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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
09 March 2023 (09.03.2023)

(10) International Publication Number
WO 2023/034629 A1

(51) International Patent Classification:
H04N 19/157 (2014.01) *H04N 19/70* (2014.01)
H04N 19/11 (2014.01) *H04N 19/132* (2014.01)
H04N 19/593 (2014.01) *H04N 19/44* (2014.01)

(21) International Application Number: PCT/US2022/042590

(22) International Filing Date: 05 September 2022 (05.09.2022)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/240,991 05 September 2021 (05.09.2021) US

(71) Applicant: BEIJING DAJIA INTERNET INFORMATION TECHNOLOGY CO., LTD. [CN/CN]; Room 101D1-7, 1st Floor, Building 1, No. 6, Shangdi West Road, Haidian District, Beijing, Beijing 100085 (CN).

(72) Inventor; and
(71) Applicant *(for US only)*: JHU, Hong-Jheng [CN/US]; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US).

(72) Inventors: CHEN, Yi-Wen; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US). XIU, Xiaoyu; 6691 Aliso Valley Way, San Diego, California 92130 (US). WANG, Xianglin; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US). YU, Bing; Room 101D1-7, 1st Floor, Building 1, No. 6, Shangdi West Road, Haidian District, Beijing, Beijing 100085 (CN).

(74) Agent: HAO TAN; ARCH & LAKE LLP, 203 N. LaSalle Street, Suite 2100, Chicago, Illinois 60601 (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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, 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,

(54) **Title:** INTRA PREDICTION MODES SIGNALING

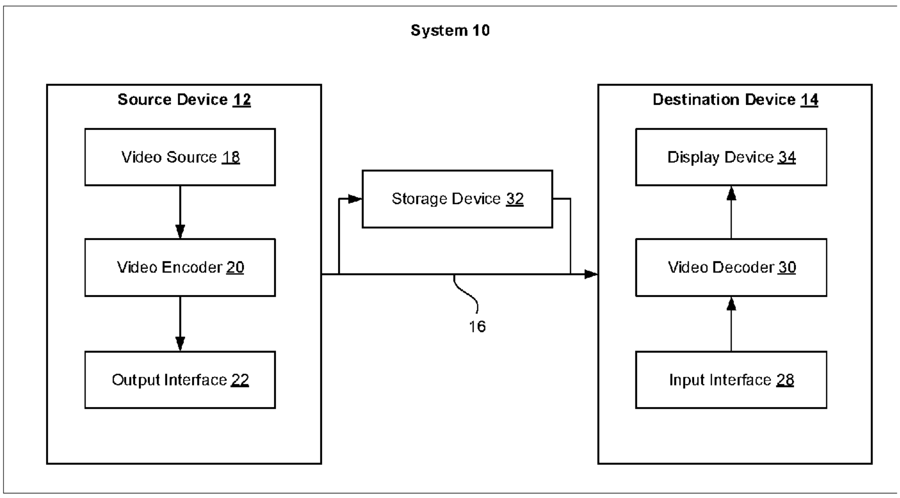

FIG. 1

(57) **Abstract:** Methods, apparatuses, and non-transitory computer-readable storage mediums are provided for video decoding. In one method, a decoder constructs a plurality of intra prediction mode groups (IPMGs), and each IPMG of the plurality of IPMGs comprises a plurality of intra prediction modes; in response to determining that a current IPMG is indicated to generate an intra prediction mode for a current block, the decoder may derive a scheme of intra prediction modes to determine a best mode within the current IPMG to generate intra prediction samples for the current block, and the scheme of intra prediction modes comprises decoder-side intra mode derivation (DIMD) and template-based intra mode derivation (TIMD).

WO 2023/034629 A1

[Continued on next page]

RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

INTRA PREDICTION MODES SIGNALING

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application is based upon and claims priority to Provisional Applications No. 63/240,991, filed on September 5, 2021, the entire content of which is incorporated herein by reference for all purposes.

TECHNICAL FIELD

[0002] This disclosure is related to video coding and compression. More specifically, this disclosure relates to methods and apparatus on improving the coding efficiency or simplifying the intra prediction modes signaling.

BACKGROUND

[0003] Digital video is supported by a variety of electronic devices, such as digital televisions, laptop or desktop computers, tablet computers, digital cameras, digital recording devices, digital media players, video gaming consoles, smart phones, video teleconferencing devices, video streaming devices, etc. The electronic devices transmit and receive or otherwise communicate digital video data across a communication network, and/or store the digital video data on a storage device. Due to a limited bandwidth capacity of the communication network and limited memory resources of the storage device, video coding may be used to compress the video data according to one or more video coding standards before it is communicated or stored. For example, video coding standards include Versatile Video Coding (VVC), Joint Exploration test Model (JEM), High-Efficiency Video Coding (HEVC/H.265), Advanced Video Coding (AVC/H.264), Moving Picture Expert Group (MPEG) coding, or the like. Video coding generally utilizes prediction methods (e.g., inter-prediction, intra-prediction, or the like) that take advantage of redundancy inherent in the video data. Video coding aims to compress video data into a form that uses a lower bit rate, while avoiding or minimizing degradations to video quality.

SUMMARY

[0004] Examples of the present disclosure provide methods and apparatus for video coding using intra prediction.

[0005] According to a first aspect of the present disclosure, a method for video decoding is provided. The method may include: constructing, by a decoder, a plurality of intra prediction mode groups (IPMGs), each IPMG of the plurality of IPMGs comprises a plurality of intra prediction modes; and in response to determining that a current IPMG is indicated to generate an intra prediction mode for a current block, deriving, by the decoder, a scheme of intra prediction modes to determine a best mode within the current IPMG to generate intra prediction samples for the current block, and the scheme of intra prediction modes comprises decoder-side intra mode derivation (DIMD) and template-based intra mode derivation (TIMD).

[0006] According to a second aspect of the present disclosure, an apparatus for video decoding is provided. The apparatus may include: one or more processors; and a memory configured to store instructions executable by the one or more processors; the one or more processors, upon execution of the instructions, are configured to perform the method as disclosed in the first aspect.

[0007] According to a third aspect of the present disclosure, a non-transitory computer-readable storage medium for video decoding is provided. The non-transitory computer-readable storage medium storing computer-executable instructions that, when executed by one or more computer processors, cause the one or more computer processors to perform the method as disclosed in the first aspect.

[0008] It is to be understood that the above general descriptions and detailed descriptions below are only exemplary and explanatory and not intended to limit the present disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate examples consistent with the present disclosure and, together with the description, serve to explain the principles of the disclosure.

[0010] FIG. 1 is a block diagram illustrating an exemplary system for encoding and decoding video blocks in accordance with some implementations of the present disclosure.

[0011] FIG. 2 is a block diagram illustrating an exemplary video encoder in accordance with some implementations of the present disclosure.

[0012] FIG. 3 is a block diagram illustrating an exemplary video decoder in accordance with some implementations of the present disclosure.

[0013] FIGS. 4A through 4E are block diagrams illustrating how a frame is recursively partitioned into multiple video blocks of different sizes and shapes in accordance with some implementations of the present disclosure.

[0014] FIG. 5A illustrates a definition of samples used by PDPC applied to a prediction mode in accordance with some implementations of the present disclosure.

[0015] FIG. 5B illustrates a definition of samples used by PDPC applied to a prediction mode in accordance with some implementations of the present disclosure.

[0016] FIG. 5C illustrates a definition of samples used by PDPC applied to a prediction mode in accordance with some implementations of the present disclosure.

[0017] FIG. 5D illustrates a definition of samples used by PDPC applied to a prediction mode in accordance with some implementations of the present disclosure.

[0018] FIG. 6 illustrates examples of the intra modes in the VVC in accordance with some implementations of the present disclosure.

[0019] FIG. 7 illustrates an example of chosen pixels on which a gradient analysis is performed in accordance with some implementations of the present disclosure.

[0020] FIG. 8 illustrates a convolution process in accordance with some implementations of the present disclosure.

[0021] FIG. 9 illustrates prediction fusion by weighted averaging of two HoG mode and one planar mode in accordance with some implementations of the present disclosure.

[0022] FIG. 10 illustrates template and its reference samples used om TIMD in accordance with some implementations of the present disclosure.

[0023] FIG. 11 is a block diagram showing a video decoding process in accordance with some implementations of the present disclosure.

[0024] FIG. 12 is a block diagram illustrating a computing environment coupled with a user interface in accordance with some implementations of the present disclosure.

DETAILED DESCRIPTION

[0025] Reference will now be made in detail to example embodiments, examples of which are illustrated in the accompanying drawings. The following description refers to the accompanying drawings in which the same numbers in different drawings represent the same or similar elements unless otherwise represented. The implementations set forth in the following description of example embodiments do not represent all implementations consistent with the disclosure. Instead, they are merely examples of apparatuses and methods consistent with aspects related to the disclosure as recited in the appended claims.

[0026] The terminology used in the present disclosure is for the purpose of describing particular embodiments only and is not intended to limit the present disclosure. As used in the present disclosure and the appended claims, the singular forms “a,” “an,” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It shall also be understood that the term “and/or” used herein is intended to signify and include any or all possible combinations of one or more of the associated listed items.

[0027] It shall be understood that, although the terms “first,” “second,” “third,” etc., may be used herein to describe various information, the information should not be limited by these terms. These terms are only used to distinguish one category of information from another. For example, without departing from the scope of the present disclosure, first information may be termed as second information; and similarly, second information may also be termed as first information. As used herein, the term “if” may be understood to mean “when” or “upon” or “in response to a judgment” depending on the context.

[0028] Various video coding techniques may be used to compress video data. Video coding is performed according to one or more video coding standards. For example, nowadays, some well-known video coding standards include Versatile Video Coding (VVC), High Efficiency Video Coding (HEVC, also known as H.265 or MPEG-H Part2) and Advanced Video Coding (AVC, also known as H.264 or MPEG-4 Part 10), which are jointly developed by ISO/IEC MPEG and ITU-T VECG. AOMedia Video 1 (AV1) was developed by Alliance for Open Media (AOM) as a successor to its preceding standard VP9. Audio Video Coding (AVS), which refers to digital audio and digital video compression standard, is another video compression standard series developed by the Audio and Video Coding Standard Workgroup of China. Most of the existing video coding standards are built upon the famous hybrid video coding framework i.e., using block-based prediction methods (e.g., inter-prediction, intra-prediction) to reduce redundancy present in video images or sequences and using transform coding to compact the energy of the prediction errors. An important goal of video coding techniques is to compress video data into a form that uses a lower bit rate while avoiding or minimizing degradations to video quality.

[0029] The first generation AVS standard includes Chinese national standard “Information Technology, Advanced Audio Video Coding, Part 2: Video” (known as AVS1) and “Information Technology, Advanced Audio Video Coding Part 16: Radio Television Video” (known as AVS+). It can offer around 50% bit-rate saving at the same perceptual quality compared to

MPEG-2 standard. The AVS1 standard video part was promulgated as the Chinese national standard in February 2006. The second generation AVS standard includes the series of Chinese national standard "Information Technology, Efficient Multimedia Coding" (knows as AVS2), which is mainly targeted at the transmission of extra HD TV programs. The coding efficiency of the AVS2 is double of that of the AVS+. On May 2016, the AVS2 was issued as the Chinese national standard. Meanwhile, the AVS2 standard video part was submitted by Institute of Electrical and Electronics Engineers (IEEE) as one international standard for applications. The AVS3 standard is one new generation video coding standard for UHD video application aiming at surpassing the coding efficiency of the latest international standard HEVC. In March 2019, at the 68-th AVS meeting, the AVS3-P2 baseline was finished, which provides approximately 30% bit-rate savings over the HEVC standard. Currently, there is one reference software, called high performance model (HPM), is maintained by the AVS group to demonstrate a reference implementation of the AVS3 standard.

[0030] FIG. 1 is a block diagram illustrating an exemplary system 10 for encoding and decoding video blocks in parallel in accordance with some implementations of the present disclosure. As shown in FIG. 1, the system 10 includes a source device 12 that generates and encodes video data to be decoded at a later time by a destination device 14. The source device 12 and the destination device 14 may comprise any of a wide variety of electronic devices, including desktop or laptop computers, tablet computers, smart phones, set-top boxes, digital televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or the like. In some implementations, the source device 12 and the destination device 14 are equipped with wireless communication capabilities.

[0031] In some implementations, the destination device 14 may receive the encoded video data to be decoded via a link 16. The link 16 may comprise any type of communication medium or device capable of moving the encoded video data from the source device 12 to the destination device 14. In one example, the link 16 may comprise a communication medium to enable the source device 12 to transmit the encoded video data directly to the destination device 14 in real time. The encoded video data may be modulated according to a communication standard, such as a wireless communication protocol, and transmitted to the destination device 14. The communication medium may comprise any wireless or wired communication medium, such as a Radio Frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from the source device 12 to the destination device 14.

[0032] In some other implementations, the encoded video data may be transmitted from an output interface 22 to a storage device 32. Subsequently, the encoded video data in the storage device 32 may be accessed by the destination device 14 via an input interface 28. The storage device 32 may include any of a variety of distributed or locally accessed data storage media such as a hard drive, Blu-ray discs, Digital Versatile Disks (DVDs), Compact Disc Read-Only Memories (CD-ROMs), flash memory, volatile or non-volatile memory, or any other suitable digital storage media for storing the encoded video data. In a further example, the storage device 32 may correspond to a file server or another intermediate storage device that may hold the encoded video data generated by the source device 12. The destination device 14 may access the stored video data from the storage device 32 via streaming or downloading. The file server may be any type of computer capable of storing the encoded video data and transmitting the encoded video data to the destination device 14. Exemplary file servers include a web server (e.g., for a website), a File Transfer Protocol (FTP) server, Network Attached Storage (NAS) devices, or a local disk drive. The destination device 14 may access the encoded video data through any standard data connection, including a wireless channel (e.g., a Wireless Fidelity (Wi-Fi) connection), a wired connection (e.g., Digital Subscriber Line (DSL), cable modem, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server. The transmission of the encoded video data from the storage device 32 may be a streaming transmission, a download transmission, or a combination of both.

[0033] As shown in FIG. 1, the source device 12 includes a video source 18, a video encoder 20 and the output interface 22. The video source 18 may include a source such as a video capturing device, e.g., a video camera, a video archive containing previously captured video, a video feeding interface to receive video from a video content provider, and/or a computer graphics system for generating computer graphics data as the source video, or a combination of such sources. As one example, if the video source 18 is a video camera of a security surveillance system, the source device 12 and the destination device 14 may form camera phones or video phones. However, the implementations described in the present application may be applicable to video coding in general, and may be applied to wireless and/or wired applications.

[0034] The captured, pre-captured, or computer-generated video may be encoded by the video encoder 20. The encoded video data may be transmitted directly to the destination device

14 via the output interface 22 of the source device 12. The encoded video data may also (or alternatively) be stored onto the storage device 32 for later access by the destination device 14 or other devices, for decoding and/or playback. The output interface 22 may further include a modem and/or a transmitter.

[0035] The destination device 14 includes the input interface 28, a video decoder 30, and a display device 34. The input interface 28 may include a receiver and/or a modem and receive the encoded video data over the link 16. The encoded video data communicated over the link 16, or provided on the storage device 32, may include a variety of syntax elements generated by the video encoder 20 for use by the video decoder 30 in decoding the video data. Such syntax elements may be included within the encoded video data transmitted on a communication medium, stored on a storage medium, or stored on a file server.

[0036] In some implementations, the destination device 14 may include the display device 34, which may be an integrated display device and an external display device that is configured to communicate with the destination device 14. The display device 34 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a Liquid Crystal Display (LCD), a plasma display, an Organic Light Emitting Diode (OLED) display, or another type of display device.

[0037] The video encoder 20 and the video decoder 30 may operate according to proprietary or industry standards, such as VVC, HEVC, MPEG-4, Part 10, AVC, or extensions of such standards. It should be understood that the present application is not limited to a specific video encoding/decoding standard and may be applicable to other video encoding/decoding standards. It is generally contemplated that the video encoder 20 of the source device 12 may be configured to encode video data according to any of these current or future standards. Similarly, it is also generally contemplated that the video decoder 30 of the destination device 14 may be configured to decode video data according to any of these current or future standards.

[0038] The video encoder 20 and the video decoder 30 each may be implemented as any of a variety of suitable encoder and/or decoder circuitry, such as one or more microprocessors, Digital Signal Processors (DSPs), Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When implemented partially in software, an electronic device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the video encoding/decoding operations disclosed in the present disclosure. Each of the video encoder

20 and the video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device.

[0039] FIG. 2 is a block diagram illustrating an exemplary video encoder 20 in accordance with some implementations described in the present application. The video encoder 20 may perform intra and inter predictive coding of video blocks within video frames. Intra predictive coding relies on spatial prediction to reduce or remove spatial redundancy in video data within a given video frame or picture. Inter predictive coding relies on temporal prediction to reduce or remove temporal redundancy in video data within adjacent video frames or pictures of a video sequence. It should be noted that the term “frame” may be used as synonyms for the term “image” or “picture” in the field of video coding.

