



(51) International Patent Classification:

H04N 19/176 (2014.01) H04N 19/60 (2014.01)

H04N 19/103 (2014.01) H04N 19/184 (2014.01)

(21) International Application Number:

PCT/CN2020/094907

(22) International Filing Date:

08 June 2020 (08.06.2020)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

PCT/CN2019/090261

06 June 2019 (06.06.2019) CN

PCT/CN2019/091435

15 June 2019 (15.06.2019) 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, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **LIU, Hongbin**; Jinritoutiao Post Office, China Satellite Communications Tower No. 63, Zhichun Road, Haidian District, Beijing 100080 (CN). **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, 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))

(54) Title: APPLICABILITY OF IMPLICIT TRANSFORM SELECTION

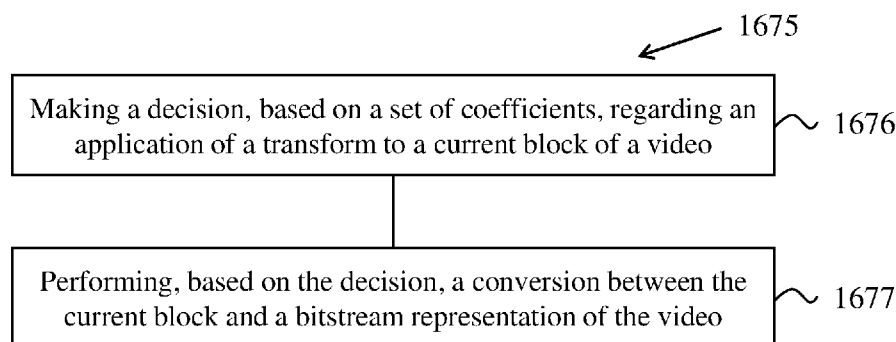


FIG. 16A

(57) Abstract: Devices, systems and methods for digital video coding, which includes using multiple transforms, are described. In a representative aspect, a method for video processing includes making a decision, for a current block of a video, regarding an application of a transform to the current block based on an indication in a bitstream representation of the video at a coding unit (CU) level, a coding block (CB) level, or a transform unit (TU) level, and performing, based on the decision, a conversion between the current block and the bitstream representation.

Published:

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

APPLICABILITY OF IMPLICIT TRANSFORM SELECTION

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 benefits of International Patent Application No. PCT/CN2019/090261 filed on June 6, 2019, and International Patent Application No. PCT/CN2019/091435 filed on June 15, 2019. For all purposes under the law, the entire disclosures of the aforementioned applications are incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[0002] This patent document relates to video coding techniques, devices and systems.

BACKGROUND

[0003] In spite of the advances in video compression, digital video still accounts for the largest bandwidth use on the internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, it is expected that the bandwidth demand for digital video usage will continue to grow.

SUMMARY

[0004] Devices, systems and methods related to digital video coding, and specifically, methods on multiple transforms for video coding, are described. The described methods may be applied to both the existing video coding standards (e.g., High Efficiency Video Coding (HEVC)) and future video coding standards (e.g., Versatile Video Coding (VVC)) or codecs.

[0005] In yet another representative aspect, the disclosed technology may be used to provide an example method for video processing. This method includes making a decision, based on a set of coefficients, regarding an application of a transform to a current block of a video, and performing, based on the decision, a conversion between the current block and a bitstream representation of the video.

[0006] In yet another representative aspect, the disclosed technology may be used to provide an example method for video processing. This method includes making a decision, based on

coding information associated with a current block of a video, regarding an application of a transform to the current block, and performing, based on the decision, a conversion between the current block and a bitstream representation of the video.

[0007] In yet another representative aspect, the disclosed technology may be used to provide an example method for video processing. This method includes making a decision, for a current block of a video, regarding an application of a transform to the current block based on an indication in a bitstream representation of the video at a coding unit (CU) level, a coding block (CB) level, or a transform unit (TU) level, and performing, based on the decision, a conversion between the current block and the bitstream representation.

[0008] In yet another representative aspect, the above-described method is embodied in the form of processor-executable code and stored in a computer-readable program medium.

[0009] In yet another representative aspect, a device that is configured or operable to perform the above-described method is disclosed. The device may include a processor that is programmed to implement this method.

[0010] In yet another representative aspect, a video decoder apparatus may implement a method as described herein.

[0011] 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

[0012] FIG. 1 shows an example of an encoder block diagram.

[0013] FIG. 2 shows an example of 67 intra prediction modes.

[0014] FIGS. 3A and 3B show examples of reference samples for wide-angle intra prediction modes for non-square blocks.

[0015] FIG. 4 shows an example of a discontinuity when using wide-angle intra prediction.

[0016] FIGS. 5A–5D show examples of samples used by a position-dependent intra prediction combination (PDPC) method.

[0017] FIG. 6 shows an example of divisions of 4×8 and 8×4 blocks.

[0018] FIG. 7 shows an example of divisions all blocks except 4×8, 8×4 and 4×4.

[0019] FIG. 8 shows an example of dividing a block of 4×8 samples into two independently decodable areas.

- [0020] FIG. 9 shows an example of the order of processing of the rows of pixels to maximize throughput for 4xN blocks with vertical predictor.
- [0021] FIG. 10 shows an example of a secondary transform in JEM.
- [0022] FIG. 11 shows an example of the proposed reduced secondary transform (RST).
- [0023] FIG. 12 shows examples of the forward and inverse reduced transforms.
- [0024] FIG. 13 shows an example of a forward RST 8x8 process with a 16x48 matrix.
- [0025] FIG. 14 shows an example of scanning positions 17 through 64 in an 8x8 block for a non-zero element.
- [0026] FIG. 15 shows an example of sub-block transform modes SBT-V and SBT-H.
- [0027] FIGS. 16A–16C show flowcharts of example methods for video processing.
- [0028] FIG. 17 is a block diagram of an example of a hardware platform for implementing a visual media decoding or a visual media encoding technique described in the present document.
- [0029] FIG. 18 is a block diagram of an example video processing system in which disclosed techniques may be implemented.

DETAILED DESCRIPTION

1 Introduction

- [0030] Due to the increasing demand of higher resolution video, video coding methods and techniques are ubiquitous in modern technology. Video codecs typically include an electronic circuit or software that compresses or decompresses digital video, and are continually being improved to provide higher coding efficiency. A video codec converts uncompressed video to a compressed format or vice versa. There are complex relationships between the video quality, the amount of data used to represent the video (determined by the bit rate), the complexity of the encoding and decoding algorithms, sensitivity to data losses and errors, ease of editing, random access, and end-to-end delay (latency). The compressed format usually conforms to a standard video compression specification, e.g., the High Efficiency Video Coding (HEVC) standard (also known as H.265 or MPEG-H Part 2), the Versatile Video Coding (VVC) standard to be finalized, or other current and/or future video coding standards.
- [0031] Embodiments of the disclosed technology may be applied to existing video coding standards (e.g., HEVC, H.265) and future standards to improve runtime performance. Section headings are used in the present document to improve readability of the description and do not in

any way limit the discussion or the embodiments (and/or implementations) to the respective sections only.

2 Embodiments and examples for methods on multiple transforms

2.1 Color space and chroma subsampling

[0032] Color space, also known as the color model (or color system), is an abstract mathematical model which simply describes the range of colors as tuples of numbers, typically as 3 or 4 values or color components (e.g. RGB). Basically speaking, color space is an elaboration of the coordinate system and sub-space.

[0033] For video compression, the most frequently used color spaces are YCbCr and RGB.

[0034] YCbCr, Y'CbCr, or Y Pb/Cb Pr/Cr, also written as YCBCR or Y'CBCR, is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y' is the luma component and CB and CR are the blue-difference and red-difference chroma components. Y' (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries.

[0035] Chroma subsampling is the practice of encoding images by implementing less resolution for chroma information than for luma information, taking advantage of the human visual system's lower acuity for color differences than for luminance.

[0036] 4:4:4 Format. Each of the three Y'CbCr components have the same sample rate, thus there is no chroma subsampling. This scheme is sometimes used in high-end film scanners and cinematic post production.

[0037] 4:2:2 Format. The two chroma components are sampled at half the sample rate of luma: the horizontal chroma resolution is halved. This reduces the bandwidth of an uncompressed video signal by one-third with little to no visual difference

[0038] 4:2:0 Format. In 4:2:0, the horizontal sampling is doubled compared to 4:1:1, but as the Cb and Cr channels are only sampled on each alternate line in this scheme, the vertical resolution is halved. The data rate is thus the same. Cb and Cr are each subsampled at a factor of 2 both horizontally and vertically. There are three variants of 4:2:0 schemes, having different horizontal and vertical siting.

[0039] ○ In MPEG-2, Cb and Cr are cosited horizontally. Cb and Cr are sited between pixels in the vertical direction (sited interstitially).

[0040] ○ In JPEG/JFIF, H.261, and MPEG-1, Cb and Cr are sited interstitially, halfway

between alternate luma samples.

[0041] ○ In 4:2:0 DV, Cb and Cr are co-sited in the horizontal direction. In the vertical direction, they are co-sited on alternating lines.

2.2 Coding flow of a typical video codec

[0042] FIG. 1 shows an example of encoder block diagram of VVC, which contains three in-loop filtering blocks: deblocking filter (DF), sample adaptive offset (SAO) and ALF. Unlike DF, which uses predefined filters, SAO and ALF utilize the original samples of the current picture to reduce the mean square errors between the original samples and the reconstructed samples by adding an offset and by applying a finite impulse response (FIR) filter, respectively, with coded side information signaling the offsets and filter coefficients. ALF is located at the last processing stage of each picture and can be regarded as a tool trying to catch and fix artifacts created by the previous stages.

2.3 Intra mode coding with 67 intra prediction modes

[0043] To capture the arbitrary edge directions presented in natural video, the number of directional intra modes is extended from 33, as used in HEVC, to 65. The additional directional modes are depicted as red dotted arrows in FIG. 2, and the planar and DC modes remain the same. These denser directional intra prediction modes apply for all block sizes and for both luma and chroma intra predictions.

[0044] Conventional angular intra prediction directions are defined from 45 degrees to -135 degrees in clockwise direction as shown in FIG. 1. In VTM2, several conventional angular intra prediction modes are adaptively replaced with wide-angle intra prediction modes for the non-square blocks. The replaced modes are signaled using the original method and remapped to the indexes of wide angular modes after parsing. The total number of intra prediction modes is unchanged (e.g., 67), and the intra mode coding is unchanged.

[0045] In HEVC, every intra-coded block has a square shape and the length of each of its side is a power of 2. Thus, no division operations are required to generate an intra-predictor using DC mode. In VTM2, blocks can have a rectangular shape that necessitates the use of a division operation per block in the general case. To avoid division operations for DC prediction, only the longer side is used to compute the average for non-square blocks.

2.4 Wide-angle intra prediction for non-square blocks

[0046] In some embodiments, conventional angular intra prediction directions are defined

from 45 degrees to -135 degrees in clockwise direction. In VTM2, several conventional angular intra prediction modes are adaptively replaced with wide-angle intra prediction modes for non-square blocks. The replaced modes are signaled using the original method and remapped to the indexes of wide angular modes after parsing. The total number of intra prediction modes for a certain block is unchanged, e.g., 67, and the intra mode coding is unchanged.

[0047] To support these prediction directions, the top reference with length $2W+1$, and the left reference with length $2H+1$, are defined as shown in the examples in FIGS. 3A and 3B.

[0048] In some embodiments, the mode number of replaced mode in wide-angular direction mode is dependent on the aspect ratio of a block. The replaced intra prediction modes are illustrated in Table 1.

Table 1: Intra prediction modes replaced by wide-angle modes

<i>Condition</i>	<i>Replaced intra prediction modes</i>
$W / H == 2$	Modes 2,3,4,5,6,7
$W / H > 2$	Modes 2,3,4,5,6,7,8,9,10,11
$W / H == 1$	None
$H / W == 1/2$	Modes 61,62,63,64,65,66
$H / W < 1/2$	Mode 57,58,59,60,61,62,63,64,65,66

[0049] As shown in FIG. 4, two vertically-adjacent predicted samples may use two non-adjacent reference samples in the case of wide-angle intra prediction. Hence, low-pass reference samples filter and side smoothing are applied to the wide-angle prediction to reduce the negative effect of the increased gap Δp_a .

2.5 Examples of position dependent intra prediction combination (PDPC)

[0050] In the VTM2, the results of intra prediction of planar mode are further modified by a position dependent intra prediction combination (PDPC) method. PDPC is an intra prediction method which invokes a combination of the un-filtered boundary reference samples and HEVC style intra prediction with filtered boundary reference samples. PDPC is applied to the following intra modes without signaling: planar, DC, horizontal, vertical, bottom-left angular mode and its *eight* adjacent angular modes, and top-right angular mode and its *eight* adjacent angular modes.

[0051] The prediction sample $pred(x,y)$ is predicted using an intra prediction mode (DC, planar, angular) and a linear combination of reference samples according to the Equation as follows:

[0052] $pred(x,y)=(wL \times R_{-1,y} + wT \times R_{x,-1} - wTL \times R_{-1,-1} + (64 - wL - wT + wTL) \times pred(x,y) + 32) \gg shift$

[0053] Herein, $R_{x,-1}$, $R_{-1,y}$ represent the reference samples located at the top and left of current sample (x, y) , respectively, and $R_{-1,-1}$ represents the reference sample located at the top-left corner of the current block.

[0054] In some embodiments, and if PDPC is applied to DC, planar, horizontal, and vertical intra modes, additional boundary filters are not needed, as required in the case of HEVC DC mode boundary filter or horizontal/vertical mode edge filters.

[0055] FIGS. 5A–5D illustrate the definition of reference samples ($R_{x,-1}$, $R_{-1,y}$ and $R_{-1,-1}$) for PDPC applied over various prediction modes. The prediction sample $pred(x', y')$ is located at (x', y') within the prediction block. The coordinate x of the reference sample $R_{x,-1}$ is given by: $x = x' + y' + 1$, and the coordinate y of the reference sample $R_{-1,y}$ is similarly given by: $y = x' + y' + 1$.

[0056] In some embodiments, the PDPC weights are dependent on prediction modes and are shown in Table 2, where $S = shift$.

Table 2: Examples of PDPC weights according to prediction modes

Prediction modes	wT	wL	wTL
Diagonal top-right	$16 \gg ((y' < 1) \gg S)$	$16 \gg ((x' < 1) \gg S)$	0
Diagonal bottom-left	$16 \gg ((y' < 1) \gg S)$	$16 \gg ((x' < 1) \gg S)$	0
Adjacent diag. top-right	$32 \gg ((y' < 1) \gg S)$	0	0
Adjacent diag. bottom-left	0	$32 \gg ((x' < 1) \gg S)$	0

2.6 Intra subblock partitioning

[0057] 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 3. FIG. 6 and FIG. 7 show examples of the two possibilities. All sub-partitions fulfill the condition of having at least 16 samples.

Table 3: Number of sub-partitions depending on the block size

Block Size	Number of Sub-Partitions
4×4	Not divided
4×8 and 8×4	2
All other cases	4

[0058] 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.

[0059] Based on the intra mode and the split utilized, two different classes of processing orders are used, which are referred to as normal and reversed order. In the normal order, the first sub-partition to be processed is the one containing the top-left sample of the CU and then continuing downwards (horizontal split) or rightwards (vertical split). As a result, reference samples used to generate the sub-partitions prediction signals are only located at the left and above sides of the lines. On the other hand, the reverse processing order either starts with the sub-partition containing the bottom-left sample of the CU and continues upwards or starts with sub-partition containing the top-right sample of the CU and continues leftwards.

2.7 Block differential pulse-code modulation coding (BDPCM)

[0060] BDPCM is proposed in JVET-M0057. Due to the shape of the horizontal (resp. vertical) predictors, which use the left (A) (resp. top (B)) pixel for prediction of the current pixel, the most throughput-efficient way of processing the block is to process all the pixels of one column (resp. line) in parallel, and to process these columns (resp. lines) sequentially. In order to increase throughput, we introduce the following process: a block of width 4 is divided into two halves with a horizontal frontier when the predictor chosen on this block is vertical, and a block of height 4 is divided into two halves with a vertical frontier when the predictor chosen on this block is horizontal.

[0061] When a block is divided, samples from one area are not allowed to use pixels from another area to compute the prediction: if this situation occurs, the prediction pixel is replaced by the reference pixel in the prediction direction. This is shown on FIG. 8 for different positions of current pixel X in a 4x8 block predicted vertically.

[0062] Because of this property, it becomes now possible to process a 4x4 block in 2 cycles, and a 4x8 or 8x4 block in 4 cycles, and so on, as shown on FIG. 9.

[0063] Table 4 summarizes the number of cycles required to process the block, depending on the block size. It is trivial to show that any block which has both dimensions larger or equal to 8 can be processed in 8 pixels per cycle or more.

Table 4: Throughput for blocks of size 4xN, Nx4

Block size	4x4	4x8, 8x4	4x16, 16x4	4x32, 32x4
Cycles	2	4	8	16
Pixels	16	32	64	128
Throughput (pixels/cycle)	8	8	8	8

2.8 Quantized residual domain BDPCM

[0064] In JVET-N0413, quantized residual domain BDPCM (denote as RBDPCM hereinafter) is proposed. 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.

