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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 April 2020 (16.04.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/076513 A1

(51) International Patent Classification:
H04N 19/96 (2014.01) *H04N 19/176* (2014.01)
H04N 19/70 (2014.01) *H04N 19/44* (2014.01)

(21) International Application Number:
PCT/US2019/053221

(22) International Filing Date:
26 September 2019 (26.09.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/742,433 07 October 2018 (07.10.2018) US
16/567,718 11 September 2019 (11.09.2019) US

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

(72) Inventors: **XU, Meng**; 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). **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, 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,

(54) Title: METHOD AND APPARATUS FOR VIDEO CODING

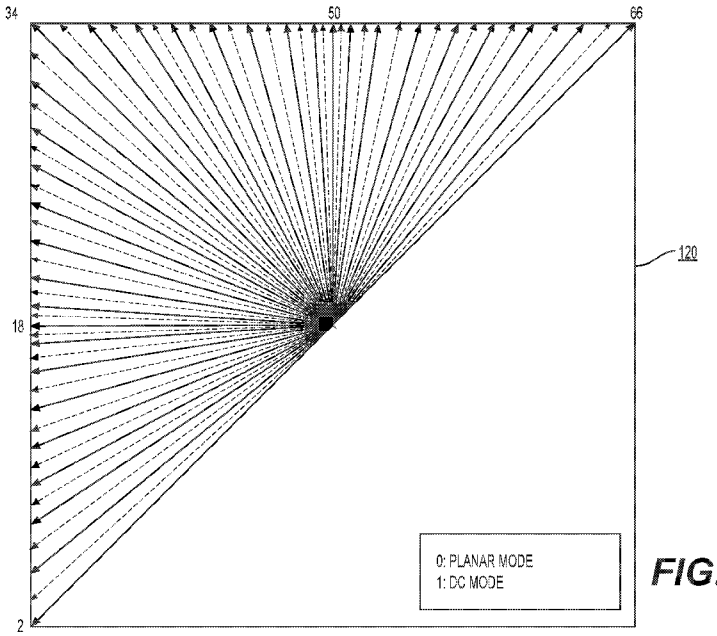


FIG. 1B

(57) Abstract: Aspects of the disclosure provide methods and apparatuses for video encoding/decoding. In some examples, an apparatus for video decoding includes receiving circuitry and processing circuitry. The processing circuitry decodes partition information of a block in a picture from a coded video bitstream. The partition information is indicative of a partition tree structure for a partition of the block into coding blocks. Then, the processing circuitry groups to-be reconstructed samples of the block into processing groups according to the partition of the block into the coding blocks. A shape of a combined region of a processing group is limited by constraints in the partition of the block into the coding blocks. Then, the processing circuitry reconstructs the to-be reconstructed samples of the block by multiple pipeline stages that process the processing groups in a pipeline processing manner.

WO 2020/076513 A1

[Continued on next page]

OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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))*

METHOD AND APPARATUS FOR VIDEO CODING

INCORPORATION BY REFERENCE

[0001] This present application claims the benefit of priority to U.S. Patent Application No. 16/567,718, "METHOD AND APPARATUS FOR VIDEO CODING" filed on September 11, 2019, which claims the benefit of priority to U.S. Provisional Application No. 62/742,433, "Coding unit partition constraints" filed on October 7, 2018. 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 itself be predicted.

[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 (120) 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 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 here is a technique henceforth referred to as "spatial merge".

[0021] Referring to FIG. 1C, a current block (131) comprises 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 (132 through 136, 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 receiving circuitry and processing circuitry. The processing circuitry decodes partition information of a block in a picture from a coded video bitstream. The partition information is indicative of a partition tree structure for a partition of the block into coding blocks. Then the processing circuitry groups to-be reconstructed samples of the block into processing groups according to the partition of the block into the coding blocks. A shape of a combined region of a processing group is limited by constraints in the partition of the block into the coding blocks. Then, the processing circuitry reconstructs the to-be reconstructed samples of the block by multiple pipeline stages that process the processing groups in a pipeline processing manner.

[0023] In an embodiment, the shape of the combined region of the processing group is limited to be one of a square shape and a rectangular shape. In another embodiment, the shape of the combined region of the processing group has no more than a fixed number of edges or has no more than a fixed number of vertices.

[0024] In another embodiment, the shape of the combined region of the processing group is limited by a constraint of no more than a fixed number of concatenating ternary tree (TT) splits in the partition tree structure for the partition of the block into the coding blocks.

[0025] In another embodiment, the combined region of the processing group has at least one of a width or a height equal to or larger than a fixed number.

[0026] In another embodiment, the processing group includes samples from multiple coding blocks, and a number of the multiple coding blocks is constrained to be no more than a limit value. In an example, the limit value is determined based on a size of the combined region.

[0027] In another embodiment, the shape of the combined region of the processing group is limited by constraints that limit usages of ternary tree split and binary tree split in the partition tree structure when the current picture is a non-intra picture. In an example, a usage of ternary tree split is allowed on a to-be-split block corresponding to a non-leaf node in the partition tree structure when the to-be split block is at coding tree unit (CTU) level or the to-be-split block is contained in a specific area of a coding tree unit.

[0028] In some examples, the combined region includes a portion that is outside a boundary of the picture.

[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 the method 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 (200) in accordance with an embodiment.

[0035] FIG. 3 is a schematic illustration of a simplified block diagram of a communication system (300) 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 illustrates an example of a block (801) that is partitioned using a QTBT structure (800).

[0041] FIG. 9A shows an example of a vertical center-side TT split.

[0042] FIG. 9B shows an example of a horizontal center-side TT split.

[0043] FIGs. 10A and 10B show partition examples according to some embodiments of the disclosure.

[0044] FIG. 11A shows some VPDU examples (1101)-(1104) having the same width (W) but different shapes.

[0045] FIG. 11B shows some VPDU examples (1111)-(1114) having the same height (H) but different shapes.

[0046] FIG. 12 shows an example of a TT split at CTU level and corresponding VPDU partition.

[0047] FIG. 13 shows an example of CTU partition according to an embodiment of the disclosure.

[0048] FIG. 14 shows an example of CTU partition and corresponding VPDU partition.

[0049] Fig. 15 shows a flow chart outlining a process example according to some embodiments of the disclosure.

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

DETAILED DESCRIPTION OF EMBODIMENTS

[0051] 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.

[0052] 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.

[0053] 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.

[0054] 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, and so on.

[0055] 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.

[0056] 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.

[0057] 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.

[0058] 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).

[0059] 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.

[0060] 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).

[0061] 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.

[0062] 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.

[0063] 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).

[0064] 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 the intra prediction unit (452) has generated to the output sample information as provided by the scaler / inverse transform unit (451).

[0065] 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.

[0066] 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.

[0067] 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.

[0068] 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.

[0069] 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.

[0070] 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.

[0071] 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.

[0072] 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).

[0073] 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.

[0074] 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 search range, 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.

[0075] 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.

[0076] 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 parser (420) may not be fully implemented in the local decoder (533).

[0077] 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.

[0078] 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.

[0079] 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).

[0080] 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).

[0081] 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.

[0082] 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.

[0083] 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).

[0084] 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:

[0085] 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.

[0086] 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.

[0087] 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.

[0088] 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.

[0089] 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.

[0090] 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.

[0091] 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.

[0092] 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.

[0093] Further, a merge mode technique can be used in the inter-picture prediction to improve coding efficiency.

[0094] 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 quadtree 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.

[0095] 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.

[0096] 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.

[0097] 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.

[0098] 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.

[0099] 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.

[0100] 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.

[0101] 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.

[0102] 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.

[0103] 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.

[0104] 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.

[0105] 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).

[0106] The inter decoder (780) is configured to receive the inter prediction information, and generate inter prediction results based on the inter prediction information.

[0107] The intra decoder (772) is configured to receive the intra prediction information, and generate prediction results based on the intra prediction information.

[0108] 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).

[0109] 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.

[0110] 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 (503), and the video decoders (310), (410), and (710) can be implemented using one or more processors that execute software instructions.

[0111] Generally, a picture is partitioned into a plurality of blocks for encoding and decoding. The partition techniques that partition the picture into blocks can influence coding performance. Aspects of the disclosure provide techniques to constrain the partitions for the blocks in order to improve coding performance.

[0112] In some examples, according to the HEVC standard, a picture can be split into a plurality of coding tree units (CTUs). Further, a CTU may be partitioned into coding units (CUs) by using a quad-tree (QT) structure denoted as a coding tree in order to adapt to various local characteristics of the picture. Each CU corresponds to a leaf of the quad-tree structure. The decision whether to code a picture area using an inter-picture prediction (also referred to as a temporal prediction or an inter-prediction type), an intra-picture prediction (also referred to as a spatial prediction, or an intra-prediction type), and the like is made at the CU level. Each CU can be further split into one, two or four prediction units (PUs) according to a PU splitting type which has its root at the CU level. For each PU, the same prediction process is applied and the relevant prediction information is transmitted to a decoder on a PU basis. After obtaining prediction residual data by applying the prediction process based on the PU splitting type, the CU can be partitioned into transform units (TUs) according to another QT structure which has its root at the CU level. According to HEVC standard, in some examples, the partition conceptions include CU, PU, and TU. In some examples, a CU, a PU that is associated with the CU, and a TU that is associated with the CU may have

different block sizes. In addition, a CU or a TU has to be in a square shape in the QT structure, while a PU can have a square or a rectangular shape.

[0113] It is noted that, in some implementation examples, encoding/decoding are performed on blocks. For example, a coding tree block (CTB), a coding block (CB), a prediction block (PB), and a transform block (TB) can be used to specify, for example, 2D sample arrays of one color component associated with a corresponding CTU, CU, PU, and TU, respectively. For example, a CTU can include one luma CTB and two chroma CTBs, a CU can include one luma CB and two chroma CBs.

[0114] In order to outperform HEVC in compression capability, many other partition structures have been proposed for next generation video coding standard beyond HEVC, i.e., so-called Versatile Video Coding (VVC). One of these proposed partition structures is called QTBT structure which employs both quad-tree (QT) and binary-tree (BT). Compared to the QT structure in HEVC, the QTBT structure in VVC removes separation among the CU, PU and TU concepts. In other words, the CU, the PU that is associated with the CU, and the TU that is associated with the CU have same block size in the QTBT structure in VVC. In addition, the QTBT structure supports more flexibility for CU partition shapes. A CU can have a square shape or a rectangle shape in the QTBT structure.

[0115] FIG. 8 illustrates an example of a block (801) that is partitioned using a QTBT structure (800). In the FIG. 8 example, QT splits are represented by solid lines and BT splits are represented by dotted lines. Specifically, the QTBT structure (800) includes nodes corresponding to various blocks during the partitions for in the block (801). When a node in the QTBT structure (800) has branches, the node is referred to as a non-leaf node, and when a node in the QTBT structure (800) has no branches, the node is referred to as a leaf node. The non-leaf node corresponds to an intermediate block that is further split, and the leaf node corresponds to a final block without further splitting. When a non-leaf node has four branches, a QT split splits a block corresponding to the non-leaf node into four smaller blocks of equal size. When a non-leaf node has two branches, a BT split splits a block corresponding to the non-leaf node into two smaller blocks of equal size. The BT split has two splitting types, a symmetric horizontal split and a symmetric vertical split. In some examples, for each non-leaf BT split node, a flag is signaled (e.g., in the coded video bitstream) to indicate a splitting type (i.e., "0" for symmetric horizontal split or "1" for symmetric vertical split). For a QT split node in the QTBT structure (800), the QT split splits a block corresponding to the QT split node both horizontally and vertically to produce four smaller blocks with an equal size.

[0116] Specifically, in the FIG. 8 example, the QTBT structure (800) includes a root node (850) corresponding to the block (801). The root node (850) has four branches that respectively generate nodes (861)-(864), thus the block (801) is split by a QT split into four blocks (811)-(814) of equal sizes. The nodes (861)-(864) respectively correspond to four blocks (811)-(814).

