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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 July 2021 (01.07.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/133456 A1

(51) International Patent Classification:
H04N 19/70 (2014.01) *H04N 19/105* (2014.01)
H04N 19/176 (2014.01) *H04N 19/126* (2014.01)
H04N 19/159 (2014.01)

(21) International Application Number:
PCT/US2020/054524

(22) International Filing Date:
07 October 2020 (07.10.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/953,457	24 December 2019 (24.12.2019)	US
62/955,825	31 December 2019 (31.12.2019)	US
17/063,149	05 October 2020 (05.10.2020)	US

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

(72) Inventors: **CHEN, Lien-Fei**; 2747 Park Boulevard, Palo Alto, California 94306 (US). **LI, Xiang**; c/o TENCENT AMERICA LLC, 2747 Park Boulevard, Palo Alto, California 94306 (US). **LI, Guichun**; 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, IT, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

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

(54) Title: METHOD AND APPARATUS FOR VIDEO CODING

(57) Abstract: Aspects of the disclosure provide methods, apparatuses, and non-transitory computer-readable storage mediums for video encoding/decoding. In a method, prediction information of a current block of a current picture in a coded bitstream is decoded. The prediction information indicates a geometric partitioning mode (GPM) for the current block. The current block is partitioned into two partitions in the GPM mode. Each of the partitions is associated with a respective predictor. A weighting index for a sample of the current block is determined based on a position of the sample. A weighting factor is calculated based on the weighting index of the sample according to an equation that converts the weighting index to the weighting factor. The sample is reconstructed based on the weighting factor and the predictor corresponding to the sample.

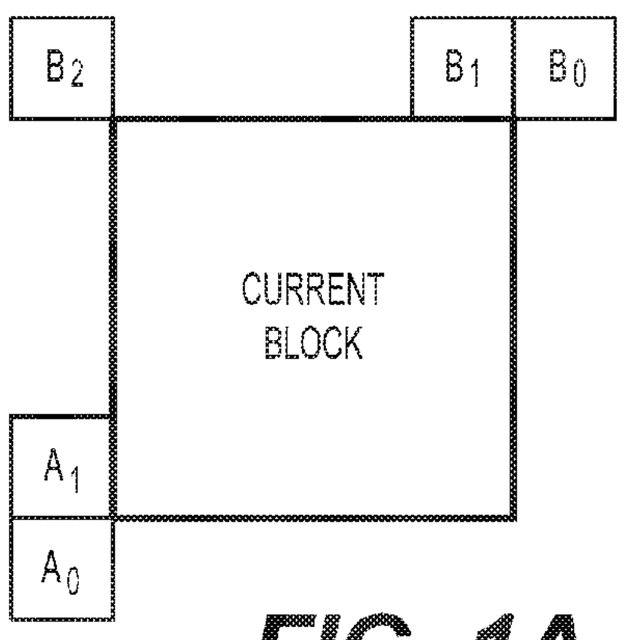


FIG. 1A

WO 2021/133456 A1

METHOD AND APPARATUS FOR VIDEO CODING

INCORPORATION BY REFERENCE

[0001] This present application claims the benefit of priority to U.S. Patenet Application No. 17/063,149, "METHOD AND APPARATUS FOR VIDEO CODING" filed on October 5, 2020, which claims the benefit of priority to U.S. Provisional Application No. 62/953,457, "SIMPLIFICATION FOR GEO INTER BLOCK" filed on December 24, 2019, and U.S. Provisional Application No. 62/955,825, "LOOK-UP TABLE FREE METHOD IN WEIGHTING INDEX TO WEIGHT CONVERSION FOR GEO INTER BLOCK" filed on December 31, 2019. The entire disclosures of the prior applications are hereby incorporated by reference in their entirety.

TECHNICAL FIELD

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

BACKGROUND

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

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

[0005] One purpose of video coding and decoding can be the reduction of redundancy in the input video signal, through compression. Compression can help reduce the aforementioned bandwidth or storage space requirements, in some cases by two orders of magnitude or more. Both lossless and lossy compression, as well as a combination thereof can be employed.

Lossless compression refers to techniques where an exact copy of the original signal can be reconstructed from the compressed original signal. When using lossy compression, the reconstructed signal may not be identical to the original signal, but the distortion between original and reconstructed signals is small enough to make the reconstructed signal useful for the intended application. In the case of video, lossy compression is widely employed. The amount of distortion tolerated depends on the application; for example, users of certain consumer streaming applications may tolerate higher distortion than users of television distribution applications. The compression ratio achievable can reflect that: higher allowable/tolerable distortion can yield higher compression ratios.

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

[0007] Video codec technologies can include techniques known as inter prediction. For each inter-predicted coding unit (CU), motion parameters include motion vectors, reference picture indices and reference picture list usage index, and additional information to be used for inter-predicted sample generation. The motion parameters can be signaled in an explicit or implicit manner. When a CU is coded with skip mode, the CU is associated with one prediction unit (PU) and has no significant residual coefficients, no coded motion vector delta or reference picture index. A merge mode is specified whereby the motion parameters for a current CU are obtained from neighboring CUs, including spatial and temporal candidates, and additional schedules introduced, for example in VVC. The merge mode can be applied to any inter-predicted CU, not only for skip mode. The alternative to merge mode is the explicit transmission of the motion parameters. Motion vector, corresponding reference picture index for each reference picture list and reference picture list usage flag and other needed information are signaled explicitly per each CU.

[0008] Some inter prediction coding tools include extended merge prediction, merge mode with motion vector difference (MMVD), advanced motion vector prediction mode (AMVP) with symmetric motion vector difference (MVD) signaling, affine motion compensated prediction, subblock-based temporal motion vector prediction (SbTMVP), adaptive motion vector resolution (AMVR), motion field storage (1/16th luma sample MV storage and 8x8 motion field compression), bi-prediction with weighted averaging (BWA), bi-directional optical

flow (BDOF), decoder side motion vector refinement (DMVR), triangular partitioning mode (TPM), and combined inter and intra prediction (CIIP).

[0009] In some cases, a merge candidate list is constructed by including the following five types of candidates in order: (1) spatial MVP from spatial neighbor CUs; (2) temporal MVP from collocated CUs; (3) history-based MVP from an FIFO table; (4) pairwise average MVP; and (5) zero MVs.

[0010] The size of the merge list is signaled in a slice header and the maximum allowed size of the merge list is for example 6 in some cases. For each CU code in merge mode, an index of the best merge candidate is encoded using truncated unary binarization (TU). The first bin of the merge index is coded with context and bypass coding is used for other bins.

[0011] FIG. 1A shows exemplary positions of spatial merge candidates. In some cases, up to four merge candidates can be selected among candidates located in the positions depicted in FIG. 1A. The selection order is B1, A1, B0, A0, and B2. The candidate at position B2 is considered only when any CU at positions A0, B0, B1, or A1 is not available (e.g., the CU at position A0 belongs to another slice or tile) or not inter coded. After the candidate at position A1 is added to the merge candidate list, the addition of the remaining candidates is subject to a redundancy check that ensures that candidates with same motion information are excluded from the merge candidate list so that coding efficiency is improved.

[0012] FIG. 1B shows candidate pairs considered for the redundancy check of the spatial merge candidates. To reduce computational complexity, not all possible candidate pairs are considered in the redundancy check. Instead, only the pairs linked with an arrow in FIG. 1B are considered and a candidate is only added to the merge candidate list if the corresponding candidate used for the redundancy check has not the same motion information.

[0013] FIG. 1C shows a motion vector scaling for a temporal merge candidate. In some cases, only one temporal merge candidate can be added to the merge candidate list. Particularly, in the derivation of this temporal merge candidate, a scaled motion vector is derived based on a co-located CU belonging to the collocated reference picture. The reference picture list used for the derivation of the co-located CU is explicitly signaled in the slice header. The scaled motion vector for the temporal merge candidate is obtained as illustrated by the dotted line in FIG. 1C. The scaled motion vector is derived from the motion vector of the co-located CU using the picture order count (POC) distances tb and td, where tb is defined as the POC difference between

a reference picture of the current picture and the current picture and td is defined as the POC difference between a reference picture of the co-located picture and the co-located picture. The reference picture index of the temporal merge candidate can be set equal to zero.

[0014] FIG. 1D shows exemplary positions for the temporal merge candidate. The temporal merge candidate is selected between CUs at positions C0 and C1. If the CU at position C0 is not available, not inter coded, or outside of the current row of CTUs, the CU at position C1 is used. Otherwise, the CU at position C0 is used in the derivation of the temporal merge candidate.

SUMMARY

[0015] Aspects of the disclosure provide methods and apparatuses for video encoding/decoding. In some examples, an apparatus for video decoding includes processing circuitry.

[0016] According to aspects of the disclosure, there is provided a method for video decoding in a decoder. In the method, prediction information of a current block of a current picture in a coded bitstream is decoded. The prediction information indicates a geometric partitioning mode (GPM) for the current block. The current block is partitioned into two partitions in the GPM mode. Each of the partitions is associated with a respective predictor. A weighting index for a sample of the current block is determined based on a position of the sample. A weighting factor is calculated based on the weighting index of the sample according to an equation that converts the weighting index to the weighting factor. The sample is reconstructed based on the weighting factor and the predictor corresponding to the sample.

[0017] In an embodiment, a right shift operation is performed on a sum of the weighting index and an offset value. A result of the right shift operation is clipped to be within a predefined range.

[0018] In an embodiment, the offset value is based on a number of bits shifted by the right shift operation, and the number of bits shifted by the right shift operation is based on at least one of the weighting index and a size of a cosine table used to calculate the weighting index.

[0019] In an embodiment, an angle index and a distance index that define a split boundary between the partitions of the current block are determined based on the GPM. The

weighting index for the sample is determined based on the position of the sample, the angle index, and the distance index.

[0020] In an embodiment, a partition index is determined based on the angle index. The weighting factor is calculated based on the partition index.

[0021] In an embodiment, the equation is

$$weight = Clip3(0, 8, (wIdxL + (1 \ll (idx2wShiftBit - 1))) \gg idx2wShiftBit),$$

in which $wIdxL = (1 \ll (idx2wShiftBit + 2)) + (partIdx\ ?\ wIdx : -wIdx)$,

where $idx2wShiftBit$ indicates the number of bits shifted by the right shift operation, $weight$ is the weighting factor, $partIdx$ is the partition index, $wIdx$ is the weighting index, and $Clip3()$ is a clip function.

[0022] In an embodiment, the equation is a piecewise constant function that includes an initial value and a plurality of unit-step functions. The initial value is one of a minimum weighting factor value and a maximum weighting factor value, and a number of the plurality of unit-step functions is equal to a total number of different weighting factor values minus one.

[0023] Aspects of the disclosure provide an apparatus configured to perform any one or a combination of the methods for video decoding. In an embodiment, the apparatus includes processing circuitry that decodes prediction information of a current block of a current picture in a coded bitstream. The prediction information indicates a geometric partitioning mode (GPM) for the current block. The current block is partitioned into two partitions in the GPM mode. Each of the partitions is associated with a respective predictor. The processing circuitry determines a weighting index for a sample of the current block based on a position of the sample. The processing circuitry calculates a weighting factor based on the weighting index of the sample according to an equation that converts the weighting index to the weighting factor. The processing circuitry reconstructs the sample based on the weighting factor and the predictor corresponding to the sample.

[0024] Aspects of the disclosure also provide a non-transitory computer-readable medium storing instructions which when executed by a computer for video decoding cause the computer to perform any one or a combination of the methods for video decoding.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] 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:

[0026] FIG. 1A shows exemplary positions of spatial merge candidates;

[0027] FIG. 1B shows exemplary candidate pairs considered for a redundancy check of the spatial merge candidates;

[0028] FIG. 1C shows an example of motion vector scaling for a temporal merge candidate;

[0029] FIG. 1D shows exemplary positions for the temporal merge candidate;

[0030] FIG. 2 shows a schematic illustration of a simplified block diagram of a communication system in accordance with an embodiment;

[0031] FIG. 3 shows a schematic illustration of a simplified block diagram of a communication system in accordance with an embodiment;

[0032] FIG. 4 shows a schematic illustration of a simplified block diagram of a decoder in accordance with an embodiment;

[0033] FIG. 5 shows a schematic illustration of a simplified block diagram of an encoder in accordance with an embodiment;

[0034] FIG. 6 shows a block diagram of an encoder in accordance with another embodiment;

[0035] FIG. 7 shows a block diagram of a decoder in accordance with another embodiment;

[0036] FIGs. 8A and 8B show two exemplary triangular partitions in accordance with some embodiments;

[0037] FIG. 9 shows a uni-prediction motion vector selection for the triangle partition mode in accordance with some embodiments;

[0038] FIGs. 10A and 10B show exemplary weight maps for luma and chroma in accordance with some embodiments;

[0039] FIG. 11 shows an exemplary geometric partitioning mode according to some embodiments of the disclosure;

[0040] FIG. 12 shows a flow chart outlining an exemplary process in accordance with an embodiment; and

[0041] FIG. 13 shows a schematic illustration of a computer system in accordance with an embodiment.

DETAILED DESCRIPTION OF EMBODIMENTS

[0042] The present disclosure includes embodiments directed to geometric merge mode (GEO), which can also be referred to as geometric partitioning mode (GPM). The embodiments include methods, apparatuses, and non-transitory computer-readable storage mediums for improving the GEO. In addition, a block may refer to a prediction block, a coding block, or a coding unit.

[0043] **I. Video Encoder and Decoder**

[0044] 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.

[0045] 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.

[0046] 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.

[0047] 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.

[0048] 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.

[0049] 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.

[0050] 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.

[0051] 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).

[0052] 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.

[0053] 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).

[0054] 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.

[0055] 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.

[0056] 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).

[0057] In some cases, the output samples of the scaler / inverse transform (451) can pertain to an intra coded block; that is: a block that is not using predictive information from previously reconstructed pictures, but can use predictive information from previously reconstructed parts of the current picture. Such predictive information can be provided by an intra picture prediction unit (452). In some cases, the intra picture prediction unit (452) generates a block of the same size and shape of the block under reconstruction, using surrounding already reconstructed information fetched from the current picture buffer (458). The current picture buffer (458) buffers, for example, partly reconstructed current picture and/or fully reconstructed current picture. The aggregator (455), in some cases, adds, on a per sample basis, the prediction information that the intra prediction unit (452) has generated to the output sample information as provided by the scaler / inverse transform unit (451).

[0058] 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.

[0059] 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.

[0060] 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.

[0061] 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.

[0062] 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.

[0063] 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.

[0064] 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.

[0065] 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).

[0066] 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.