[0065] For a block of size M (rows) $\times$ N (cols), let $r_{i,j}$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$, 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 $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ 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 $M \times N$ array $\tilde{R}$ with elements $\tilde{r}_{i,j}$. When vertical BDPCM is signaled:

$$[0066] \quad \tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0, \quad 0 \leq j \leq (N-1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \end{cases}$$

[0067] For horizontal prediction, similar rules apply, and the residual quantized samples are obtained by

$$[0068] \quad \tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M-1), \quad j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \leq i \leq (M-1), \quad 1 \leq j \leq (N-1) \end{cases}$$

[0069] The residual quantized samples $\tilde{r}_{i,j}$ are sent to the decoder.

[0070] On the decoder side, the above calculations are reversed to produce $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$. For vertical prediction case,

$$[0071] \quad Q(r_{i,j}) = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1)$$

[0072] For horizontal case,

[0073] $Q(r_{i,j}) = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1)$

[0074] The invert quantized residuals, $Q^{-1}(Q(r_{i,j}))$, are added to the intra block prediction values to produce the reconstructed sample values.

[0075] One of the benefits of this scheme is that inverting the 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.

[0076] Transform skip is always used in quantized residual domain BDPCM.

2.9 Multiple Transform Set (MTS) in VVC

[0077] In VTM4, large block-size transforms, up to 64×64 in size, are enabled, which is primarily useful for higher resolution video, e.g., 1080p and 4K sequences. High frequency transform coefficients are zeroed out for the transform blocks with size (width or height, or both width and height) equal to 64, so that only the lower-frequency coefficients are retained. For example, for an M×N transform block, with M as the block width and N as the block height, when M is equal to 64, only the left 32 columns of transform coefficients are kept. Similarly, when N is equal to 64, only the top 32 rows of transform coefficients are kept. When transform skip mode is used for a large block, the entire block is used without zeroing out any values.

[0078] In addition to DCT-II which has been employed in HEVC, a Multiple Transform Selection (MTS) scheme is used for residual coding both inter and intra coded blocks. It uses multiple selected transforms from the DCT8/DST7. The newly introduced transform matrices are DST-VII and DCT-VIII. The Table 4 below shows the basis functions of the selected DST/DCT.

Table 4: Basis functions of transform matrices used in VVC

Transform Type	Basis function $T_i(j), i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

[0079] In order to keep the orthogonality of the transform matrix, the transform matrices are quantized more accurately than the transform matrices in HEVC. To keep the intermediate values of the transformed coefficients within the 16-bit range, after horizontal and after vertical transform, all the coefficients are to have 10-bit.

[0080] In order to control MTS scheme, separate enabling flags are specified at SPS level for intra and inter, respectively. When MTS is enabled at SPS, a CU level flag is signalled to indicate whether MTS is applied or not. Here, MTS is applied only for luma. The MTS CU level flag is signalled when the following conditions are satisfied.

[0081] ○ Both width and height smaller than or equal to 32

[0082] ○ CBF flag is equal to one

[0083] If MTS CU flag is equal to zero, then DCT2 is applied in both directions. However, if MTS CU flag is equal to one, then two other flags are additionally signalled to indicate the transform type for the horizontal and vertical directions, respectively. Transform and signalling mapping table as shown in Table 5. When it comes to transform matrix precision, 8-bit primary transform cores are used. Therefore, all the transform cores used in HEVC are kept as the same, including 4-point DCT-2 and DST-7, 8-point, 16-point and 32-point DCT-2. Also, other transform cores including 64-point DCT-2, 4-point DCT-8, 8-point, 16-point, 32-point DST-7 and DCT-8, use 8-bit primary transform cores.

MTS_CU_flag	MTS_Hor_flag	MTS_Ver_flag	Intra/inter	
			Horizontal	Vertical
0			DCT2	
1	0	0	DST7	DST7
	0	1	DCT8	DST7
	1	0	DST7	DCT8
	1	1	DCT8	DCT8

[0084] To reduce the complexity of large size DST-7 and DCT-8, High frequency transform coefficients are zeroed out for the DST-7 and DCT-8 blocks with size (width or height, or both width and height) equal to 32. Only the coefficients within the 16x16 lower-frequency region are retained.

[0085] As in HEVC, the residual of a block can be coded with transform skip mode. To

avoid the redundancy of syntax coding, the transform skip flag is not signalled when the CU level MTS_CU_flag is not equal to zero. The block size limitation for transform skip is the same to that for MTS in JEM4, which indicate that transform skip is applicable for a CU when both block width and height are equal to or less than 32.

[0086] Implicit MTS is a recent tool in VVC. The variable implicitMtsEnabled is derived as follows:

- If sps_mts_enabled_flag is equal to 1 and one of the following conditions is true, implicitMtsEnabled is set equal to 1:
 - IntraSubPartitionsSplitType is not equal to ISP_NO_SPLIT
 - cu_sbt_flag is equal to 1 and $\text{Max}(nTbW, nTbH)$ is less than or equal to 32
 - sps_explicit_mts_intra_enabled_flag is equal to 0 and $\text{CuPredMode}[0][xTbY][yTbY]$ is equal to MODE_INTRA and $\text{lnst_idx}[x0][y0]$ is equal to 0 and $\text{intra_mip_flag}[x0][y0]$ is equal to 0
- Otherwise, implicitMtsEnabled is set equal to 0.

[0087] In VVC Draft v6, if the current coding block is luma component and implicitMtsEnabled is equal to 1, the implicit MTS transform selection is decided by the following formula:

$$\text{trTypeHor} = (nTbW \geq 4 \ \&\& \ nTbW \leq 16) ? \text{DST7} : \text{DCT2} \quad (8-965)$$

$$\text{trTypeVer} = (nTbH \geq 4 \ \&\& \ nTbH \leq 16) ? \text{DST7} : \text{DCT2} \quad (8-966)$$

[0088] where trTypeHor and trTypeVer specify the horizontal and vertical transform kernels, and nTbW and nTbH specify the width and height of the transform block. The two formulas indicate that the implicit MTS transform is selected from the following pairs: {DST7, DST7}, {DST7, DCT2}, {DCT2, DST7}, {DCT2, DCT2}

2.10 Reduced Secondary Transform (RST) proposed in JVET-N0193

2.10.1 Non-Separable Secondary Transform (NSST) in JEM

[0089] In JEM, secondary transform is applied between forward primary transform and quantization (at encoder) and between de-quantization and invert primary transform (at decoder side). As shown in FIG. 10, 4x4 (or 8x8) secondary transform is performed depends on block size. For example, 4x4 secondary transform is applied for small blocks (i.e., $\min(\text{width}, \text{height}) < 8$) and 8x8 secondary transform is applied for larger blocks (i.e., $\min(\text{width}, \text{height}) > 4$) per

8x8 block.

[0090] Application of a non-separable transform is described as follows using input as an example. To apply the non-separable transform, the 4x4 input block X

$$[0091] \quad X = \begin{bmatrix} X_{00} & X_{01} & X_{02} & X_{03} \\ X_{10} & X_{11} & X_{12} & X_{13} \\ X_{20} & X_{21} & X_{22} & X_{23} \\ X_{30} & X_{31} & X_{32} & X_{33} \end{bmatrix}$$

[0092] is first represented as a vector $\vec{X}$:

$$[0093] \quad \vec{X} =$$

$$[X_{00} \ X_{01} \ X_{02} \ X_{03} \ X_{10} \ X_{11} \ X_{12} \ X_{13} \ X_{20} \ X_{21} \ X_{22} \ X_{23} \ X_{30} \ X_{31} \ X_{32} \ X_{33}]^T$$

[0094] The non-separable transform is calculated as $\vec{F} = T \cdot \vec{X}$, where $\vec{F}$ indicates the transform coefficient vector, and T is a 16x16 transform matrix. The 16x1 coefficient vector $\vec{F}$ is subsequently re-organized as 4x4 block using the scanning order for that block (horizontal, vertical or diagonal). The coefficients with smaller index will be placed with the smaller scanning index in the 4x4 coefficient block. There are totally 35 transform sets and 3 non-separable transform matrices (kernels) per transform set are used. The mapping from the intra prediction mode to the transform set is pre-defined. For each transform set, the selected non-separable secondary transform (NSST) candidate is further specified by the explicitly signalled secondary transform index. The index is signalled in a bit-stream once per Intra CU after transform coefficients.

2.10.2 Reduced Secondary Transform (RST) in JVET-N0193

[0095] The RST (a.k.a. Low Frequency Non-Separable Transform (LFNST)) was introduced in JVET-K0099 and 4 transform set (instead of 35 transform sets) mapping introduced in JVET-L0133. In this JVET-N0193, 16x64 (further reduced to 16x48) and 16x16 matrices are employed. For notational convenience, the 16x64 (reduced to 16x48) transform is denoted as RST8x8 and the 16x16 one as RST4x4. FIG. 11 shows an example of RST.

RST computation

[0096] The main idea of a Reduced Transform (RT) is to map an N dimensional vector to an R dimensional vector in a different space, where R/N ($R < N$) is the reduction factor.

[0097] The RT matrix is an $R \times N$ matrix as follows:

$$T_{R \times N} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & t_{1N} \\ t_{21} & t_{22} & t_{23} & & t_{2N} \\ & \vdots & & \ddots & \vdots \\ t_{R1} & t_{R2} & t_{R3} & \dots & t_{RN} \end{bmatrix}$$

[0098] where the **R** rows of the transform are **R** bases of the **N** dimensional space. The invert transform matrix for RT is the transpose of its forward transform. The forward and invert RT are depicted in FIG. 12.

[0099] In this contribution, the RST8x8 with a reduction factor of 4 (1/4 size) is applied. Hence, instead of 64x64, which is conventional 8x8 non-separable transform matrix size, 16x64 direct matrix is used. In other words, the 64x16 invert RST matrix is used at the decoder side to generate core (primary) transform coefficients in 8x8 top-left regions. The forward RST8x8 uses 16x64 (or 8x64 for 8x8 block) matrices so that it produces non-zero coefficients only in the top-left 4x4 region within the given 8x8 region. In other words, if RST is applied then the 8x8 region except the top-left 4x4 region will have only zero coefficients. For RST4x4, 16x16 (or 8x16 for 4x4 block) direct matrix multiplication is applied.

[00100] An invert RST is conditionally applied when the following two conditions are satisfied:

[00101] ○ Block size is greater than or equal to the given threshold ($W \geq 4 \ \&\& \ H \geq 4$)

[00102] ○ Transform skip mode flag is equal to zero

[00103] If both width (W) and height (H) of a transform coefficient block is greater than 4, then the RST8x8 is applied to the top-left 8x8 region of the transform coefficient block.

Otherwise, the RST4x4 is applied on the top-left $\min(8, W) \times \min(8, H)$ region of the transform coefficient block.

[00104] If RST index is equal to 0, RST is not applied. Otherwise, RST is applied, of which kernel is chosen with the RST index. The RST selection method and coding of the RST index are explained later.

[00105] Furthermore, RST is applied for intra CU in both intra and inter slices, and for both Luma and Chroma. If a dual tree is enabled, RST indices for Luma and Chroma are signaled separately. For inter slice (the dual tree is disabled), a single RST index is signaled and used for both Luma and Chroma.

[00106] In 13th JVET meeting, Intra Sub-Partitions (ISP), as a new intra prediction mode, was adopted. When ISP mode is selected, RST is disabled and RST index is not signaled, because performance improvement was marginal even if RST is applied to every feasible partition block. Furthermore, disabling RST for ISP-predicted residual could reduce encoding complexity.

RST selection

[00107] A RST matrix is chosen from four transform sets, each of which consists of two transforms. Which transform set is applied is determined from intra prediction mode as the following:

[00108] (1) If one of three CCLM modes is indicated, transform set 0 is selected.

[00109] (2) Otherwise, transform set selection is performed according to the following table:

The transform set selection table

IntraPredMode	Tr. set index
IntraPredMode < 0	1
0 <= IntraPredMode <= 1	0
2 <= IntraPredMode <= 12	1
13 <= IntraPredMode <= 23	2
24 <= IntraPredMode <= 44	3
45 <= IntraPredMode <= 55	2
56 <= IntraPredMode	1

[00110] The index to access the above table, denoted as IntraPredMode, have a range of [-14, 83], which is a transformed mode index used for wide angle intra prediction.

RST matrices of reduced dimension

[00111] As a further simplification, 16x48 matrices are applied instead of 16x64 with the same transform set configuration, each of which takes 48 input data from three 4x4 blocks in a top-left 8x8 block excluding right-bottom 4x4 block (as shown in FIG. 13).

RST signaling

[00112] The forward RST8x8 with R =16 uses 16x64 matrices so that it produces non-zero coefficients only in the top-left 4x4 region within the given 8x8 region. In other words, if RST is applied then the 8x8 region except the top-left 4x4 region generates only zero coefficients. As a result, RST index is not coded when any non-zero element is detected within 8x8 block region

other than top-left 4×4 (which is depicted in FIG. 14) because it implies that RST was not applied. In such a case, RST index is inferred to be zero.

Zero-out range

[00113] Usually, before applying the invert RST on a 4×4 sub-block, any coefficient in the 4×4 sub-block may be non-zero. However, it is constrained that in some cases, some coefficients in the 4×4 sub-block must be zero before invert RST is applied on the sub-block.

[00114] Let nonZeroSize be a variable. It is required that any coefficient with the index no smaller than nonZeroSize when it is rearranged into a 1-D array before the invert RST must be zero.

[00115] When nonZeroSize is equal to 16, there is no zero-out constrain on the coefficients in the top-left 4×4 sub-block.

[00116] In JVET-N0193, when the current block size is 4×4 or 8×8, nonZeroSize is set equal to 8 (that is, coefficients with the scanning index in the range [8, 15] as show in FIG. 14, shall be 0). For other block dimensions, nonZeroSize is set equal to 16.

Description of RST in working Draft

Sequence parameter set RBSP syntax

seq_parameter_set_rbsp() {	Descriptor
.....	
sps_mts_enabled_flag	u(1)
if(sps_mts_enabled_flag) {	
sps_explicit_mts_intra_enabled_flag	u(1)
sps_explicit_mts_inter_enabled_flag	u(1)
}	
.....	
sps_st_enabled_flag	u(1)
.....	
}	

Residual coding syntax

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cIdx) {	Descriptor
...	
if(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) && (xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag[xC][yC]	ae(v)
remBinsPass1 --	

if(sig_coeff_flag[xC][yC])	
inferSbDcSigCoeffFlag = 0	
}	
if(sig_coeff_flag[xC][yC]) {	
if(!transform_skip_flag[x0][y0]) {	
numSigCoeff++	
if(((log2TbWidth == 2 && log2TbHeight == 2) (log2TbWidth == 3 && log2TbHeight == 3)) && n >= 8 && i == 0) ((log2TbWidth >= 3 && log2TbHeight >= 3 && (i == 1 i == 2)))) {	
numZeroOutSigCoeff++	
}	
}	
abs_level_gt1_flag[n]	ae(v)
...	

Coding unit syntax

coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
...	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0]) {	
if(cbWidth <= MaxSbtSize && cbHeight <= MaxSbtSize) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH allowSbtVerQ allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
numZeroOutSigCoeff = 0	

transform_tree(x0, y0, cbWidth, cbHeight, treeType)	
if(Min(cbWidth, cbHeight) >= 4 && sps_st_enabled_flag == 1 && CuPredMode[x0][y0] == MODE_INTRA && IntraSubPartitionsSplitType == ISP_NO_SPLIT) {	
if((numSigCoeff > ((treeType == SINGLE_TREE) ? 2 : 1)) && numZeroOutSigCoeff == 0) {	
st_idx[x0][y0]	ae(v)
}	
}	
}	
}	
}	

Sequence parameter set RBSP semantics

.....

sps_st_enabled_flag equal to 1 specifies that st_idx may be present in the residual coding syntax for intra coding units. sps_st_enabled_flag equal to 0 specifies that st_idx is not present in the residual coding syntax for intra coding units.

.....

Coding unit semantics

.....

st_idx[x0][y0] specifies which secondary transform kernel is applied between two candidate kernels in a selected transform set. st_idx[x0][y0] equal to 0 specifies that the secondary transform is not applied. The array indices x0, y0 specify the location (x0, y0) of the top-left sample of the considered transform block relative to the top-left sample of the picture. When st_idx[x0][y0] is not present, st_idx[x0][y0] is inferred to be equal to 0.

Transformation process for scaled transform coefficients

General

Inputs to this process are:

- a luma location (xTbY, yTbY) specifying the top-left sample of the current luma transform block relative to the top-left luma sample of the current picture,
- a variable nTbW specifying the width of the current transform block,
- a variable nTbH specifying the height of the current transform block,
- a variable cIdx specifying the colour component of the current block,
- an (nTbW)x(nTbH) array d[x][y] of scaled transform coefficients with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

Output of this process is the (nTbW)x(nTbH) array r[x][y] of residual samples with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$.