[0117] Further, the node (861) has two branches that generate nodes (871) and (872), the node (862) has two branches that generate nodes (873) and (874), and the node (863) has four branches that generate nodes (875)-(878). It is noted that the node (864) has no branch, and thus is a leaf node. It is also noted that the node (862) is a non-leaf BT split node with a splitting type "1", and the node (863) is a non-leaf BT split node with a splitting type "0". Accordingly, the block (811) is split by a BT split vertically into two blocks (821)-(822), and the block (812) is split by a BT split horizontally into two blocks (823)-(824), and the block (813) is split by a QT split into four blocks (825)-(828).

[0118] Similarly, the node (871) has two branches that generate nodes (881)-(882), the node (875) has two branches that generate nodes (883)-(884), the node 878 has two branches that generate nodes (885)-(886), the node (884) has two branches that generate nodes (891)-(892). Accordingly, the corresponding blocks are split into smaller blocks as shown in FIG. 8. Then nodes (881), (882), (872), (873), (874), (883), (891), (892), (876), (877), (885) and (886) are similar to the node (864) that has no branches, and are thus leaf nodes.

[0119] In the FIG. 8 example, the blocks corresponding to the leaf nodes (864), (881), (882), (872), (873), (874), (883), (891), (892), (876), (877), (885) and (886) are not further split, and are CUs that are used for prediction and transform processing. In some examples, such as VVC, the CUs are respectively used as PUs and TUs.

[0120] In addition, in the QTBT structure, a CU can include CBs of different colour components. For example, in a 4:2:0 format, one CU can include one luma CB and two chroma CBs in a P or B slice. However, in some other embodiments, a CU may include CBs in a single component. For example, in an I slice, one CU may include one luma CB or two chroma CBs. That is to say, the QTBT structure supports an ability for luma and chroma to have different partitioning structures.

[0121] In addition to the QTBT structure described above, another splitting structure called multi-type-tree (MTT) structure is also used in VVC and can be more flexible than QTBT structure. In an MTT structure, other than quad-tree and binary-tree, horizontal and vertical center-side triple-tree (TT) splits are introduced, as shown in FIG. 9A and FIG. 9B.

A triple-tree split can also be referred to as a ternary tree (TT) split, or a ternary split. FIG. 9A shows an example of a vertical center-side TT split. For example, a block (910) is vertically split into three sub-blocks (911)-(913) where the sub-block (912) is located in the middle of the block (910). FIG. 9B shows an example of a horizontal center-side TT split. For example, a block (920) is horizontally split into three sub-blocks (921)-(923) where the sub-block (922) is located in the middle of the block (920). Similar to BT split, in TT split, a flag can be signaled, for example in the video bitstream from the encoder side to the decoder side, to indicate a splitting type (i.e., a symmetric horizontal split or a symmetric vertical split). In an example, "0" indicates a symmetric horizontal split and "1" indicates a symmetric vertical split.

[0122] One of MTT structures is called QTBTTT structure that includes quad-tree, binary-tree, and ternary-tree splitting types. Similar to QTBT structure, a QTBTTT structure also supports luma and chroma to have different structures. For examples, in an I slice, a QTBTTT structure used to partition a luma CTB can be different from a QTBTTT structure used to partition a chroma CTB. However, in a P or B slice, a luma CTB can share same QTBTTT structure with a chroma CTB in one CTU. This means that when separated tree structures are enabled, a CU consists of one luma CB or two chroma CBs; and when separated tree structure is disabled, a CU consists of all three CBs, i.e., one luma CB and two chroma CBs.

[0123] In some examples, such as a VCC test model (VTM), a max TU is set to 64x64 for luma and 32x32 for chroma, thus 64 is set as a width/height threshold for luma and 32 is set as a width/height threshold for chroma. For a block with either width or height larger than the width/height threshold (64 for luma, 32 for chroma), the block is implicitly split into smaller blocks based on threshold. For example, a 128x128 block is implicitly split into four 64x64 TUs, a 128x64 CU is implicitly split into two 64x64 TUs, and a 128x32 CU is implicitly split into two 64x32 TUs.

[0124] In some embodiments, implicit QT or BT are used at picture boundary. For example, when a CTU crosses the picture boundary, the CTU includes areas both inside and outside of the picture boundary, the CTU is enforced to be split into smaller units. In HEVC, a boundary coding tree unit is split recursively using the implicit quadtree splits. The split flag is implied and not signaled in an example.

[0125] In some embodiments, combined techniques, such as QT+BT, QT+BT+TT, QT+BT+ABT, QT+BTS and the like, are used at the picture boundary. In an example, at each split level, when multiple split techniques, such as QT and non-QT splits are both

available at the picture boundary, one or more flags are signaled to indicate which split is applied. In an example, such as VTM version 2, QT+BT is used in the partitioning of the picture boundary CUs.

[0126] In some embodiments, QT, BT and TT splits are constrained in some scenarios. For example, in VTM version 2, in the center part of a TT split, a BT split along the TT split direction is prohibited. In another example, separate luma/chroma trees for intra slices use an implicit split of 64x64. In another example, ternary tree split along the direction with length greater than 64 is prohibited.

[0127] In addition to various partitioning structures, a concept called virtual pipeline data unit (VPDU) is used to assist encoding/decoding in some examples. VPDU is defined as a non-overlapping MxM-luma (L)/NxN-chroma (C) unit in a picture. In some hardware decoder implementations, successive VPDUs are processed by multiple pipeline stages at the same time. Further, different stages of the multiple pipeline stages process different VPDUs simultaneously. In some examples, the size of a VPDU is almost proportional to a buffer size in most pipeline stages. In an example, when the VPDU size is relatively small, the buffer can be implemented using for example on-chip buffer that can be accessed with fast speed compared to off-chip buffer. Thus, in some examples, the VPDU size is desired to keep relatively small.

[0128] In some HEVC hardware decoders, a VPDU size is set to be a maximum transform block (TB) size. Enlarging a maximum TB size from 32x32-L/16x16-C in HEVC to 64x64-L/32x32-C in current VVC can bring a coding gain, which expectedly results in 4X of VPDU size in comparison with HEVC. Further, the VVC uses TT splits and BT splits in addition to QT splits to partition a CTU into CUs, and can achieve additional coding gains. In some examples, TT and BT splits can be applied to 128x128-L/64x64-C coding tree blocks (CTUs) recursively, which leads to 16X of VPDU size (128x128-L/64x64-C) in comparison with HEVC.

[0129] To reduce the VPDU size in VVC, some constraints for TT splits and BT splits can be used, and the VPDU size is defined as 64x64-L/32x32-C. In a constraint example, for each CTU, the processing order of CUs in the CTU shall not leave a VPDU and re-visit the VPDU later. In another constraint example, for each VPDU that includes one or multiple CUs, the CUs should be completely contained in the VPDU. In another constraint example, for each CU containing one or more VPDUs, the VPDUs should be completely contained in the CU.

[0130] In some embodiments, a VPDU size can be signaled in a coded video bitstream, such as in the sequence parameter set (SPS), the picture parameter set (PPS) and the like. In an example, the VPDU size can be signaled in terms of luma samples. In another example, the VPDU size can be signaled in terms of chroma samples.

[0131] In some embodiments, a VPDU size is predefined and stored in both encoder and decoder, thus the VPDU size is not signaled. In an example, the VPDU size can be stored in profile and/or level definitions. In some examples, the VPDU size can be stored in terms of luma or chroma samples.

[0132] In some embodiments, VPDUs share same size, and they may be in different shapes. For example, a VPDU size is 4096 in terms of luma samples. The VPDU can have a square shape of 64x64 or a rectangular shape of 32x128. The VPDU can also have other shapes, such as an L shape, or reverse L shape, etc., as long as the VPDU size is 4096 in terms of luma samples.

[0133] In some embodiments, a transform unit (TU) may not cross a VPDU boundary. This means that the TU is completely contained in one VPDU.

[0134] FIGs. 10A and 10B show partition examples according to some embodiments of the disclosure.

[0135] In FIG. 10A, a CTU (1010) has a size of 128x128 and the size of a VPDU is 4096 in terms of luma samples. The CTU (1010) is first split by a vertical TT split into blocks (1011), (1012) and (1013). The block (1012) is split by a horizontal TT split into blocks (1021), (1022) and (1023). The block (1022) is split by a vertical BT split into blocks (1031) and (1032). The blocks (1011), (1013), (1021), (1023), (1031) and (1032) are leaf nodes, and are CUs in the FIG. 10A example. In the FIG. 10A example, the block (1011) is named CU1, has a size of 32x128, and fits in a first VPDU. The block (1021) is named CU2, has a size of 64x32. The block (1031) is named CU3, and has a size of 32x64. The blocks (1021) and (1031) together fit into a second VPDU. The block (1032) is named CU4, and has size of 32x64. The block (1023) is named CU5, and has a size of 64x32. The blocks (1032) and (1023) together fit into a third VPDU. The block (1013) is named CU6, has a size of 32x128, and fits into a fourth VPDU. Accordingly, the CTU (1010) can be covered by four non-overlapping VPDUs. In other words, the partition structure in FIG. 10A is valid.

[0136] In some embodiments, a transform unit (TU) may cross a VPDU boundary when the TU doesn't have significant (e.g., non-zero) coefficients. Boundary crossing between a TU and a VPDU means that one part of the TU is in one VPDU while another part

of the TU is in another VPDU. That is to say, the TU is not completely contained in one VPDU.

[0137] FIG. 10B shows a partition example of a CTU in which not every TU is completely contained in a VPDU and thus boundary crossing does happen. In the FIG. 10B example, a CTU (1050) has a size of 128x128 and the size of a VPDU is 4096 in terms of luma samples. The CTU (1050) is first split by a vertical TT split into blocks (1051), (1052) and (1053). The block (1052) is split by a horizontal TT split into blocks (1061), (1062) and (1063). The blocks (1051), (1053), (1061), (1062) and (1063) are leaf nodes, and are CUs in the FIG. 10B example. In the FIG. 10B example, the block (1051) is named CU1, and has a size of 32x128. The block (1061) is named CU2, and has a size of 64x32. The block (1062) is named CU3, and has a size of 64x64. The block (1063) is named CU4, and has size of 32x64. The block (1053) is named CU5, and has a size of 32x128. As the VPDU size is 4096, the CU1 fits into a first VPDU and the CU2 fits into a second VPDU, respectively. However, the CU2 size is 2048 and the CU3 size is 4096, the CU3 crosses a VPDU boundary, which means half of the CU3 is in within the second VPDU and the other half of the CU3 is within a third VPDU. In an example, TU of CU3 doesn't have significant (non-zero) coefficients, then the partition example in FIG. 10B can be allowed. In another example, when the TU of CU3 has significant (non-zero) coefficients, the partitioning example in FIG. 10B is not allowed.

[0138] In some embodiments in which separate luma/chroma trees are used, chroma partitions may be inferred from luma partitions and thus is not signaled, so that luma CUs and chroma CUs may share same VPDUs.

[0139] In the above examples, non-square VPDUs can be used. Each of the VPDUs includes the same number of samples, but the shapes of the VPDUs may or may not be square. The non-square VPDUs provide flexible in CTU partitions, but may result in complicated shapes for VPDUs which require additional buffer space, for example, for specifying the shapes of the VPDUs, and add complexity for hardware implementation.

[0140] Aspects of the disclosure provide methods to constraints to CTU partition in order to limit the complicated shapes for the VPDUs, and alleviate the complexity for hardware implementation. The methods may be used separately or combined in any order. Further, embodiments may be implemented by processing circuitry (e.g., one or more processors or one or more integrated circuits). In one example, one or more processors execute a program that is stored in a non-transitory computer-readable medium to perform these methods. In addition, it is noted that these methods are not dependent on the definition

of VPDU and can be applied to all architectures. For convenience, the definition of VPDU described above is used in following descriptions.

[0141] For the following description, the VPDU is not limited to square or rectangular shape, the width (W) of a VPDU is defined as the maximum width of sample rows in the VPDU, the height (H) of a VPDU is defined as the maximum height of sample columns in the VPDU.

[0142] FIG. 11A shows some VPDU examples (1101)-(1104) having the same width (W) but different shapes; and FIG. 11B shows some VPDU examples (1111)-(1114) having the same height (H) but different shapes.

[0143] According to some aspects of the disclosure, additional constraints are applied to VPDUs.