[0040] As shown in FIG. 2, the video encoder 20 includes a video data memory 40, a prediction processing unit 41, a Decoded Picture Buffer (DPB) 64, a summer 50, a transform processing unit 52, a quantization unit 54, and an entropy encoding unit 56. The prediction processing unit 41 further includes a motion estimation unit 42, a motion compensation unit 44, a partition unit 45, an intra prediction processing unit 46, and an intra Block Copy (BC) unit 48. In some implementations, the video encoder 20 also includes an inverse quantization unit 58, an inverse transform processing unit 60, and a summer 62 for video block reconstruction. An in-loop filter 63, such as a deblocking filter, may be positioned between the summer 62 and the DPB 64 to filter block boundaries to remove blockiness artifacts from reconstructed video. Another in-loop filter, such as Sample Adaptive Offset (SAO) filter and/or Adaptive in-Loop Filter (ALF), may also be used in addition to the deblocking filter to filter an output of the summer 62. In some examples, the in-loop filters may be omitted, and the decoded video block may be directly provided by the summer 62 to the DPB 64. The video encoder 20 may take the form of a fixed or programmable hardware unit or may be divided among one or more of the illustrated fixed or programmable hardware units.

[0041] The video data memory 40 may store video data to be encoded by the components of the video encoder 20. The video data in the video data memory 40 may be obtained, for example, from the video source 18 as shown in FIG. 1. The DPB 64 is a buffer that stores reference video data (for example, reference frames or pictures) for use in encoding video data by the video encoder 20 (e.g., in intra or inter predictive coding modes). The video data memory 40 and the DPB 64 may be formed by any of a variety of memory devices. In various examples, the video data memory 40 may be on-chip with other components of the video encoder 20, or off-chip relative to those components.

[0042] As shown in FIG. 2, after receiving the video data, the partition unit 45 within the prediction processing unit 41 partitions the video data into video blocks. This partitioning may also include partitioning a video frame into slices, tiles (for example, sets of video blocks), or other larger Coding Units (CUs) according to predefined splitting structures such as a Quad-Tree (QT) structure associated with the video data. The video frame is or may be regarded as a two-dimensional array or matrix of samples with sample values. A sample in the array may also be referred to as a pixel or a pel. A number of samples in horizontal and vertical directions (or axes) of the array or picture define a size and/or a resolution of the video frame. The video frame may be divided into multiple video blocks by, for example, using QT partitioning. The video block again is or may be regarded as a two-dimensional array or matrix of samples with sample values, although of smaller dimension than the video frame. A number of samples in horizontal and vertical directions (or axes) of the video block define a size of the video block. The video block may further be partitioned into one or more block partitions or sub-blocks (which may form again blocks) by, for example, iteratively using QT partitioning, Binary-Tree (BT) partitioning or Triple-Tree (TT) partitioning or any combination thereof. It should be noted that the term “block” or “video block” as used herein may be a portion, in particular a rectangular (square or non- square) portion, of a frame or a picture. With reference, for example, to HEVC and VVC, the block or video block may be or correspond to a Coding Tree Unit (CTU), a CU, a Prediction Unit (PU) or a Transform Unit (TU) and/or may be or correspond to a corresponding block, e.g. a Coding Tree Block (CTB), a Coding Block (CB), a Prediction Block (PB) or a Transform Block (TB) and/or to a sub-block.

[0043] The prediction processing unit 41 may select one of a plurality of possible predictive coding modes, such as one of a plurality of intra predictive coding modes or one of a plurality of inter predictive coding modes, for the current video block based on error results (e.g., coding rate and the level of distortion). The prediction processing unit 41 may provide the resulting intra or inter prediction coded block to the summer 50 to generate a residual block and to the summer 62 to reconstruct the encoded block for use as part of a reference frame subsequently. The prediction processing unit 41 also provides syntax elements, such as motion vectors, intra-mode indicators, partition information, and other such syntax information, to the entropy encoding unit 56.

[0044] In order to select an appropriate intra predictive coding mode for the current video block, the intra prediction processing unit 46 within the prediction processing unit 41 may perform intra predictive coding of the current video block relative to one or more neighbor blocks in the same frame as the current block to be coded to provide spatial prediction. The motion estimation unit 42 and the motion compensation unit 44 within the prediction processing unit 41 perform inter predictive coding of the current video block relative to one or more predictive blocks in one or more reference frames to provide temporal prediction. The video encoder 20 may perform multiple coding passes, e.g., to select an appropriate coding mode for each block of video data.

[0045] In some implementations, the motion estimation unit 42 determines the inter prediction mode for a current video frame by generating a motion vector, which indicates the displacement of a video block within the current video frame relative to a predictive block within a reference video frame, according to a predetermined pattern within a sequence of video frames. Motion estimation, performed by the motion estimation unit 42, is the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a video block within a current video frame or picture relative to a predictive block within a reference frame relative to the current block being coded within the current frame. The predetermined pattern may designate video frames in the sequence as P frames or B frames. The intra BC unit 48 may determine vectors, e.g., block vectors, for intra BC coding in a manner similar to the determination of motion vectors by the motion estimation unit 42 for inter prediction, or may utilize the motion estimation unit 42 to determine the block vector.

[0046] A predictive block for the video block may be or may correspond to a block or a reference block of a reference frame that is deemed as closely matching the video block to be coded in terms of pixel difference, which may be determined by Sum of Absolute Difference (SAD), Sum of Square Difference (SSD), or other difference metrics. In some implementations, the video encoder 20 may calculate values for sub-integer pixel positions of reference frames stored in the DPB 64. For example, the video encoder 20 may interpolate values of one-quarter pixel positions, one-eighth pixel positions, or other fractional pixel positions of the reference frame. Therefore, the motion estimation unit 42 may perform a motion search relative to the full pixel positions and fractional pixel positions and output a motion vector with fractional pixel precision.

[0047] The motion estimation unit 42 calculates a motion vector for a video block in an inter prediction coded frame by comparing the position of the video block to the position of a predictive block of a reference frame selected from a first reference frame list (List 0) or a second reference frame list (List 1), each of which identifies one or more reference frames stored in the DPB 64. The motion estimation unit 42 sends the calculated motion vector to the motion compensation unit 44 and then to the entropy encoding unit 56.

[0048] Motion compensation, performed by the motion compensation unit 44, may involve fetching or generating the predictive block based on the motion vector determined by the motion estimation unit 42. Upon receiving the motion vector for the current video block, the motion compensation unit 44 may locate a predictive block to which the motion vector points in one of the reference frame lists, retrieve the predictive block from the DPB 64, and forward the predictive block to the summer 50. The summer 50 then forms a residual video block of pixel difference values by subtracting pixel values of the predictive block provided by the motion compensation unit 44 from the pixel values of the current video block being coded. The pixel difference values forming the residual video block may include luma or chroma difference components or both. The motion compensation unit 44 may also generate syntax elements associated with the video blocks of a video frame for use by the video decoder 30 in decoding the video blocks of the video frame. The syntax elements may include, for example, syntax elements defining the motion vector used to identify the predictive block, any flags indicating the prediction mode, or any other syntax information described herein. Note that the motion estimation unit 42 and the motion compensation unit 44 may be highly integrated, but are illustrated separately for conceptual purposes.

[0049] In some implementations, the intra BC unit 48 may generate vectors and fetch predictive blocks in a manner similar to that described above in connection with the motion estimation unit 42 and the motion compensation unit 44, but with the predictive blocks being in the same frame as the current block being coded and with the vectors being referred to as block vectors as opposed to motion vectors. In particular, the intra BC unit 48 may determine an intra-prediction mode to use to encode a current block. In some examples, the intra BC unit 48 may encode a current block using various intra-prediction modes, e.g., during separate encoding passes, and test their performance through rate-distortion analysis. Next, the intra BC unit 48 may select, among the various tested intra-prediction modes, an appropriate intra-prediction mode to use and generate an intra-mode indicator accordingly. For example, the intra BC unit 48 may calculate rate-distortion values using a rate-distortion analysis for the various tested intra-prediction modes, and select the intra-prediction mode having the best rate-distortion characteristics among the tested modes as the appropriate intra-prediction mode to use. Rate-distortion analysis generally determines an amount of distortion (or error) between an encoded block and an original, unencoded block that was encoded to produce the encoded block, as well as a bitrate (i.e., a number of bits) used to produce the encoded block. Intra BC unit 48 may calculate ratios from the distortions and rates for the various encoded blocks to determine which intra-prediction mode exhibits the best rate-distortion value for the block.

[0050] In other examples, the intra BC unit 48 may use the motion estimation unit 42 and the motion compensation unit 44, in whole or in part, to perform such functions for Intra BC prediction according to the implementations described herein. In either case, for Intra block copy, a predictive block may be a block that is deemed as closely matching the block to be coded, in terms of pixel difference, which may be determined by SAD, SSD, or other difference metrics, and identification of the predictive block may include calculation of values for sub-integer pixel positions.

[0051] Whether the predictive block is from the same frame according to intra prediction, or a different frame according to inter prediction, the video encoder 20 may form a residual video block by subtracting pixel values of the predictive block from the pixel values of the current video block being coded, forming pixel difference values. The pixel difference values forming the residual video block may include both luma and chroma component differences.

[0052] The intra prediction processing unit 46 may intra-predict a current video block, as an alternative to the inter-prediction performed by the motion estimation unit 42 and the motion compensation unit 44, or the intra block copy prediction performed by the intra BC unit 48, as described above. In particular, the intra prediction processing unit 46 may determine an intra prediction mode to use to encode a current block. To do so, the intra prediction processing unit 46 may encode a current block using various intra prediction modes, e.g., during separate encoding passes, and the intra prediction processing unit 46 (or a mode selection unit, in some examples) may select an appropriate intra prediction mode to use from the tested intra prediction modes. The intra prediction processing unit 46 may provide information indicative of the selected intra-prediction mode for the block to the entropy encoding unit 56. The entropy encoding unit 56 may encode the information indicating the selected intra-prediction mode in the bitstream.

[0053] After the prediction processing unit 41 determines the predictive block for the current video block via either inter prediction or intra prediction, the summer 50 forms a residual video block by subtracting the predictive block from the current video block. The residual video data in the residual block may be included in one or more TUs and is provided to the transform processing unit 52. The transform processing unit 52 transforms the residual video data into residual transform coefficients using a transform, such as a Discrete Cosine Transform (DCT) or a conceptually similar transform.

[0054] The transform processing unit 52 may send the resulting transform coefficients to the quantization unit 54. The quantization unit 54 quantizes the transform coefficients to further reduce the bit rate. The quantization process may also reduce the bit depth associated with some or all of the coefficients. The degree of quantization may be modified by adjusting a quantization parameter. In some examples, the quantization unit 54 may then perform a scan of a matrix including the quantized transform coefficients. Alternatively, the entropy encoding unit 56 may perform the scan.

[0055] Following quantization, the entropy encoding unit 56 entropy encodes the quantized transform coefficients into a video bitstream using, e.g., Context Adaptive Variable Length Coding (CAVLC), Context Adaptive Binary Arithmetic Coding (CABAC), Syntax-based context-adaptive Binary Arithmetic Coding (SBAC), Probability Interval Partitioning Entropy (PIPE) coding or another entropy encoding methodology or technique. The encoded bitstream may then be transmitted to the video decoder 30 as shown in FIG. 1, or archived in the storage device 32 as shown in FIG. 1 for later transmission to or retrieval by the video decoder 30. The entropy encoding unit 56 may also entropy encode the motion vectors and the other syntax elements for the current video frame being coded.

[0056] The inverse quantization unit 58 and the inverse transform processing unit 60 apply inverse quantization and inverse transformation, respectively, to reconstruct the residual video block in the pixel domain for generating a reference block for prediction of other video blocks. As noted above, the motion compensation unit 44 may generate a motion compensated predictive block from one or more reference blocks of the frames stored in the DPB 64. The motion compensation unit 44 may also apply one or more interpolation filters to the predictive block to calculate sub-integer pixel values for use in motion estimation.

[0057] The summer 62 adds the reconstructed residual block to the motion compensated predictive block produced by the motion compensation unit 44 to produce a reference block for storage in the DPB 64. The reference block may then be used by the intra BC unit 48, the motion estimation unit 42 and the motion compensation unit 44 as a predictive block to inter predict another video block in a subsequent video frame.

[0058] FIG. 3 is a block diagram illustrating an exemplary video decoder 30 in accordance with some implementations of the present application. The video decoder 30 includes a video data memory 79, an entropy decoding unit 80, a prediction processing unit 81, an inverse quantization unit 86, an inverse transform processing unit 88, a summer 90, and a DPB 92. The prediction processing unit 81 further includes a motion compensation unit 82, an intra prediction unit 84, and an intra BC unit 85. The video decoder 30 may perform a decoding process generally reciprocal to the encoding process described above with respect to the video encoder 20 in connection with FIG. 2. For example, the motion compensation unit 82 may generate prediction data based on motion vectors received from the entropy decoding unit 80, while the intra-prediction unit 84 may generate prediction data based on intra-prediction mode indicators received from the entropy decoding unit 80.

[0059] In some examples, a unit of the video decoder 30 may be tasked to perform the implementations of the present application. Also, in some examples, the implementations of the present disclosure may be divided among one or more of the units of the video decoder 30. For example, the intra BC unit 85 may perform the implementations of the present application, alone, or in combination with other units of the video decoder 30, such as the motion compensation unit 82, the intra prediction unit 84, and the entropy decoding unit 80. In some examples, the video decoder 30 may not include the intra BC unit 85 and the functionality of intra BC unit 85 may be performed by other components of the prediction processing unit 81, such as the motion compensation unit 82.

[0060] The video data memory 79 may store video data, such as an encoded video bitstream, to be decoded by the other components of the video decoder 30. The video data stored in the video data memory 79 may be obtained, for example, from the storage device 32, from a local video source, such as a camera, via wired or wireless network communication of video data, or by accessing physical data storage media (e.g., a flash drive or hard disk). The video data memory 79 may include a Coded Picture Buffer (CPB) that stores encoded video data from an encoded video bitstream. The DPB 92 of the video decoder 30 stores reference video data for use in decoding video data by the video decoder 30 (e.g., in intra or inter predictive coding modes). The video data memory 79 and the DPB 92 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including Synchronous DRAM (SDRAM), Magneto-resistive RAM (MRAM), Resistive RAM (RRAM), or other types of memory devices. For illustrative purpose, the video data memory 79 and the DPB 92 are depicted as two distinct components of the video decoder 30 in FIG. 3. But it will be apparent to one skilled in the art that the video data memory 79 and the DPB 92 may be provided by the same memory device or separate memory devices. In some examples, the video data memory 79 may be on-chip with other components of the video decoder 30, or off-chip relative to those components.

[0061] During the decoding process, the video decoder 30 receives an encoded video bitstream that represents video blocks of an encoded video frame and associated syntax elements. The video decoder 30 may receive the syntax elements at the video frame level and/or the video block level. The entropy decoding unit 80 of the video decoder 30 entropy decodes the bitstream to generate quantized coefficients, motion vectors or intra-prediction mode indicators, and other syntax elements. The entropy decoding unit 80 then forwards the motion vectors or intra-prediction mode indicators and other syntax elements to the prediction processing unit 81.

[0062] When the video frame is coded as an intra predictive coded (I) frame or for intra coded predictive blocks in other types of frames, the intra prediction unit 84 of the prediction processing unit 81 may generate prediction data for a video block of the current video frame based on a signaled intra prediction mode and reference data from previously decoded blocks of the current frame.

[0063] When the video frame is coded as an inter-predictive coded (i.e., B or P) frame, the motion compensation unit 82 of the prediction processing unit 81 produces one or more predictive blocks for a video block of the current video frame based on the motion vectors and other syntax elements received from the entropy decoding unit 80. Each of the predictive blocks may be produced from a reference frame within one of the reference frame lists. The video decoder 30 may construct the reference frame lists, List 0 and List 1, using default construction techniques based on reference frames stored in the DPB 92.

[0064] In some examples, when the video block is coded according to the intra BC mode described herein, the intra BC unit 85 of the prediction processing unit 81 produces predictive blocks for the current video block based on block vectors and other syntax elements received from the entropy decoding unit 80. The predictive blocks may be within a reconstructed region of the same picture as the current video block defined by the video encoder 20.

[0065] The motion compensation unit 82 and/or the intra BC unit 85 determines prediction information for a video block of the current video frame by parsing the motion vectors and other syntax elements, and then uses the prediction information to produce the predictive blocks for the current video block being decoded. For example, the motion compensation unit 82 uses some of the received syntax elements to determine a prediction mode (e.g., intra or inter prediction) used to code video blocks of the video frame, an inter prediction frame type (e.g., B or P), construction information for one or more of the reference frame lists for the frame, motion vectors for each inter predictive encoded video block of the frame, inter prediction status for each inter predictive coded video block of the frame, and other information to decode the video blocks in the current video frame.

[0066] Similarly, the intra BC unit 85 may use some of the received syntax elements, e.g., a flag, to determine that the current video block was predicted using the intra BC mode, construction information of which video blocks of the frame are within the reconstructed region and should be stored in the DPB 92, block vectors for each intra BC predicted video block of the frame, intra BC prediction status for each intra BC predicted video block of the frame, and other information to decode the video blocks in the current video frame.

[0067] The motion compensation unit 82 may also perform interpolation using the interpolation filters as used by the video encoder 20 during encoding of the video blocks to calculate interpolated values for sub-integer pixels of reference blocks. In this case, the motion compensation unit 82 may determine the interpolation filters used by the video encoder 20 from the received syntax elements and use the interpolation filters to produce predictive blocks.