If st_idx[xTbY][yTbY] is not equal to 0, the following applies:

1. The variables nStSize, log2StSize, numStX, numStY, and nonZeroSize are derived as follows:

- If both nTbW and nTbH are greater than or equal to 8, log2StSize is set to 3 and nStOutSize is set to 48.
 - Otherwise, log2StSize is set to 2 and nStOutSize is set to 16.
 - nStSize is set to $(1 \ll \log2StSize)$.
 - If nTbH is equal to 4 and nTbW is greater than 8, numStX set equal to 2.
 - Otherwise, numStX set equal to 1.
 - If nTbW is equal to 4 and nTbH is greater than 8, numStY set equal to 2.
 - Otherwise, numStY set equal to 1.
 - If both nTbW and nTbH are equal to 4 or both nTbW and nTbH are equal to 8, nonZeroSize is set equal to 8.
 - Otherwise, nonZeroSize set equal to 16.
2. For xSbIdx = 0..numStX – 1 and ySbIdx = 0..numStY – 1, the following applies:
- The variable array u[x] with x = 0..nonZeroSize – 1 are derived as follows:

$$xC = (xSbIdx \ll \log2StSize) + \text{DiagScanOrder}[\log2StSize][\log2StSize][x][0]$$

$$yC = (ySbIdx \ll \log2StSize) + \text{DiagScanOrder}[\log2StSize][\log2StSize][x][1]$$

$$u[x] = d[xC][yC]$$
 - u[x] with x = 0..nonZeroSize – 1 is transformed to the variable array v[x] with x = 0..nStOutSize – 1 by invoking the one-dimensional transformation process as specified in clause 8.7.4.4 with the transform input length of the scaled transform coefficients nonZeroSize, the transform output length nStOutSize the list u[x] with x = 0..nonZeroSize – 1, the index for transform set selection stPredModeIntra, and the index for transform selection in a transform set st_idx[xTbY][yTbY] as inputs, and the output is the list v[x] with x = 0..nStOutSize – 1. The variable stPredModeIntra is set to the predModeIntra specified in clause 8.4.4.2.1.
 - The array d[(xSbIdx << log2StSize) + x][(ySbIdx << log2StSize) + y] with x = 0..nStSize – 1, y = 0..nStSize – 1 are derived as follows:
 - If stPredModeIntra is less than or equal to 34, or equal to INTRA_LT_CCLM, INTRA_T_CCLM, or INTRA_L_CCLM, the following applies:

$$d[(xSbIdx \ll \log2StSize) + x][(ySbIdx \ll \log2StSize) + y] = (y < 4) ? v[x + (y \ll \log2StSize)] : ((x < 4) ? v[32 + x + ((y - 4) \ll 2)] : d[(xSbIdx \ll \log2StSize) + x][(ySbIdx \ll \log2StSize) + y])$$
 - Otherwise, the following applies:

$$d[(xSbIdx \ll \log2StSize) + x][(ySbIdx \ll \log2StSize) + y] = (y < 4) ? v[y + (x \ll \log2StSize)] : ((x < 4) ? v[32 + (y - 4) + (x \ll 2)] : d[(xSbIdx \ll \log2StSize) + x][(ySbIdx \ll \log2StSize) + y])$$

Secondary transformation process

Inputs to this process are:

- a variable nTrS specifying the transform output length,
- a variable nonZeroSize specifying the transform input length,
- a list of transform input $x[j]$ with $j = 0..nonZeroSize - 1$,
- a variable stPredModeIntra specifying the index for transform set selection,
- a variable stIdx specifying the index for transform selection in a set.

Output of this process is the list of transformed samples $y[i]$ with $i = 0..nTrS - 1$.

The transformation matrix derivation process as specified in clause 8.7.4.5 is involved with the transform output length nTrS, the index for transform set selection stPredModeIntra, and the index for transform selection in a transform set stIdx as inputs, and the transformation matrix secTransMatrix as output.

The list of transformed samples $y[i]$ with $i = 0..nTrS - 1$ is derived as follows:

$$y[i] = \text{Clip3}(\text{CoeffMin}, \text{CoeffMax}, ((\sum_{j=0}^{nonZeroSize-1} \text{secTransMatrix}[j][i] * x[j]) + 64) \gg 7)$$

with $i = 0..nTrS - 1$

CoeffMin = $-(1 \ll 15)$ and CoeffMax = $(1 \ll 15) - 1$;

Secondary Transformation matrix derivation process

Inputs to this process are:

- a variable nTrS specifying the transform output length,
- a variable stPredModeIntra specifying the index for transform set selection,
- a variable stIdx specifying the index for transform selection in the designated transform set.

Output of this process is the transformation matrix secTransMatrix.

The variable stTrSetIdx is derived as follows:

stPredModeIntra	stTrSetIdx
stPredModeIntra < 0	1
0 <= stPredModeIntra <= 1	0
2 <= stPredModeIntra <= 12	1
13 <= stPredModeIntra <= 23	2
24 <= stPredModeIntra <= 44	3
45 <= stPredModeIntra <= 55	2
56 <= stPredModeIntra	1

The transformation matrix secTransMatrix is derived based on nTrS, stTrSetIdx, and stIdx as follows:

– If nTrS is equal to 16, stTrSetIdx is equal to 0, and stIdx is equal to 1, the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 16, stTrSetIdx is equal to 0, and stIdx is equal to 2, the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 16, stTrSetIdx is equal to 1, and stIdx is equal to 1, the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 16, stTrSetIdx is equal to 1, and stIdx is equal to 2, the following applies:

secTransMatrix [m][n] = ...

– If nTrS is equal to 16, stTrSetIdx is equal to 2, and stIdx is equal to 1, the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 16, stTrSetIdx is equal to 2, and stIdx is equal to 2, the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 16, stTrSetIdx is equal to 3, and stIdx is equal to 1, the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 16, stTrSetIdx is equal to 3, and stIdx is equal to 2, the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 48, stTrSetIdx is equal to 0, and stIdx is equal to 1 the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 48, stTrSetIdx is equal to 0, and stIdx is equal to 2 the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 48, stTrSetIdx is equal to 1, and stIdx is equal to 1 the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 48, stTrSetIdx is equal to 1, and stIdx is equal to 2 the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 48, stTrSetIdx is equal to 2, and stIdx is equal to 1 the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 48, stTrSetIdx is equal to 2, and stIdx is equal to 2 the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 48, stTrSetIdx is equal to 3, and stIdx is equal to 1 the following applies:
secTransMatrix [m][n] = ...

– If nTrS is equal to 48, stTrSetIdx is equal to 3, and stIdx is equal to 2 the following applies:
secTransMatrix [m][n] = ...

2.11 Clipping of Dequantization in HEVC

[00117] In HEVC, the scaled transform coefficient d' is calculated as

[00118] $d' = \text{Clip3}(\text{coeffMin}, \text{coeffMax}, d)$,

[00119] where d is the scaled transform coefficient before clipping.

[00120] For luma component,

[00121] $\text{coeffMin} = \text{CoeffMinY}, \text{coeffMax} = \text{CoeffMaxY};$

[00122] For chroma components,

[00123] $\text{coeffMin} = \text{CoeffMinC}, \text{coeffMax} = \text{CoeffMaxC}.$

[00124] Herein,

[00125] $\text{CoeffMinY} = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} +$

6): 15))

[00126] $\text{CoeffMinC} = -(1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15))$

[00127] $\text{CoeffMaxY} = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthY} + 6) : 15)) - 1$

[00128] $\text{CoeffMaxC} = (1 \ll (\text{extended_precision_processing_flag} ? \text{Max}(15, \text{BitDepthC} + 6) : 15)) - 1$

[00129] `extended_precision_processing_flag` is a syntax element signaled in SPS.

2.12 Affine linear weighted intra prediction (ALWIP, or matrix based intra prediction)

[00130] Affine linear weighted intra prediction (ALWIP, or matrix based intra prediction (MIP)) is proposed in JVET-N0217.

[00131] In JVET-N0217, two tests are conducted. In test 1, ALWIP is designed with a memory restriction of 8K bytes and at most 4 multiplications per sample. Test 2 is similar to test 1, but further simplifies the design in terms of memory requirement and model architecture.

[00132] ○ Single set of matrices and offset vectors for all block shapes.

[00133] ○ Reduction of number of modes to 19 for all block shapes.

[00134] ○ Reduction of memory requirement to 5760 10-bit values, that is 7.20 Kilobyte.

[00135] ○ Linear interpolation of predicted samples is carried out in a single step per direction replacing iterative interpolation as in the first test.

2.13 Sub-block transform

[00136] 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.

[00137] In sub-block transform, 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. 15. 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.

3 Examples of drawbacks in existing implementations

[00138] The current design of MTS has the following problems:

[00139] The signaled index may cause overhead bits.

[00140] Some combinations of transforms may not be efficient in MTS and SBT.

4 Exemplary methods for multiple transforms

[00141] Embodiments of the presently disclosed technology overcome drawbacks of existing implementations, thereby providing video coding with higher coding efficiencies but lower computational complexity. Methods for multiple transforms, and as described in the present document, may enhance both existing and future video coding standards, is elucidated in the following examples described for various implementations. The examples of the disclosed technology provided below explain general concepts, and are not meant to be interpreted as limiting. In an example, unless explicitly indicated to the contrary, the various features described in these examples may be combined.

[00142] In the following examples, it is assumed that:

[00143] ○ Max(x, y) returns the larger one of x and y, and

[00144] ○ Min(x, y) returns the smaller one of x and y.

Implicit MTS

It is proposed to decide the transform (horizontal and/or vertical transform) applied to one block according to decoded coefficients without receiving transform indices.

1. The decoded coefficients may be associated with one or multiple representative blocks in the same color component as the current block or different color component.
 - a. In one example, the determination on the transform of a first block may depend on the decoded coefficients of the first block.
 - b. In one example, the determination on the transform of a first block may depend on the decoded coefficients of a second block which may be different to the first block.

- i. In one example, the second block may be in the same color component, such as the luma component, as the color component of the first block.
 - 1) In one example, the second block may be neighbouring to the first block.
 - 2) In one example, the second block may be with the same intra prediction mode as the first block.
 - 3) In one example, the second block may be with the same block dimensions as the first block.
 - 4) In one example, the second block may be the last decoded block satisfying certain conditions, such as the same intra-prediction mode or the same dimensions, before the first block in the decoding order.
- ii. In one example, the second block may be in a different color component from that of the first block.
 - 1) In one example, the first block may be in the luma component, and the second block may be in a chroma component (e.g., the Cb/Cr, B/R component).
 - a) In one example, the three blocks are in the same coding unit.
 - b) Alternatively, furthermore, implicit MTS is applied only to the luma blocks, not to chroma blocks.
 - 2) In one example, the first block in the first color component and the second block in the second color component may be at the corresponding locations of a picture with each other.
- c. In one example, the determination on the transform of a first block may depend on the decoded coefficients of multiple blocks comprising at least one block not identical to the first block.
 - i. In one example, multiple blocks may comprise the first block.
 - ii. In one example, multiple blocks may comprise one block or plurality of neighbouring to the first block.

- iii. In one example, multiple blocks may comprise one block or plurality of blocks with the same block dimensions as the first block.
 - iv. In one example, multiple blocks may comprise last N decoded block satisfying certain conditions, such as the same intra-prediction mode or the same dimensions, before the first block in the decoding order. N is an integer larger than 1.
 - v. In one example, multiple blocks may comprise one block or plurality of blocks not in the same color component as the first block.
 - 1) In one example, the first block may be in the luma component. Multiple blocks may comprise blocks in chroma components (e.g., a second block in the Cb/B component, and a third block in the Cr/R component).
 - a) In one example, the three blocks are in the same coding unit.
 - b) Alternatively, furthermore, implicit MTS is applied only to the luma blocks, not to chroma blocks.
 - 2) In one example, the first block in the first color component and the plurality of blocks not in the first component color component comprised in the multiple blocks may be at the corresponding locations of a picture with the first block.
2. The decoded coefficients utilized for transform determination are those unequal to zero (denoted as significant coefficients). Those coefficients utilized for transform determination are called representative coefficients.
- a. In one example, representative coefficients are all the significant coefficients in representative blocks.
 - b. Alternatively, representative coefficients are partial of significant coefficients in representative blocks.
 - i. In one example, representative coefficients are those decoded significant coefficients which are larger than or no smaller than a threshold
 - ii. In one example, representative coefficients are those decoded significant coefficients which are smaller than or no greater than a threshold

- iii. In one example, representative coefficients are the first K ($K \geq 1$) decoded significant coefficients in the decoding order.
- iv. In one example, representative coefficients are the last K ($K \geq 1$) decoded significant coefficients in the decoding order.
- v. In one example, representative coefficients may be those at a predefined location in a block.
 - 1) In one example, representative coefficients may comprise only one coefficient located at (xPos, yPos) coordinate relative to the representative block. E.g. xPos = yPos = 0.
 - 2) For example, the positions may depend on the dimensions of the block.
- vi. In one example, representative coefficients may be those at a predefined position in the coefficient scanning order.
- c. Alternatively, representative coefficients may also comprise those zero coefficients.
- d. Alternatively, representative coefficients may be those derived from decoded coefficients, such as via clipping to a range, via quantization.
- 3. The transform determination may depend on a function of representative coefficients, such as a function with a value V as the output, using representative coefficients as inputs.
 - a. In one example, V is derived as the number of representative coefficients.
 - i. Alternatively, V is derived as the sum of representative coefficients.
 - 1) Alternatively, furthermore, the sum may be clipped to derive V .
 - ii. Alternatively, V is derived as the sum of absolute values of representative coefficients.
 - 1) Alternatively, furthermore, the sum may be clipped to derive V .
 - b. In one example, the selection may be determined implicitly at the decoder depending on the parity of V .
 - i. For example, if V is even, a first kind of transform is selected as the horizontal transform and a second kind of transform is selected as the vertical transform; and if V is odd, a third kind of transform is selected

as the horizontal transform and a forth kind of transform is selected as the vertical transform.

- 1) In one example, the first kind of transform is the same to the second kind of transform.
 - a) Alternatively, the first kind of transform is different to the second kind of transform.
- 2) In one example, the third kind of transform is the same to the forth kind of transform.
 - a) Alternatively, the third kind of transform is different to the forth kind of transform.
- 3) The first/second/third/forth kind of transform is a specific transform such as DCT-X or DST-Y. X may be an integer such as 2 or 8. Y may be an integer such as 7 or 8.
- 4) Alternatively, furthermore, at least one of the third and fourth kinds of transforms is different from the first and second kinds of transform.
 - a) In one example, when V is even, the first and second kinds of transforms are DCT-2 and when V is odd, the third and fourth kinds of transforms are DST-7.
 - b) Alternatively, when V is odd, the first and second kinds of transforms are DCT-2 and when V is even, the third and fourth kinds of transforms are DST-7.
- c. In one example, if V is smaller than a threshold T1, a fifth kind of transform is selected as the horizontal transform and a sixth kind of transform is selected as the vertical transform. e.g. T1 = 1 or 2.
 - i. Alternatively, if V is larger than a threshold T2, a fifth kind of transform is selected as the horizontal transform and a sixth kind of transform is selected as the vertical transform.
 - ii. For example, the threshold may depend on the dimensions of the block.
 - iii. For example, the threshold may depend on the QP.

- iv. In one example, the fifth kind of transform is the same to the sixth kind of transform.
 - 1) Alternatively, the fifth kind of transform is different to the sixth kind of transform.
 - v. In one example, the fifth/sixth kind of transform is a specific transform such as DCT-X or DST-Y. X may be an integer such as 2 or 8. Y may be an integer such as 7 or 8.
4. The transform determination may further depend on the coded information of current block.
- a. In one example, if current intra-coded block is in I-slice/picture, when V is even, DST-7 may be applied to current block and when V is odd, DCT-2 may be applied to current block.
 - b. In one example, if current intra-coded block is in P/B-slice/picture, and V is even, DCT-2 may be applied to current block and when V is odd, DST-7 may be applied to current block.
 - c. In one example, the determination may further depend on the mode information (e.g., intra or inter).
5. Transform sets which the implicit MTS may choose transforms from may be pre-defined.
- a. In one example, the horizontal and vertical transform sets may be not identical.
 - i. Alternatively, the horizontal and vertical transform sets may be not identical.
 - b. In one example, the transform set may include DCT-2 and DST-7.
 - c. In one example, the transform set may include DCT-2, DST-7 and identify transform.
 - d. In one example, the transform set may be dependent on coded information, color component, partitioning structure (e.g., dual tree/single tree; quadtree/binary tree/ternary tree/extended quadtree), slice/picture types etc. al.
 - i. In one example, the transform set may be dependent on block dimension.

- ii. In one example, for intra-coded blocks, DCT-2 and DST-7 may be included.
 - iii. In one example, for blocks coded with reference samples in the same picture (e.g., intra block copy), DST-7 and identity transform (i.e., without transform being applied) may be included.
6. In one example, one or multiple of the methods disclosed in bullet 1- bullet 5 can only be applied to specific blocks.
- a. For example, one or multiple of the methods disclosed in bullet 1- bullet 5 can only be applied to intra-coded blocks.