[0144] In some embodiments, the shapes of the VPDU are constrained. For example, only certain VPDU shapes are allowed. In an example, the allowed VPDU shapes are signaled in coded video bitstream from the encoder side to the decoder side. In another example, the allowed VPDU shapes are pre-determined at both the encoder side and the decoder side. It is noted that, in some embodiments, the boundaries or edges of a VPDU are in the vertical direction or horizontal direction.

[0145] In an embodiment, only square or rectangular VPDUs are allowed.

[0146] In another embodiment, the VPDU shapes are limited according to the number of edges. For example, the number of edges of an allowed VPDU should be equal to or less than a fixed number, such as such as 4 or 6, etc. In an example, when the number of edges of an allowed VPDU is limited to be equal to or less than 4, the shapes of the allowed VPDUs can be square or rectangle. In another examples, when the number of edges of an allowed VPDU is limited to be equal to or less than 6, the shapes of the allowed VPDUs can be square, rectangle, L-shape, or reverse L-shape.

[0147] In another embodiment, the VPDU shapes are limited according to the number of vertices. For example, the number of vertices of an allowed VPDU should be equal to or less than a fixed number, such as such as 4 or 6, etc. In an example, when the number of vertices of an allowed VPDU is limited to be equal to or less than 4, the shapes of the allowed VPDUs can be square or rectangle. In another examples, when the number of vertices of an allowed VPDU is limited to be equal to or less than 6, the shapes of the allowed VPDUs can be square, rectangle, L-shape, or reverse L-shape.

[0148] In another embodiment, when a block (e.g., CTU, a block corresponding to a non-leaf node) spans over multiple VPDUs, then the number of concatenating TT splits (at

the consecutive partition depth) is limited to constrain the partition of the block. The number of concatenating TT splits is a fixed number, such as 0, 1, 2, 3. For example, when the number of concatenating TT splits is limited to be zero, then no TT split is allowed. In another example, when the number of concatenating TT split is limited to be equal or less than 1, then a block resulted from a TT split cannot be split by another TT split. In an example, the number of concatenating TT splits is signaled in the coded video bitstream that is sent from the encoder side to the decoder side. In another example, the number of concatenating TT splits is predetermined, and known (e.g., stored in memory) at the encoder side and the decoder side.

[0149] In another embodiment, the widths (W) of VPDUs are limited. For example, only VPDUs with widths (W) larger than or equal to a width threshold are allowed. The width threshold is a fixed number, such as 64.

[0150] In another embodiment, the heights (H) of VPDUs are limited. For example, only VPDUs with height (H) larger than or equal to a height threshold are allowed. The height threshold is a fixed number, such as 64.

[0151] In another embodiment, when multiple CUs in a VPDU, the widths of the CUs in the VPDU are the same.

[0152] In another embodiment, when multiple CUs in a VPDU, the multiple CUs in the VPDU can have at most two different width values.

[0153] In some embodiments, the number of CUs in a VPDU is limited. In an embodiment, the number of CUs in a VPDU is limited to be no more than an upper limit value. In an example, the t upper limit value is a fixed number, such as 2 or 3. In another example, the upper limit value can be determined based on the width (W) of the VPDU. In another example, the upper limit value can be determined based on the height (H) of the VPDU. In another example, the upper limit value can be determined based on the area of the VPDU, such as the number of luma pixels in the VPDU.

[0154] According to some aspects of the disclosure, BT split and TT split at non-intra pictures are constrained.

[0155] In some embodiments, TT split at the CTU level is allowed.

[0156] FIG. 12 shows an example of a TT split at CTU level and corresponding VPDU partition. In the FIG. 12 example, a CTU (1210) is partitioned into three CUs (1211)-(1213) by a TT split (in solid lines). In an example, the number of samples of the VPDU is ¼ of the number of samples in CTU. Then, the CTU (shown as 1220) can be partitioned into four VPDU (1221)-(1224), as shown by dashed lines.

[0157] In some embodiments, when a block is contained in a specific rectangular area, TT split is allowed. Otherwise, TT split is prohibited. The specific rectangular area can be defined as MxN area. In an embodiment, M and N are fixed numbers, such as the width/height of a square of the VPDU size. In another embodiment, M and N depends on the CTU size. For example, M is equal to half of the CTU width, and N is equal to half of the CTU height, and a partition example is shown in FIG. 13.

[0158] FIG. 13 shows an example of CTU partition according to an embodiment of the disclosure. In the FIG. 13 example, when a block is in a square of VPDU size, TT split is allowed to split the block. For example, the CTU 1310 is split into four blocks (1311)-(1314) by QT split. The number of samples in the VPDU is ¼ of the number of samples in the CTU. Then, the block (1311) is in a square of the VPDU size, and TT split is allowed to further split the block (1311) into smaller blocks.

[0159] In some embodiments, when the length of one side of a block (i.e., width or height) is larger than N and the length of the other side of the block (i.e., height or width) is less than or equal to N, then BT is allowed, where N is the width of the square VPDU. For example, when the CTU size is 128 and N is equal to 64, for 128x64 and 64x128 blocks, BT is allowed in both horizontal and vertical directions.

[0160] In some embodiments, when a block is not at CTU level and is not contained in the specific rectangular area, TT is prohibited. In one embodiment, when one side of a block (i.e., width or height) is equal to the width/height of the CTU (for example, 128), and the other side of the block (i.e., height or width) is not equal to the width/height of the CTU (for example, 128), then TT split is not allowed for the block. For example, when the width/height of a CTU is 128, then TT split for a 128x64 or 64x128 block is not allowed.

[0161] FIG. 14 shows an example of CTU partition and corresponding VPDU partition. In the FIG. 14 example, a CTU (1410) is split into three blocks (1411)-(1413) by a TT split (in solid lines) in vertical direction. The block (1412) is split into three blocks (1414)-(1416) by a TT split in horizontal direction. The block (1415) is split into two blocks (1417) and (1418) by a BT in vertical direction. In an example, the number of samples of the VPDU is ¼ of the number of samples in CTU. Then, the CTU (as shown by 1420) can be partitioned into four VPDU (1421)-(1424), as shown by dashed lines. In some embodiments, the TT split of the block (1412) is not allowed to avoid the possibility of non-rectangular shape VPDU. In an example, the block (1412) is 64x128, and TT split of the block (1412) is not allowed. In another example, the block (1412) is a result of a TT split, and thus a concatenating TT split of the block (1412) is not allowed.

[0162] According to some aspects of the disclosure, at the picture boundary, in order to keep the VPDU processing consistent with the non-boundary areas, the padding area outside the picture boundary is considered as normal block when VPDU constraints are applied, regardless of the picture boundary.

[0163] FIG. 15 shows a flow chart outlining a process (1500) according to an embodiment of the disclosure. The process (1500) can be used in the reconstruction of a block coded in intra mode, so to generate a prediction block for the block under reconstruction. In various embodiments, the process (1500) are 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 video encoder (503), 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). The process starts at (S1501) and proceeds to (S1510).

[0164] At (S1510), partition information of a block in a picture is decoded from a coded video bitstream. In an example, the block is at CTU level. The partition information is indicative of a partition tree structure for a partition of the block into coding blocks. For example, according to the partition tree structure, a CTU is partitioned into CUs that are leaf nodes in the partition tree structure.

[0165] At (S1520), to-be-reconstructed samples of the block are grouped into processing groups according to the partition of the block. For example, the processing groups respectively correspond to the VPDUs. The shapes of regions of the processing groups are limited by constraints in the partition of the block.

[0166] At (S1530), the to-be-reconstructed samples are reconstructed by multiple pipeline stages that process the processing groups in a pipeline processing manner. For example, each processing group is processed successively by the multiple pipeline stages. A pipeline stage provides processing results of a processing group to a next pipeline stage, and then processes a next processing group. The multiple pipeline stages process different processing groups at the same time. Then, the process proceeds to (S1599) and terminates.

[0167] 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.

[0168] 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.

[0169] 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.

[0170] 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).

[0171] 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).

[0172] 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).

[0173] 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).

[0174] 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.

[0175] 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.

[0176] Computer system (1600) can also include an interface to one or more communication networks. Networks can for example be wireless, wireline, optical. Networks can further be local, wide-area, metropolitan, vehicular and industrial, real-time, delay-tolerant, and so on. Examples of networks 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.

[0177] Aforementioned human interface devices, human-accessible storage devices, and network interfaces can be attached to a core (1640) of the computer system (1600).

[0178] 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 a 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.

[0179] 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.

[0180] 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.

[0181] 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.

Appendix A: Acronyms

JEM: joint exploration model

VVC: versatile video coding

BMS: benchmark set

MV: Motion Vector

HEVC: High Efficiency Video Coding

SEI: Supplementary Enhancement Information

VUI: Video Usability Information

GOPs: Groups of Pictures

TUs: Transform Units,

PUs: Prediction Units

CTUs: Coding Tree Units

CTBs: Coding Tree Blocks

PBs: Prediction Blocks

HRD: Hypothetical Reference Decoder

SNR: Signal Noise Ratio

CPUs: Central Processing Units

GPUs: Graphics Processing Units

CRT: Cathode Ray Tube

LCD: Liquid-Crystal Display

OLED: Organic Light-Emitting Diode

CD: Compact Disc

DVD: Digital Video Disc

ROM: Read-Only Memory

RAM: Random Access Memory

ASIC: Application-Specific Integrated Circuit

PLD: Programmable Logic Device

LAN: Local Area Network

GSM: Global System for Mobile communications

LTE: Long-Term Evolution

CANBus: Controller Area Network Bus

USB: Universal Serial Bus

PCI: Peripheral Component Interconnect

FPGA: Field Programmable Gate Areas

SSD: solid-state drive

IC: Integrated Circuit

CU: Coding Unit

[0182] 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.

WHAT IS CLAIMED IS:

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

decoding partition information of a block in a picture from a coded video bitstream, the partition information being indicative of a partition tree structure for a partition of the block into coding blocks;

grouping to-be reconstructed samples of the block into processing groups according to the partition of the block into the coding blocks, a shape of a combined region of a processing group being limited by constraints in the partition of the block into the coding blocks; and

reconstructing the to-be reconstructed samples of the block by multiple pipeline stages that process the processing groups in a pipeline processing manner.

2. The method of claim 1, wherein the shape of the combined region of the processing group is limited to be one of a square shape and a rectangular shape.

3. The method of claim 1, wherein the shape of the combined region of the processing group has no more than a fixed number of edges or has no more than a fixed number of vertices.

4. The method of claim 1, wherein the shape of the combined region of the processing group is limited by a constraint of no more than a fixed number of concatenating ternary tree (TT) splits in the partition tree structure for the partition of the block into the coding blocks.

5. The method of claim 1, wherein the combined region of the processing group has at least one of a width or a height equal to or larger than a fixed number.

6. The method of claim 1, wherein the processing group includes samples from multiple coding blocks, and a number of the multiple coding blocks is constrained to be no more than a limit value.

7. The method of claim 6, wherein the limit value is determined based on a size of the combined region.

8. The method of claim 1, wherein the shape of the combined region of the processing group is limited by constraints that limit usages of ternary tree split and binary tree split in the partition tree structure when the current picture is a non-intra picture.

9. The method of claim 8, wherein a usage of ternary tree split is allowed on a to-be-split block corresponding to a non-leaf node in the partition tree structure when the to-be split block is at coding tree unit (CTU) level or the to-be-split block is contained in a specific area of a coding tree unit.

10. The method of claim 1, wherein the combined region includes a portion that is outside a boundary of the picture.

11. An apparatus for video decoding, comprising:

processing circuitry configured to:

decode partition information of a block in a picture from a coded video bitstream, the partition information being indicative of a partition tree structure for a partition of the block into coding blocks;

group to-be reconstructed samples of the block into processing groups according to the partition of the block into the coding blocks, a shape of a combined region of a processing group being limited by constraints in the partition of the block into the coding blocks; and

reconstruct the to-be reconstructed samples of the block by multiple pipeline stages that process the processing groups in a pipeline processing manner.

12. The apparatus of claim 11, wherein the shape of the combined region of the processing group is limited to be one of a square shape and a rectangular shape.

