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

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

(43) **International Publication Date**
17 September 2020 (17.09.2020)

WIPO | PCT

(10) **International Publication Number**
WO 2020/185943 A1

(51) **International Patent Classification:**
H04N 19/593 (2014.01) *H04N 19/50* (2014.01)
H04N 19/12 (2014.01) *H04N 19/176* (2014.01)

(21) **International Application Number:**
PCT/US2020/022163

(22) **International Filing Date:**
11 March 2020 (11.03.2020)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
62/816,855 11 March 2019 (11.03.2019) US
16/816,011 11 March 2020 (11.03.2020) US

(71) **Applicant: TENCENT AMERICA LLC** [US/US]; 2747 Park Boulevard, Palo Alto, California 94306 (US).

(72) **Inventors: ZHAO, Liang**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **LI, Xiang**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **ZHAO, Xin**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **LIU, Shan**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US).

(74) **Agent: MA, Johnny** et al.; Oblon, McClelland, Maier & Neustadt, L.L.P., 1940 Duke Street, Alexandria, Virginia 22314 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, 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,

(54) **Title:** IMPROVEMENT FOR INTER PDPC MODE

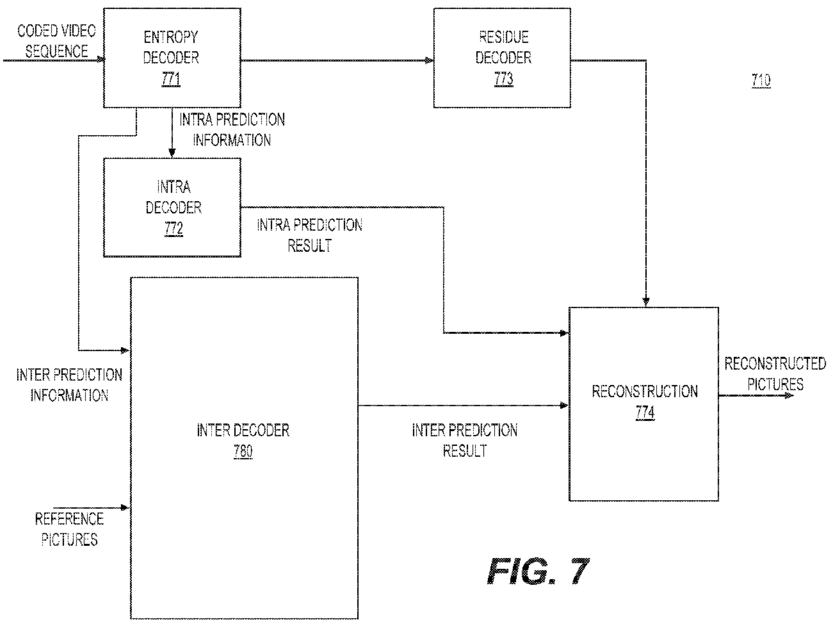


FIG. 7

(57) **Abstract:** Aspects of the disclosure provide methods and apparatuses for video encoding/decoding. An apparatus for video decoding includes processing circuitry that decodes prediction information for a current block in a current picture that is a part of a coded video sequence. The prediction information indicates a specific inter prediction mode, a position dependent prediction combination (PDPC) process, and a coded residual for the current block. The processing circuitry determines at least one of (i) a partition of the current block to which a transform process is to be applied and (ii) a transform type of the transform process for the current block based on the prediction information. The processing circuitry performs the transform process on the coded residual to generate a decoded residual. The processing circuitry reconstructs the current block based on the decoded residual.

WO 2020/185943 A1

[Continued on next page]

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).

Published:

— *with international search report (Art. 21(3))*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

IMPROVEMENT FOR INTER PDPC MODE

INCORPORATION BY REFERENCE

[0001] This present application claims the benefit of priority to U.S. Patent Application No. 16/816,011, "IMPROVEMENT FOR INTER PDPC MODE" filed on March 11, 2020, which claims the benefit of priority to U.S. Provisional Application No. 62/816,855, "IMPROVEMENT FOR INTER PDPC MODE" filed on March 11, 2019. The entire disclosures of the prior applications are hereby incorporated by reference in their entirety.

TECHNICAL FIELD

[0002] The present disclosure describes embodiments generally related to video coding.

BACKGROUND

[0003] The background description provided herein is for the purpose of generally presenting the context of the disclosure. Work of the presently named inventors, to the extent the work is described in this background section, as well as aspects of the description that may not otherwise qualify as prior art at the time of filing, are neither expressly nor impliedly admitted as prior art against the present disclosure.

[0004] Video coding and decoding can be performed using inter-picture prediction with motion compensation. Uncompressed digital video can include a series of pictures, each picture having a spatial dimension of, for example, 1920 x 1080 luminance samples and associated chrominance samples. The series of pictures can have a fixed or variable picture rate (informally also known as frame rate) of, for example, 60 pictures per second or 60 Hz. Uncompressed video has significant bitrate requirements. For example, 1080p60 4:2:0 video at 8 bit per sample (1920x1080 luminance sample resolution at 60 Hz frame rate) requires close to 1.5 Gbit/s bandwidth. An hour of such video requires more than 600 GBytes of storage space.

[0005] One purpose of video coding and decoding can be the reduction of redundancy in the input video signal, through compression. Compression can help reduce the aforementioned bandwidth or storage space requirements, in some cases by two orders of magnitude or more. Both lossless and lossy compression, as well as a combination thereof can be employed. Lossless compression refers to techniques where an exact copy of the original signal can be reconstructed from the compressed original signal. When using lossy compression, the reconstructed signal may not be identical to the original signal, but the distortion between original and reconstructed signals is small enough to make the reconstructed signal useful for the intended application. In the case of video, lossy compression is widely employed. The amount of distortion tolerated depends on the application; for example, users of certain consumer streaming applications may tolerate higher distortion than users of television distribution applications. The compression ratio achievable can reflect that: higher allowable/tolerable distortion can yield higher compression ratios.

[0006] A video encoder and decoder can utilize techniques from several broad categories, including, for example, motion compensation, transform, quantization, and entropy coding.

[0007] Video codec technologies can include techniques known as intra coding. In intra coding, sample values are represented without reference to samples or other data from previously reconstructed reference pictures. In some video codecs, the picture is spatially subdivided into blocks of samples. When all blocks of samples are coded in intra mode, that picture can be an intra picture. Intra pictures and their derivations such as independent decoder refresh pictures, can be used to reset the decoder state and can, therefore, be used as the first picture in a coded video bitstream and a video session, or as a still image. The samples of an intra block can be exposed to a transform, and the transform coefficients can be quantized before entropy coding. Intra prediction can be a technique that minimizes sample values in the pre-transform domain. In some cases, the smaller the DC value after a transform is, and the smaller the AC coefficients are, the fewer the bits that are required at a given quantization step size to represent the block after entropy coding.

[0008] Traditional intra coding such as known from, for example MPEG-2 generation coding technologies, does not use intra prediction. However, some newer video compression technologies include techniques that attempt, from, for example, surrounding sample data and/or metadata obtained during the encoding/decoding of spatially neighboring, and preceding in decoding order, blocks of data. Such techniques are henceforth called "intra prediction" techniques. Note that in at least some cases, intra prediction is only using reference data from the current picture under reconstruction and not from reference pictures.

[0009] There can be many different forms of intra prediction. When more than one of such techniques can be used in a given video coding technology, the technique in use can be coded in an intra prediction mode. In certain cases, modes can have submodes and/or parameters, and those can be coded individually or included in the mode codeword. Which codeword to use for a given mode/submode/parameter combination can have an impact in the coding efficiency gain through intra prediction, and so can the entropy coding technology used to translate the codewords into a bitstream.

[0010] A certain mode of intra prediction was introduced with H.264, refined in H.265, and further refined in newer coding technologies such as joint exploration model (JEM), versatile video coding (VVC), and benchmark set (BMS). A predictor block can be formed using neighboring sample values belonging to already available samples. Sample values of neighboring samples are copied into the predictor block according to a direction. A reference to the direction in use can be coded in the bitstream or may be predicted itself.

[0011] Referring to FIG. 1A, depicted in the lower right is a subset of nine predictor directions known from H.265's 33 possible predictor directions (corresponding to the 33 angular modes of the 35 intra modes). The point where the arrows converge (101) represents the sample being predicted. The arrows represent the direction from which the sample is being predicted. For example, arrow (102) indicates that sample (101) is predicted from a sample or samples to the upper right, at a 45 degree angle from the horizontal. Similarly, arrow (103) indicates that sample (101) is predicted from a sample or samples to the lower left of sample (101), in a 22.5 degree angle from the horizontal.

[0012] Still referring to FIG. 1A, on the top left there is depicted a square block (104) of 4 x 4 samples (indicated by a dashed, boldface line). The square block (104) includes 16 samples, each labelled with an "S", its position in the Y dimension (e.g., row index) and its position in the X dimension (e.g., column index). For example, sample S21 is the second sample in the Y dimension (from the top) and the first (from the left) sample in the X dimension. Similarly, sample S44 is the fourth sample in block (104) in both the Y and X dimensions. As the block is 4 x 4 samples in size, S44 is at the bottom right. Further shown are reference samples that follow a similar numbering scheme. A reference sample is labelled with an R, its Y position (e.g., row index) and X position (column index) relative to block (104). In both H.264 and H.265, prediction samples neighbor the block under reconstruction; therefore no negative values need to be used.

[0013] Intra picture prediction can work by copying reference sample values from the neighboring samples as appropriated by the signaled prediction direction. For example, assume the coded video bitstream includes signaling that, for this block, indicates a prediction direction consistent with arrow (102)—that is, samples are predicted from a prediction sample or samples to the upper right, at a 45 degree angle from the horizontal. In that case, samples S41, S32, S23, and S14 are predicted from the same reference sample R05. Sample S44 is then predicted from reference sample R08.

[0014] In certain cases, the values of multiple reference samples may be combined, for example through interpolation, in order to calculate a reference sample; especially when the directions are not evenly divisible by 45 degrees.

[0015] The number of possible directions has increased as video coding technology has developed. In H.264 (year 2003), nine different direction could be represented. That increased to 33 in H.265 (year 2013), and JEM/VVC/BMS, at the time of disclosure, can support up to 65 directions. Experiments have been conducted to identify the most likely directions, and certain techniques in the entropy coding are used to represent those likely directions in a small number of bits, accepting a certain penalty for less likely directions. Further, the directions themselves can sometimes be predicted from neighboring directions used in neighboring, already decoded, blocks.

[0016] FIG. 1B shows a schematic (105) that depicts 65 intra prediction directions according to JEM to illustrate the increasing number of prediction directions over time.

[0017] The mapping of intra prediction directions bits in the coded video bitstream that represent the direction can be different from video coding technology to video coding technology; and can range, for example, from simple direct mappings of prediction direction to intra prediction mode, to codewords, to complex adaptive schemes involving most probable modes, and similar techniques. In all cases, however, there can be certain directions that are statistically less likely to occur in video content than certain other directions. As the goal of video compression is the reduction of redundancy, those less likely directions will, in a well working video coding technology, be represented by a larger number of bits than more likely directions.

[0018] Motion compensation can be a lossy compression technique and can relate to techniques where a block of sample data from a previously reconstructed picture or part thereof (reference picture), after being spatially shifted in a direction indicated by a motion vector (MV henceforth), is used for the prediction of a newly reconstructed picture or picture part. In some cases, the reference picture can be the same as the picture currently under reconstruction. MVs can have two dimensions X and Y, or three dimensions, the third being an indication of the reference picture in use (the latter, indirectly, can be a time dimension).

[0019] In some video compression techniques, an MV applicable to a certain area of sample data can be predicted from other MVs, for example from those related to another area of sample data spatially adjacent to the area under reconstruction, and preceding that MV in decoding order. Doing so can substantially reduce the amount of data required for coding the MV, thereby removing redundancy and increasing compression. MV prediction can work effectively, for example, because when coding an input video signal derived from a camera (known as natural video) there is a statistical likelihood that areas larger than the area to which a single MV is applicable move in a similar direction and, therefore, can in some cases be predicted using a similar motion vector derived from MVs of a neighboring area. That results in the MV found for a given area to be similar or the same as the MV predicted from the surrounding MVs, and that in turn can be represented, after entropy coding, in a smaller number of bits than what would be used if coding the MV directly. In some cases, MV prediction can be an example of lossless compression of a signal (namely: the MVs) derived from the original signal (namely: the sample stream). In other cases, MV prediction itself can be lossy, for example because of rounding errors when calculating a predictor from several surrounding MVs.

[0020] Various MV prediction mechanisms are described in H.265/HEVC (ITU-T Rec. H.265, "High Efficiency Video Coding", December 2016). Out of the many MV prediction mechanisms that H.265 offers, described herein is a technique henceforth referred to as "spatial merge."

[0021] Referring to FIG. 1C, a current block (111) can include samples that have been found by the encoder during the motion search process to be predictable from a previous block of the same size that has been spatially shifted. Instead of coding that MV directly, the MV can be derived from metadata associated with one or more reference pictures, for example from the most recent (in decoding order) reference picture, using the MV associated with either one of five surrounding samples, denoted A0, A1, and B0, B1, B2 (112 through 116, respectively). In H.265, the MV prediction can use predictors from the same reference picture that the neighboring block is using.

SUMMARY

[0022] Aspects of the disclosure provide methods and apparatuses for video encoding/decoding. In some examples, an apparatus for video decoding includes processing circuitry.

[0023] According to aspects of the disclosure, there is provided a method for video decoding in a decoder. In the method, the processing circuitry decodes prediction information for a current block in a current picture that is a part of a coded video sequence. The prediction information indicates a specific inter prediction mode, a position dependent prediction combination (PDPC) process, and a coded residual for the current block. The processing circuitry determines at least one of (i) a partition of the current block to which a transform process is to be applied and (ii) a transform type of the transform process for the current block based on the prediction information. The processing circuitry performs the transform process on the coded residual to generate a decoded residual. The processing circuitry reconstructs the current block based on the decoded residual.

[0024] According to aspects of the disclosure, the processing circuitry determines the transform type to be one of a plurality of sub-block transform (SBT) types. Each of the plurality of SBT types is applied to a different part of the current block. In an embodiment, application of the one of the plurality of SBT types is independent of the specific inter prediction mode of the current block. In an embodiment, the part that the one of the plurality of SBT types is applied to is one of a rightmost part and a bottommost part of a plurality of parts of the current block.

[0025] According to aspects of the disclosure, the processing circuitry determines a subset of a DCT-2 transform, a DST-7 transform, and a DCT-8 transform from which the transform type is selected.

[0026] In an embodiment, the prediction information includes a signaled flag that is (i) context coded and (ii) indicates the PDPC process for the current block.

[0027] In an embodiment, an intra block copy (IBC) mode is not permitted for the current block.

[0028] In an embodiment, a plurality of weighting parameters of the PDPC process is based on prediction information of a neighboring block adjacent to the current block.

[0029] Aspects of the disclosure also provide a non-transitory computer-readable medium storing instructions which when executed by a computer for video decoding cause the computer to perform any one or a combination of the methods for video decoding.

BRIEF DESCRIPTION OF THE DRAWINGS

[0030] Further features, the nature, and various advantages of the disclosed subject matter will be more apparent from the following detailed description and the accompanying drawings in which:

[0031] FIG. 1A is a schematic illustration of an exemplary subset of intra prediction modes;

[0032] FIG. 1B is an illustration of exemplary intra prediction directions;

[0033] FIG. 1C is a schematic illustration of a current block and its surrounding spatial merge candidates in one example;

[0034] FIG. 2 is a schematic illustration of a simplified block diagram of a communication system in accordance with an embodiment;

[0035] FIG. 3 is a schematic illustration of a simplified block diagram of a communication system in accordance with an embodiment;

[0036] FIG. 4 is a schematic illustration of a simplified block diagram of a decoder in accordance with an embodiment;

[0037] FIG. 5 is a schematic illustration of a simplified block diagram of an encoder in accordance with an embodiment;

[0038] FIG. 6 shows a block diagram of an encoder in accordance with another embodiment;

[0039] FIG. 7 shows a block diagram of a decoder in accordance with another embodiment;

[0040] FIG. 8 shows an illustration of exemplary intra prediction directions and corresponding intra prediction modes in some examples (e.g., VVC);

[0041] FIG. 9A shows exemplary weighting factors for a prediction sample at (0, 0) in DC mode in accordance with an embodiment;

[0042] FIG. 9B shows exemplary weighting factors for a prediction sample at (1, 0) in DC mode in accordance with an embodiment;

[0043] FIG. 10 shows a table that illustrates an exemplary intra mode coding of a chroma block in some examples (e.g., VVC);

[0044] FIG. 11A shows the diagonal split of a CU in accordance with an embodiment;

[0045] FIG. 11B shows the anti-diagonal split of a CU in accordance with an embodiment;

[0046] FIGs. 12A – 12D show exemplary sub-block transform modes in accordance with some embodiments;

[0047] FIG. 13 shows an exemplary table of the basis functions of the selected DST/DCT;

[0048] FIG. 14 shows an exemplary table of the horizontal and vertical transform types depending on an multiple transform selection (MTS) index;

[0049] FIG. 15 shows a flow chart outlining an exemplary process in accordance with an embodiment; and

[0050] FIG. 16 is a schematic illustration of a computer system in accordance with an embodiment.

DETAILED DESCRIPTION OF EMBODIMENTS

[0051] **Video encoder and decoder**

[0052] FIG. 2 illustrates a simplified block diagram of a communication system (200) according to an embodiment of the present disclosure. The communication system (200) includes a plurality of terminal devices that can communicate with each other, via, for example, a network (250). For example, the communication system (200) includes a first pair of terminal devices (210) and (220) interconnected via the network (250). In the FIG. 2 example, the first pair of terminal devices (210) and (220) performs unidirectional transmission of data. For example, the terminal device (210) may code video data (e.g., a stream of video pictures that are captured by the terminal device (210)) for transmission to the other terminal device (220) via the network (250). The encoded video data can be transmitted in the form of one or more coded video bitstreams. The terminal device (220) may receive the coded video data from the network (250), decode the coded video data to recover the video pictures and display video pictures according to the recovered video data. Unidirectional data transmission may be common in media serving applications and the like.

[0053] In another example, the communication system (200) includes a second pair of terminal devices (230) and (240) that performs bidirectional transmission of coded video data that may occur, for example, during videoconferencing. For bidirectional transmission of data, in an example, each terminal device of the terminal devices (230) and (240) may code video data (e.g., a stream of video pictures that are captured by the terminal device) for transmission to the other terminal device of the terminal devices (230) and (240) via the network (250). Each terminal device of the terminal devices (230) and (240) also may receive the coded video data transmitted by the other terminal device of the terminal devices (230) and (240), and may decode the coded video data to recover the video pictures and may display video pictures at an accessible display device according to the recovered video data.

[0054] In the FIG. 2 example, the terminal devices (210), (220), (230) and (240) may be illustrated as servers, personal computers and smart phones but the principles of the present disclosure may be not so limited. Embodiments of the present disclosure find application with laptop computers, tablet computers, media players and/or dedicated video conferencing equipment. The network (250) represents any number of networks that convey coded video data among the terminal devices (210), (220), (230) and (240), including for example wireline (wired) and/or wireless communication networks. The communication network (250) may exchange data in circuit-switched and/or packet-switched channels. Representative networks include telecommunications networks, local area networks, wide area networks and/or the Internet. For the purposes of the present discussion, the architecture and topology of the network (250) may be immaterial to the operation of the present disclosure unless explained herein below.

[0055] FIG. 3 illustrates, as an example for an application for the disclosed subject matter, the placement of a video encoder and a video decoder in a streaming environment. The disclosed subject matter can be equally applicable to other video enabled applications, including, for example, video conferencing, digital TV, storing of compressed video on digital media including CD, DVD, memory stick, and the like.

[0056] A streaming system may include a capture subsystem (313) that can include a video source (301), for example a digital camera, creating for example a stream of video pictures (302) that are uncompressed. In an example, the stream of video pictures (302) includes samples that are taken by the digital camera. The stream of video pictures (302), depicted as a bold line to emphasize a high data volume when compared to encoded video data (304) (or coded video bitstreams), can be processed by an electronic device (320) that includes a video encoder (303) coupled to the video source (301). The video encoder (303) can include hardware, software, or a combination thereof to enable or implement aspects of the disclosed subject matter as described in more detail below. The encoded video data (304) (or encoded video bitstream (304)), depicted as a thin line to emphasize the lower data volume when compared to the stream of video pictures (302), can be stored on a streaming server (305) for future use. One or more streaming client subsystems, such as client subsystems (306) and (308) in FIG. 3 can access the streaming server (305) to retrieve copies (307) and (309) of the encoded video data (304). A client subsystem (306) can include a video decoder (310), for example, in an electronic device (330). The video decoder (310) decodes the incoming copy (307) of the encoded video data and creates an outgoing stream of video pictures (311) that can be rendered on a display (312) (e.g., display screen) or other rendering device (not depicted). In some streaming systems, the encoded video data (304), (307), and (309) (e.g., video bitstreams) can be encoded according to certain video coding/compression standards. Examples of those standards include ITU-T Recommendation H.265. In an example, a video coding standard under development is informally known as Versatile Video Coding (VVC). The disclosed subject matter may be used in the context of VVC.

