ 0 \le j \le (N-1) \quad (2\text{-}7\text{-}3)$$

For horizontal case,

$$Q(r_{i,j}) = \sum_{k=0}^{j} \tilde{r}_{i,k},\quad 0 \le i \le (M-1),\ \ 0 \le j \le (N-1) \quad (2\text{-}7\text{-}4)$$

[00100] The inverse quantized residuals, $Q^{-1}\left(Q(r_{i,j})\right)$, are added to the intra block prediction values to produce the reconstructed sample values.

[00101] The main benefit of this scheme is that the inverse DPCM can be done on the fly during coefficient parsing simply adding the predictor as the coefficients are parsed or it can be performed after parsing.

2.10. Intra chroma coding

[00102] In addition to the intra chroma prediction modes, CCLM and joint chroma residual

coding are introduced in VVC.

2.10.1. Cross-component linear model (CCLM)

[00103] To reduce the cross-component redundancy, a cross-component linear model (CCLM) prediction mode is used in the VTM4, for which the chroma samples are predicted based on the reconstructed luma samples of the same CU by using a linear model as follows:

$$pred_C(i,j) = \alpha \cdot rec_L'(i,j) + \beta$$

[00104] where $pred_C(i,j)$ represents the predicted chroma samples in a CU and $rec_L(i,j)$ represents the downsampled reconstructed luma samples of the same CU. Linear model parameter α and β are derived from the relationship between luma values and chroma values from four samples at specific positions. Among the four samples, the two larger values are averaged, and the two smaller values are averaged. The averaged values are then utilized to derive the linear model parameters.

2.10.2. Joint chroma residual coding

[00105] If chrominance reshaper is active, reshaping is applied to the received residual identically to what is done in separate coding modes (that is, the joint residual signal is reshaped). On the encoder side the average of positive Cb residual and negative Cr residual are used as the joint residual when testing this mode:

$$resJoint = (resCb - resCr) / 2$$

[00106] One bin indicator is signaled in the bitstream to enable the mode. In the case the mode is enabled a joint residual signal is coded in the bitstream. On the decoder side the joint residual is used for Cb component and a negative version of the residual is applied for Cr.

2.11. Sub-block Transform (SBT) in VVC

[00107] For an inter-predicted CU with cu_cbf equal to 1, cu_sbt_flag may be signaled to indicate whether the whole residual block or a sub-part of the residual block is decoded. In the former case, inter MTS information is further parsed to determine the transform type of the CU. In the latter case, a part of the residual block is coded with inferred adaptive transform and the other part of the residual block is zeroed out. The SBT is not applied to the combined inter-intra mode, since almost no coding gain is achieved.

2.11.1. Sub-block TU tiling

[00108] When SBT is used for a inter CU, SBT type and SBT position information are further decoded from the bitstream. There are two SBT types and two SBT positions, as indicated in FIG. 12. For SBT-V (or SBT-H), the TU width (or height) may equal to half of the CU width (or height) or 1/4 of the CU width (or height), signaled by another flag, resulting in 2:2 split or 1:3/3:1 split. The 2:2 split is like a binary tree (BT) split while the 1:3/3:1 split is like an asymmetric binary tree (ABT) split. If one side of CU is 8 in luma samples, the 1:3/3:1 split along this side is not allowed. Hence, there are at most 8 SBT modes for a CU.

[00109] Quad-tree (QT) split is further used to tile one CU into 4 sub-blocks, and still one sub-block has residual, as shown in FIG. 13. This SBT type is denoted as SBT-Q. This part was not adopted by VVC.

[00110] SBT-V, SBT-H and SBT-Q are allowed for CU with width and height both no larger than maxSbtSize. The maxSbtSize is signaled in SPS. For HD and 4K sequences, maxSbtSize is set as 64 by encoder; for other smaller resolution sequences, maxSbtSize is set as 32.

2.11.2. Transform type of the sub-block

[00111] Position-dependent transform is applied on luma transform blocks in SBT-V and SBT-H (chroma TB always using DCT-2). The two positions of SBT-H and SBT-V are associated with different core transforms. More specifically, the horizontal and vertical transforms for each SBT position is specified in FIG. 12. For example, the horizontal and vertical transforms for SBT-V position 0 is DCT-8 and DST-7, respectively. When one side of the residual TU is greater than 32, the corresponding transform is set as DCT-2. Therefore, the sub-block transform jointly specifies the TU tiling, cbf, and horizontal and vertical transforms of a residual block, which may be considered a syntax shortcut for the cases that the major residual of a block is at one side of the block.

[00112] FIG. 12 is an illustration of sub-block transform modes SBT-V and SBT-H (The grey area is a TU which may have non-zero coefficients; the white area is a zero-out TU with all zero coefficients).

[00113] FIG. 13 is an illustration of sub-block transform modes SBT-Q.

2.12. Partition tree

[00114] In VTM5, the coding tree scheme supports the ability for the luma and chroma to have a separate block tree structure. Currently, 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 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.

3. Examples of technical problems addressed by the solutions described in this document

[00115] How to apply ACT to the VVC design needs to be studied, especially the interaction between ACT and other tools needs to be resolved:

1. ILR is to convert the luma component from a whole picture/slice/tile from the original domain to reshaped domain and code everything in the reshaped domain. However, for the chroma component, it is coded in the original domain. ACT requires to get the residual signal of pixels for three-color components.
2. How to handle ACT when dual tree is enabled.

4. Examples of solutions and embodiments

[00116] The listing below should be considered as examples to explain general concepts. These inventions should not be interpreted in a narrow way. Furthermore, these techniques can be combined in any manner.

[00117] In the following discussion, a CU may comprise information associated to all the three-color components with the single tree coding structure. Or a CU may comprise information only associated to the luma color component with the mono-color coding. Or a CU may comprise information only associated to the luma color component (e.g., Y component in YCbCr format or G component in GBR format) with the dual tree coding structure. Or a CU may comprise information only associated to the two chroma components (e.g., Cb and Cr components in YCbCr format or B and R components in GBR format) with the dual-tree coding structure.

[00118] In the following description, a "block" may refer to coding unit (CU) or a transform unit (TU) or any rectangle or polygonal region of video data. a "current block" may refer to a current being decoded/coded coding unit (CU) or a current being decoded/coded transform unit (TU) or any being decoded/coded coding rectangle region of video data. "CU" or "TU" may be also known as "coding block" and" transform block".

[00119] In the following discussions, the term 'ACT' may represent any technology that may convert the original signals/prediction signals/reconstructed signals/residual signals of three-color components from one domain to another domain, not necessarily to be the same design in HEVC SCC.

1. ACT is disabled for all blocks in a video unit when dual tree partitioning structure is enabled for the video unit (e.g., slice/tile/brick/picture/a region covering one or multiple CTUs).
    a. Indications of usage of ACT may be conditionally signaled based on the usage of the dual-tree partitioning structure.
        i. In one example, when the dual-tree partitioning structure is applied, the signaling of indications of usage of ACT may be skipped.
2. ACT is disabled for all blocks in a video unit when ILR is enabled for the video unit (e.g., slice/tile/brick/picture/a region covering one or multiple CTUs).
    a. Indications of usage of ACT may be conditionally signaled based on the usage of ILR.
        i. In one example, when ILR is applied, the signaling of indications of usage of ACT may be skipped.
3. ACT and a coding tool X are exclusively applied for a video block (e.g., CU/TU).
    a. In one example, X is CCLM. If CCLM is enabled for chroma components of the video block, ACT is disabled; and/or vice versa.
    b. In one example, X is joint chroma residual coding. If joint chroma residual coding is enabled for chroma components of the video block, ACT is disabled; and/or vice versa.

c. In one example, X is matrix based intra prediction method. If the matrix based intra prediction method is enabled for the luma component of the video block, ACT is disabled; and/or vice versa.
d. In one example, X is QR-BDPCM. If QR-BDPCM is enabled for the luma components of the video block, ACT is disabled; and/or vice versa.
e. In one example, X is sub-block transform (SBT). If SBT is enabled for the luma components of the video block, ACT is disabled; and/or vice versa.
f. In one example, X is multiple transform selection (MTS). If MTS is enabled for the luma components of the video block, ACT is disabled; and/or vice versa.
g. In one example, X is Low frequency non-separable transform (LFNST). If LFNST is enabled, ACT is disabled; and/or vice versa.
h. In one example, X is Pulse Code Modulation (PCM). If PCM is enabled, ACT is disabled; and/or vice versa.
i. In one example, X is Transform Skip (TS). If TS is enabled, ACT is disabled; and/or vice versa.
j. In one example, X is Intra Subblock Partitioning (ISP). If ISP is enabled, ACT is disabled; and/or vice versa.
k. Alternatively, furthermore, indications of usage of ACT may be conditionally signaled based on the usage of the coding tool X.
    i. In one example, when the coding tool X is enabled, the signaling of indications of usage of ACT may be skipped.
l. Alternatively, furthermore, indications of usage of tool X may be conditionally signaled based on the usage of ACT.
    i. In one example, when ACT is enabled, the signaling of indications of usage of the coding tool X may be skipped.
m. Alternatively, the above mentioned tools and ACT may be both enabled for one video block.

4. ACT and dual tree partition structure may be both enabled for one video unit (e.g., picture/slice/tile/brick)
    a. Alternatively, furthermore, the signaling of usage of dual tree partition structure is moved from video unit-level to video block (e.g., CTU/CTB or VPDU)-level.

b. ACT and dual tree partition structure may be both enabled for one video block
    i. In one example, at the encoder side, ACT may be firstly applied to a CTU/CTB before partitioning of the CTU/CTB.
    ii. In one example, at the decoder side, a CTU/CTB may be first decoded, followed by inverse color-space transform.

5. ACT and ILR may be both enabled for one video unit (e.g., picture/slice/tile/brick)
    a. Alternatively, furthermore, the signaling of usage of ILR is moved from video unit-level to video block (e.g., CU/TU)-level.
    b. ACT and ILR may be both enabled for one video block (e.g., CU/TU).
        i. In one example, at the encoder side, ACT may be firstly applied, followed by ILR. That is, the prediction signal and residual signal is firstly generated in the original domain, ACT is applied to convert the residual signal from the original domain to a different color-space domain; and ILR is further applied to convert the residual signal to the reshaped domain.
        ii. In one example, at the decoder side, ILR may be firstly applied, followed by inverse color-space transform. That is, ILR is firstly applied to convert the decoded residual signal from the reshaped domain to the color-space domain; then ACT is applied to convert from the color-space domain to the original domain.
6. ACT and SBT may be both enabled for one video block (e.g., CU/TU).
    a. In one example, the predictor error in a converted color-space domain (e.g., original domain is RGB, converted domain is YCoCg with ACT) is coded with two TUs. One of them is all zero coefficients and the other one has non-zero coefficients.
        i. Alternatively, furthermore, the one TU that has non-zero coefficients may be obtained via transforms or transform skip.
        ii. In one example, how to split it to 2 TUs; and/or what kinds of transforms may be applied to one of the two TUs may be signalled, e.g., in a similar way as SBT.
7. For a video unit (e.g., slice/tile/brick/picture), ACT may be enabled in different levels, such as CU-level and TU-level.

a. In one example, the signaling of usage of ACT may be in different levels, such as CU-level and TU-level, for different video blocks in the video unit.
b. Whether to enable/signal the ACT in CU or TU level and/or whether to signal the usage of ACT may be determined based on coding characteristics.
    i. In one example, whether to enable/signal the ACT in CU or TU level may be determined based on the dimensions of the current CU. Suppose the width and height of the current CU are denoted as W and H, respectively.
        1. For example, whether to enable/signal the ACT in CU or TU level may be determined based on whether the current CU has a size greater than the VPDU size.
            a. In one example, if current CU has a size greater than the VPDU size, CU-level signaling/usage of ACT may be applied (i.e., all TUs share the same on/off control of ACT). Otherwise, TU-level signaling/usage of ACT may be applied.
            b. Alternatively, if current CU has a size greater than the VPDU size, TU-level signaling/usage of ACT may be applied (i.e., all TUs share the same on/off control of ACT). Otherwise, CU-level signaling/usage of ACT may be applied.
            c. In one example, if current CU has a size greater than the VPDU size, ACT is disabled without being siganled.
        2. In one example, whether to enable/signal the ACT in CU or TU level may be determined based on the current CU has a size greater than the maximum TU size.
        3. In one example, ACT is disabled when W>=T1 and H >=T2. E.g. T1=T2=32.
            a. Alternatively, ACT is disabled when W>=T1 or H >=T2. E.g. T1=T2=32.
            b. Alternatively, ACT is disabled when W<=T1 and H <=T2. E.g. T1=T2=8.

c. Alternatively, ACT is disabled when W<=T1 or H <=T2. E.g. T1=T2=8.
d. Alternatively, ACT is disabled when W * H >=T. E.g. T = 1024.
e. Alternatively, ACT is disabled when W * H <=T. E.g. T = 64.

ii. In one example, whether to enable/signal the ACT in CU or TU level may be determined based on the current CU is coded with sub-block partition tools, such as ISP.