13. The apparatus of claim 11, wherein the shape of the combined region of the processing group has no more than a fixed number of edges or has no more than a fixed number of vertices.

14. The apparatus of claim 11, wherein the shape of the combined region of the processing group is limited by a constraint of no more than a fixed number of concatenating ternary tree (TT) splits in the partition tree structure for the partition of the block into the coding blocks.

15. The apparatus of claim 11, wherein the combined region of the processing group has at least one of a width or a height equal to or larger than a fixed number.

16. The apparatus of claim 11, wherein the processing group includes samples from multiple coding blocks, and a number of the multiple coding blocks is constrained to be no more than a limit value.

17. The apparatus of claim 16, wherein the limit value is determined based on a size of the combined region.

18. The apparatus of claim 11, wherein the shape of the combined region of the processing group is limited by constraints that limit usages of ternary tree split and binary tree split in the partition tree structure when the current picture is a non-intra picture.

19. The apparatus of claim 18, wherein a usage of ternary tree split is allowed on a to-be-split block corresponding to a non-leaf node in the partition tree structure when the to-

be split block is at coding tree unit (CTU) level or the to-be-split block is contained in a specific area of a coding tree unit.

20. A non-transitory computer-readable medium storing instructions which when executed by a computer for video decoding cause the computer to perform:

decoding partition information of a block in a picture from a coded video bitstream, the partition information being indicative of a partition tree structure for a partition of the block into coding blocks;

grouping to-be reconstructed samples of the block into processing groups according to the partition of the block into the coding blocks, a shape of a combined region of a processing group being limited by constraints in the partition of the block into the coding blocks; and

reconstructing the to-be reconstructed samples of the block by multiple pipeline stages that process the processing groups in a pipeline processing manner.

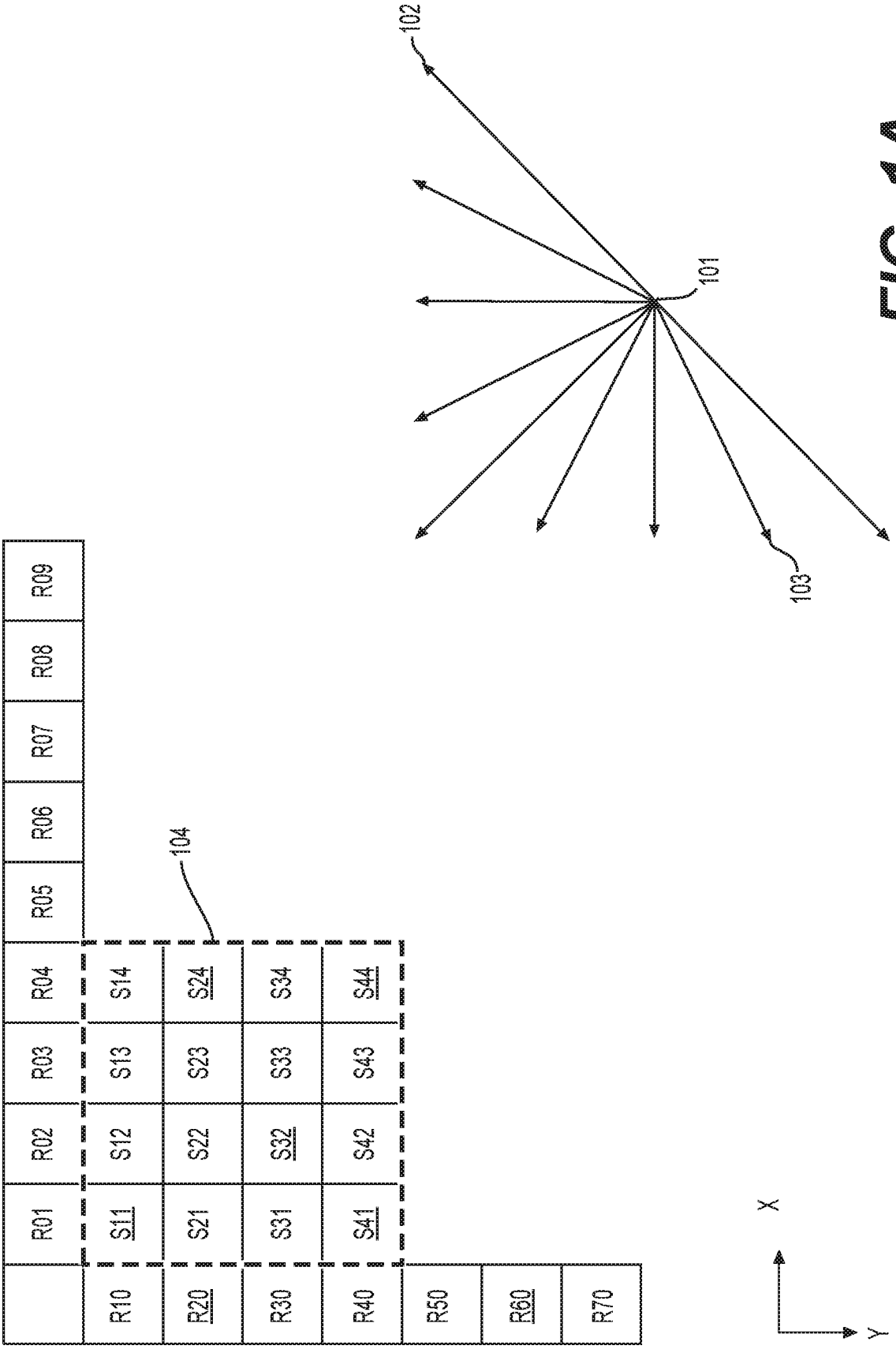

R01
R02
R03
R04
R05
R06
R07
R08
R09
R10
S11
S12
S13
S14
R20
S21
S22
S23
S24
R30
S31
S32
S33
S34
R40
S41
S42
S43
S44
R50
R60
R70
104
102
101
103
X
Y