[0057] It is noted that the electronic devices (320) and (330) can include other components (not shown). For example, the electronic device (320) can include a video decoder (not shown) and the electronic device (330) can include a video encoder (not shown) as well.

[0058] FIG. 4 shows a block diagram of a video decoder (410) according to an embodiment of the present disclosure. The video decoder (410) can be included in an electronic device (430). The electronic device (430) can include a receiver (431) (e.g., receiving circuitry). The video decoder (410) can be used in the place of the video decoder (310) in the FIG. 3 example.

[0059] The receiver (431) may receive one or more coded video sequences to be decoded by the video decoder (410); in the same or another embodiment, one coded video sequence at a time, where the decoding of each coded video sequence is independent from other coded video sequences. The coded video sequence may be received from a channel (401), which may be a hardware/software link to a storage device which stores the encoded video data. The receiver (431) may receive the encoded video data with other data, for example, coded audio data and/or ancillary data streams, that may be forwarded to their respective using entities (not depicted). The receiver (431) may separate the coded video sequence from the other data. To combat network jitter, a buffer memory (415) may be coupled in between the receiver (431) and an entropy decoder / parser (420) ("parser (420)" henceforth). In certain applications, the buffer memory (415) is part of the video decoder (410). In others, it can be outside of the video decoder (410) (not depicted). In still others, there can be a buffer memory (not depicted) outside of the video decoder (410), for example to combat network jitter, and in addition another buffer memory (415) inside the video decoder (410), for example to handle playout timing. When the receiver (431) is receiving data from a store/forward device of sufficient bandwidth and controllability, or from an isosynchronous network, the buffer memory (415) may not be needed, or can be small. For use on best effort packet networks such as the Internet, the buffer memory (415) may be required, can be comparatively large and can be advantageously of adaptive size, and may at least partially be implemented in an operating system or similar elements (not depicted) outside of the video decoder (410).

[0060] The video decoder (410) may include the parser (420) to reconstruct symbols (421) from the coded video sequence. Categories of those symbols include information used to manage operation of the video decoder (410), and potentially information to control a rendering device such as a render device (412) (e.g., a display screen) that is not an integral part of the electronic device (430) but can be coupled to the electronic device (430), as was shown in FIG. 4. The control information for the rendering device(s) may be in the form of Supplemental Enhancement Information (SEI messages) or Video Usability Information (VUI) parameter set fragments (not depicted). The parser (420) may parse / entropy-decode the coded video sequence that is received. The coding of the coded video sequence can be in accordance with a video coding technology or standard, and can follow various principles, including variable length coding, Huffman coding, arithmetic coding with or without context sensitivity, and so forth. The parser (420) may extract from the coded video sequence, a set of subgroup parameters for at least one of the subgroups of pixels in the video decoder, based upon at least one parameter corresponding to the group. Subgroups can include Groups of Pictures (GOPs), pictures, tiles, slices, macroblocks, Coding Units (CUs), blocks, Transform Units (TUs), Prediction Units (PUs) and so forth. The parser (420) may also extract from the coded video sequence information such as transform coefficients, quantizer parameter values, motion vectors, and so forth.

[0061] The parser (420) may perform an entropy decoding / parsing operation on the video sequence received from the buffer memory (415), so as to create symbols (421).

[0062] Reconstruction of the symbols (421) can involve multiple different units depending on the type of the coded video picture or parts thereof (such as: inter and intra picture, inter and intra block), and other factors. Which units are involved, and how, can be controlled by the subgroup control information that was parsed from the coded video sequence by the parser (420). The flow of such subgroup control information between the parser (420) and the multiple units below is not depicted for clarity.

[0063] Beyond the functional blocks already mentioned, the video decoder (410) can be conceptually subdivided into a number of functional units as described below. In a practical implementation operating under commercial constraints, many of these units interact closely with each other and can, at least partly, be integrated into each other. However, for the purpose of describing the disclosed subject matter, the conceptual subdivision into the functional units below is appropriate.

[0064] A first unit is the scaler / inverse transform unit (451). The scaler / inverse transform unit (451) receives a quantized transform coefficient as well as control information, including which transform to use, block size, quantization factor, quantization scaling matrices, etc. as symbol(s) (421) from the parser (420). The scaler / inverse transform unit (451) can output blocks comprising sample values that can be input into aggregator (455).

[0065] In some cases, the output samples of the scaler / inverse transform (451) can pertain to an intra coded block; that is: a block that is not using predictive information from previously reconstructed pictures, but can use predictive information from previously reconstructed parts of the current picture. Such predictive information can be provided by an intra picture prediction unit (452). In some cases, the intra picture prediction unit (452) generates a block of the same size and shape of the block under reconstruction, using surrounding already reconstructed information fetched from the current picture buffer (458). The current picture buffer (458) buffers, for example, partly reconstructed current picture and/or fully reconstructed current picture. The aggregator (455), in some cases, adds, on a per sample basis, the prediction information that the intra prediction unit (452) has generated to the output sample information as provided by the scaler / inverse transform unit (451).

[0066] In other cases, the output samples of the scaler / inverse transform unit (451) can pertain to an inter coded, and potentially motion compensated block. In such a case, a motion compensation prediction unit (453) can access reference picture memory (457) to fetch samples used for prediction. After motion compensating the fetched samples in accordance with the symbols (421) pertaining to the block, these samples can be added by the aggregator (455) to the output of the scaler / inverse transform unit (451) (in this case called the residual samples or residual signal) so as to generate output sample information. The addresses within the reference picture memory (457) from where the motion compensation prediction unit (453) fetches prediction samples can be controlled by motion vectors, available to the motion compensation prediction unit (453) in the form of symbols (421) that can have, for example X, Y, and reference picture components. Motion compensation also can include interpolation of sample values as fetched from the reference picture memory (457) when sub-sample exact motion vectors are in use, motion vector prediction mechanisms, and so forth.

[0067] The output samples of the aggregator (455) can be subject to various loop filtering techniques in the loop filter unit (456). Video compression technologies can include in-loop filter technologies that are controlled by parameters included in the coded video sequence (also referred to as coded video bitstream) and made available to the loop filter unit (456) as symbols (421) from the parser (420), but can also be responsive to meta-information obtained during the decoding of previous (in decoding order) parts of the coded picture or coded video sequence, as well as responsive to previously reconstructed and loop-filtered sample values.

[0068] The output of the loop filter unit (456) can be a sample stream that can be output to the render device (412) as well as stored in the reference picture memory (457) for use in future inter-picture prediction.

[0069] Certain coded pictures, once fully reconstructed, can be used as reference pictures for future prediction. For example, once a coded picture corresponding to a current picture is fully reconstructed and the coded picture has been identified as a reference picture (by, for example, the parser (420)), the current picture buffer (458) can become a part of the reference picture memory (457), and a fresh current picture buffer can be reallocated before commencing the reconstruction of the following coded picture.

[0070] The video decoder (410) may perform decoding operations according to a predetermined video compression technology in a standard, such as ITU-T Rec. H.265. The coded video sequence may conform to a syntax specified by the video compression technology or standard being used, in the sense that the coded video sequence adheres to both the syntax of the video compression technology or standard and the profiles as documented in the video compression technology or standard. Specifically, a profile can select certain tools as the only tools available for use under that profile from all the tools available in the video compression technology or standard. Also necessary for compliance can be that the complexity of the coded video sequence is within bounds as defined by the level of the video compression technology or standard. In some cases, levels restrict the maximum picture size, maximum frame rate, maximum reconstruction sample rate (measured in, for example megasamples per second), maximum reference picture size, and so on. Limits set by levels can, in some cases, be further restricted through Hypothetical Reference Decoder (HRD) specifications and metadata for HRD buffer management signaled in the coded video sequence.

[0071] In an embodiment, the receiver (431) may receive additional (redundant) data with the encoded video. The additional data may be included as part of the coded video sequence(s). The additional data may be used by the video decoder (410) to properly decode the data and/or to more accurately reconstruct the original video data. Additional data can be in the form of, for example, temporal, spatial, or signal noise ratio (SNR) enhancement layers, redundant slices, redundant pictures, forward error correction codes, and so on.

[0072] FIG. 5 shows a block diagram of a video encoder (503) according to an embodiment of the present disclosure. The video encoder (503) is included in an electronic device (520). The electronic device (520) includes a transmitter (540) (e.g., transmitting circuitry). The video encoder (503) can be used in the place of the video encoder (303) in the FIG. 3 example.

[0073] The video encoder (503) may receive video samples from a video source (501) (that is not part of the electronic device (520) in the FIG. 5 example) that may capture video image(s) to be coded by the video encoder (503). In another example, the video source (501) is a part of the electronic device (520).

[0074] The video source (501) may provide the source video sequence to be coded by the video encoder (503) in the form of a digital video sample stream that can be of any suitable bit depth (for example: 8 bit, 10 bit, 12 bit, ...), any colorspace (for example, BT.601 Y CrCB, RGB, ...), and any suitable sampling structure (for example Y CrCb 4:2:0, Y CrCb 4:4:4). In a media serving system, the video source (501) may be a storage device storing previously prepared video. In a videoconferencing system, the video source (501) may be a camera that captures local image information as a video sequence. Video data may be provided as a plurality of individual pictures that impart motion when viewed in sequence. The pictures themselves may be organized as a spatial array of pixels, wherein each pixel can comprise one or more samples depending on the sampling structure, color space, etc. in use. A person skilled in the art can readily understand the relationship between pixels and samples. The description below focuses on samples.

[0075] According to an embodiment, the video encoder (503) may code and compress the pictures of the source video sequence into a coded video sequence (543) in real time or under any other time constraints as required by the application. Enforcing appropriate coding speed is one function of a controller (550). In some embodiments, the controller (550) controls other functional units as described below and is functionally coupled to the other functional units. The coupling is not depicted for clarity. Parameters set by the controller (550) can include rate control related parameters (picture skip, quantizer, lambda value of rate-distortion optimization techniques, ...), picture size, group of pictures (GOP) layout, maximum motion vector allowed reference area, and so forth. The controller (550) can be configured to have other suitable functions that pertain to the video encoder (503) optimized for a certain system design.

[0076] In some embodiments, the video encoder (503) is configured to operate in a coding loop. As an oversimplified description, in an example, the coding loop can include a source coder (530) (e.g., responsible for creating symbols, such as a symbol stream, based on an input picture to be coded, and a reference picture(s)), and a (local) decoder (533) embedded in the video encoder (503). The decoder (533) reconstructs the symbols to create the sample data in a similar manner as a (remote) decoder also would create (as any compression between symbols and coded video bitstream is lossless in the video compression technologies considered in the disclosed subject matter). The reconstructed sample stream (sample data) is input to the reference picture memory (534). As the decoding of a symbol stream leads to bit-exact results independent of decoder location (local or remote), the content in the reference picture memory (534) is also bit exact between the local encoder and remote encoder. In other words, the prediction part of an encoder "sees" as reference picture samples exactly the same sample values as a decoder would "see" when using prediction during decoding. This fundamental principle of reference picture synchronicity (and resulting drift, if synchronicity cannot be maintained, for example because of channel errors) is used in some related arts as well.

[0077] The operation of the "local" decoder (533) can be the same as of a "remote" decoder, such as the video decoder (410), which has already been described in detail above in conjunction with FIG. 4. Briefly referring also to FIG. 4, however, as symbols are available and encoding/decoding of symbols to a coded video sequence by an entropy coder (545) and the parser (420) can be lossless, the entropy decoding parts of the video decoder (410), including the buffer memory (415) and the parser (420) may not be fully implemented in the local decoder (533).

[0078] An observation that can be made at this point is that any decoder technology except the parsing/entropy decoding that is present in a decoder also necessarily needs to be present, in substantially identical functional form, in a corresponding encoder. For this reason, the disclosed subject matter focuses on decoder operation. The description of encoder technologies can be abbreviated as they are the inverse of the comprehensively described decoder technologies. Only in certain areas a more detail description is required and provided below.

[0079] During operation, in some examples, the source coder (530) may perform motion compensated predictive coding, which codes an input picture predictively with reference to one or more previously-coded picture from the video sequence that were designated as "reference pictures". In this manner, the coding engine (532) codes differences between pixel blocks of an input picture and pixel blocks of reference picture(s) that may be selected as prediction reference(s) to the input picture.

[0080] The local video decoder (533) may decode coded video data of pictures that may be designated as reference pictures, based on symbols created by the source coder (530). Operations of the coding engine (532) may advantageously be lossy processes. When the coded video data may be decoded at a video decoder (not shown in FIG. 5), the reconstructed video sequence typically may be a replica of the source video sequence with some errors. The local video decoder (533) replicates decoding processes that may be performed by the video decoder on reference pictures and may cause reconstructed reference pictures to be stored in the reference picture cache (534). In this manner, the video encoder (503) may store copies of reconstructed reference pictures locally that have common content as the reconstructed reference pictures that will be obtained by a far-end video decoder (absent transmission errors).

[0081] The predictor (535) may perform prediction searches for the coding engine (532). That is, for a new picture to be coded, the predictor (535) may search the reference picture memory (534) for sample data (as candidate reference pixel blocks) or certain metadata such as reference picture motion vectors, block shapes, and so on, that may serve as an appropriate prediction reference for the new pictures. The predictor (535) may operate on a sample block-by-pixel block basis to find appropriate prediction references. In some cases, as determined by search results obtained by the predictor (535), an input picture may have prediction references drawn from multiple reference pictures stored in the reference picture memory (534).

[0082] The controller (550) may manage coding operations of the source coder (530), including, for example, setting of parameters and subgroup parameters used for encoding the video data.

[0083] Output of all aforementioned functional units may be subjected to entropy coding in the entropy coder (545). The entropy coder (545) translates the symbols as generated by the various functional units into a coded video sequence, by lossless compressing the symbols according to technologies such as Huffman coding, variable length coding, arithmetic coding, and so forth.

[0084] The transmitter (540) may buffer the coded video sequence(s) as created by the entropy coder (545) to prepare for transmission via a communication channel (560), which may be a hardware/software link to a storage device which would store the encoded video data. The transmitter (540) may merge coded video data from the video coder (503) with other data to be transmitted, for example, coded audio data and/or ancillary data streams (sources not shown).

[0085] The controller (550) may manage operation of the video encoder (503). During coding, the controller (550) may assign to each coded picture a certain coded picture type, which may affect the coding techniques that may be applied to the respective picture. For example, pictures often may be assigned as one of the following picture types:

[0086] An Intra Picture (I picture) may be one that may be coded and decoded without using any other picture in the sequence as a source of prediction. Some video codecs allow for different types of intra pictures, including, for example Independent Decoder Refresh ("IDR") Pictures. A person skilled in the art is aware of those variants of I pictures and their respective applications and features.

[0087] A predictive picture (P picture) may be one that may be coded and decoded using intra prediction or inter prediction using at most one motion vector and reference index to predict the sample values of each block.

[0088] A bi-directionally predictive picture (B Picture) may be one that may be coded and decoded using intra prediction or inter prediction using at most two motion vectors and reference indices to predict the sample values of each block. Similarly, multiple-predictive pictures can use more than two reference pictures and associated metadata for the reconstruction of a single block.

[0089] Source pictures commonly may be subdivided spatially into a plurality of sample blocks (for example, blocks of 4x4, 8x8, 4x8, or 16x16 samples each) and coded on a block-by-block basis. Blocks may be coded predictively with reference to other (already coded) blocks as determined by the coding assignment applied to the blocks' respective pictures. For example, blocks of I pictures may be coded non-predictively or they may be coded predictively with reference to already coded blocks of the same picture (spatial prediction or intra prediction). Pixel blocks of P pictures may be coded predictively, via spatial prediction or via temporal prediction with reference to one previously coded reference picture. Blocks of B pictures may be coded predictively, via spatial prediction or via temporal prediction with reference to one or two previously coded reference pictures.

[0090] The video encoder (503) may perform coding operations according to a predetermined video coding technology or standard, such as ITU-T Rec. H.265. In its operation, the video encoder (503) may perform various compression operations, including predictive coding operations that exploit temporal and spatial redundancies in the input video sequence. The coded video data, therefore, may conform to a syntax specified by the video coding technology or standard being used.

[0091] In an embodiment, the transmitter (540) may transmit additional data with the encoded video. The source coder (530) may include such data as part of the coded video sequence. Additional data may comprise temporal/spatial/SNR enhancement layers, other forms of redundant data such as redundant pictures and slices, SEI messages, VUI parameter set fragments, and so on.

[0092] A video may be captured as a plurality of source pictures (video pictures) in a temporal sequence. Intra-picture prediction (often abbreviated to intra prediction) makes use of spatial correlation in a given picture, and inter-picture prediction makes uses of the (temporal or other) correlation between the pictures. In an example, a specific picture under encoding/decoding, which is referred to as a current picture, is partitioned into blocks. When a block in the current picture is similar to a reference block in a previously coded and still buffered reference picture in the video, the block in the current picture can be coded by a vector that is referred to as a motion vector. The motion vector points to the reference block in the reference picture, and can have a third dimension identifying the reference picture, in case multiple reference pictures are in use.

[0093] In some embodiments, a bi-prediction technique can be used in the inter-picture prediction. According to the bi-prediction technique, two reference pictures, such as a first reference picture and a second reference picture that are both prior in decoding order to the current picture in the video (but may be in the past and future, respectively, in display order) are used. A block in the current picture can be coded by a first motion vector that points to a first reference block in the first reference picture, and a second motion vector that points to a second reference block in the second reference picture. The block can be predicted by a combination of the first reference block and the second reference block.

[0094] Further, a merge mode technique can be used in the inter-picture prediction to improve coding efficiency.

[0095] According to some embodiments of the disclosure, predictions, such as inter-picture predictions and intra-picture predictions are performed in the unit of blocks. For example, according to the HEVC standard, a picture in a sequence of video pictures is partitioned into coding tree units (CTU) for compression, the CTUs in a picture have the same size, such as 64x64 pixels, 32x32 pixels, or 16x16 pixels. In general, a CTU includes three coding tree blocks (CTBs), which are one luma CTB and two chroma CTBs. Each CTU can be recursively quad-tree split into one or multiple coding units (CUs). For example, a CTU of 64x64 pixels can be split into one CU of 64x64 pixels, or 4 CUs of 32x32 pixels, or 16 CUs of 16x16 pixels. In an example, each CU is analyzed to determine a prediction type for the CU, such as an inter prediction type or an intra prediction type. The CU is split into one or more prediction units (PUs) depending on the temporal and/or spatial predictability. Generally, each PU includes a luma prediction block (PB), and two chroma PBs. In an embodiment, a prediction operation in coding (encoding/decoding) is performed in the unit of a prediction block. Using a luma prediction block as an example of a prediction block, the prediction block includes a matrix of values (e.g., luma values) for pixels, such as 8x8 pixels, 16x16 pixels, 8x16 pixels, 16x8 pixels, and the like.

[0096] FIG. 6 shows a diagram of a video encoder (603) according to another embodiment of the disclosure. The video encoder (603) is configured to receive a processing block (e.g., a prediction block) of sample values within a current video picture in a sequence of video pictures, and encode the processing block into a coded picture that is part of a coded video sequence. In an example, the video encoder (603) is used in the place of the video encoder (303) in the FIG. 3 example.

[0097] In an HEVC example, the video encoder (603) receives a matrix of sample values for a processing block, such as a prediction block of 8x8 samples, and the like. The video encoder (603) determines whether the processing block is best coded using intra mode, inter mode, or bi-prediction mode using, for example, rate-distortion optimization. When the processing block is to be coded in intra mode, the video encoder (603) may use an intra prediction technique to encode the processing block into the coded picture; and when the processing block is to be coded in inter mode or bi-prediction mode, the video encoder (603) may use an inter prediction or bi-prediction technique, respectively, to encode the processing block into the coded picture. In certain video coding technologies, merge mode can be an inter picture prediction submode where the motion vector is derived from one or more motion vector predictors without the benefit of a coded motion vector component outside the predictors. In certain other video coding technologies, a motion vector component applicable to the subject block may be present. In an example, the video encoder (603) includes other components, such as a mode decision module (not shown) to determine the mode of the processing blocks.

[0098] In the FIG. 6 example, the video encoder (603) includes the inter encoder (630), an intra encoder (622), a residue calculator (623), a switch (626), a residue encoder (624), a general controller (621), and an entropy encoder (625) coupled together as shown in FIG. 6.

[0099] The inter encoder (630) is configured to receive the samples of the current block (e.g., a processing block), compare the block to one or more reference blocks in reference pictures (e.g., blocks in previous pictures and later pictures), generate inter prediction information (e.g., description of redundant information according to inter encoding technique, motion vectors, merge mode information), and calculate inter prediction results (e.g., predicted block) based on the inter prediction information using any suitable technique. In some examples, the reference pictures are decoded reference pictures that are decoded based on the encoded video information.