[00120] FIG. 14 is a block diagram of a video processing apparatus 1400. The apparatus 1400 may be used to implement one or more of the methods described herein. The apparatus 1400 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1400 may include one or more processors 1402, one or more memories 1404 and video processing hardware 1406. The processor(s) 1402 may be configured to implement one or more methods described in the present document. The memory (memories) 1404 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1406 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00121] FIG. 15 is another example of a block diagram of a video processing system in which disclosed techniques may be implemented. FIG. 15 is a block diagram showing an example video processing system 1510 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 1510. The system 1510 may include input 1512 for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input 1512 may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as Wi-Fi or cellular interfaces.

[00122] The system 1510 may include a coding component 1514 that may implement the various coding or encoding methods described in the present document. The coding component

1514 may reduce the average bitrate of video from the input 1512 to the output of the coding component 1514 to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component 1514 may be either stored, or transmitted via a communication connected, as represented by the component 1516. The stored or communicated bitstream (or coded) representation of the video received at the input 1512 may be used by the component 1518 for generating pixel values or displayable video that is sent to a display interface 1520. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as "coding" operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

[00123] Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on. Examples of storage interfaces include SATA (serial advanced technology attachment), PCI, IDE interface, and the like. The techniques described in the present document may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

[00124] In some embodiments, the video coding methods may be implemented using an apparatus that is implemented on a hardware platform as described with respect to FIG. 14 or 15.

[00125] FIG. 16A is a flowchart of an example method 1610 of video processing. The method 1610 includes, at step 1612, determining, due to a dual tree partitioning structure being used for a conversion between a video unit and a coded representation of the video unit, that use of an adaptive color space transformation (ACT) tool is disabled for the video unit. The method 1610 further includes, at step 1614, performing, based on the determining, the conversion by disabling the ACT tool for the video unit. In some implementations of the present method 1610 and other examples methods, use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[00126] FIG. 16B is a flowchart of an example method 1620 of video processing. The method 1620 includes, at step 1622, determining that a dual tree partitioning structure and an adaptive

color space transformation (ACT) tool are used for a conversion between a video unit and a coded representation of the video unit. The method 1620 further includes, at step 1624, performing, based on the determining, the conversion by enabling the ACT tool for the video unit.

[00127] FIG. 17A is a flowchart of an example method 1710 of video processing. The method 1710 includes, at step 1712, determining, for a conversion between a current video block of a video and a coded representation of the video, that applicability of a first coding tool and a second coding tool is mutually exclusive. The method 1710 further includes, at step 1714, performing the conversion based on the determining.

[00128] FIG. 17B is a flowchart of an example method 1720 of video processing. The method 1720 includes, at step 1712, determining that both a coding tool and an adaptive color space transformation (ACT) tool are used for a conversion between a current video block of a video and a coded representation of the video. The method 1720 further includes, at step 1724, performing, based on the determining, the conversion by enabling the ACT tool for the current video block.

[00129] FIG. 17C is a flowchart of an example method 1730 of video processing. The method 1730 includes, at step 1732, determining, for a conversion between a current video block of a video unit of a video and a coded representation of the video, that an adaptive color space transformation (ACT) tool is disabled for the conversion due to an in-loop reshaping (ILR) tool being enabled for the video unit. The method 1730 further includes, at step 1734, performing, based on the determining, the conversion. In some implementations of the present method 1730 and other examples methods, the use of the ILR tool includes constructing the video unit based on a luma reshaping between a first domain and a second domain and/or a chroma residue scaling in a luma-dependent manner.

[00130] FIG. 17D is a flowchart of an example method 1740 of video processing. The method 1740 includes, at step 1742, determining that both an in-loop reshaping (ILR) tool and an adaptive color space transformation (ACT) tool are enabled for a conversion between a video unit and a coded representation of the video unit. The method 1740 further includes, at step 1744, performing, based on the determining, the conversion.

[00131] FIG. 17E is a flowchart of an example method 1750 of video processing. The method 1750 includes, at step 1752, determining that both a sub-block transform (SBT) tool and an adaptive color space transformation (ACT) coding tool are enabled for a conversion between a current video block and a coded representation of the current video block. The method 1750 further

includes, at step 1754, performing, based on the determining, the conversion. In some implementations of the present method 1750 and other example methods, use of the SBT tool comprises applying a transform process or an inverse transform process on a sub-part of a prediction residual block.

[00132] FIG. 18 is a flowchart of an example method 1800 of video processing. The method 1800 includes, at step 1810, performing a conversion between a video unit of a video and a coded representation of the video, where the video unit comprises one or more partitions at a first level comprising one or more partitions at a second level. In some implementations, the coded representation conforms to a formatting rule, wherein the formatting rule specifies whether to include, or a partition level at which a syntax element indicative of use of an adaptive color space transformation (ACT) tool for representing the one or more second level partitions in the coded representation is included in the coded representation, wherein the partition level is one of the first level, the second level or the video unit.

[00133] Some embodiments of the disclosed technology include making a decision or determination to enable a video processing tool or mode. In an example, when the video processing tool or mode is enabled, the encoder will use or implement the tool or mode in the processing of a block of video, but may not necessarily modify the resulting bitstream based on the usage of the tool or mode. That is, a conversion from the block of video to the bitstream representation of the video will use the video processing tool or mode when it is enabled based on the decision or determination. In another example, when the video processing tool or mode is enabled, the decoder will process the bitstream with the knowledge that the bitstream has been modified based on the video processing tool or mode. That is, a conversion from the bitstream representation of the video to the block of video will be performed using the video processing tool or mode that was enabled based on the decision or determination.

[00134] Some embodiments of the disclosed technology include making a decision or determination to disable a video processing tool or mode. In an example, when the video processing tool or mode is disabled, the encoder will not use the tool or mode in the conversion of the block of video to the bitstream representation of the video. In another example, when the video processing tool or mode is disabled, the decoder will process the bitstream with the knowledge that the bitstream has not been modified using the video processing tool or mode that was disabled based on the decision or determination.

[00135] In the present document, the term "video processing" may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream.

[00136] Various solutions and embodiments described in the present document are further described using a list of clauses. The first set of clauses describe certain features and aspects of the disclosed techniques in the previous section.

[00137] Item 1 in previous section provides additional examples of the following clauses.

[00138] 1. A method of video processing, comprising: determining that due to a dual tree partitioning structure being used for a conversion between a video unit and a bitstream representation of the video unit, that use of adaptive color space transformation (ACT) is disabled for the video unit; and performing, based on the determining, the conversion by disabling the ACT for the vide unit.

[00139] 2. The method of clause 1, wherein the bitstream representation excludes bits for providing information about usage of the ACT.

[00140] Item 2 in previous section provides additional examples of the following clauses.

[00141] 1. A method of video processing, comprising: determining, due to an in-loop reshaper being enabled for use for a conversion between a video unit and a bitstream representation of the video unit, to disable an adaptive color space transformation (ACT) for the conversion; and performing, based on the determining, the conversion by disabling the ACT for the video unit.

[00142] 2. The method of clause 1, wherein the bitstream representation includes bits for providing information about usage of the ILR.

[00143] Item 3 in previous section provides additional examples of the following clauses.

[00144] 3. A method of video processing. comprising: determining, due to a coding tool being used for a conversion between a video unit and a bitstream representation of the video unit, that an adaptive color space transformation tool is disabled for the conversion; and performing, based

on the determining, the conversion by disabling the ACT for the video unit.

[00145] 4. The method of clause 3, wherein the coding tool corresponds to a cross-component linear model tool.

[00146] 5. The method of clause 3, wherein the coding tool corresponds to a joint chroma residual coding.

[00147] 6. The method of clause 3, wherein the coding tool corresponds to a multiple transform selection (MTS) coding tool.

[00148] Item 4 in previous section provides additional examples of the following clauses.

[00149] 7. A method of video processing, comprising: determining that both a dual tree partitioning structure and an adaptive color space transformation (ACT) coding tool are used for a conversion between a video unit and a bitstream representation of the video unit; and performing, based on the determining, the conversion by enabling the ACT for the video unit.

[00150] 8. The method of clause 7, wherein the bitstream representation includes signaling of the dual partition tree structure at a video block level that is a finer level than the video unit level.

[00151] Item 5 in previous section provides additional examples of the following clauses.

[00152] 9. A method of video processing, comprising: determining that both an in-loop reshaping (ILR) and an adaptive color space transformation (ACT) coding tool are used for a conversion between a video unit and a bitstream representation of the video unit; and performing, based on the determining, the conversion using the ILR and the ACT coding tool.

[00153] 10. The method of clause 9, wherein the performing the conversion includes, during encoding, first applying the ACT coding tool and next applying the ILR on a result of the applying the ACT.

[00154] 11. The method of clause 9, wherein the performing the conversion includes, first applying the ILR and then an inverse color space transform is applied to a result of the ILR.

[00155] Item 6 in previous section provides additional examples of the following clauses.

[00156] 12. A method of video processing, comprising: determining that both an (SBT) and an adaptive color space transformation (ACT) coding tool are used for a conversion between a video unit and a bitstream representation of the video unit; and performing, based on the determining, the conversion using the SBT and the ACT coding tool.

[00157] 13. The method of clause 12, wherein a prediction error during the conversion is transformed from an RGB color space to a YCoCg color space.

[00158] 14. The method of clause 13, wherein the prediction error, after the transforming, is coded using at least two transform uint (TU) partitions.

[00159] Item 7 in previous section provides additional examples of the following clauses.

[00160] 15. The method of any of above clauses, wherein the video unit comprises slice or a tile or a brick or a picture.

[00161] 16. The method of clause 15, wherein the determining is performed at a sub-video unit level, wherein the sub-unit level corresponds to a coding unit (CU) or a transform unit (TU).

[00162] 17. The method of clause 16, wherein the determining at the sub-video unit level is based on coding characteristics at the sub-unit level.

[00163] 18. The method of clause 17, wherein the coding characteristics include a size of the CU and/or a size of the TU and/or a relationship between the size of the CU and the size of the TU.

[00164] 19. The method of any of above clauses, wherein the conversion includes decoding the bitstream representation to generate the video unit.

[00165] 20. The method of any of above clauses, wherein the conversion includes encoding the video unit into the bitstream representation.

[00166] 21. A video encoder apparatus comprising a processor configured to implement a method recited in any one or more of above clauses.

[00167] 22. A video decoder apparatus comprising a processor configured to implement a method recited in any one or more of above clauses.

[00168] 23. A computer-readable medium having code for implementing a method described in any one or more of above clauses stored thereupon.

[00169] The second set of clauses describe certain features and aspects of the disclosed techniques in the previous section, for example, Example Implementations 1 and 4.

[00170] 1. A method of video processing, comprising: determining, due to a dual tree partitioning structure being used for a conversion between a video unit and a coded representation of the video unit, that use of an adaptive color space transformation (ACT) tool is disabled for the video unit; and performing, based on the determining, the conversion by disabling the ACT tool for the video unit, wherein the use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color

domain.

[00171] 2. The method of clause 1, wherein the visual signal comprises an original signal, a prediction signal, a reconstructed signal or a residual signal.

[00172] 3. The method of clause 1, wherein the video unit corresponds to a slice, a tile, a brick, a picture, or a video region covering one or more coding tree units.

[00173] 4. The method of clause 1, wherein the ACT tool is disabled for all coding blocks in the video unit and the coding blocks are generated based on the dual tree partitioning structure.

[00174] 5. The method of clause 1, wherein the dual tree partitioning structure uses separate partition trees for luma and chroma components.

[00175] 6. The method of clause 1, wherein the use of the ACT tool is signaled based on use of the dual tree partitioning structure.

[00176] 7. The method of clause 1, wherein the coded representation excludes bits for providing information about the use of the ACT tool due to usage of the dual tree partitioning structure.