[0068] The inverse quantization unit 86 inverse quantizes the quantized transform coefficients provided in the bitstream and entropy decoded by the entropy decoding unit 80 using the same quantization parameter calculated by the video encoder 20 for each video block in the video frame to determine a degree of quantization. The inverse transform processing unit 88 applies an inverse transform, e.g., an inverse DCT, an inverse integer transform, or a conceptually similar inverse transform process, to the transform coefficients in order to reconstruct the residual blocks in the pixel domain.

[0069] After the motion compensation unit 82 or the intra BC unit 85 generates the predictive block for the current video block based on the vectors and other syntax elements, the summer 90 reconstructs decoded video block for the current video block by summing the residual block from the inverse transform processing unit 88 and a corresponding predictive block generated by the motion compensation unit 82 and the intra BC unit 85. An in-loop filter 91 such as deblocking filter, SAO filter and/or ALF may be positioned between the summer 90 and the DPB 92 to further process the decoded video block. In some examples, the in-loop filter 91 may be omitted, and the decoded video block may be directly provided by the summer 90 to the DPB 92. The decoded video blocks in a given frame are then stored in the DPB 92, which stores reference frames used for subsequent motion compensation of next video blocks. The DPB 92, or a memory device separate from the DPB 92, may also store decoded video for later presentation on a display device, such as the display device 34 of FIG. 1.

[0070] In a typical video coding process, a video sequence typically includes an ordered set of frames or pictures. Each frame may include three sample arrays, denoted SL, SCb, and SCr. SL is a two-dimensional array of luma samples. SCb is a two-dimensional array of Cb chroma samples. SCr is a two-dimensional array of Cr chroma samples. In other instances, a frame may be monochrome and therefore includes only one two-dimensional array of luma samples.

[0071] As shown in FIG. 4A, the video encoder 20 (or more specifically the partition unit 45) generates an encoded representation of a frame by first partitioning the frame into a set of CTUs. A video frame may include an integer number of CTUs ordered consecutively in a raster scan order from left to right and from top to bottom. Each CTU is a largest logical coding unit and the width and height of the CTU are signaled by the video encoder 20 in a sequence parameter set, such that all the CTUs in a video sequence have the same size being one of 128×128, 64×64, 32×32, and 16×16. But it should be noted that the present application is not necessarily limited to a particular size. As shown in FIG. 4B, each CTU may comprise one CTB of luma samples, two corresponding coding tree blocks of chroma samples, and syntax elements used to code the samples of the coding tree blocks. The syntax elements describe properties of different types of units of a coded block of pixels and how the video sequence may be reconstructed at the video decoder 30, including inter or intra prediction, intra prediction mode, motion vectors, and other parameters. In monochrome pictures or pictures having three separate color planes, a CTU may comprise a single coding tree block and syntax elements used to code the samples of the coding tree block. A coding tree block may be an NxN block of samples.

[0072] To achieve a better performance, the video encoder 20 may recursively perform tree partitioning such as binary-tree partitioning, ternary-tree partitioning, quad-tree partitioning or a combination thereof on the coding tree blocks of the CTU and divide the CTU into smaller CUs. As depicted in FIG. 4C, the 64x64 CTU 400 is first divided into four smaller CUs, each having a block size of 32x32. Among the four smaller CUs, CU 410 and CU 420 are each divided into four CUs of 16x16 by block size. The two 16x16 CUs 430 and 440 are each further divided into four CUs of 8x8 by block size. FIG. 4D depicts a quad-tree data structure illustrating the end result of the partition process of the CTU 400 as depicted in FIG. 4C, each leaf node of the quad-tree corresponding to one CU of a respective size ranging from 32x32 to 8x8. Like the CTU depicted in FIG. 4B, each CU may comprise a CB of luma samples and two corresponding coding blocks of chroma samples of a frame of the same size, and syntax elements used to code the samples of the coding blocks. In monochrome pictures or pictures having three separate color planes, a CU may comprise a single coding block and syntax structures used to code the samples of the coding block. It should be noted that the quad-tree partitioning depicted in FIGS. 4C and 4D is only for illustrative purposes and one CTU may be split into CUs to adapt to varying local characteristics based on quad/ternary/binary-tree partitions. In the multi-type tree structure, one CTU is partitioned by a quad-tree structure and each quad-tree leaf CU may be further partitioned by a binary and ternary tree structure. As shown in FIG. 4E, there are five possible partitioning types of a coding block having a width W and a height H, i.e., quaternary partitioning, horizontal binary partitioning, vertical binary partitioning, horizontal ternary partitioning, and vertical ternary partitioning.

[0073] In some implementations, the video encoder 20 may further partition a coding block of a CU into one or more MxN PBs. A PB is a rectangular (square or non-square) block of samples on which the same prediction, inter or intra, is applied. A PU of a CU may comprise a PB of luma samples, two corresponding PBs of chroma samples, and syntax elements used to predict the PBs. In monochrome pictures or pictures having three separate color planes, a PU may comprise a single PB and syntax structures used to predict the PB. The video encoder 20 may generate predictive luma, Cb, and Cr blocks for luma, Cb, and Cr PBs of each PU of the CU.

[0074] The video encoder 20 may use intra prediction or inter prediction to generate the predictive blocks for a PU. If the video encoder 20 uses intra prediction to generate the predictive blocks of a PU, the video encoder 20 may generate the predictive blocks of the PU based on decoded samples of the frame associated with the PU. If the video encoder 20 uses inter prediction to generate the predictive blocks of a PU, the video encoder 20 may generate the predictive blocks of the PU based on decoded samples of one or more frames other than the frame associated with the PU.

[0075] After the video encoder 20 generates predictive luma, Cb, and Cr blocks for one or more PUs of a CU, the video encoder 20 may generate a luma residual block for the CU by subtracting the CU’s predictive luma blocks from its original luma coding block such that each sample in the CU’s luma residual block indicates a difference between a luma sample in one of the CU's predictive luma blocks and a corresponding sample in the CU's original luma coding block. Similarly, the video encoder 20 may generate a Cb residual block and a Cr residual block for the CU, respectively, such that each sample in the CU's Cb residual block indicates a difference between a Cb sample in one of the CU's predictive Cb blocks and a corresponding sample in the CU's original Cb coding block and each sample in the CU's Cr residual block may indicate a difference between a Cr sample in one of the CU's predictive Cr blocks and a corresponding sample in the CU's original Cr coding block.

[0076] Furthermore, as illustrated in FIG. 4C, the video encoder 20 may use quad-tree partitioning to decompose the luma, Cb, and Cr residual blocks of a CU into one or more luma, Cb, and Cr transform blocks respectively. A transform block is a rectangular (square or non-square) block of samples on which the same transform is applied. A TU of a CU may comprise a transform block of luma samples, two corresponding transform blocks of chroma samples, and syntax elements used to transform the transform block samples. Thus, each TU of a CU may be associated with a luma transform block, a Cb transform block, and a Cr transform block. In some examples, the luma transform block associated with the TU may be a sub-block of the CU's luma residual block. The Cb transform block may be a sub-block of the CU's Cb residual block. The Cr transform block may be a sub-block of the CU's Cr residual block. In monochrome pictures or pictures having three separate color planes, a TU may comprise a single transform block and syntax structures used to transform the samples of the transform block.

[0077] The video encoder 20 may apply one or more transforms to a luma transform block of a TU to generate a luma coefficient block for the TU. A coefficient block may be a two-dimensional array of transform coefficients. A transform coefficient may be a scalar quantity. The video encoder 20 may apply one or more transforms to a Cb transform block of a TU to generate a Cb coefficient block for the TU. The video encoder 20 may apply one or more transforms to a Cr transform block of a TU to generate a Cr coefficient block for the TU.

[0078] After generating a coefficient block (e.g., a luma coefficient block, a Cb coefficient block or a Cr coefficient block), the video encoder 20 may quantize the coefficient block. Quantization generally refers to a process in which transform coefficients are quantized to possibly reduce the amount of data used to represent the transform coefficients, providing further compression. After the video encoder 20 quantizes a coefficient block, the video encoder 20 may entropy encode syntax elements indicating the quantized transform coefficients. For example, the video encoder 20 may perform CABAC on the syntax elements indicating the quantized transform coefficients. Finally, the video encoder 20 may output a bitstream that includes a sequence of bits that forms a representation of coded frames and associated data, which is either saved in the storage device 32 or transmitted to the destination device 14.

[0079] After receiving a bitstream generated by the video encoder 20, the video decoder 30 may parse the bitstream to obtain syntax elements from the bitstream. The video decoder 30 may reconstruct the frames of the video data based at least in part on the syntax elements obtained from the bitstream. The process of reconstructing the video data is generally reciprocal to the encoding process performed by the video encoder 20. For example, the video decoder 30 may perform inverse transforms on the coefficient blocks associated with TUs of a current CU to reconstruct residual blocks associated with the TUs of the current CU. The video decoder 30 also reconstructs the coding blocks of the current CU by adding the samples of the predictive blocks for PUs of the current CU to corresponding samples of the transform blocks of the TUs of the current CU. After reconstructing the coding blocks for each CU of a frame, video decoder 30 may reconstruct the frame.

[0080] As noted above, video coding achieves video compression using primarily two modes, i.e., intra-frame prediction (or intra-prediction) and inter-frame prediction (or inter-prediction). It is noted that IBC could be regarded as either intra-frame prediction or a third mode. Between the two modes, inter-frame prediction contributes more to the coding efficiency than intra-frame prediction because of the use of motion vectors for predicting a current video block from a reference video block.

[0081] But with the ever improving video data capturing technology and more refined video block size for preserving details in the video data, the amount of data required for representing motion vectors for a current frame also increases substantially. One way of overcoming this challenge is to benefit from the fact that not only a group of neighboring CUs in both the spatial and temporal domains have similar video data for predicting purpose but the motion vectors between these neighboring CUs are also similar. Therefore, it is possible to use the motion information of spatially neighboring CUs and/or temporally co-located CUs as an approximation of the motion information (e.g., motion vector) of a current CU by exploring their spatial and temporal correlation, which is also referred to as “Motion Vector Predictor (MVP)” of the current CU.

[0082] Instead of encoding, into the video bitstream, an actual motion vector of the current CU determined by the motion estimation unit 42 as described above in connection with FIG. 2, the motion vector predictor of the current CU is subtracted from the actual motion vector of the current CU to produce a Motion Vector Difference (MVD) for the current CU. By doing so, there is no need to encode the motion vector determined by the motion estimation unit 42 for each CU of a frame into the video bitstream and the amount of data used for representing motion information in the video bitstream may be significantly decreased.

[0083] Like the process of choosing a predictive block in a reference frame during inter-frame prediction of a code block, a set of rules need to be adopted by both the video encoder 20 and the video decoder 30 for constructing a motion vector candidate list (also known as a "merge list") for a current CU using those potential candidate motion vectors associated with spatially neighboring CUs and/or temporally co-located CUs of the current CU and then selecting one member from the motion vector candidate list as a motion vector predictor for the current CU. By doing so, there is no need to transmit the motion vector candidate list itself from the video encoder 20 to the video decoder 30 and an index of the selected motion vector predictor within the motion vector candidate list is sufficient for the video encoder 20 and the video decoder 30 to use the same motion vector predictor within the motion vector candidate list for encoding and decoding the current CU.

[0084] Intra mode coding in VVC

[0085] FIG. 6 shows 67 intra prediction modes. To keep the complexity of the most probable mode (MPM) list generation low, an intra mode coding method with 6 MPMs is used by considering two available neighboring intra modes. The following three aspects are considered to construct the MPM list:

- Default intra modes
- Neighboring intra modes
- Derived intra modes

[0086] A unified 6-MPM list is used for intra blocks irrespective of whether MRL and ISP coding tools are applied or not. The MPM list is constructed based on intra modes of the left and above neighboring blocks. Suppose the mode of the left neighboring block is denoted as Left and the mode of the above neighboring block is denoted as Above, the unified MPM list is constructed as follows:

- When a neighboring block is not available, its intra mode is set to Planar by default.
  - If both modes Left and Above are non-angular modes:
    - MPM list → {Planar, DC, V, H, V − 4, V + 4}
  - If one of modes Left and Above is angular mode, and the other is non-angular:
    - Set a mode Max as the larger mode in Left and Above

  - MPM list → {Planar, Max, Max − 1, Max + 1, Max — 2, Max + 2}
- If Left and Above are both angular and they are different:
  - Set a mode Max as the larger mode in Left and Above
  - Set a mode Min as the smaller mode in Left and Above
  - If Max – Min is equal to 1 :
    - MPM list → {Planar, Left, Above, Min – 1, Max + 1, Min – 2}
  - Otherwise, if Max – Min is greater than or equal to 62 :
    - MPM list → {Planar, Left, Above, Min + 1, Max – 1, Min + 2}
- Otherwise, if Max – Min is equal to 2 :
    - MPM list → {Planar, Left, Above, Min + 1, Min – 1, Max + 1}
  - Otherwise:
    - MPM list → {Planar, Left, Above, Min – 1, –Min + 1, Max – 1}
- If Left and Above are both angular and they are the same:
  - MPM list → {Planar, Left, Left − 1, Left + 1, Left – 2, Left + 2}

[0087] Besides, the first bin of the mpm index codeword is CABAC context coded. In total three contexts are used, corresponding to whether the current intra block is MRL enabled, ISP enabled, or a normal intra block.

[0088] During 6 MPM list generation process, pruning is used to remove duplicated modes so that only unique modes can be included into the MPM list. For entropy coding of the 61 non-MPM modes, a Truncated Binary Code (TBC) is used.

[0089] Position-dependent intra prediction combination

[0090] In VVC, the results of intra prediction of DC, planar and several angular modes are further modified by a position dependent intra prediction combination (PDPC) method. PDPC is an intra prediction method which invokes a combination of the boundary reference samples and HEVC style intra prediction with filtered boundary reference samples. PDPC is applied to the following intra modes without signaling: planar, DC, intra angles less than or equal to horizontal, and intra angles greater than or equal to vertical and less than or equal to 80. If the current block is Bdpcm mode or MRL index is larger than 0, PDPC is not applied.

[0091] The prediction sample pred(x’,y’) is predicted using an intra prediction mode (DC, planar, angular) and a linear combination of reference samples according to the equation as follows:

$$pred(x',y') = Clip(0, (1 << BitDepth) - 1, (wL \times R_{-1,y'} + wT \times R_{x',-1} + (64 - wL - wT) \times pred(x',y') + 32) >> 6)$$

where $R_{x,-1}$, $R_{-1,y}$ represent the reference samples located at the top and left boundaries of current sample (x, y), respectively.

[0092] If PDPC is applied to DC, planar, horizontal, and vertical intra modes, additional boundary filters are not needed, as required in the case of HEVC DC mode boundary filter or horizontal/vertical mode edge filters. PDPC process for DC and Planar modes is identical. For angular modes, if the current angular mode is HOR_IDX or VER_IDX, left or top reference samples is not used, respectively. The PDPC weights and scale factors are dependent on prediction modes and the block sizes. PDPC is applied to the block with both width and height greater than or equal to 4.

[0093] FIGS. 5A-5D illustrate the definition of reference samples ($R_{x,-1}$ and $R_{-1,y}$) for PDPC applied over various prediction modes. FIG. 5A shows an example of a diagonal top-right mode. FIG. 5B shows an example of a diagonal bottom-left mode. FIG. 5C shows an example of an adjacent diagonal top-right mode. FIG. 5D shows an example of an adjacent diagonal bottom-left mode. The prediction sample pred(x’, y’) is located at (x’, y’) within the prediction block. As an example, the coordinate x of the reference sample $R_{x,-1}$ is given by: x = x’ + y’ + 1, and the coordinate y of the reference sample $R_{-1,y}$ is similarly given by: y = x’ + y’ + 1 for the diagonal modes. For the other angular mode, the reference samples $R_{x,-1}$ and $R_{-1,y}$ could be located in fractional sample position. In this case, the sample value of the nearest integer sample location is used.

[0094] As mentioned earlier, the intra prediction samples are generated from either a non-filtered or a filtered set of neighboring reference samples, which may introduce discontinuities along the block boundaries between the current coding block and its neighbors. To resolve such problem, boundary filtering is applied in the HEVC by combing the first row/column of prediction samples of DC, horizontal (i.e., mode 18) and vertical (i.e., mode 50) prediction modes with the unfiltered reference samples utilizing a 2-tap filter (for DC mode) or a gradient-based smoothing filter (for horizontal and vertical prediction modes).

[0095] Gradient PDPC

[0096] In VVC, for a few scenarios, PDPC may not be applied due to the unavailability of the secondary reference samples. A gradient based PDPC, extended from horizontal/vertical mode, is applied. The PDPC weights (wT / wL) and nScale parameter for determining the decay in PDPC weights with respect to the distance from left/top boundary are set equal to corresponding parameters in horizontal/vertical mode, respectively. When the secondary reference sample is at a fractional sample position, bilinear interpolation is applied.