[0100] The intra encoder (622) is configured to receive the samples of the current block (e.g., a processing block), in some cases compare the block to blocks already coded in the same picture, generate quantized coefficients after transform, and in some cases also intra prediction information (e.g., an intra prediction direction information according to one or more intra encoding techniques). In an example, the intra encoder (622) also calculates intra prediction results (e.g., predicted block) based on the intra prediction information and reference blocks in the same picture.

[0101] The general controller (621) is configured to determine general control data and control other components of the video encoder (603) based on the general control data. In an example, the general controller (621) determines the mode of the block, and provides a control signal to the switch (626) based on the mode. For example, when the mode is the intra mode, the general controller (621) controls the switch (626) to select the intra mode result for use by the residue calculator (623), and controls the entropy encoder (625) to select the intra prediction information and include the intra prediction information in the bitstream; and when the mode is the inter mode, the general controller (621) controls the switch (626) to select the inter prediction result for use by the residue calculator (623), and controls the entropy encoder (625) to select the inter prediction information and include the inter prediction information in the bitstream.

[0102] The residue calculator (623) is configured to calculate a difference (residue data) between the received block and prediction results selected from the intra encoder (622) or the inter encoder (630). The residue encoder (624) is configured to operate based on the residue data to encode the residue data to generate the transform coefficients. In an example, the residue encoder (624) is configured to convert the residue data from a spatial domain to a frequency domain, and generate the transform coefficients. The transform coefficients are then subject to quantization processing to obtain quantized transform coefficients. In various embodiments, the video encoder (603) also includes a residue decoder (628). The residue decoder (628) is configured to perform inverse-transform, and generate the decoded residue data. The decoded residue data can be suitably used by the intra encoder (622) and the inter encoder (630). For example, the inter encoder (630) can generate decoded blocks based on the decoded residue data and inter prediction information, and the intra encoder (622) can generate decoded blocks based on the decoded residue data and the intra prediction information. The decoded blocks are suitably processed to generate decoded pictures and the decoded pictures can be buffered in a memory circuit (not shown) and used as reference pictures in some examples.

[0103] The entropy encoder (625) is configured to format the bitstream to include the encoded block. The entropy encoder (625) is configured to include various information according to a suitable standard, such as the HEVC standard. In an example, the entropy encoder (625) is configured to include the general control data, the selected prediction information (e.g., intra prediction information or inter prediction information), the residue information, and other suitable information in the bitstream. Note that, according to the disclosed subject matter, when coding a block in the merge submode of either inter mode or bi-prediction mode, there is no residue information.

[0104] FIG. 7 shows a diagram of a video decoder (710) according to another embodiment of the disclosure. The video decoder (710) is configured to receive coded pictures that are part of a coded video sequence, and decode the coded pictures to generate reconstructed pictures. In an example, the video decoder (710) is used in the place of the video decoder (310) in the FIG. 3 example.

[0105] In the FIG. 7 example, the video decoder (710) includes an entropy decoder (771), an inter decoder (780), a residue decoder (773), a reconstruction module (774), and an intra decoder (772) coupled together as shown in FIG. 7.

[0106] The entropy decoder (771) can be configured to reconstruct, from the coded picture, certain symbols that represent the syntax elements of which the coded picture is made up. Such symbols can include, for example, the mode in which a block is coded (such as, for example, intra mode, inter mode, bi-predicted mode, the latter two in merge submode or another submode), prediction information (such as, for example, intra prediction information or inter prediction information) that can identify certain sample or metadata that is used for prediction by the intra decoder (772) or the inter decoder (780), respectively, residual information in the form of, for example, quantized transform coefficients, and the like. In an example, when the prediction mode is inter or bi-predicted mode, the inter prediction information is provided to the inter decoder (780); and when the prediction type is the intra prediction type, the intra prediction information is provided to the intra decoder (772). The residual information can be subject to inverse quantization and is provided to the residue decoder (773).

[0107] The inter decoder (780) is configured to receive the inter prediction information, and generate inter prediction results based on the inter prediction information.

[0108] The intra decoder (772) is configured to receive the intra prediction information, and generate prediction results based on the intra prediction information.

[0109] The residue decoder (773) is configured to perform inverse quantization to extract de-quantized transform coefficients, and process the de-quantized transform coefficients to convert the residual from the frequency domain to the spatial domain. The residue decoder (773) may also require certain control information (to include the Quantizer Parameter (QP)), and that information may be provided by the entropy decoder (771) (data path not depicted as this may be low volume control information only).

[0110] The reconstruction module (774) is configured to combine, in the spatial domain, the residual as output by the residue decoder (773) and the prediction results (as output by the inter or intra prediction modules as the case may be) to form a reconstructed block, that may be part of the reconstructed picture, which in turn may be part of the reconstructed video. It is noted that other suitable operations, such as a deblocking operation and the like, can be performed to improve the visual quality.

[0111] It is noted that the video encoders (303), (503), and (603), and the video decoders (310), (410), and (710) can be implemented using any suitable technique. In an embodiment, the video encoders (303), (503), and (603), and the video decoders (310), (410), and (710) can be implemented using one or more integrated circuits. In another embodiment, the video encoders (303), (503), and (603), and the video decoders (310), (410), and (710) can be implemented using one or more processors that execute software instructions.

[0112] **Intra prediction in VVC**

[0113] FIG. 8 shows an illustration of exemplary intra prediction directions and corresponding intra prediction modes in some examples (e.g., VVC). In FIG. 8, there are a total of 95 intra prediction modes (mode -14 to mode 80), among which mode 18 is a horizontal mode, mode 50 is a vertical mode, and mode 2, mode 34, and mode 66 are diagonal modes. Modes -1 ~ -14 and Modes 67 ~ 80 are referred to as wide-angle intra prediction (WAIP) modes.

[0114] **Position dependent prediction combination (PDPC) filtering process**

[0115] According to aspects of the disclosure, position dependent prediction combination (PDPC) is applied to the following intra modes without signaling: planar, DC, WAIP modes, horizontal, vertical, bottom-left angular mode (mode 2) and its 8 adjacent angular modes (mode

3 to mode10), and top-right angular mode (mode 66) and its 8 adjacent angular modes (mode 58 to mode 65).

[0116] In an embodiment, a prediction sample $pred'[x][y]$ located at a position (x, y) in a current block is predicted using an intra prediction mode (e.g., DC, planar, or an angular mode) and a linear combination of reference samples according to Eq. 1:

$$pred'[x][y] = (wL \times R(-1, y) + wT \times R(x, -1) - wTL \times R(-1, -1) + (64 - wL - wT + wTL) \times pred[x][y] + 32) \gg 6 \quad \text{(Eq. 1)}$$

where $pred[x][y]$ is an intra prediction value resulted from the intra prediction mode, $R(x, -1)$ represents a (unfiltered) reference sample that is located at a top reference line of the current sample (x, y) and has the same horizontal coordinate as the current sample (x, y), $R(-1, y)$ represents a (unfiltered) reference sample that is located at a left reference line of the current sample (x, y) and has the same vertical coordinate as the current sample (x, y), $R(-1, -1)$ represents a reference sample located at a top-left corner of the current block, and wT, wL, and wTL denote weighting factors.

[0117] In an embodiment, when the intra prediction mode is DC mode, the weight factors are calculated by Eq. 2 - Eq. 5:

$$wT = 32 \gg ((y \ll 1) \gg nScale) \quad \text{(Eq. 2)}$$

$$wL = 32 \gg ((x \ll 1) \gg nScale) \quad \text{(Eq. 3)}$$

$$wTL = +(wL \gg 4) + (wT \gg 4) \quad \text{(Eq. 4)}$$

$$nScale = (log2(width) + log2(height) - 2) \gg 2 \quad \text{(Eq. 5)}$$

where wT denotes the weighting factor for the reference sample $(x, -1)$, wL denotes the weighting factor for the reference sample $(-1, y)$, and wTL denotes the weighting factor for the top-left reference sample $(-1, -1)$, $nScale$ (referred to as weighting factor decrement rate) specifies how fast these weighting factors decrease along an axis (e.g., wL decreases from left to right along the x-axis, or wT decreases from top to bottom along the y-axis). The constant 32 in Eq. 2 and Eq. 3 denotes an initial weighting factor of a neighboring sample (e.g., a top neighboring sample, a left neighboring sample, or a top-left neighboring sample). The initial weighting factor is also assigned to a top-left sample of the current block. The weighting factors of neighboring samples in the PDPC filtering process are equal to or less than the initial weighting factor.

[0118] In an embodiment, when the intra prediction mode is planar mode, $wTL = 0$; when the intra prediction mode is horizontal mode, $wTL = wT$; and when the intra prediction mode is vertical mode, $wTL = wL$. The PDPC weighting factors can be calculated with add operations and shift operations. The value of $pred'[x][y]$ can be computed in a single step using Eq. 1.

[0119] FIG. 9A shows exemplary weighting factors for a prediction sample at (0, 0) in DC mode. In the FIG. 9A example, the current block is a 4x4 block (width = height = 4), thus $nScale$ is 0. Then, wT is 32, wL is 32, and wTL is 4.

[0120] FIG. 9B shows exemplary weighting factors for a prediction sample at (1, 0) in DC mode. In the FIG. 9B example, the current block is a 4x4 block (width = height = 4), thus $nScale$ is 0. Then, wT is 32, wL is 8, and wTL is 2.

[0121] In some embodiments, when the PDPC filtering process is applied to DC, planar, horizontal, and vertical intra modes, additional boundary filters are not needed, such as the HEVC DC mode boundary filter or horizontal/vertical mode edge filters.

[0122] In some embodiments, inputs of the PDPC filtering process includes:

- the intra prediction mode that is represented by $preModeIntra$;
- the width of the current block that is represented $nTbW$;
- the height of the current block that is represented by $nTbH$;
- the width of the reference samples that is represented by $refW$;
- the height of the reference samples that is represented by $refH$;
- the predicted samples that are represented by $predSamples[x][y]$, with $x = 0..nTbW - 1$ and $y = 0..nTbH - 1$;
- the unfiltered reference (also referred to as neighboring) samples that are represented by $p[x][y]$, with $x = -1, y = -1..refH - 1$ and $x = 0..refW - 1, y = -1$; and
- the color component of the current block that is represented by $cIdx$.

[0123] Depending on the value of $cIdx$, the function $clip1Cmp$ is set as follows:

- If $cIdx$ is equal to 0, $clip1Cmp$ is set equal to $Clip1_Y$.
- Otherwise, $clip1Cmp$ is set equal to $Clip1_C$.

[0124] Further, outputs of the PDPC filtering process are the modified predicted samples $predSamples'[x][y]$ with $x = 0..nTbW - 1$ and $y = 0..nTbH - 1$.

[0125] Then, a scaling factor $nScale$ is calculated by Eq. 6:

$$nScale = ((\,Log2(\,nTbW\,) \;+\; Log2(\,nTbH\,) \;-\; 2\,) \;>>\; 2\,) \qquad \text{(Eq. 6)}$$

[0126] Further, in some embodiments, a reference sample array $mainRef[\,x\,]$ with $x = 0..refW$ is defined as the array of unfiltered reference samples above the current block and another reference sample array $sideRef[\,y\,]$ with $y = 0..refH$ is defined as the array of unfiltered reference samples to the left of the current block. The reference sample arrays $mainRef[\,x\,]$and $sideRef[\,y\,]$ can be derived from unfiltered reference samples according to Eq. 7 - Eq. 8, respectively:

$$mainRef[\,x\,] \;=\; p[\,x\,][\,-1\,] \qquad \text{(Eq. 7)}$$

$$sideRef[\,y\,] \;=\; p[\,-1\,][\,y\,] \qquad \text{(Eq. 8)}$$

[0127] For each location (x, y) in the current block, the PDPC calculation may use a reference sample at the top that is denoted as $refT[\,x\,][\,y\,]$, a reference sample at the left that is denoted as $refL[\,x\,][\,y\,]$, and a reference sample at the corner $p[-1,-1]$. In some examples, the modified predicted sample is calculated by Eq. 9, and the result is suitably clipped according to the variable $cIdx$ that is indicative of the color component.

$$predSamples'[x][y] = (wL \times refL(x,y) + wT \times refT(x,y) - wTL \times p(-1,-1) + (64 - wL - wT + wTL) \times predSamples[x][y] + 32) \gg 6 \qquad \text{(Eq.9)}$$

[0128] The reference samples $refT[\,x\,][\,y\,]$, $refL[\,x\,][\,y\,]$, and the weighting factors wL, wT, and wTL can be determined based on the intra prediction mode $preModeIntra$.

[0129] In an example, when the intra prediction mode $preModeIntra$ is equal to INTRA_PLANAR (e.g., 0, planar mode, or mode 0), INTRA_DC (e.g., 1, DC mode, or mode 1), INTRA_ ANGULAR18 (e.g., 18, horizontal mode, or mode 18), or INTRA_ ANGULAR50 (e.g., 50, vertical mode, or mode 50), reference samples $refT[\,x\,][\,y\,]$, $refL[\,x\,][\,y\,]$, and the weighting factors wL, wT, and wTL can be determined according to Eq. 10 - Eq. 14:

$$refL[\,x\,][\,y\,] \;=\; p[\,-1\,][\,y\,] \qquad \text{(Eq. 10)}$$

$$refT[\,x\,][\,y\,] \;=\; p[\,x\,][\,-1\,] \qquad \text{(Eq. 11)}$$

$$wT[\,y\,] \;=\; 32 \;>>\; ((\,y \;<<\; 1\,) \;>>\; nScale\,) \qquad \text{(Eq. 12)}$$

$$wL[\,x\,] \;=\; 32 \;>>\; ((\,x \;<<\; 1\,) \;>>\; nScale\,) \qquad \text{(Eq. 13)}$$

$wTL[\,x\,][\,y\,] = (\,predModeIntra\ = = \ INTRA\_DC\,)\ ?\ (\,(\,wL[\,x\,]\ >>\ 4\,) + (\,wT[\,y\,]\ >>\ 4\,)\,)\ :\ 0$ (Eq. 14)

[0130] In another example, when the intra prediction mode $preModeIntra$ is equal to INTRA_ANGULAR2 (e.g., 2, or mode 2) or INTRA_ANGULAR66 (e.g., 66, or mode 66), reference samples $refT[\,x\,][\,y\,]$, $refL[\,x\,][\,y\,]$, and the weighting factors wL, wT, and wTL can be determined according to Eq. 15 - Eq. 19:

$$refL[\,x\,][\,y\,] = p[\,-1\,][\,x+y+1\,] \quad \text{(Eq. 15)}$$

$$refT[\,x\,][\,y\,] = p[\,x+y+1\,][\,-1\,] \quad \text{(Eq. 16)}$$

$$wT[\,y\,] = 32\ >>\ (\,(\,y\ <<\ 1\,)\ >>\ nScale\,) \quad \text{(Eq. 17)}$$

$$wL[\,x\,] = 32\ >>\ (\,(\,x\ <<\ 1\,)\ >>\ nScale\,) \quad \text{(Eq. 18)}$$

$$wTL[\,x\,][\,y\,] = 0 \quad \text{(Eq. 19)}$$

[0131] In another example, when the intra prediction mode $preModeIntra$ is less than or equal to INTRA_ANGULAR10 (e.g., 10, or mode 10), for location (x, y), variables $dXPos[\,y\,]$, $dXFrac[\,y\,]$, $dXInt[\,y\,]$, and $dX[\,y\,]$ are derived based on a variable $invAngle$ that is a function of the intra prediction mode $preModeIntra$. In an example, the $invAngle$ can be determined based on a look-up table that stores a corresponding $invAngle$ value to each intra prediction mode, and then the reference samples $refT[\,x\,][\,y\,]$, $refL[\,x\,][\,y\,]$, and the weighting factors wL, wT, and wTL can be determined based on the variables $dXPos[\,y\,]$, $dXFrac[\,y\,]$, $dXInt[\,y\,]$, and $dX[\,y\,]$.

[0132] For example, the variables $dXPos[\,y\,]$, $dXFrac[\,y\,]$, $dXInt[\,y\,]$, and $dX[\,y\,]$ are determined according to Eq. 20 - Eq.23:

$$dXPos[\,y\,] = (\,(\,y\ +\ 1\,) \times invAngle\ +\ 2\,)\ >>\ 2 \quad \text{(Eq. 20)}$$

$$dXFrac[\,y\,] = dXPos[\,y\,]\ \&\ 63 \quad \text{(Eq. 21)}$$

$$dXInt[\,y\,] = dXPos\,[\,y\,]\ >>\ 6 \quad \text{(Eq. 22)}$$

$$dX[\,y\,] = x\ +\ dXInt[\,y\,] \quad \text{(Eq. 23)}$$

[0133] Then, the reference samples $refT[\,x\,][\,y\,]$, $refL[\,x\,][\,y\,]$, and the weighting factors wL, wT, and wTL may be determined according to Eq. 24 - Eq. 28.

$$refL[\,x\,][\,y\,] = 0 \quad \text{(Eq. 24)}$$

$refT[x][y] = (dX[y] < refW - 1)?\,((64 - dXFrac[y]) \times mainRef[dX[y]] + dXFrac[y] \times mainRef[dX[y] + 1] + 32) \gg 6 : 0$ (Eq. 25)

$$wT[\,y\,] = (\,dX[\,y\,] < refW - 1\,)\,?\ 32\ >> (\,(\,y << 1\,) >> nScale\,) : 0 \qquad \text{(Eq. 26)}$$

$$wL[\,x\,] = 0 \qquad \text{(Eq. 27)}$$

$$wTL[\,x\,][\,y\,] = 0 \qquad \text{(Eq. 28)}$$

[0134] In another example, when the intra prediction mode $preModeIntra$ is greater than or equal to INTRA_ANGULAR58 (e.g., 58, or mode 58), variables $dXPos[\,x\,]$, $dXFrac[\,x\,]$, $dXInt[\,x\,]$, and $dX[\,x\,]$ are derived based on a variable $invAngle$ that is a function of the intra prediction mode $preModeIntra$. In an example, the $invAngle$ can be determined based on a look-up table that stores a corresponding $invAngle$ value for each intra prediction mode, and then the reference samples $refT[\,x\,][\,y\,]$, $refL[\,x\,][\,y\,]$, and the weighting factors wL, wT, and wTL are determined based on the variables $dXPos[\,x\,]$, $dXFrac[\,x\,]$, $dXInt[\,x\,]$, and $dX[\,x\,]$.

[0135] For example, the variables $dXPos[\,x\,]$, $dXFrac[\,x\,]$, $dXInt[\,x\,]$, and $dX[\,x\,]$ are determined according to Eq. 29 - Eq.32:

$$dYPos[\,x\,] = (\,(\,x + 1\,) \times invAngle + 2\,) \gg 2 \qquad \text{(Eq. 29)}$$

$$dYFrac[\,x\,] = dYPos[\,x\,]\ \&\ 63 \qquad \text{(Eq. 30)}$$

$$dYInt[\,x\,] = dYPos\,[\,x\,] \gg 6 \qquad \text{(Eq. 31)}$$

$$dY[\,x\,] = x + dYInt[\,x\,] \qquad \text{(Eq. 32)}$$

[0136] Then, the reference samples $refT[\,x\,][\,y\,]$, $refL[\,x\,][\,y\,]$, and the weighting factors wL, wT, and wTL are determined according to Eq. 33 - Eq. 37.

$$refL[x][y] = (dY[x] < refH - 1)?\,((64 - dYFrac[x]) \times sideRef[dY[x]] + dYFrac[x] \times sideRef[dY[x] + 1] + 32) \gg 6 : 0 \qquad \text{(Eq. 33)}$$

$$refT[\,x\,][\,y\,] = 0 \qquad \text{(Eq. 34)}$$

$$wT[\,y\,] = 0 \qquad \text{(Eq. 35)}$$

$$wL[\,x\,] = (\,dY[x] < refH - 1\,)\ ?\ 32 \gg ((\,x \ll 1) \gg nScale\,)\ :\ 0 \qquad \text{(Eq. 36)}$$

$$wTL[\,x\,][\,y\,] = 0 \qquad \text{(Eq. 37)}$$

[0137] In some examples, when the variable $preModeIntra$ is between modes 11 - 57 and is not one of mode 18 and mode 50, then the $refT[\,x\,][\,y\,]$, $refL[\,x\,][\,y\,]$, and the weighting factors wL, wT, and wTL are all set equal to 0. Then, the values of the filtered samples $filtSamples[\,x\,][\,y\,]$, with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$ are derived according to Eq. 38:

$$filtSamples[x][y] = clip1Cmp((refL[x][y] \times wL + refT[x][y] \times wT - p[-1][-1] \times wTL[x][y] + (64 - wL[x] - wT[y] + wTL[x][y]) \times predSamples[x][y] + 32) >> 6) \quad \text{(Eq. 38)}$$

[0138] Chroma intra prediction modes

[0139] FIG. 10 shows a table that illustrates an exemplary intra mode coding of a chroma block in some examples (e.g., VVC). In FIG. 10, there are 5 intra modes for the intra mode coding of the chroma block: Planar mode (mode index 0), DC mode (mode index 1), Horizontal mode (mode index 18), Vertical mode (mode index 50), Diagonal mode (mode index 66), and Derived mode (DM) which is a direct copy of an intra prediction mode of a luma component associated with the chroma component.