[00177] 8. A method of video processing, comprising: determining that a dual tree partitioning structure and an adaptive color space transformation (ACT) tool are used for a conversion between a video unit and a coded representation of the video unit; and performing, based on the determining, the conversion by enabling the ACT tool for the video unit, wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[00178] 9. The method of clause 8, wherein the visual signal comprises an original signal, a prediction signal, a reconstructed signal or a residual signal.

[00179] 10. The method of clause 8, wherein the video unit corresponds to a slice, a tile, a brick, a picture, or a vide region covering one or more coding tree units.

[00180] 11. The method of clause 8, wherein the coded representation includes signaling of the dual partition tree structure at a video block level that is a finer level than the video unit level.

[00181] 12. The method of clause 8, wherein both the dual tree partitioning structure and the ACT tool are enabled for a video block of the video unit.

[00182] 13. The method of clause 12, wherein the performing the conversion includes, during encoding, applying the ACT tool to the video block prior to a partitioning of the video block.

[00183] 14. The method of clause 12, wherein the performing the conversion includes decoding the vide block and then performing an inverse color space transform on a result of the decoding.

[00184] 15. The method of any of clauses 1 to 14, wherein the conversion includes decoding the coded representation to generate the video unit.

[00185] 16. The method of any of clauses 1 to 14, wherein the conversion includes encoding the video unit into the coded representation.

[00186] 17. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of clauses 1 to 16.

[00187] 18. A computer readable medium storing program code that, when executed, causes a processor to implement a method recited in any one or more of clauses 1 to 16.

[00188] The third set of clauses describe certain features and aspects of the disclosed techniques in the previous section, for example, Example Implementations 2, 3, 5 and 6.

[00189] 1. A video processing method, comprising: determining, for a conversion between a current video block of a video and a coded representation of the video, that applicability of a first coding tool and a second coding tool is mutually exclusive; and performing the conversion based on the determining, wherein the first coding tool corresponds to an adaptive color space transformation (ACT) tool; wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[00190] 2. The method of clause 1, wherein the visual signal comprises an original signal, a prediction signal, a reconstructed signal or a residual signal.

[00191] 3. The method of clause 1, wherein the second coding tool corresponds to a cross-component linear model tool that uses a linear mode to derive prediction values of a chroma component from another component.

[00192] 4. The method of clause 1, wherein the second coding tool corresponds to a joint chroma residual coding tool in which prediction residual block of two color components are jointly processed.

[00193] 5. The method of clause 1, wherein the second coding tool corresponds to a matrix based intra prediction tool that includes generating a prediction signal based on predefined matrices and samples obtained along two axes of boundaries of the current video block.

[00194] 6. The method of clause 1, wherein the second coding tool corresponds to a quantized residual block differential pulse-code modulation (QR-BDPCM) tool that includes coding residual differences between quantized residual and prediction residual into the coded representation or deriving quantized residual from the residual differences included in the coded representation.

[00195] 7. The method of clause 1, wherein the second coding tool corresponds to a sub-block transform (SBT) tool in which the current video block is split into multiple sub-blocks and a transform or an inverse transform is only performed on a part of sub-blocks.

[00196] 8. The method of clause 1, wherein the second coding tool corresponds to a multiple transform selection (MTS) tool that selects, for the current video block, a transform among multiple transforms.

[00197] 9. The method of clause 1, wherein the second coding tool corresponds to a low frequency non-separable transform (LFNST) tool that includes applying, during encoding, a forward secondary transform to an output of a forward primary transform applied to a residual of a video block prior to quantization, or includes applying, during decoding, an inverse secondary transform to an output of dequantization to the video block before applying an inverse primary transform

[00198] 10. The method of clause 1, wherein the second coding tool corresponds to a pulse code modulation (PCM) tool that digitally represents a sampled analog signal.

[00199] 11. The method of clause 1, wherein the second coding tool corresponds to a transform skip (TS) mode in which a transform is bypassed or an identify transform is applied

[00200] 12. The method of clause 1, wherein the second coding tool corresponds to an intra subblock partitioning (ISP) tool that includes dividing luma intra-predicted blocks vertically or horizontally into sub-partitions.

[00201] 13. The method of clause 1, wherein the use of the first coding tool is signaled based on use of the second coding tool.

[00202] 14. The method of clause 13, wherein the coded representation excludes bits for providing information about the use of the first coding tool due to usage of the second coding tool.

[00203] 15. The method of clause 1, wherein use of the second coding tool is signaled based on the use of the first coding tool.

[00204] 16. The method of clause 15, wherein the coded representation excludes bits for

providing information about the use of the second coding tool due to usage of the first coding tool.

[00205] 17. A method of video processing, comprising: determining that both a coding tool and an adaptive color space transformation (ACT) tool are used for a conversion between a current video block of a video and a coded representation of the video; and performing, based on the determining, the conversion by enabling the ACT tool for the current video block, wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[00206] 18. The method of clause 17, wherein the coding tool comprises a matrix based intra prediction (MIP) tool, a sub-block transform (SBT) tool, a multiple transform selection (MTS), a low frequency non-separable transform (LFNST) tool, or a transform skip (TS) tool.

[00207] 19. A method of video processing, comprising: determining, for a conversion between a current video block of a video unit of a video and a coded representation of the video, that an adaptive color space transformation (ACT) tool is disabled for the conversion due to an in-loop reshaping (ILR) tool being enabled for the video unit; and performing, based on the determining, the conversion, and wherein the use of the ILR tool includes constructing the video unit based on a luma reshaping between a first domain and a second domain and/or a chroma residue scaling in a luma-dependent manner, and wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[00208] 20. The method of clause 19, wherein the video unit corresponds to a slice, a tile, a brick, a picture, or a video region covering one or more coding tree units.

[00209] 21. The method of clause 19, wherein the use of the ACT tool is signaled based on the use of the ILR tool.

[00210] 22. The method of clause 19, wherein the coded representation excludes bits for providing information about the use of the ACT due to usage of the ILR.

[00211] 23. A method of video processing, comprising: determining that both an in-loop reshaping (ILR) tool and an adaptive color space transformation (ACT) tool are enabled for a conversion between a video unit and a coded representation of the video unit; and performing, based on the determining, the conversion, and wherein use of the ILR tool includes constructing

the current video unit based on a first domain and a second domain and/or scaling chroma residue in a luma-dependent manner, and wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[00212] 24. The method of clause 23, wherein the video unit corresponds to a slice, a tile, a brick, a picture, or a video region covering one or more coding tree units.

[00213] 25. The method of clause 23, wherein the coded representation includes signaling of the dual partition tree structure at a video block level that is a finer level than the video unit level.

[00214] 26. The method of clause 23, wherein both the ILR tool and the ACT tool are enabled for a video block of the video unit.

[00215] 27. The method of clause 26, wherein the performing the conversion includes, during encoding, first applying the ACT tool to the video block and next applying the ILR tool on a result of the applying the ACT tool.

[00216] 28. The method of clause 26, wherein the performing the conversion includes first applying the ILR tool to the video block and then applying an inverse color space transform on a result of the ILR tool.

[00217] 29. A method of video processing, comprising: determining that both a sub-block transform (SBT) tool and an adaptive color space transformation (ACT) coding tool are enabled for a conversion between a current video block and a coded representation of the current video block; and performing, based on the determining, the conversion, wherein use of the SBT tool comprises applying a transform process or an inverse transform process on a sub-part of a prediction residual block, and wherein use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

[00218] 30. The method of clause 29, wherein a prediction error is converted from an RGB color space to a YCoCg color space before the transform process in encoding or converted from a YCoCg color space to an RGB color space after the inverse transform process in decoding.

[00219] 31. The method of clause 30, wherein the prediction error is coded using two

transform units (TUs) including a first transform unit (TU) and a second transform unit (TU).

[00220] 32. The method of clause 31, wherein the first transform unit (TU) has all zero coefficient and the second transform unit (TU) has non-zero coefficients.

[00221] 33. The method of clause 32, wherein the second transform unit (TU) is obtained by performing a transform or a transform skip.

[00222] 34. The method of clause 31, wherein the coded representation includes information as to how to split the current video block to the two transform units (TUs) and/or types of transforms applied to at least one of the two transform units (TUs).

[00223] 35. The method of any of clauses 1 to 34, wherein the conversion includes generating the video unit or the current video block from the coded representation.

[00224] 36. The method of any of clauses 1 to 34, wherein the conversion includes generating the coded representation from the video unit or the current video block.

[00225] 37. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of clauses 1 to 36.

[00226] 38. A computer readable medium storing program code that, when executed, causes a processor to implement a method recited in any one or more of clauses 1 to 36.

[00227] The fourth set of clauses describe certain features and aspects of the disclosed techniques in the previous section, for example, Example Implementation 7.

[00228] 1. A video processing method, comprising: performing a conversion between a video unit of a video and a coded representation of the video, where the video unit comprises one or more partitions at a first level comprising one or more partitions at a second level, wherein the coded representation conforms to a formatting rule, wherein the formatting rule specifies whether to include, or a partition level at which a syntax element indicative of use of an adaptive color space transformation (ACT) tool for representing the one or more second level partitions in the coded representation is included in the coded representation, wherein the partition level is one of the first level, the second level or the video unit.

[00229] 2. The method of clause 1, wherein the video unit corresponds to a slice or a tile or a brick or a picture.

[00230] 3. The method of clause 1, wherein the use of the ACT tool comprises: converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or converting, during decoding, a representation of a visual signal from the second color

domain to the first color domain.

[00231] 4. The method of any of clauses 1 to 3, wherein the formatting rule specifies that different levels for different video blocks in the video unit.

[00232] 5. The method of clause 4, wherein the different levels correspond to the coding unit (CU) and/or the transform unit (TU).

[00233] 6. The method of any of clauses 1 to 5, wherein the formatting rule specifies the partition level based on coding characteristics of the conversion.

[00234] 7. The method of clause 6, wherein the coding characteristics include a width (W) and/or a height (H) of a current coding unit.

[00235] 8. The method of clause 7, wherein the formatting rule specifies the partition level based on whether the current coding unit has a size greater than a size of a virtual pipelining data unit (VPDU).

[00236] 9. The method of clause 8, wherein the partition level corresponds to a coding unit (CU) level in a case that the current coding unit has a size greater than the size of the VPDU.

[00237] 10. The method of clause 8, wherein the partition level corresponds a transform unit (TU) level in a case that the current coding unit has a size greater than the size of the VPDU.

[00238] 11. The method of clause 8, wherein the formatting rule specifies not to include the syntax element due to the ACT tool disabled in a case that the current coding unit has a size greater than the size of the VPDU.

[00239] 12. The method of clause 7, wherein the formatting rule specifies the partition level based on whether the current coding unit has a size greater than a maximum size of the transform unit (TU).

[00240] 13. The method of clause 1, wherein the formatting rule specifies not to include the syntax element due to the ACT tool disabled in a case that W>=T1 and H >=T2, W and H corresponding to a width and a height of a current coding unit, respectively.

[00241] 14. The method of clause 1, wherein the formatting rule specifies not to include the syntax element due to the ACT tool disabled in a case that W>=T1 or H >=T2, W and H corresponding to a width and a height of a current coding unit, respectively.

[00242] 15. The method of clause 7, wherein the formatting rule specifies not to include the syntax element due to the ACT tool disabled in a case that W<=T1 and H <=T2, W and H corresponding to a width and a height of a current coding unit, respectively.

[00243] 16. The method of clause 7, wherein the formatting rule specifies not to include the syntax element due to the ACT tool disabled in a case that W<=T1 or H <=T2, W and H corresponding to a width and a height of a current coding unit, respectively.

[00244] 17. The method of clause 7, wherein the formatting rule specifies not to include the syntax element due to the ACT tool disabled in a case that W * H >=T, W and H corresponding to a width and a height of a current coding unit, respectively.

[00245] 18. The method of clause 7, wherein the formatting rule specifies not to include the syntax element due to the ACT tool disabled in a case that W * H <=T, W and H corresponding to a width and a height of a current coding unit, respectively.

[00246] 19. The method of clause 7, wherein the formatting rule specifies the partition level based on whether the current coding unit is coded with a sub-block partition tool.

[00247] 20. The method of any of clauses 1 to 19, wherein the conversion includes generating the video unit from the coded representation.

[00248] 21. The method of any of clauses 1 to 19, wherein the conversion includes generating the coded representation from the video unit.