[0097] Decoder-side Intra Mode Derivation (DIMD)

[0098] DIMD is an intra coding tool wherein the luma intra prediction mode (IPM) is not transmitted via the bitstream. Instead, it is derived using previously encoded/decoded pixels, in an identical fashion at the encoder and at the decoder. The DIMD method performs a texture gradient processing to derive 2 best modes. These two modes and planar mode are then applied to the block and their predictors are weighted averaged. The selection of DIMD is signalled in the bitstream for intra coded blocks using a flag. At the decoder, in response to determining that the DIMD flag is true, the intra prediction mode is derived in the reconstruction process using the same previously encoded neighboring pixels. In response to determining that the DIMD flag is not true, the intra prediction mode is parsed from the bitstream as in classical intra coding mode.

[0099] To derive the intra prediction mode for a block, an apparatus such as an encoder or a decoder first selects a set of neighboring pixels to perform a gradient analysis. For normativity purposes, these pixels should be in the decoded / reconstructed pool of pixels. The apparatus may choose, as shown in Figure 6, a template surrounding the current block by T pixels to the left, and T pixels above. Next, the apparatus may perform a gradient analysis on the pixels of the template. This allows to determine a main angular direction for the template, which is assumed to have a high chance to be identical to the one of the current blocks. The apparatus may thus use a simple 3x3 Sobel gradient filter, defined by the following matrices that will be convoluted with the template:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

[00100] For each pixel of the template, the apparatus may point-by-point multiply each of these two matrices with the 3x3 window centered around the current pixel and composed of its 8 direct neighbors, and sum the result. Thus, the apparatus may obtain two values Gx (from the multiplication with Mx), and Gy (from the multiplication with My) corresponding to the gradient at the current pixel, in the horizontal and vertical direction respectively.

[00101] FIG. 8 shows the convolution process. The blue pixel is the current pixel. Red pixels (including the blue) are pixels on which the gradient analysis is possible. Gray pixels are pixels on which the gradient analysis is not possible due to lack of some neighbors. Violet pixels are available (reconstructed) pixels outside of the considered template, used in the gradient analysis of the red pixels. In case a violet pixel is not available (due to blocks being too close to the border of the picture for instance), the gradient analysis of all red pixels that use this violet pixel is not performed. For each red pixel, the apparatus may compute the intensity (G) and the orientation (O) of the gradient using Gx and Gy as such:

$$G = |G_x| + |G_y| \text{ and } O = \text{atan}\,(\frac{G_y}{G_x})$$

[00102] The orientation of the gradient is then converted into an intra angular prediction mode, used to index a histogram (first initialized to zero). The histogram value at that intra angular mode is increased by G. Once all the red pixels in the template have been processed, the histogram will contain cumulative values of gradient intensities, for each intra angular mode. The IPMs corresponding to two tallest histogram bars are selected for the current block. If the maximum value in the histogram is 0 (meaning no gradient analysis was able to be made, or the area composing the template is flat), then the DC mode is selected as intra prediction mode for the current block.

[00103] The two IPMs corresponding to two tallest HoG bars are combined with the Planar mode. In one or more examples, the prediction fusion is applied as a weighted average of the above three predictors. To this aim, the weight of planar is fixed to 21/64 (~1/3). The remaining weight of 43/64 (~2/3) is then shared between the two HoG IPMs, proportionally to the amplitude of their HoG bars. FIG. 9 illustrates an example of this process.

[00104] Derived intra modes are included into the primary list of intra most probable modes (MPM), so the DIMD process is performed before the MPM list is constructed. The primary derived intra mode of a DIMD block is stored with a block and is used for MPM list construction of the neighboring blocks.

[00105] Template-based intra mode derivation (TIMD)

[00106] For each intra mode in MPMs, the sum of absolute transformed differences (SATD) between prediction and reconstruction samples of the template region shown in FIG. 10 is computed and the intra modes with the first two modes with the smallest SATD cost are chosen and then fuse them with the weights, and such weighted intra prediction is used to code the current CU.

[00107] The costs of the two selected modes are compared with a threshold, in the test the cost factor of 2 is applied as follows:

costMode2 < 2*costMode1.

[00108] If this condition is true, the fusion is applied, otherwise the only mode1 is used.

[00109] Weights of the modes are computed from their SATD costs as follows:

weight1 = costMode2/(costMode1+ costMode2)

weight2 = 1 - weight1.

[00110] The DIMD, TIMD modes could derive the intra prediction modes at the decoder side, the derived modes may provide good intra prediction samples (in terms of prediction accuracy) to some extent. On the other hands, the schemes to derive the intra prediction modes at the decoder side (e.g., DIMD, TIMD) do not always derive the best intra prediction modes in terms of prediction accuracy. Here, the prediction accuracy is defined as the sum of absolute difference of the prediction error by applying the derived prediction modes to generate the intra prediction samples. In this disclosure, some schemes are proposed to improve the prediction efficiency of the schemes to derive the intra modes at the decoder. It is noted that the disclosed methods may be applied independently or jointly.

[00111] In one embodiment according to the present disclosure, it is proposed that the intra prediction modes are grouped into several groups which are termed as intra prediction modes groups (IPMG). Each IPMG group contains multiple intra prediction modes. When one IPMG is indicated to be used to generate the intra prediction for the current block, a scheme of intra mode derivation at the decoder side (e.g., DIMD, TIMD or any other schemes) is used to determine the best modes (could be one or more than one) within this IPMG to generate the intra prediction samples for the current block. When multiple modes are chosen, the weighted averaging of all the prediction samples generated by the different modes are used for the current block. As illustrated in FIG. 11, in one example of the present disclosure, the decoder may construct a plurality of IPMGs in Step 1102, and in each IPMG, there are multiple intra prediction modes. In Step 1104, when a current IPMG is indicated to generate an intra prediction mode for a current block , the decoder may derive a scheme of intra prediction modes to determine a best mode within the current IPMG to generate intra prediction samples for the current block, and the scheme of intra prediction modes includes DIMD and TIMD.

[00112] In one example, an IPMG list is constructed by inserting IPMG into the IPMG list based on a predefined order. The same IPMG list construction process is performed both at the encoder and the decoder side.

[00113] At the encoder side, a scheme of intra mode derivation at the decoder side (e.g., DIMD, TIMD or any other schemes) is used to determine the best modes within each IPMG. After all the best modes within each IPMG in the IPMG list are determined, the encoder may select the best one among all the best modes within each IPMG based on the rate-distortion costs (or any other criterion) and signal the index into the bitstream to indicate which IPMG could derive the best intra prediction mode for the current block.

[00114] At the decoder side, the same IPMG list is also constructed by inserting IPMG into the list based on a predefined order. The index is parsed from the bitstream, the decoder than uses the index to indicate which IPMG is used to derive the intra prediction mode for the current block. After the IPMG is selected by the index, a scheme of intra mode derivation at the decoder side (e.g., DIMD, TIMD or any other schemes) is used to determine the final best mode among all the modes within the indicated IPMG for the current block.

[00115] The construction of the IPMG

[00116] In another embodiment, the IPMG is constructed based on intra modes of the neighboring blocks. Suppose the mode of the neighboring block is denoted as M_Neig, the IPMG could be derived as a modes set containing M_Neig, (M_Neig+K)%L, (M_Neig+2K)%L, (M_Neig+3K)%L, …,(M_Neig+N*K)%L, where N is the size of the IPMG, K is a predefined integer and L is another predefined integer.

[00117] In one example, L is set as the number of the available angular intra prediction modes. K is set as L divided by N. When the intra prediction mode is identical to the one in any of the other IPMG, the redundant one is not put into the IPMG, an alternative intra prediction mode is used instead. The alternative mode could be the redundant mode +1, -1, +2, -2, … and so on.

[00118] In yet another embodiment, the IPMGs may be predefined based on the available intra prediction modes. For examples, the IPMGs could be defined as {(I+C)%L, (I+C+D*1)%L, (I+C+D*2)%L, (I+C+D*3)%L,… (I+C+D*(N-1))%L }, where N is the size of the IPMG, I is the starting intra prediction mode, D is the predefined constant to indicate the difference between modes within each IPMG, L is the available angular intra prediction modes and C is the IPMG index starting from 0 to (L/N)-1.

[00119] In one example, I is set as 0; N is 4; L is 67; D is 16. All the IPMGs are listed below:

[00120] {0 ,16 ,32 ,48}

[00121] {1 ,17 ,33 ,49}

[00122] {2 ,18 ,34 ,50}

[00123] {3 ,19 ,35 ,51}

[00124] {4 ,20 ,36 ,52}

[00125] {5 ,21 ,37 ,53}

[00126] {6 ,22 ,38 ,54}

[00127] {7 ,23 ,39 ,55}

[00128] {8 ,24 ,40 ,56}

[00129] {9 ,25 ,41 ,57}

[00130] {10 ,26 ,42 ,58}

[00131] {11 ,27 ,43 ,59}

[00132] {12 ,28 ,44 ,60}

[00133] {13 ,29 ,45 ,61}

[00134] {14 ,30 ,46 ,62}

[00135] {15 ,31 ,47 ,63}

[00136] It is noted that the remaining modes could form another IPMG. In this example, the remaining IPMG is {64, 65, 66}. In yet another way, the remaining modes could be put into the other IPMGs as illustrated below:

[00137] {0 ,16 ,32 ,48, 64}

[00138] {1 ,17 ,33 ,49, 65}

[00139] {2 ,18 ,34 ,50, 66}

[00140] In another embodiment, the DC(0) and Planar(1) are put into one IPMG. The other IPMGs are derived from the other intra prediction modes as {(I+C)%L, (I+C+D*1)%L, (I+C+D*2)%L, (I+C+D*3)%L,… (I+C+D*(N-1))%L }, where N is the size of the IPMG, I is the starting intra prediction mode, D is the predefined constant to indicate the difference between modes within each IPMG, L is the available angular intra prediction modes and C is the IPMG index starting from 0 to (L/N)-1.

[00141] In one example, I is set as 2; N is 4; L is 65 which is derived from 67 (the number of all the available intra modes) subtracted by 2 (DC and Planar); D is 16. All the IPMGs are listed below:

[00142] {2 ,18 ,34 ,50}

[00143] {3 ,19 ,35 ,51}

[00144] {4 ,20 ,36 ,52}

[00145] {5 ,21 ,37 ,53}

[00146] {6 ,22 ,38 ,54}

[00147] {7 ,23 ,39 ,55}

[00148] {8 ,24 ,40 ,56}

[00149] {9 ,25 ,41 ,57}

[00150] {10 ,26 ,42 ,58}

[00151] {11 ,27 ,43 ,59}

[00152] {12 ,28 ,44 ,60}

[00153] {13 ,29 ,45 ,61}

[00154] {14 ,30 ,46 ,62}

[00155] {15 ,31 ,47 ,63}

[00156] {16 ,32 ,48 ,64}

[00157] {17 ,33 ,49 ,65}

[00158] It is noted that the remaining modes could form another IPMG, in this example, the remaining IPMG is {66}. In yet another way, the remaining modes could be put into the other IPMGs as illustrated below:

[00159] {2 ,18 ,34 ,50, 66}

[00160] It is noted that the construction of the IPMG is not limited to the schemes introduced in this disclosure, and any other means to construct the IPMG could be used in the method illustrated in the first embodiment.

[00161] The construction of the IPMG list

[00162] In another embodiment, the IPMG list is constructed by the following steps.

[00163] 1. Check the intra prediction modes of the neighboring blocks based on a predefined order.

[00164] 2. For each neighboring block, perform the following steps until the maximum size of IPMG list is reached.

[00165] a. When a neighboring block is not available, its intra mode is set to Planar by default.

[00166] b. Use the mode of the neighboring block to derive an IPMG using the scheme introduced in second embodiment

[00167] c. If the derived IPMG is not redundant to anyone inserted in the IPMG list, insert this IPMG into the IPMG list at the last position.

[00168] In another embodiment, the IPMG list is constructed by the following steps.

[00169] 1. Check the intra prediction modes of the neighboring blocks based on a predefined order.

[00170] 2. For each neighboring block, perform the following steps until the maximum size of IPMG list is reached.

[00171] a. When a neighboring block is not available, its intra mode is set to Planar by default.

[00172] b. Based on the predefined IPMGs using the schemes illustrated in the third embodiment, insert the IPMG which the neighboring block's mode belongs to into the IPMG list.

[00173] c. If the derived IPMG is not redundant to anyone inserted in the IPMG list, insert this IPMG into the IPMG list at the last position.

[00174] In another embodiment, the IPMG list is constructed by the following steps.

[00175] 1. Check the intra prediction modes of the neighboring blocks based on a predefined order.

[00176] 2. For each neighboring block, perform the following steps until the maximum size of IPMG list is reached.

[00177] a. When a neighboring block is not available, its intra mode is set to Planar by default.

[00178] b. Based on the predefined IPMGs using the schemes illustrated in the fourth embodiment, insert the IPMG which the neighboring block's mode belongs to into the IPMG list.

[00179] c. If the derived IPMG is not redundant to anyone inserted in the IPMG list, insert this IPMG into the IPMG list at the last position.

[00180] It is noted that the construction of the IPMG list is not limited to the schemes introduced in this disclosure, and any other means to construct the IPMG list could be used in the method illustrated in the first embodiment.

[00181] The above methods may be implemented using an apparatus that includes one or more circuitries, which include application specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), controllers, micro-controllers, microprocessors, or other electronic components. The apparatus may use the circuitries in combination with the other hardware or software components for performing the above-described methods. Each module, sub-module, unit, or sub-unit disclosed above may be implemented at least partially using the one or more circuitries.

[00182] FIG. 12 shows a computing environment 1610 coupled with a user interface 1650. The computing environment 1610 may be part of a data processing server. The computing environment 1610 includes a processor 1620, a memory 1630, and an Input/Output (I/O) interface 1640.

[00183] The processor 1620 typically controls overall operations of the computing environment 1610, such as the operations associated with display, data acquisition, data communications, and image processing. The processor 1620 may include one or more processors to execute instructions to perform all or some of the steps in the above-described methods. Moreover, the processor 1620 may include one or more modules that facilitate the interaction between the processor 1620 and other components. The processor may be a Central Processing Unit (CPU), a microprocessor, a single chip machine, a Graphical Processing Unit (GPU), or the like.

[00184] The memory 1630 is configured to store various types of data to support the operation of the computing environment 1610. The memory 1630 may include predetermined software 1632. Examples of such data includes instructions for any applications or methods operated on the computing environment 1610, video datasets, image data, etc. The memory 1630 may be implemented by using any type of volatile or non-volatile memory devices, or a combination thereof, such as a Static Random Access Memory (SRAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), an Erasable Programmable Read-Only Memory (EPROM), a Programmable Read-Only Memory (PROM), a Read-Only Memory (ROM), a magnetic memory, a flash memory, a magnetic or optical disk.

[00185] The I/O interface 1640 provides an interface between the processor 1620 and peripheral interface modules, such as a keyboard, a click wheel, buttons, and the like. The buttons may include but are not limited to, a home button, a start scan button, and a stop scan button. The I/O interface 1640 may be coupled with an encoder and decoder.

[00186] In an embodiment, there is also provided a non-transitory computer-readable storage medium comprising a plurality of programs, for example, in the memory 1630, executable by the processor 1620 in the computing environment 1610, for performing the above-described methods. Alternatively, the non-transitory computer-readable storage medium may have stored therein a bitstream or a data stream comprising encoded video information (for example, video information comprising one or more syntax elements) generated by an encoder (for example, the video encoder 20 in FIG. 2) using, for example, the encoding method described above for use by a decoder (for example, the video decoder 30 in FIG. 3) in decoding video data. The non-transitory computer-readable storage medium may be, for example, a ROM, a Random Access Memory (RAM), a CD-ROM, a magnetic tape, a floppy disc, an optical data storage device or the like.

[00187] In an embodiment, the is also provided a computing device comprising one or more processors (for example, the processor 1620); and the non-transitory computer-readable storage medium or the memory 1630 having stored therein a plurality of programs executable by the one or more processors, wherein the one or more processors, upon execution of the plurality of programs, are configured to perform the above-described methods.

[00188] In an embodiment, there is also provided a computer program product comprising a plurality of programs, for example, in the memory 1630, executable by the processor 1620 in the computing environment 1610, for performing the above-described methods. For example, the computer program product may include the non-transitory computer-readable storage medium.

[00189] In an embodiment, the computing environment 1610 may be implemented with one or more ASICs, DSPs, Digital Signal Processing Devices (DSPDs), Programmable Logic Devices (PLDs), FPGAs, GPUs, controllers, micro-controllers, microprocessors, or other electronic components, for performing the above methods.

[00190] The description of the present disclosure has been presented for purposes of illustration and is not intended to be exhaustive or limited to the present disclosure. Many modifications, variations, and alternative implementations will be apparent to those of ordinary skill in the art having the benefit of the teachings presented in the foregoing descriptions and the associated drawings.

[00191] Unless specifically stated otherwise, an order of steps of the method according to the present disclosure is only intended to be illustrative, and the steps of the method according to the present disclosure are not limited to the order specifically described above, but may be changed according to practical conditions. In addition, at least one of the steps of the method according to the present disclosure may be adjusted, combined or deleted according to practical requirements.

[00192] The examples were chosen and described in order to explain the principles of the disclosure and to enable others skilled in the art to understand the disclosure for various implementations and to best utilize the underlying principles and various implementations with various modifications as are suited to the particular use contemplated. Therefore, it is to be understood that the scope of the disclosure is not to be limited to the specific examples of the implementations disclosed and that modifications and other implementations are intended to be included within the scope of the present disclosure.