[0140] In an embodiment, to avoid a duplicate mode, the four modes other than DM are assigned according to the intra prediction mode of the associated luma component. For example, in FIG. 10, when the intra prediction mode number of the chroma component is 4, the intra prediction direction of the associated luma component is used for the intra prediction sample generation of the chroma component. When the intra prediction mode number of the chroma component is not 4 and it is identical to the intra prediction mode number of the associated luma component, the intra prediction direction of 66 is used for the intra prediction sample generation of the chroma component.

[0141] Merge and skip mode in VVC

[0142] According to aspects of the disclosure, inter-picture prediction (also referred to as inter prediction) includes merge mode and skip mode.

[0143] In the merge mode for inter-picture prediction, the motion data (e.g., motion vector) of a block is inferred instead of being explicitly signaled. In an example, a merge candidate list of candidate motion parameters is constructed and an index is signaled to identify a candidate in the merge candidate list for predicting the current block.

[0144] In some embodiments, a merge candidate list includes a non-sub-CU merge candidate list and a sub-CU merge candidate list. The non-sub-CU merge candidate list is constructed based on spatial neighboring motion vectors, collocated temporal motion vectors, and history based motion vectors. The sub-CU merge candidate list includes affine merge candidates and ATMVP merge candidates. The sub-CU merge candidate list is used to derive multiple motion vectors for a current CU and different parts of samples in the current CU can have different motion vectors.

[0145] In the skip mode, the motion data of a block is inferred instead of being explicitly signaled and a prediction residual of the block is zero (i.e. no transform coefficients are transmitted). At the beginning of each CU in an inter-picture prediction slice, a skip_flag is signaled. The skip_flag indicates that a merge mode is used to derive the motion data and no residual data is present in the coded video bitstream.

[0146] Multi-hypothesis prediction for intra and inter mode

[0147] According to some aspects of the disclosure, intra and inter predictions can be suitably combined, such as a multi-hypothesis intra-inter prediction. The multi-hypothesis intra-inter prediction combines one intra prediction and one merge indexed prediction, and is referred to as intra-inter prediction mode in the present disclosure. In an example, when a CU is in the merge mode, a specific flag for intra mode is signaled. When the specific flag is true, an intra mode can be selected from an intra candidate list. For luma component, the intra candidate list is derived from 4 intra prediction modes including DC mode, planar mode, horizontal mode, and vertical mode, and the size of the intra mode candidate list can be 3 or 4 depending on the block shape. In an example, when the CU width is larger than twice of the CU height, the horizontal mode is removed from the intra mode candidate list and when the CU height is larger than twice of the CU width, vertical mode is removed from the intra mode candidate list. In some embodiments, an intra prediction is performed based on an intra prediction mode selected by an intra mode index and an inter prediction is performed based on a merge index. The intra prediction and the inter prediction are combined using weighted average. For chroma component, DM is always applied without extra signaling in some examples.

[0148] In some embodiments, the weights for combining the intra prediction and the inter prediction can be suitably determined. In an example, when DC or planar mode is selected or the Coding Block (CB) width or height is smaller than 4, equal weights are applied for inter prediction and intra prediction. In an example, for a CB with CB width and height larger than or equal to 4, when a horizontal or a vertical mode is selected, the CB is first vertically or horizontally split into four equal-area regions. Each region has a weight set, denoted as (w_intra_i, w_inter_i), where i is from 1 to 4. In an example, the first weight set (w_intra_1, w_inter_1) = (6, 2), the second weight set (w_intra_2, w_inter_2) = (5, 3), the third weight set

(w_intra_3, w_inter_3) = (3, 5), and fourth weight set (w_intra_4, w_inter_4) = (2, 6). Each weight set can be applied to a corresponding region. For example, the first weight set (w_intra_1, w_inter_1) is applied to a region closest to the reference samples and fourth weight set (w_intra_4, w_inter_4) is applied to a region farthest away from the reference samples.

[0149] In an embodiment, the combined prediction can be calculated by summing up the two weighted predictions and right-shifting 3 bits. Moreover, the intra prediction mode for the intra hypothesis of predictors can be saved for the intra mode coding of the following neighboring CBs when the neighboring CBs are intra coded.

[0150] Inter PDPC mode

[0151] According to aspects of the disclosure, the PDPC filtering process can be applied to an inter prediction mode. The inter prediction mode that employs the PDPC filter process is referred to as inter PDPC mode. In the inter PDPC mode, the PDPC filtering process is applied to inter prediction samples (or reconstructed samples of an inter coded CU). In an example, Eq. 1 can be suitably modified for the inter PDPC mode. For example, $pred[x][y]$ is modified to denote inter prediction value in the inter PDPC mode. In some examples, a flag, referred to as $interPDPCFlag$, is signaled to indicate whether to apply the PDPC filtering process in inter prediction. For example, when the flag $interPDPCFlag$ is true, the prediction samples (or reconstructed samples of an inter coded CU) are further modified in the PDPC filtering process.

[0152] It is noted that the inter PDPC mode can be combined with any suitable inter coded modes to further improve the inter prediction samples. However, there may be some restrictions on which inter coded mode the inter PDPC mode can be combined with. In one example, the inter PDPC mode is allowed to be only applied to regular merge mode and/or sub-block merge mode. In another example, the inter PDPC mode is not allowed to be applied to merge skip mode.

[0153] Triangle partition for inter prediction

[0154] According to aspects of the disclosure, triangle partition can be used in inter prediction. For example, in VVC test model (VTM) 3, a new triangle partition mode is introduced for inter prediction. The triangle partition mode is only applied to CUs that are 8x8 or larger and are coded in skip or merge mode. For a CU satisfying these conditions, a CU-level flag is signaled to indicate whether the triangle partition mode is applied or not.

[0155] When the triangular partition mode is used, a CU is split evenly into two triangle-shaped partitions, using either a diagonal split or an anti-diagonal split.

[0156] FIG. 11A shows the diagonal split of a CU and FIG. 11B shows the anti-diagonal split of a CU. Each triangle partition in a CU has its own motion information, and can be inter-predicted using its own motion information. In an example, only uni-prediction is allowed for each triangle partition. That is, each partition has one motion vector and one reference index. The uni-prediction motion constraint is applied to ensure that same as the conventional bi-prediction, only two motion compensated predictions are needed for each CU. The uni-prediction motion for each partition is derived from a uni-prediction candidate list constructed using the process.

[0157] In some examples, when a CU-level flag indicates that the current CU is coded using the triangle partition mode, then an index in the range of [0, 39] is further signaled. Using this triangle partition index, the direction of the triangle partition (diagonal or anti-diagonal), as well as the motion for each of the partitions can be obtained through a look-up table. After predicting each of the triangle partitions, the sample values along the diagonal or anti-diagonal edge are adjusted using a blending processing with adaptive weights. After the whole CU is predicted, the transform and quantization process will be applied to the whole CU as in other prediction modes. Finally, the motion field of a CU predicted using the triangle partition mode is stored in 4x4 units.

[0158] **Sub-block transform (SBT)**

[0159] In some embodiments, a sub-block transform (SBT), also referred to as spatially varying transform (SVT), is employed. The SBT can be applied to inter prediction residuals. For example, a coding block can be partitioned into sub-blocks, only part of the sub-blocks is treated at a residual block. Zero residual is assumed for the remaining part of the sub-blocks. Therefore, the residual block is smaller than the coding block, and a transform size in SBT is smaller than the coding block size. For the region which is not covered by the residual block, no transform processing is performed.

[0160] FIGs. 12A – 12D show exemplary SBT modes according to some embodiments of the disclosure. The SBT modes support different SBT types (SVT-H, SVT-V) (e.g., vertically or horizontally partitioned), sizes and positions (e.g., left half, left quarter, right half, right quarter, top half, top quarter, bottom half, bottom quarter). The shaded regions labeled by letter

"A" is residual blocks with transform, and the other regions is assumed to be zero residual without transform.

[0161] Multiple Transform Selection (MTS)

[0162] In HEVC, DCT-2 and 4×4 DST-7 are used as transform basis functions. In VVC, in addition to DCT-2 and 4×4 DST-7, an Adaptive Multiple Transform (AMT, also referred to as Enhanced Multiple Transform (EMT) or Multiple Transform Selection (MTS)) scheme has been used for residual coding for both inter and intra coded blocks. The AMT scheme may use multiple selected transforms from the DCT/DST families other than the current transforms in HEVC. The newly introduced transform matrices are DST-7, DCT-8. FIG. 13 shows an exemplary table of the basis functions of the selected DST/DCT.

[0163] In some embodiments, the primary transform matrices in VVC may be used with 8-bit representation. The AMT applies transform matrices to the CUs with both a width and a height smaller than or equal to 32. Whether the AMT is applied may be controlled by a flag called mts_flag. In an example, when the mts_flag is equal to 0, only DCT-2 is applied for coding the residue. In an example, when the mts_flag is equal to 1, an index mts_idx may be further signalled using 2 bins to identify the horizontal and vertical transform types to be used. FIG. 14 shows an exemplary table of the horizontal transform type $trTypeHor$ and the vertical transform type $trTypeVer$ depending on the index mts_idx, where, for the transform types, value 1 means using DST-7, and value 2 means using DCT-8.

[0164] A transform core is a matrix composed by basis vectors. APPENDIX B shows some transform cores of DST-7 and DCT-8.

[0165] In some embodiments (e.g., in VVC), when both the height and width of the coding block is smaller than or equal to 64, the transform size is always the same as the coding block size. When either the height or width of the coding block is larger than 64, the coding block is further split into multiple sub-blocks for residual transform or intra prediction. The width and height of each sub-block is smaller than or equal to 64, and a respective transform is performed on each sub-block.

[0166] Inter PDPC improvement techniques

[0167] Since inter PDPC mode can be combined with different inter prediction modes, such as regular merge mode or sub-block merge mode, the energy distribution of inter predicted residuals after applying the PDPC filtering process may be different for different inter prediction modes. Therefore, different transform designs may be considered for the different combinations of inter PDPC mode and other inter prediction modes.

[0168] Aspects of the disclosure provide improvement techniques for inter PDPC mode. The present methods or embodiments may be used separately or combined in any order. In the following description, the term block may be interpreted as a prediction block, a coding block, or a coding unit (CU). The PDPC filtering process as described above in the disclosure, or a variant, can be used in the inter PDPC mode.

[0169] According to aspects of the disclosure, a transform design of a current block that is used for inter predicted residuals after applying the PDPC filtering process is dependent on coded information of the current block and its neighboring blocks. The coded information includes, but is not limited to, inter prediction mode, merge flag, block size, and color component, etc.

[0170] In an embodiment, when the inter PDPC mode is employed (e.g., the inter PDPC flag is on/true), SBT is always allowed (e.g., the SBT flag is always on/true) or not allowed (e.g., the SBT flag is always off/false) regardless which inter coded modes the inter PDPC mode is applied to. In an example, when the inter PDPC mode is employed and SBT is always allowed, an SBT flag is not signaled but derived as on/true. In another example, when the inter PDPC mode is employed and SBT is always not allowed, an SBT flag is not signaled but derived as off/false.

[0171] In an embodiment, when the inter PDPC mode is applied to a sub-block merge mode, SBT is allowed (e.g., the SBT flag is on/true). Otherwise, when the inter PDPC mode is applied to a merge mode except the sub-block merge mode, SBT is not allowed (e.g., the SBT flag is always off/false). In an example, when the inter PDPC mode is applied to a sub-block merge mode and SBT is allowed, an SBT flag is signaled. In another example, when the inter PDPC mode is applied to a non-sub-block merge mode and SBT is not allowed, an SBT flag is not signaled but derived as off/false.

[0172] In an embodiment, when the inter PDPC mode is applied to a triangle merge mode, SBT is allowed (e.g., the SBT flag is on/true). Alternatively, when the inter PDPC mode is applied to a triangle merge mode, SBT is not allowed (e.g., the SBT flag is always off/false). In an example, when the inter PDPC mode is applied to a triangle merge mode and SBT is allowed, an SBT flag is signaled. In another example, when the inter PDPC mode is applied to a triangle merge mode and SBT is not allowed, an SBT flag is not signaled but derived as off/false.

[0173] In an embodiment, when the inter PDPC mode is employed (e.g., the inter PDPC flag is on/true), only a subset of SBT types can be allowed. In an example, when both of the inter PDPC flag and SBT flag are on/true, the residual partition (i.e., a partition that is allowed to have non-zero coefficients) of a current CU is always the rightmost or bottommost partition of the current CU. In another example, when both of the inter PDPC flag and SBT flag are on/true, the leftmost or topmost partition of the current CU is always the zero-residual partition.

[0174] In an embodiment, when inter PDPC mode is employed (e.g., the inter PDPC flag is on/true), only a subset of available transform types can be used. In an example, when the inter PDPC mode is employed (e.g., the inter PDPC flag is on/true), DCT-8 is excluded from the allowed transform types. In an example, when the inter PDPC mode is employed, only DCT-2 and DST-7 can be used for the inter predicted residuals. In an example, when the inter PDPC mode is employed, DST-7 is excluded from the allowed transform types. In an example, when the inter PDPC mode is employed, only DCT-2 and DCT-8 can be used for the inter predicted residuals. In an example, when the inter PDPC mode is employed, only DST-7 can be applied.

[0175] In one embodiment, when the inter PDPC mode is employed (e.g., the inter PDPC flag is on/true), MTS is always allowed (e.g., MTS flag is always on/true) or not allowed (e.g., MTS flag is always off/false) regardless which inter coded modes the inter PDPC mode is applied to. In an example, when the inter PDPC mode is employed and MTS is always allowed, an MTS flag is not signaled but derived as on/true. In another example, when the inter PDPC mode is employed and MTS is always not allowed, an MTS flag is not signaled but derived as off/false.

[0176] According to aspects of the disclosure, the inter PDPC flag is context coded. The context is dependent on coded information of a current block and its neighboring blocks. The coded information includes, but is not limited to, intra-inter flag, intra mode flag, inter mode flag, skip flag, merge flag, and inter PDPC flag, etc.

[0177] In an embodiment, only one context is used for the entropy coding of the inter PDPC mode. In another embodiment, M contexts are used for the entropy coding of inter PDPC mode. For example, M can be any positive integer greater than 1, such as 2 or 3. In an example where M is greater than 1, if one of the neighboring blocks is intra coded or intra-inter coded or inter PDPC coded, a first context is used. Otherwise, a second context is used. In another example where M is greater than 2, if both neighboring blocks are intra coded or intra-inter coded or inter PDPC coded, a first context is used. If only one of the neighboring blocks is intra coded or intra-inter coded or inter PDPC coded, a second context is used. Otherwise, a third context is used.

[0178] According to aspects of the disclosure, the inter PDPC mode is not allowed to be used together with Intra Block Copy (IBC) mode.

[0179] According to aspects of the disclosure, the weighting factors (e.g., wL, wT and/or wTL) applied in the inter PDPC mode depend on coded information of a neighboring block of a current block. The coded information of the neighboring block includes whether the neighboring block is coded by intra-inter mode, intra prediction mode, inter prediction mode, skip mode, merge mode, or inter PDPC mode, etc.

[0180] FIG. 15 shows a flow chart outlining an exemplary process (1500) according to an embodiment of the disclosure. In various embodiments, the process (1500) is executed by processing circuitry, such as the processing circuitry in the terminal devices (210), (220), (230) and (240), the processing circuitry that performs functions of the video encoder (303), the processing circuitry that performs functions of the video decoder (310), the processing circuitry that performs functions of the video decoder (410), the processing circuitry that performs functions of the intra prediction module (452), the processing circuitry that performs functions of the video encoder (503), the processing circuitry that performs functions of the predictor (535), the processing circuitry that performs functions of the intra encoder (622), the processing circuitry that performs functions of the intra decoder (772), and the like. In some embodiments, the process (1500) is implemented in software instructions, thus when the processing circuitry executes the software instructions, the processing circuitry performs the process (1500).

[0181] The process (1500) may generally start at step (S1510), where the process (1500) decodes prediction information for a current block in a current picture that is a part of a coded video sequence. The prediction information indicates a specific inter prediction mode, a position dependent prediction combination (PDPC) process, and a coded residual for the current block. Then the process (1500) proceeds to step (S1520).

[0182] At step (S1520), the process (1500) determines at least one of (i) a partition of the current block to which a transform process is to be applied and (ii) a transform type of the transform process for the current block based on the prediction information. Then the process (1500) proceeds to step (S1530).

[0183] At step (S1530), the process (1500) performs the transform process on the coded residual to generate a decoded residual. Then process (1500) proceeds to step (S1540).

[0184] At step (S1540), the process (1500) reconstructs the current block based on the decoded residual.

[0185] After reconstructing the current block, the process (1500) terminates.

[0186] In some embodiments, the process (1500) determines the transform type to be one of a plurality of sub-block transform (SBT) types, each of the plurality of SBT types being applied to a different part of the current block. In an embodiment, the application of the one of the plurality of SBT types is independent of the specific inter prediction mode of the current block. In an embodiment, the part that the one of the plurality of SBT types is applied to is one of a rightmost part and a bottommost part of a plurality of parts of the current block.

[0187] In an embodiment, the process (1500) determines a subset of a DCT-2 transform, a DST-7 transform, and a DCT-8 transform from which the transform type is selected.

[0188] In an embodiment, the prediction information includes a signaled flag that is (i) context coded and (ii) indicates the PDPC process for the current block.

[0189] In an embodiment, an intra block copy (IBC) mode is not permitted for the current block.

[0190] In an embodiment, a plurality of weighting parameters of the PDPC process is based on prediction information of a neighboring block adjacent to the current block.

[0191] **Computer system**

[0192] The techniques described above, can be implemented as computer software using computer-readable instructions and physically stored in one or more computer-readable media. For example, FIG. 16 shows a computer system (1600) suitable for implementing certain embodiments of the disclosed subject matter.

[0193] The computer software can be coded using any suitable machine code or computer language, that may be subject to assembly, compilation, linking, or like mechanisms to create code comprising instructions that can be executed directly, or through interpretation, micro-code execution, and the like, by one or more computer central processing units (CPUs), Graphics Processing Units (GPUs), and the like.

[0194] The instructions can be executed on various types of computers or components thereof, including, for example, personal computers, tablet computers, servers, smartphones, gaming devices, internet of things devices, and the like.

[0195] The components shown in FIG. 16 for computer system (1600) are exemplary in nature and are not intended to suggest any limitation as to the scope of use or functionality of the computer software implementing embodiments of the present disclosure. Neither should the configuration of components be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary embodiment of a computer system (1600).

[0196] Computer system (1600) may include certain human interface input devices. Such a human interface input device may be responsive to input by one or more human users through, for example, tactile input (such as: keystrokes, swipes, data glove movements), audio input (such as: voice, clapping), visual input (such as: gestures), olfactory input (not depicted). The human interface devices can also be used to capture certain media not necessarily directly related to conscious input by a human, such as audio (such as: speech, music, ambient sound), images (such as: scanned images, photographic images obtain from a still image camera), video (such as two-dimensional video, three-dimensional video including stereoscopic video).

[0197] Input human interface devices may include one or more of (only one of each depicted): keyboard (1601), mouse (1602), trackpad (1603), touch screen (1610), data-glove (not shown), joystick (1605), microphone (1606), scanner (1607), camera (1608).

[0198] Computer system (1600) may also include certain human interface output devices. Such human interface output devices may be stimulating the senses of one or more human users through, for example, tactile output, sound, light, and smell/taste. Such human interface output devices may include tactile output devices (for example tactile feedback by the touch-screen (1610), data-glove (not shown), or joystick (1605), but there can also be tactile feedback devices that do not serve as input devices), audio output devices (such as: speakers (1609), headphones (not depicted)), visual output devices (such as screens (1610) to include CRT screens, LCD screens, plasma screens, OLED screens, each with or without touch-screen input capability, each with or without tactile feedback capability—some of which may be capable to output two dimensional visual output or more than three dimensional output through means such as stereographic output; virtual-reality glasses (not depicted), holographic displays and smoke tanks (not depicted)), and printers (not depicted). These visual output devices (such as screens (1610)) can be connected to a system bus (1648) through a graphics adapter (1650).

[0199] Computer system (1600) can also include human accessible storage devices and their associated media such as optical media including CD/DVD ROM/RW (1620) with CD/DVD or the like media (1621), thumb-drive (1622), removable hard drive or solid state drive (1623), legacy magnetic media such as tape and floppy disc (not depicted), specialized ROM/ASIC/PLD based devices such as security dongles (not depicted), and the like.

[0200] Those skilled in the art should also understand that term "computer readable media" as used in connection with the presently disclosed subject matter does not encompass transmission media, carrier waves, or other transitory signals.