[00249] 22. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of clauses 1 to 21.

[00250] 23. A computer readable medium storing program code that, when executed, causes a processor to implement a method recited in any one or more of clauses 1 to 21.

[00251] From the foregoing, it will be appreciated that specific embodiments of the presently disclosed technology have been described herein for purposes of illustration, but that various modifications may be made without deviating from the scope of the invention. Accordingly, the presently disclosed technology is not limited except as by the appended claims.

[00252] Implementations of the subject matter and the functional operations described in this patent document can be implemented in various systems, digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Implementations of the subject matter described in this specification can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a tangible and non-transitory computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage

device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more of them. The term "data processing unit" or "data processing apparatus" encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them.

[00253] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[00254] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

[00255] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or

transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of nonvolatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[00256] It is intended that the specification, together with the drawings, be considered exemplary only, where exemplary means an example. As used herein, the use of "or" is intended to include "and/or", unless the context clearly indicates otherwise.

[00257] While this patent document contains many specifics, these should not be construed as limitations on the scope of any invention or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this patent document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[00258] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described in this patent document should not be understood as requiring such separation in all embodiments.

[00259] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this patent document.

CLAIMS

1. A method of video processing, comprising:

determining, due to a dual tree partitioning structure being used for a conversion between a video unit and a coded representation of the video unit, that use of an adaptive color space transformation (ACT) tool is disabled for the video unit; and

performing, based on the determining, the conversion by disabling the ACT tool for the video unit,

wherein the use of the ACT tool comprises:

converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or

converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

2. The method of claim 1, wherein the visual signal comprises an original signal, a prediction signal, a reconstructed signal or a residual signal.

3. The method of claim 1, wherein the video unit corresponds to a slice, a tile, a brick, a picture, or a video region covering one or more coding tree units.

4. The method of claim 1, wherein the ACT tool is disabled for all coding blocks in the video unit and the coding blocks are generated based on the dual tree partitioning structure.

5. The method of claim 1, wherein the dual tree partitioning structure uses separate partition trees for luma and chroma components.

6. The method of claim 1, wherein the use of the ACT tool is signaled based on use of the dual tree partitioning structure.

7. The method of claim 1, wherein the coded representation excludes bits for providing information about the use of the ACT tool due to usage of the dual tree partitioning structure.

8. A method of video processing, comprising:

determining that a dual tree partitioning structure and an adaptive color space transformation (ACT) tool are used for a conversion between a video unit and a coded representation of the video unit; and

performing, based on the determining, the conversion by enabling the ACT tool for the video unit,

wherein use of the ACT tool comprises:

converting, during encoding, a representation of a visual signal from a first color domain to a second color domain, or

converting, during decoding, a representation of a visual signal from the second color domain to the first color domain.

9. The method of claim 8, wherein the visual signal comprises an original signal, a prediction signal, a reconstructed signal or a residual signal.

10. The method of claim 8, wherein the video unit corresponds to a slice, a tile, a brick, a picture, or a vide region covering one or more coding tree units.

11. The method of claim 8, wherein the coded representation includes signaling of the dual partition tree structure at a video block level that is a finer level than the video unit level.

12. The method of claim 8, wherein both the dual tree partitioning structure and the ACT tool are enabled for a video block of the video unit.

13. The method of claim 12, wherein the performing the conversion includes, during encoding, applying the ACT tool to the video block prior to a partitioning of the video block.

14. The method of claim 12, wherein the performing the conversion includes decoding the vide block and then performing an inverse color space transform on a result of the decoding.

15. The method of any of claims 1 to 14, wherein the conversion includes decoding the coded representation to generate the video unit.

16. The method of any of claims 1 to 14, wherein the conversion includes encoding the video unit into the coded representation.

17. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of claims 1 to 16.

18. A computer readable medium storing program code that, when executed, causes a processor to implement a method recited in any one or more of claims 1 to 16.

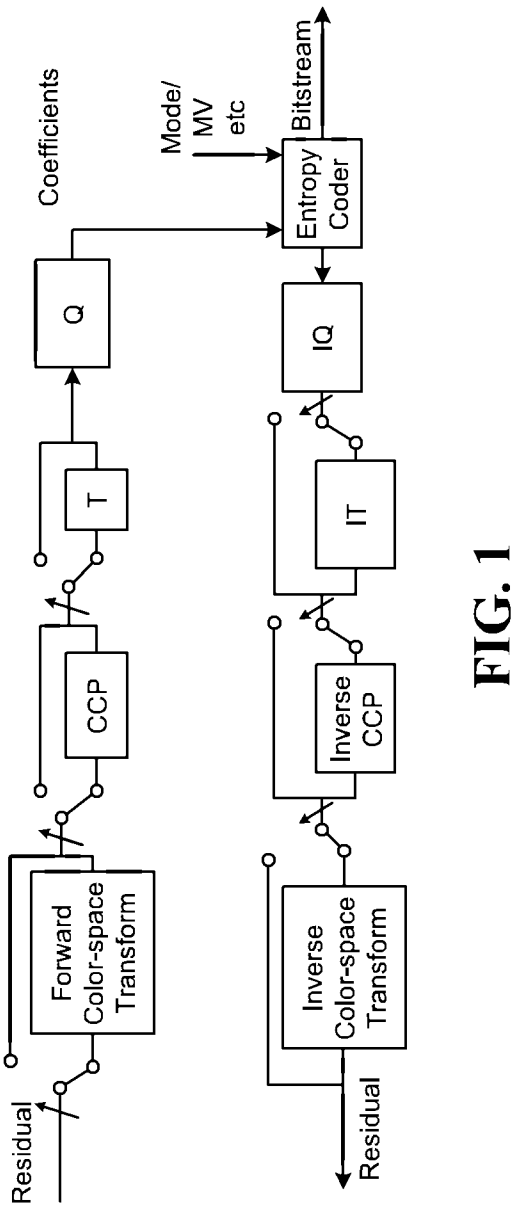

Residual
Forward Color-space Transform
CCP
T
Q
Coefficients
Mode/ MV etc
Entropy Coder
Bitstream
IQ
IT
Inverse CCP
Inverse Color-space Transform
Residual

FIG. 1

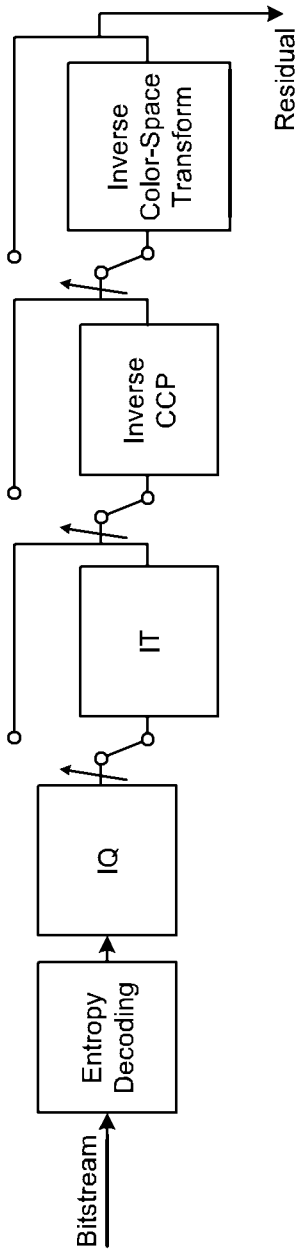

Bitstream
Entropy Decoding
IQ
IT
Inverse CCP
Inverse Color-Space Transform
Residual

FIG. 2

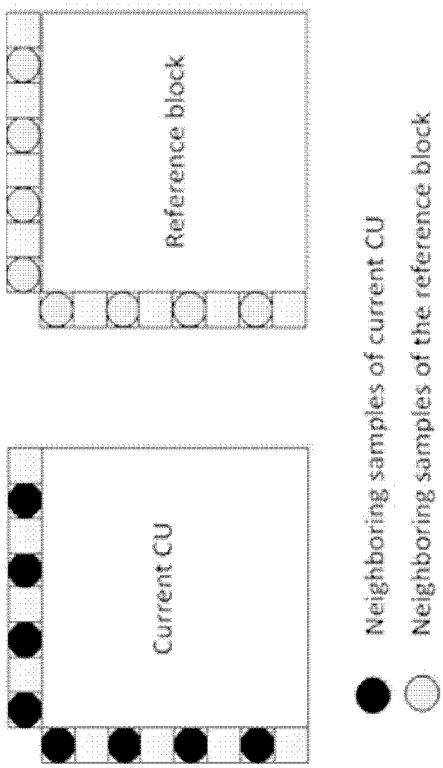

Reference block
Current CU
Neighboring samples of current CU
Neighboring samples of the reference block

FIG. 3

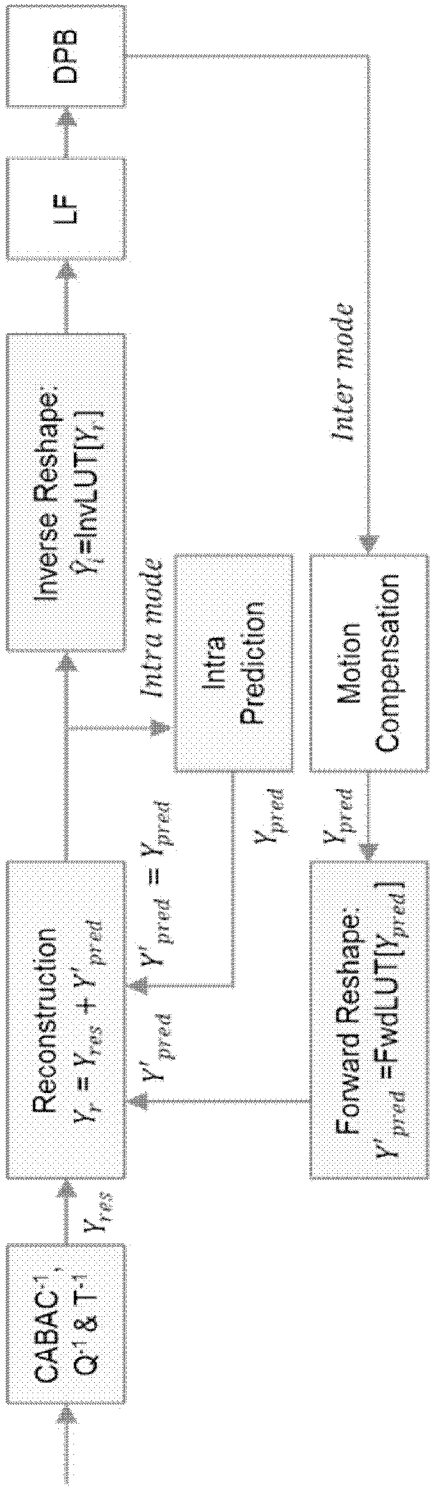

CABAC-1, Q-1 & T-1
Yres
Reconstruction
Yr = Yres + Y'pred
Y'pred
Y'pred = Ypred
Ypred
Inverse Reshape:
Ŷt=InvLUT[Yt]
LF
DPB
Intra mode
Intra Prediction
Ypred
Forward Reshape:
Y'pred =FwdLUT[Ypred]
Inter mode
Motion Compensation
Ypred