[0067] According to an embodiment, the video encoder (503) may code and compress the pictures of the source video sequence into a coded video sequence (543) in real time or under any other time constraints as required by the application. Enforcing appropriate coding speed is one function of a controller (550). In some embodiments, the controller (550) controls other functional units as described below and is functionally coupled to the other functional units. The coupling is not depicted for clarity. Parameters set by the controller (550) can include rate control related parameters (picture skip, quantizer, lambda value of rate-distortion optimization techniques, ...), picture size, group of pictures (GOP) layout, maximum motion vector allowed reference area, and so forth. The controller (550) can be configured to have other suitable functions that pertain to the video encoder (503) optimized for a certain system design.

[0068] 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.

[0069] The operation of the "local" decoder (533) can be the same as of a "remote" decoder, such as the video decoder (410), which has already been described in detail above in conjunction with FIG. 4. Briefly referring also to FIG. 4, however, as symbols are available and encoding/decoding of symbols to a coded video sequence by an entropy coder (545) and the parser (420) can be lossless, the entropy decoding parts of the video decoder (410), including the

buffer memory (415) and the parser (420) may not be fully implemented in the local decoder (533).

[0070] 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.

[0071] 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.

[0072] 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).

[0073] 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).

[0074] 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.

[0075] 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.

[0076] 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).

[0077] 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:

[0078] 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.

[0079] 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.

[0080] 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.

[0081] 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.

[0082] 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.

[0083] 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.

[0084] 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.

[0085] 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.

[0086] Further, a merge mode technique can be used in the inter-picture prediction to improve coding efficiency.

[0087] According to some embodiments of the disclosure, predictions, such as inter-picture predictions and intra-picture predictions are performed in the unit of blocks. For example, according to the HEVC standard, a picture in a sequence of video pictures is partitioned into coding tree units (CTU) for compression, the CTUs in a picture have the same size, such as 64x64 pixels, 32x32 pixels, or 16x16 pixels. In general, a CTU includes three coding tree blocks (CTBs), which are one luma CTB and two chroma CTBs. Each CTU can be recursively quad-tree split into one or multiple 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.

[0088] 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.

[0089] 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.

[0090] 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.

[0091] 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.

[0092] 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.

[0093] 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.

[0094] 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.

[0095] 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.

[0096] 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.

[0097] 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.

[0098] 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).

[0099] The inter decoder (780) is configured to receive the inter prediction information, and generate inter prediction results based on the inter prediction information.

[0100] The intra decoder (772) is configured to receive the intra prediction information, and generate prediction results based on the intra prediction information.

[0101] 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).

[0102] 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.

[0103] It is noted that the video encoders (303), (503), and (603), and the video decoders (310), (410), and (710) can be implemented using any suitable technique. In an embodiment, the video encoders (303), (503), and (603), and the video decoders (310), (410), and (710) can be implemented using one or more integrated circuits. In another embodiment, the video encoders (303), (503), and (603), and the video decoders (310), (410), and (710) can be implemented using one or more processors that execute software instructions.

[0104] **II. Triangle Partition Mode (TPM) for Inter Prediction**

[0105] In some cases, a TPM can be supported for inter prediction. The TPM can only be applied to CUs that are 8x8 or larger. The TPM can be signaled using a CU-level flag as one kind of merge mode, with other merge modes, such as the regular merge mode, the MMVD mode, the CIIP mode, and the subblock merge mode.

[0106] When the TPM is used, a CU can be split evenly into two triangle-shaped partitions, using either a diagonal split or an anti-diagonal split, as shown in FIGs 8A and 8B for example. Each triangle partition in the CU can be inter-predicted using its own motion parameters. Only uni-prediction can be allowed for each partition. That is, each partition has one motion vector and one reference index. The uni-prediction motion constraint is applied to ensure that the triangle partition is the same as the conventional bi-prediction. That is, only two motion compensated predictions are needed for each CU. The uni-prediction motion for each partition can be derived using the process described in FIGs. 1A-1D.

[0107] If the TPM is used for a current CU, then a flag indicating a direction (diagonal or anti-diagonal) of the TPM and two merge indices (one for each partition) can be further signaled. The maximum TPM candidate size can be signaled explicitly at the slice level and specify syntax binarization for the TMP merge indices. After predicting each of the triangular partitions, the sample values along the diagonal or anti-diagonal edge can be adjusted using a blending process with adaptive weighting values. After deriving the prediction signal for the whole CU, transform and quantization process can be further applied to the whole CU as in other prediction modes. Finally, the motion field of the CU that is predicted using the TPM can be stored.

[0108] In some cases, the TPM cannot be used in combination with sub-block transform (SBT). That is, when the signaled triangle mode is equal to 1, the cu_sbt_flag is inferred to be 0 without signaling.

[0109] The uni-prediction candidate list can be derived directly from the merge candidate list constructed according to the extended merge prediction process. Denote N as the index of the uni-prediction motion in the triangle uni-prediction candidate list. The LX motion vector of the N-th extended merge candidate, with X equal to the parity of N, is used as the N-th uni-prediction motion vector for TPM. These motion vectors are marked with "X" in FIG. 9. In case a corresponding LX motion vector of the N-the extended merge candidate does not exist, the L(1-X) motion vector of the same candidate is used instead as the uni-prediction motion vector for TPM.

[0110] After predicting each triangular partition using its own motion parameters, the blending process can be applied to the two prediction signals to derive samples around the diagonal or anti-diagonal edge. The following weighting values are used in the blending process: {7/8, 6/8, 5/8, 4/8, 3/8, 2/8, 1/8} for luma and {6/8, 4/8, 2/8} for chroma, as shown in FIGs. 10A and 10B.

[0111] In some cases, the motion vectors of a CU coded in TPM can be generated using the following process. If Mv1 and Mv2 are from different reference picture lists (e.g., one from L0 and the other from L1), then Mv1 and Mv2 are simply combined to form the bi-prediction motion vector. Otherwise, if Mv1 and Mv2 are from the same list, only uni-prediction motion Mv2 is stored.

[0112] **III. Geometric Merge Mode (GEO)**

[0113] Geometric merge mode (GEO), also referred to as geometric partitioning mode (GPM), can support a plurality of different partitioning manners. The partitioning manners can be defined by angles and edges. For example, 140 different partitioning manners can be differentiated by 32 angles (quantized between 0 and 360° with 11.25° of equal separation) and 5 edges relative to the center of a CU.

[0114] FIG. 11 shows an exemplary GEO. In FIG. 11, an angle φ_i represents a quantized angle between 0 and 360 degrees and a distance offset ρ_i represents a quantized offset of the largest distance ρ_{max}. The value of ρ_{max} can be geometrically derived by Eq. 1 for either a w or h that is equal to 8 and scaled with log2 scaled short edge length. The variables h and w represent the height and width of the current block. When φ is equal to 0°, ρ_{max} is equal to w/2. When φ is equal to 90°, ρ_{max} is equal to h/2. In Eq. 1, ρ_{margin} =1.0 is to prevent the split boundary from being too close to the corner of the current block.

$$\rho_{\max}(\varphi,w,h) = \cos(\varphi)\left(\frac{h}{2\tan\left(\frac{\pi}{2}-\varphi\right)}+\frac{w}{2}\right) - \rho_{\text{margin}},\ 0<\varphi<\frac{\pi}{2} \qquad \text{(Eq. 1)}$$

[0115] Each partition mode (i.e., a pair of an angle index and an edge index) in GEO can be assigned with a pixel-adaptive weight table to blend samples in the two partitioned parts. The weighting value of a sample can range from for example 0 to 8 and be determined by the L2 distance from the center position of a pixel to the edge. A unit-gain constraint can be followed when the weighting values are assigned. For example, when a small weighting value is assigned to a GEO partition, a large complementary one is assigned to the other partition, summing up to 8.

[0116] **IV. Weighted Sample Prediction Process for GEO**

[0117] In GEO, a final sample predictor P_B can be derived with two 3-bit blending masks (i.e., weighting values or weighting factors) W_0 and W_1 and two predictors P_0 and P_1 according to Eq. 2.

$$P_B = (W_0P_0 + W_1P_1 + 4) \gg 3 \qquad \text{(Eq. 2)}$$

[0118] The blending masks W_0 and W_1 can be derived from a look-up table based on their weighting indices. A weighting index can be derived based on a distance between the sample position (x, y) and the split boundary, as shown in Eq. 3.

$$wIdx(x,y) = x\times cos(\varphi) + y\times\cos\left(\varphi+\frac{\pi}{2}\right) - \left(\rho+\frac{w}{2}\times cos(\varphi)+\frac{h}{2}\times cos\left(\varphi+\frac{\pi}{2}\right)\right) \qquad \text{(Eq. 3)}$$

$$\text{where } \rho = i \times \frac{\rho_{\max}(\varphi,w,h)}{n} = i \times ((\frac{h}{2} \times tan(\varphi) + \frac{w}{2}) \times cos\,(\varphi) - \rho_{\text{margin}})/n$$
$$= i \times ((\frac{w}{2} \times cos\,(\varphi) + \frac{h}{2} \times cos(\varphi + \frac{\pi}{2})) - \rho_{\text{margin}})/n \quad \text{(Eq. 4)}$$

[0119] In Eq. 4, n represents the total distance quantization steps, and i represents the distance step index for the GEO with the angle φ, and i<n.

[0120] Finally, the blending masks W_0 and W_1 (or the weighting values) of the sample can be set using Table 3 denoted as GeoFilter, as shown in Eq. 5.

$$weight(x,y) = wIdx(x,y) \leq 0\ ?GeoFilter[[wIdx(x,y)]]:8 - GeoFilter[[wIdx(x,y)]] \quad \text{(Eq. 5)}$$

[0121] An exemplary weighted sample prediction process is described as follows. Inputs to this process include two variables nCbW and nCbH specifying the width and the height of the current coding block, two (nCbW)x(nCbH) arrays predSamplesLA and predSamplesLB, a variable angleIdx specifying the angle index of the geometric partition, a variable distanceIdx specifying the distance idx of the geometric partition, and a variable cIdx specifying a color component index. Outputs of this process include a (nCbW)x(nCbH) array pbSamples of prediction sample values and a (nCbW>>2)x(nCbH>>2) array motionIdx.

[0122] The variable bitDepth can be derived as follows: if cIdx=0, bitDepth=BitDepthY; otherwise, bitDepth=BitDepthC.

[0123] The variables shift1=Max(5, 17−bitDepth) and offset1=1<<(shift1−1).

[0124] The weights array sampleWeightL[x][y] for luma and sampleWeightC[x][y] for chroma with x=0..nCbW−1 and y=0..nCbH−1 can be derived as follows:

[0125] The variables wIdx=log2(nCbW) and hIdx=log2(nCbH).

[0126] The variable whRatio=(wIdx>= hIdx)?wIdx−hIdx:hIdx−wIdx, scaleIdx=(wIdx >= hIdx)?hIdx−3:wIdx−3.

[0127] The variables displacementX=angleIdx and displacementY=(displacementX+8)%32.

[0128] The variable angleN=(wIdx>=hIdx)?(angleIdx>>3&1)?angleIdx%8:8−angleIdx%8:(angleIdx>>3&1)?8−angleIdx%8:angleIdx%8.

[0129] The variable rho can be set to the following value using the look-up tables denoted as stepDis and Dis, specified in Table 1 and Table 2.

rho=distanceIdx*(stepDis[whRatio][angleN]<<scaleIdx)+(Dis[displacementX]<<wIdx)+(Dis[displacementY]<<hIdx).

[0130] The variable weightIdx and weightIdxAbs can be calculated using the look-up Table 2 with x=0..nCbW−1 and y=0…nCbH−1.

weightIdx=((x<<1)+1)*Dis[displacementX]+((y<<1)+1))*Dis[displacementY]−rho.

weightIdxAbs=Clip3(0, 26, (abs(weightIdx)+4) >>3).

[0131] The variable partIdx can be set to weightIdx>0, if x=0 and y=nCbH−1.

[0132] The value of sampleWeightL[x][y] with x=0..nCbW−1 and y=0..nCbH−1 can be set according to Table 3 denoted as GeoFilter.

sampleWeightL[x][y]=weightIdx<=0?GeoFilter[weightIdxAbs]:8−GeoFilter[weightIdxAbs].

[0133] The value sampleWeightC[x][y] with x = 0..nCbW−1 and y=0..nCbH−1 can be set as follows: sampleWeightC[x][y]=sampleWeightL[(x<<(SubWidthC−1))][(y<<(SubHeightC−1))].

Table 1

whRatio	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
angleN	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
stepDis[whRatio] [angleN]	77	95	108	116	119	116	108	95	77	77	115	147	173	192	202	203	195	179
whRatio	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
angleN	0	1	2	3	4	5	6	7	8	0	1	2	3	4	5	6	7	8
stepDis[whRatio] [angleN]	77	155	226	287	336	372	392	396	384	77	235	382	515	626	712	770	798	794

Table 2

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dis[idx]	64	63	59	53	45	36	24	12	0	-12	-24	-36	-45	-53	-59	-63
idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Dis[idx]	-64	-63	-59	-53	-45	-36	-24	-12	0	12	24	36	45	53	59	63

Table 3

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13
GeoFilter[idx]	4	4	4	4	5	5	5	5	5	5	5	6	6	6
idx	14	15	16	17	18	19	20	21	22	23	24	25	26	
GeoFilter[idx]	6	6	6	6	7	7	7	7	7	7	7	7	8	

[0134] In order to reduce the storage requirement of the precalculated blending masks, a minimum blending mask storage method can achieve an 84-91% memory requirement reduction for the blending weights storage.

[0135] Let $g_sampleWeight_L$ [] represent the pre-defined masks for blending weights. Assuming that N represents the number of pre-defined masks in each set, and N is set to NA>>1, where NA is the number of angles supported in GEO. M×M represents the size of the pre-defined masks for the blending weights, and M is set to 128+((ND–1)×(128>>S))<<1, where ND is the number of steps supported in GEO and S is set to ND–1. For the 32 angles with 5 steps setting, N is set to 16 and M is set to 192. For the 24 angles with 4 steps setting, N is set to 12 and M is set to 224.