[0201] Computer system (1600) can also include a network interface (1654) to one or more communication networks (1655). The one or more communication networks (1655) can for example be wireless, wireline, optical. The one or more communication networks (1655) can further be local, wide-area, metropolitan, vehicular and industrial, real-time, delay-tolerant, and so on. Examples of the one or more communication networks (1655) include local area networks such as Ethernet, wireless LANs, cellular networks to include GSM, 3G, 4G, 5G, LTE and the like, TV wireline or wireless wide area digital networks to include cable TV, satellite TV, and terrestrial broadcast TV, vehicular and industrial to include CANBus, and so forth. Certain networks commonly require external network interface adapters that attached to certain general purpose data ports or peripheral buses (1649) (such as, for example USB ports of the computer system (1600)); others are commonly integrated into the core of the computer system (1600) by attachment to a system bus as described below (for example Ethernet interface into a PC computer system or cellular network interface into a smartphone computer system). Using any of these networks, computer system (1600) can communicate with other entities. Such communication can be uni-directional, receive only (for example, broadcast TV), uni-directional send-only (for example CANbus to certain CANbus devices), or bi-directional, for example to other computer systems using local or wide area digital networks. Certain protocols and protocol stacks can be used on each of those networks and network interfaces as described above.

[0202] Aforementioned human interface devices, human-accessible storage devices, and network interfaces can be attached to a core (1640) of the computer system (1600).

[0203] The core (1640) can include one or more Central Processing Units (CPU) (1641), Graphics Processing Units (GPU) (1642), specialized programmable processing units in the form of Field Programmable Gate Areas (FPGA) (1643), hardware accelerators for certain tasks (1644), and so forth. These devices, along with Read-only memory (ROM) (1645), Random-access memory (1646), internal mass storage such as internal non-user accessible hard drives, SSDs, and the like (1647), may be connected through the system bus (1648). In some computer systems, the system bus (1648) can be accessible in the form of one or more physical plugs to enable extensions by additional CPUs, GPU, and the like. The peripheral devices can be attached either directly to the core's system bus (1648), or through a peripheral bus (1649). Architectures for a peripheral bus include PCI, USB, and the like.

[0204] CPUs (1641), GPUs (1642), FPGAs (1643), and accelerators (1644) can execute certain instructions that, in combination, can make up the aforementioned computer code. That computer code can be stored in ROM (1645) or RAM (1646). Transitional data can be also be stored in RAM (1646), whereas permanent data can be stored for example, in the internal mass storage (1647). Fast storage and retrieve to any of the memory devices can be enabled through the use of cache memory, that can be closely associated with one or more CPU (1641), GPU (1642), mass storage (1647), ROM (1645), RAM (1646), and the like.

[0205] The computer readable media can have computer code thereon for performing various computer-implemented operations. The media and computer code can be those specially designed and constructed for the purposes of the present disclosure, or they can be of the kind well known and available to those having skill in the computer software arts.

[0206] As an example and not by way of limitation, the computer system having architecture (1600), and specifically the core (1640) can provide functionality as a result of processor(s) (including CPUs, GPUs, FPGA, accelerators, and the like) executing software embodied in one or more tangible, computer-readable media. Such computer-readable media can be media associated with user-accessible mass storage as introduced above, as well as certain storage of the core (1640) that are of non-transitory nature, such as core-internal mass storage (1647) or ROM (1645). The software implementing various embodiments of the present disclosure can be stored in such devices and executed by core (1640). A computer-readable medium can include one or more memory devices or chips, according to particular needs. The software can cause the core (1640) and specifically the processors therein (including CPU, GPU, FPGA, and the like) to execute particular processes or particular parts of particular processes described herein, including defining data structures stored in RAM (1646) and modifying such data structures according to the processes defined by the software. In addition or as an alternative, the computer system can provide functionality as a result of logic hardwired or otherwise embodied in a circuit (for example: accelerator (1644)), which can operate in place of or together with software to execute particular processes or particular parts of particular processes described herein. Reference to software can encompass logic, and vice versa, where appropriate. Reference to a computer-readable media can encompass a circuit (such as an integrated circuit (IC)) storing software for execution, a circuit embodying logic for execution, or both, where appropriate. The present disclosure encompasses any suitable combination of hardware and software.

[0207] While this disclosure has described several exemplary embodiments, there are alterations, permutations, and various substitute equivalents, which fall within the scope of the disclosure. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the disclosure and are thus within the spirit and scope thereof.

Appendix A: Acronyms

AMVP: Advanced Motion Vector Prediction

ASIC: Application-Specific Integrated Circuit

ATMVP: Alternative/Advanced Temporal Motion Vector Prediction

BDOF: Bi-directional Optical Flow

BIO: Bi-directional Optical Flow

BMS: Benchmark Set

BV: Block Vector

CANBus: Controller Area Network Bus

CB: Coding Block

CBF: Coded Block Flag

CCLM: Cross-Component Linear Mode/Model

CD: Compact Disc

CPR: Current Picture Referencing

CPUs: Central Processing Units

CRT: Cathode Ray Tube

CTBs: Coding Tree Blocks

CTUs: Coding Tree Units

CU: Coding Unit

DPB: Decoder Picture Buffer

DVD: Digital Video Disc

FPGA: Field Programmable Gate Areas

GOPs: Groups of Pictures

GPUs: Graphics Processing Units

GSM: Global System for Mobile communications

HDR: High Dynamic Range

HEVC: High Efficiency Video Coding

HRD: Hypothetical Reference Decoder

IBC: Intra Block Copy

IC: Integrated Circuit

JEM: Joint Exploration Model

JVET: Joint Video Exploration Team

LAN: Local Area Network

LCD: Liquid-Crystal Display

LTE: Long-Term Evolution

MPM: Most Probable Mode

MTS: Multiple Transform Selection

MV: Motion Vector

OLED: Organic Light-Emitting Diode

PBs: Prediction Blocks

PCI: Peripheral Component Interconnect

PDPC: Position Dependent Prediction Combination

PLD: Programmable Logic Device

PU: Prediction Unit

RAM: Random Access Memory

ROM: Read-Only Memory

SBT: Sub-block Transform

SCC: Screen Content Coding

SDR: Standard Dynamic Range

SEI: Supplementary Enhancement Information

SNR: Signal Noise Ratio

SSD: Solid-state Drive

TUs: Transform Units

USB: Universal Serial Bus

VPDU: Visual Process Data Unit

VUI: Video Usability Information

VVC: Versatile Video Coding

WAIP: Wide-Angle Intra Prediction

Appendix B: Transform cores

4-point DST-7

{ a, b, c, d }

{ c, c, 0, -c }

{ d, -a, -c, b }

{ b, -d, c, -a }

where {a, b, c, d} = { 29, 55, 74, 84}

8-point DST-7:

{ a, b, c, d, e, f, g, h,}

{ c, f, h, e, b, -a, -d, -g,}

{ e, g, b, -c, -h, -d, a, f,}

{ g, c, -d, -f, a, h, b, -e,}

{ h, -a, -g, b, f, -c, -e, d,}

{ f, -e, -a, g, -d, -b, h, -c,}

{ d, -h, e, -a, -c, g, -f, b,}

{ b, -d, f, -h, g, -e, c, -a,}

where {a, b, c, d, e, f, g, h} = { 17, 32, 46, 60, 71, 78, 85, 86}

16-point DST-7

{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p,}

{ c, f, i, l, o, o, l, i, f, c, 0, -c, -f, -i, -l, -o,}

{ e, j, o, m, h, c, -b, -g, -l, -p, -k, -f, -a, d, i, n,}

{ g, n, l, e, -b, -i, -p, -j, -c, d, k, o, h, a, -f, -m,}

{ i, o, f, -c, -l, -l, -c, f, o, i, 0, -i, -o, -f, c, l,}

{ k, k, 0, -k, -k, 0, k, k, 0, -k, -k, 0, k, k, 0, -k,}

{ m, g, -f, -n, -a, l, h, -e, -o, -b, k, i, -d, -p, -c, j,}

{ o, c, -l, -f, i, i, -f, -l, c, o, 0, -o, -c, l, f, -i,}

{ p, -a, -o, b, n, -c, -m, d, l, -e, -k, f, j, -g, -i, h,}

{ n, -e, -i, j, d, -o, a, m, -f, -h, k, c, -p, b, l, -g,}

{ l, -i, -c, o, -f, -f, o, -c, -i, l, 0, -l, i, c, -o, f,}

{ j, -m, c, g, -p, f, d, -n, i, a, -k, l, -b, -h, o, -e,}

{ h, -p, i, -a, -g, o, -j, b, f, -n, k, -c, -e, m, -l, d,}

{ f, -l, o, -i, c, c, -i, o, -l, f, 0, -f, l, -o, i, -c,}

{ d, -h, l, -p, m, -i, e, -a, -c, g, -k, o, -n, j, -f, b,}

{ b, -d, f, -h, j, -l, n, -p, o, -m, k, -i, g, -e, c, -a,}

where {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p} = { 9, 17, 25, 33, 41, 49, 56, 62, 66, 72, 77, 81, 83, 87, 89, 90}

32-point DST-7

{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F,}

{ c, f, i, l, o, r, u, x, A, D, F, C, z, w, t, q, n, k, h, e, b, -a, -d, -g, -j, -m, -p, -s, -v, -y, -B, -E,}

{ e, j, o, t, y, D, D, y, t, o, j, e, 0, -e, -j, -o, -t, -y, -D, -D, -y, -t, -o, -j, -e, 0, e, j, o, t, y, D,}

{ g, n, u, B, D, w, p, i, b, -e, -l, -s, -z, -F, -y, -r, -k, -d, c, j, q, x, E, A, t, m, f, -a, -h, -o, -v, -C,}

{ i, r, A, C, t, k, b, -g, -p, -y, -E, -v, -m, -d, e, n, w, F, x, o, f, -c, -l, -u, -D, -z, -q, -h, a, j, s, B,}

{ k, v, F, u, j, -a, -l, -w, -E, -t, -i, b, m, x, D, s, h, -c, -n, -y, -C, -r, -g, d, o, z, B, q, f, -e, -p, -A,}

{ m, z, z, m, 0, -m, -z, -z, -m, 0, m, z, z, m, 0, -m, -z, -z, -m, 0, m, z, z, m, 0, -m, -z, -z, -m, 0, m, z,}

{ o, D, t, e, -j, -y, -y, -j, e, t, D, o, 0, -o, -D, -t, -e, j, y, y, j, -e, -t, -D, -o, 0, o, D, t, e, -j, -y,}

{ q, E, n, -c, -t, -B, -k, f, w, y, h, -i, -z, -v, -e, l, C, s, b, -o, -F, -p, a, r, D, m, -d, -u, -A, -j, g, x,}

{ s, A, h, -k, -D, -p, c, v, x, e, -n, -F, -m, f, y, u, b, -q, -C, -j, i, B, r, -a, -t, -z, -g, l, E, o, -d, -w,}

{ u, w, b, -s, -y, -d, q, A, f, -o, -C, -h, m, E, j, -k, -F, -l, i, D, n, -g, -B, -p, e, z, r, -c, -x, -t, a, v,}

{ w, s, -d, -A, -o, h, E, k, -l, -D, -g, p, z, c, -t, -v, a, x, r, -e, -B, -n, i, F, j, -m, -C, -f, q, y, b, -u,}

{ y, o, -j, -D, -e, t, t, -e, -D, -j, o, y, 0, -y, -o, j, D, e, -t, -t, e, D, j, -o, -y, 0, y, o, -j, -D, -e, t,}

{ A, k, -p, -v, e, F, f, -u, -q, j, B, a, -z, -l, o, w, -d, -E, -g, t, r, -i, -C, -b, y, m, -n, -x, c, D, h, -s,}

{ C, g, -v, -n, o, u, -h, -B, a, D, f, -w, -m, p, t, -i, -A, b, E, e, -x, -l, q, s, -j, -z, c, F, d, -y, -k, r,}

{ E, c, -B, -f, y, i, -v, -l, s, o, -p, -r, m, u, -j, -x, g, A, -d, -D, a, F, b, -C, -e, z, h, -w, -k, t, n, -q,}

{ F, -a, -E, b, D, -c, -C, d, B, -e, -A, f, z, -g, -y, h, x, -i, -w, j, v, -k, -u, l, t, -m, -s, n, r, -o, -q, p,}

{ D, -e, -y, j, t, -o, -o, t, j, -y, -e, D, 0, -D, e, y, -j, -t, o, o, -t, -j, y, e, -D, 0, D, -e, -y, j, t, -o,}

{ B, -i, -s, r, j, -A, -a, C, -h, -t, q, k, -z, -b, D, -g, -u, p, l, -y, -c, E, -f, -v, o, m, -x, -d, F, -e, -w, n,}

{ z, -m, -m, z, 0, -z, m, m, -z, 0, z, -m, -m, z, 0, -z, m, m, -z, 0, z, -m, -m, z, 0, -z, m, m, -z, 0, z, -m,}

{ x, -q, -g, E, -j, -n, A, -c, -u, t, d, -B, m, k, -D, f, r, -w, -a, y, -p, -h, F, -i, -o, z, -b, -v, s, e, -C, l,}

{ v, -u, -a, w, -t, -b, x, -s, -c, y, -r, -d, z, -q, -e, A, -p, -f, B, -o, -g, C, -n, -h, D, -m, -i, E, -l, -j, F, -k,}

{ t, -y, e, o, -D, j, j, -D, o, e, -y, t, 0, -t, y, -e, -o, D, -j, -j, D, -o, -e, y, -t, 0, t, -y, e, o, -D, j,}

{ r, -C, k, g, -y, v, -d, -n, F, -o, -c, u, -z, h, j, -B, s, -a, -q, D, -l, -f, x, -w, e, m, -E, p, b, -t, A, -i,}

{ p, -F, q, -a, -o, E, -r, b, n, -D, s, -c, -m, C, -t, d, l, -B, u, -e, -k, A, -v, f, j, -z, w, -g, -i, y, -x, h,}

{ n, -B, w, -i, -e, s, -F, r, -d, -j, x, -A, m, a, -o, C, -v, h, f, -t, E, -q, c, k, -y, z, -l, -b, p, -D, u, -g,}

{ l, -x, C, -q, e, g, -s, E, -v, j, b, -n, z, -A, o, -c, -i, u, -F, t, -h, -d, p, -B, y, -m, a, k, -w, D, -r, f,}

{ j, -t, D, -y, o, -e, -e, o, -y, D, -t, j, 0, -j, t, -D, y, -o, e, e, -o, y, -D, t, -j, 0, j, -t, D, -y, o, -e,}

{ h, -p, x, -F, y, -q, i, -a, -g, o, -w, E, -z, r, -j, b, f, -n, v, -D, A, -s, k, -c, -e, m, -u, C, -B, t, -l, d,}

{ f, -l, r, -x, D, -C, w, -q, k, -e, -a, g, -m, s, -y, E, -B, v, -p, j, -d, -b, h, -n, t, -z, F, -A, u, -o, i, -c,}

{ d, -h, l, -p, t, -x, B, -F, C, -y, u, -q, m, -i, e, -a, -c, g, -k, o, -s, w, -A, E, -D, z, -v, r, -n, j, -f, b,}

{ b, -d, f, -h, j, -l, n, -p, r, -t, v, -x, z, -B, D, -F, E, -C, A, -y, w, -u, s, -q, o, -m, k, -i, g, -e, c, -a,}

where {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F } = { 4, 9, 13, 17, 21, 26, 30, 34, 38, 42, 45, 50, 53, 56, 60, 63, 66, 68, 72, 74, 77, 78, 80, 82, 84, 85, 86, 88, 88, 89, 90, 90 }

4-point DCT-8

{ a, b, c, d,}

{ b, 0, -b, -b,}
{ c, -b, -d, a,}
{ d, -b, a, -c,}

where {a, b, c, d} = { 84,74,55,29}

8-point DCT-8:

{ a, b, c, d, e, f, g, h,}
{ b, e, h, -g, -d, -a, -c, -f,}
{ c, h, -e, -a, -f, g, b, d,}
{ d, -g, -a, -h, c, e, -f, -b,}
{ e, -d, -f, c, g, -b, -h, a,}
{ f, -a, g, e, -b, h, d, -c,}
{ g, -c, b, -f, -h, d, -a, e,}
{ h, -f, d, -b, a, -c, e, -g,}

where {a, b, c, d, e, f, g, h} = { 86,85,78,71,60,46,32,17}

16-point DCT-8

{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p,}
{ b, e, h, k, n, 0, -n, -k, -h, -e, -b, -b, -e, -h, -k, -n,}
{ c, h, m, -p, -k, -f, -a, -e, -j, -o, n, i, d, b, g, l,}
{ d, k, -p, -i, -b, -f, -m, n, g, a, h, o, -l, -e, -c, -j,}
{ e, n, -k, -b, -h, 0, h, b, k, -n, -e, -e, -n, k, b, h,}
{ f, 0, -f, -f, 0, f, f, 0, -f, -f, 0, f, f, 0, -f, -f,}
{ g, -n, -a, -m, h, f, -o, -b, -l, i, e, -p, -c, -k, j, d,}
{ h, -k, -e, n, b, 0, -b, -n, e, k, -h, -h, k, e, -n, -b,}
{ i, -h, -j, g, k, -f, -l, e, m, -d, -n, c, o, -b, -p, a,}
{ j, -e, -o, a, -n, -f, i, k, -d, -p, b, -m, -g, h, l, -c,}
{ k, -b, n, h, -e, 0, e, -h, -n, b, -k, -k, b, -n, -h, e,}
{ l, -b, i, o, -e, f, -p, -h, c, -m, -k, a, -j, -n, d, -g,}
{ m, -e, d, -l, -n, f, -c, k, o, -g, b, -j, -p, h, -a, i,}
{ n, -h, b, -e, k, 0, -k, e, -b, h, -n, -n, h, -b, e, -k,}
{ o, -k, g, -c, b, -f, j, -n, -p, l, -h, d, -a, e, -i, m,}

{ p, -n, l, -j, h, -f, d, -b, a, -c, e, -g, i, -k, m, -o,}

where {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p} = { 90,89,87,83,81,77,72,66,62,56,49,41,33,25,17, 9}

32-point DCT-8

{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F,}

{ b, e, h, k, n, q, t, w, z, C, F, -E, -B, -y, -v, -s, -p, -m, -j, -g, -d, -a, -c, -f, -i, -l, -o, -r, -u, -x, -A, -D,}

{ c, h, m, r, w, B, 0, -B, -w, -r, -m, -h, -c, -c, -h, -m, -r, -w, -B, 0, B, w, r, m, h, c, c, h, m, r, w, B,}

{ d, k, r, y, F, -A, -t, -m, -f, -b, -i, -p, -w, -D, C, v, o, h, a, g, n, u, B, -E, -x, -q, -j, -c, -e, -l, -s, -z,}

{ e, n, w, F, -y, -p, -g, -c, -l, -u, -D, A, r, i, a, j, s, B, -C, -t, -k, -b, -h, -q, -z, E, v, m, d, f, o, x,}

{ f, q, B, -A, -p, -e, -g, -r, -C, z, o, d, h, s, D, -y, -n, -c, -i, -t, -E, x, m, b, j, u, F, -w, -l, -a, -k, -v,}

{ g, t, 0, -t, -g, -g, -t, 0, t, g, g, t, 0, -t, -g, -g, -t, 0, t, g, g, t, 0, -t, -g, -g, -t, 0, t, g, g, t,}

{ h, w, -B, -m, -c, -r, 0, r, c, m, B, -w, -h, -h, -w, B, m, c, r, 0, -r, -c, -m, -B, w, h, h, w, -B, -m, -c, -r,}

{ i, z, -w, -f, -l, -C, t, c, o, F, -q, -a, -r, E, n, d, u, -B, -k, -g, -x, y, h, j, A, -v, -e, -m, -D, s, b, p,}

{ j, C, -r, -b, -u, z, g, m, F, -o, -e, -x, w, d, p, -E, -l, -h, -A, t, a, s, -B, -i, -k, -D, q, c, v, -y, -f, -n,}

{ k, F, -m, -i, -D, o, g, B, -q, -e, -z, s, c, x, -u, -a, -v, w, b, t, -y, -d, -r, A, f, p, -C, -h, -n, E, j, l,}

{ l, -E, -h, -p, A, d, t, -w, -a, -x, s, e, B, -o, -i, -F, k, m, -D, -g, -q, z, c, u, -v, -b, -y, r, f, C, -n, -j,}

{ m, -B, -c, -w, r, h, 0, -h, -r, w, c, B, -m, -m, B, c, w, -r, -h, 0, h, r, -w, -c, -B, m, m, -B, -c, -w, r, h,}

{ n, -y, -c, -D, i, s, -t, -h, E, d, x, -o, -m, z, b, C, -j, -r, u, g, -F, -e, -w, p, l, -A, -a, -B, k, q, -v, -f,}

{ o, -v, -h, C, a, D, -g, -w, n, p, -u, -i, B, b, E, -f, -x, m, q, -t, -j, A, c, F, -e, -y, l, r, -s, -k, z, d,}

{ p, -s, -m, v, j, -y, -g, B, d, -E, -a, -F, c, C, -f, -z, i, w, -l, -t, o, q, -r, -n, u, k, -x, -h, A, e, -D, -b,}

{ q, -p, -r, o, s, -n, -t, m, u, -l, -v, k, w, -j, -x, i, y, -h, -z, g, A, -f, -B, e, C, -d, -D, c, E, -b, -F, a,}