CLAIMS

What is claimed is:

1. A method for video decoding, comprising:

constructing, by a decoder, a plurality of intra prediction mode groups (IPMGs), wherein each IPMG of the plurality of IPMGs comprises a plurality of intra prediction modes; and

in response to determining that a current IPMG is indicated to generate an intra prediction mode for a current block, deriving, by the decoder, a scheme of intra prediction modes to determine a best mode within the current IPMG to generate intra prediction samples for the current block, wherein the scheme of intra prediction modes comprises decoder-side intra mode derivation (DIMD) and template-based intra mode derivation (TIMD).

2. The method for video decoding of claim 1, further comprising:

in response to determining that the best mode within the current IPMG comprises a plurality of modes, determining a weighted average of intra prediction samples generated by the plurality of modes; and

applying the weighted average of the intra prediction samples for the current block.

3. The method for video decoding of claim 1, further comprising:

constructing an IPMG list by inserting the plurality of IPMGs into the IPMG list based on a predefined order.

4. The method for video decoding of claim 1, further comprising:

parsing, by the decoder, the index from the bitstream to indicate the IPMG to derive the intra prediction mode for the current block; and

selecting a final best mode for the current block from all modes within the IPMG indicated based on the scheme of intra prediction mode derivation.

5. The method for video decoding of claim 1, further comprising:

constructing, by the decoder, an IPMG based on intra prediction modes of neighboring blocks, wherein the IPMG comprises the intra prediction modes of the neighboring blocks and a plurality of intra prediction modes derived from the intra prediction modes of the neighboring blocks.

6. The method for video decoding of claim 5, further comprising:

constructing the IPMG based on based on the intra prediction modes of the neighboring blocks and a number of available angular intra prediction modes.

7. The method for video decoding of claim 5, further comprising:

in response to determining that an intra prediction mode is identical to one intra prediction mode in the IPMG, selecting another intra prediction mode to add into the IPMG.

8. The method for video decoding of claim 1, further comprising:

constructing, by the decoder, an IPMG based on available intra prediction modes.

9. The method for video decoding of claim 8, further comprising:

constructing the IPMG based on based on available angular intra prediction modes, a starting intra prediction mode, a size of the IPMG, and a predefined constant to indicate difference between modes within each IPMG.

10. The method for video decoding of claim 1, further comprising:

constructing, by the decoder, a first one IPMG to comprise a direct current (DC) mode, and a planar mode; and

constructing, by the decoder, other IPMGs with deriving other intra prediction modes based on available angular intra prediction modes, a starting intra prediction mode, a size of the IPMGs, and a predefined constant to indicate difference between modes within each IPMG.

11. The method for video decoding of claim 10, further comprising:

grouping remaining modes that are not selected into another IPMG.

12. The method for video decoding of claim 3, wherein constructing the IPMG list further comprises:

checking intra prediction modes of neighboring blocks based on the predefined order; and performing following steps on each neighboring block until a maximum size of the IPMG:

in response to determining that a neighboring block is unavailable, selecting a planar mode by default as an intra prediction mode;

deriving an IPMG based on intra prediction modes of neighboring blocks, wherein the IPMG comprises the intra prediction modes of the neighboring blocks and a plurality of intra prediction modes derived from the intra prediction modes of the neighboring blocks; and

in response to determining that the IPMG derived is not identical to any existing IPMG in the IPMG list, inserting the IPMG derived into the IPMG list at last position.

13. The method for video decoding of claim 3, wherein constructing the IPMG list further comprises:

checking intra prediction modes of neighboring blocks based on the predefined order; and

performing following steps on each neighboring block until a maximum size of the IPMG:

in response to determining that a neighboring block is unavailable, selecting a planar mode by default as an intra prediction mode;

deriving the IPMG based on based on available angular intra prediction modes, a starting intra prediction mode, a size of the IPMG, and a predefined constant to indicate difference between modes within each IPMG; and

in response to determining that the IPMG derived is not identical to any existing IPMG in the IPMG list, inserting the IPMG derived into the IPMG list at last position.

14. The method for video decoding of claim 3, wherein constructing the IPMG list further comprises:

checking intra prediction modes of neighboring blocks based on the predefined order; and

performing following steps on each neighboring block until a maximum size of the IPMG:

in response to determining that a neighboring block is unavailable, selecting a planar mode by default as an intra prediction mode;

deriving a first one IPMG to comprise a direct current (DC) mode, and a planar mode;

deriving other IPMGs with deriving other intra prediction modes based on available angular intra prediction modes, a starting intra prediction mode, a size of the IPMGs, and a predefined constant to indicate difference between modes within each IPMG; and

in response to determining that the IPMG derived is not identical to any existing IPMG in the IPMG list, inserting the IPMG derived into the IPMG list at last position.

15. An apparatus for video decoding, comprising:

one or more processors; and

a memory configured to store instructions executable by the one or more processors; wherein the one or more processors, upon execution of the instructions, are configured to perform the method in any of claims 1-14.

16. A non-transitory computer-readable storage medium for video decoding storing computer-executable instructions that, when executed by one or more computer processors, cause the one or more computer processors to perform the method in any of claims 1-14.

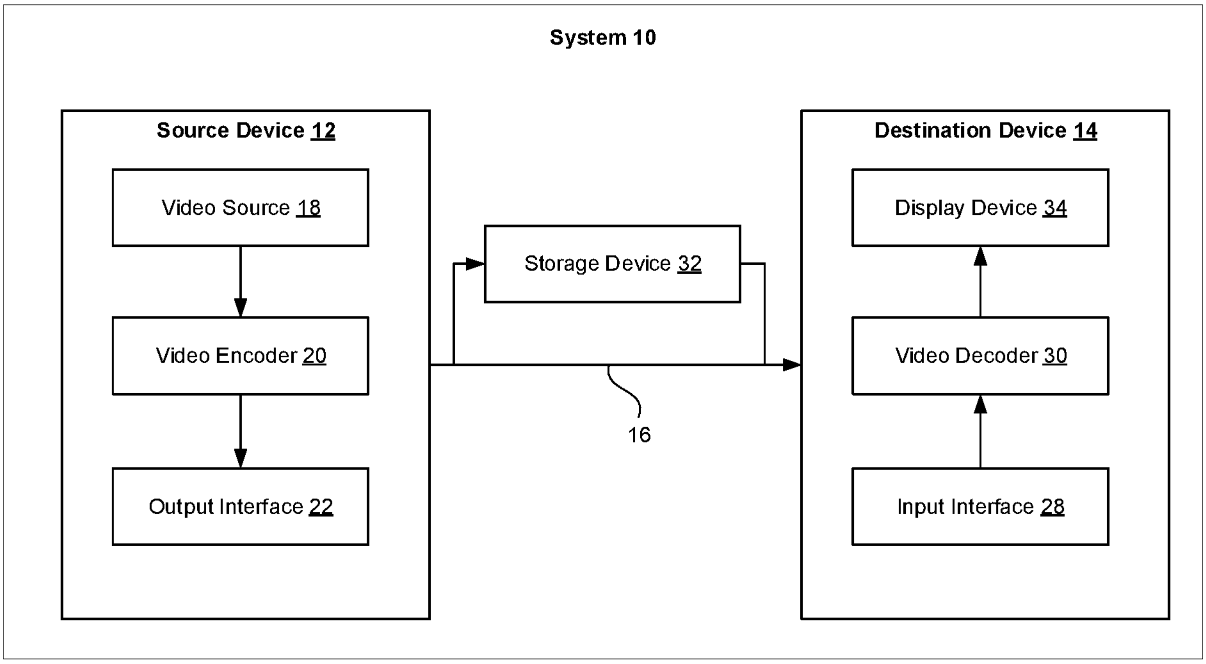
System 10
Source Device 12
Video Source 18
Video Encoder 20
Output Interface 22
Storage Device 32
16
Destination Device 14
Display Device 34
Video Decoder 30
Input Interface 28

FIG. 1

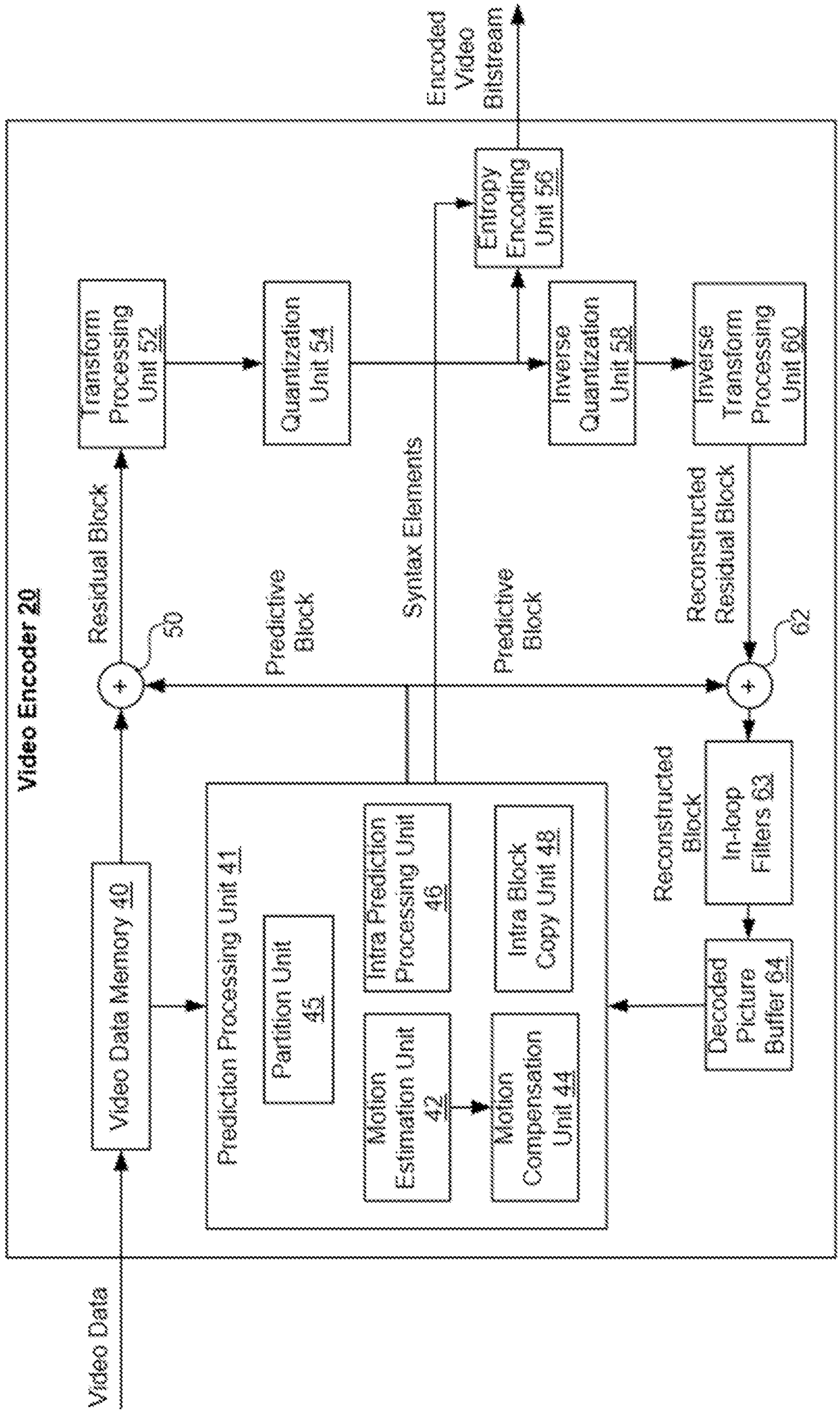
Video Encoder 20
Video Data
Video Data Memory 40
Prediction Processing Unit 41
Partition Unit 45
Motion Estimation Unit 42
Motion Compensation Unit 44
Intra Prediction Processing Unit 46
Intra Block Copy Unit 48
Residual Block
50
Predictive Block
Syntax Elements
Predictive Block
Transform Processing Unit 52
Quantization Unit 54
Entropy Encoding Unit 56
Encoded Video Bitstream
Inverse Quantization Unit 58
Inverse Transform Processing Unit 60
Reconstructed Residual Block
62
Reconstructed Block
In-loop Filters 63
Decoded Picture Buffer 64

FIG. 2

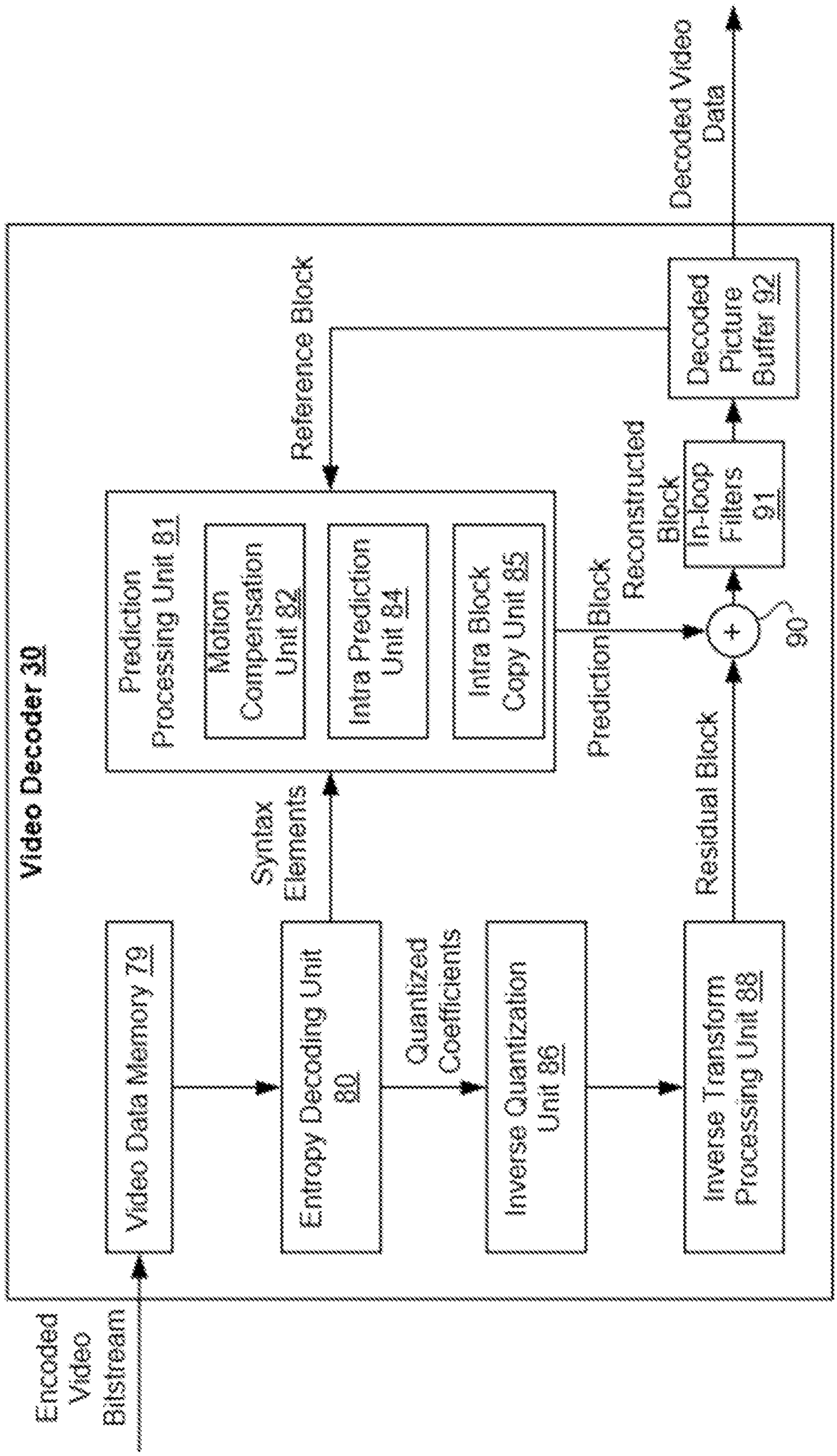
Video Decoder 30
Encoded Video Bitstream
Video Data Memory 79
Entropy Decoding Unit 80
Quantized Coefficients
Inverse Quantization Unit 86
Inverse Transform Processing Unit 88
Syntax Elements
Prediction Processing Unit 81
Motion Compensation Unit 82
Intra Prediction Unit 84
Intra Block Copy Unit 85
Reference Block
Prediction Block
Residual Block
+
90
Reconstructed Block
In-loop Filters 91
Decoded Picture Buffer 92
Decoded Video Data

FIG. 3

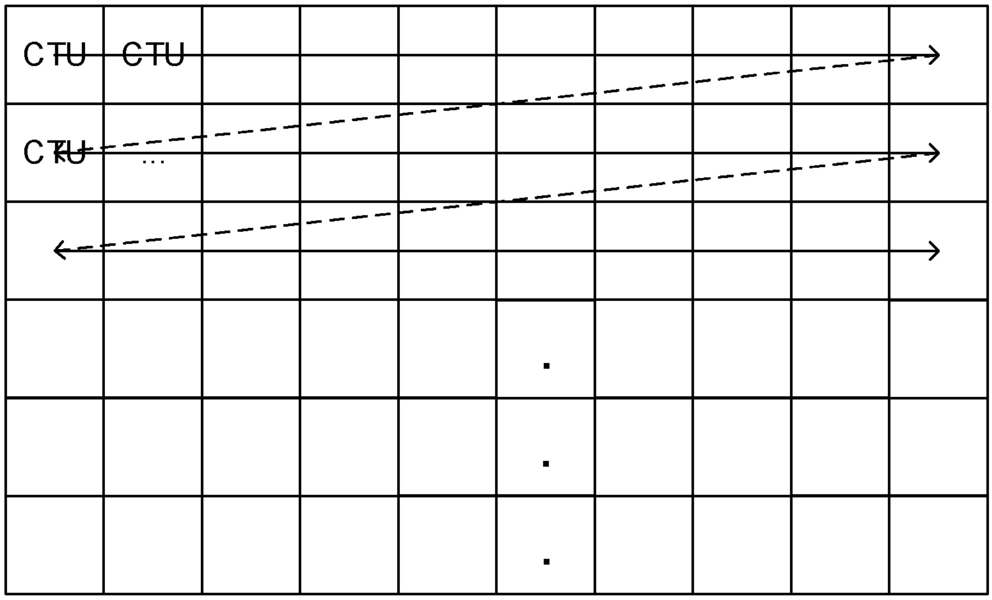
CTU
CTU
CTU
...