Simplified MTS/SBT

7. In one example, the vertical transform and the horizontal transform must be the same in MTS.
- a. In one example, a block can only choose one of the two kinds of transforms as below:
 - i. DCT-2 in horizontal transform and DCT-2 in vertical transform;
 - ii. DST-7 in horizontal transform and DST-7 in vertical transform;
 - b. In one example, the signaling for MTS may comprise at most one flag for a block.
 - i. In one example, if the flag is equal to 0, DCT-2 in horizontal transform and DCT-2 in vertical transform; if the flag is equal to 1, DST-7 in horizontal transform and DST-7 in vertical transform.
 - ii. In one example, if the flag is equal to 1, DCT-2 in horizontal transform and DCT-2 in vertical transform; if the flag is equal to 0, DST-7 in horizontal transform and DST-7 in vertical transform.
8. Suppose the block width and height are W and H, respectively. Bullet 7 may be applied only if
- a. $W \geq T1$ and $H \geq T2$, e.g. $T1=T2=8$;
 - b. $W \leq T1$ and $H \leq T2$, e.g. $T1=T2=16$;
 - c. $\text{Min}(W, H) \geq T1$, e.g. $T1=8$;
 - d. $\text{Max}(W, H) \leq T1$, e.g. $T1=32$;

- e. $W \cdot H \geq T1$, e.g. $T1=64$;
 - f. $W \cdot H \leq T1$, e.g. $T1=256$.
9. In one example, transforms excluding DST-8 can be applied in a block coded with SBT.
- a. In one example, only DCT-2 and DST-7 can be applied in a block coded with SBT.
 - b. In one example, in the case of (SBT-V, position 0) for SBT as shown in FIG. 15, DCT-2 is applied horizontally and DST-7 is applied vertically.
 - c. In one example, in the case of (SBT-H, position 0) for SBT as shown in FIG. 15, DST-7 is applied horizontally and DCT-2 is applied vertically.
10. Suppose the transform block width and height are W and H , respectively. In one example, the selection of transforms for a block coded with SBT may depend on transform block dimensions, wherein the transform block may be smaller than the coding block when SBT is applied.
- a. In one example, in the case of (SBT-V, position 0) for SBT as shown in FIG. 15, DCT-2 is applied horizontally and DST-7 is applied vertically if $W \geq T1$; Otherwise, DCT-8 is applied horizontally and DST-7 is applied vertically. For example, $T1 = 8$.
 - b. In one example, in the case of (SBT-H, position 0) for SBT as shown in FIG. 15, DST-7 is applied horizontally and DCT-2 is applied vertically if $H \geq T1$; Otherwise, DST-7 is applied horizontally and DCT-8 is applied vertically. For example, $T1 = 8$.

General claims

11. The decision of transform matrix may be done in CU/CB-level or TU-level.
- a. In one example, the decision is made in CU level wherein all TUs share the same transform matrix.
 - i. Alternatively, furthermore, when one CU is split to multiple TUs, coefficients in one TU (e.g., the first or the last TU) or partial or all TUs may be utilized to determine the transform matrix.

- b. Whether to use the CU-level solution or TU-level solution may depend on the block size and/or VPDU size and/or maximum CTU size and/or coded information of one block.
 - i. In one example, when the block size is larger than the VPDU size, CU-level determination method may be applied.
- 12. In the disclosed methods in the document, the coefficients or representative coefficients may be quantized or dequantized.
- 13. Transform skip may also be determined by the coefficients or representative coefficients implicitly with any disclosed method in the document.
- 14. In the disclosed methods in the document, the coefficients or representative coefficients may be modified before being used to derive the transforms.
 - a. For example, a coefficient or representative coefficients may be clipped before being used to derive the transforms.
 - b. For example, a coefficient or representative coefficients may be scaled before being used to derive the transforms.
 - c. For example, a coefficient or representative coefficients may be added by an offset before being used to derive the transforms.
 - d. For example, coefficients or representative coefficients may be filtered before being used to derive the transforms.
 - e. For example, a coefficient or representative coefficients may be mapped to other values (e.g., via look up tables) before being used to derive the transforms.
- 15. The disclosed methods in the document may also be used to derive other coding modes/information by the coefficients or representative coefficients implicitly.
 - a. In one example, the disclosed methods may be used to derive the secondary transforms which may be applied to sub-region of a block.
 - b. Alternatively, furthermore, the representative coefficients are from those corresponding to the sub-region instead of the whole block.
- 16. In one example, whether to and/or how to apply the disclosed methods above may be signaled at sequence level/picture level/slice level/tile group level, such as in sequence header/picture header/SPS/VPS/DPS/PPS/APS/slice header/tile group header.

17. In one example, whether to and/or how to apply the disclosed methods above may depend on coding information which may include:

- a. Block dimensions.
 - i. In one example, for blocks with both width and height no greater than a threshold (e.g., 32), the above-mentioned implicit MTS method may be applied.
- b. QPs
- c. Picture or slice type (such as I-frame or P/B-frame, I-slice or P/B-slice)
 - i. In one example, the proposed method may be enabled on I-frames but be disabled on P/B frames.
- d. Structure partitioning method (single tree or dual tree)
 - i. In one example, for single tree partitioning applied slices/pictures/bricks/tiles, the above-mentioned implicit MTS method may be applied.
- e. Coding mode (such as inter mode/intra mode/IBC mode etc.)
 - i. In one example, for intra-coded blocks, the above-mentioned implicit MTS method may be applied.
- f. Coding methods (such as Intra Sub-block partition, Derived Tree (DT) method, etc.)
 - i. In one example, for intra-coded blocks with DT applied, the above-mentioned implicit MTS method may be disabled.
 - ii. In one example, for intra-coded blocks with ISP applied, the above-mentioned implicit MTS method may be disabled.
- g. Color components
 - i. In one example, for luma blocks, the above-mentioned implicit MTS method may be applied while for chroma blocks, it is not applied.
- h. Intra-prediction mode (such as DC, vertical, horizontal, etc.)
- i. Motion information (such as MV and reference index).
- j. Standard Profiles/Levels/Tiers

[00145] The examples described above may be incorporated in the context of the methods

described below, e.g., methods 1675, 1680 and 1685, which may be implemented at a video encoder and/or decoder.

[00146] FIG. 16A shows a flowchart of an exemplary method for video processing. The method 1675 includes, at step 1676, making a decision, based on a set of coefficients, regarding an application of a transform to a current block of a video.

[00147] The method 1675 includes, at step 1677, performing, based on the decision, a conversion between the current block and a bitstream representation of the video.

[00148] FIG. 16B shows a flowchart of an exemplary method for video processing. The method 1680 includes, at step 1681, making a decision, based on coding information associated with a current block of a video, regarding an application of a transform to the current block.

[00149] The method 1680 includes, at step 1682, performing, based on the decision, a conversion between the current block and a bitstream representation of the video.

[00150] FIG. 16C shows a flowchart of an exemplary method for video processing. The method 1685 includes, at step 1686, making a decision, for a current block of a video, regarding an application of a transform to the current block based on an indication in a bitstream representation of the video at a coding unit (CU) level, a coding block (CB) level, or a transform unit (TU) level.

[00151] The method 1685 includes, at step 1687, performing, based on the decision, a conversion between the current block and the bitstream representation.

5 Example implementations of the disclosed technology

[00152] In the following embodiments, additions are indicated using bolded double braces, e.g., **{{a}}** indicates that “a” has been added, whereas deletions are indicated using bolded double brackets, e.g., **[[a]]** indicates that “a” has been deleted.

5.1 Embodiment #1

[00153] The working draft specified in JVET-N1001-v7 may be changed as below.

8.7.4 Transformation process for scaled transform coefficients

8.7.4.1 General

...

Table 8-17 – Specification of trTypeHor and trTypeVer depending on cu_sbt_horizontal_flag and cu_sbt_pos_flag

cu_sbt_horizontal_flag	cu_sbt_pos_flag	trTypeHor	trTypeVer
0	0	{{nTbW >= 8? 0:}} 2	1
0	1	1	1
1	0	1	{{nTbH >= 8? 0:}} 2
1	1	1	1

...

[00154] In some embodiments, the following technical solutions can be implemented:

[00155] C1. A method for video processing, comprising making a decision, based on a set of coefficients, regarding an application of a transform to a current block of a video; and performing, based on the decision, a conversion between the current block and a bitstream representation of the video.

[00156] C2. The method of solution C1, wherein the set of coefficients correspond to a set of quantized coefficients in the bitstream representation.

[00157] C3. The method of solution C1, wherein the set of coefficients correspond to a dequantized version of a set of quantized coefficients in the bitstream representation.

[00158] C4. The method of any of solutions C1 to C3, wherein an application of a transform skip to the current block is based on the set of coefficients.

[00159] C5. The method of any of solutions C1 to C3, further comprising modifying the set of coefficients; and deriving, prior to making the decision, the transform based on the modified set of coefficients.

[00160] C6. The method of solution C5, wherein modifying the set of coefficients comprises clipping at least one of the set of coefficients.

[00161] C7. The method of solution C5, wherein modifying the set of coefficients comprises scaling at least one of the set of coefficients.

[00162] C8. The method of solution C5, wherein modifying the set of coefficients comprises adding an offset to at least one of the set of coefficients.

[00163] C9. The method of solution C5, wherein modifying the set of coefficients comprises filtering at least one of the set of coefficients.

[00164] C10. The method of solution C5, wherein modifying the set of coefficient comprises mapping the set of coefficients, based on at least one look-up table (LUT), to the modified set of coefficients.

[00165] C11. The method of any of solutions C1 to C3, wherein deriving a coding mode

and/or information associated with the current block is implicitly based on the set of coefficients.

[00166] C12. The method of any of solutions C1 to C3, wherein deriving a secondary transform applicable to a sub-region of the current block is based on the set of coefficients.

[00167] C13. The method of solution C12, wherein the set of coefficients correspond to the sub-region.

[00168] C14. The method of any of solutions C1 to C3, wherein making the decision is further based on an indication that is signaled in the bitstream representation at a sequence level, a picture level, a slice level, a tile group level.

[00169] C15. The method of any of solutions C1 to C3, wherein the indication is signaled in a sequence header, a picture header, a sequence parameter set (SPS), a video parameter set (VPS), a decoder parameter set (DPS), a picture parameter set (PPS), an adaptation parameter set (APS), a slice header, or a tile group header.

[00170] C16. The method of any of solutions C1 to C15, wherein the set of coefficients comprises a set of representative coefficients.

[00171] C17. A method for video processing, comprising making a decision, based on coding information associated with a current block of a video, regarding an application of a transform to the current block; and performing, based on the decision, a conversion between the current block and a bitstream representation of the video.

[00172] C18. The method of solution C17, wherein the transform is associated with an implicit selection of transforms (IST) mode, and wherein the IST mode comprises a selectively applying a transform type other than a Discrete Cosine Transform of Type II (DCT-II).

[00173] C19. The method of solution C17, wherein the coding information comprises a dimension of the current block.

[00174] C20. The method of solution C19, wherein the transform is applied when a height and a width of the current block is no greater than a threshold (T), and wherein T is a positive integer.

[00175] C21. The method of solution C20, wherein $T = 32$.

[00176] C22. The method of solution C17, wherein the coding information comprises a quantization parameter (QP).

[00177] C23. The method of solution C17, wherein the coding information comprises a picture type of a current picture comprising the current block.

- [00178] C24. The method of solution C23, wherein the picture type is an I-frame or a P/B-frame.
- [00179] C25. The method of solution C24, wherein the transform is applied when the picture type is an I-frame, and wherein the transform is not applied when the picture type is a P/B-frame.
- [00180] C26. The method of solution C17, wherein the coding information comprises a slice type of a current slice comprising the current block.
- [00181] C27. The method of solution C26, wherein the slice type is an I-slice or a P/B-slice.
- [00182] C28. The method of solution C17, wherein the coding information comprises a partitioning method of the current block.
- [00183] C29. The method of solution C28, wherein the partitioning method is a single tree or a dual tree.
- [00184] C30. The method of solution C17, wherein the coding information comprises a coding mode of the current block.
- [00185] C31. The method of solution C30, wherein the coding mode comprises at least one of an inter mode, an intra mode, or an intra block copy (IBC) mode.
- [00186] C32. The method of solution C30 or C31, wherein the transform is applied when the current block is an intra-coded block.
- [00187] C33. The method of solution C17, wherein the coding information comprises a coding method applied to the current block.
- [00188] C34. The method of solution C33, wherein the coding method is an intra sub-block partitioning (ISP) method and/or a derived tree (DT) method.
- [00189] C35. The method of solution C33, wherein the transform is not applied when the current block is an intra-coded block with a derived tree (DT) method applied to the current block.
- [00190] C36. The method of solution C33, wherein the transform is not applied when the current block is an intra-coded block with an intra sub-block partitioning (ISP) method applied to the current block.
- [00191] C37. The method of solution C17, wherein the coding information comprises a color component of the video.
- [00192] C38. The method of solution C37, wherein the transform is applied when the color component is a luma component of the video.

- [00193] C39. The method of solution C37, wherein the transform is not applied when the color component is a chroma component of the video.
- [00194] C40. The method of solution C17, wherein the coding information comprises an intra-prediction mode of the current block.
- [00195] C41. The method of solution C17, wherein the coding information comprises motion information associated with the current block.
- [00196] C42. The method of solution C41, wherein the motion information comprises one or more motion vectors and/or a reference index.
- [00197] C43. The method of solution C17, wherein the coding information comprises a standard profile, level, or tier of the current block.
- [00198] C44. A method for video processing, comprising making a decision, for a current block of a video, regarding an application of a transform to the current block based on an indication in a bitstream representation of the video at a coding unit (CU) level, a coding block (CB) level, or a transform unit (TU) level; and performing, based on the decision, a conversion between the current block and the bitstream representation.
- [00199] C45. The method of solution C44, wherein making the decision is based on the indication at the CU level, and wherein all TUs share an identical transform matrix.
- [00200] C46. The method of solution C44, wherein one CU is split into a plurality of TUs, and wherein a transform matrix corresponding to the transform is determined based on coefficients in one or more TUs of the plurality of TUs .
- [00201] C47. The method of solution C46, wherein the one or more TUs comprise a first TU of the plurality of TUs.
- [00202] C48. The method of solution C46, wherein the one or more TUs comprise a last TU of the plurality of TUs.
- [00203] C49. The method of solution C46, wherein the one or more TUs exclude at least one TU of the plurality of TUs.
- [00204] C50. The method of solution C46, wherein the one or more TUs comprise each of the plurality of TUs.
- [00205] C51. The method of solution C44, whether making the decision is based on the indication at the CU level or the TU level is based on at least one of a size of the current block, a size of a virtual pipeline data unit (VPDU), a maximum size of a coding tree unit (CTU), and/or

coded information of the current block.

[00206] C52. The method of solution C51, wherein the decision is based on the indication at the CU level when the size of the current block is larger than the size of the VPDU.

[00207] C53. The method of any of solutions C1 to C52, wherein the conversion generates the current block from the bitstream representation.

[00208] C54. The method of any of solutions C1 to C52, wherein the conversion generates the bitstream representation from the current block.

[00209] C55. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method in any one of solutions C1 to C54.

[00210] C56. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method in any one of solutions C1 to C54.

[00211] FIG. 17 is a block diagram of a video processing apparatus 1700. The apparatus 1700 may be used to implement one or more of the methods described herein. The apparatus 1700 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1700 may include one or more processors 1702, one or more memories 1704 and video processing hardware 1706. The processor(s) 1702 may be configured to implement one or more methods (including, but not limited to, methods 1675, 1680 and 1685) described in the present document. The memory (memories) 1704 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1706 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00212] 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. 17.

[00213] 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.

[00214] 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 enabled based on the decision or determination.

[00215] FIG. 18 is a block diagram showing an example video processing system 1800 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 1800. The system 1800 may include input 1802 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 1802 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.

[00216] The system 1800 may include a coding component 1804 that may implement the various coding or encoding methods described in the present document. The coding component 1804 may reduce the average bitrate of video from the input 1802 to the output of the coding component 1804 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 1804 may be either stored, or transmitted via a communication connected, as represented by the component 1806. The stored or communicated bitstream (or coded) representation of the video received at the input 1802 may be used by the component 1808 for generating pixel values or displayable video that is sent to a display interface 1810. 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.

[00217] 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.

[00218] 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.

[00219] 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.

[00220] 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.

[00221] 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).

[00222] 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.

[00223] 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.

[00224] 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.

[00225] 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.

WHAT IS CLAIMED IS:

1. A method for video processing, comprising:
making a decision, based on a set of coefficients, regarding an application of a transform to a current block of a video; and
performing, based on the decision, a conversion between the current block and a bitstream representation of the video.
2. The method of claim 1, wherein the set of coefficients correspond to a set of quantized coefficients in the bitstream representation.
3. The method of claim 1, wherein the set of coefficients correspond to a dequantized version of a set of quantized coefficients in the bitstream representation.
4. The method of any of claims 1 to 3, wherein an application of a transform skip to the current block is based on the set of coefficients.
5. The method of any of claims 1 to 3, further comprising:
modifying the set of coefficients; and
deriving, prior to making the decision, the transform based on the modified set of coefficients.
6. The method of claim 5, wherein modifying the set of coefficients comprises clipping at least one of the set of coefficients.
7. The method of claim 5, wherein modifying the set of coefficients comprises scaling at least one of the set of coefficients.
8. The method of claim 5, wherein modifying the set of coefficients comprises adding an offset to at least one of the set of coefficients.
9. The method of claim 5, wherein modifying the set of coefficients comprises filtering at least one of the set of coefficients.

10. The method of claim 5, wherein modifying the set of coefficient comprises mapping the set of coefficients, based on at least one look-up table (LUT), to the modified set of coefficients.
11. The method of any of claims 1 to 3, wherein deriving a coding mode and/or information associated with the current block is implicitly based on the set of coefficients.
12. The method of any of claims 1 to 3, wherein deriving a secondary transform applicable to a sub-region of the current block is based on the set of coefficients.
13. The method of claim 12, wherein the set of coefficients correspond to the sub-region.
14. The method of any of claims 1 to 3, wherein making the decision is further based on an indication that is signaled in the bitstream representation at a sequence level, a picture level, a slice level, a tile group level.
15. The method of any of claims 1 to 3, wherein the indication is signaled in a sequence header, a picture header, a sequence parameter set (SPS), a video parameter set (VPS), a decoder parameter set (DPS), a picture parameter set (PPS), an adaptation parameter set (APS), a slice header, or a tile group header.
16. The method of any of claims 1 to 15, wherein the set of coefficients comprises a set of representative coefficients.
17. A method for video processing, comprising:
 - making a decision, based on coding information associated with a current block of a video, regarding an application of a transform to the current block; and
 - performing, based on the decision, a conversion between the current block and a bitstream representation of the video.
18. The method of claim 17, wherein the transform is associated with an implicit selection of transforms (IST) mode, and wherein the IST mode comprises a selectively applying a transform type other than a Discrete Cosine Transform of Type II (DCT-II).