[0136] For a block of size W×H with geometric partitioning index K, the blending weights for luma samples are derived as follows. Variables angle φ and distance ρ are obtained from a look-up table using the geometric partitioning index K. Variables offsetX and offsetY can be calculated as follows:

$$offsetX = \begin{cases} (M - W) \gg 1, \ \varphi \,\%\, N = 8 \text{ or } (\varphi \,\%\, N \neq 0 \text{ and H} \geq \text{W}) \\ ((M - W) \gg 1) + \varphi < N \,?(\rho \times W) \gg S : -((\rho \times W) \gg S), \ otherwise \end{cases} \quad \text{(Eq. 6)}$$

$$(offsetY = \begin{cases} ((M - H) \gg 1) + \varphi < N \,?(\rho \times H) \gg S : -((\rho \times H) \gg S), \\ \varphi \,\%\, N = 8 \text{ or } (\varphi \,\%\, N \neq 0 \text{ and H} \geq \text{W}) \\ (M - H) \gg 1, \ otherwise \end{cases} \quad \text{(Eq. 7)}$$

$$sampleWeight_L[x][y] = g_sampleWeight_L[\varphi \,\%\, N][x + offsetX][y + offsetY] \quad \text{(Eq. 8)}$$

[0137] An exemplary weighted sample prediction process with minimum blending weight mask storage for GEO is described as follows. Inputs to this process include two variables nCbW and nCbH specifying the width and the height of the current coding block, two (nCbW)x(nCbH) arrays predSamplesLA and predSamplesLB, a variable angleIdx specifying the angle index of the geometric partition, a variable distanceIdx specifying the distance idx of the geometric partition, and a variable cIdx specifying a color component index. Outputs of this process include a (nCbW)x(nCbH) array pbSamples of prediction sample values and a variable partIdx.

[0138] The variable bitDepth can be derived as follows: if cIdx=0, bitDepth=BitDepthY; otherwise, bitDepth=BitDepthC.

[0139] The variables shift1=Max(5, 17−bitDepth) and offset1=1 <<(shift1−1).

[0140] The weights array sampleWeightL[x][y] for luma and sampleWeightC[x][y] for chroma with x=0..nCbW−1 and y=0..nCbH−1 can be derived as follows:

[0141] The variable hwRatio=nCbH/nCbW.

[0142] The variables displacementX=angleIdx and displacementY=(displacementX+8)%32.

[0143] The variable partIdx=(angleIdx>=13&& angleIdx<=27) ? 1:0.

[0144] The variable rho can be set to the following value using the look-up Table 2. rho=(Dis[displacementX]<< 8)+(Dis[displacementY]<<8).

[0145] If one of the following conditions is true, the variable shiftHor=0: (1) angleIdx%16=8; and (2) angleIdx%16!=0 and hwRatio≥1. Otherwise, shiftHor=1.

[0146] If shiftHor=0, offsetX=(256−nCbW)>>1, offsetY=(256−nCbH)>>1+angleIdx<16?(distanceIdx*nCbH)>>3:−((distanceIdx*nCbH)>>3).

[0147] Otherwise, if shiftHor=1, offsetX=(256−nCbW)>>1+angleIdx<16?(distanceIdx*nCbW)>>3:−((distanceIdx*nCbW)>>3), offsetY=(256−nCbH)>>1.

[0148] The variable weightIdx and weightIdxAbs can be calculated using the look-up Table 2 with x=0..nCbW−1 and y=0..nCbH−1 as follows: weightIdx=(((x+offsetX)<<1)+1)*Dis[displacementX]+(((y+offsetY)<<1)+1))*Dis[displacemen tY]−rho, weightIdxAbs=Clip3(0, 26, abs(weightIdx)).

[0149] The value of sampleWeightL[x][y] with x=0..nCbW−1 and y=0..nCbH−1 can be set according to Table 3 denoted as GeoFilter. sampleWeightL[x][y]=weightIdx<=0?GeoFilter[weightIdxAbs]:8−GeoFilter[weightIdxAbs].

[0150] The value sampleWeightC[x][y] with x=0..nCbW−1 and y=0..nCbH−1 can be set as follows: sampleWeightC[x][y]=sampleWeightL[(x<<(SubWidthC−1))][(y<<(SubHeightC−1))].

[0151] **V. Motion Vector Storing Process for GEO**

[0152] In some cases, luminance sample weights at four corners of a 4x4 motion storage unit can be summed up. The sum can be compared with 2 thresholds to determine whether one of two uni-prediction motion information and bi-prediction motion information is stored. The bi-prediction motion information can be derived using the same process as TPM.

[0153] An exemplary motion vector storage process for GEO is described as follows.

[0154] The array motionIdx[xSbIdx][ySbIdx] with xSbIdx=0.. (nCbW>>2)−1 and ySbIdx=0..(nCbH>>2)−1 can be derived as follows:

[0155] The variables threshScaler=(wIdx+hIdx)>>1)−1, threshLower =32>>threshScaler, threshUpper=32−threshLower, Cnt=sampleWeightL[(xSbIdx<<2)][(ySbIdx<<2)]+sampleWeightL[(xSbIdx<<2)+3][(ySbIdx<<2)]+sampleWeightL[(xSbIdx<<2)][(ySbIdx<<2)+3]+sampleWeightL[(xSbIdx<<2)+3][(ySbIdx<<2)+3], Cnt=partIdx?32–Cnt:Cnt, motionIdx[xSbIdx][ySbIdx]=Cnt<=threshLower?0:Cnt>=threshUpper?1:2.

[0156] If the merge_geo_flag[xCb][yCb]=1, sType=motionIdx[xSbIdx][ySbIdx] with xSbIdx=0..numSbX−1, and ySbIdx=0..numSbY−1.

[0157] In some cases, the motion vector storage process is further simplified. The distance between the central position of a 4x4 motion storage unit and the split boundary can be calculated and compared with a fixed threshold to determine whether uni- or bi-prediction motion information is stored for the 4x4 motion storage unit. The sign of the distance indicates which uni-prediction motion information should be stored in the uni-prediction storage case. The dependency of blending mask and motion storage can be removed.

[0158] If the merge_geo_flag[xCb][yCb]=1, wIdx=log2(cbWidth), hIdx=log2(cbHeight), whRatio=(wIdx >= hIdx)?wIdx−hIdx:hIdx−wIdx, scaleIdx=(wIdx>=hIdx)?hIdx−3:wIdx−3, displacementX=angleIdx, displacementY=(displacementX+8)%32, angleN=(wIdx>=hIdx)?(angleIdx>>3&1)?angleIdx% 8:8−angleIdx%8:(angleIdx>>3&1)?8−angleIdx%8:angleIdx%8.

[0159] The variable rho is set to the following value using the look-up tables denoted as stepDis and Dis, specified in Table 1 and Table 2. rho=distanceIdx*(stepDis[whRatio][angleN]<<scaleIdx)+(Dis[displacementX]<<wIdx)+(Dis[displacementY]<<hIdx) .

[0160] The variable motionOffset is set to the following value using the look-up tables denoted as Dis, specified in Table 1 and Table 2.

motionOffset=3*Dis[displacementX]+3*Dis[displacementY] .

[0161] The variable motionIdx is calculated using the look-up Table 2 as follows: motionIdx=((xSbIdx <<3)+1)*Dis[displacementX]+((xSbIdx<<3)+1))*Dis[displacementY]–rho+motionOffset.

[0162] The variable sType is derived as follows: if partIdx=1, sType=abs(motionIdx)<256?2:motionIdx<=0?1:0; otherwise, sType=abs(motionIdx)<256?2:motionIdx <= 0?0:1.

[0163] To reduce the memory needed for storing the masks for motion field storage, in a process, all the information from a pre-defined mask can be derived for the masks of motion field storage. This process is invoked when decoding a coding unit with MergeWedgeFlag[xCb][yCb]=1. Inputs to this process include a luma location (xCb, yCb) specifying the top-left sample of the current coding block relative to the top left luma sample of the current picture, a variable cbWidth specifying the width of the current coding block in luma samples, a variable cbHeight specifying the height of the current coding block in luma samples, the luma motion vectors in 1/16 fractional-sample accuracy mvA and mvB, the reference indices refIdxA and refIdxB, and the prediction list flags predListFlagA and predListFlagB.

[0164] The variables numSbX and numSbY specifying the number of 4x4 blocks in the current coding block in horizontal and vertical direction are set equal to numSbX=cbWidth>>2 and numSbY=cbHeight>>2.

[0165] The variables displacementX=angleIdx, displacementY=(displacementX+8)%32, hwRatio=nCbH / nCbW.

[0166] If one of the following conditions is true, the variable shiftHor=0: (1) angleIdx%16=8; and (2) angleIdx%16!=0 and hwRatio≥1. Otherwise, shiftHor=1.

[0167] The variable partIdx=(angleIdx>=13&&angleIdx<=27)?1:0.

[0168] If shiftHor=0, offsetX=(64–numSbX)>>1, offsetY=(64−numSbY)>>1+angleIdx<16?(distanceIdx*nCbH)>> 5:−((distanceIdx*nCbH)>>5). Otherwise, if shiftHor=1, offsetX=(64–numSbX)>>1+angleIdx<16?(distanceIdx*nCbW)>>5:−((distanceIdx*nCbW)>> 5), offsetY=(64– numSbY)>>1.

[0169] The value of the variable rho is derived according to the equation below and the Dis lookup table specified in Table 2. rho=(Dis[displacementX]<<8)+(Dis[displacementY]<<8).

[0170] The variable motionOffset is set equal to the following value using the look-up tables denoted as Dis, specified in Table 1 and Table 2. motionOffset=3*Dis[displacementX]+3*Dis[displacementY] .

[0171] For each 4x4 subblock at subblock index (xSbIdx, ySbIdx) with xSbIdx=0..numSbX−1, and ySbIdx=0..numSbY−1, the variable motionIdx is calculated using the look-up Table 2 as follows:

motionIdx=(((xSbIdx+offsetX)<<3)+1)*Dis[displacementX]+(((xSbIdx+offsetY<<3)+1))*Dis[displacementY]–rho+motionOffset.

[0172] **VI. On-the-fly Weighting Index to Weighting Value Calculation**

[0173] In the above weighted sample process for GEO, the weighting index wIdx(x, y) to the weighting value (or weighting factor) weight(x, y) conversion can be derived based on a linear function, as shown in Eq. 9-Eq.11.

$$weightIdxL(x,y) = fl2intOp(\ s \times wIdx(x,y) + bias) \quad \text{(Eq. 9)}$$

$$wValue(x,y) = min(maxValue, weightIdxL(x,y)) \quad \text{(Eq. 10)}$$

$$weight(x,y) = (wIdx(x,y) < 0)\ ?\ wValue(x,y) : -wValue(x,y) \quad \text{(Eq. 11)}$$

[0174] In an example, maxValue=8, bias=4, the quantization step size s=1/7.2. The fl2intOp operation is used to convert a floating value to an integer number and can be a combination of round, floor, or ceil. A look-up table, such as Table 3, is an exemplary implementation method for the above equations.

[0175] However, since different samples may have different input values for the look-up table, multiple copies of the look-up table may be needed to perform parallel processing on these samples when the above equations are used. Accordingly, the above weighted sample process for GEO is not friendly for hardware and software implementation. To address this issue, this disclosure includes embodiments for calculation based conversion so that the weighting values for different samples can be derived in parallel.

[0176] According to aspects of the disclosure, the weighting index wIdx(x, y) to the weighting value weight(x, y) conversion can be derived based on an on-the-fly calculation with a right shift operation.

[0177] In some embodiments, the on-the-fly calculation is a sum of an offset value and the weighting index wIdx(x, y) and the offset value is a function of the right shift operand. In addition, the offset value can be rounded.

[0178] In an embodiment, the weighting index wIdx(x, y) to the weighting value weight(x, y) conversion can be derived according to Eq. 12.

$$weight(x,y) = \left(2^{idx2wShiftBit+2} - wIdx(x,y)\right) \gg idx2wShiftBit \quad \text{(Eq. 12)}$$

[0179] In Eq. 12, the offset value $2^{idx2wShiftBit+2}$ is an exponential function (with base 2) of the right shift operand idx2wShiftBit. In an example, the right shift operand idx2wShiftBit=m+3, and the weighting value weight(x, y)=(2^{m+5}-wIdx(x, y))>>(m+3). m can represent a precision bit number (e.g., 3 or 6) of the cosine table (e.g., Table 2) that is used to calculate the weighting index.

[0180] In an embodiment, the weighting index wIdx(x, y) to the weighting value weight(x, y) conversion can be derived according to Eq. 13 or Eq. 14.

$$weight(x,y) = Clip3(0, 8, (idx2wOffset - wIdx(x,y)) \gg idx2wShiftBit) \quad \text{(Eq. 13)}$$

$$weight(x,y) = Clip3(0, 8, (idx2wOffset + wIdx(x,y)) \gg idx2wShiftBit) \quad \text{(Eq. 14)}$$

[0181] In Eq. 13 and Eq. 14, a clip function Clip3() is used to limit the weighting value weight(x, y) within a predefined range, such as [0, 8]. In an example, the offset value idx2wOffset is a linear function of the right shift operand idx2wShiftBit, as shown in Eq. 15.

$$idx2wOffset = 1 \ll (idx2wShiftBit + 2) \quad \text{(Eq. 15)}$$

[0182] The right shift operand idx2wShiftBit can be the precision bit number (e.g., 3, 6, or 9) of the cosine table (e.g., Table 2) that is used to calculate the weighting index and/or a calculated distance value (e.g., the weighting index).

[0183] In an embodiment, the weighting index can have a sign conversion according to a partition index partIdx, as shown in Eq. 16.

$$wIdxFlip(x,y) = partIdx \ ? -wIdx(x,y) : wIdx(x,y) \quad \text{(Eq. 16)}$$

[0184] Accordingly, the weighting value can be derived according to Eq. 17 or Eq. 18.

$$weight(x,y) = Clip3(0, 8,(idx2wOffset - wIdxFlip(x,y)) \gg idx2wShiftBit) \quad \text{(Eq. 17)}$$

$$weight(x,y) = Clip3(0, 8,(idx2wOffset + wIdxFlip(x,y)) \gg idx2wShiftBit) \quad \text{(Eq. 18)}$$

[0185] In an example, the partition index partIdx can be set according to the weighting index of the left-bottom corner sample of the current block. In another example, the partition index partIdx can be set according to the angle index angleIdx in Eq. 19, where T_1 and T_2 are two thresholds. In one embodiment, T_1=10 and T_2=20.

$$partIdx = (angleIdx \geq T_1 \ \&\& \ angleIdx \leq T_2) \ ? \ 0 : 1 \quad \text{(Eq. 19)}$$

[0186] In an embodiment, the partition index partIdx can be used as a definition of a partition for different angles. The weighting value can be derived according to Eq. 20 and Eq. 21.

$$wIdxL = (1 \ll (idx2wShiftBit + 2)) + (partIdx \ ? \ wIdx : \ -wIdx) \quad \text{(Eq. 20)}$$

$$weight = Clip3(0, 8,(wIdxL + (1 \ll (idx2wShiftBit - 1))) \gg idx2wShiftBit) \quad \text{(Eq. 21)}$$

[0187] In an embodiment, the weighting value can be derived according to Eq. 22 and Eq. 23.

$$wIdxL = (1 \ll (idx2wShiftBit + 2)) + wIdx \quad \text{(Eq. 22)}$$

$$weight = Clip3(0, 8,(wIdxL + (1 \ll (idx2wShiftBit - 1))) \gg idx2wShiftBit) \quad \text{(Eq. 23)}$$

[0188] Then, the final blending mask of the sample of the current block can be flipped according to the condition of the angle index angleIdx, which is described in Eq. 19.