FIG. 4

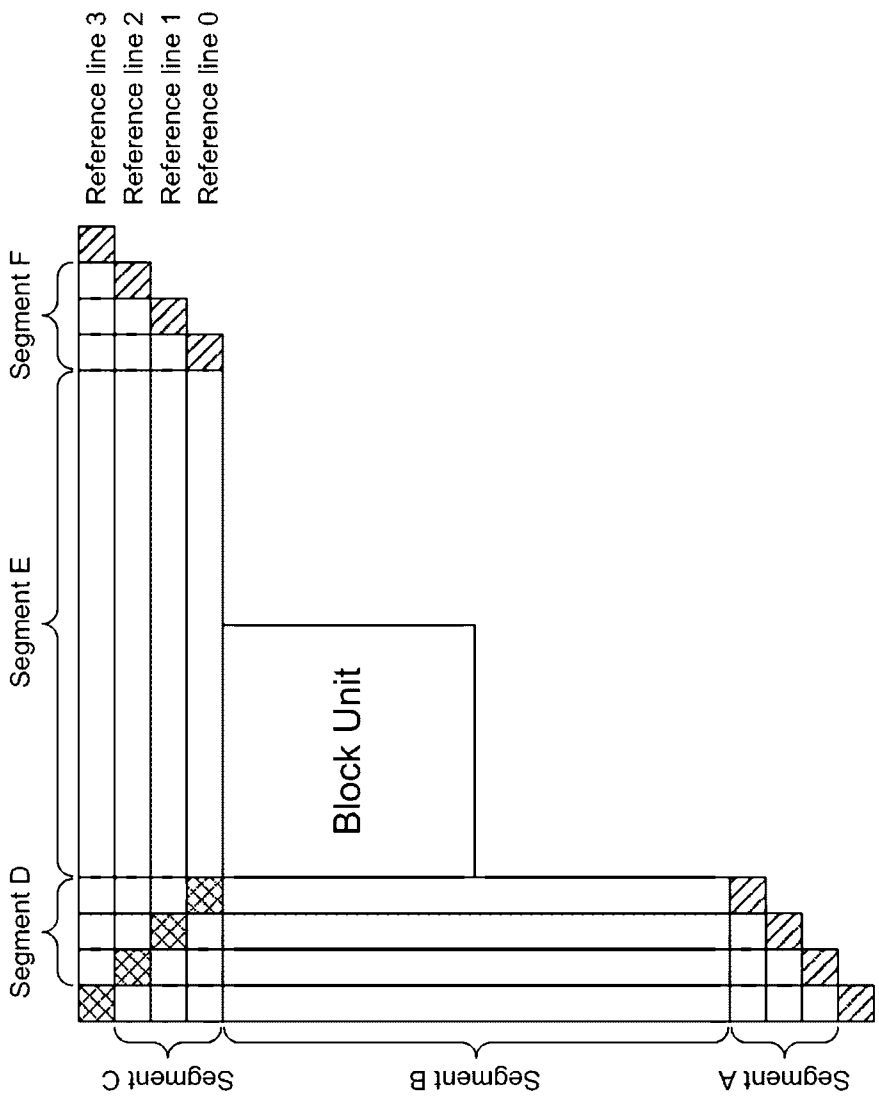

Reference line 3
Reference line 2
Reference line 1
Reference line 0
Segment F
Segment E
Segment D
Block Unit
Segment C
Segment B
Segment A

FIG. 5

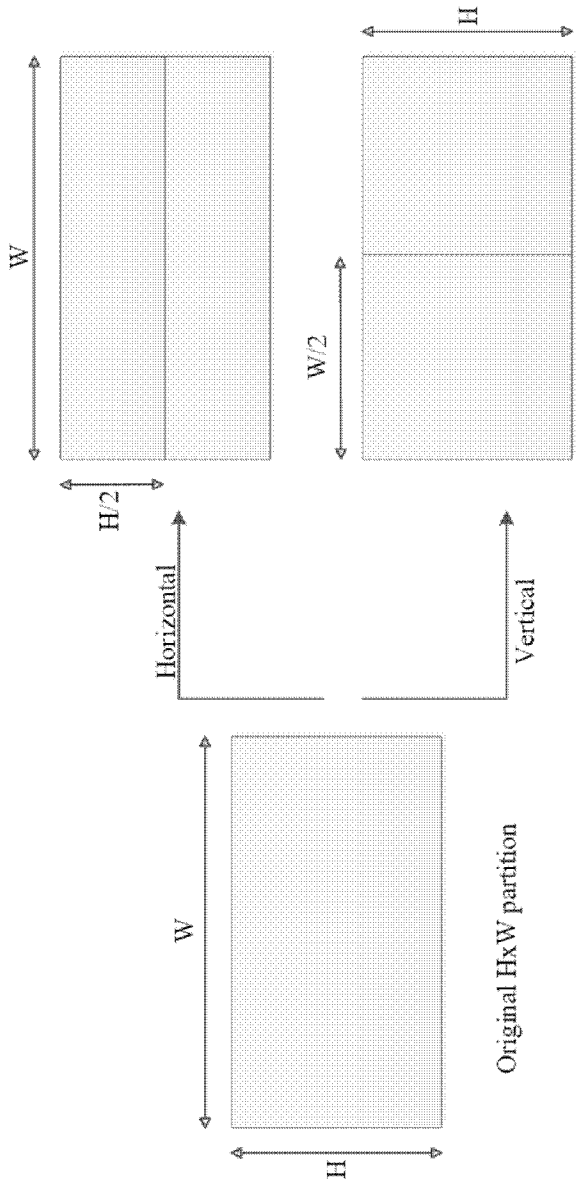


FIG. 6

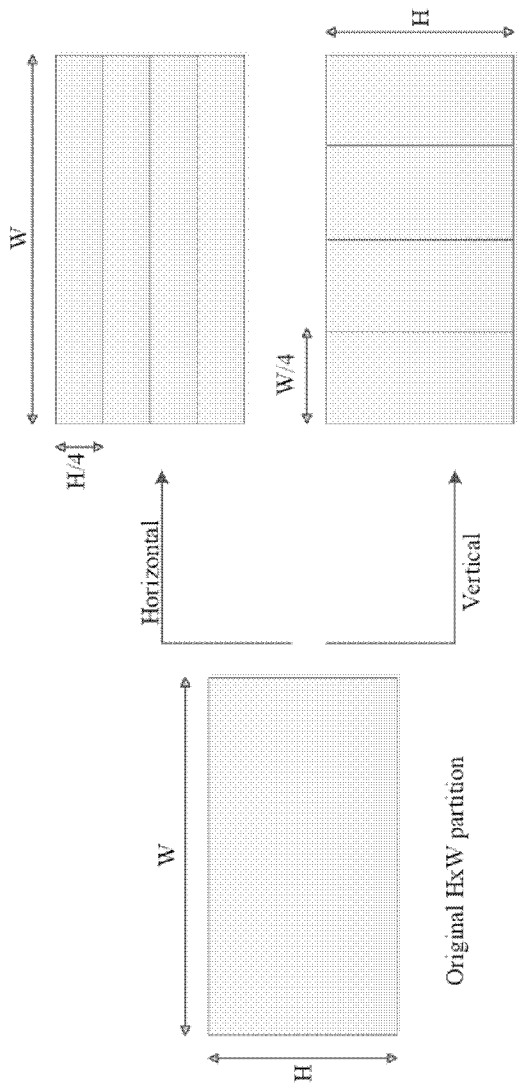


FIG. 7

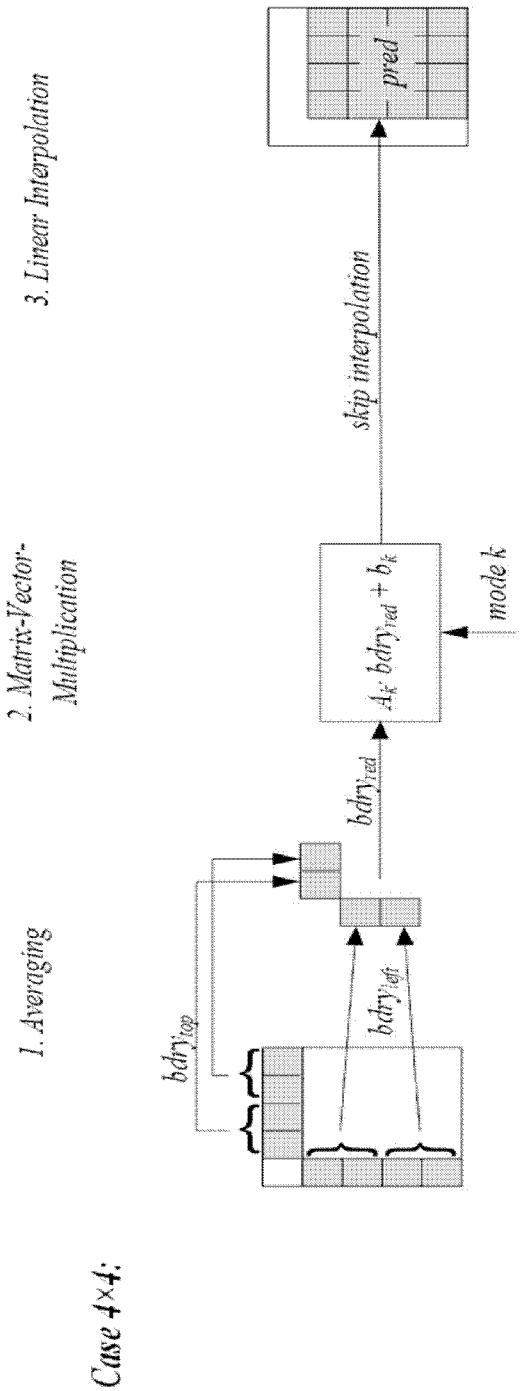

Case 4×4:
1. Averaging
2. Matrix-Vector-Multiplication
3. Linear Interpolation
$bdry_{top}$
$bdry_{left}$
$bdry_{red}$
$A_k \cdot bdry_{red} + b_k$
mode k
skip interpolation
pred

FIG.8

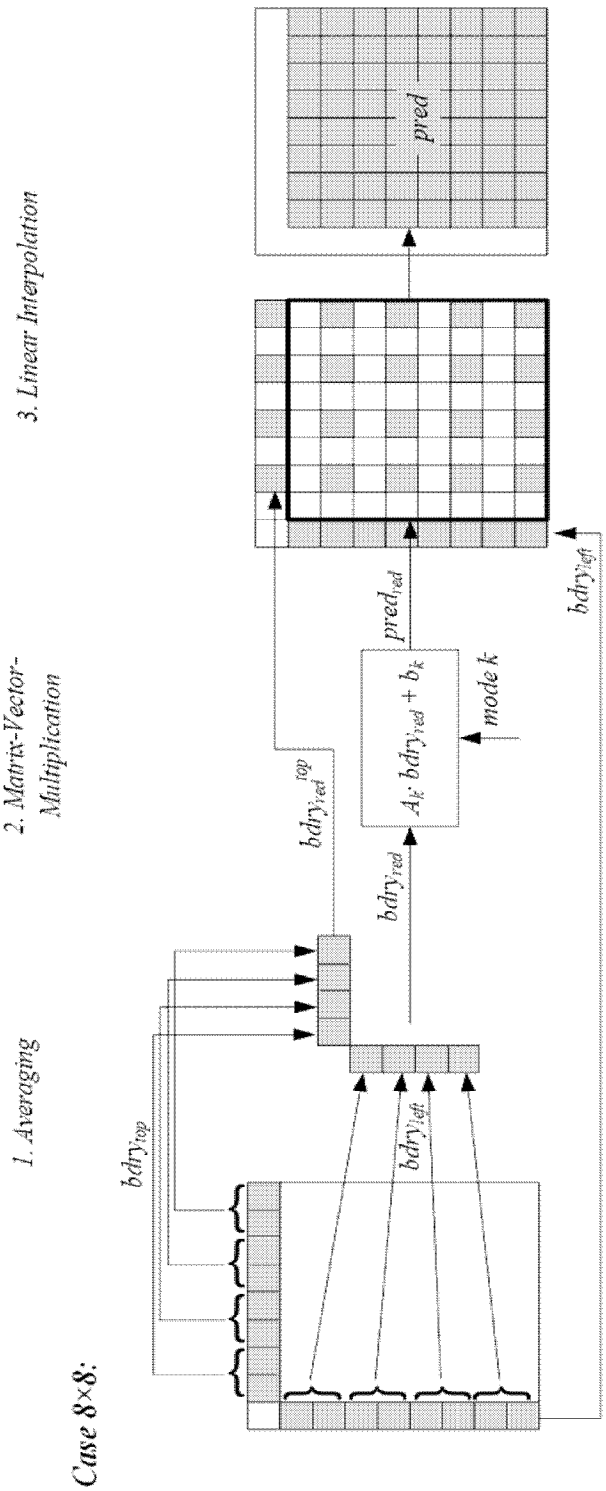

Case 8×8:
1. Averaging
2. Matrix-Vector-Multiplication
3. Linear Interpolation
$bdry_{top}$
$bdry_{left}$
$bdry_{red}$
$bdry_{red}^{top}$
$A_k \cdot bdry_{red} + b_k$
mode k
$pred_{red}$
$bdry_{left}$
pred

