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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 September 2017 (28.09.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/165395 A1

(51) International Patent Classification:
H04N 19/593 (2014.01) *H04N 19/167* (2014.01)
H04N 19/11 (2014.01) *H04N 19/176* (2014.01)

(21) International Application Number: PCT/US2017/023378

(22) International Filing Date: 21 March 2017 (21.03.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/311,265 21 March 2016 (21.03.2016) US
15/463,474 20 March 2017 (20.03.2017) US

(71) Applicant: **QUALCOMM INCORPORATED** [US/US]; ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(72) Inventors: **SEREGIN, Vadim**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **ZHAO, Xin**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **SAID, Amir**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **KARCZEWICZ, Marta**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(74) Agent: **EVANS, Matthew J.**; Shumaker & Sieffert, P.A., 1625 Radio Drive, Suite 100, Woodbury, Minnesota 55125 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

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

(54) **Title**: DETERMINING PREDICTION PARAMETERS FOR NON-SQUARE BLOCKS IN VIDEO CODING

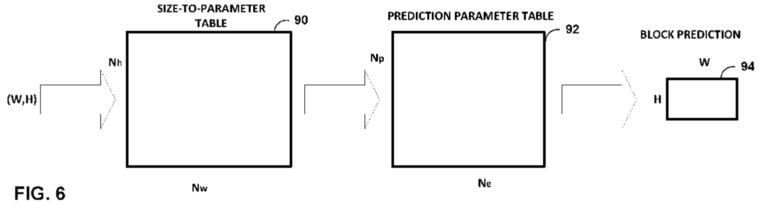


FIG. 6

(57) **Abstract**: A method of decoding video data comprising receiving a block of video data encoded using a position dependent intra prediction combination (PDPC) mode, the block of video data having a non-square shape defined by a width and a height, determining one or more PDPC parameters based on one or more of the width or the height of the block of video data, and decoding the block of video data using the PDPC mode and the determined PDPC parameters.

WO 2017/165395 A1

DETERMINING PREDICTION PARAMETERS FOR NON-SQUARE BLOCKS IN VIDEO CODING

[0001] This application claims the benefit of U.S. Provisional Application No. 62/311,265, filed March 21, 2016, the entire content of which is incorporated by reference herein.

TECHNICAL FIELD

[0002] This disclosure relates to video encoding and video decoding.

BACKGROUND

[0003] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, tablet computers, e-book readers, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, so-called “smart phones,” video teleconferencing devices, video streaming devices, and the like. Digital video devices implement video coding techniques, such as those described in the standards defined by MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), the High Efficiency Video Coding (HEVC or H.265) standard, and extensions of such standards. The video devices may transmit, receive, encode, decode, and/or store digital video information more efficiently by implementing such video coding techniques.

[0004] Video coding techniques include spatial (intra picture) prediction and/or temporal (inter picture) prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video slice (e.g., a video frame or a portion of a video frame) may be partitioned into video blocks, which may also be referred to as treeblocks, coding units (CUs) and/or coding nodes. Pictures may be referred to as frames, and reference pictures may be referred to as reference frames.

[0005] Spatial or temporal prediction results in a predictive block for a block to be coded. Residual data represents pixel differences between the original block to be coded and the predictive block. For further compression, the residual data may be transformed from the pixel domain to a transform domain, resulting in residual transform coefficients, which then may be quantized. Entropy coding may be applied to achieve even more compression.

SUMMARY

[0006] This disclosure describes techniques for coding video data that has been partitioned using an independent luma and chroma partition framework. In some examples, this disclosure describes techniques for determining how to reuse coding information from luma blocks for chroma blocks when there are two or more luma blocks that correspond to the chroma block (e.g., when two or more luma blocks are co-located with a chroma block).

[0007] In other examples, this disclosure describes techniques for determining parameters for a position dependent intra prediction comparison (PDPC) mode when blocks of video data may be partitioned into non-square blocks. In some examples, PDPC parameters may be determined using multiple lookup tables, including separate tables for vertical-related parameters and horizontal-related parameters.

[0008] In one example of the disclosure, a method of decoding video data comprises receiving a bitstream of encoded video data, the encoded video data representing partitioned luma blocks and partitioned chroma blocks, wherein the chroma blocks are partitioned independently of the luma blocks, determining a respective coding mode corresponding to the respective partitioned luma blocks, decoding the respective partitioned luma blocks according to the determined respective coding modes, decoding a first syntax element indicating that the respective coding modes associated with the respective partitioned luma blocks are to be used for decoding a first partitioned chroma block, wherein the first partitioned chroma block is aligned with two or more partitioned luma blocks, determining a chroma coding mode for the first partitioned chroma block according to a function of the respective coding modes of the two or more partitioned luma blocks, and decoding the first partitioned chroma block in accordance with the determined chroma coding mode.

[0009] In another example of the disclosure, an apparatus configured to decode video data comprises a memory configured to store a bitstream of encoded video data and one or more processors configured to receive the bitstream of encoded video data, the encoded video data representing partitioned luma blocks and partitioned chroma blocks, wherein the chroma blocks are partitioned independently of the luma blocks, determine a respective coding mode corresponding to the respective partitioned luma blocks, decode the respective partitioned luma blocks according to the determined respective coding modes, decode a first syntax element indicating that the respective coding modes associated with the respective partitioned luma blocks are to be used for decoding a first partitioned chroma block, wherein the first partitioned chroma block is aligned with two or more partitioned luma blocks, determine a chroma coding mode for the first partitioned chroma block according to a function of the respective coding modes of the two or more partitioned luma blocks, and decode the first partitioned chroma block in accordance with the determined chroma coding mode.