FIG. 1A

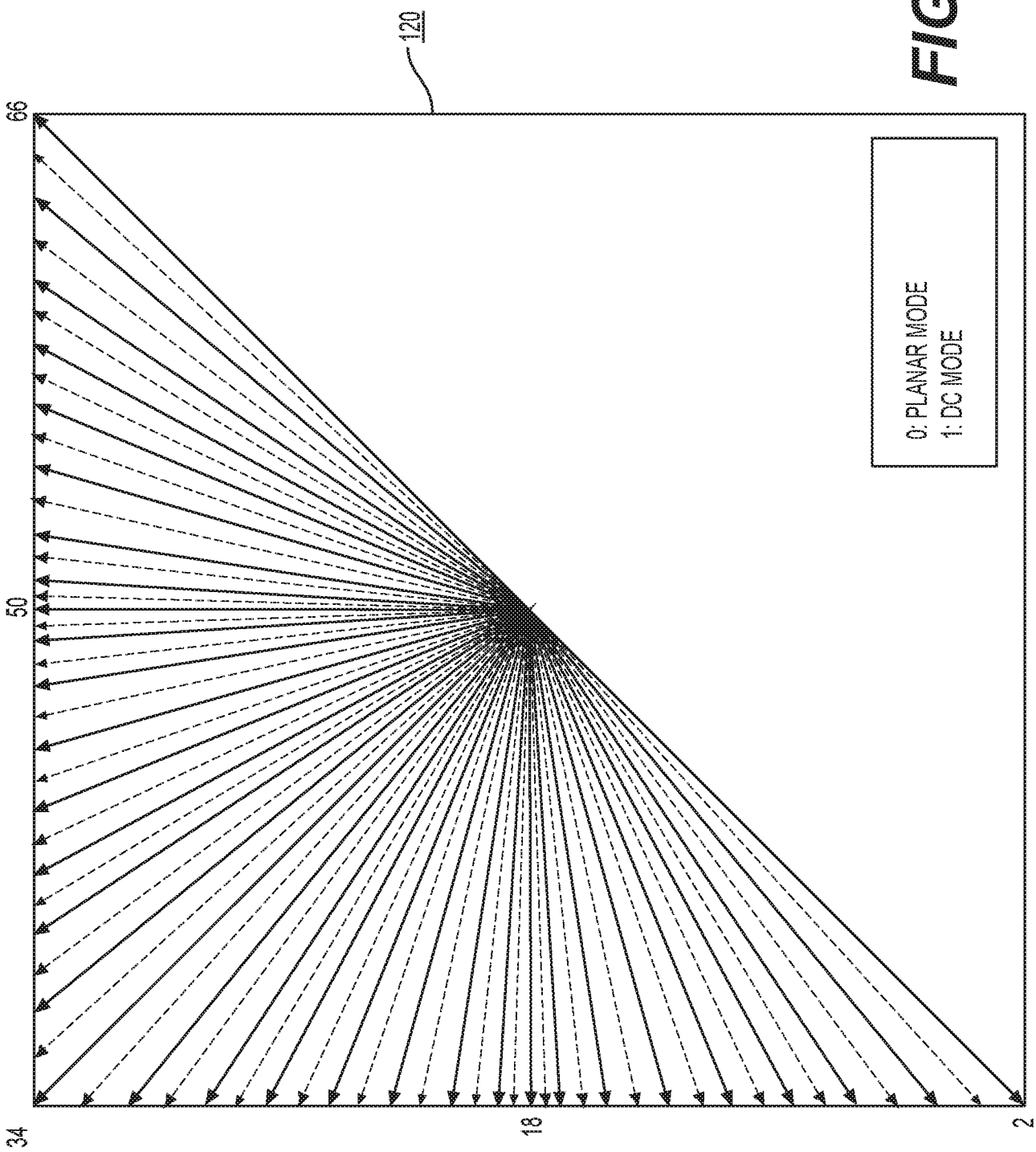


FIG. 1B

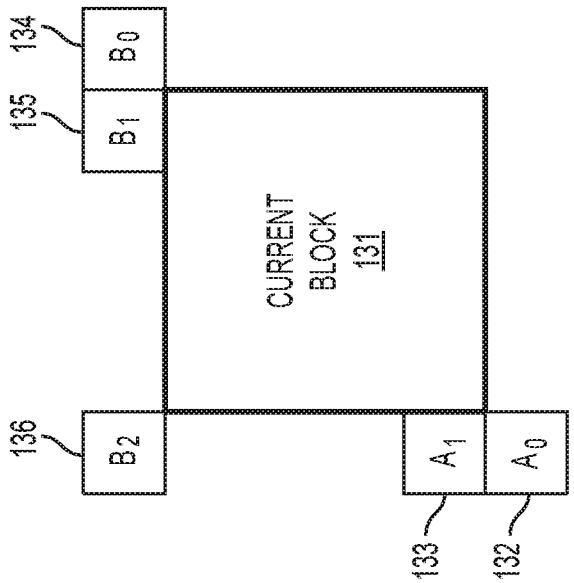


FIG. 1C

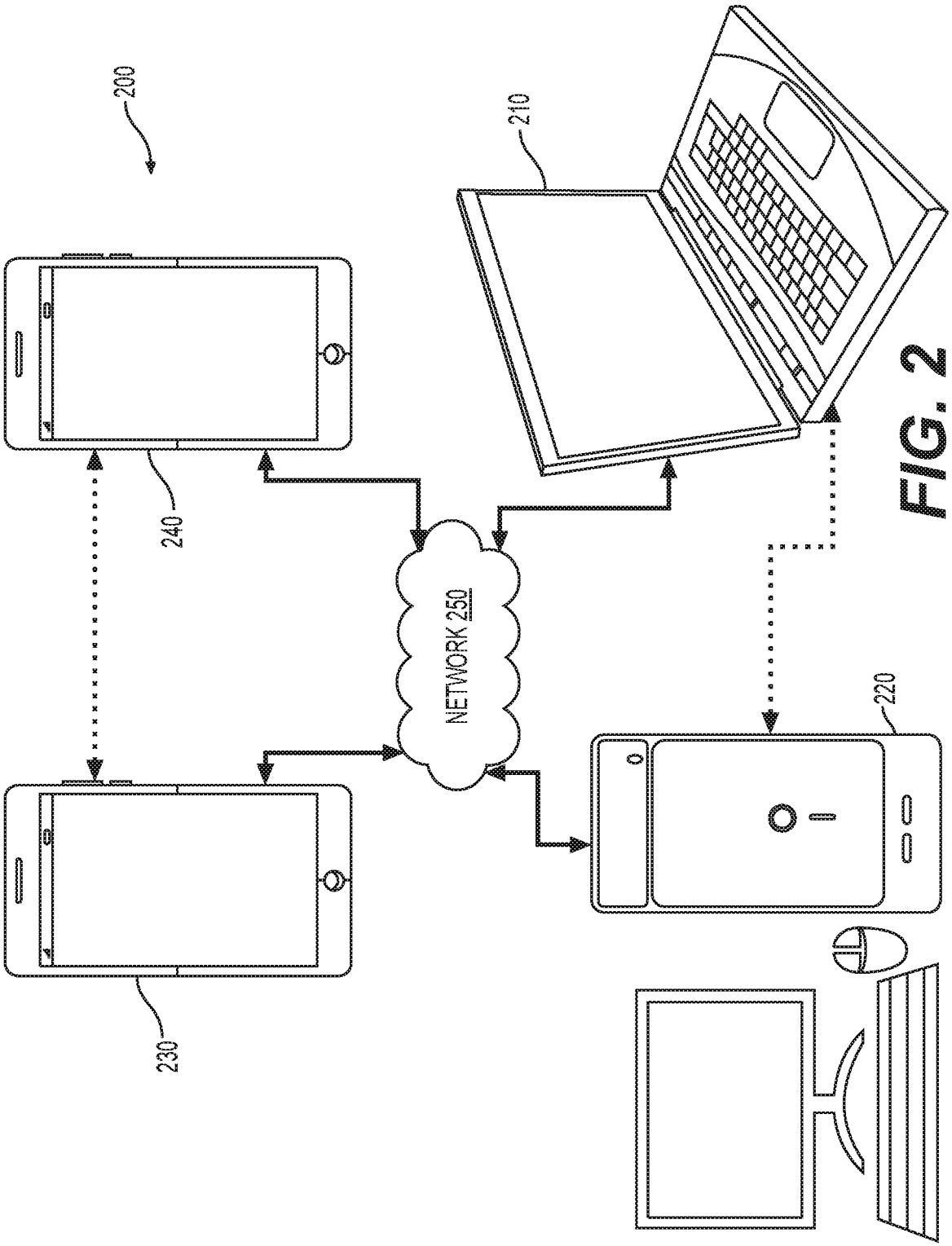


FIG. 2

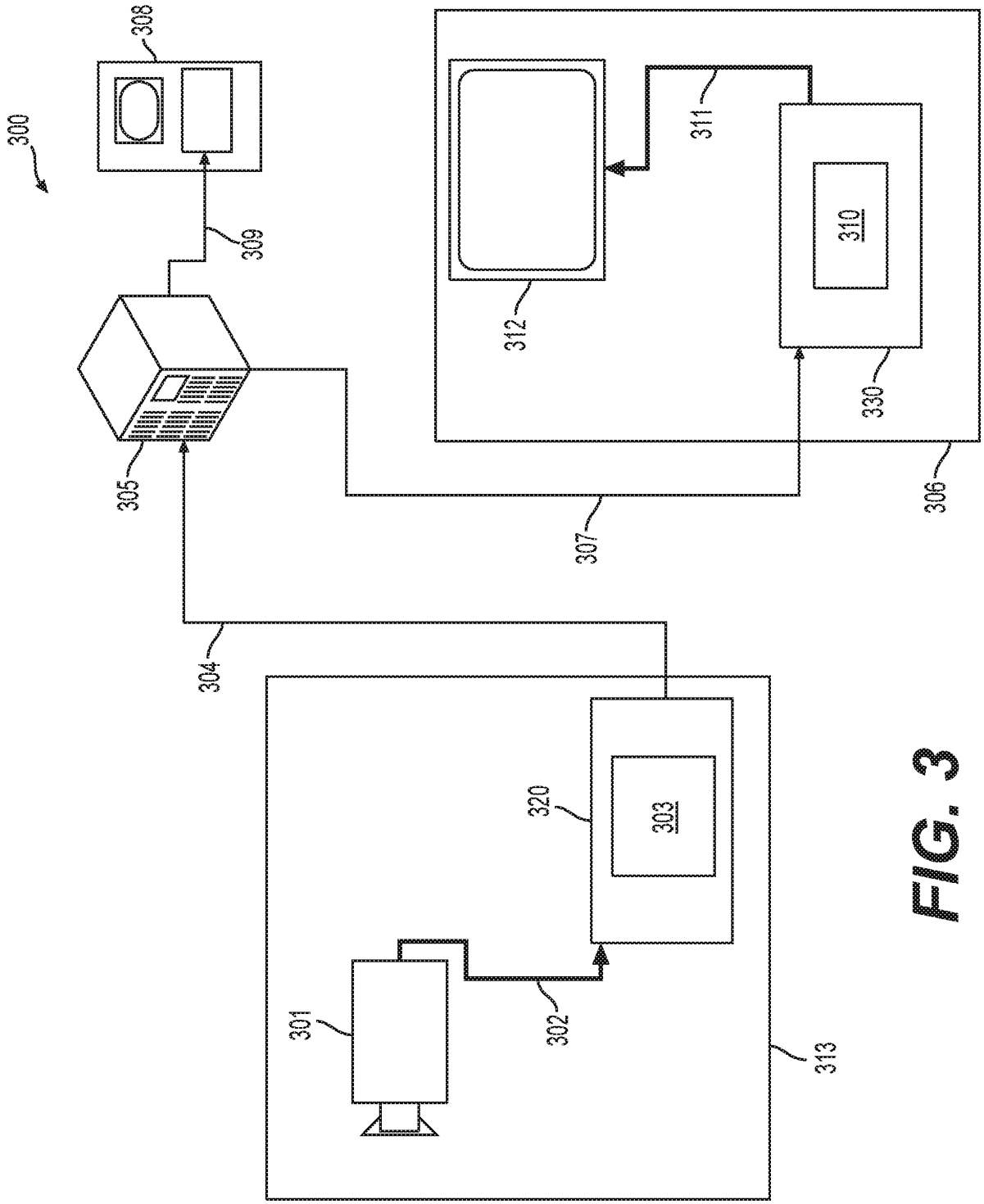


FIG. 3

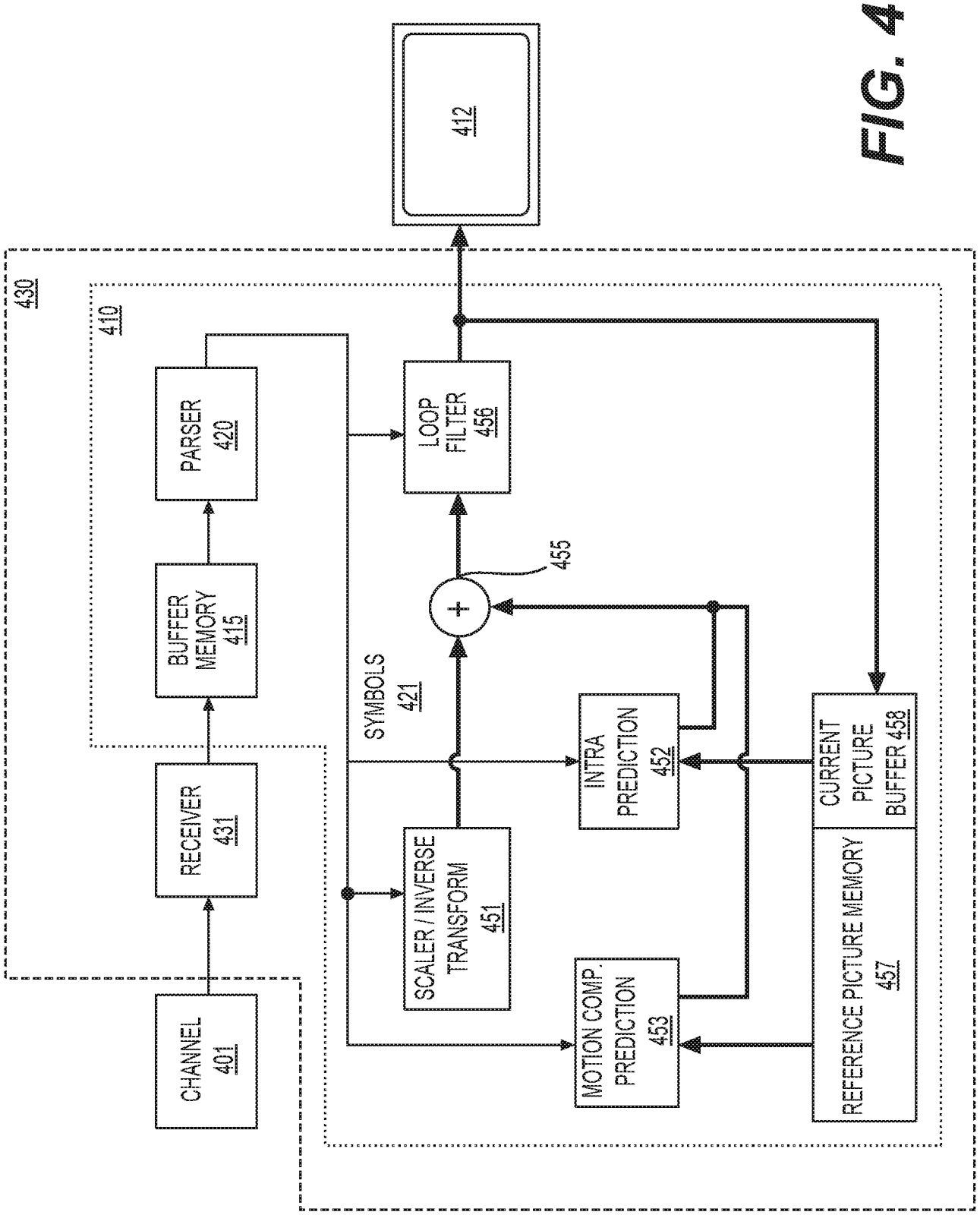

412
430
410
PARSER 420
BUFFER MEMORY 415
RECEIVER 431
CHANNEL 401
LOOP FILTER 456
455
+
SYMBOLS 421
SCALER / INVERSE TRANSFORM 451
INTRA PREDICTION 452
MOTION COMP. PREDICTION 453
CURRENT PICTURE BUFFER 458
REFERENCE PICTURE MEMORY 457