FIG. 9

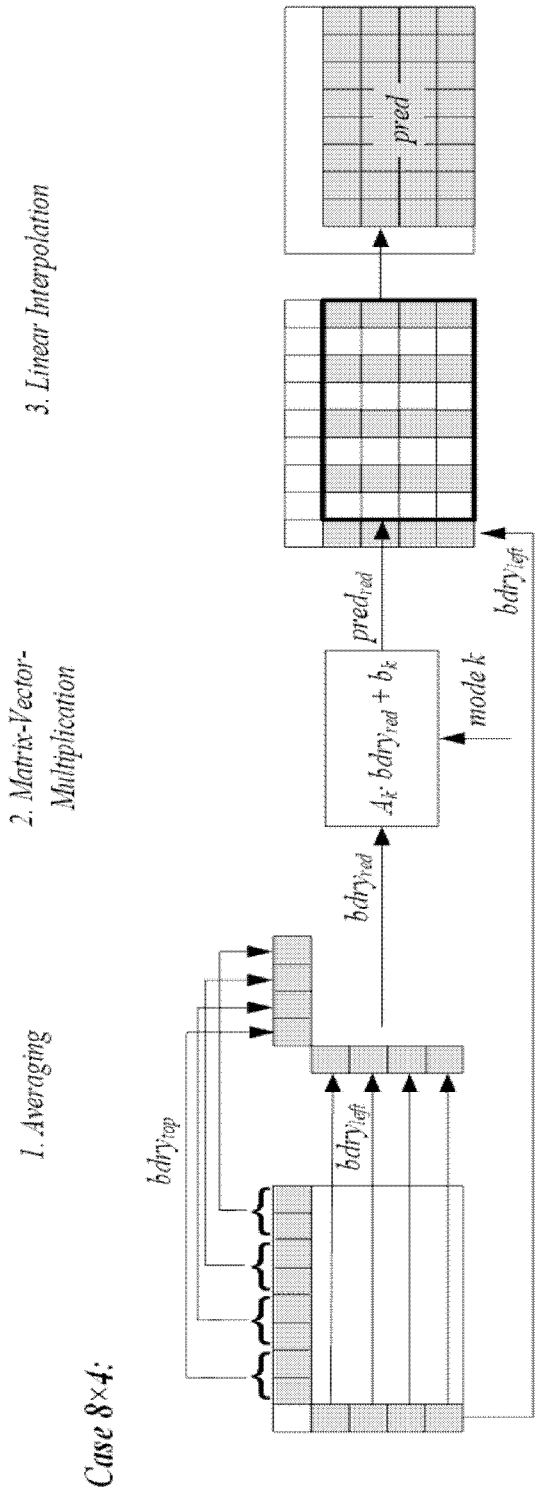

Case 8×4:
1. Averaging
2. Matrix-Vector-Multiplication
3. Linear Interpolation
$bdry_{top}$
$bdry_{left}$
$bdry_{red}$
$A_k \cdot bdry_{red} + b_k$
mode k
$pred_{red}$
$bdry_{left}$
pred

FIG. 10

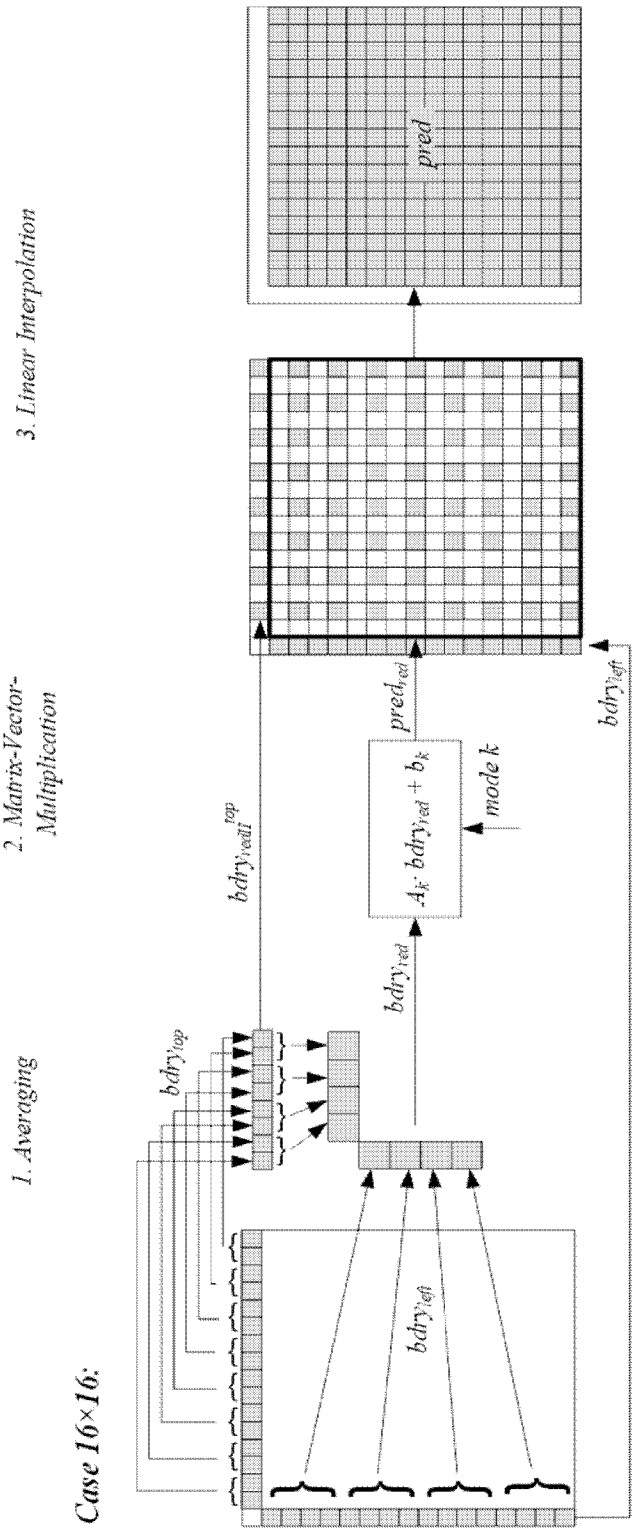

Case 16×16:
1. Averaging
2. Matrix-Vector-Multiplication
3. Linear Interpolation
$bdry_{top}$
$bdry_{left}$
$bdry_{red}$
$bdry_{red}^{top}$
$A_k \cdot bdry_{red} + b_k$
mode k
$pred_{red}$
pred
$bdry_{left}$

FIG. 11

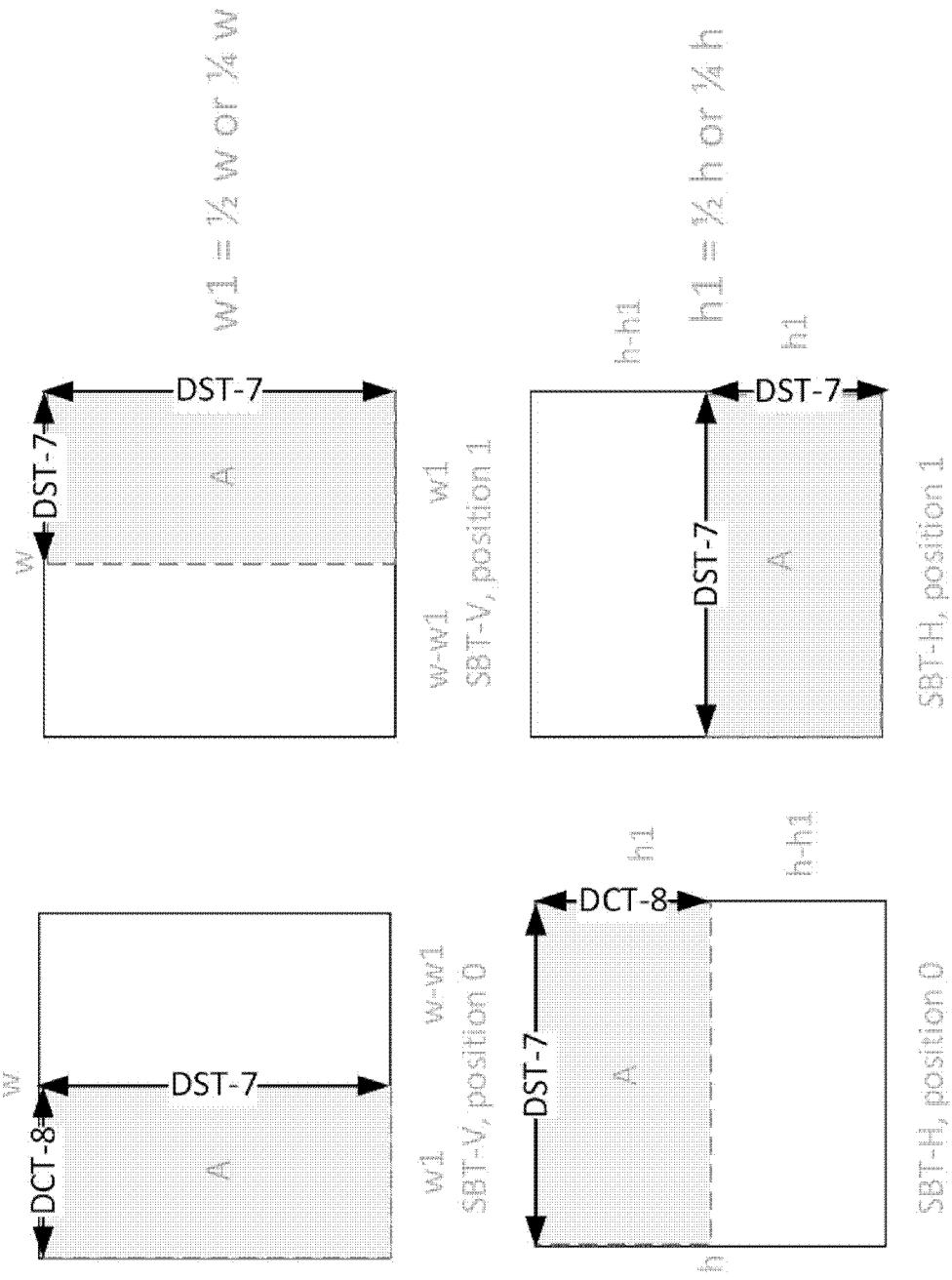

w1 = ½ w or ¼ w
DST-7
DST-7
A
w
w1
w-w1
SBT-V, position 1
h-h1
h1 = ½ h or ¼ h
h1
DST-7
DST-7
A
SBT-H, position 1
DCT-8
DST-7
A
w
w1
w-w1
SBT-V, position 0
h1
h-h1
DCT-7
DCT-8
A
h
SBT-H, position 0