{ r, -m, -w, h, B, -c, 0, c, -B, -h, w, m, -r, -r, m, w, -h, -B, c, 0, -c, B, h, -w, -m, r, r, -m, -w, h, B, -c,}

{ s, -j, -B, a, -C, -i, t, r, -k, -A, b, -D, -h, u, q, -l, -z, c, -E, -g, v, p, -m, -y, d, -F, -f, w, o, -n, -x, e,}

{ t, -g, 0, g, -t, -t, g, 0, -g, t, t, -g, 0, g, -t, -t, g, 0, -g, t, t, -g, 0, g, -t, -t, g, 0, -g, t, t, -g,}

{ u, -d, B, n, -k, -E, g, -r, -x, a, -y, -q, h, -F, -j, o, A, -c, v, t, -e, C, m, -l, -D, f, -s, -w, b, -z, -p, i,}

{ v, -a, w, u, -b, x, t, -c, y, s, -d, z, r, -e, A, q, -f, B, p, -g, C, o, -h, D, n, -i, E, m, -j, F, l, -k,}

{ w, -c, r, B, -h, m, 0, -m, h, -B, -r, c, -w, -w, c, -r, -B, h, -m, 0, m, -h, B, r, -c, w, w, -c, r, B, -h, m,}

{ x, -f, m, -E, -q, b, -t, -B, j, -i, A, u, -c, p, F, -n, e, -w, -y, g, -l, D, r, -a, s, C, -k, h, -z, -v, d, -o,}

{ y, -i, h, -x, -z, j, -g, w, A, -k, f, -v, -B, l, -e, u, C, -m, d, -t, -D, n, -c, s, E, -o, b, -r, -F, p, -a, q,}

{ z, -l, c, -q, E, u, -g, h, -v, -D, p, -b, m, -A, -y, k, -d, r, -F, -t, f, -i, w, C, -o, a, -n, B, x, -j, e, -s,}

{ A, -o, c, -j, v, F, -t, h, -e, q, -C, -y, m, -a, l, -x, -D, r, -f, g, -s, E, w, -k, b, -n, z, B, -p, d, -i, u,}

{ B, -r, h, -c, m, -w, 0, w, -m, c, -h, r, -B, -B, r, -h, c, -m, w, 0, -w, m, -c, h, -r, B, B, -r, h, -c, m, -w,}

{ C, -u, m, -e, d, -l, t, -B, -D, v, -n, f, -c, k, -s, A, E, -w, o, -g, b, -j, r, -z, -F, x, -p, h, -a, i, -q, y,}
{ D, -x, r, -l, f, -a, g, -m, s, -y, E, C, -w, q, -k, e, -b, h, -n, t, -z, F, B, -v, p, -j, d, -c, i, -o, u, -A,}
{ E, -A, w, -s, o, -k, g, -c, b, -f, j, -n, r, -v, z, -D, -F, B, -x, t, -p, l, -h, d, -a, e, -i, m, -q, u, -y, C,}
{ F, -D, B, -z, x, -v, t, -r, p, -n, l, -j, h, -f, d, -b, a, -c, e, -g, i, -k, m, -o, q, -s, u, -w, y, -A, C, -E,}

where {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F } = {90,90,89,88,88,86,85,84,82,80,78,77,74,72,68,66,63,60,56,53,50,45,42,38,34,30,26,21,17,13, 9, 4}

WHAT IS CLAIMED IS:

1. A method for video decoding in a decoder, comprising:

decoding prediction information for a current block in a current picture that is a part of a coded video sequence, the prediction information indicating a specific inter prediction mode, a position dependent prediction combination (PDPC) process, and a coded residual for the current block;

determining at least one of (i) a partition of the current block to which a transform process is to be applied and (ii) a transform type of the transform process for the current block based on the prediction information;

performing the transform process on the coded residual to generate a decoded residual; and

reconstructing the current block based on the decoded residual.

2. The method of claim 1, wherein the determining further comprises:

determining the transform type to be one of a plurality of sub-block transform (SBT) types, each of the plurality of SBT types being applied to a different part of the current block.

3. The method of claim 2, wherein application of the one of the plurality of SBT types is independent of the specific inter prediction mode of the current block.

4. The method of claim 2, wherein the part that the one of the plurality of SBT types is applied to is one of a rightmost part and a bottommost part of a plurality of parts of the current block.

5. The method of claim 1, further comprising:

determining a subset of a DCT-2 transform, a DST-7 transform, and a DCT-8 transform from which the transform type is selected.

6. The method of claim 1, wherein the prediction information includes a signaled flag that is (i) context coded and (ii) indicates the PDPC process for the current block.

7. The method of claim 1, wherein an intra block copy (IBC) mode is not permitted for the current block.

8. The method of claim 1, wherein a plurality of weighting parameters of the PDPC process is based on prediction information of a neighboring block adjacent to the current block.

9. An apparatus, comprising a processing circuitry configured to:

decode prediction information for a current block in a current picture that is a part of a coded video sequence, the prediction information indicating a specific inter prediction mode, a position dependent prediction combination (PDPC) process, and a coded residual for the current block;

determine at least one of (i) a partition of the current block to which a transform process is to be applied and (ii) a transform type of the transform process for the current block based on the prediction information;

perform the transform process on the coded residual to generate a decoded residual; and

reconstruct the current block based on the decoded residual.

10. The apparatus of claim 9, wherein the processing circuitry is further configured to:

determine the transform type to be one of a plurality of sub-block transform (SBT) types, each of the plurality of SBT types being applied to a different part of the current block.

11. The apparatus of claim 10, wherein application of the one of the plurality of SBT types is independent of the specific inter prediction mode of the current block.

12. The apparatus of claim 10, wherein the part that the one of the plurality of SBT types is applied to is one of a rightmost part and a bottommost part of a plurality of parts of the current block.

13. The apparatus of claim 9, wherein the processing circuitry is further configured to:

determine a subset of a DCT-2 transform, a DST-7 transform, and a DCT-8 transform from which the transform type is selected.

14. The apparatus of claim 9, wherein the prediction information includes a signaled flag that is (i) context coded and (ii) indicates the PDPC process for the current block.

15. The apparatus of claim 9, wherein an intra block copy (IBC) mode is not permitted for the current block.

16. The apparatus of claim 9, wherein a plurality of weighting parameters of the PDPC process is based on prediction information of a neighboring block adjacent to the current block.

17. A non-transitory computer-readable storage medium storing a program executable by at least one processor to perform:

decoding prediction information for a current block in a current picture that is a part of a coded video sequence, the prediction information indicating a specific inter prediction mode, a position dependent prediction combination (PDPC) process, and a coded residual for the current block;

determining at least one of (i) a partition of the current block to which a transform process is to be applied and (ii) a transform type of the transform process for the current block based on the prediction information;

performing the transform process on the coded residual to generate a decoded residual; and

reconstructing the current block based on the decoded residual.

18. The non-transitory computer-readable storage medium of claim 17, wherein the storing program is further to perform:

determining the transform type to be one of a plurality of sub-block transform (SBT) types, each of the plurality of SBT types being applied to a different part of the current block.

19. The non-transitory computer-readable storage medium of claim 18, wherein application of the one of the plurality of SBT types is independent of the specific inter prediction mode of the current block.

20. The non-transitory computer-readable storage medium of claim 18, wherein the part that the one of the plurality of SBT types is applied to is one of a rightmost part and a bottommost part of a plurality of parts of the current block.

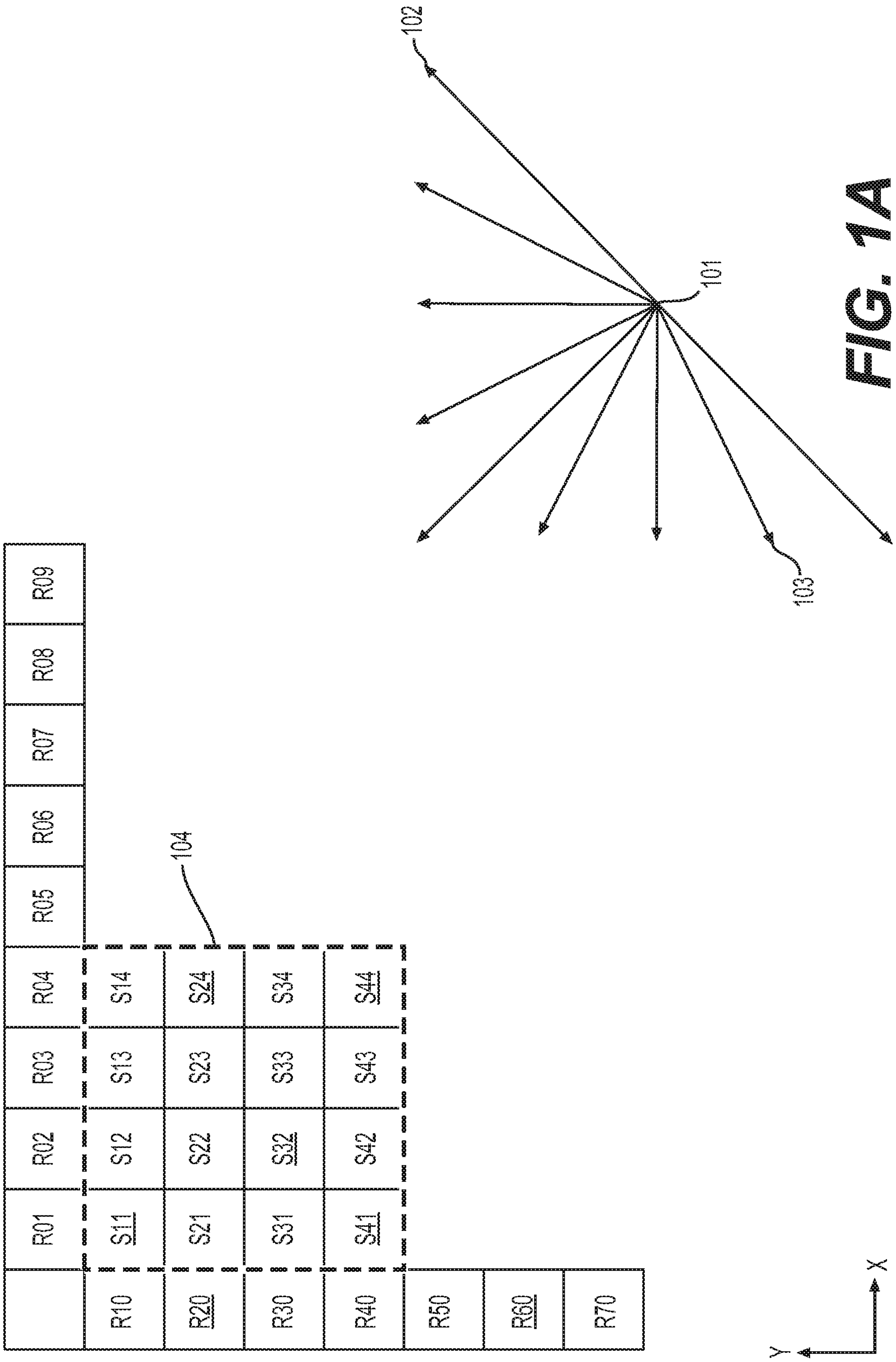


FIG. 1A
(Related Art)

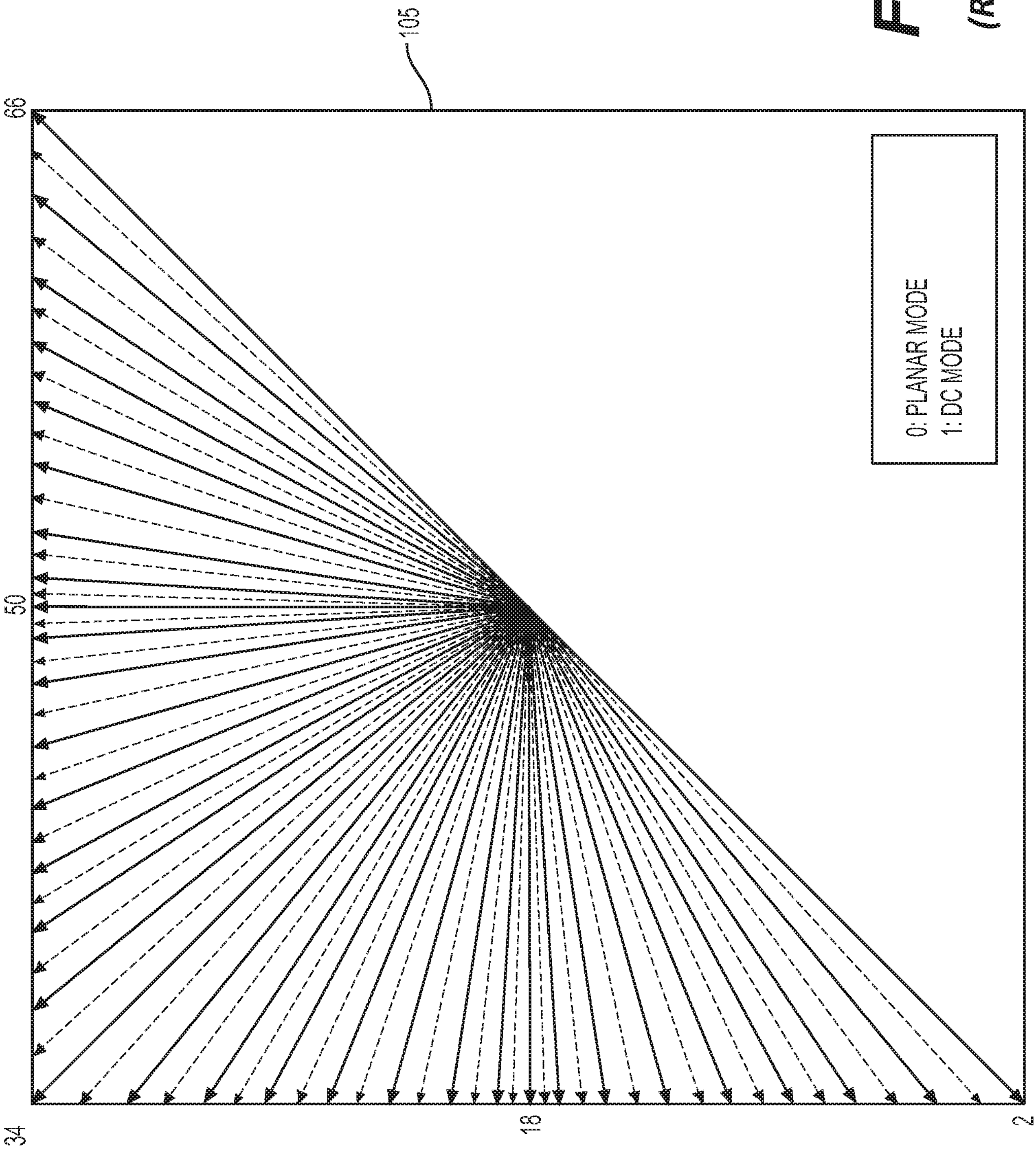


FIG. 1B
(Related Art)

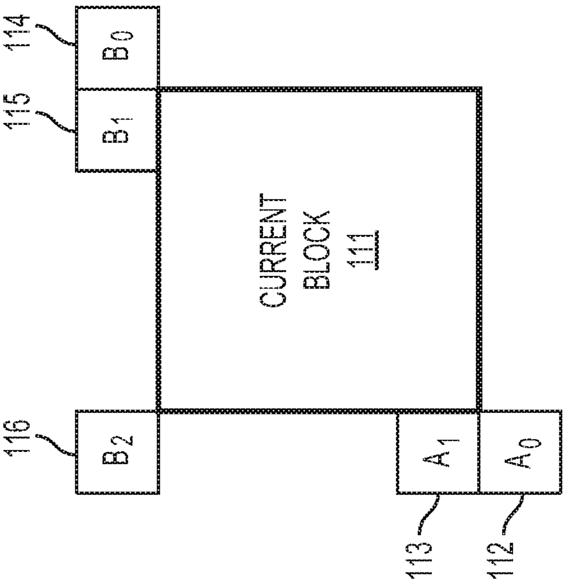


FIG. 1C
(Related Art)