FIG. 4

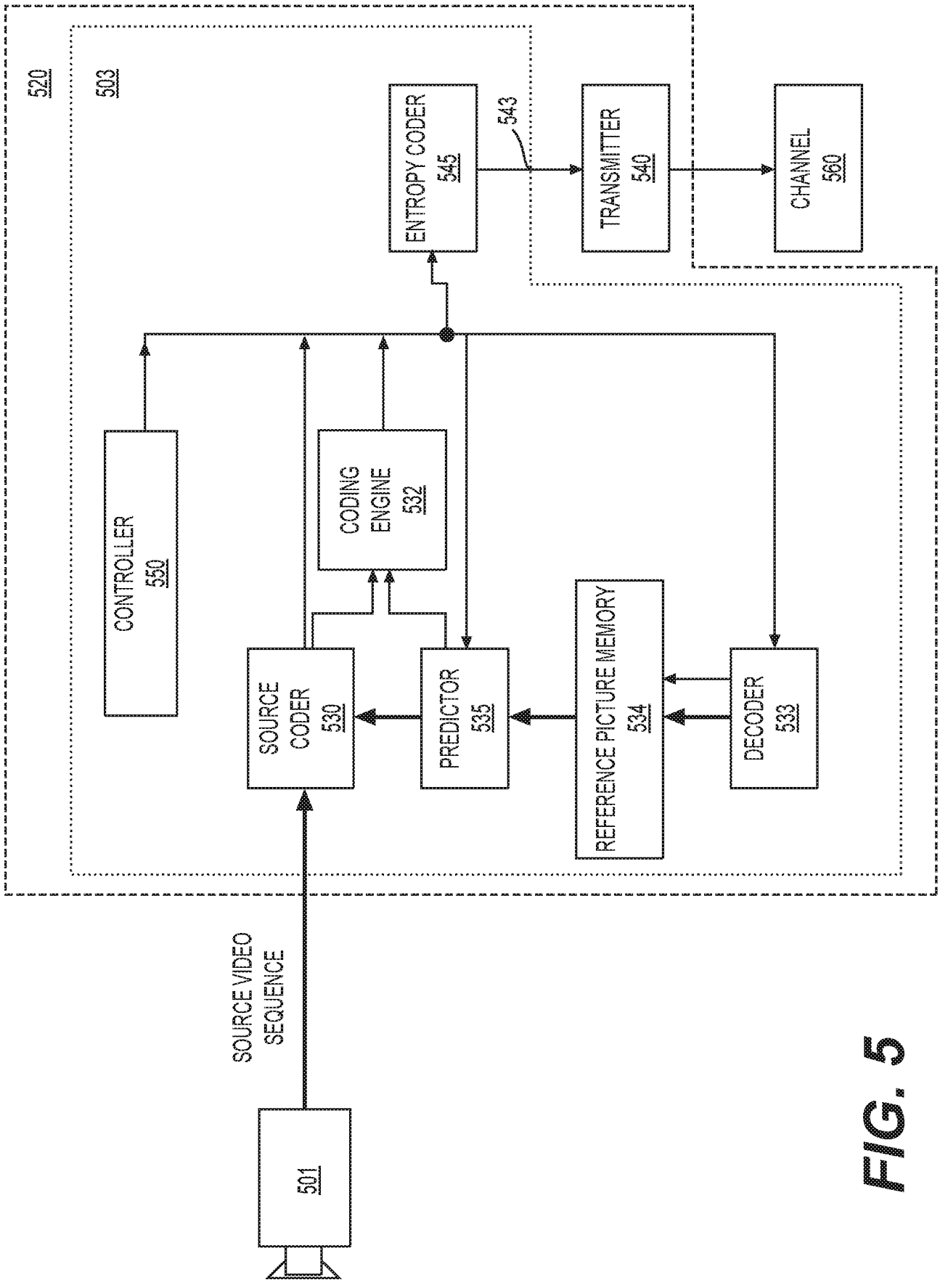

520
503
ENTROPY CODER
545
543
TRANSMITTER
540
CHANNEL
560
CONTROLLER
550
CODING
ENGINE
532
SOURCE
CODER
530
PREDICTOR
535
REFERENCE PICTURE MEMORY
534
DECODER
533
SOURCE VIDEO
SEQUENCE
501

FIG. 5

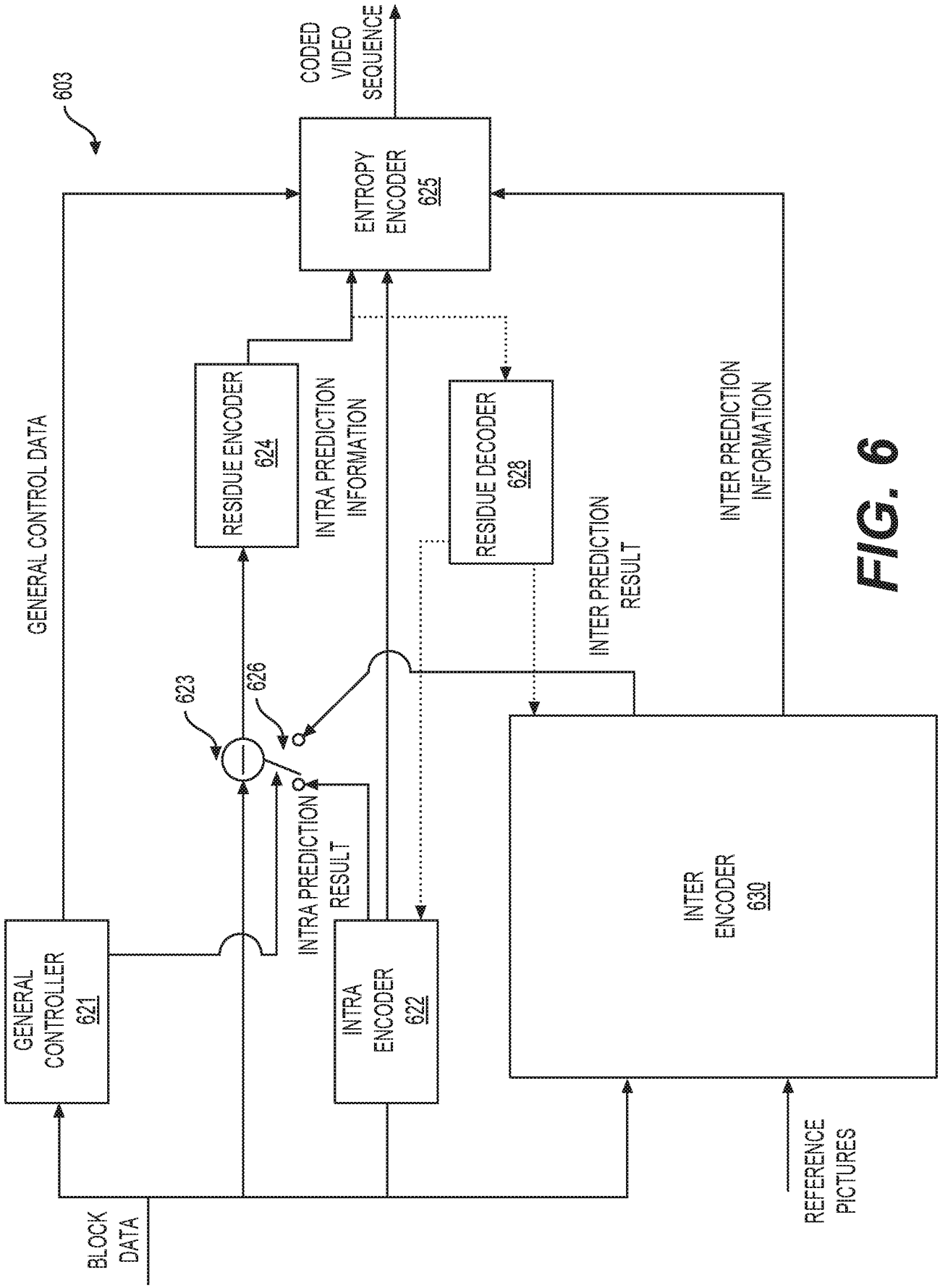

603
CODED VIDEO SEQUENCE
ENTROPY ENCODER 625
GENERAL CONTROL DATA
RESIDUE ENCODER 624
INTRA PREDICTION INFORMATION
RESIDUE DECODER 628
INTER PREDICTION INFORMATION
INTER PREDICTION RESULT
623
626
INTRA PREDICTION RESULT
GENERAL CONTROLLER 621
INTRA ENCODER 622
INTER ENCODER 630
BLOCK DATA
REFERENCE PICTURES

FIG. 6

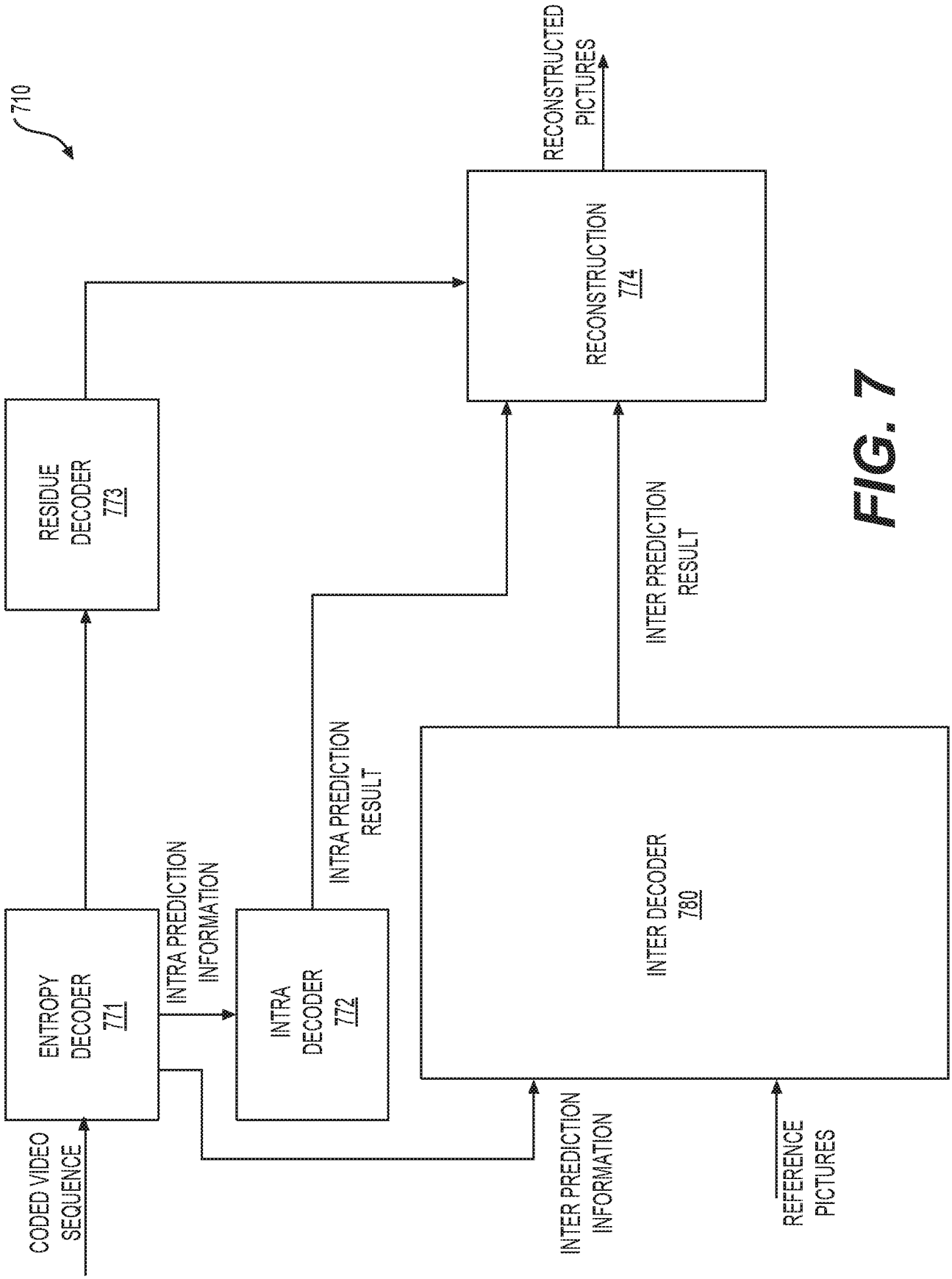

710
RECONSTRUCTED PICTURES
RECONSTRUCTION 774
RESIDUE DECODER 773
INTER PREDICTION RESULT
INTRA PREDICTION RESULT
INTER DECODER 780
ENTROPY DECODER 771
INTRA PREDICTION INFORMATION
INTRA DECODER 772
CODED VIDEO SEQUENCE
INTER PREDICTION INFORMATION
REFERENCE PICTURES