[0189] According to aspects of the disclosure, the weighting index wIdx(x, y) to the weighting value weight(x, y) conversion can be derived based on a piecewise constant function. The piecewise constant function can include an initial value and a plurality of uni-step functions. Among the range of weight factor values, the initial value can be one of a minimum weight factor value and a maximum weighting factor value. Further, a number of the plurality of unit-step functions can be equal to a total number of the weighting factor values minus 1.

[0190] In an embodiment, the weight value weight(x, y) can be derived from a predefined initial value smallestWeight and four unit-step functions, as shown in Eq. 24.

$$weight(x,y) = smallestWeight + \sum_{i=0}^{3} \omega_i(v) \quad \text{(Eq. 24)}$$

where v is the weighting index and $\omega_i(v)$ can be derived according to Eq. 25 or Eq. 26.

$$\omega_i(v) = \begin{cases} 1 & \text{if } v \geq c_i, \\ 0 & \text{otherwise} \end{cases} \quad \text{(Eq. 25)}$$

$$\omega_i(v) = \begin{cases} 1 & \text{if } v > p_i, \\ 0 & \text{otherwise} \end{cases} \quad \text{(Eq. 26)}$$

[0191] Table 4 shows an exemplary look-up table for mapping the weighting index to the weighting value.

Table 4

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13
GeoFilter[idx]	4	4	4	4	5	5	5	5	5	5	5	6	6	6
idx	14	15	16	17	18	19	20	21	22	23	24	25	26	
GeoFilter[idx]	6	6	6	6	7	7	7	7	7	7	7	8	8	

[0192] If using Eq. 24 and Eq. 25 to represent the Table 4, the initial value smallestWeight=4, and the threshold weighting indices C_0-C_3 can be 4, 11, 18, and 25, respectively.

[0193] If using Eq. 24 and Eq. 26 to represent the Table 4, the initial value smallestWeight=4, and the threshold weighting indices P_0-P_3 can be 3, 10, 17, and 24, respectively.

[0194] Table 5 shows another exemplary look-up table for mapping the weighting index to the weighting value.

Table 5

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13
GeoFilter[idx]	4	4	4	5	5	5	5	5	5	5	6	6	6	6
idx	14	15	16	17	18	19	20	21	22	23	24	25	26	
GeoFilter[idx]	6	6	6	7	7	7	7	7	7	7	7	7	8	

[0195] If using Eq. 24 and Eq. 25 to represent the Table 5, the initial value smallestWeight=4, and the threshold weighting indices C_0-C_3 can be 3, 10, 17, and 26, respectively.

[0196] If using Eq. 24 and Eq. 26 to represent the Table 5, the initial value smallestWeight=4, and the threshold weighting indices P_0-P_3 can be 2, 9, 16, and 25, respectively.

[0197] In an embodiment, the weight value weight(x, y) can be derived from a predefined initial value largestWeight and four unit-step functions, as shown in Eq. 27.

$$weight(x,y) = largestWeight - \sum_{i=0}^{3} \omega_i(v) \quad \text{(Eq. 27)}$$

where $\omega_i(v)$ can be derived according to Eq. 28 or Eq. 29.

$$\omega_i(v) = \begin{cases} 1 & \text{if } v \le c_i \\ 0 & \text{otherwise} \end{cases} \quad \text{(Eq. 28)}$$

$$\omega_i(v) = \begin{cases} 1 & \text{if } v < p_i \\ 0 & \text{otherwise} \end{cases} \quad \text{(Eq. 29)}$$

[0198] If using Eq. 27 and Eq. 28 to represent the Table 4, the initial value largestWeight=8, and the threshold weighting indices C_0-C_3 can be 3, 10, 17, and 24, respectively.

[0199] If using Eq. 27 and Eq. 29 to represent the Table 4, the initial value largestWeight=8, and the threshold weighting indices P_0-P_3 can be 4, 11, 18, and 25, respectively.

[0200] If using Eq. 27 and Eq. 28 to represent the Table 5, the initial value largestWeight=8, and the threshold weighting indices C_0-C_3 can be 2, 9, 16, and 25, respectively.

[0201] If using Eq. 27 and Eq. 29 to represent the Table 5, the initial value largestWeight=8, and the threshold weighting indices P_0-P_3 can be 3, 10, 17, and 26, respectively.

[0202] It is noted that the relational operators (≥) and (>) yield 1 if the corresponding relation is true and 0 if the corresponding relation is false, as defined in C and C++ for example.

[0203] **VII. Flowchart**

[0204] FIG. 12 shows a flow chart outlining an exemplary process (1200) according to an embodiment of the disclosure. In various embodiments, the process (1200) is executed by processing circuitry, such as the processing circuitry in the terminal devices (210), (220), (230) and (240), the processing circuitry that performs functions of the video encoder (303), the processing circuitry that performs functions of the video decoder (310), the processing circuitry that performs functions of the video decoder (410), the processing circuitry that performs functions of the intra prediction module (452), the processing circuitry that performs functions of the video encoder (503), the processing circuitry that performs functions of the predictor (535), the processing circuitry that performs functions of the intra encoder (622), the processing circuitry that performs functions of the intra decoder (772), and the like. In some embodiments, the process (1200) is implemented in software instructions, thus when the processing circuitry executes the software instructions, the processing circuitry performs the process (1200).

[0205] The process (1200) may generally start at step (S1210), where the process (1200) decodes prediction information of a current block of a current picture in a coded bitstream. The prediction information indicates a geometric partitioning mode (GPM) for the current block. The current block is partitioned into two partitions in the GPM mode. Each of the partitions is associated with a respective predictor. Then, the process (1200) proceeds to step (S1220).

[0206] At step (S1220), the process (1200) determines a weighting index for a sample of the current block based on a position of the sample. Then, the process (1200) proceeds to step (S1230).

[0207] At step (S1230), the process (1200) calculates a weighting factor based on the weighting index of the sample according to an equation that converts the weighting index to the weighting factor. Then, the process (1200) proceeds to step (S1240).

[0208] At step (S1240), the process (1200) reconstructs the sample based on the weighting factor and the predictor corresponding to the sample. After reconstructing the sample, the process (1200) terminates.

[0209] In an embodiment, a right shift operation is performed on a sum of the weighting index and an offset value. A result of the right shift operation is clipped to be within a predefined range.

[0210] In an embodiment, the offset value is based on a number of bits shifted by the right shift operation, and the number of bits shifted by the right shift operation is based on at least one of the weighting index and a size of a cosine table used to calculate the weighting index.

[0211] In an embodiment, an angle index and a distance index that define a split boundary between the partitions of the current block are determined based on the GPM. The weighting index for the sample is determined based on the position of the sample, the angle index, and the distance index.

[0212] In an embodiment, a partition index is determined based on the angle index. The weighting factor is calculated based on the partition index.

[0213] In an embodiment, the equation is

$$weight = Clip3(0, 8, (wIdxL + (1 \ll (idx2wShiftBit - 1))) \gg idx2wShiftBit),$$

wherein $$wIdxL = (1 \ll (idx2wShiftBit + 2)) + (partIdx\ ?\ wIdx : -wIdx),$$

where $idx2wShiftBit$ indicates the number of bits shifted by the right shift operation, $partIdx$ is the partition index, and $wIdx$ is the weighting index.

[0214] In an embodiment, the equation is a piecewise constant function that includes an initial value and a plurality of unit-step functions. The initial value is one of a minimum weighting factor value or a maximum weighting factor value, and a number of the plurality of unit-step functions is equal to a total number of different weighting factor values minus one.

[0215] VIII. Computer System

[0216] 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. 13 shows a computer system (1300) suitable for implementing certain embodiments of the disclosed subject matter.

[0217] 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.

[0218] 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.

[0219] The components shown in FIG. 13 for computer system (1300) 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 (1300).

[0220] Computer system (1300) 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).

[0221] Input human interface devices may include one or more of (only one of each depicted): keyboard (1301), mouse (1302), trackpad (1303), touch screen (1310), data-glove (not shown), joystick (1305), microphone (1306), scanner (1307), camera (1308).

[0222] Computer system (1300) 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 (1310), data-glove (not shown), or joystick (1305), but there can also be tactile feedback devices that do not serve as input devices), audio output devices (such as: speakers (1309), headphones

(not depicted)), visual output devices (such as screens (1310) to include CRT screens, LCD screens, plasma screens, OLED screens, each with or without touch-screen input capability, each with or without tactile feedback capability—some of which may be capable to output two dimensional visual output or more than three dimensional output through means such as stereographic output; virtual-reality glasses (not depicted), holographic displays and smoke tanks (not depicted)), and printers (not depicted). These visual output devices (such as screens (1310)) can be connected to a system bus (1348) through a graphics adapter (1350).

[0223] Computer system (1300) can also include human accessible storage devices and their associated media such as optical media including CD/DVD ROM/RW (1320) with CD/DVD or the like media (1321), thumb-drive (1322), removable hard drive or solid state drive (1323), 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.

[0224] 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.

[0225] Computer system (1300) can also include a network interface (1354) to one or more communication networks (1355). The one or more communication networks (1355) can for example be wireless, wireline, optical. The one or more communication networks (1355) can further be local, wide-area, metropolitan, vehicular and industrial, real-time, delay-tolerant, and so on. Examples of the one or more communication networks (1355) 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 (1349) (such as, for example USB ports of the computer system (1300)); others are commonly integrated into the core of the computer system (1300) 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 (1300) 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.

[0226] Aforementioned human interface devices, human-accessible storage devices, and network interfaces can be attached to a core (1340) of the computer system (1300).

[0227] The core (1340) can include one or more Central Processing Units (CPU) (1341), Graphics Processing Units (GPU) (1342), specialized programmable processing units in the form of Field Programmable Gate Areas (FPGA) (1343), hardware accelerators for certain tasks (1344), and so forth. These devices, along with Read-only memory (ROM) (1345), Random-access memory (1346), internal mass storage such as internal non-user accessible hard drives, SSDs, and the like (1347), may be connected through the system bus (1348). In some computer systems, the system bus (1348) 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 (1348), or through a peripheral bus (1349). Architectures for a peripheral bus include PCI, USB, and the like.

[0228] CPUs (1341), GPUs (1342), FPGAs (1343), and accelerators (1344) can execute certain instructions that, in combination, can make up the aforementioned computer code. That computer code can be stored in ROM (1345) or RAM (1346). Transitional data can be also be stored in RAM (1346), whereas permanent data can be stored for example, in the internal mass storage (1347). 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 (1341), GPU (1342), mass storage (1347), ROM (1345), RAM (1346), and the like.

[0229] 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.

[0230] As an example and not by way of limitation, the computer system having architecture (1300), and specifically the core (1340) 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 (1340) that are of non-transitory nature, such as core-internal mass storage

(1347) or ROM (1345). The software implementing various embodiments of the present disclosure can be stored in such devices and executed by core (1340). A computer-readable medium can include one or more memory devices or chips, according to particular needs. The software can cause the core (1340) 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 (1346) 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 (1344)), 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.

[0231] While this disclosure has described several exemplary embodiments, there are alterations, permutations, and various substitute equivalents, which fall within the scope of the disclosure. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the disclosure and are thus within the spirit and scope thereof.

Appendix A: Acronyms

AMT: Adaptive Multiple Transform

AMVP: Advanced Motion Vector Prediction

ASIC: Application-Specific Integrated Circuit

ATMVP: Alternative/Advanced Temporal Motion Vector Prediction

BDOF: Bi-directional Optical Flow

BDPCM (or RDPCM): Residual Difference Pulse Coded Modulation

BIO: Bi-directional Optical Flow

BMS: Benchmark Set

BT: Binary Tree

BV: Block Vector

CANBus: Controller Area Network Bus

CB: Coding Block

CBF: Coded Block Flag

CCLM: Cross-Component Linear Mode/Model

CD: Compact Disc

CPR: Current Picture Referencing

CPU: Central Processing Unit

CRT: Cathode Ray Tube

CTB: Coding Tree Block

CTU: Coding Tree Unit

CU: Coding Unit

DM: Derived Mode

DPB: Decoder Picture Buffer

DVD: Digital Video Disc

EMT: Enhanced Multiple Transform

FPGA: Field Programmable Gate Areas

GOP: Group of Picture

GPU: Graphics Processing Unit

GSM: Global System for Mobile communications

HDR: High Dynamic Range

HEVC: High Efficiency Video Coding

HRD: Hypothetical Reference Decoder

IBC: Intra Block Copy

IC: Integrated Circuit

IDT: Identify Transform

ISP: Intra Sub-Partitions

JEM: Joint Exploration Model

JVET: Joint Video Exploration Team

LAN: Local Area Network

LCD: Liquid-Crystal Display

LFNST: Low Frequency Non-Separable Transform, or Low Frequency Non-Separable Secondary Transform

LTE: Long-Term Evolution

L_CCLM: Left-Cross-Component Linear Mode/Model

LT_CCLM: Left and Top Cross-Component Linear Mode/Model

MIP: Matrix based Intra Prediction

MPM: Most Probable Mode

MRLP (or MRL): Multiple Reference Line Prediction

MTS: Multiple Transform Selection

MV: Motion Vector

NSST: Non-Separable Secondary Transform

OLED: Organic Light-Emitting Diode

PBs: Prediction Blocks

PCI: Peripheral Component Interconnect

PDPC: Position Dependent Prediction Combination

PLD: Programmable Logic Device

PPR: Parallel-Processable Region

PPS: Picture Parameter Set

PU: Prediction Unit

QT: Quad-Tree

RAM: Random Access Memory

ROM: Read-Only Memory

RST: Reduced-Size Transform

SBT: Sub-block Transform

SCC: Screen Content Coding

SCIPU: Small Chroma Intra Prediction Unit

SDR: Standard Dynamic Range

SEI: Supplementary Enhancement Information

SNR: Signal Noise Ratio

SPS: Sequence Parameter Set

SSD: Solid-state Drive

SVT: Spatially Varying Transform

TSM: Transform Skip Mode

TT: Ternary Tree

TU: Transform Unit

T_CCLM: Top Cross-Component Linear Mode/Model

USB: Universal Serial Bus

VPDU: Visual Process Data Unit

VPS: Video Parameter Set

VUI: Video Usability Information

VVC: Versatile Video Coding

WAIP: Wide-Angle Intra Prediction

WHAT IS CLAIMED IS:

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

decoding prediction information of a current block of a current picture in a coded bitstream, the prediction information indicating a geometric partitioning mode (GPM) for the current block, the current block being partitioned into two partitions in the GPM mode, and each of the partitions being associated with a respective predictor;

determining a weighting index for a sample of the current block based on a position of the sample;

calculating a weighting factor based on the weighting index of the sample according to an equation that converts the weighting index to the weighting factor; and

reconstructing the sample based on the weighting factor and the predictor corresponding to the sample.