FIG. 4A

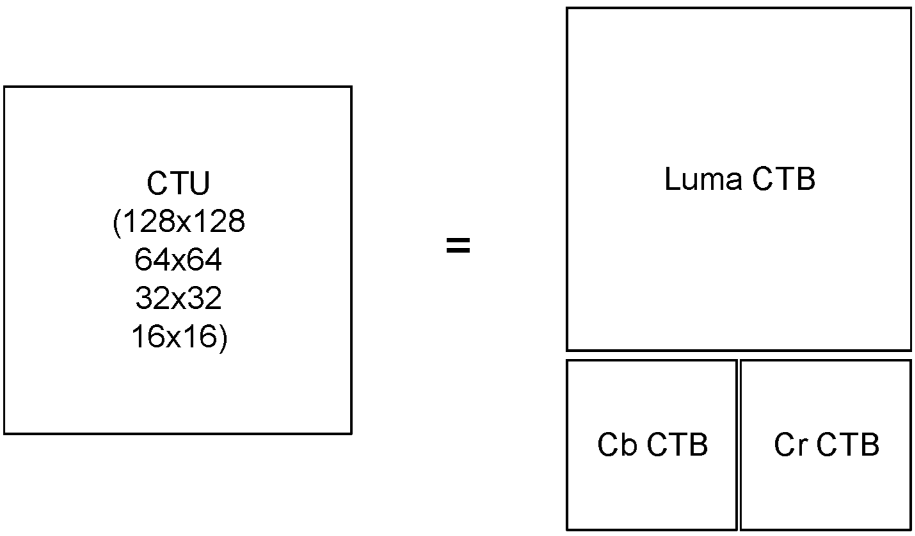
CTU
(128x128
64x64
32x32
16x16)
=
Luma CTB
Cb CTB
Cr CTB

FIG. 4B

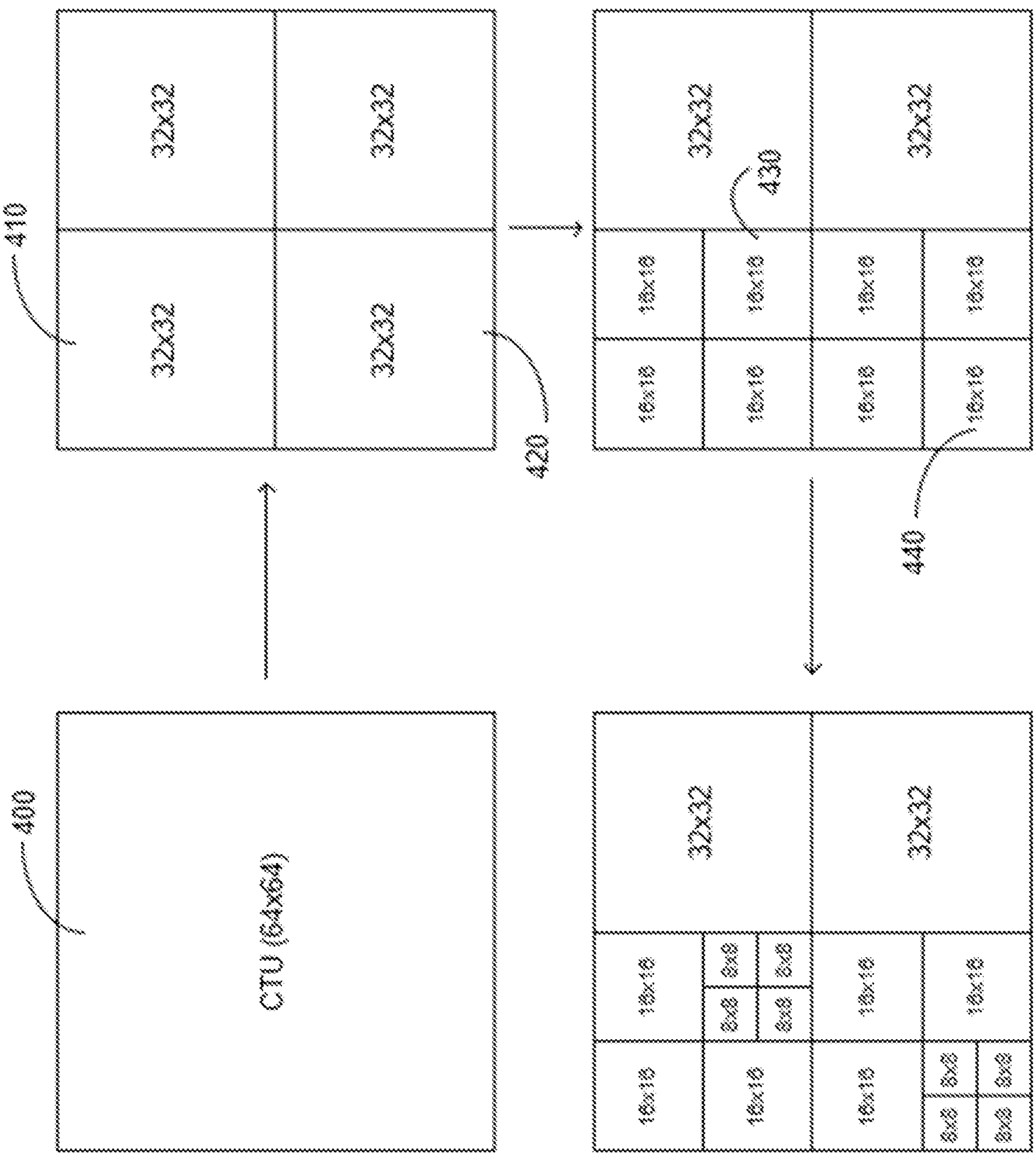

FIG. 4C

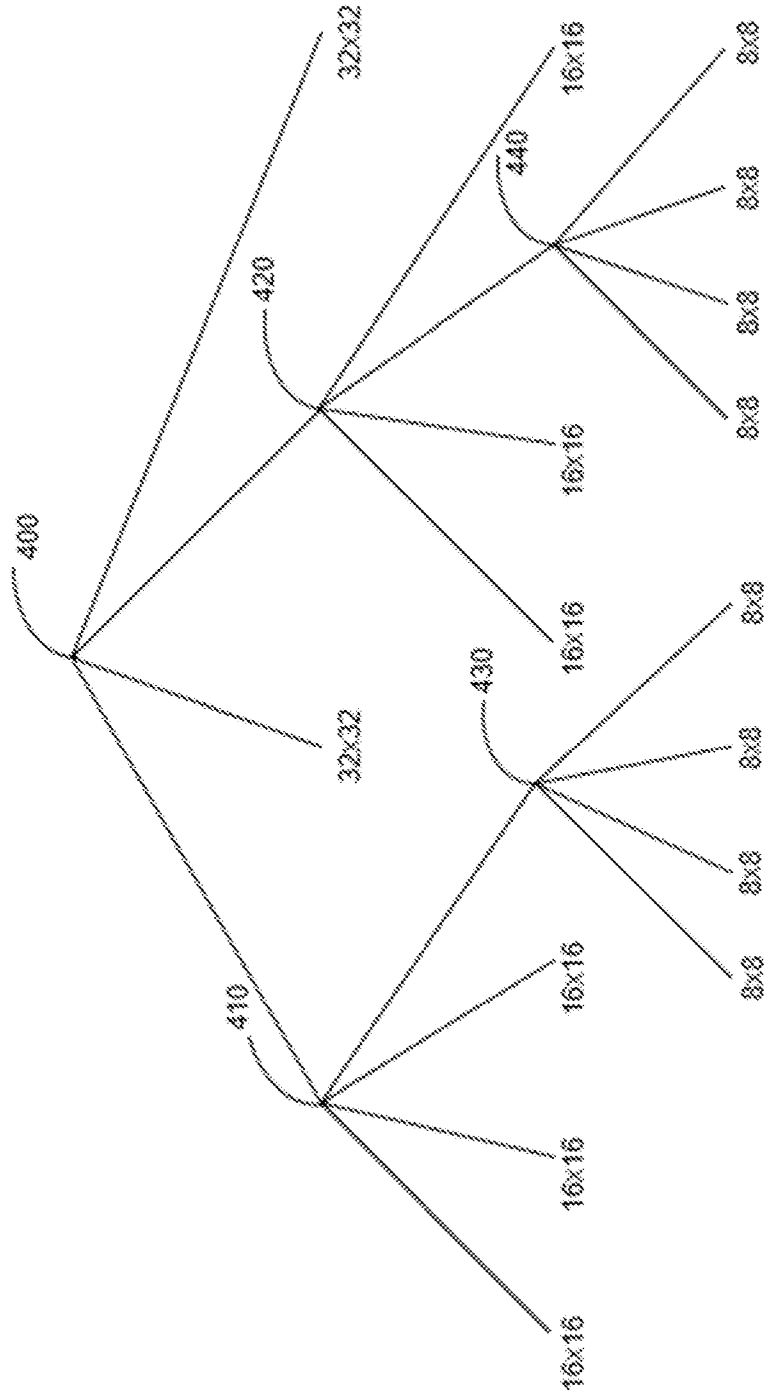
400
410
420
430
440
32x32
32x32
16x16
16x16
16x16
16x16
16x16
16x16
8x8
8x8
8x8
8x8
8x8
8x8
8x8
8x8

FIG. 4D

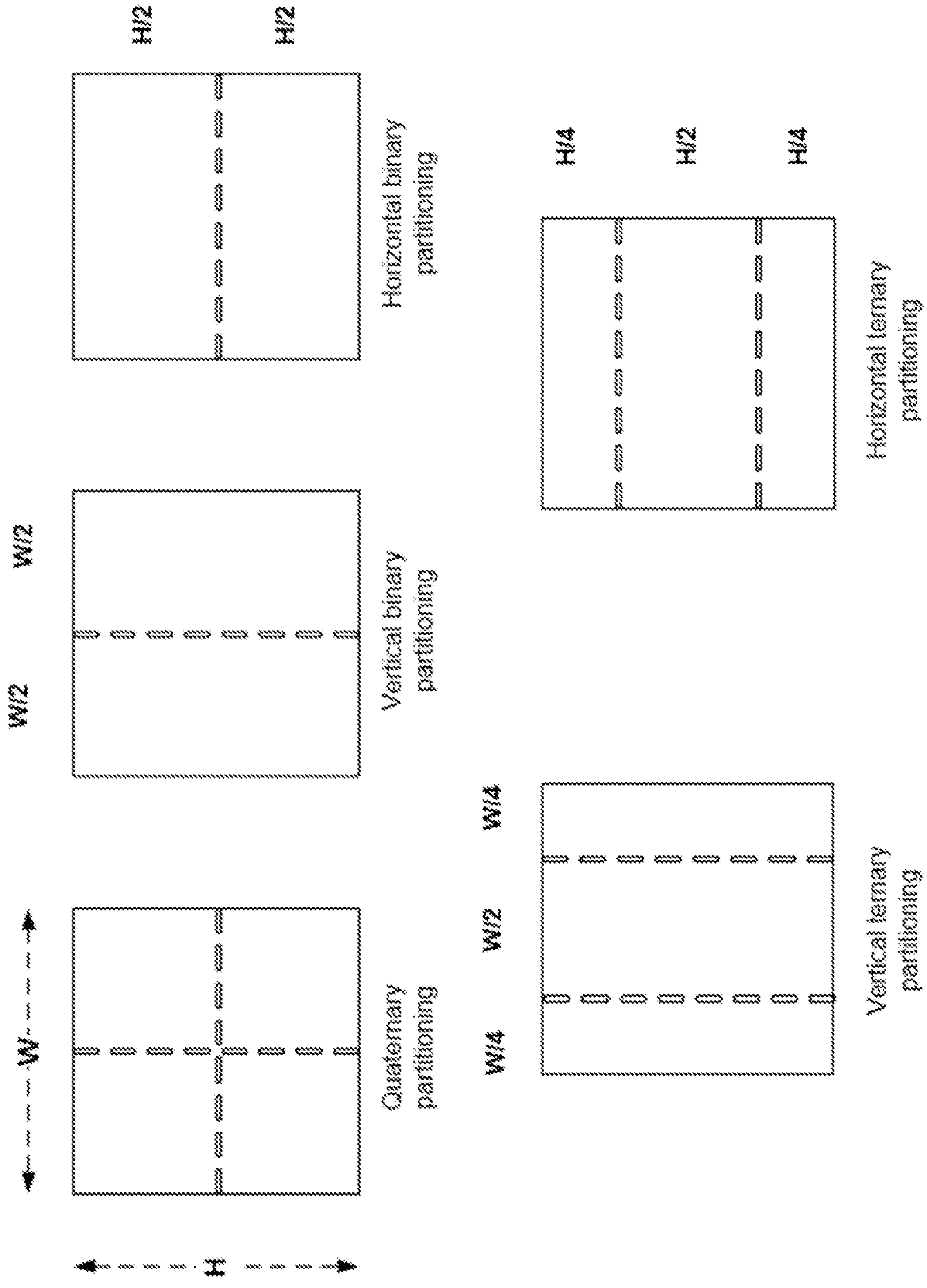
H
W
Quaternary partitioning
W/2
W/2
Vertical binary partitioning
H/2
H/2
Horizontal binary partitioning
W/4
W/2
W/4
Vertical ternary partitioning
H/4
H/2
H/4
Horizontal ternary partitioning

FIG. 4E

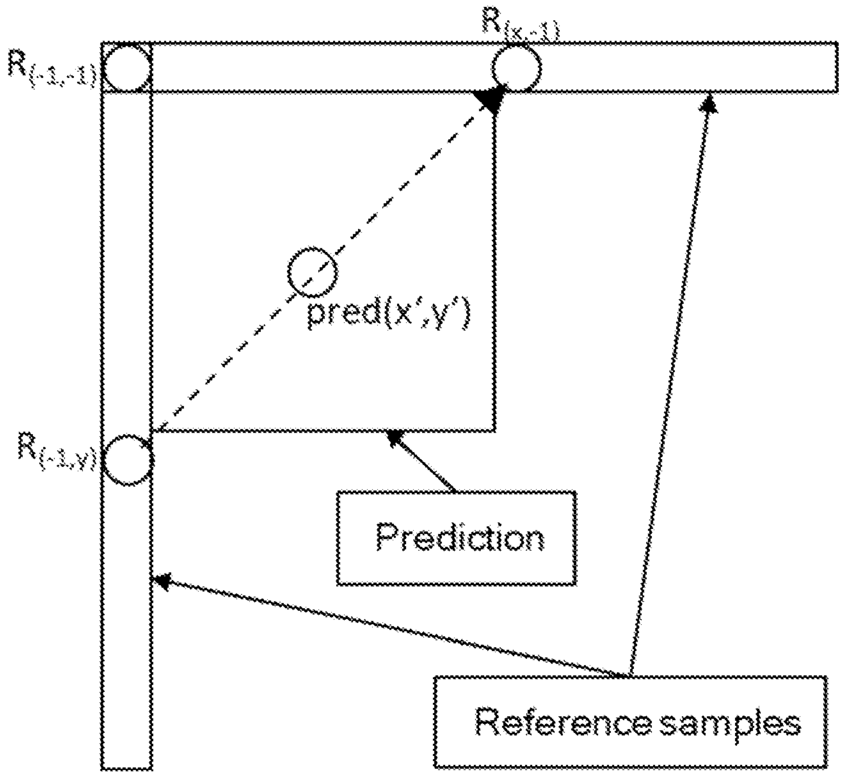
R(x,-1)
R(-1,-1)
pred(x',y')
R(-1,y)
Prediction
Reference samples