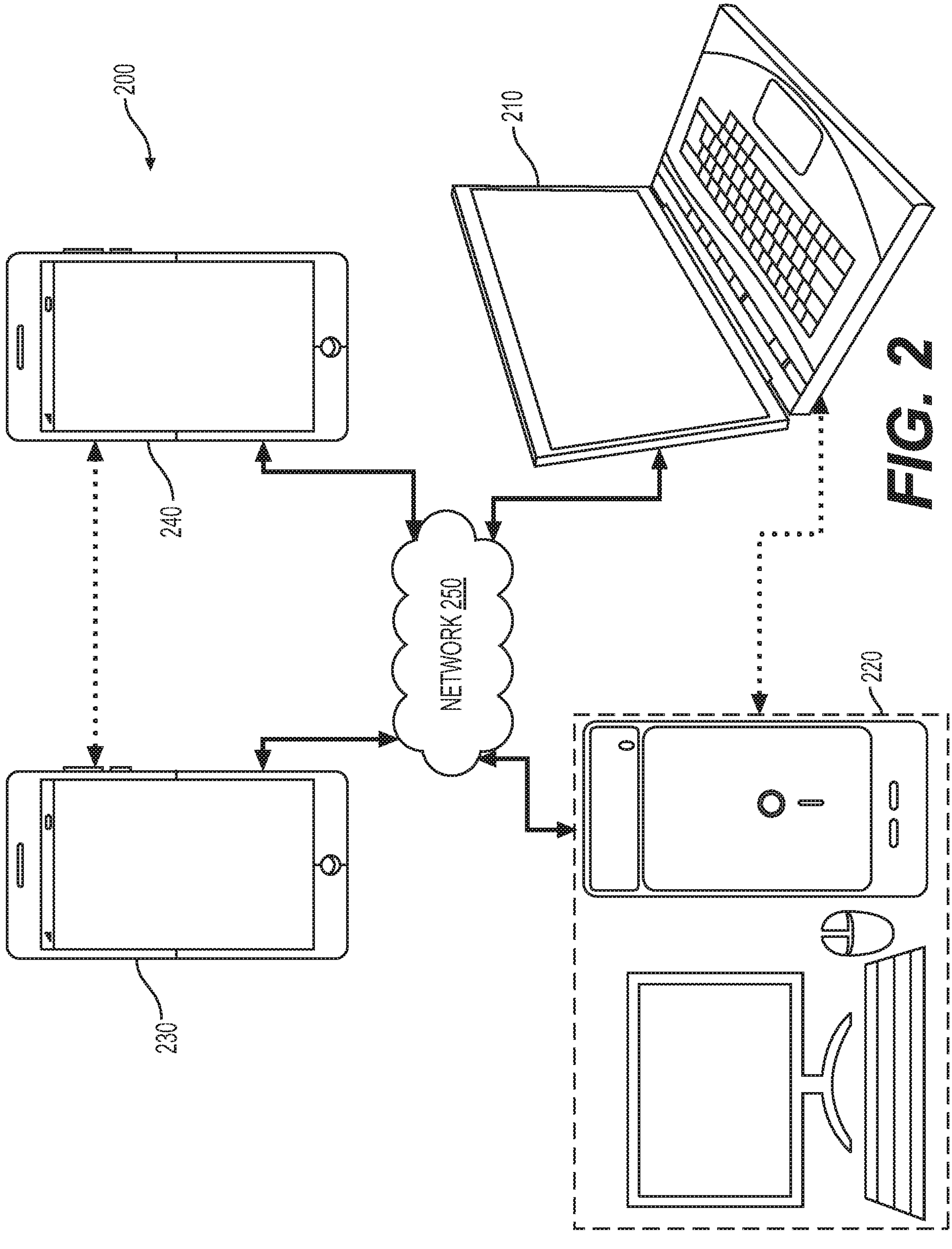


FIG. 2

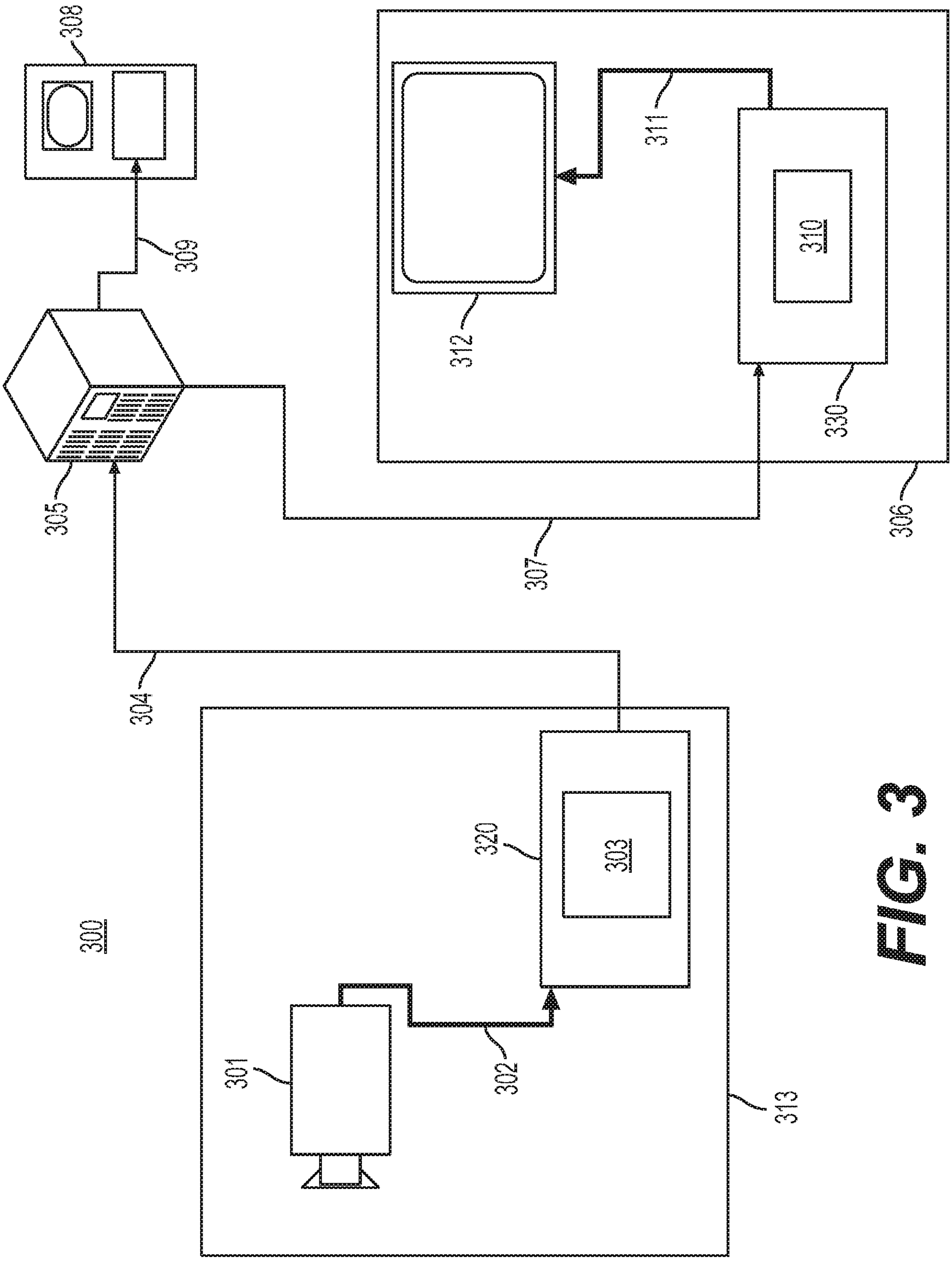


FIG. 3

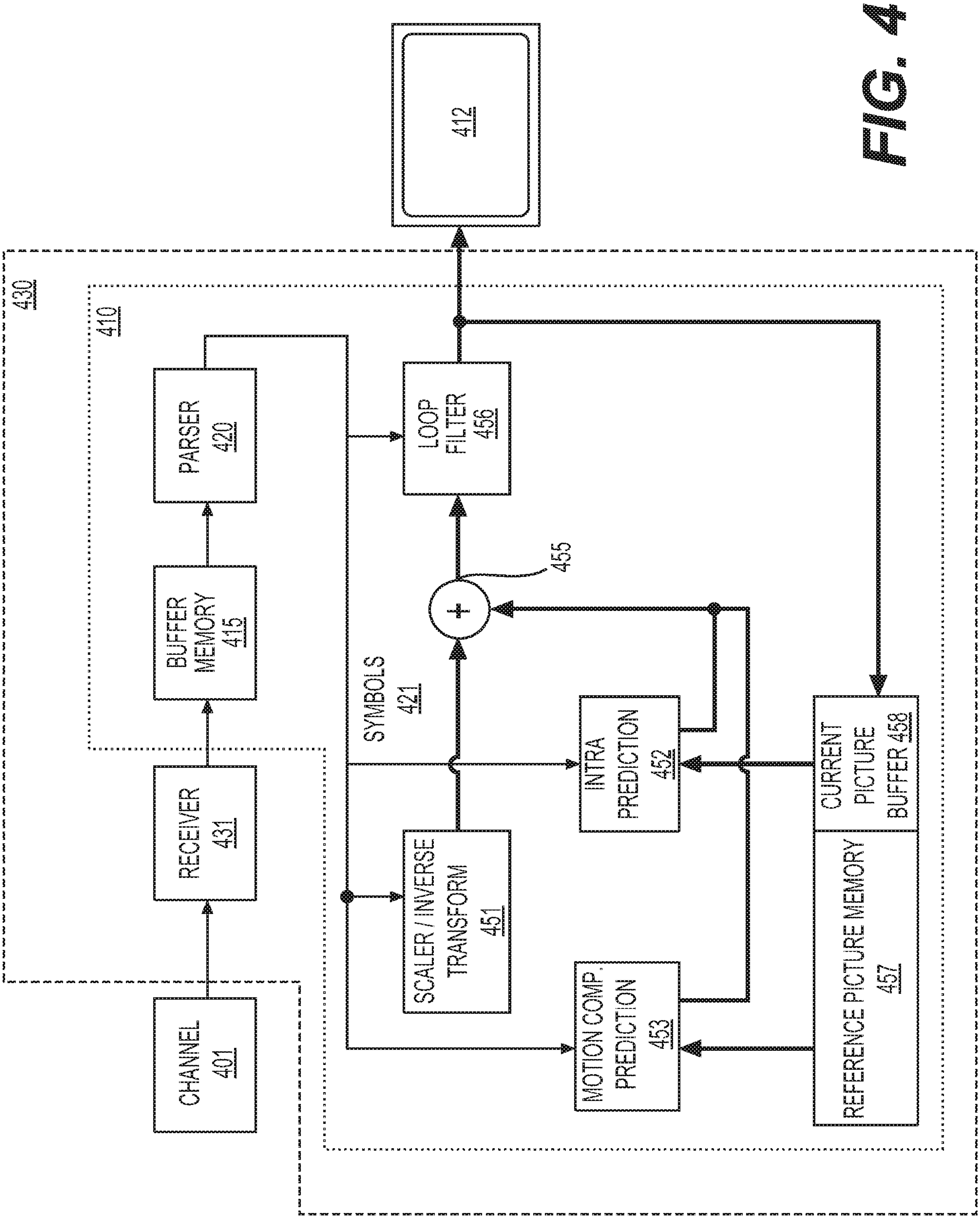

430
410
CHANNEL 401
RECEIVER 431
BUFFER MEMORY 415
PARSER 420
SYMBOLS 421
SCALER / INVERSE TRANSFORM 451
LOOP FILTER 456
455
+
412
INTRA PREDICTION 452
MOTION COMP. PREDICTION 453
CURRENT PICTURE BUFFER 458
REFERENCE PICTURE MEMORY 457

FIG. 4

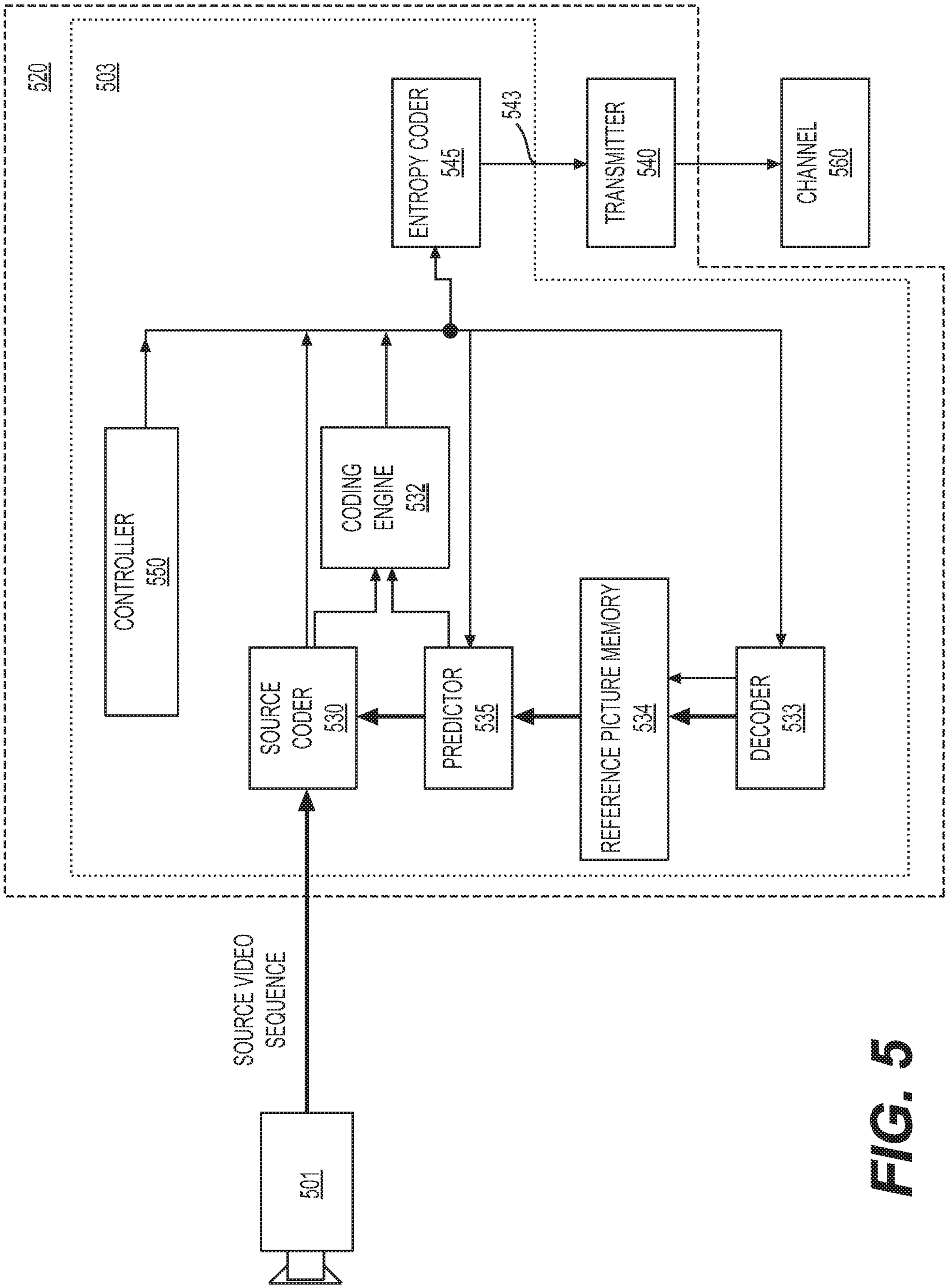

520
503
ENTROPY CODER
545
543
TRANSMITTER
540
CHANNEL
560
CONTROLLER
550
CODING
ENGINE
532
SOURCE
CODER
530
PREDICTOR
535
REFERENCE PICTURE MEMORY
534
DECODER
533
SOURCE VIDEO
SEQUENCE
501

FIG. 5

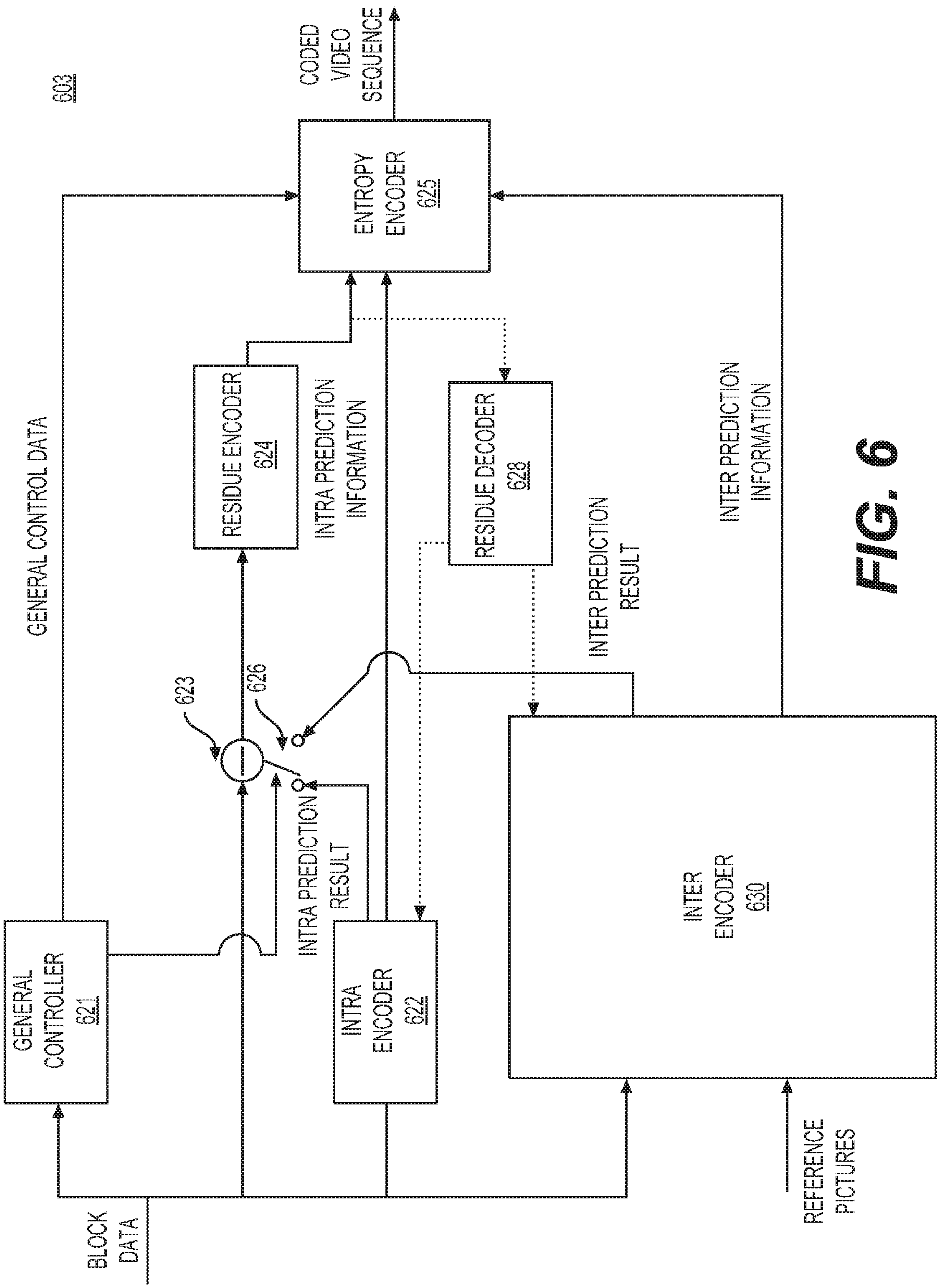

603
CODED VIDEO SEQUENCE
ENTROPY ENCODER 625
GENERAL CONTROL DATA
RESIDUE ENCODER 624
INTRA PREDICTION INFORMATION
RESIDUE DECODER 628
INTER PREDICTION INFORMATION
INTER PREDICTION RESULT
623
626
INTRA PREDICTION RESULT
GENERAL CONTROLLER 621
INTRA ENCODER 622
INTER ENCODER 630
BLOCK DATA
REFERENCE PICTURES

FIG. 6

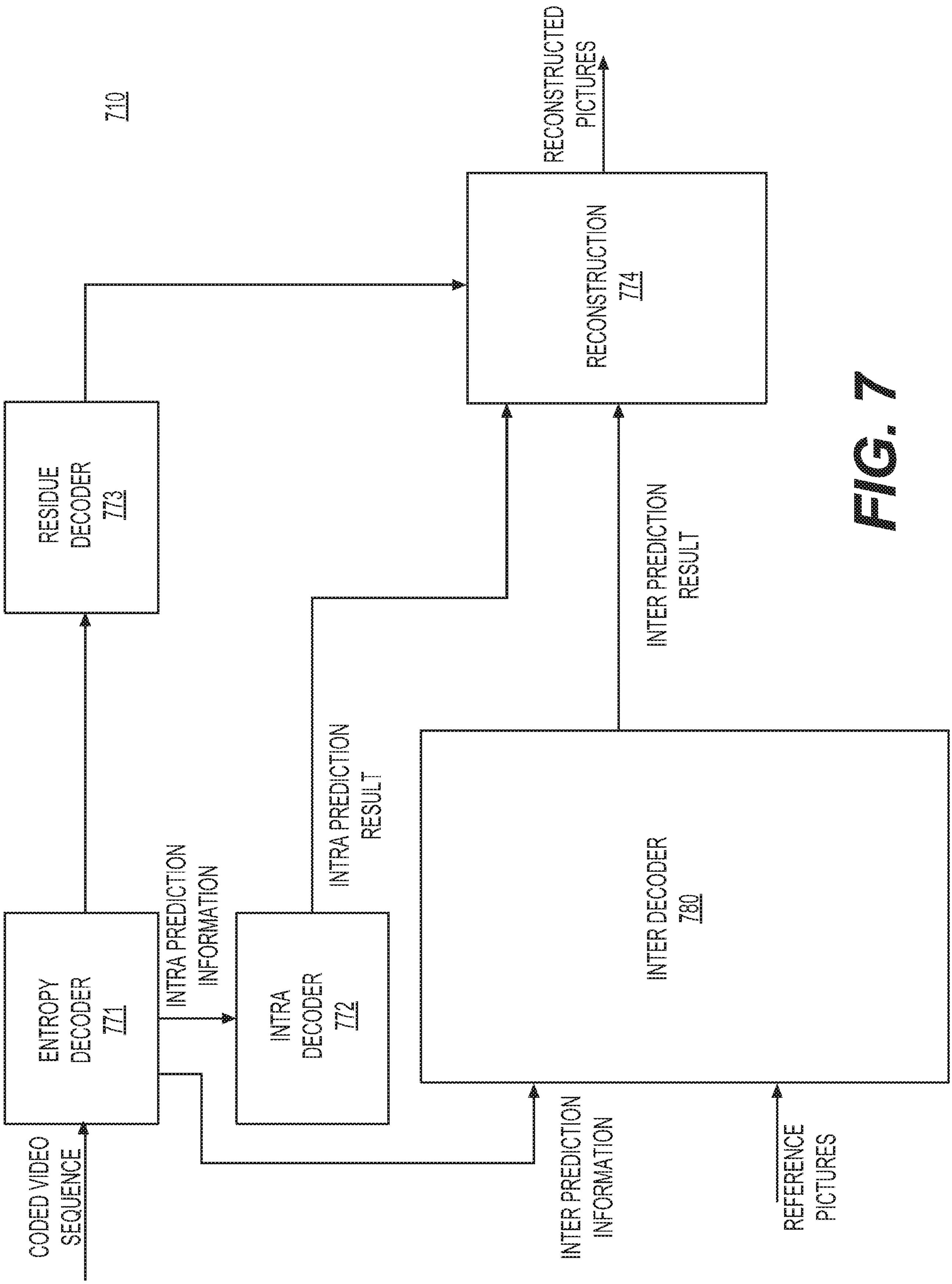

710
CODED VIDEO SEQUENCE
ENTROPY DECODER 771
INTRA PREDICTION INFORMATION
INTRA DECODER 772
INTRA PREDICTION RESULT
RESIDUE DECODER 773
RECONSTRUCTION 774
RECONSTRUCTED PICTURES
INTER PREDICTION RESULT
INTER DECODER 780
INTER PREDICTION INFORMATION
REFERENCE PICTURES