FIG. 12

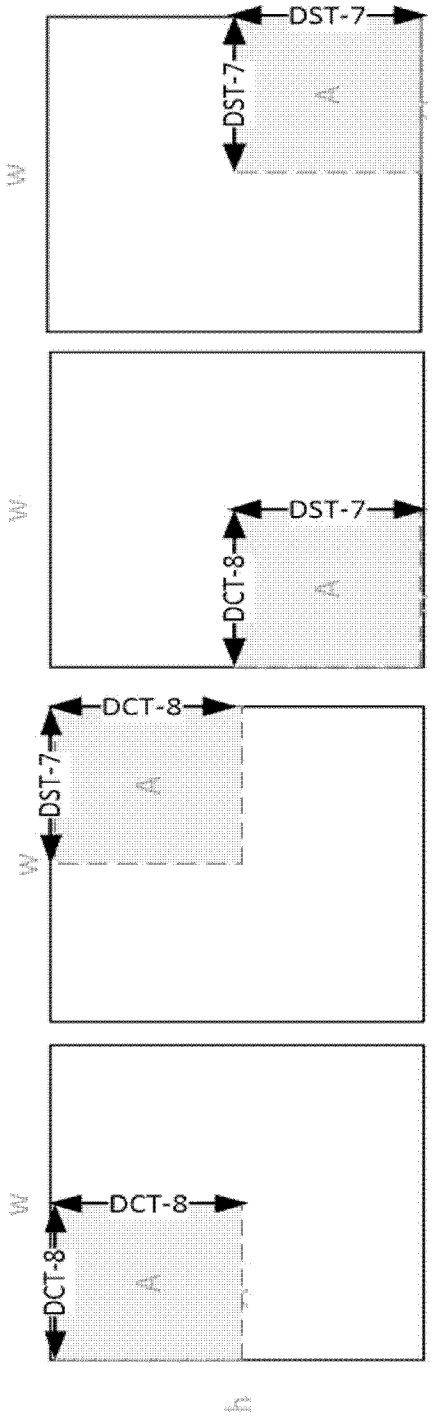


FIG. 13

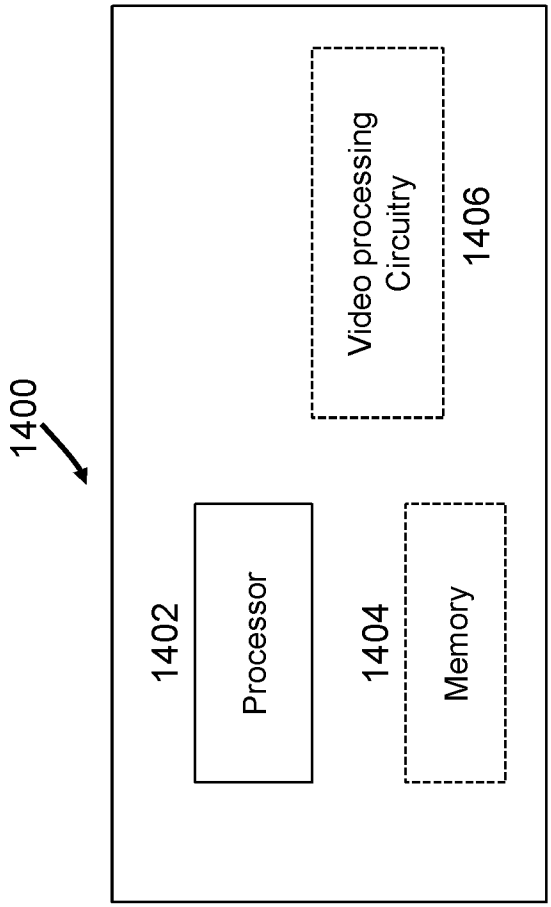

1400
1402
Processor
1404
Memory
Video processing Circuitry
1406

FIG. 14

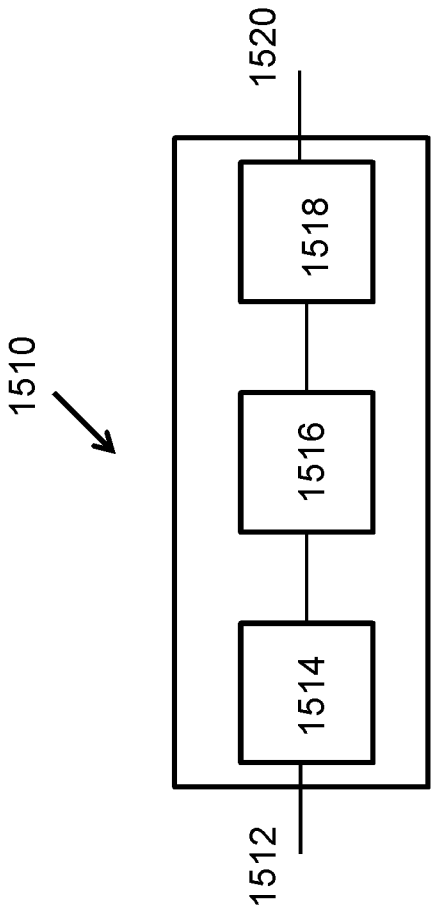


FIG. 15

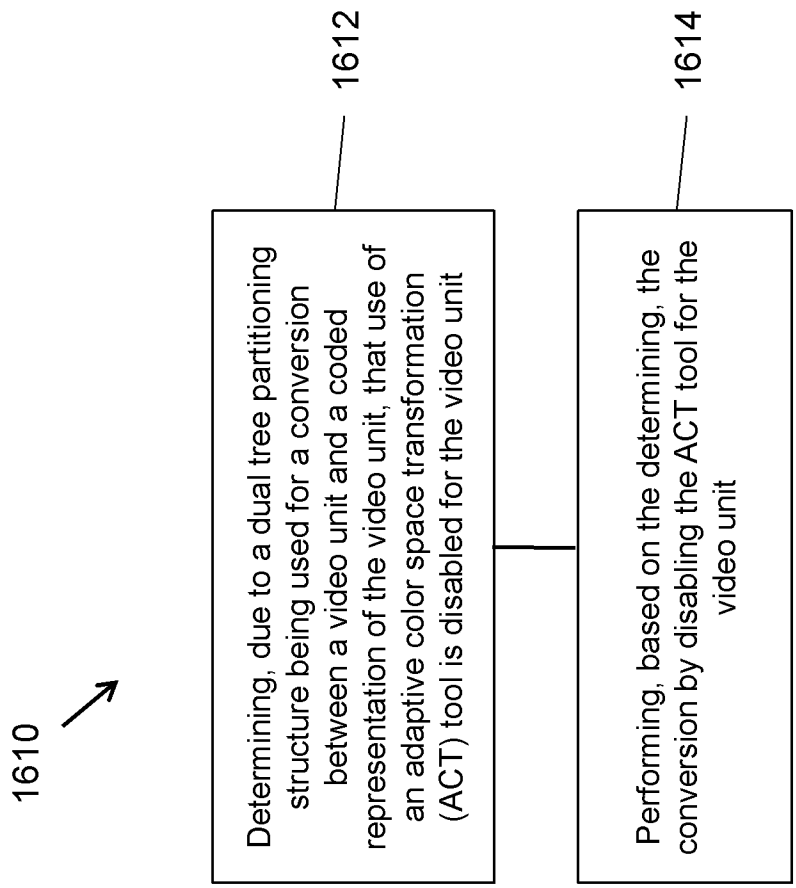

1610
1612
Determining, due to a dual tree partitioning structure being used for a conversion between a video unit and a coded representation of the video unit, that use of an adaptive color space transformation (ACT) tool is disabled for the video unit
1614
Performing, based on the determining, the conversion by disabling the ACT tool for the video unit

FIG. 16A

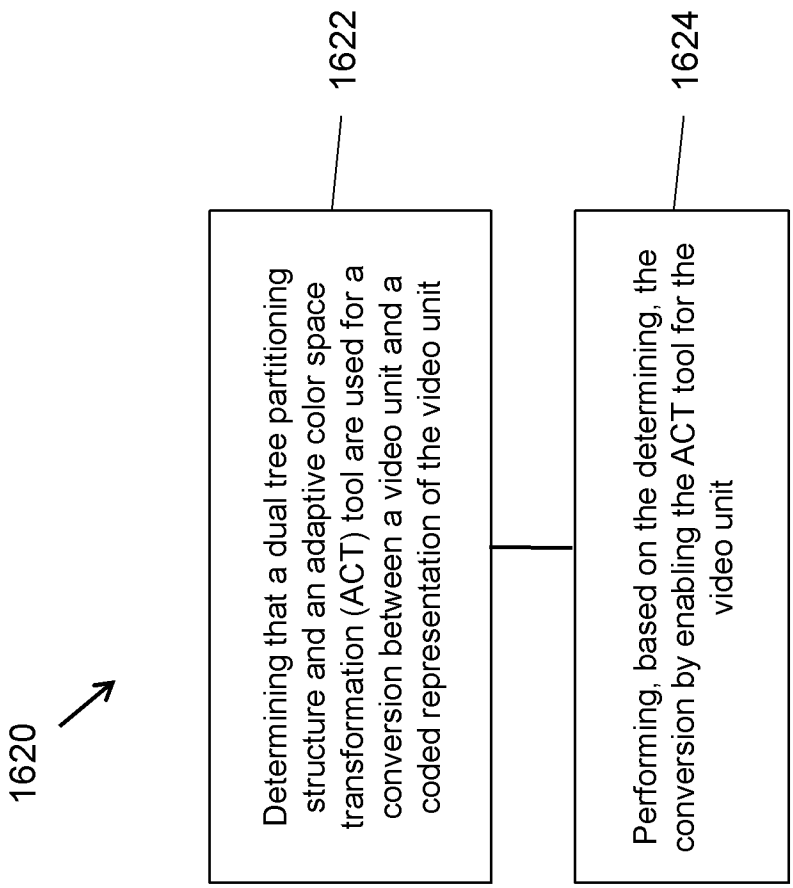

1620
1622
Determining that a dual tree partitioning structure and an adaptive color space transformation (ACT) tool are used for a conversion between a video unit and a coded representation of the video unit
1624
Performing, based on the determining, the conversion by enabling the ACT tool for the video unit

FIG. 16B

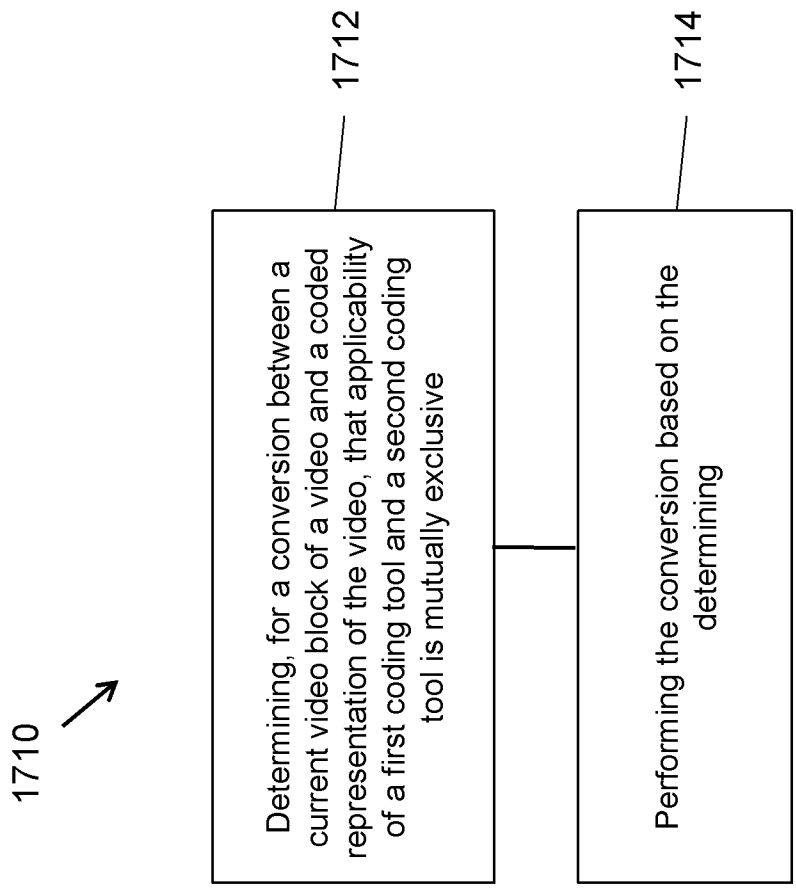

1710
1712
Determining, for a conversion between a current video block of a video and a coded representation of the video, that applicability of a first coding tool and a second coding tool is mutually exclusive
1714
Performing the conversion based on the determining

FIG. 17A

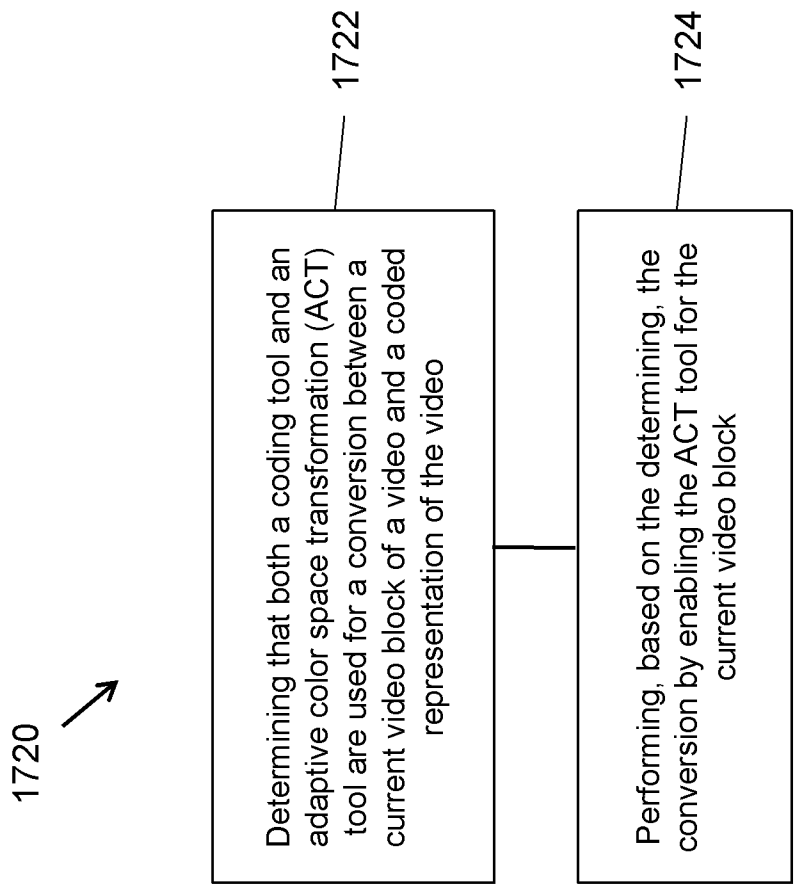

1720
1722
Determining that both a coding tool and an adaptive color space transformation (ACT) tool are used for a conversion between a current video block of a video and a coded representation of the video
1724
Performing, based on the determining, the conversion by enabling the ACT tool for the current video block