19. The method of claim 17, wherein the coding information comprises a dimension of the current block.
20. The method of claim 19, wherein the transform is applied when a height and a width of the current block is no greater than a threshold (T), and wherein T is a positive integer.
21. The method of claim 20, wherein $T = 32$.
22. The method of claim 17, wherein the coding information comprises a quantization parameter (QP).
23. The method of claim 17, wherein the coding information comprises a picture type of a current picture comprising the current block.
24. The method of claim 23, wherein the picture type is an I-frame or a P/B-frame.
25. The method of claim 24, wherein the transform is applied when the picture type is an I-frame, and wherein the transform is not applied when the picture type is a P/B-frame.
26. The method of claim 17, wherein the coding information comprises a slice type of a current slice comprising the current block.
27. The method of claim 26, wherein the slice type is an I-slice or a P/B-slice.
28. The method of claim 17, wherein the coding information comprises a partitioning method of the current block.
29. The method of claim 28, wherein the partitioning method is a single tree or a dual tree.
30. The method of claim 17, wherein the coding information comprises a coding mode of the current block.
31. The method of claim 30, wherein the coding mode comprises at least one of an inter mode, an intra mode, or an intra block copy (IBC) mode.

32. The method of claim 30 or 31, wherein the transform is applied when the current block is an intra-coded block.
33. The method of claim 17, wherein the coding information comprises a coding method applied to the current block.
34. The method of claim 33, wherein the coding method is an intra sub-block partitioning (ISP) method and/or a derived tree (DT) method.
35. The method of claim 33, wherein the transform is not applied when the current block is an intra-coded block with a derived tree (DT) method applied to the current block.
36. The method of claim 33, wherein the transform is not applied when the current block is an intra-coded block with an intra sub-block partitioning (ISP) method applied to the current block.
37. The method of claim 17, wherein the coding information comprises a color component of the video.
38. The method of claim 37, wherein the transform is applied when the color component is a luma component of the video.
39. The method of claim 37, wherein the transform is not applied when the color component is a chroma component of the video.
40. The method of claim 17, wherein the coding information comprises an intra-prediction mode of the current block.
41. The method of claim 17, wherein the coding information comprises motion information associated with the current block.
42. The method of claim 41, wherein the motion information comprises one or more motion vectors and/or a reference index.

43. The method of claim 17, wherein the coding information comprises a standard profile, level, or tier of the current block.
44. A method for video processing, comprising:
making a decision, for a current block of a video, regarding an application of a transform to the current block based on an indication in a bitstream representation of the video at a coding unit (CU) level, a coding block (CB) level, or a transform unit (TU) level; and
performing, based on the decision, a conversion between the current block and the bitstream representation.
45. The method of claim 44, wherein making the decision is based on the indication at the CU level, and wherein all TUs share an identical transform matrix.
46. The method of claim 44, wherein one CU is split into a plurality of TUs, and wherein a transform matrix corresponding to the transform is determined based on coefficients in one or more TUs of the plurality of TUs .
47. The method of claim 46, wherein the one or more TUs comprise a first TU of the plurality of TUs.
48. The method of claim 46, wherein the one or more TUs comprise a last TU of the plurality of TUs.
49. The method of claim 46, wherein the one or more TUs exclude at least one TU of the plurality of TUs.
50. The method of claim 46, wherein the one or more TUs comprise each of the plurality of TUs.
51. The method of claim 44, whether making the decision is based on the indication at the CU level or the TU level is based on at least one of a size of the current block, a size of a virtual pipeline data unit (VPDU), a maximum size of a coding tree unit (CTU), and/or coded information of the current block.

52. The method of claim 51, wherein the decision is based on the indication at the CU level when the size of the current block is larger than the size of the VPDU.
53. The method of any of claims 1 to 52, wherein the conversion generates the current block from the bitstream representation.
54. The method of any of claims 1 to 52, wherein the conversion generates the bitstream representation from the current block.
55. An apparatus in a video system comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to implement the method recited in one or more of claims 1 to 54.
56. A computer program product stored on a non-transitory computer readable media, the computer program product including program code for carrying out the method recited in one or more of claims 1 to 54.