FIG. 7

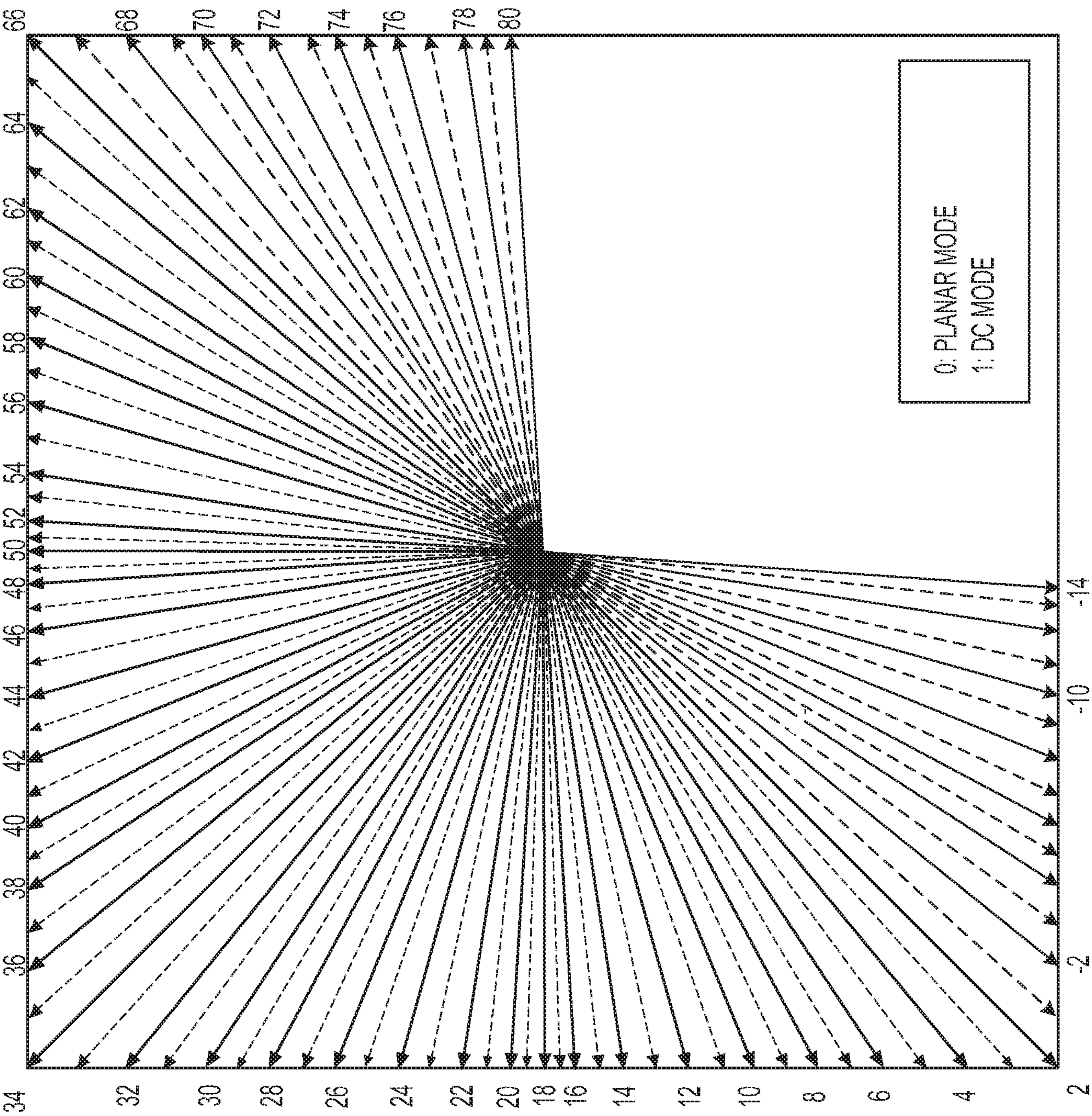

0: PLANAR MODE
1: DC MODE
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70
72
74
76
78
80
-2
-10
-14

FIG. 8

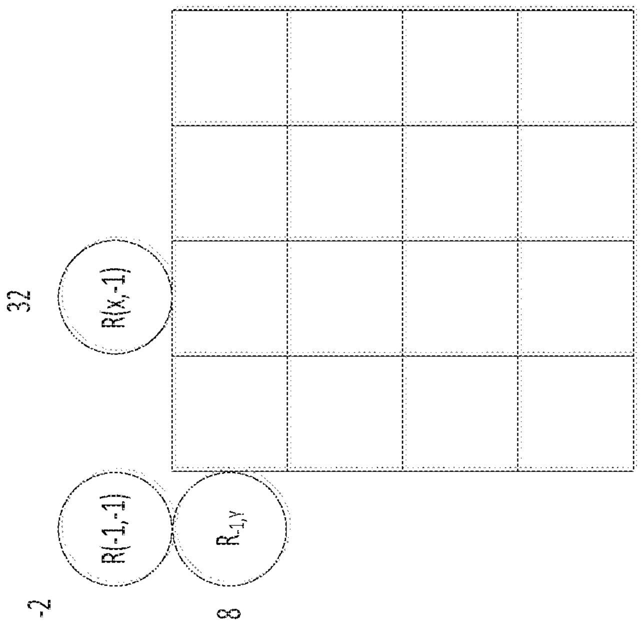


FIG. 9B

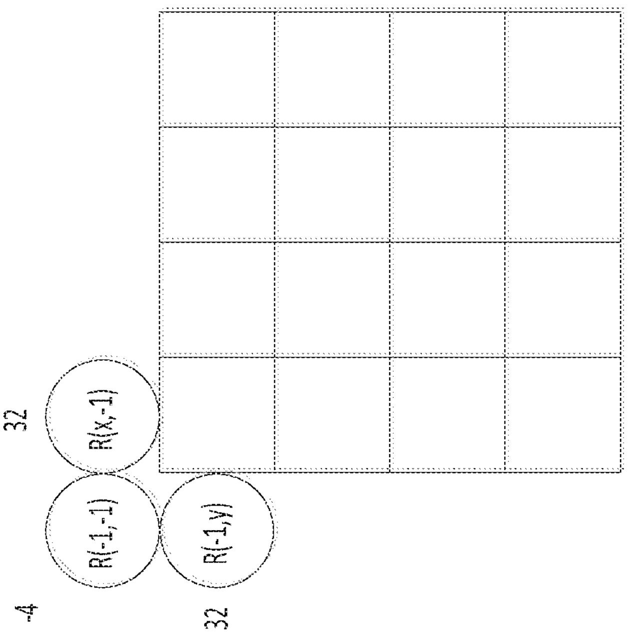


FIG. 9A

intra_chroma_pred_mode[xCb][yCb]	IntraPredModeY[xCb + cbWidth / 2][yCb + cbHeight / 2]				
	0	50	18	1	X (0 <= X <= 66)
0	66	0	0	0	0
1	50	66	50	50	50
2	18	18	66	18	18
3	1	1	1	66	1
4	0	50	18	1	X

FIG. 10

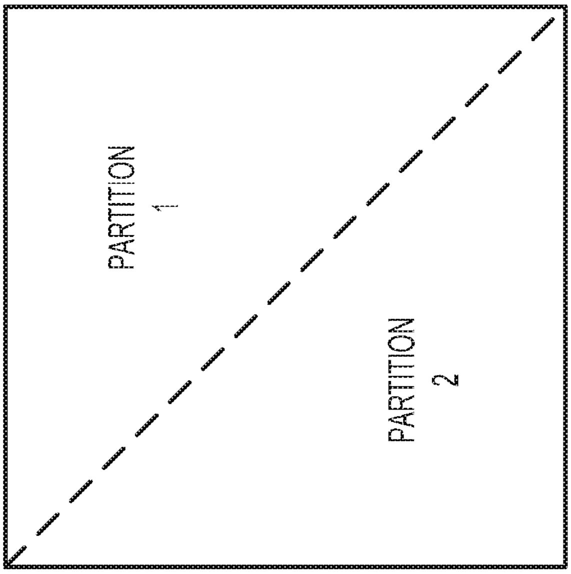


FIG. 11A

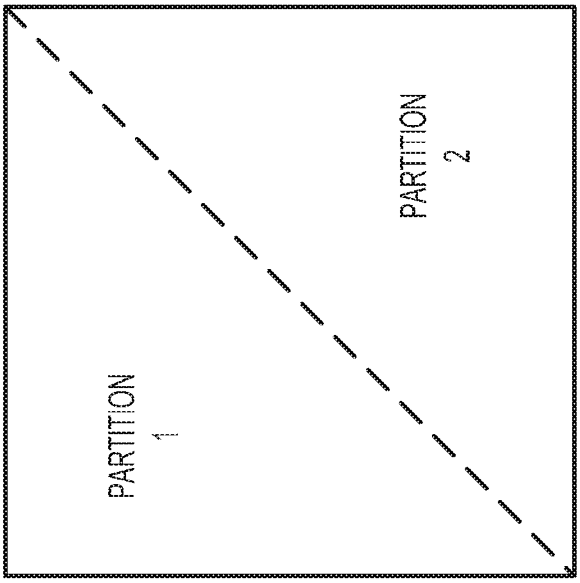
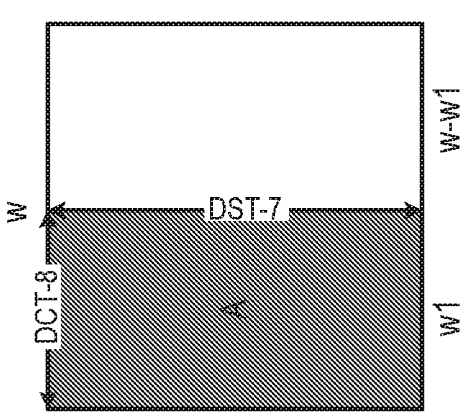
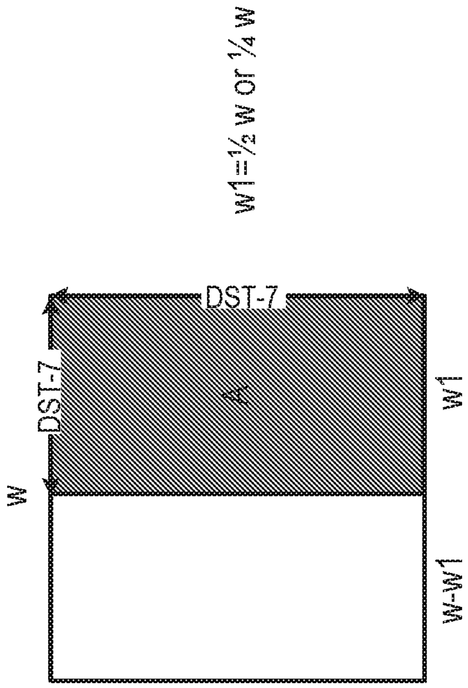


FIG. 11B



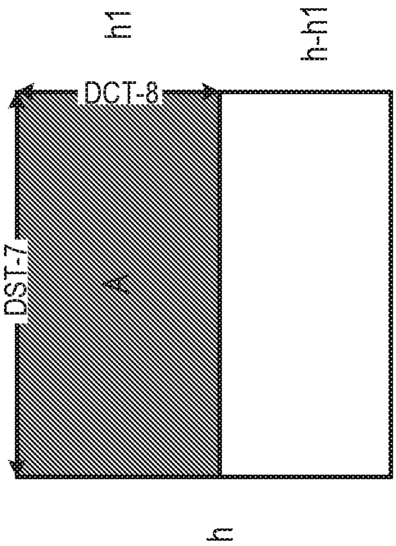
SVT-V, position 0

FIG. 12A



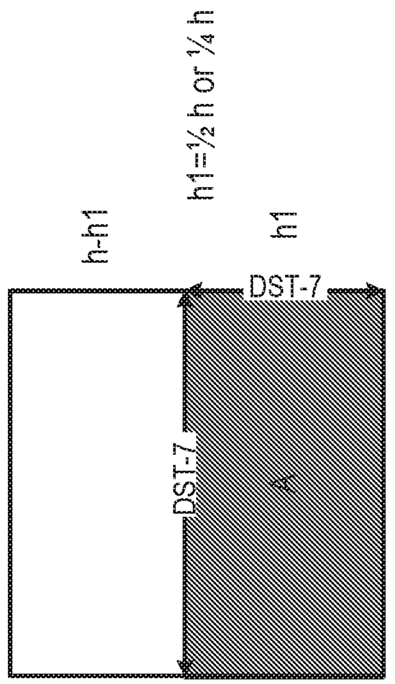
SVT-V, position 1

FIG. 12B



SVT-H, position 0

FIG. 12C



SVT-H, position 1

FIG. 12D

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \ldots, N-1$
DCT-2	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-8	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-7	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

FIG. 13

mts_idx[xTbY][yTbY][cIdx]	trTypeHor	trTypeVer
−1	0	0
0	1	1
1	2	1
2	1	2
3	2	2

FIG. 14

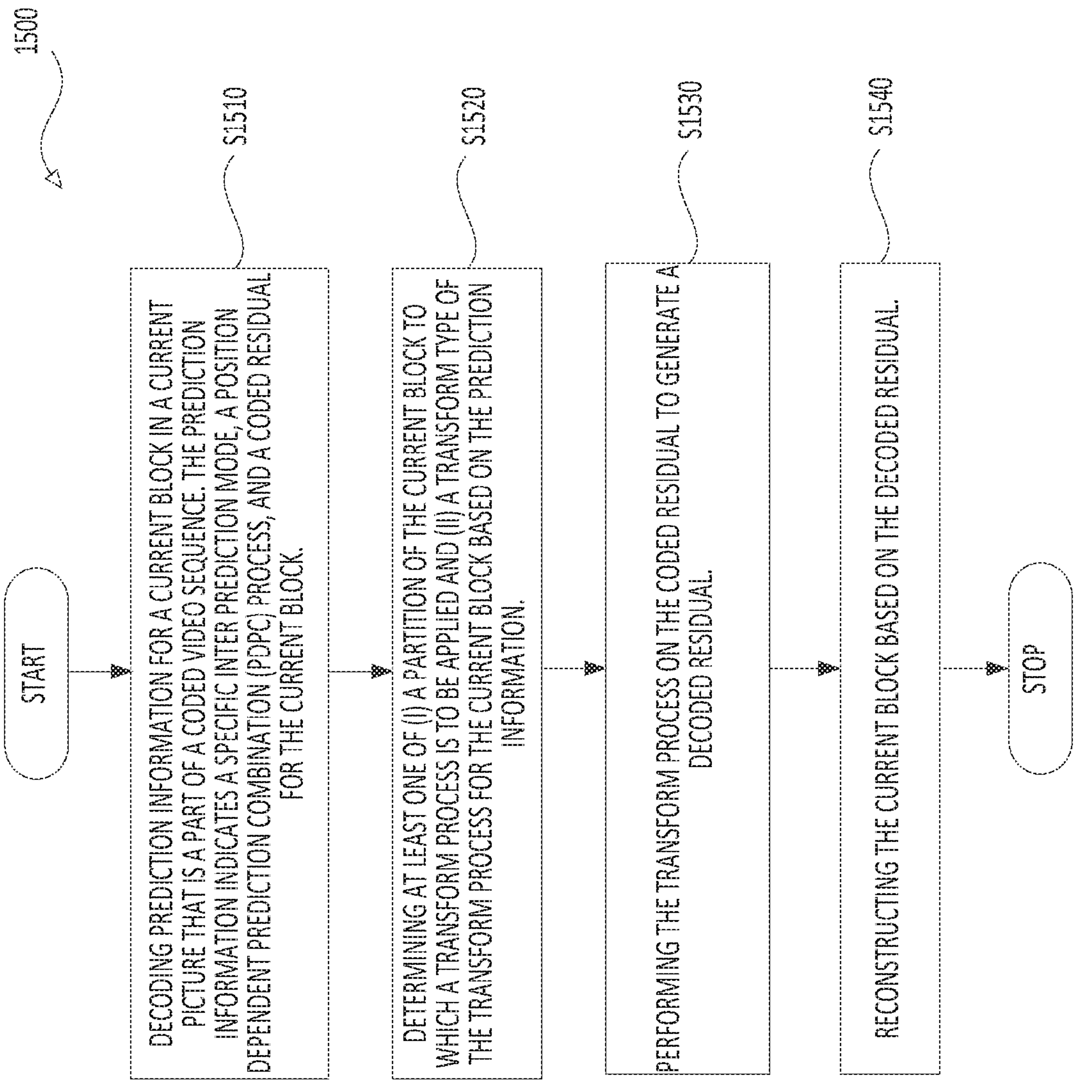

1500
START
DECODING PREDICTION INFORMATION FOR A CURRENT BLOCK IN A CURRENT PICTURE THAT IS A PART OF A CODED VIDEO SEQUENCE. THE PREDICTION INFORMATION INDICATES A SPECIFIC INTER PREDICTION MODE, A POSITION DEPENDENT PREDICTION COMBINATION (PDPC) PROCESS, AND A CODED RESIDUAL FOR THE CURRENT BLOCK.
S1510
DETERMINING AT LEAST ONE OF (I) A PARTITION OF THE CURRENT BLOCK TO WHICH A TRANSFORM PROCESS IS TO BE APPLIED AND (II) A TRANSFORM TYPE OF THE TRANSFORM PROCESS FOR THE CURRENT BLOCK BASED ON THE PREDICTION INFORMATION.
S1520
PERFORMING THE TRANSFORM PROCESS ON THE CODED RESIDUAL TO GENERATE A DECODED RESIDUAL.
S1530
RECONSTRUCTING THE CURRENT BLOCK BASED ON THE DECODED RESIDUAL.
S1540
STOP

FIG. 15

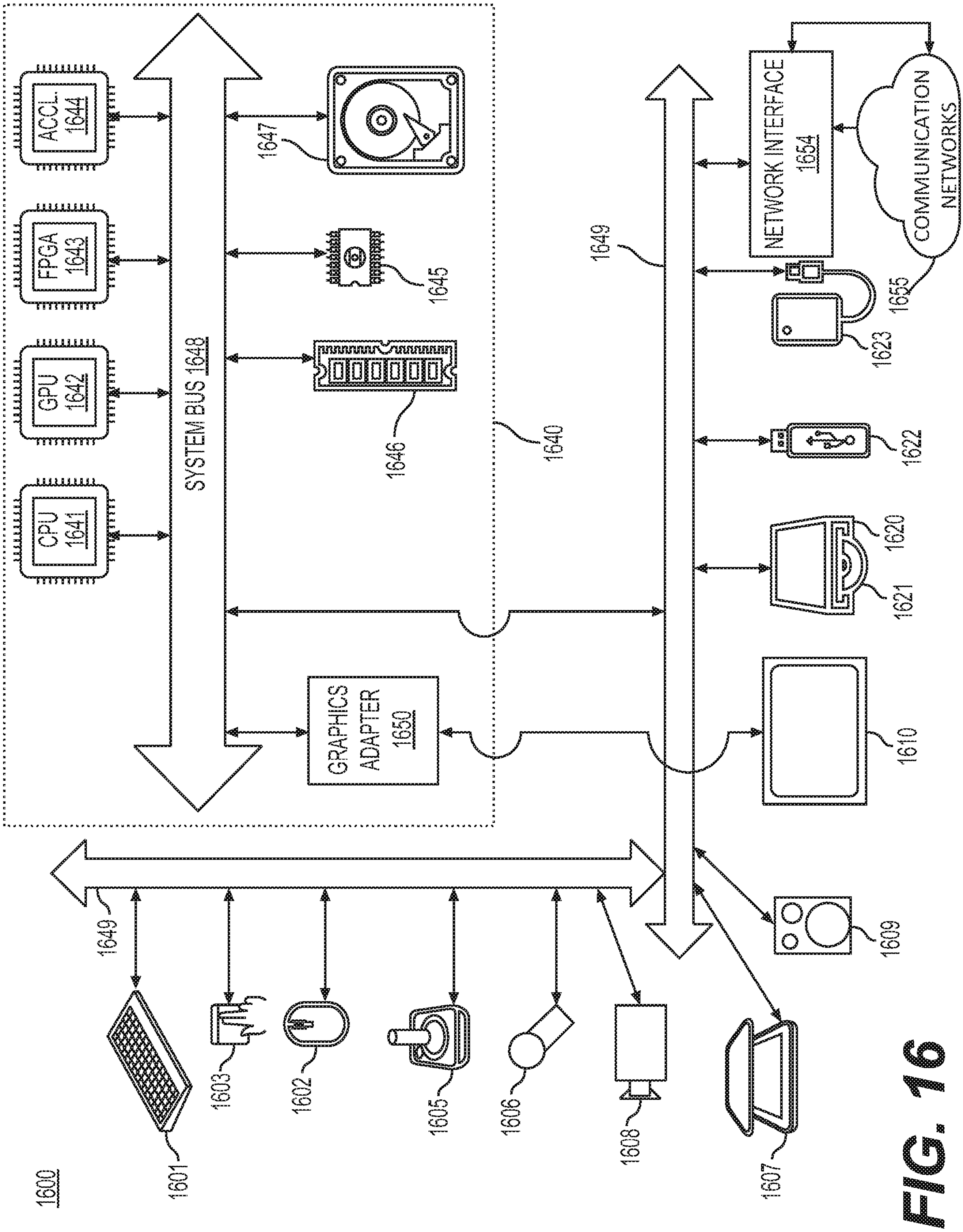

1600
ACCL. 1644
FPGA 1643
GPU 1642
CPU 1641
SYSTEM BUS 1648
1647
1645
1646
1640
GRAPHICS ADAPTER 1650
1649
NETWORK INTERFACE 1654
COMMUNICATION NETWORKS
1655
1623
1622
1620
1621
1610
1609
1601
1603
1602
1605
1606
1608
1607

FIG. 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US20/22163

A. CLASSIFICATION OF SUBJECT MATTER

IPC - H04N 19/593, 19/12, 19/50, 19/176 (2020.01)

CPC - H04N 19/593, 19/12, 19/50, 19/176, 19/11, 19/159

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 2017/0324643 A1 (QUALCOMM INCORPORATED) 09 November 2017; abstract; paragraphs [0015], [0029], [0063], [0065], [0086], [0125], [0136], [0160], [0161], [0185], [0194].	1-4, 6, 9-12, 14, 17-20 --- 5, 7-8, 13, 15-16
Y	WO 2016/123091 A1 (QUALCOMM INCORPORATED) 04 August 2016; paragraphs [0249], [0274].	5, 13
Y	CN 106797479 A (QUALCOMM INC) 31 May 2017; see machine translation	7, 15
Y	CN 108810552 A (HUAWEI TECHNOLOGY CO LTD) 13 November 2018; see machine translation.	8, 16

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

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
01 May 2020 (01.05.2020)	09 JUL 2020

Name and mailing address of the ISA/US	Authorized officer
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Shane Thomas Telephone No. PCT Helpdesk: 571-272-4300