FIG. 17B

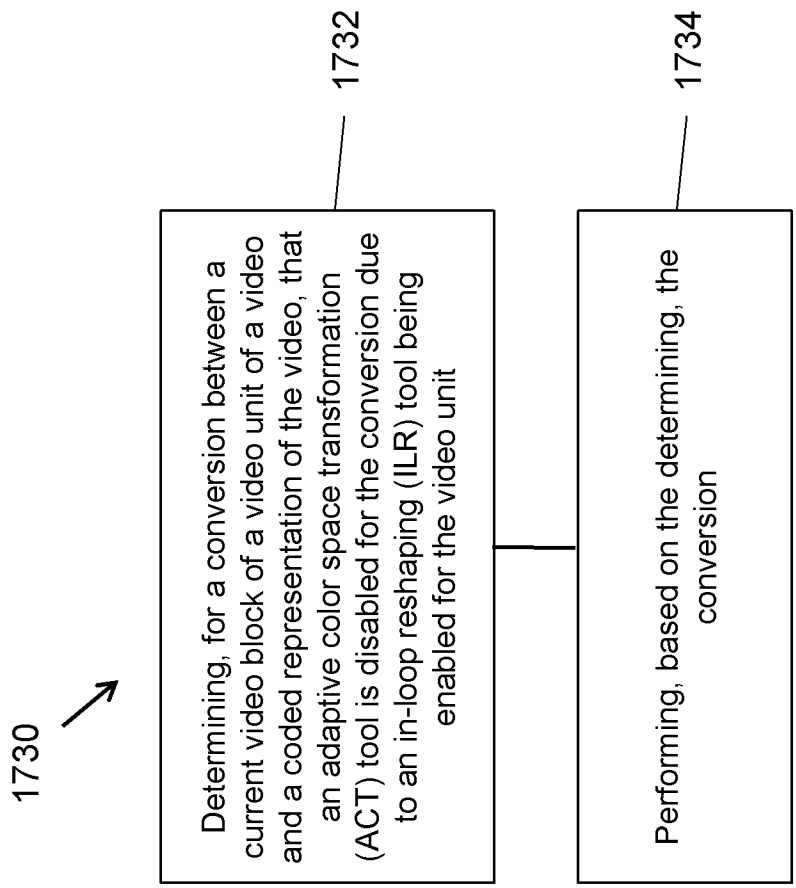

1730
Determining, for a conversion between a current video block of a video unit of a video and a coded representation of the video, that an adaptive color space transformation (ACT) tool is disabled for the conversion due to an in-loop reshaping (ILR) tool being enabled for the video unit
1732
Performing, based on the determining, the conversion
1734

FIG. 17C

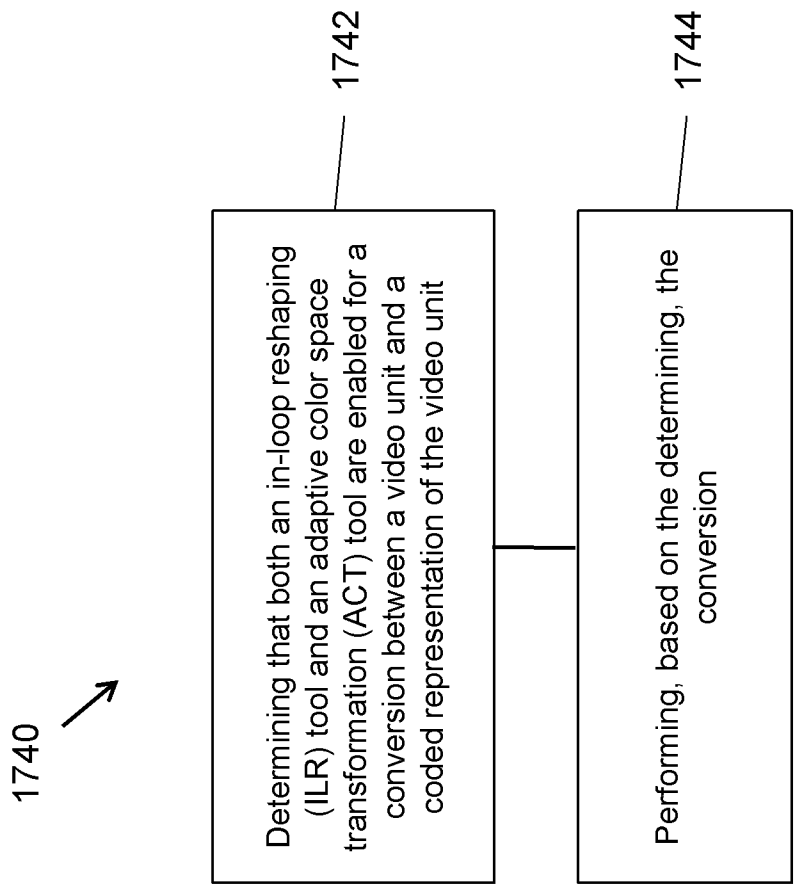

1740
1742
Determining that both an in-loop reshaping (ILR) tool and an adaptive color space transformation (ACT) tool are enabled for a conversion between a video unit and a coded representation of the video unit
1744
Performing, based on the determining, the conversion

FIG. 17D

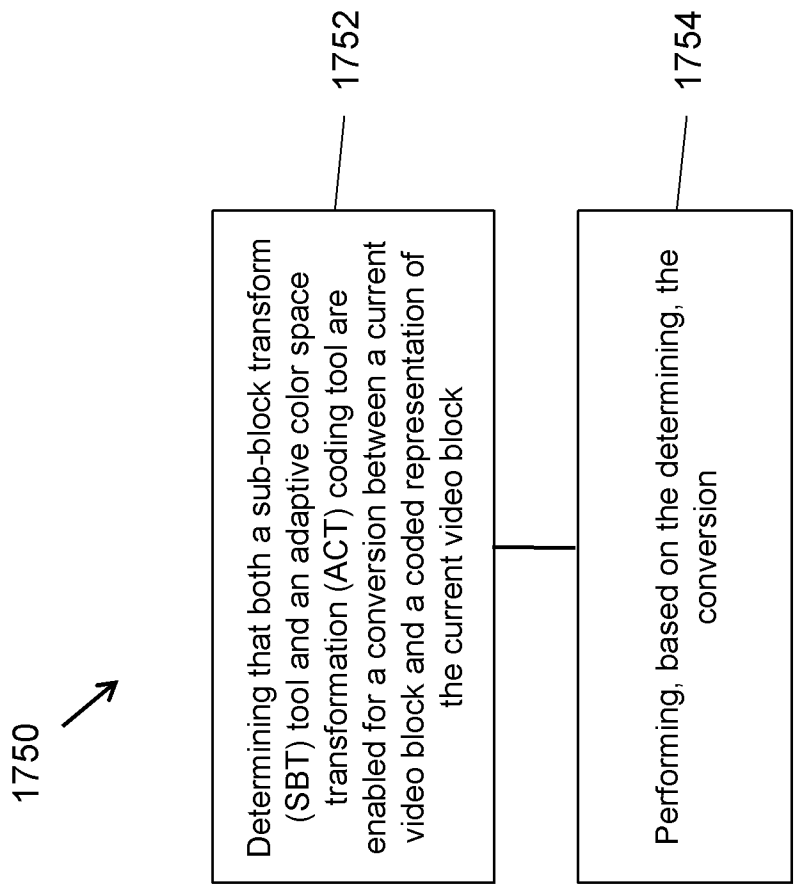

1750
1752
Determining that both a sub-block transform (SBT) tool and an adaptive color space transformation (ACT) coding tool are enabled for a conversion between a current video block and a coded representation of the current video block
1754
Performing, based on the determining, the conversion

FIG. 17E

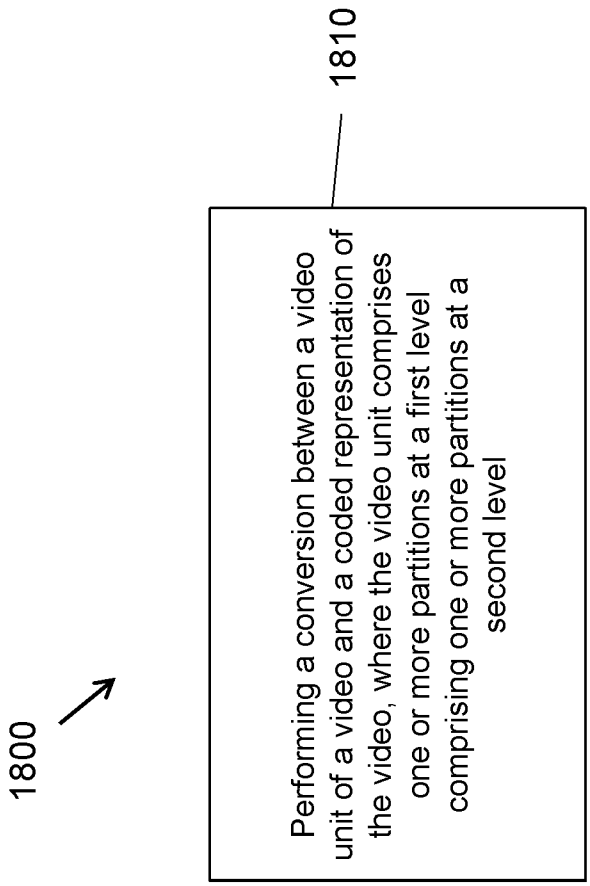

1800
1810
Performing a conversion between a video unit of a video and a coded representation of the video, where the video unit comprises one or more partitions at a first level comprising one or more partitions at a second level

FIG. 18

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/097368

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/96(2014.01)i; H04N 19/00(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

CNPAT, CNKI, WPI, EPODOC, IEEE: dual, separate, tree, partition, luma, chroma, colo?r, transform+, conversion, disable, off, ACT, enable

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ZHAO, Xin et al. "An implementation of adaptive color transform in VVC" *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14th Meeting: Geneva, CH, 19-27 March 2019, JVET-N0368,* 27 March 2019 (2019-03-27), abstract, sections 1, 2 and figure 1	1-7,15-18
PX	XIU, Xiaoyu et al. "Support of adaptive color transform for 444 video coding in VVC" *Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/ SC 29/WG 11, 16th Meeting: Geneva, CH, 1-11 October 2019, JVET-P0517_r1,* 11 October 2019 (2019-10-11), abstract, sections 2, 3	1-18
PX	WO 2020017513 A1 (SHARP KK.) 23 January 2020 (2020-01-23) description, paragraph [0061]	8-18
A	CN 109565599 A (QUALCOMM INC.) 02 April 2019 (2019-04-02) the whole document	1-18
A	WO 2019069950 A1 (SHARP KK.) 11 April 2019 (2019-04-11) the whole document	1-18

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 September 2020	**24 September 2020**
Name and mailing address of the ISA/CN **National Intellectual Property Administration, PRC** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China** Facsimile No. **(86-10)62019451**	Authorized officer **LONG,Xuanyao** Telephone No. **86-(10)-53961698**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/097368

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
WO 2020017513 A1	23 January 2020	None	
CN 109565599 A	02 April 2019	US 2018048889 A1	15 February 2018
		BR 112019002856 A2	14 May 2019
		EP 3497930 A1	19 June 2019
		JP 2019528025 A	03 October 2019
		KR 20190033559 A	29 March 2019
		WO 2018035127 A1	22 February 2018
		SG 11201900109S A	27 February 2019
		TW 201818720 A	16 May 2018
		US 10326986 B2	18 June 2019
WO 2019069950 A1	11 April 2019	None	