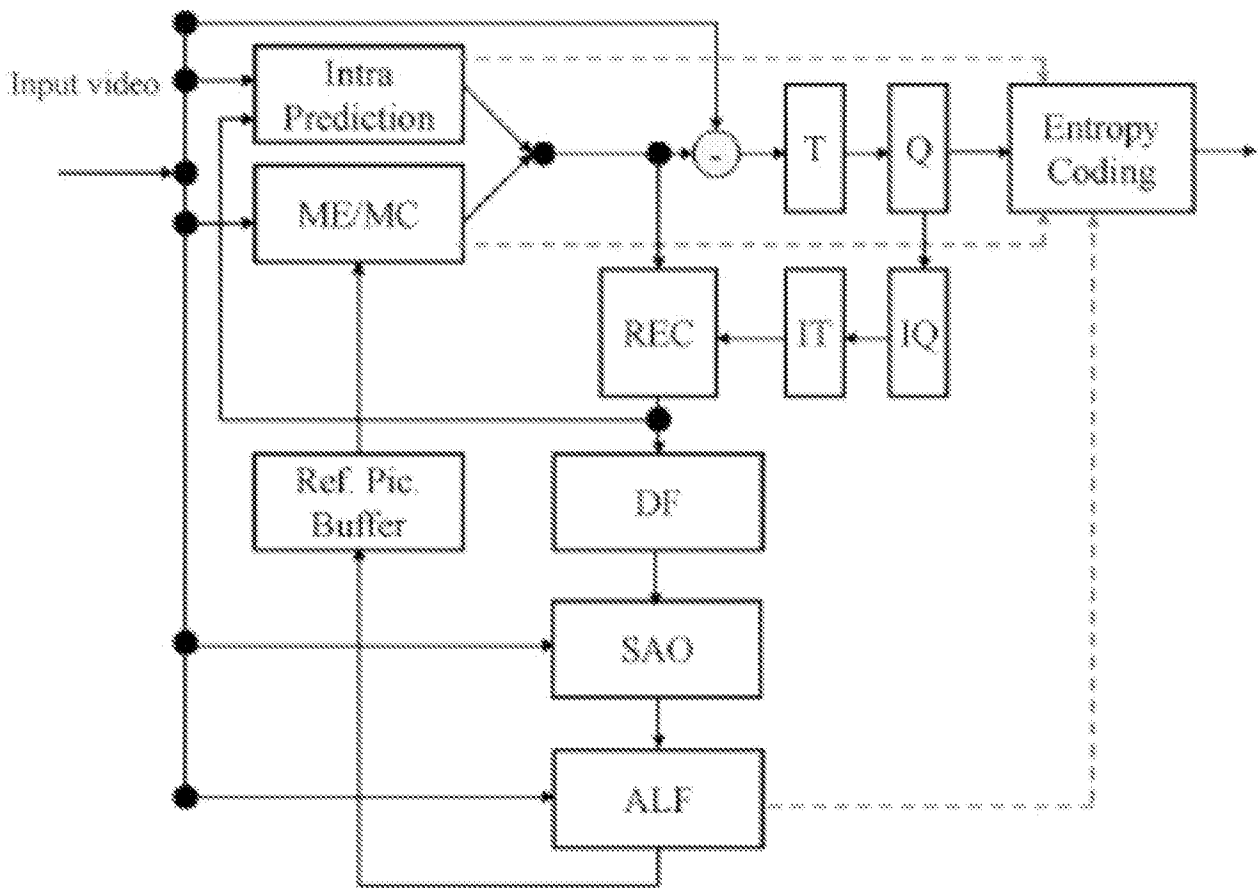


FIG. 1

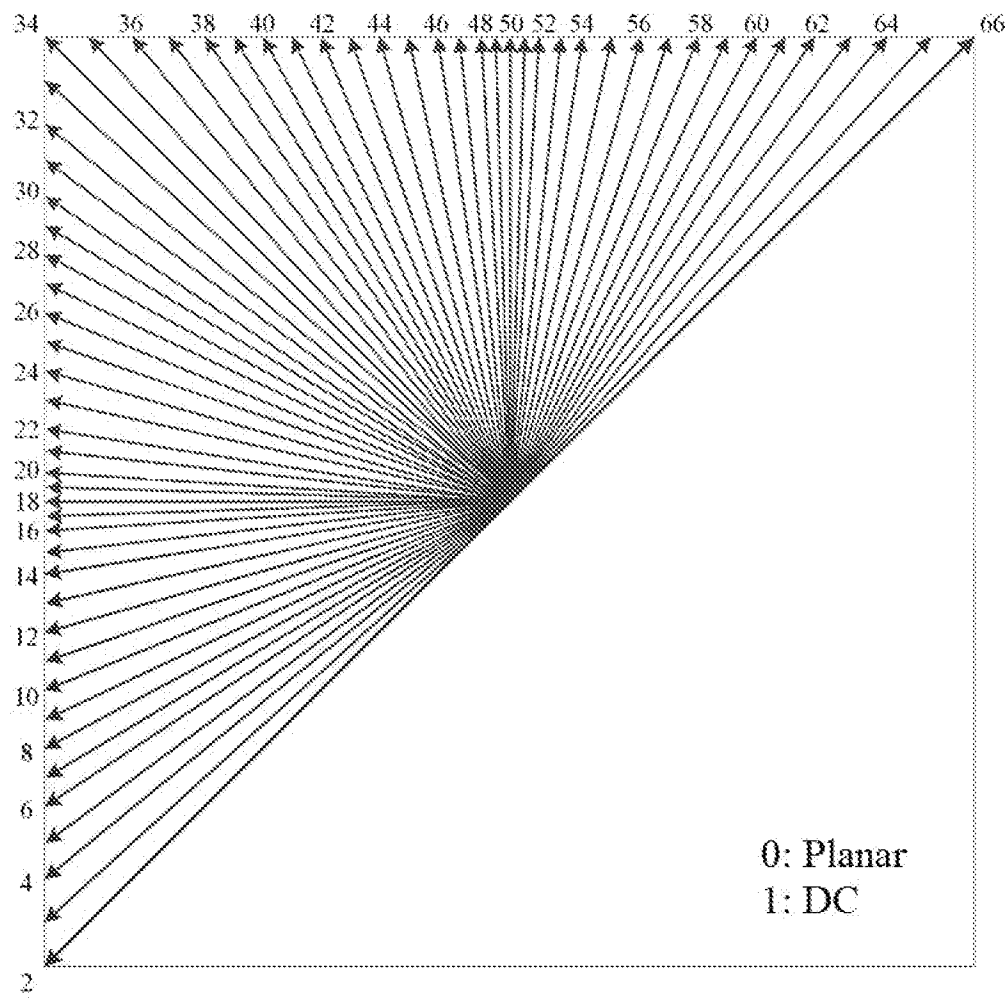


FIG. 2

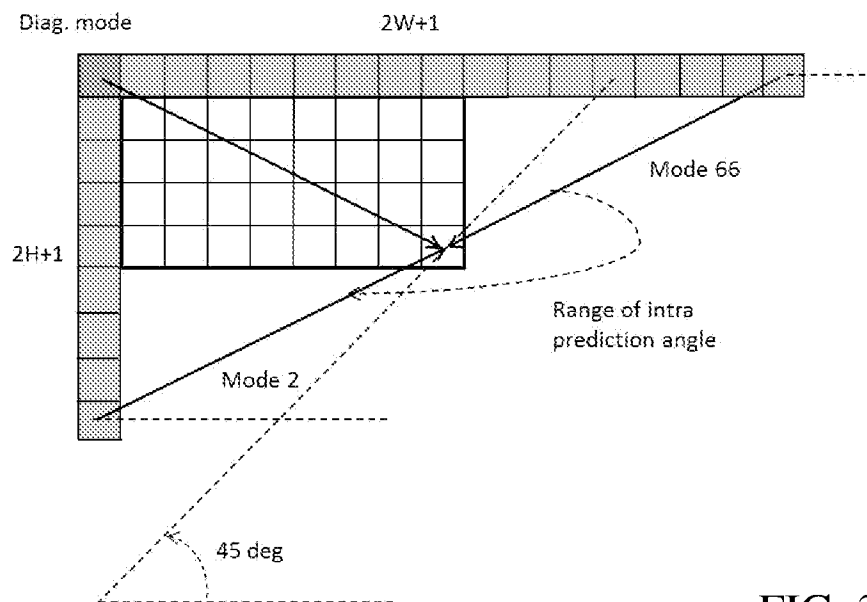


FIG. 3A

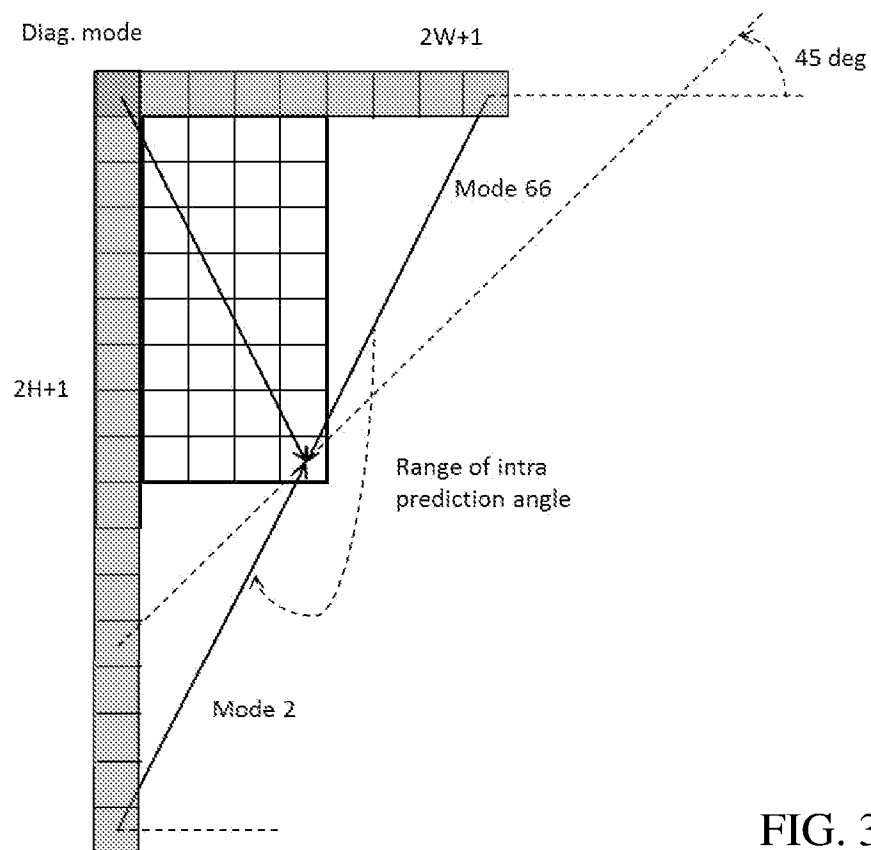


FIG. 3B

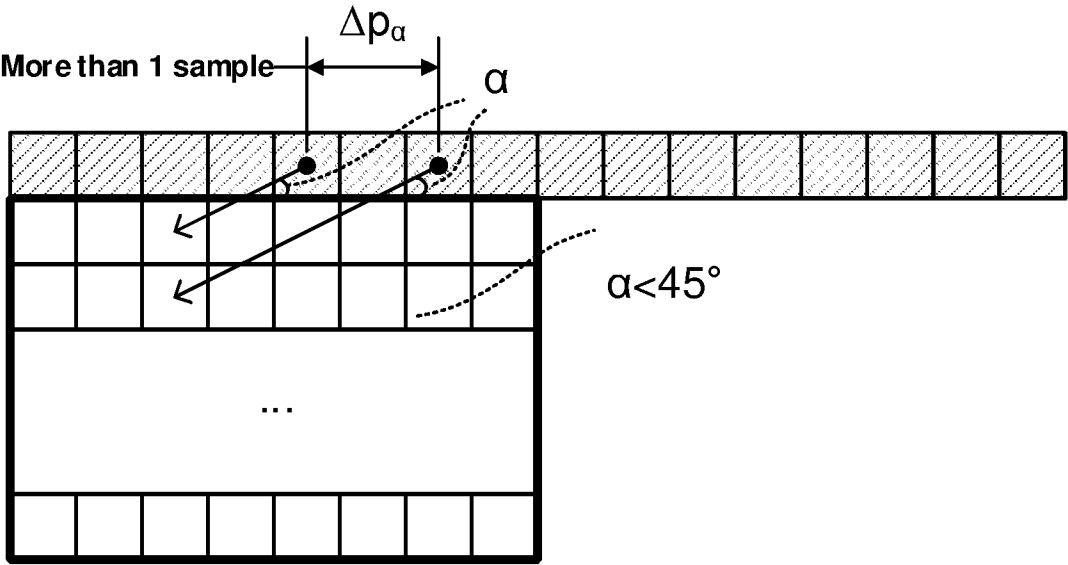


FIG. 4

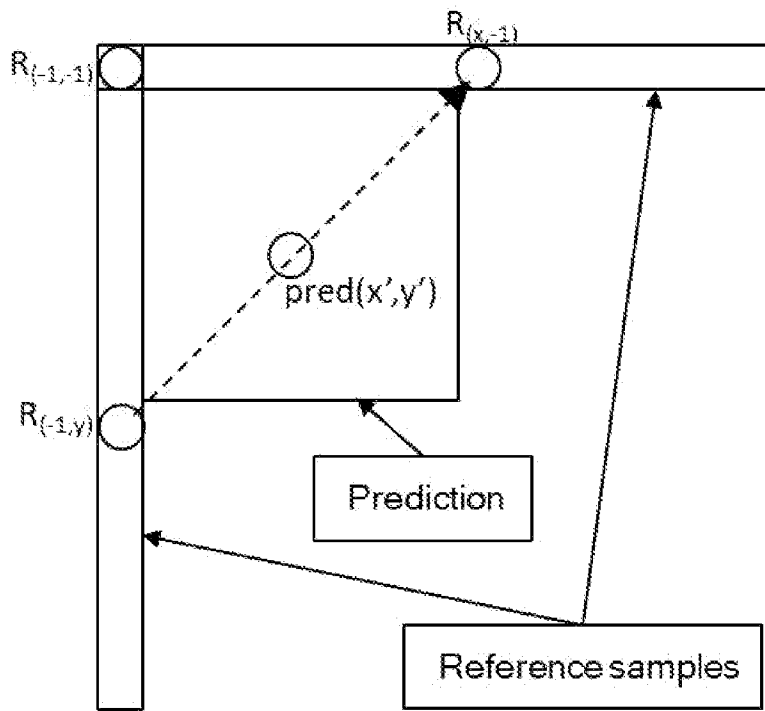


FIG. 5A

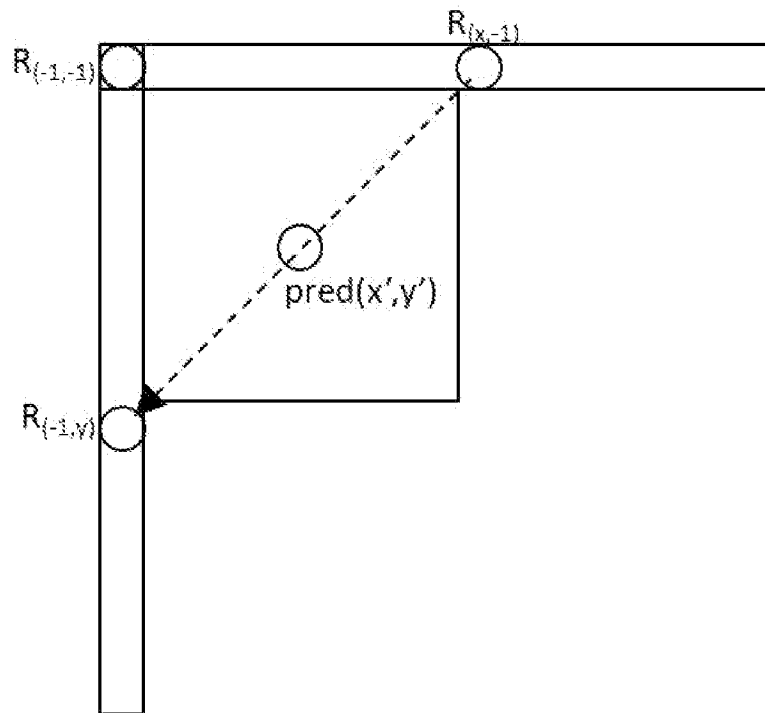


FIG. 5B

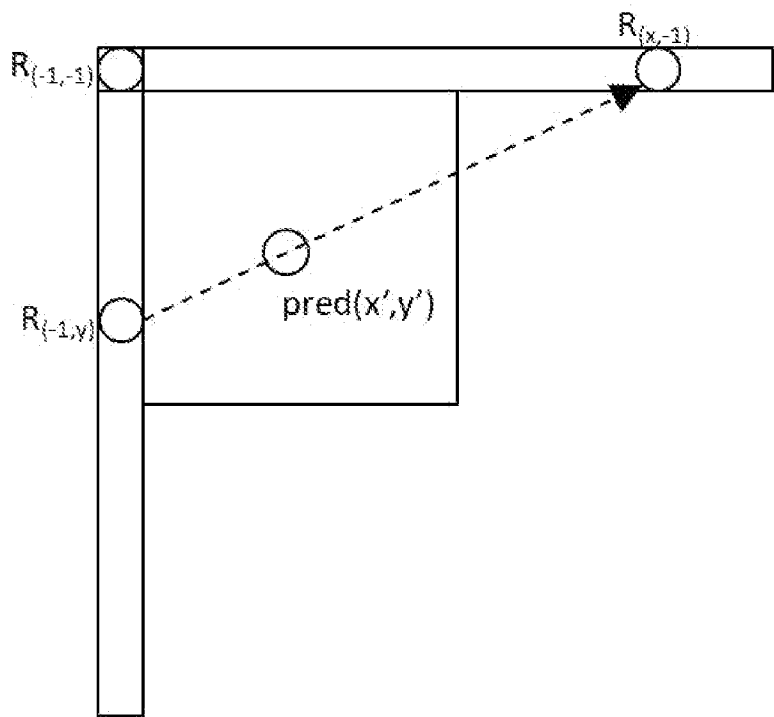


FIG. 5C

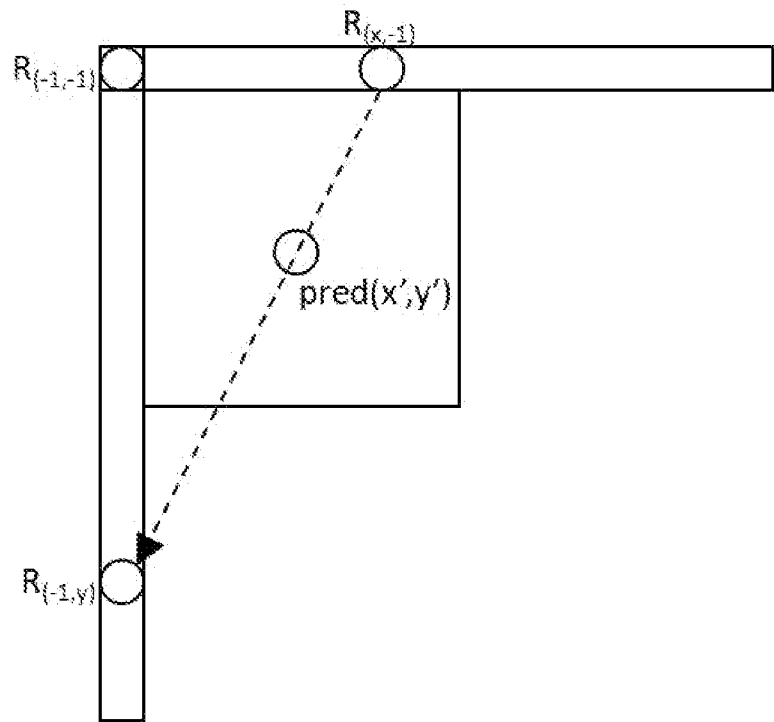


FIG. 5D

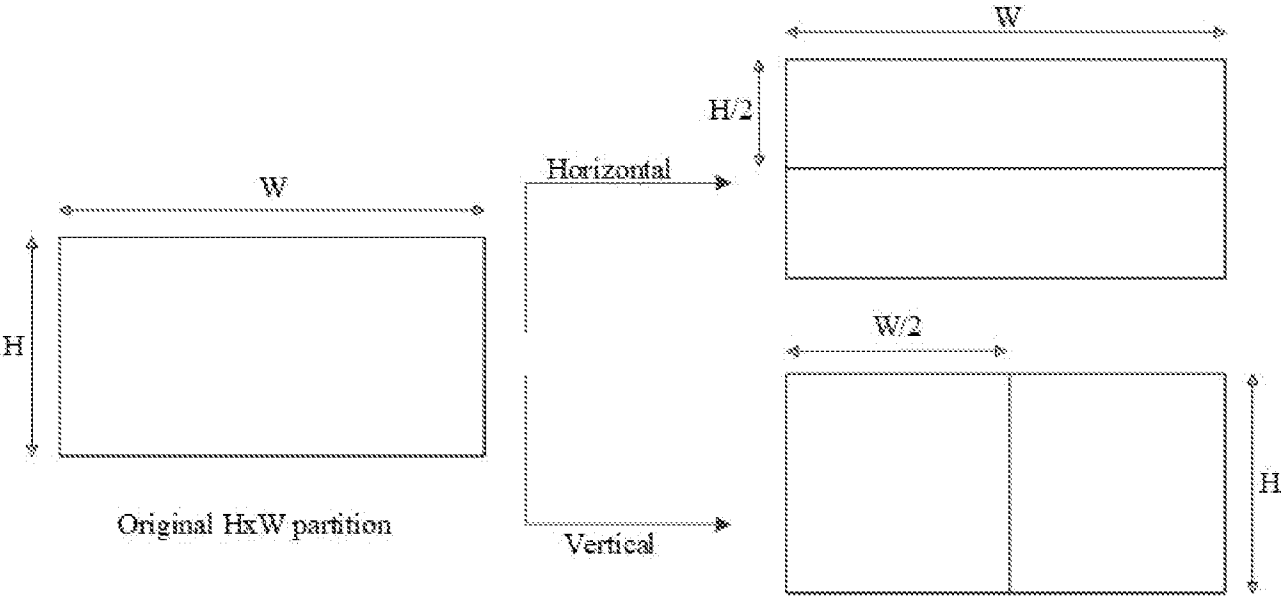


FIG. 6

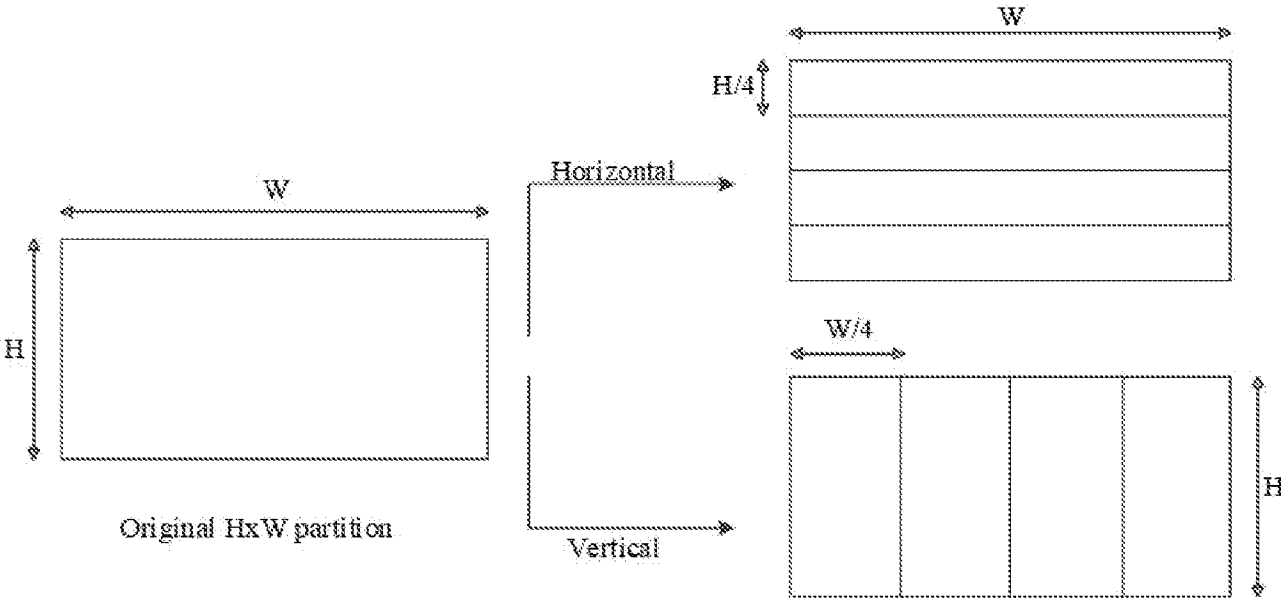


FIG. 7

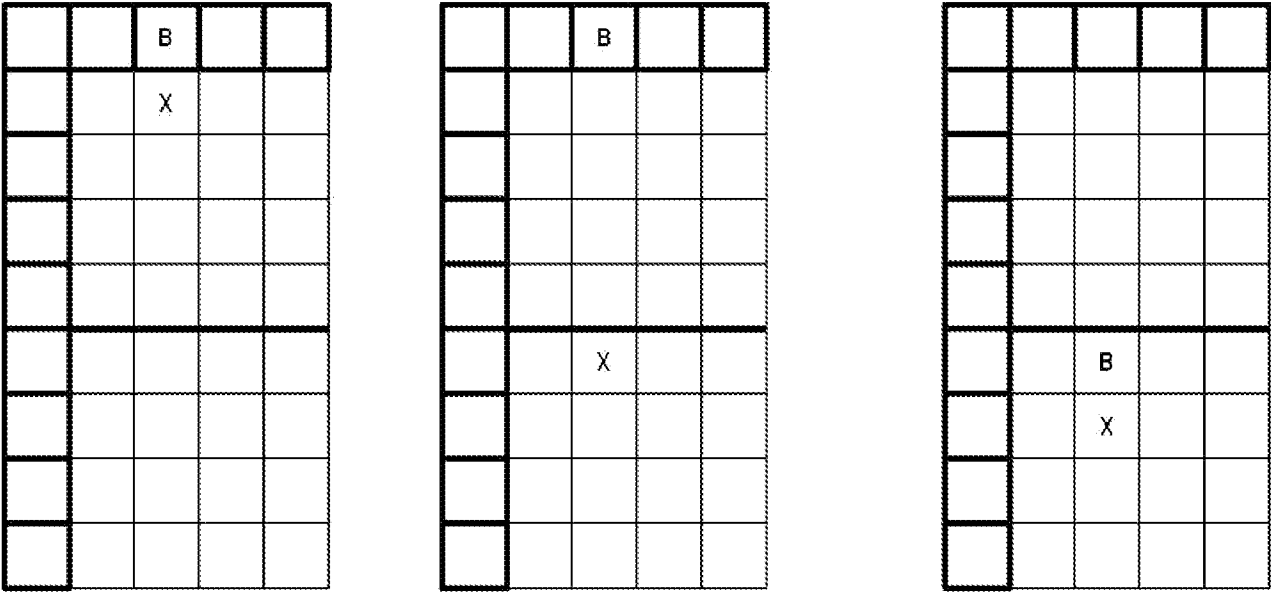


FIG. 8

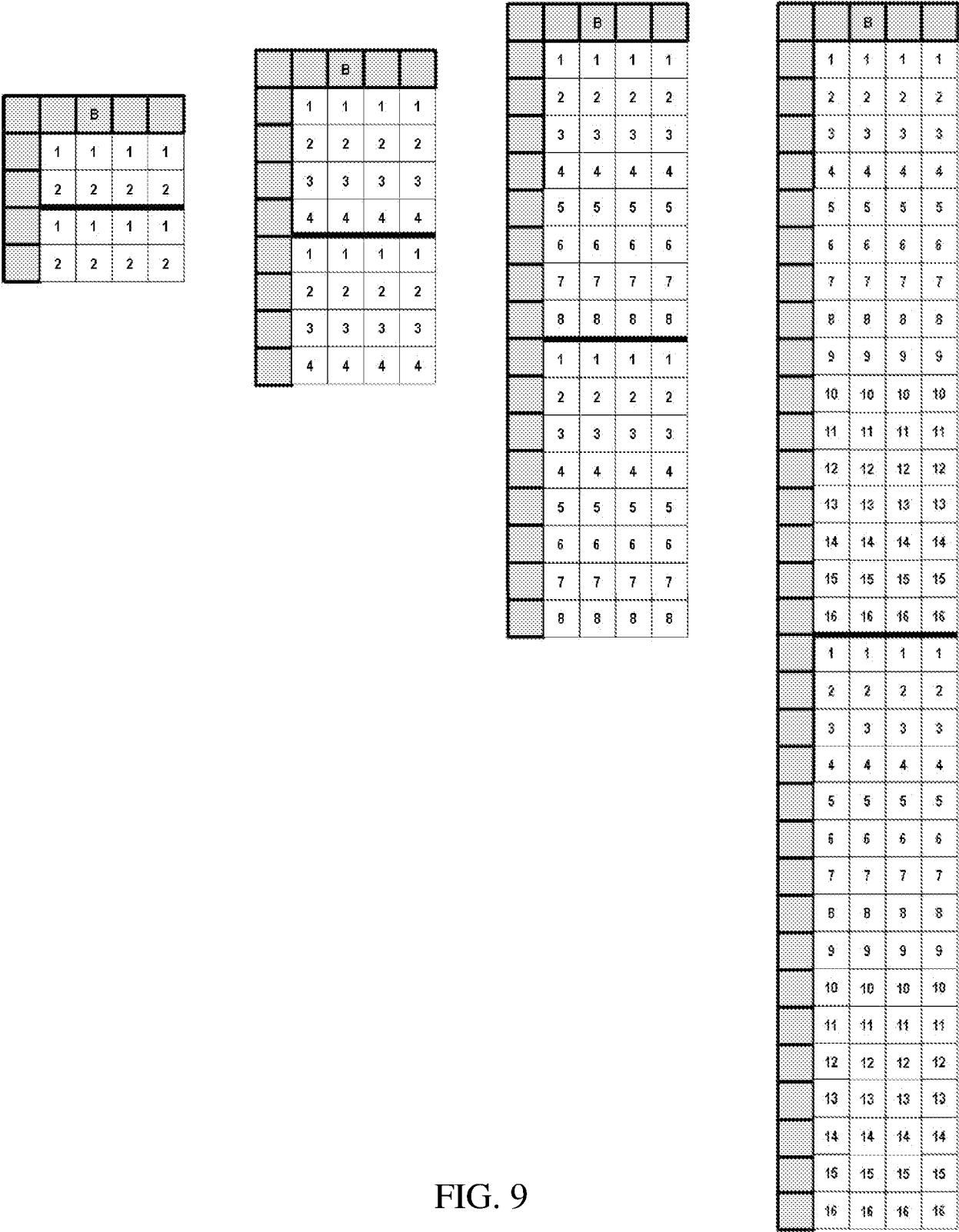


FIG. 9

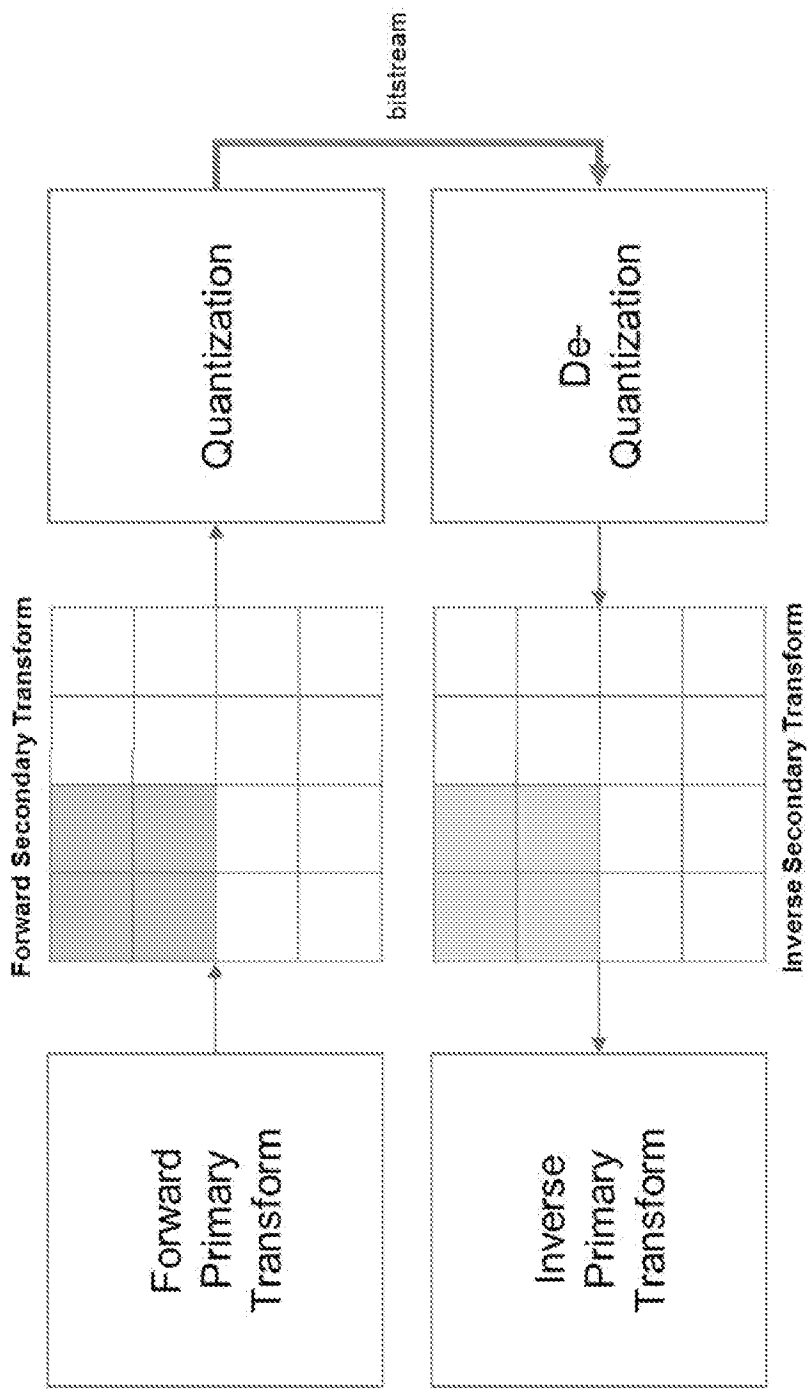


FIG. 10

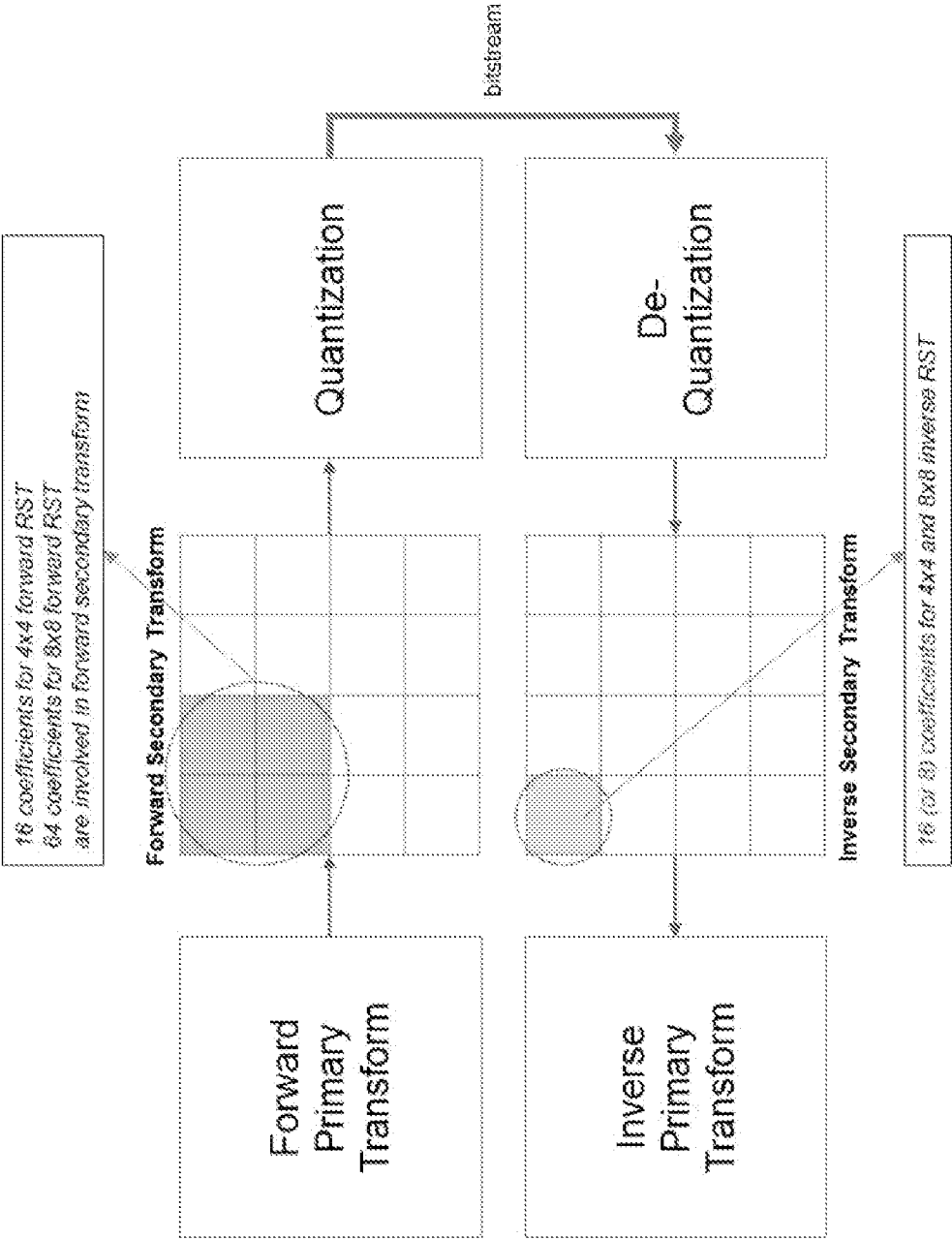


FIG. 11

12/17

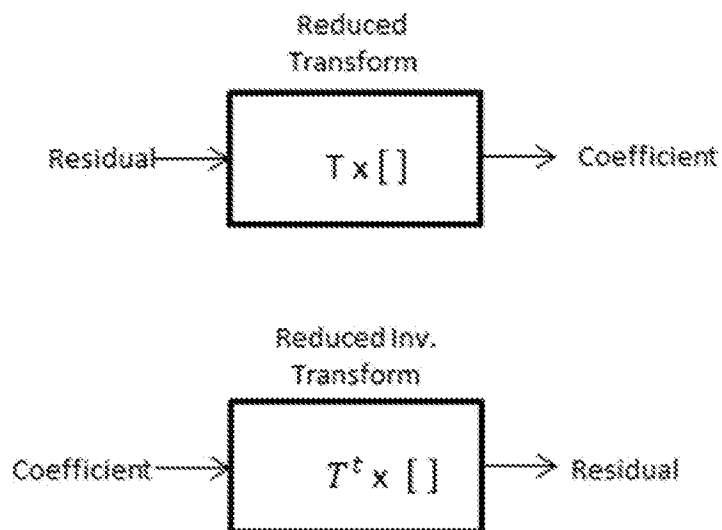


FIG. 12

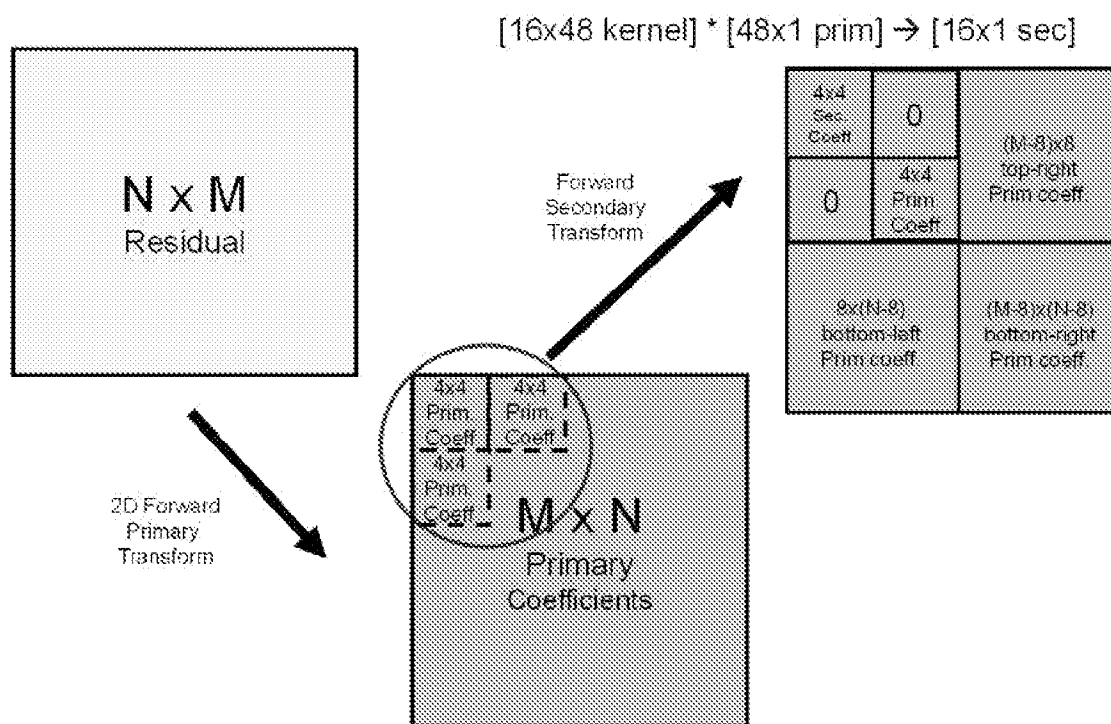
When $N \geq 8$ and $M \geq 8$,

FIG. 13

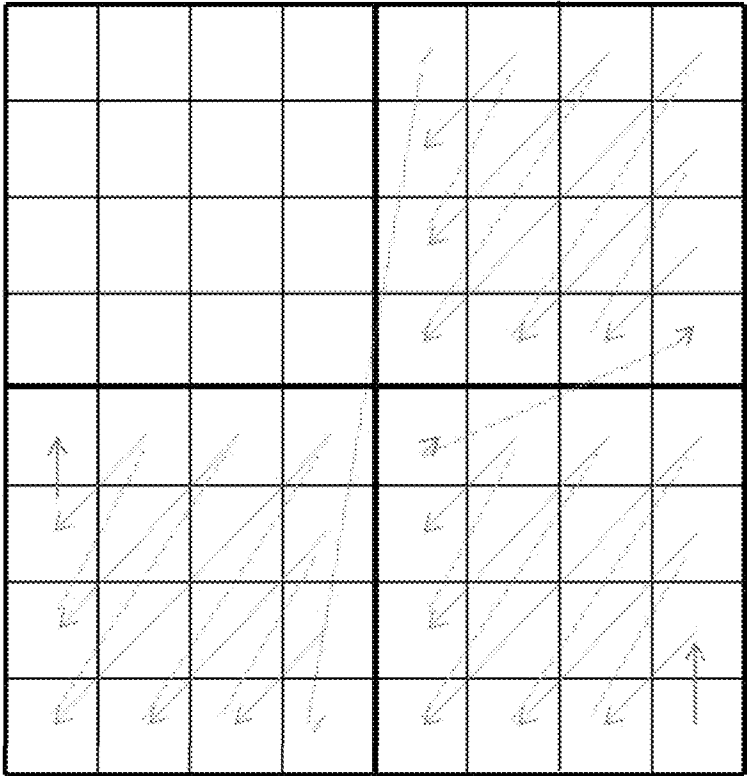


FIG. 14

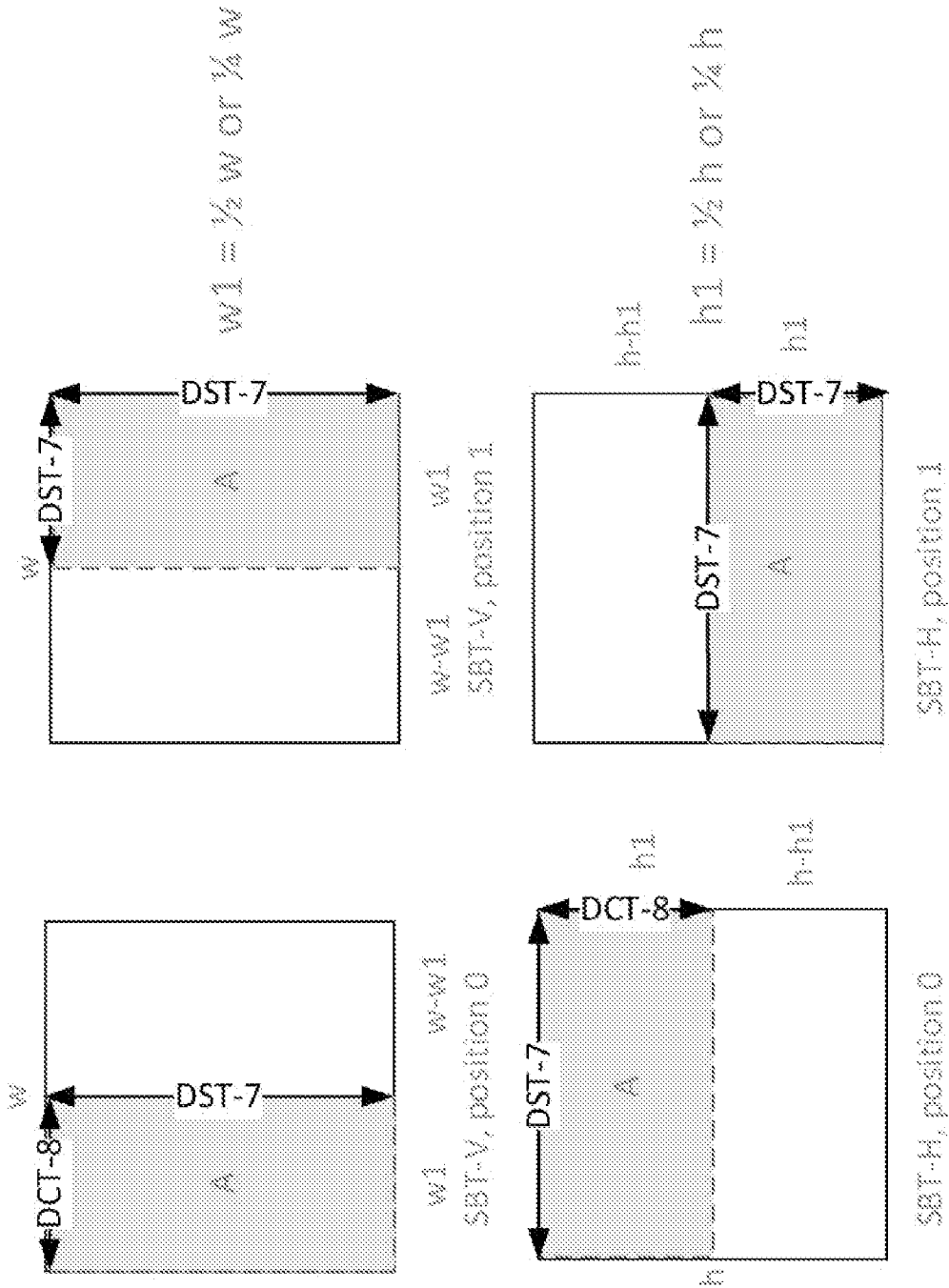


FIG. 15

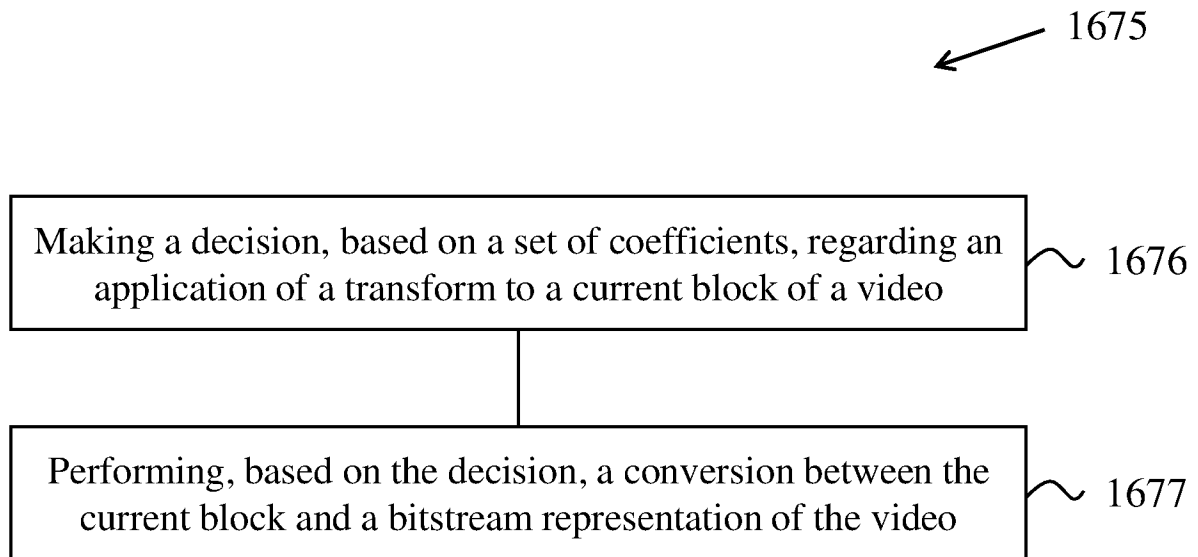


FIG. 16A

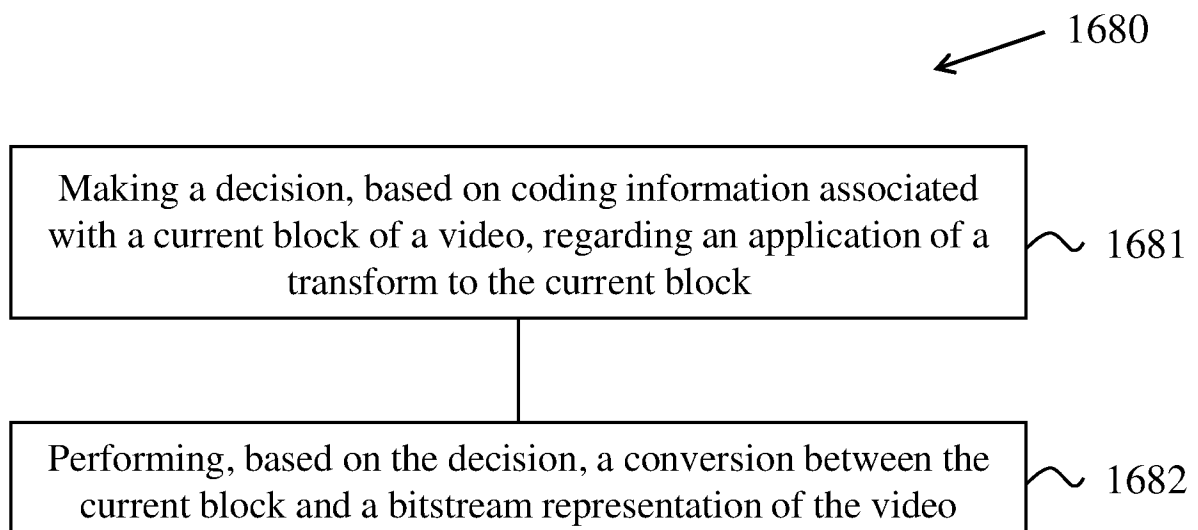


FIG. 16B

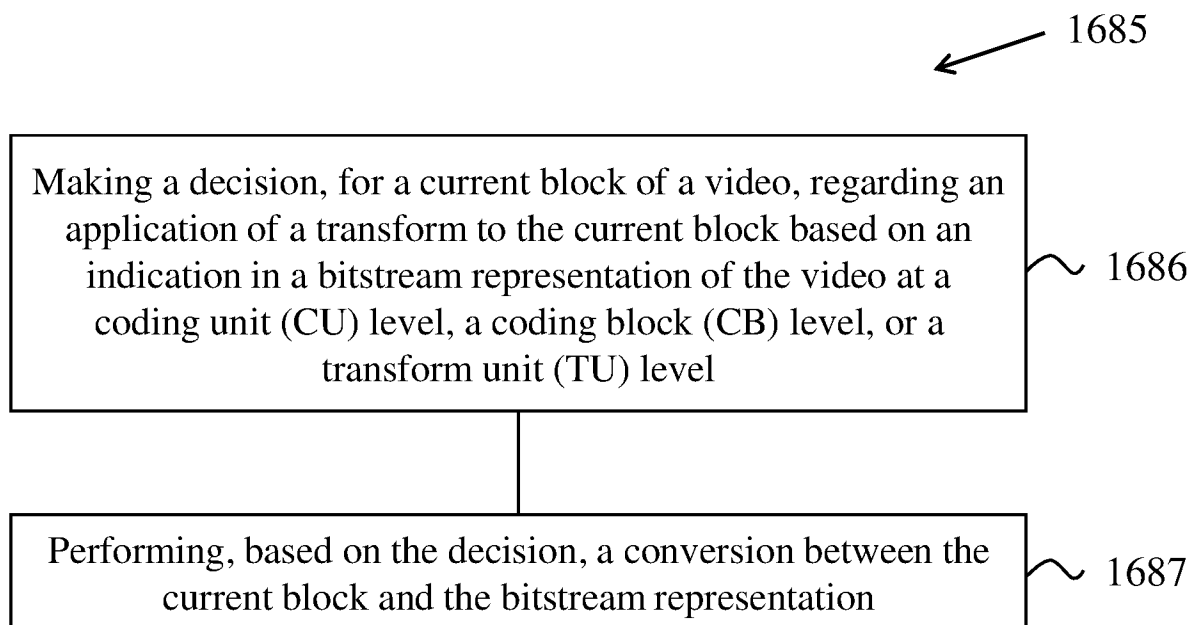


FIG. 16C

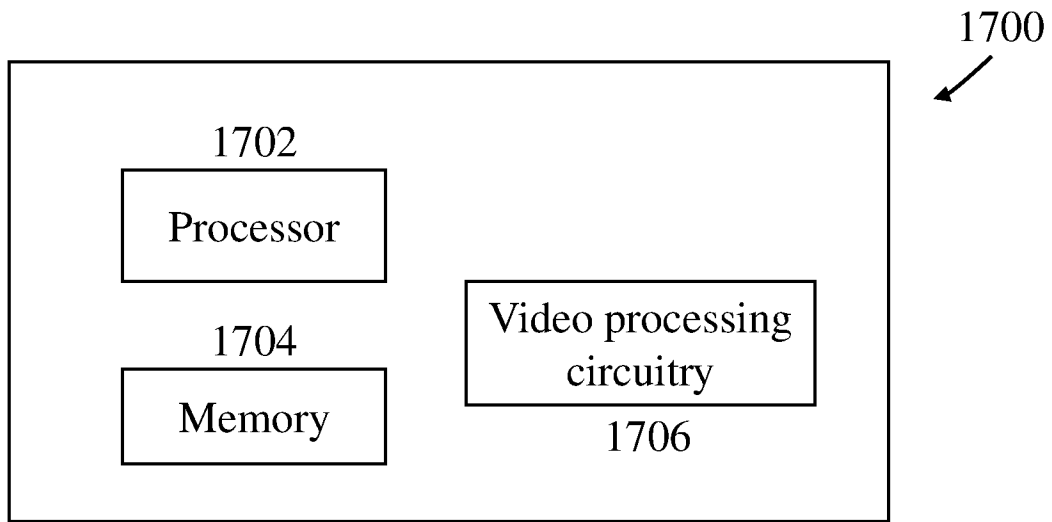


FIG. 17

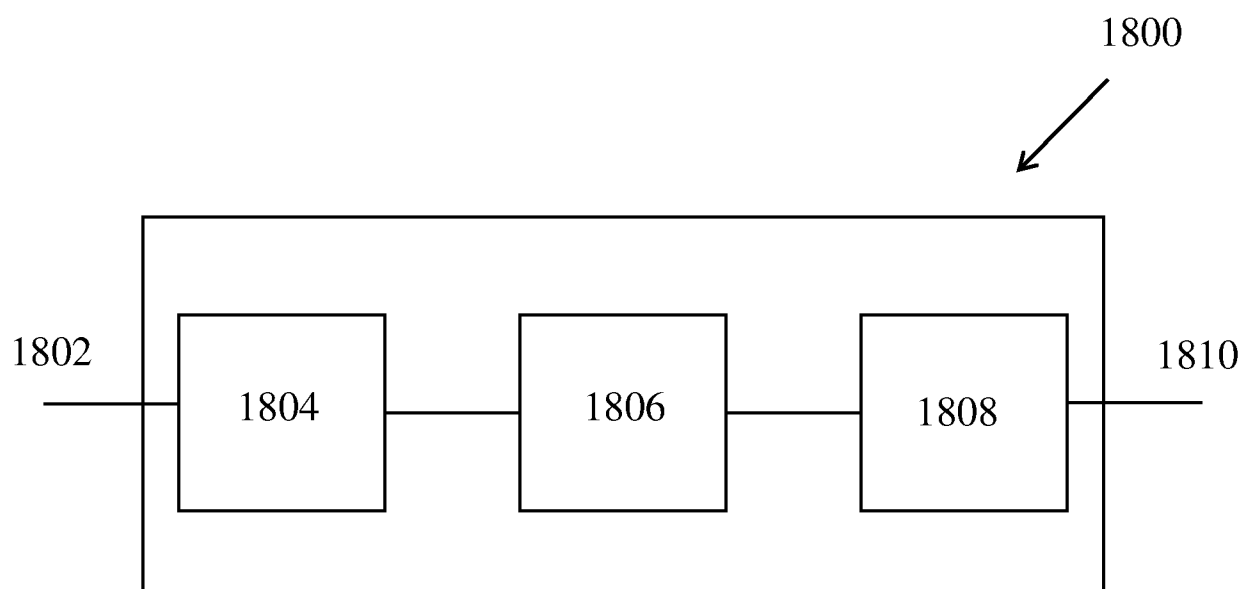


FIG. 18

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/094907

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/176(2014.01)i; H04N 19/103(2014.01)i; H04N 19/60(2014.01)i; H04N 19/184(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, EPODOC, WPI, CNKI, IEEE: video, process, coefficient, transform, block, current, conversion, bitstream, quantized, offset, LUT, coding, code, implicit, IST, DCT, dimension, size, frame, tree, IBC, transform unit, coding unit, CU, CB, TU, virtual pipeline, VPDU, CTU, slice, intra-coded

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2013266233 A1 (SONY CORPORATION) 10 October 2013 (2013-10-10) claims 9-20, description, paragraphs [0008]-[0012], [0209], [0219]-[0221], [0232], [0264]-[0268]	1-16, 53-56
X	US 2017188029 A1 (MEDIA TEK SINGAPORE PTE. LTD.) 29 June 2017 (2017-06-29) claims 1-15, description, paragraphs [0033]-[0039]	17-43, 53-56
X	WO 2017091776 A1 (QUALCOMM INCORPORATED) 01 June 2017 (2017-06-01) claims 1-14, description, paragraphs [0427]-[0429]	44-56
A	US 2014056347 A1 (MICROSOFT CORPORATION) 27 February 2014 (2014-02-27) the whole document	1-56