FIG. 7

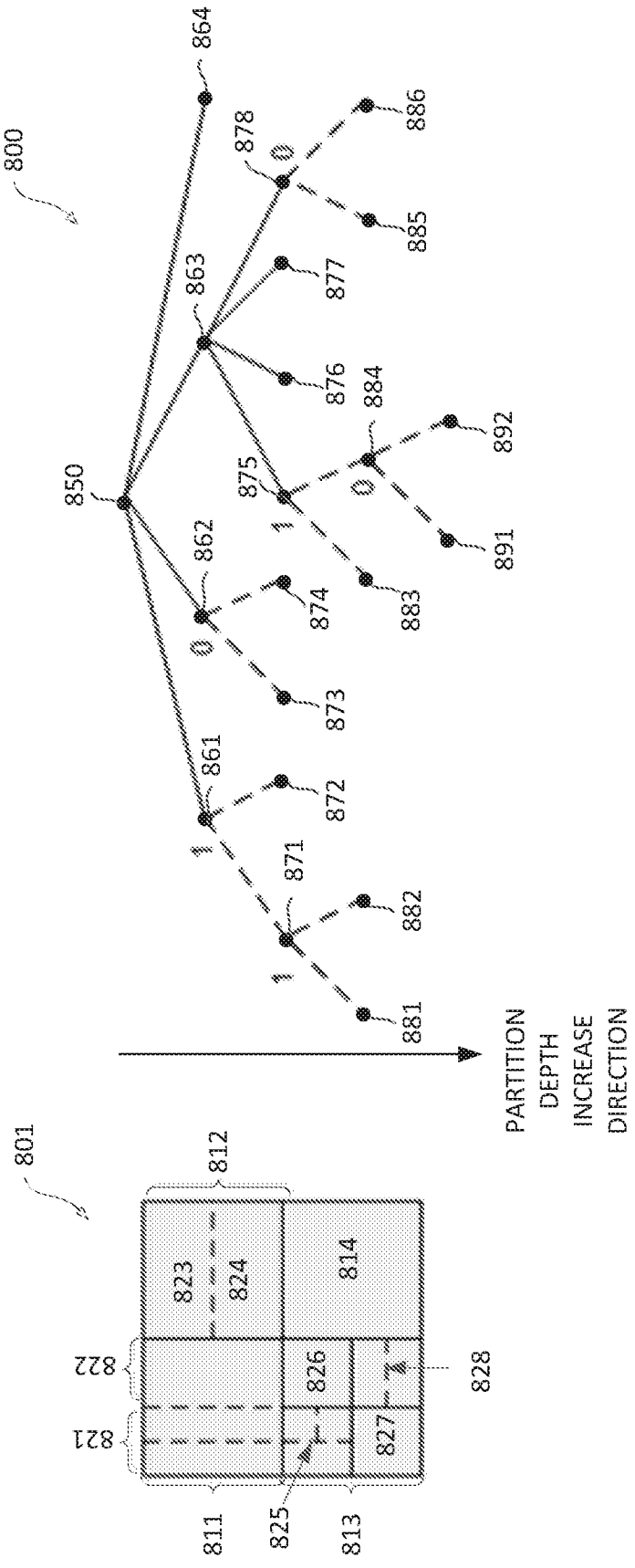

800
850
861
862
863
864
871
872
873
874
875
876
877
878
881
882
883
884
885
886
891
892
1
0
PARTITION DEPTH INCREASE DIRECTION
801
811
812
813
814
821
822
823
824
825
826
827
828