FIG. 5A

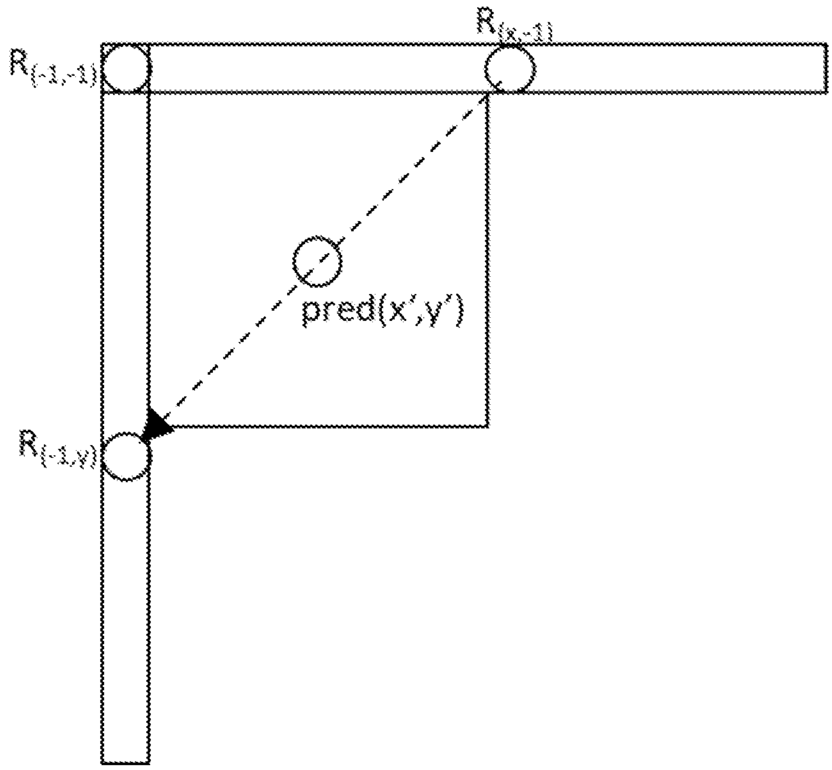
R(x,-1)
R(-1,-1)
pred(x',y')
R(-1,y)

FIG. 5B

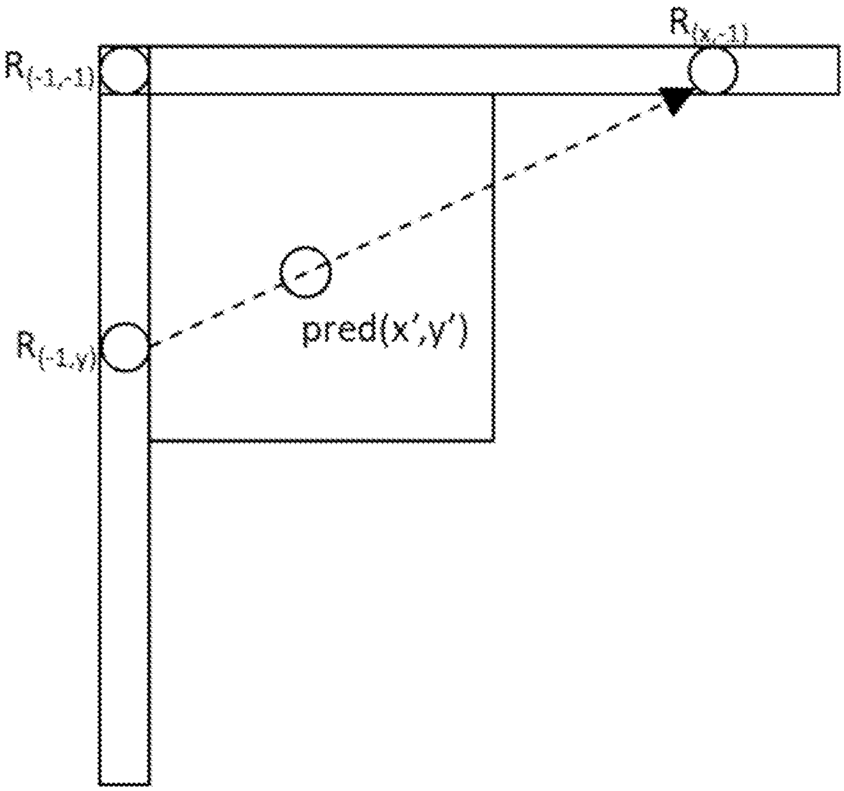

FIG. 5C

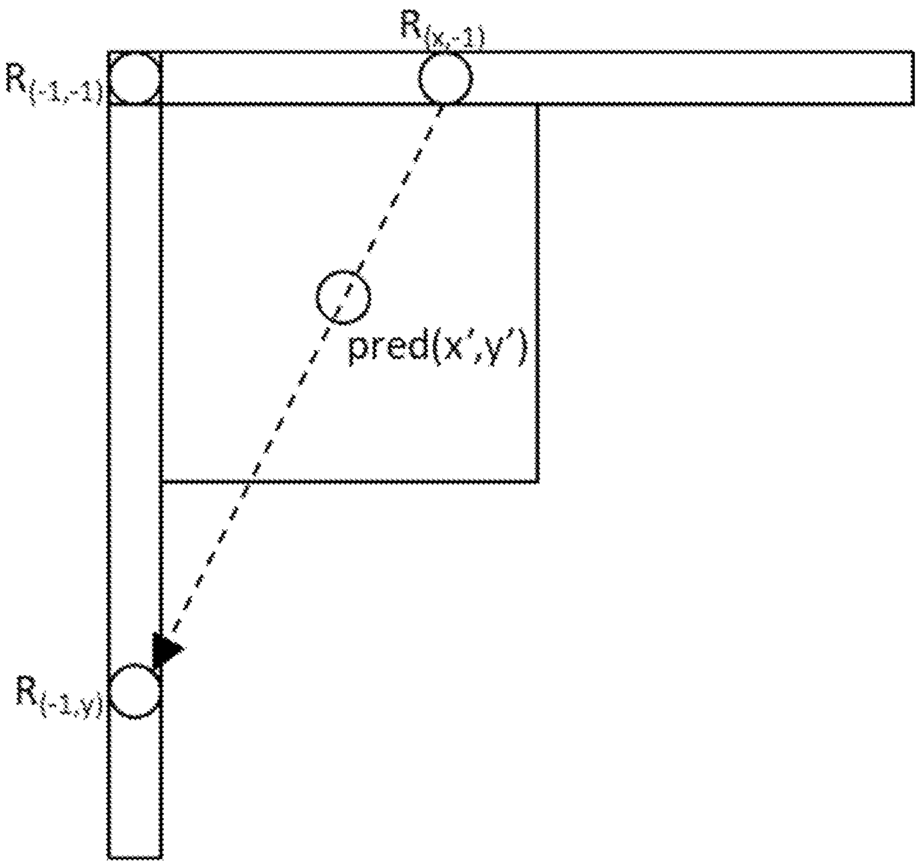

FIG. 5D

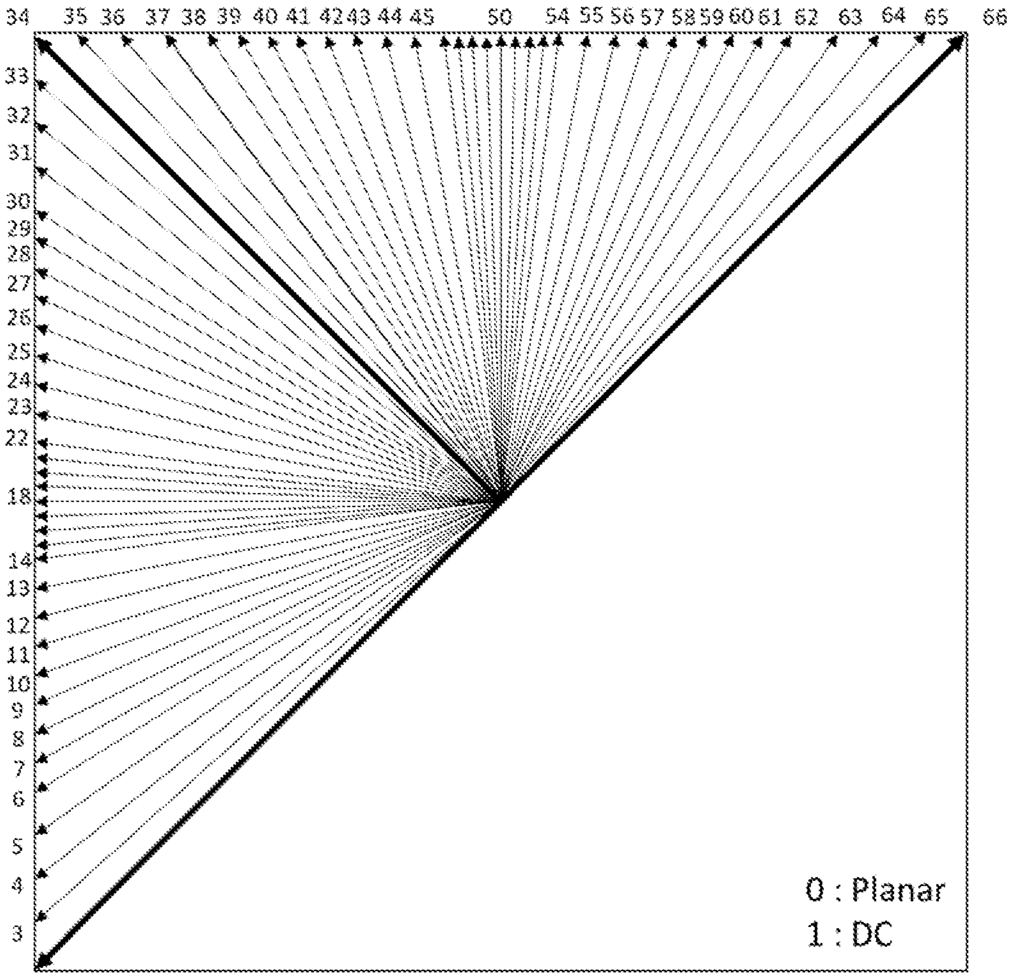
34
35
36
37
38
39
40
41
42
43
44
45
50
54
55
56
57
58
59
60
61
62
63
64
65
66
33
32
31
30
29
28
27
26
25
24
23
22
18
14
13
12
11
10
9
8
7
6
5
4
3
0 : Planar
1 : DC

FIG. 6

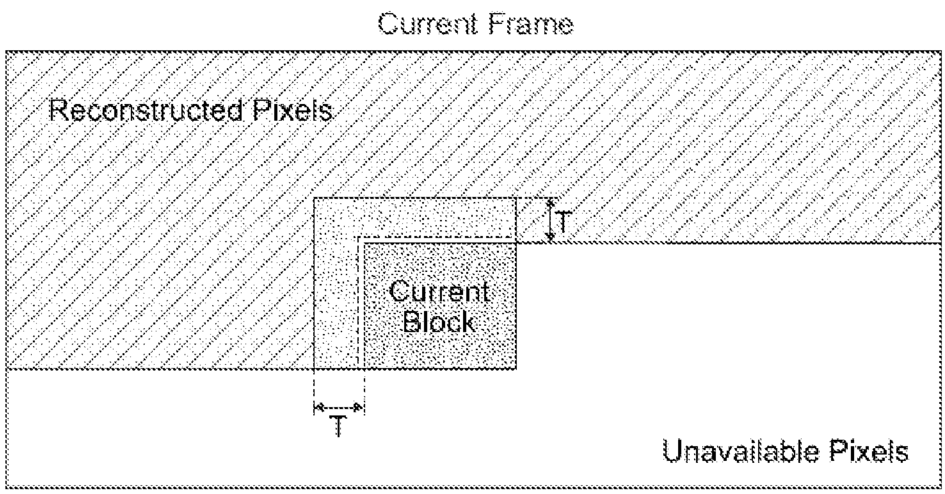
Current Frame
Reconstructed Pixels
T
Current Block
T
Unavailable Pixels

FIG. 7

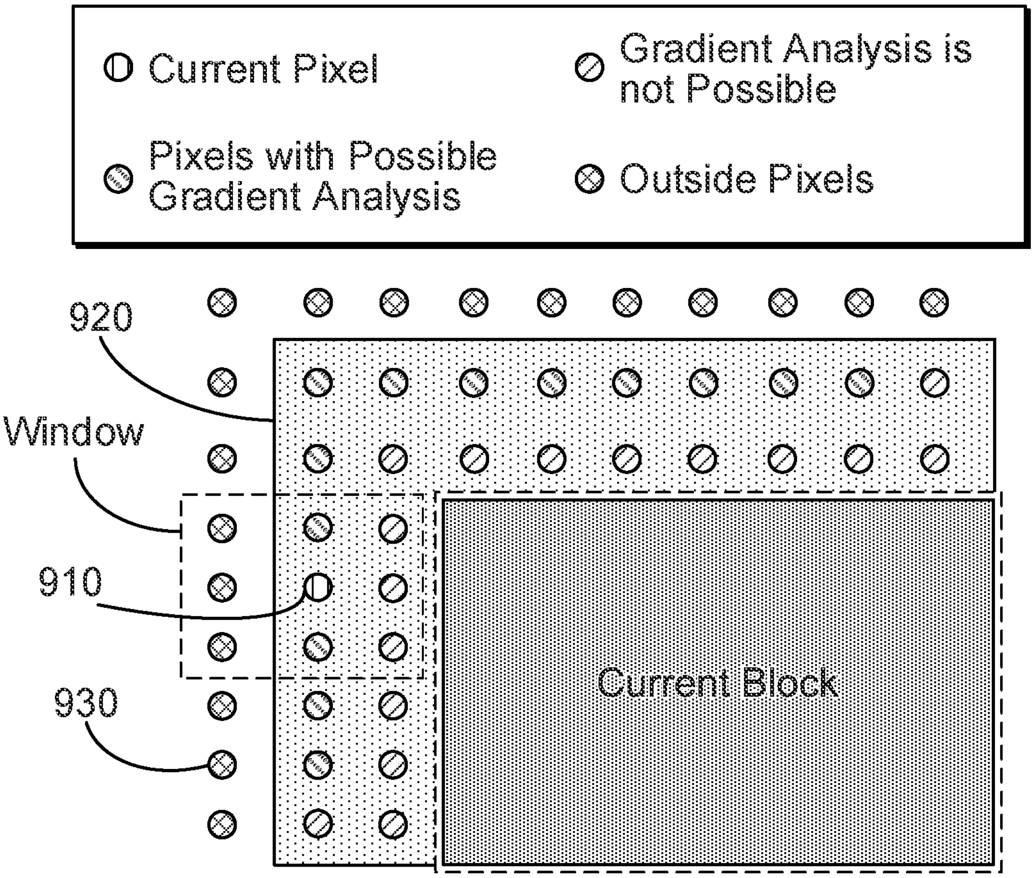
Current Pixel
Gradient Analysis is not Possible
Pixels with Possible Gradient Analysis
Outside Pixels
920
Window
910
930
Current Block

FIG. 8

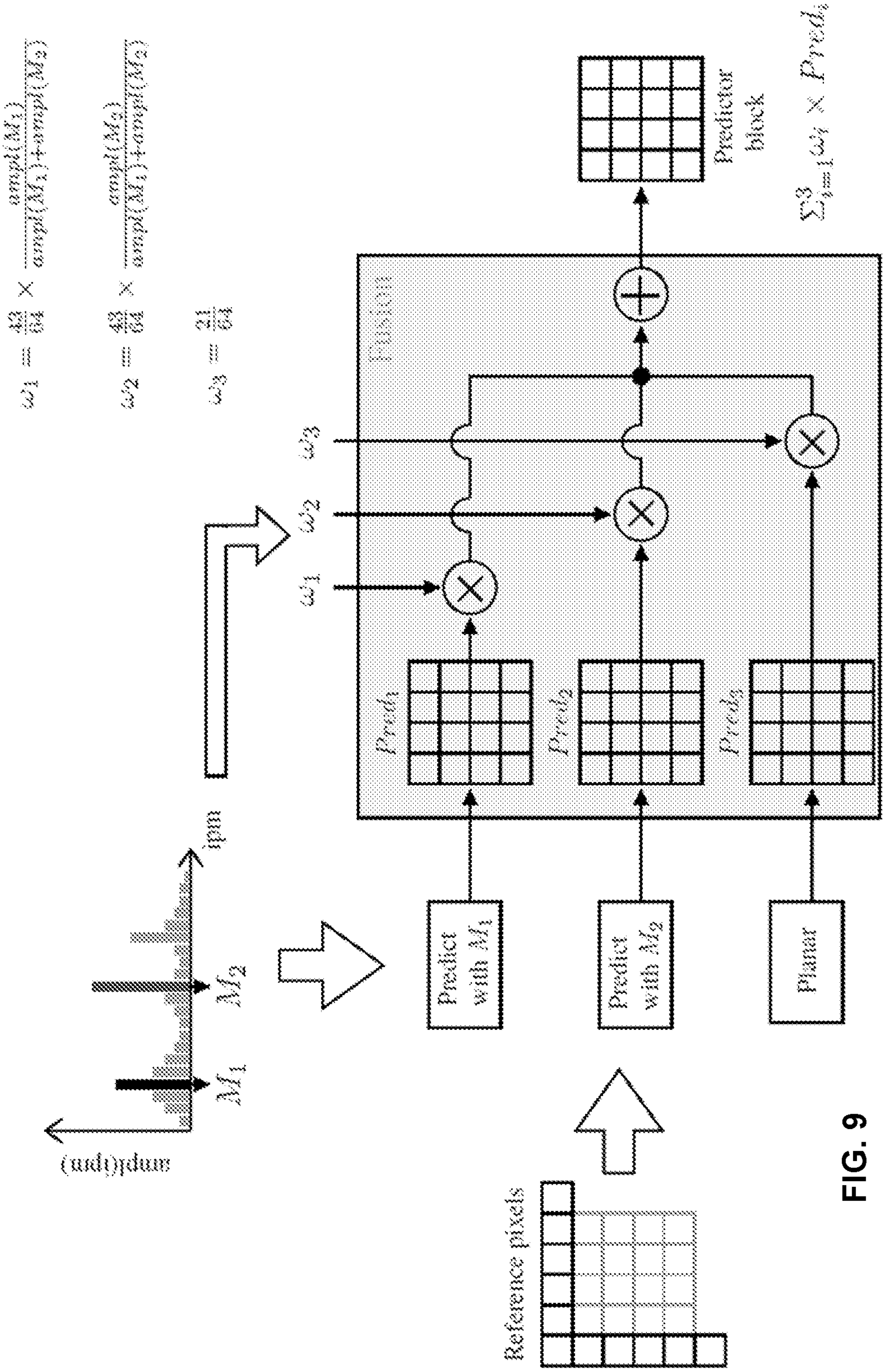
$\omega_1 = \frac{43}{64} \times \frac{ampl(M_1)}{ampl(M_1)+ampl(M_2)}$
$\omega_2 = \frac{43}{64} \times \frac{ampl(M_2)}{ampl(M_1)+ampl(M_2)}$
$\omega_3 = \frac{21}{64}$
ampl(ipm)
ipm
M_1
M_2
Predict with M_1
Predict with M_2
Planar
Reference pixels
FUSION
$Pred_1$
$Pred_2$
$Pred_3$
ω_1
ω_2
ω_3
Predictor block
$\Sigma_{i=1}^{3} \omega_i \times Pred_i$