A	WO 2019076138 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 25 April 2019 (2019-04-25) the whole document	1-56



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

24 August 2020

Date of mailing of the international search report

09 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

Authorized officer

SHI, Shuang

Facsimile No. (86-10)62019451

Telephone No. 86-(10)-53961413

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2020/094907

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2013266233	A1	10 October 2013	JP	2012170042	A	06 September 2012
				US	2013301710	A1	14 November 2013
				JP	6056122	B2	11 January 2017
				AU	2019200515	A1	14 February 2019
				US	9560348	B2	31 January 2017
				AU	2018201382	B2	08 November 2018
				CN	103329536	B	15 February 2017
				AU	2016234951	A1	13 October 2016
				CN	110839154	A	25 February 2020
				TW	201242377	A	16 October 2012
				CN	103329536	A	25 September 2013
				AU	2012210001	B2	21 July 2016
				KR	20180053425	A	21 May 2018
				CN	107087190	B	21 April 2020
				MX	2013008289	A	13 September 2013
				CN	107087190	A	22 August 2017
				EP	2645717	A1	02 October 2013
				CN	106878738	B	08 November 2019
				TW	1495350	B	01 August 2015
				MY	158425	A	14 October 2016
				US	8787453	B2	22 July 2014
				MX	336094	B	08 January 2016
				AU	2020203010	A1	28 May 2020
				CN	106878738	A	20 June 2017
				CN	110602497	A	20 December 2019
				RU	2013132921	A	27 January 2015
				US	10419761	B2	17 September 2019
				AU	2012210001	A1	04 July 2013
				SG	191994	A1	30 August 2013
				US	2018063530	A1	01 March 2018
				WO	2012102088	A1	02 August 2012
				KR	20140029381	A	10 March 2014
				AU	2019200515	B2	06 February 2020
				KR	101965119	B1	02 April 2019
				CN	107087191	B	07 January 2020
				CA	2823024	A1	02 August 2012
				AU	2018201382	A1	15 March 2018
				AU	2016234951	B2	04 January 2018
				US	2019327471	A1	24 October 2019
				CN	107087191	A	22 August 2017
				KR	101858289	B1	15 May 2018
				EP	3512198	A1	17 July 2019
				ZA	201304818	B	25 September 2013
				CO	6731124	A2	15 August 2013
				JP	2017079485	A	27 April 2017
				JP	2018191334	A	29 November 2018
US	2017188029	A1	29 June 2017	EP	3130145	A1	15 February 2017
				WO	2015172694	A1	19 November 2015
				CN	106464878	B	21 May 2019
				CA	2948197	A1	19 November 2015

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2020/094907

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