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

performing a right shift operation on a sum of the weighting index and an offset value; and

clipping a result of the right shift operation to be within a predefined range.

3. The method of claim 2, wherein the offset value is based on a number of bits shifted by the right shift operation, and the number of bits shifted by the right shift operation is based on at least one of the weighting index and a size of a cosine table used to calculate the weighting index.

4. The method of claim 1, wherein the determining the weighting index comprises:

determining an angle index and a distance index that define a split boundary between the partitions of the current block based on the GPM; and

determining the weighting index for the sample based on the position of the sample, the angle index, and the distance index.

5. The method of claim 1, wherein the calculating comprises:

determining a partition index based on the angle index; and

calculating the weighting factor based on the partition index.

6. The method of claim 5, wherein the equation is $weight = Clip3$

$(0, 8,(wIdxL + (1 \ll (idx2wShiftBit - 1))) \gg idx2wShiftBit),$

wherein $wIdxL = (1 \ll (idx2wShiftBit + 2)) + (partIdx\ ?\ wIdx : -wIdx),$

where $idx2wShiftBit$ indicates the number of bits shifted by the right shift operation, $weight$ is the weighting factor, $partIdx$ is the partition index, $wIdx$ is the weighting index, and $Clip3()$ is a clipping function.

7. The method of claim 1, wherein the equation is a piecewise constant function that includes an initial value and a plurality of unit-step functions.

8. The method of claim 7, wherein the initial value is one of a minimum weighting factor value and a maximum weighting factor value, and a number of the plurality of unit-step functions is equal to a total number of different weighting factor values minus one.

9. An apparatus, comprising processing circuitry configured to:

decode prediction information of a current block of a current picture in a coded bitstream, the prediction information indicating a geometric partitioning mode (GPM) for the current block, the current block being partitioned into two partitions in the GPM mode, and each of the partitions being associated with a respective predictor;

determine a weighting index for a sample of the current block based on a position of the sample;

calculate a weighting factor based on the weighting index of the sample according to an equation that converts the weighting index to the weighting factor; and

reconstruct the sample based on the weighting factor and the predictor corresponding to the sample.

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

perform a right shift operation on a sum of the weighting index and an offset value; and

clip a result of the right shift operation to be within a predefined range.

11. The apparatus of claim 10, wherein the offset value is based on a number of bits shifted by the right shift operation, and the number of bits shifted by the right shift operation is based on at least one of the weighting index and a size of a cosine table used to calculate the weighting index.

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

determine an angle index and a distance index that define a split boundary between the partitions of the current block based on the GPM; and

determine the weighting index for the sample based on the position of the sample, the angle index, and the distance index.

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

determine a partition index based on the angle index; and

calculate the weighting factor based on the partition index.

14. The apparatus of claim 13, wherein the equation is $weight = Clip3(0, 8, (wIdxL + (1 \ll (idx2wShiftBit - 1))) \gg idx2wShiftBit)$,

wherein $wIdxL = (1 \ll (idx2wShiftBit + 2)) + (partIdx\ ?\ wIdx : -wIdx)$,

where $idx2wShiftBit$ indicates the number of bits shifted by the right shift operation, $weight$ is the weighting factor, $partIdx$ is the partition index, $wIdx$ is the weighting index, and $Clip3()$ is a clip function.

15. The apparatus of claim 9, wherein the equation is a piecewise constant function that includes an initial value and a plurality of unit-step functions.

16. The apparatus of claim 15, wherein the initial value is one of a minimum weighting factor value and a maximum weighting factor value, and a number of the plurality of unit-step functions is equal to a total number of different weighting factor values minus one.

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

decoding prediction information of a current block of a current picture in a coded bitstream, the prediction information indicating a geometric partitioning mode (GPM) for the current block, the current block being partitioned into two partitions in the GPM mode, and each of the partitions being associated with a respective predictor;

determining a weighting index for a sample of the current block based on a position of the sample;

calculating a weighting factor based on the weighting index of the sample according to an equation that converts the weighting index to the weighting factor; and

reconstructing the sample based on the weighting factor and the predictor corresponding to the sample.

18. The non-transitory computer-readable storage medium of claim 17, wherein the calculating comprises:

performing a right shift operation on a sum of the weighting index and an offset value; and

clipping a result of the right shift operation to be within a predefined range.

19. The non-transitory computer-readable storage medium of claim 18, wherein the offset value is based on a number of bits shifted by the right shift operation, and the number of bits shifted by the right shift operation is based on at least one of the weighting index and a size of a cosine table used to calculate the weighting index.

20. The non-transitory computer-readable storage medium of claim 17, wherein the determining the weighting index comprises:

determining an angle index and a distance index that define a split boundary between the partitions of the current block based on the GPM; and

determining the weighting index for the sample based on the position of the sample, the angle index, and the distance index.