FIG. 8

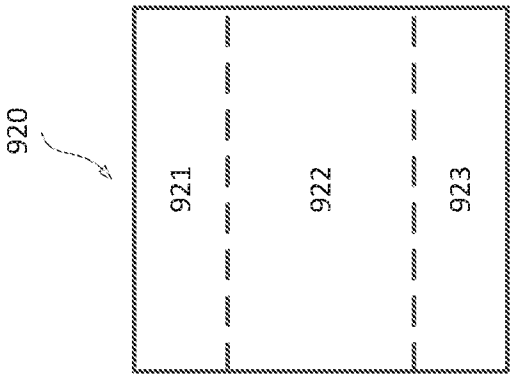


FIG. 9B

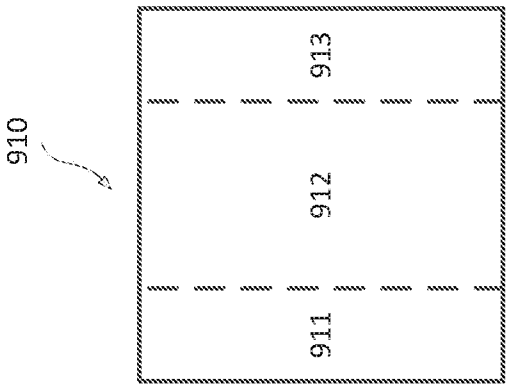


FIG. 9A

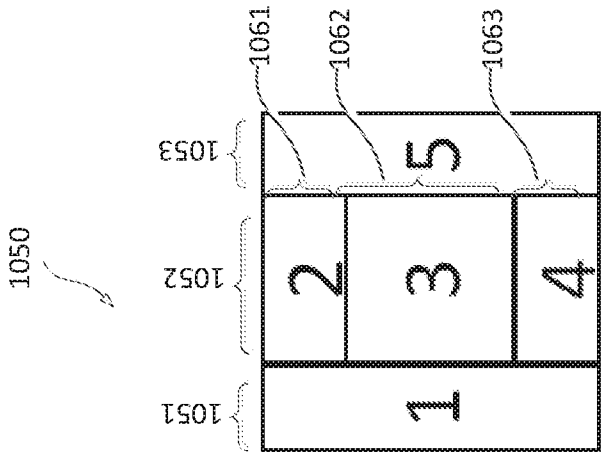


FIG. 10B

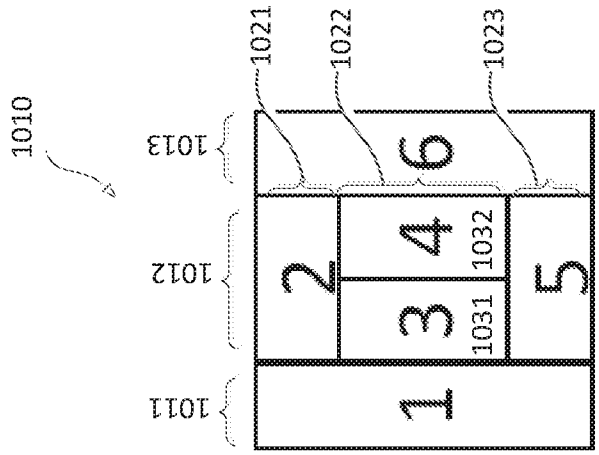


FIG. 10A

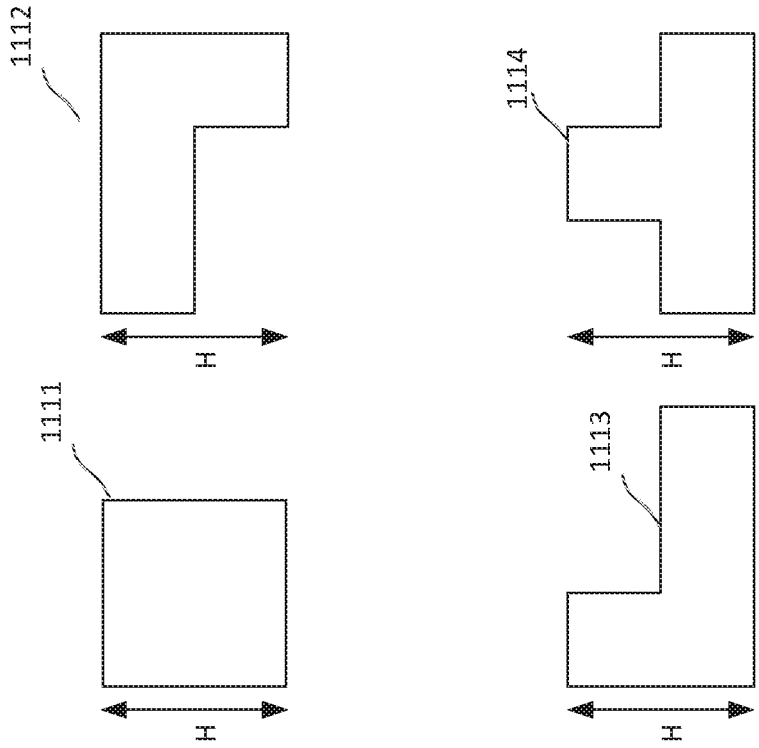


FIG. 11B

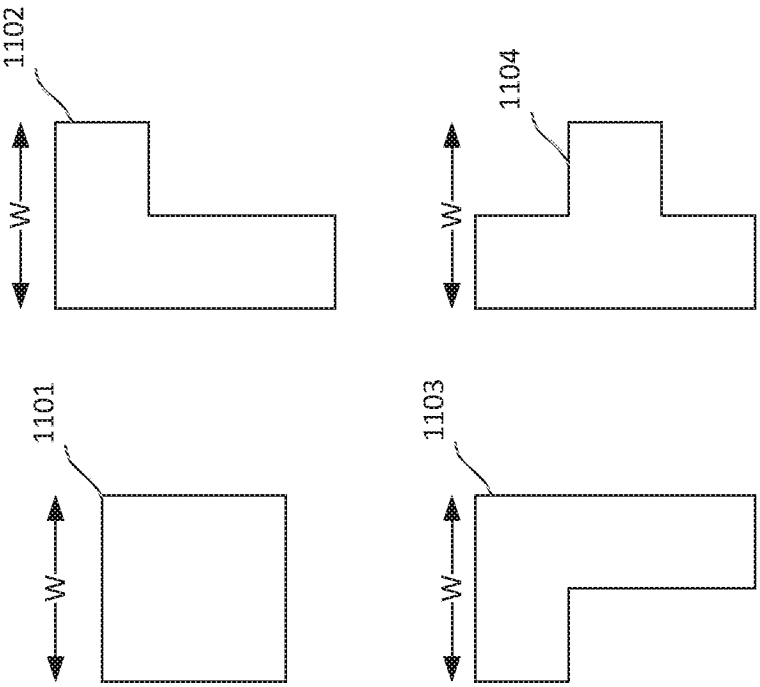


FIG. 11A

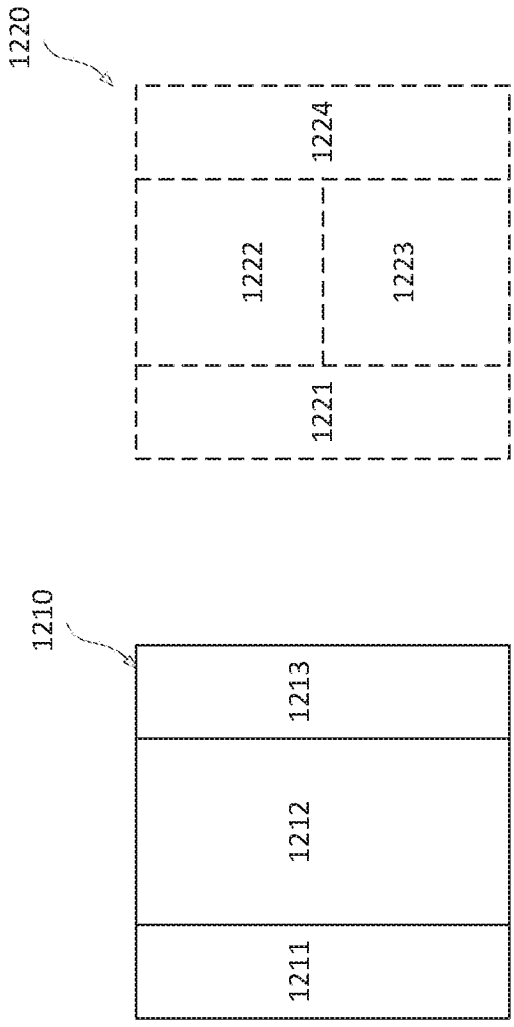


FIG. 12

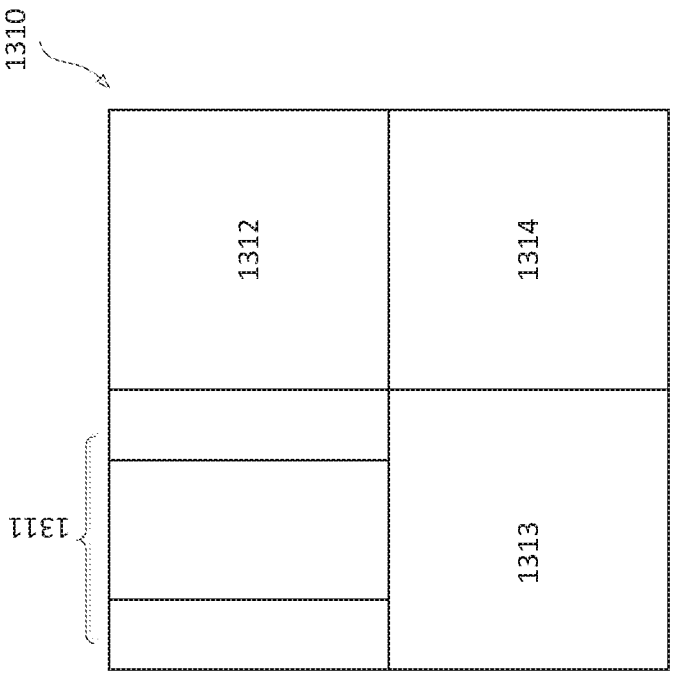


FIG. 13

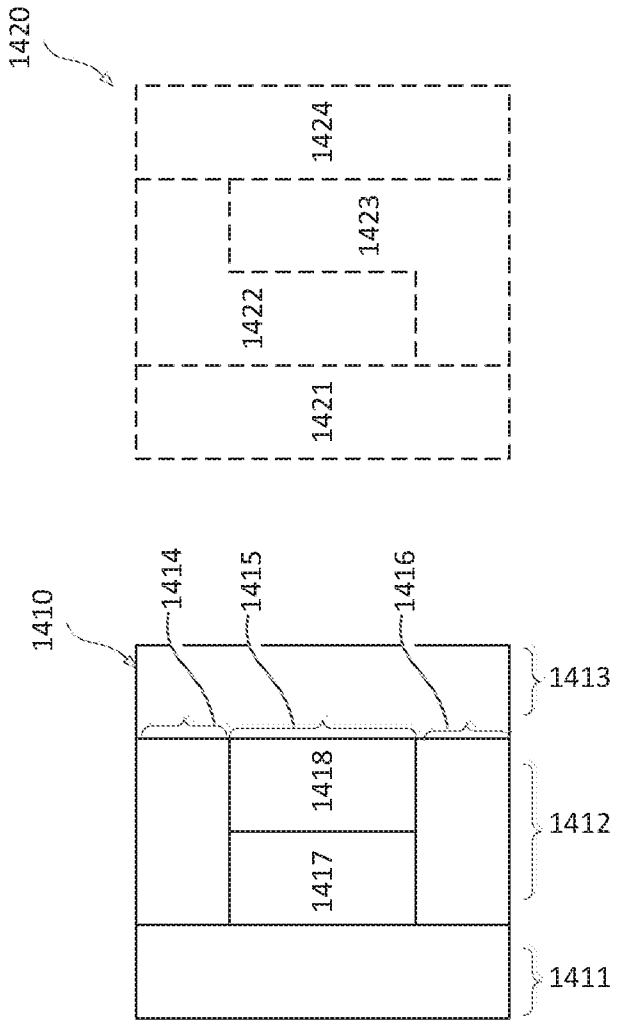

1420
1424
1423
1422
1421
1410
1414
1415
1416
1413
1418
1412
1417
1411

FIG. 14

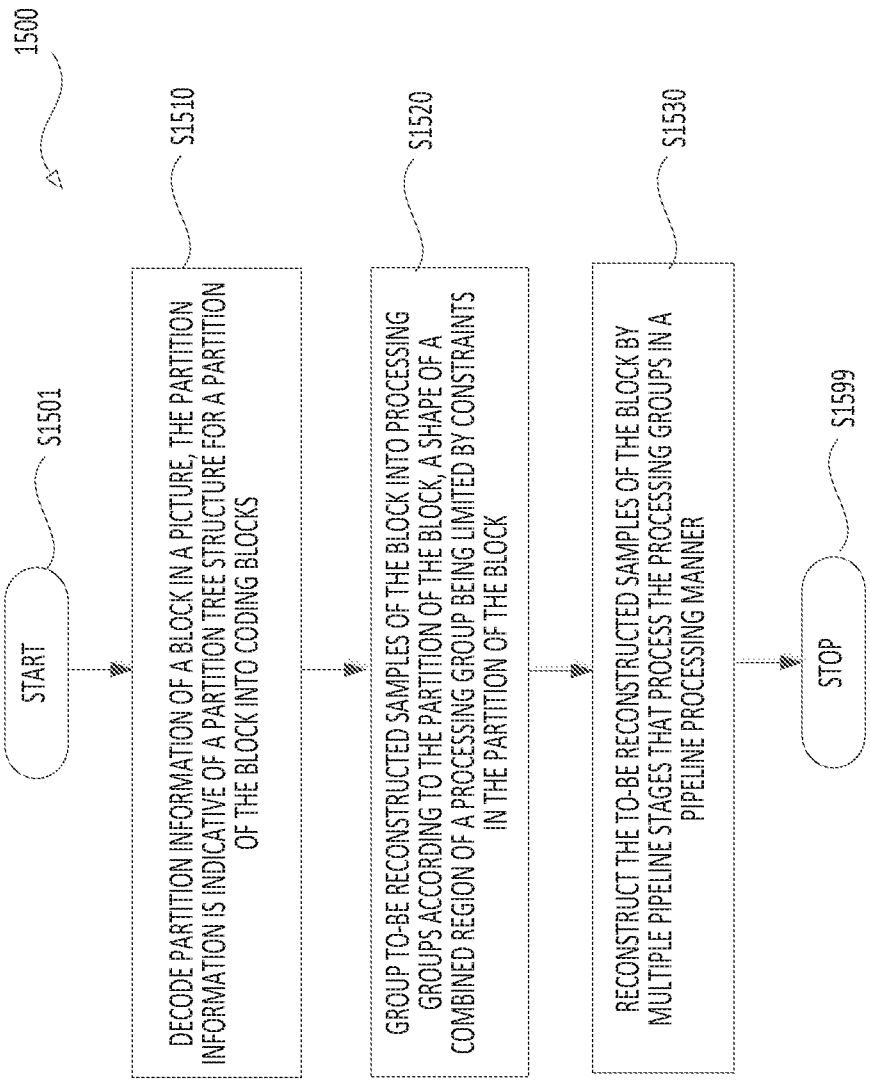

1500
START
S1501
DECODE PARTITION INFORMATION OF A BLOCK IN A PICTURE, THE PARTITION INFORMATION IS INDICATIVE OF A PARTITION TREE STRUCTURE FOR A PARTITION OF THE BLOCK INTO CODING BLOCKS
S1510
GROUP TO-BE RECONSTRUCTED SAMPLES OF THE BLOCK INTO PROCESSING GROUPS ACCORDING TO THE PARTITION OF THE BLOCK, A SHAPE OF A COMBINED REGION OF A PROCESSING GROUP BEING LIMITED BY CONSTRAINTS IN THE PARTITION OF THE BLOCK
S1520
RECONSTRUCT THE TO-BE RECONSTRUCTED SAMPLES OF THE BLOCK BY MULTIPLE PIPELINE STAGES THAT PROCESS THE PROCESSING GROUPS IN A PIPELINE PROCESSING MANNER
S1530
STOP
S1599

FIG. 15

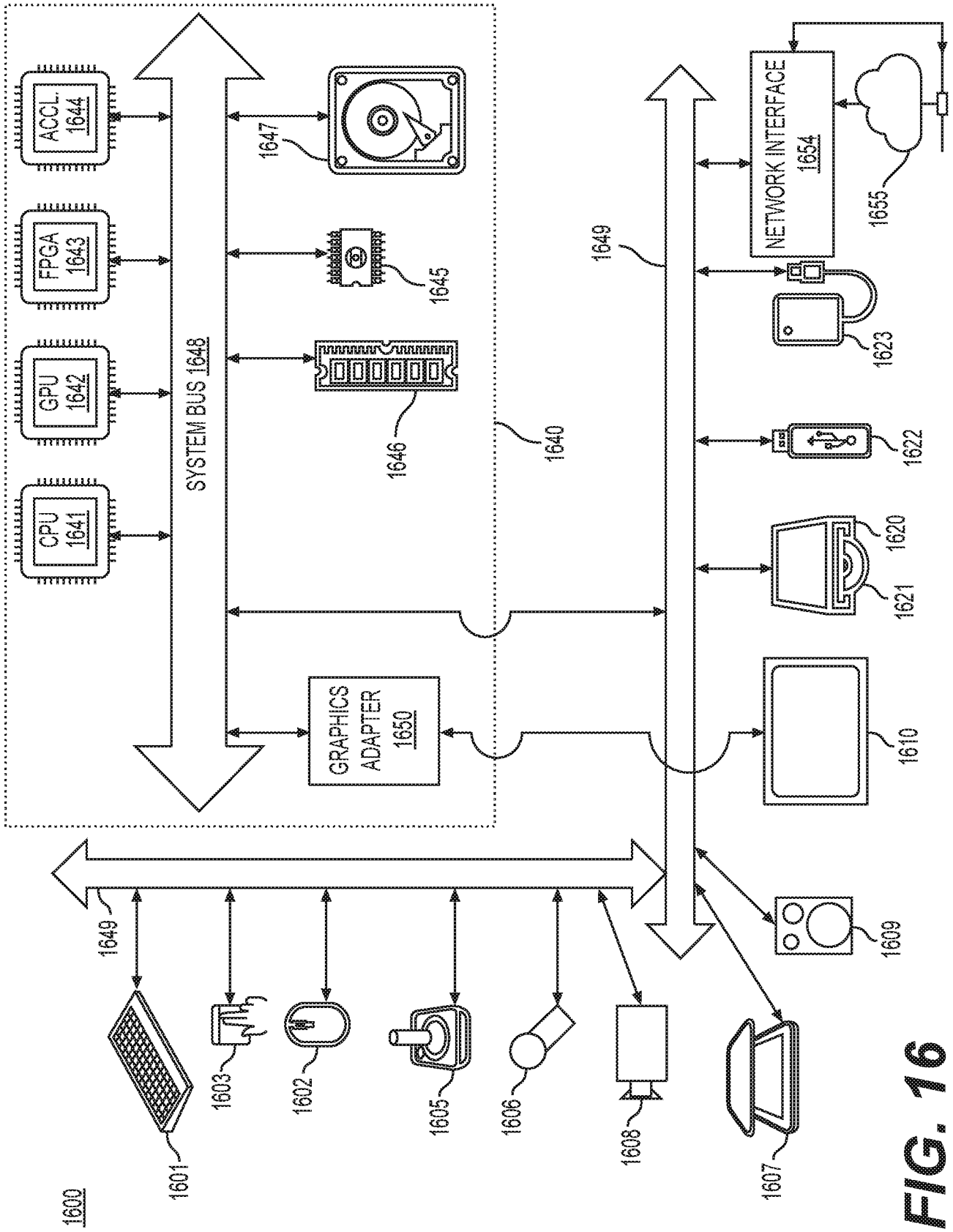

1600
1601
1603
1602
1605
1606
1608
1607
1649
1609
1610
1620
1621
1622
1623
1655
NETWORK INTERFACE 1654
1649
1647
1645
1646
1640
SYSTEM BUS 1648
ACCL. 1644
FPGA 1643
GPU 1642
CPU 1641
GRAPHICS ADAPTER 1650

FIG. 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2019/053221

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - H04N 19/96; H04N 19/70; H04N 19/176; H04N 19/44 (2019.01)
CPC - H04N 19/96; H04N 19/70; H04N 19/176; H04N 19/44 (2019.08)

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
USPC - 375/240.24; 341/79; 375/240.25 (keyword delimited)

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.
A	US 2013/0034157 A1 (FRAUNHOFER-GESELLSCHAFT ZUR FOERDERUNG DER ANGEW) 07 February 2013 (07.02.2013) entire document	1-20
A	US 2015/0016543 A1 (QUALCOMM INCORPORATED) 15 January 2015 (15.01.2015) entire document	1-20
A	US 2018/0063528 A1 (MEDIATEK SINGAPORE PTE. LTD.) 01 March 2018 (01.03.2018) entire document	1-20
A	US 2014/0003525 A1 (CISCO TECHNOLOGY, INC.) 02 January 2014 (02.01.2014) entire document	1-20
A	US 2017/0230659 A1 (JVC KENWOOD CORPORATION) 10 August 2017 (10.08.2017) entire document	1-20
A	US 2015/0139337 A1 (QUALCOMM INCORPORATED) 21 May 2015 (21.05.2015) entire document	1-20
A	US 2012/0163470 A1 (WU et al) 28 June 2012 (28.06.2012) entire document	1-20

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

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

Date of the actual completion of the international search	Date of mailing of the international search report
19 November 2019	06 DEC 2019
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774