WO	2017091776	A1	01 June 2017	CN	106464878	A	22 February 2017
				WO	2015172337	A1	19 November 2015
				US	10326990	B2	18 June 2019
WO	2017091776	A1	01 June 2017	AU	2016359155	A1	26 April 2018
				TW	201729592	A	16 August 2017
				WO	2017091759	A1	01 June 2017
WO	2017091776	A1	01 June 2017	CN	108293112	A	17 July 2018
				EP	3381186	A1	03 October 2018
				TW	201722156	A	16 June 2017
WO	2017091776	A1	01 June 2017	CN	108293119	A	17 July 2018
				JP	2018535609	A	29 November 2018
				US	2017150183	A1	25 May 2017
WO	2017091776	A1	01 June 2017	CN	108293137	A	17 July 2018
				BR	112018010664	A2	13 November 2018
				KR	20180084815	A	25 July 2018
WO	2017091776	A1	01 June 2017	JP	2018535610	A	29 November 2018
				BR	112018010534	A2	13 November 2018
				WO	2017091773	A1	01 June 2017
WO	2017091776	A1	01 June 2017	JP	2019500779	A	10 January 2019
				CN	108464002	A	28 August 2018
				KR	20180084813	A	25 July 2018
WO	2017091776	A1	01 June 2017	BR	112018010517	A2	13 November 2018
				US	2017150176	A1	25 May 2017
				US	2017150156	A1	25 May 2017
WO	2017091776	A1	01 June 2017	WO	2017091770	A1	01 June 2017
				TW	201729593	A	16 August 2017
				KR	20180084816	A	25 July 2018
WO	2017091776	A1	01 June 2017	AU	2016359165	A1	26 April 2018
				TW	201722157	A	16 June 2017
				KR	20180084814	A	25 July 2018
WO	2017091776	A1	01 June 2017	EP	3381189	A1	03 October 2018
				US	10200719	B2	05 February 2019
				JP	2019500778	A	10 January 2019
WO	2017091776	A1	01 June 2017	BR	112018010660	A2	13 November 2018
				AU	2016359168	A1	26 April 2018
				AU	2016359171	A1	10 May 2018
WO	2017091776	A1	01 June 2017	EP	3381191	A1	03 October 2018
				KR	102048169	B1	22 November 2019
				US	2017150186	A1	25 May 2017
WO	2017091776	A1	01 June 2017	EP	3381188	A1	03 October 2018
				US	10623776	B2	14 April 2020
				US	2017289575	A1	05 October 2017
WO	2017091776	A1	01 June 2017	US	9866868	B2	09 January 2018
				US	9866867	B2	09 January 2018
				US	2019306531	A1	03 October 2019
WO	2017091776	A1	01 June 2017	US	2018103270	A1	12 April 2018
				US	2017238018	A1	17 August 2017
				US	10298955	B2	21 May 2019
WO	2019076138	A1	25 April 2019	CN	111133760	A	08 May 2020
				WO	2019076206	A1	25 April 2019

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2020/094907

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
		CN 111226438 A	02 June 2020
		WO 2019076201 A1	25 April 2019
		KR 20200053602 A	18 May 2020
		EP 3677032 A1	08 July 2020
		WO 2019076200 A1	25 April 2019