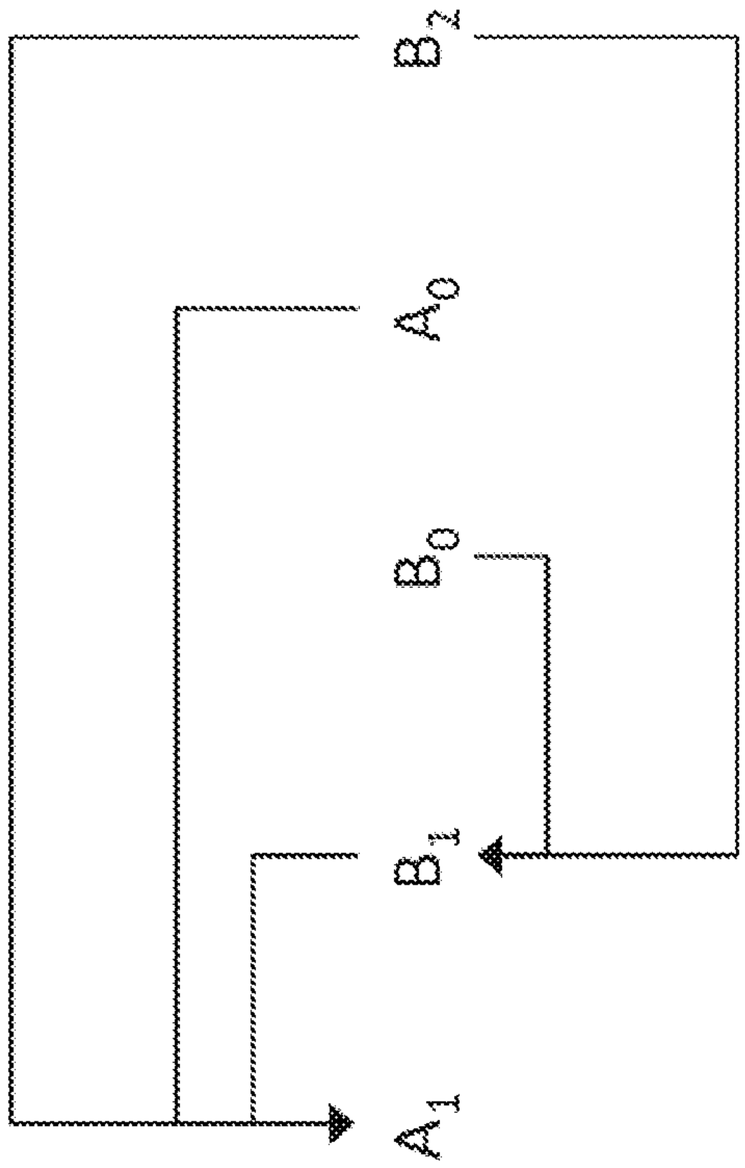


FIG. 1A

FIG. 1B

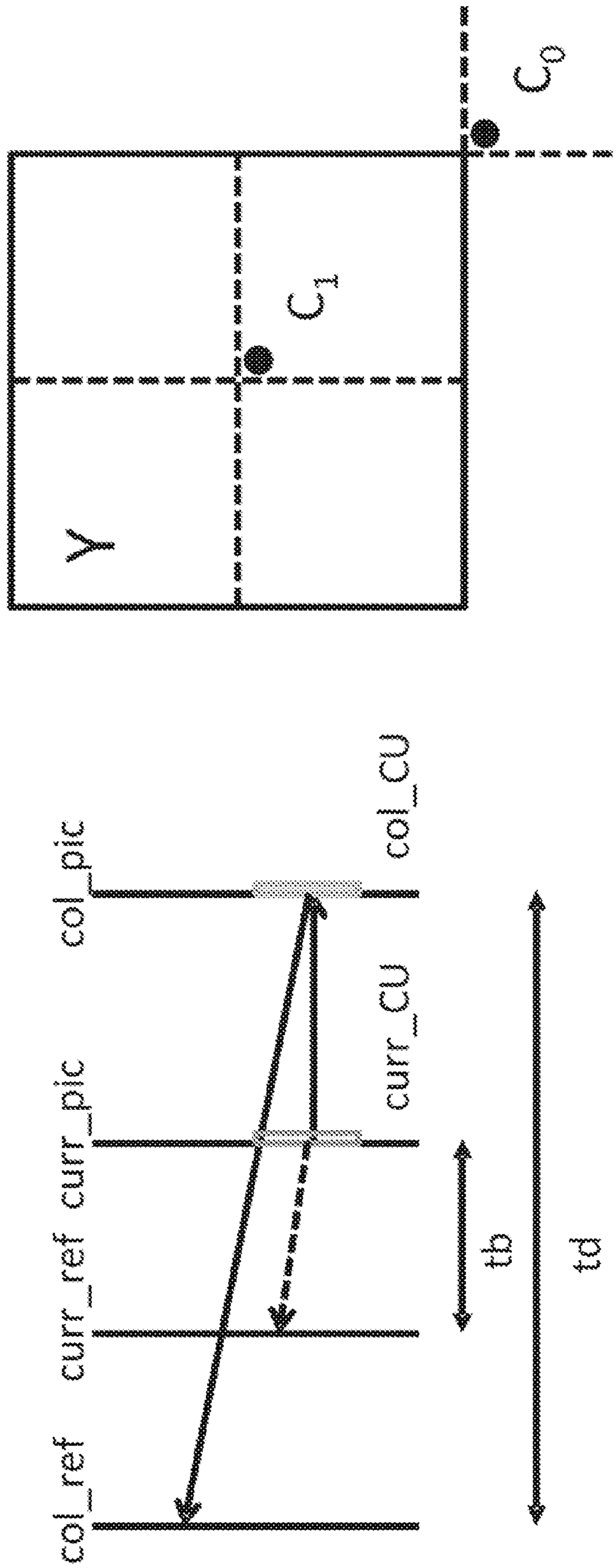


FIG. 1C

FIG. 1D

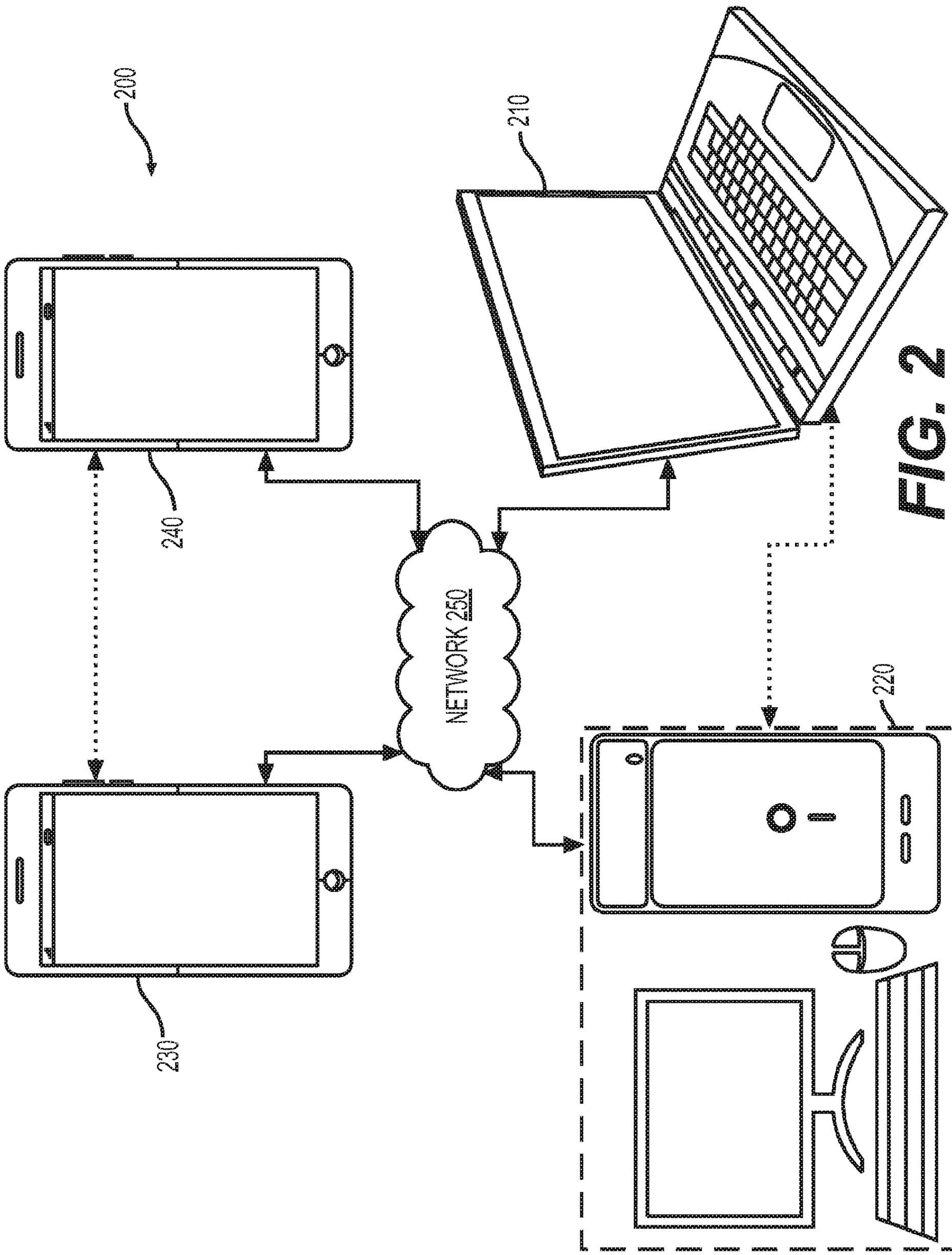


FIG. 2

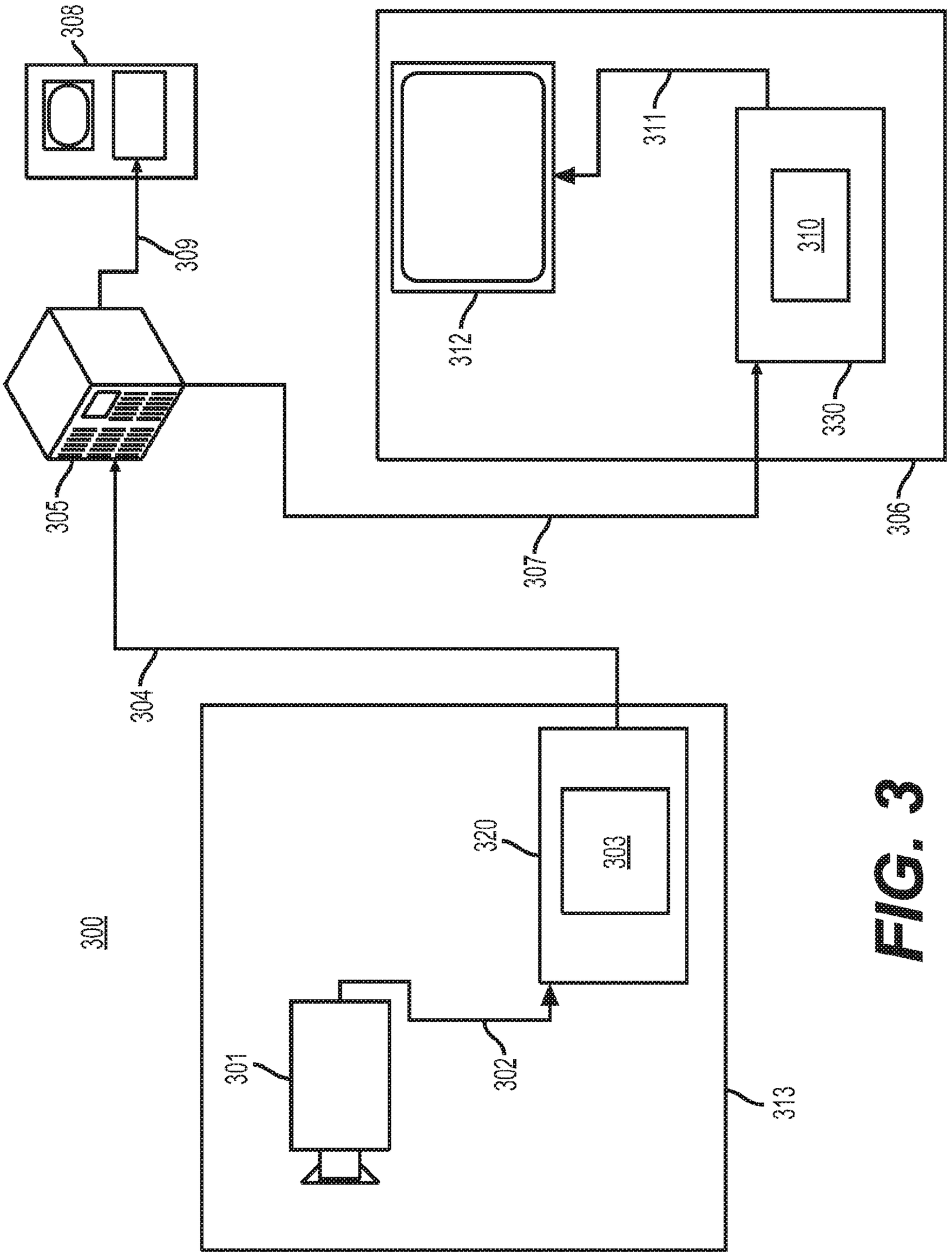


FIG. 3

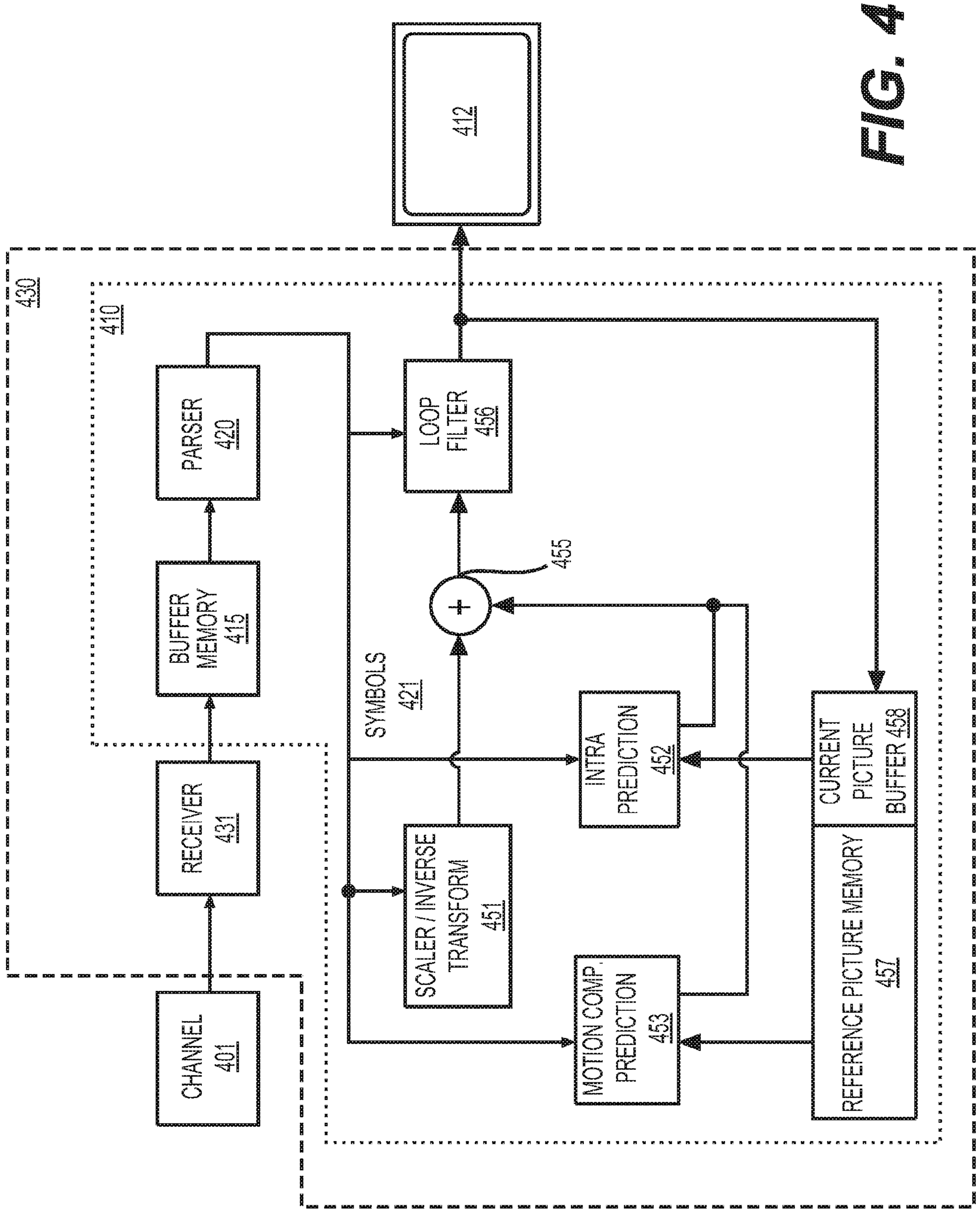

412
430
410
PARSER
420
LOOP
FILTER
456
BUFFER
MEMORY
415
RECEIVER
431
455
SYMBOLS
421
INTRA
PREDICTION
452
SCALER / INVERSE
TRANSFORM
451
CURRENT
PICTURE
BUFFER 458
MOTION COMP.
PREDICTION
453
REFERENCE PICTURE MEMORY
457
CHANNEL
401

FIG. 4

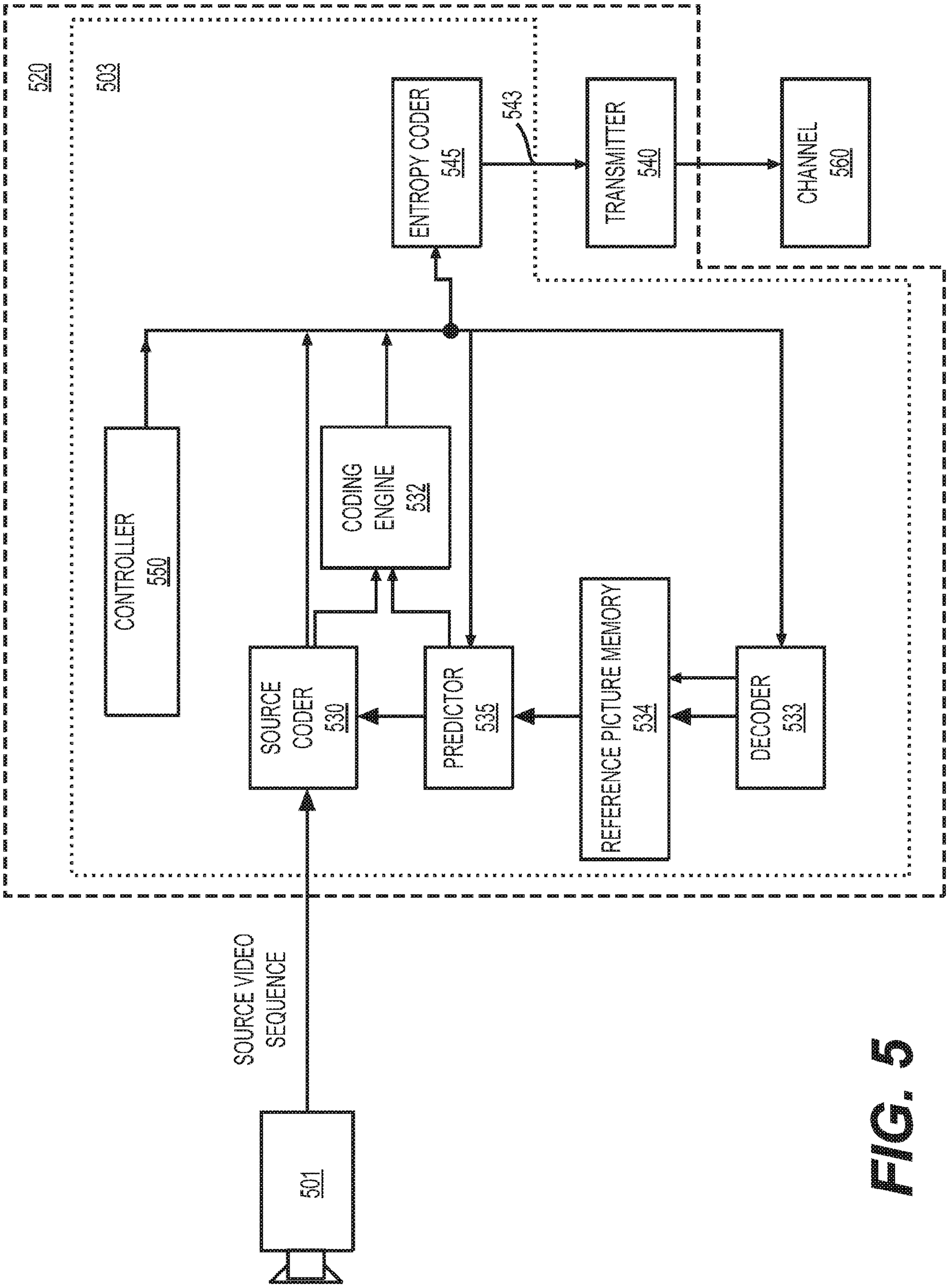

520
503
ENTROPY CODER
545
543
TRANSMITTER
540
CHANNEL
560
CONTROLLER
550
CODING
ENGINE
532
SOURCE
CODER
530
PREDICTOR
535
REFERENCE PICTURE MEMORY
534
DECODER
533
SOURCE VIDEO
SEQUENCE
501

FIG. 5

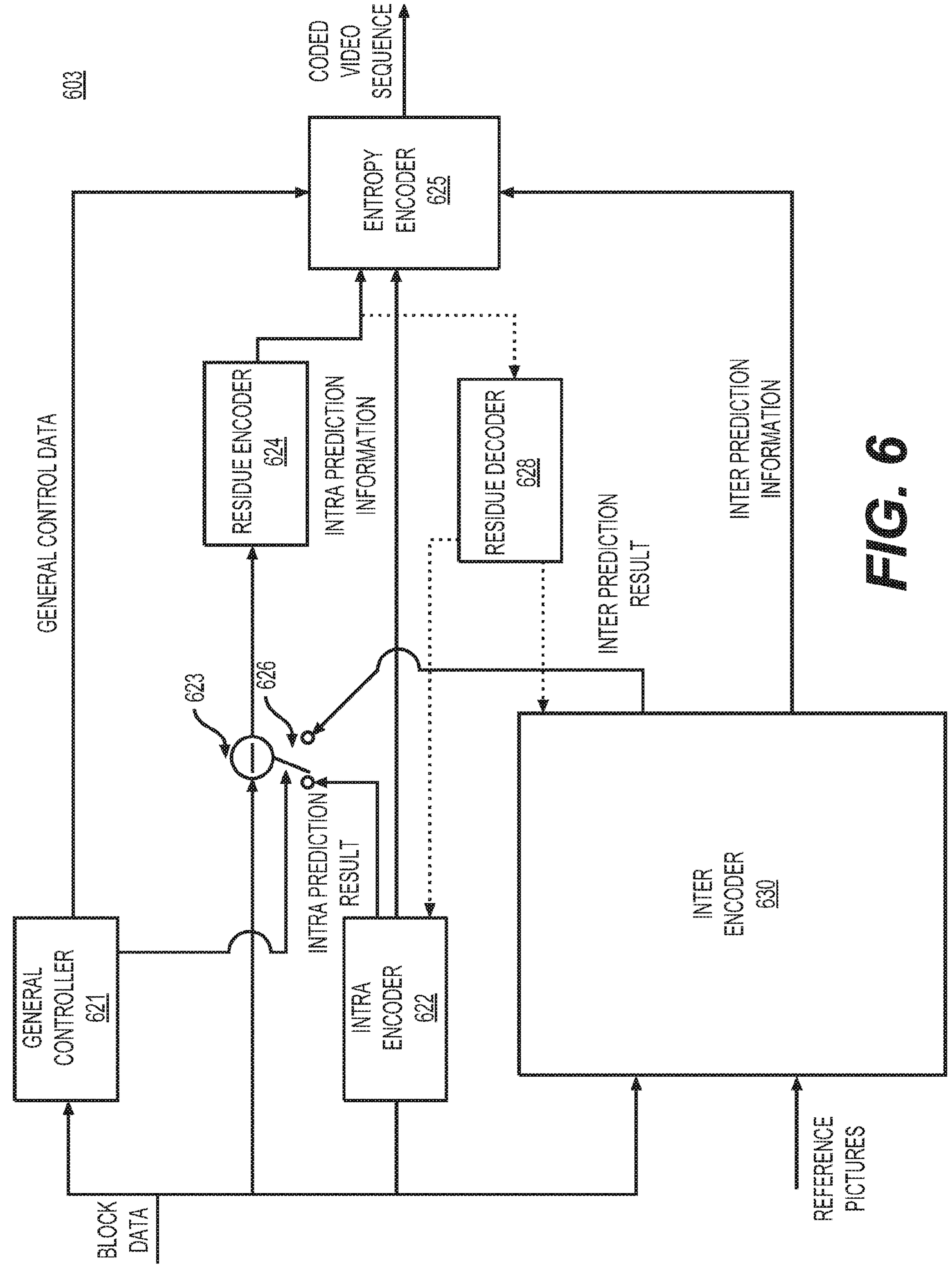

603
CODED VIDEO SEQUENCE
ENTROPY ENCODER 625
RESIDUE ENCODER 624
RESIDUE DECODER 628
GENERAL CONTROL DATA
INTRA PREDICTION INFORMATION
INTER PREDICTION RESULT
INTER PREDICTION INFORMATION
623
626
INTRA PREDICTION RESULT
GENERAL CONTROLLER 621
INTRA ENCODER 622
INTER ENCODER 630
BLOCK DATA
REFERENCE PICTURES