FIG. 9

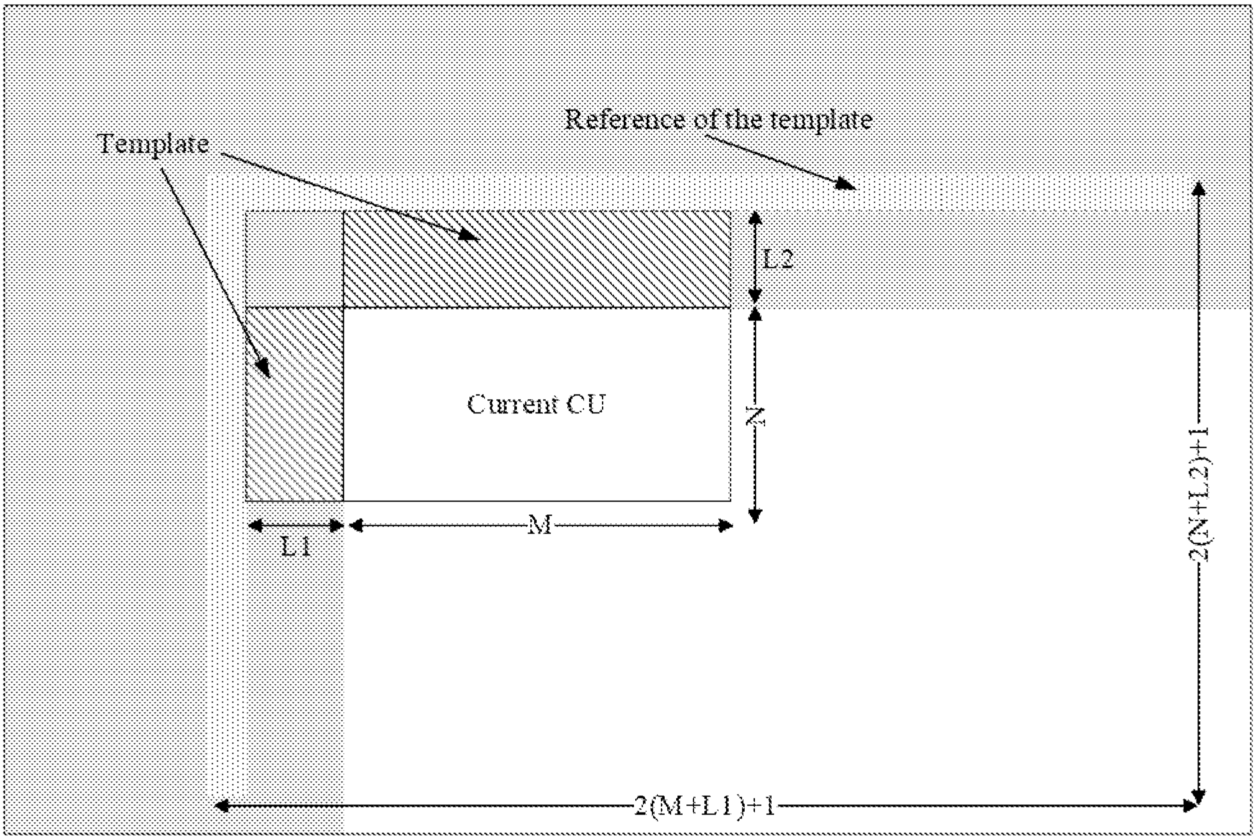
Reference of the template
Template
L2
Current CU
N
L1
M
2(N+L2)+1
2(M+L1)+1

FIG. 10

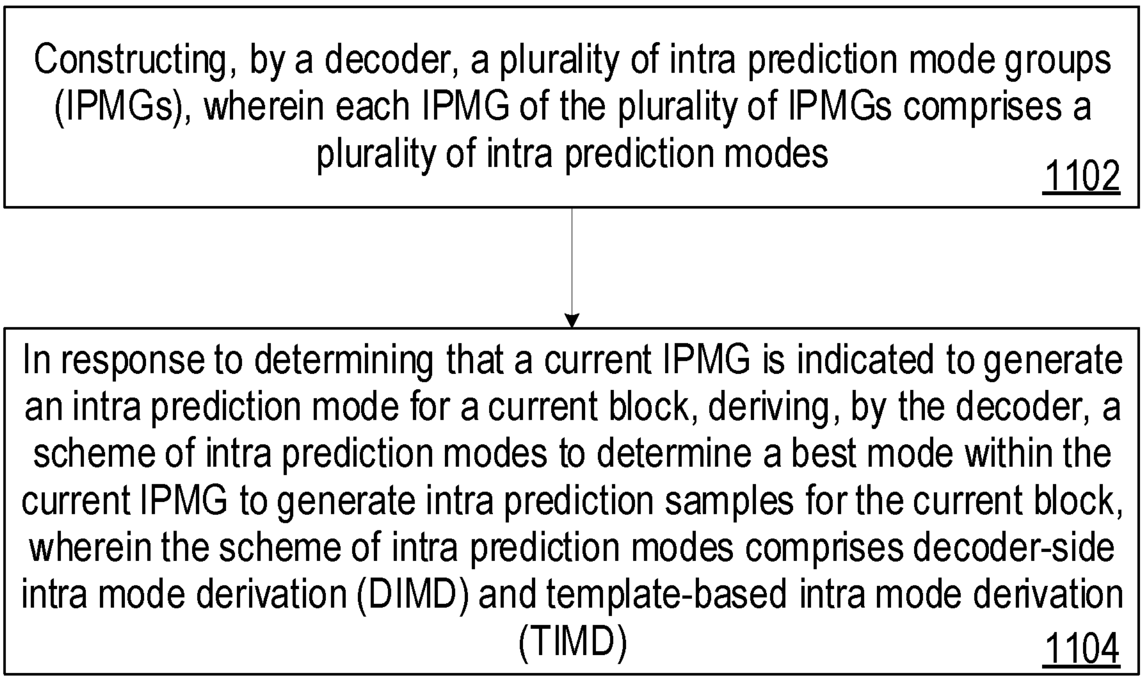
Constructing, by a decoder, a plurality of intra prediction mode groups (IPMGs), wherein each IPMG of the plurality of IPMGs comprises a plurality of intra prediction modes
1102
In response to determining that a current IPMG is indicated to generate an intra prediction mode for a current block, deriving, by the decoder, a scheme of intra prediction modes to determine a best mode within the current IPMG to generate intra prediction samples for the current block, wherein the scheme of intra prediction modes comprises decoder-side intra mode derivation (DIMD) and template-based intra mode derivation (TIMD)
1104

FIG. 11

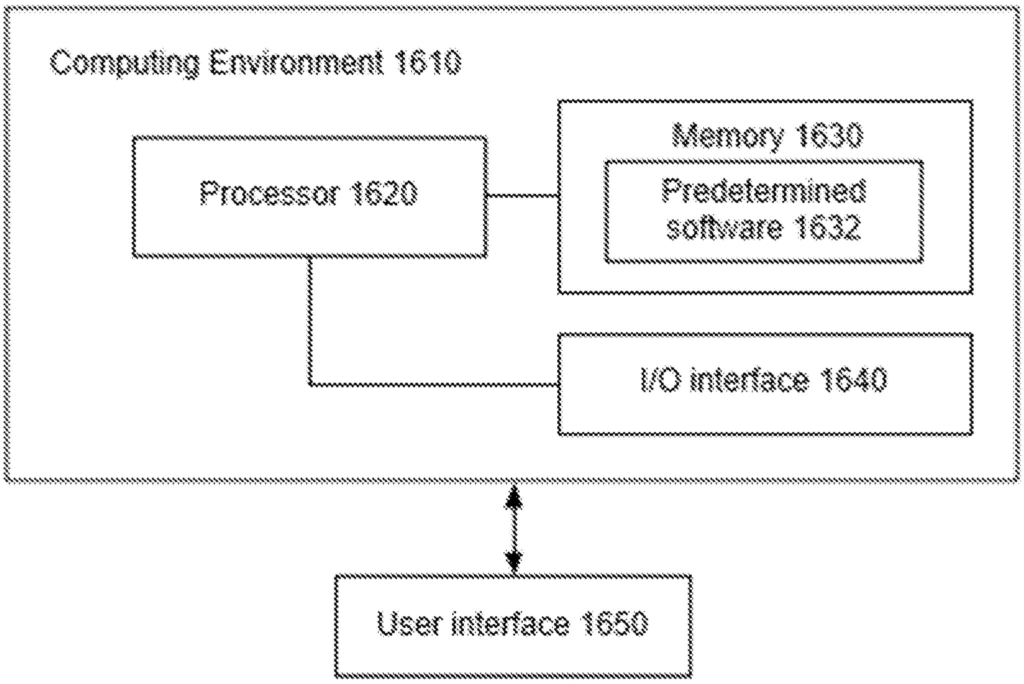
Computing Environment 1610
Processor 1620
Memory 1630
Predetermined software 1632
I/O interface 1640
User interface 1650

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2022/042590

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/157(2014.01)i; **H04N 19/11**(2014.01)i; **H04N 19/593**(2014.01)i; **H04N 19/70**(2014.01)i; **H04N 19/132**(2014.01)i; **H04N 19/44**(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N 19/157(2014.01); H04N 19/105(2014.01); H04N 19/11(2014.01); H04N 19/117(2014.01); H04N 19/13(2014.01); H04N 19/159(2014.01); H04N 19/593(2014.01)

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

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: decoder, template, selection, prediction mode, neighboring block

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	US 2018-0184082 A1 (LG ELECTRONICS INC.) 28 June 2018 (2018-06-28) paragraphs [0146], [0412]-[0537]	1-11,15,16 12-14
Y	MOHSEN ABDOLI. JVET-O0449-v2. Non-CE3: Decoder-side Intra Mode Derivation with Prediction Fusion Using Planar. In: Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. 15th Meeting: Gothenburg, SE, 3 July 2019 pp. 1-9 pages 2-3	1-11,15,16
A	US 2021-0250576 A1 (KI BAEK KIM) 12 August 2021 (2021-08-12) paragraphs [0036]-[0044]	1-16
A	US 2020-0382772 A1 (MEDIATEK INC.) 03 December 2020 (2020-12-03) claims 1-10	1-16

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

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

Date of the actual completion of the international search	Date of mailing of the international search report
16 December 2022	**16 December 2022**
Name and mailing address of the ISA/KR **Korean Intellectual Property Office** **189 Cheongsa-ro, Seo-gu, Daejeon** **35208, Republic of Korea** Facsimile No. **+82-42-481-8578**	Authorized officer **YANG, Jeong Rok** Telephone No. **+82-42-481-5709**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2022/042590

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KR 10-2019-0016989 A (DAEGARAM INC.) 19 February 2019 (2019-02-19) paragraphs [0047]-[0098]	1-16

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2022/042590

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2018-0184082 A1	28 June 2018	KR 10-2018-0008797 A	24 January 2018
		US 10531084 B2	07 January 2020
		WO 2016-204478 A1	22 December 2016
US 2021-0250576 A1	12 August 2021	CN 112335249 A	05 February 2021
		KR 10-2021-0016053 A	10 February 2021
		WO 2020-004902 A1	02 January 2020
US 2020-0382772 A1	03 December 2020	AU 2018-230328 A1	03 October 2019
		AU 2018-230328 B2	06 May 2021
		CA 3055804 A1	13 September 2018
		CN 110393011 A	29 October 2019
		CN 110393011 B	18 February 2022
		EP 3590256 A1	08 January 2020
		EP 3590256 A4	21 October 2020
		TW 201841506 A	16 November 2018
		TW I683573 B	21 January 2020
		US 11228756 B2	18 January 2022
		WO 2018-161954 A1	13 September 2018
KR 10-2019-0016989 A	19 February 2019	CN 104170379 A	26 November 2014
		CN 104170379 B	20 January 2016
		CN 105306932 A	03 February 2016
		CN 105306932 B	13 November 2018
		CN 105306933 A	03 February 2016
		CN 105306933 B	13 November 2018
		CN 105338345 A	17 February 2016
		CN 105338345 B	04 January 2019
		CN 105338346 A	17 February 2016
		CN 105338346 B	13 November 2018
		CN 105338347 A	17 February 2016
		CN 105338347 B	13 November 2018
		CN 105338348 A	17 February 2016
		CN 105338348 B	13 November 2018
		CN 105338349 A	17 February 2016
		CN 105338349 B	02 November 2018
		CN 105338349 B9	05 March 2019
		CN 105338350 A	17 February 2016
		CN 105338350 B	09 October 2018
		DK 2773118 T3	07 December 2020
		EP 2773118 A1	03 September 2014
		EP 2773118 B1	16 September 2020
		EP 3780622 A1	17 February 2021
		EP 3780623 A1	17 February 2021
		EP 3783898 A1	24 February 2021
		EP 3783899 A1	24 February 2021
		ES 2828734 T3	27 May 2021
		HK 1215634 A1	02 September 2016
		HK 1215635 A1	02 September 2016
		HK 1215636 A1	02 September 2016
		HK 1215637 A1	02 September 2016
		HK 1215638 A1	02 September 2016
		HK 1215639 A1	02 September 2016

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2022/042590

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
		HK	1215640	A1	02 September 2016
		HK	1215641	A1	02 September 2016
		HR	P20201980	T1	16 April 2021
		HU	E052290	T2	28 April 2021
		JP	2014-530556	A	17 November 2014
		JP	2015-005998	A	08 January 2015
		JP	2017-017742	A	19 January 2017
		JP	2018-143002	A	13 September 2018
		JP	2020-074638	A	14 May 2020
		JP	5611498	B1	22 October 2014
		JP	6006763	B2	12 October 2016
		JP	6660074	B2	04 March 2020
		JP	6685208	B2	22 April 2020
		JP	6974516	B2	01 December 2021
		KR	10-1947658	B1	14 February 2019
		KR	10-2013-0045150	A	03 May 2013
		KR	10-2013-0045154	A	03 May 2013
		KR	10-2019-0016985	A	19 February 2019
		KR	10-2019-0016986	A	19 February 2019
		KR	10-2019-0016987	A	19 February 2019
		KR	10-2019-0016988	A	19 February 2019
		KR	10-2020-0089807	A	28 July 2020
		KR	10-2292589	B1	24 August 2021
		KR	10-2292590	B1	24 August 2021
		KR	10-2292591	B1	24 August 2021
		KR	10-2292592	B1	24 August 2021
		LT	2773118	T	12 October 2020
		MX	2014003542	A	22 January 2015
		MX	350187	B	29 August 2017
		PT	2773118	T	25 September 2020
		RS	61146	B1	31 December 2020
		SI	2773118	T1	26 February 2021
		US	10341656	B2	02 July 2019
		US	10708584	B2	07 July 2020
		US	11206397	B2	21 December 2021
		US	2014-0219334	A1	07 August 2014
		US	2014-0226720	A1	14 August 2014
		US	2015-0222891	A1	06 August 2015
		US	2015-0222892	A1	06 August 2015
		US	2015-0222897	A1	06 August 2015
		US	2015-0222929	A1	06 August 2015
		US	2016-0353103	A1	01 December 2016
		US	2018-0152701	A1	31 May 2018
		US	2019-0289287	A1	19 September 2019
		US	2020-0314420	A1	01 October 2020
		US	9036704	B2	19 May 2015
		US	9392284	B2	12 July 2016
		US	9426470	B2	23 August 2016
		US	9426471	B2	23 August 2016
		US	9445097	B2	13 September 2016

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2022/042590

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
		US	9769479	B2	19 September 2017
		US	9912946	B2	06 March 2018
		WO	2013-062193	A1	02 May 2013
		WO	2013-062197	A1	02 May 2013