FIG. 6

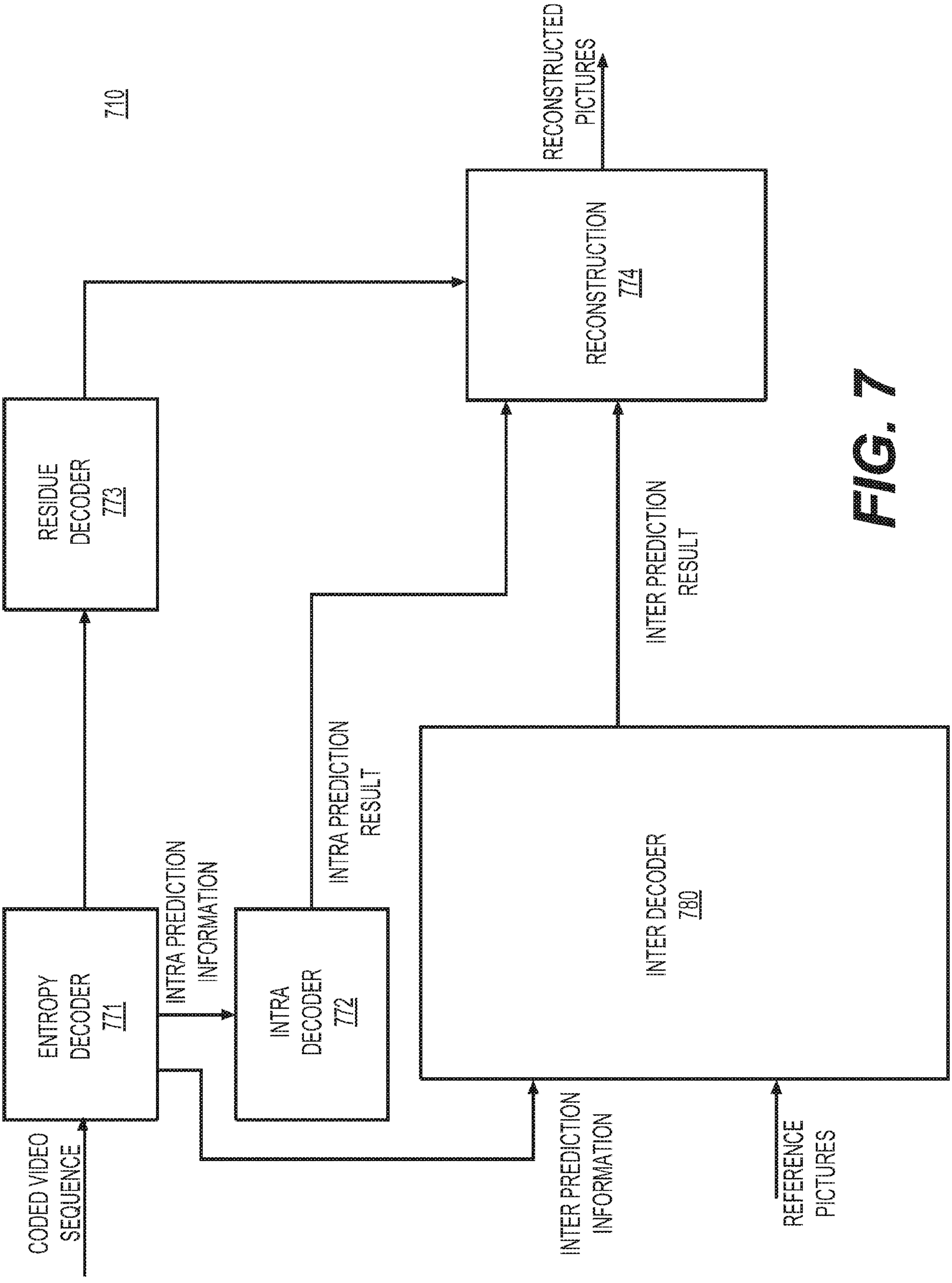

710
RECONSTRUCTED PICTURES
RECONSTRUCTION 774
RESIDUE DECODER 773
INTER PREDICTION RESULT
ENTROPY DECODER 771
INTRA PREDICTION INFORMATION
INTRA DECODER 772
INTRA PREDICTION RESULT
INTER DECODER 780
CODED VIDEO SEQUENCE
INTER PREDICTION INFORMATION
REFERENCE PICTURES

FIG. 7

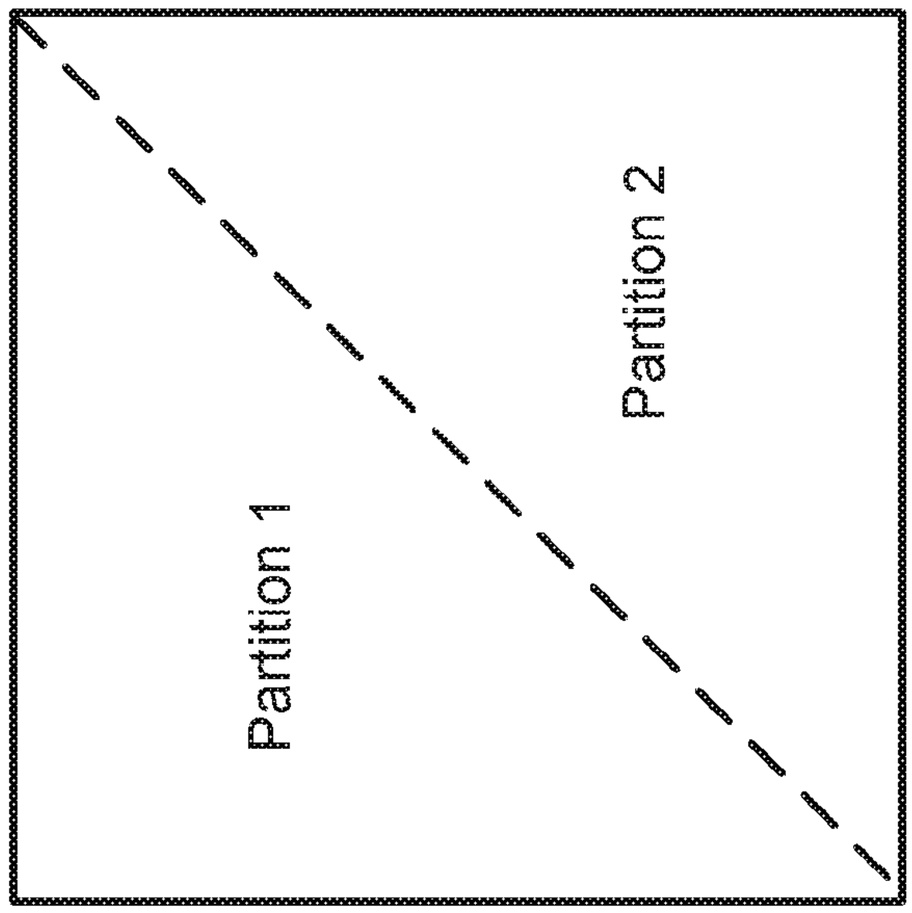


FIG. 8B

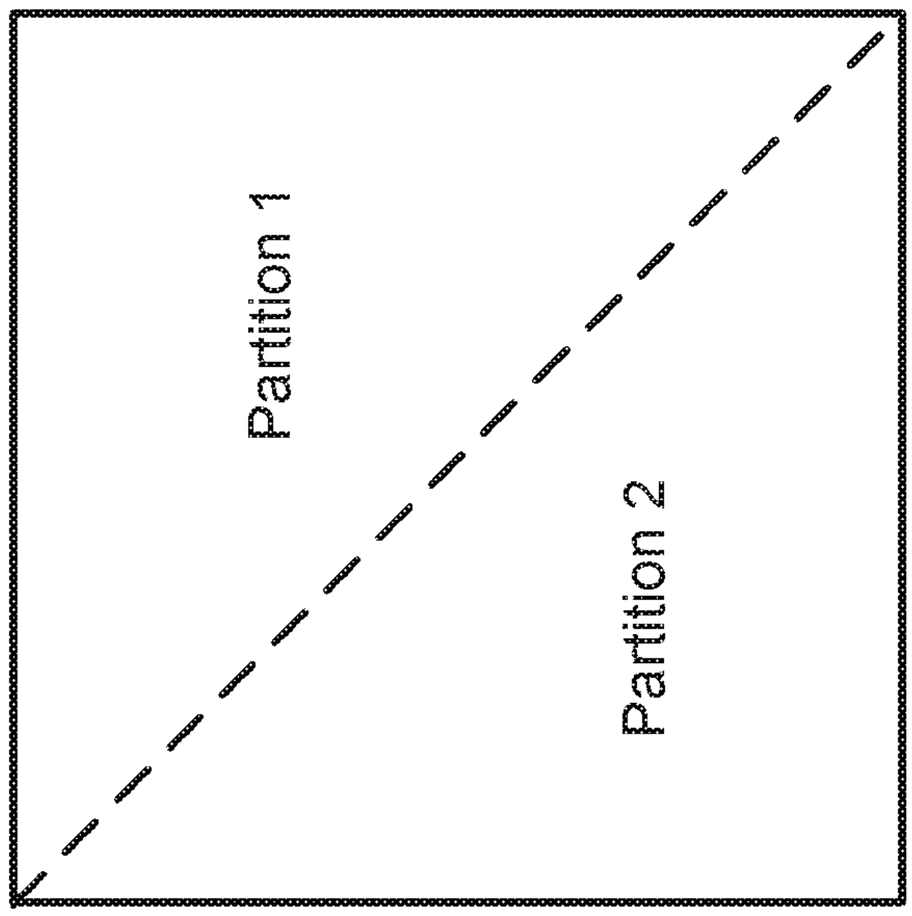


FIG. 8A

Merge Index	L0 MV	L1 MV
0	X	
1		X
2	X	
3		X
4	X	

FIG. 9

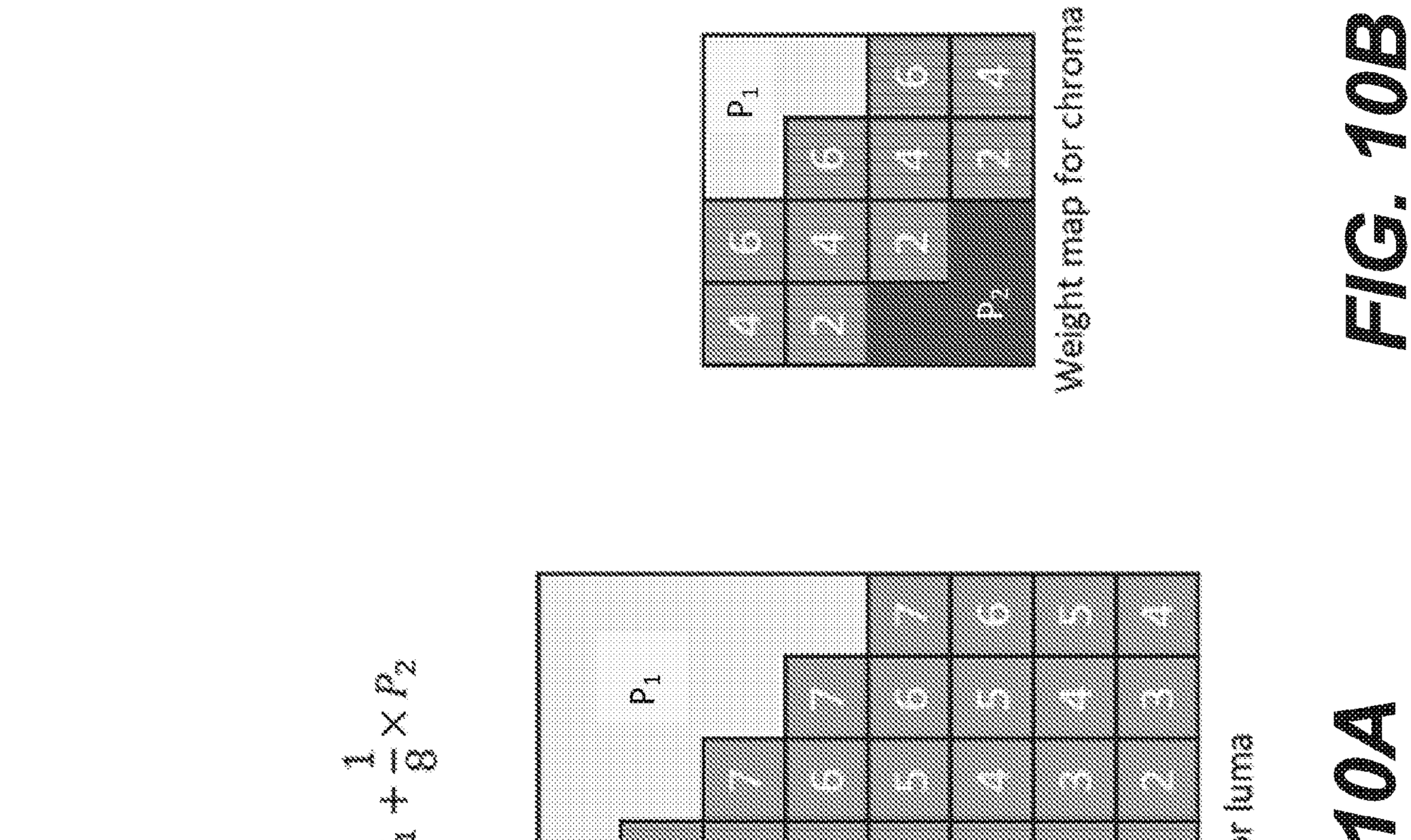


FIG. 10A

FIG. 10B

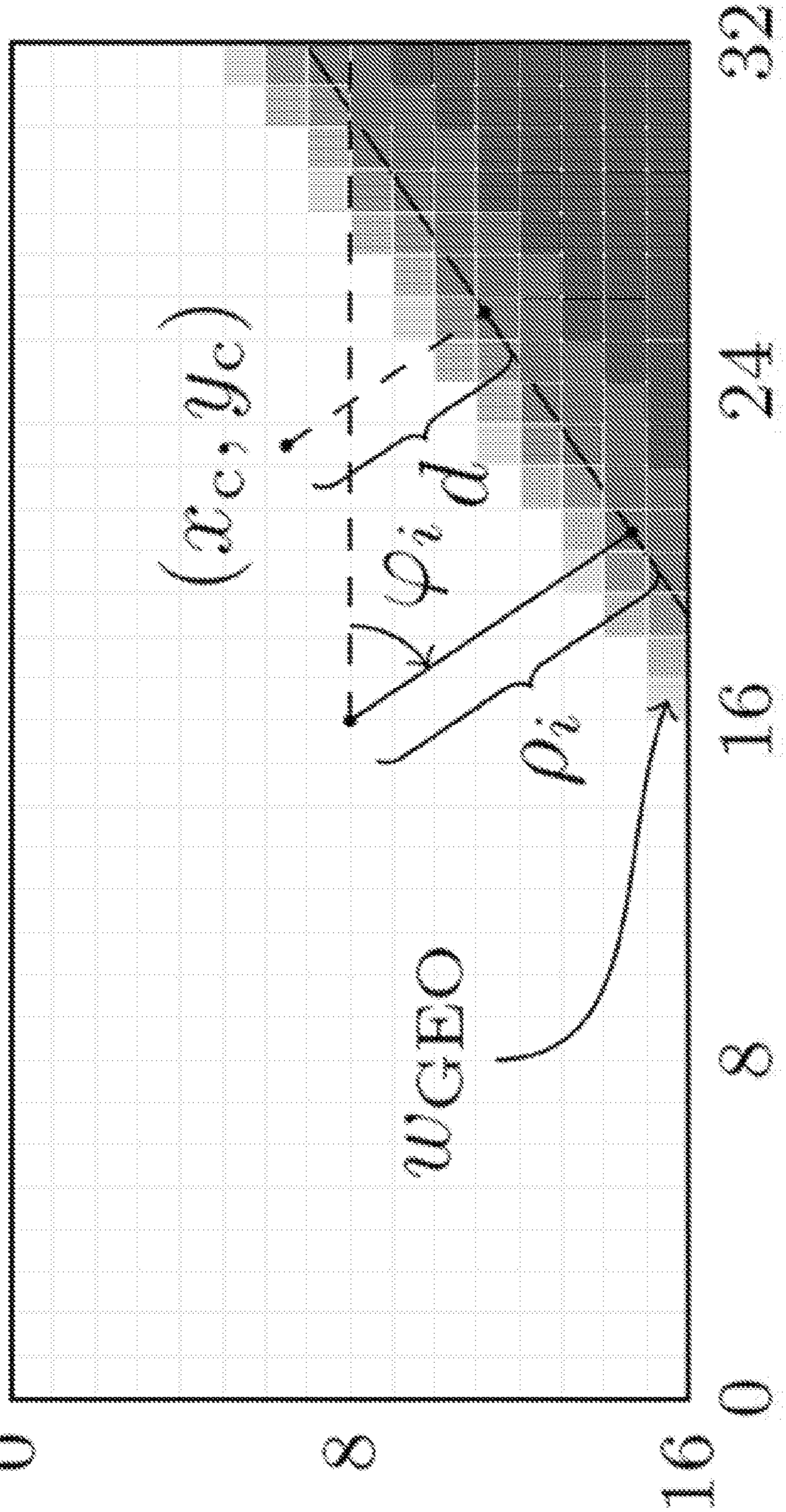


FIG. 11

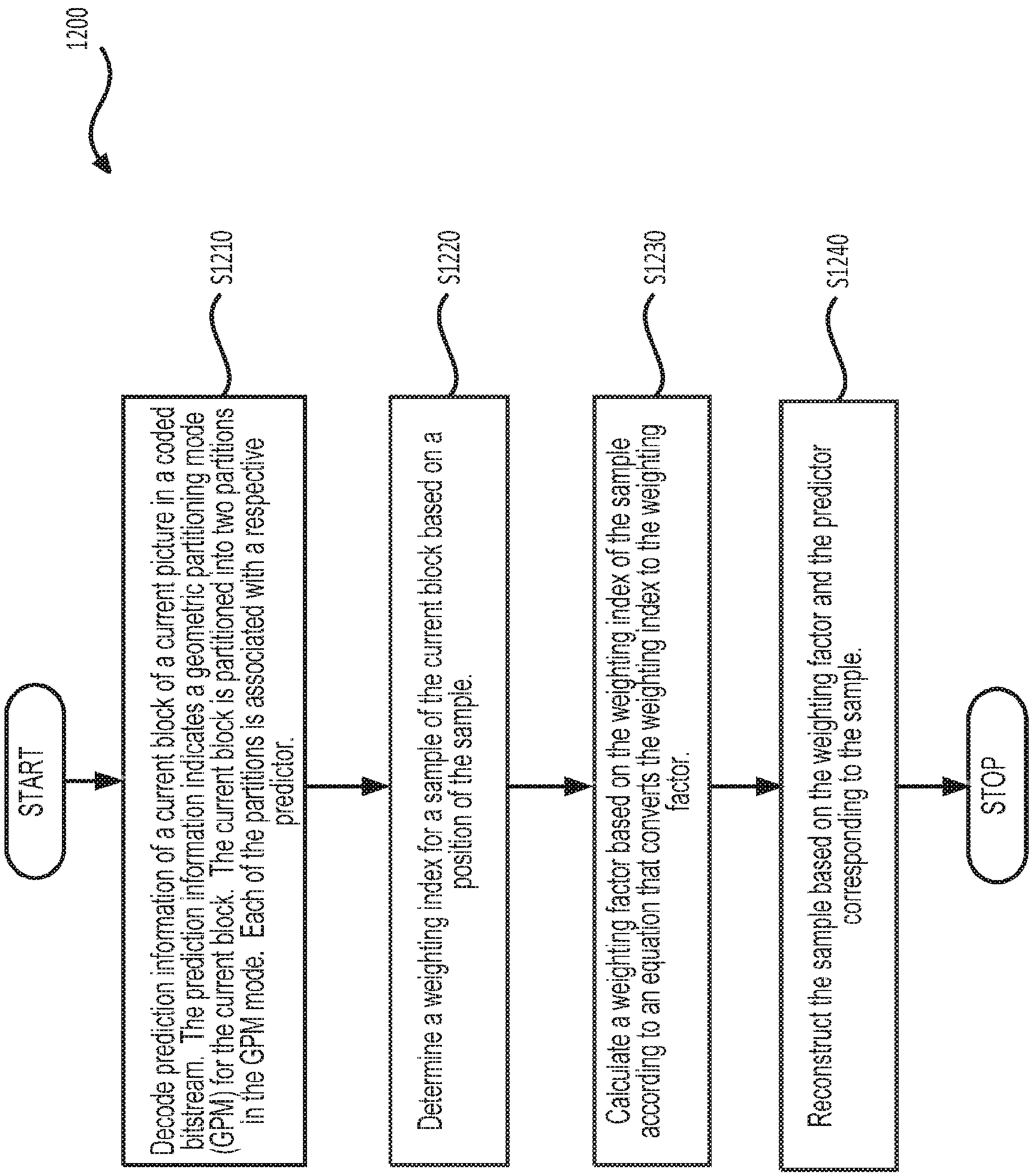

1200
START
Decode prediction information of a current block of a current picture in a coded bitstream. The prediction information indicates a geometric partitioning mode (GPM) for the current block. The current block is partitioned into two partitions in the GPM mode. Each of the partitions is associated with a respective predictor.
S1210
Determine a weighting index for a sample of the current block based on a position of the sample.
S1220
Calculate a weighting factor based on the weighting index of the sample according to an equation that converts the weighting index to the weighting factor.
S1230
Reconstruct the sample based on the weighting factor and the predictor corresponding to the sample.
S1240
STOP

FIG. 12

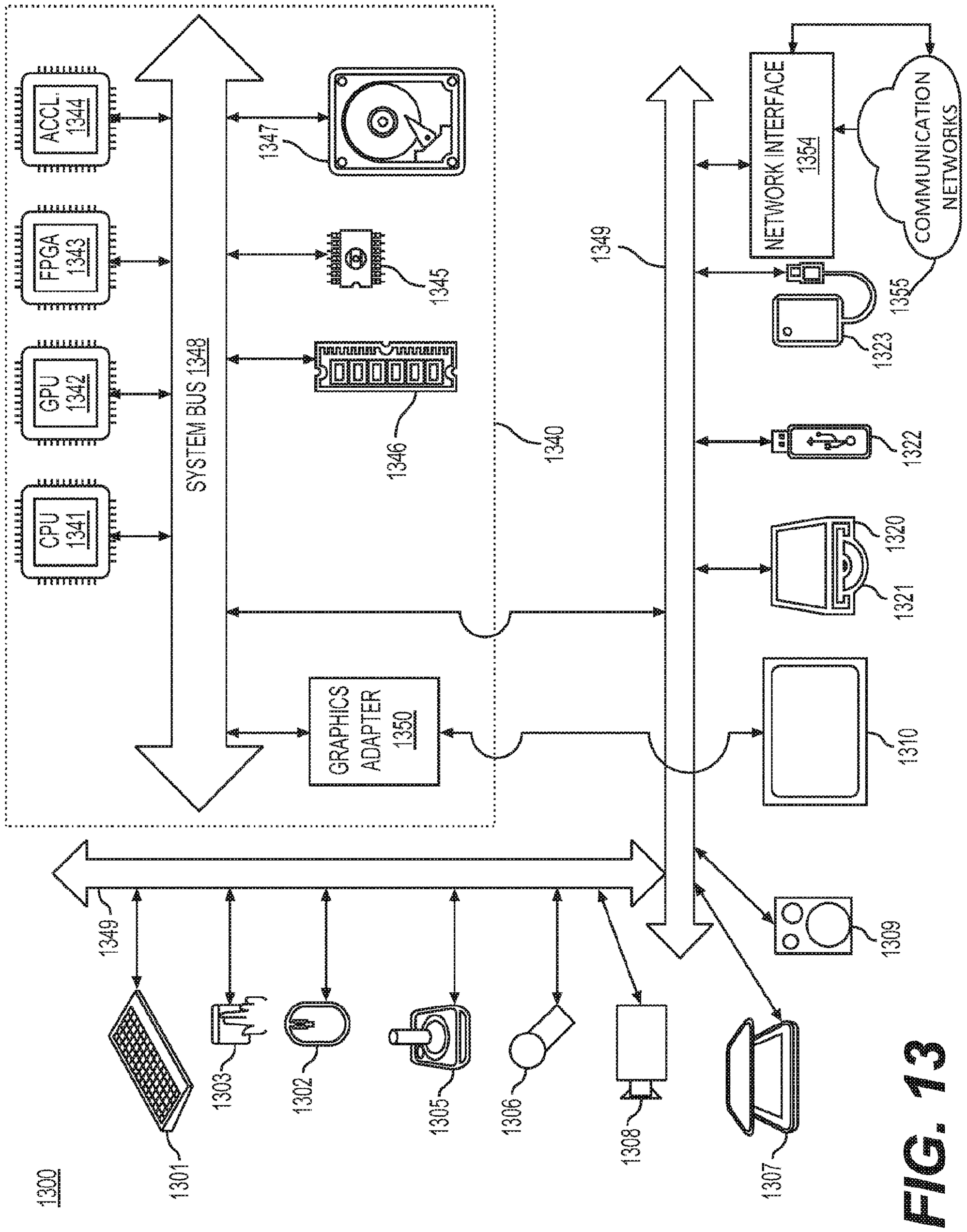

1300
1301
1302
1303
1305
1306
1307
1308
1309
1310
1320
1321
1322
1323
1340
1345
1346
1347
1348
1349
1354
1355
CPU 1341
GPU 1342
FPGA 1343
ACCL. 1344
GRAPHICS ADAPTER 1350
NETWORK INTERFACE 1354
COMMUNICATION NETWORKS
SYSTEM BUS 1348

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2020/054524

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - H04N 19/70; H04N 19/176; H04N 19/159; H04N 19/105; H04N 19/126 (2020.01)
CPC - H04N 19/70; H04N 19/176; H04N 19/159; H04N 19/105; H04N 19/126 (2020.08)

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

see Search History document

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

see Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014/0044162 A1 (QUALCOMM INCORPORATED) 13 February 2014 (13.02.2014) entire document	1-20
A	WO 2019/001733 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 03 January 2019 (03.01.2019) entire document	1-20
A	US 2017/0280156 A1 (THOMSON LICENSING) 28 September 2017 (28.09.2017) entire document	1-20
A	US 2019/0098295 A1 (GE VIDEO COMPRESSION, LLC) 28 March 2019 (28.03.2019) entire document	1-20
A	US 2012/0106627 A1 (GUO et al) 03 May 2012 (03.05.2012) entire document	1-20
A	US 2015/0010243 A1 (HUMAX HOLDINGS CO., LTD.) 08 January 2015 (08.01.2015) entire document	1-20
A	MULLER et al., "3D High-Efficiency Video Coding for Multi-View Video and Depth Data", IEEE, September 2013, retrieved on [13.12.2020]. Retrieved from the internet <URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6519266> entire document	1-20
A	US 2018/0288410 A1 (SAMSUNG ELECTRONICS CO., LTD.) 04 October 2018 (04.10.2018) 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
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 December 2020	31 DEC 2020
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 Telephone No. PCT Helpdesk: 571-272-4300