[0010] In another example of the disclosure, an apparatus configured to decode video data comprises means for receiving a bitstream of encoded video data, the encoded video data representing partitioned luma blocks and partitioned chroma blocks, wherein the chroma blocks are partitioned independently of the luma blocks, means for determining a respective coding mode corresponding to the respective partitioned luma blocks, means for decoding the respective partitioned luma blocks according to the determined respective coding modes, means for decoding a first syntax element indicating that the respective coding modes associated with the respective partitioned luma blocks are to be used for decoding a first partitioned chroma block, wherein the first partitioned chroma block is aligned with two or more partitioned luma blocks, means for determining a chroma coding mode for the first partitioned chroma block according to a function of the respective coding modes of the two or more partitioned luma blocks, and means for decoding the first partitioned chroma block in accordance with the determined chroma coding mode.

[0011] In another example, this disclosure describes a non-transitory computer-readable storage medium storing instructions that, when executed, causes one or more processors configured to decoded video data to receive the bitstream of encoded video data, the encoded video data representing partitioned luma blocks and partitioned chroma blocks, wherein the chroma blocks are partitioned independently of the luma blocks, determine a respective coding mode corresponding to the respective partitioned luma blocks, decode the respective partitioned luma blocks according to the determined respective coding modes, decode a first syntax element indicating that the respective coding modes associated with the respective partitioned luma blocks are to be used for decoding a first partitioned chroma block, wherein the first partitioned chroma block is aligned with two or more partitioned luma blocks, determine a chroma coding mode for the first partitioned chroma block according to a function of the respective coding modes of the two or more partitioned luma blocks, and decode the first partitioned chroma block in accordance with the determined chroma coding mode.

[0012] In another example of the disclosure, a method of decoding video data comprises receiving a block of video data encoded using a position dependent intra prediction combination (PDPC) mode, the block of video data having a non-square shape defined by a width and a height, determining one or more PDPC parameters based on one or more of the width or the height of the block of video data, and decoding the block of video data using the PDPC mode and the determined PDPC parameters.

[0013] In another example of the disclosure, an apparatus configured to decode video data comprises a memory configured to store a block of video data encoded using a PDPC mode, the block of video data having a non-square shape defined by a width and a height, and one or more processors configured to receive the block of video data, determine one or more PDPC parameters based on one or more of the width or the height of the block of video data, and decode the block of video data using the PDPC mode and the determined PDPC parameters.

[0014] In another example of the disclosure, an apparatus configured to decode video data comprises means for receiving a block of video data encoded using a PDPC mode, the block of video data having a non-square shape defined by a width and a height, means for determining one or more PDPC parameters based on one or more of the width or the height of the block of video data, and means for decoding the block of video data using the PDPC mode and the determined PDPC parameters.

[0015] In another example, this disclosure describes a non-transitory computer-readable storage medium storing instructions that, when executed, cause one or more processors of a device to configured to decode video data to receive a block of video data encoded using a PDPC mode, the block of video data having a non-square shape defined by a width and a height, determine one or more PDPC parameters based on one or more of the width or the height of the block of video data, and decode the block of video data using the PDPC mode and the determined PDPC parameters.

[0016] In another example of the disclosure, a method of encoding video data comprises receiving a block of video data, the block of video data having a non-square shape defined by a width and a height, determining one or more PDPC parameters based on one or more of the width or the height of the block of video data, and encoding the block of video data using a PDPC mode and the determined PDPC parameters.

[0017] In another example of the disclosure, an apparatus configured to encode video data comprises a memory configured to store a block of video data, the block of video data having a non-square shape defined by a width and a height, and one or more processors configured to receive the block of video data, determine one or more PDPC parameters based on one or more of the width or the height of the block of video data, and encode the block of video data using a PDPC mode and the determined PDPC parameters.

[0018] The details of one or more aspects of the disclosure are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the techniques described in this disclosure will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0019] FIG. 1 is a block diagram illustrating an example video encoding and decoding system configured to implement techniques of the disclosure.

[0020] FIG. 2A is a conceptual diagram illustrating an example of block partitioning using a quadtree plus binary tree (QTBT) structure.

[0021] FIG. 2B is a conceptual diagram illustrating an example tree structure corresponding to the block partitioning using the QTBT structure of FIG. 2A.

[0022] FIG. 3 is a conceptual diagram illustrating an example of luma and chroma relative partitioning in accordance with the techniques of this disclosure.

[0023] FIG. 4 is a conceptual diagram illustrating another example of luma and chroma relative partitioning in accordance with the techniques of this disclosure.

[0024] FIG. 5A illustrates a prediction of a 4x4 block using an unfiltered reference according to techniques of this disclosure.

[0025] FIG. 5B illustrates a prediction of a 4x4 block using a filtered reference according to techniques of this disclosure.

[0026] FIG. 6 is a conceptual diagram illustrating the use of nested tables for determining a set of prediction parameters used in a rectangular block in accordance with one example of the disclosure.

[0027] FIG. 7 is a block diagram illustrating an example of a video encoder configured to implement techniques of the disclosure.

[0028] FIG. 8 is a block diagram illustrating an example of a video decoder configured to implement techniques of the disclosure.

[0029] FIG. 9 is a flowchart illustrating an example operation of a video coder in accordance with a technique of this disclosure.

[0030] FIG. 10 is a flowchart illustrating an example operation of a video decoder in accordance with a technique of this disclosure.

[0031] FIG. 11 is a flowchart illustrating an example operation of a video encoder in accordance with a technique of this disclosure.

[0032] FIG. 12 is a flowchart illustrating an example operation of a video decoder in accordance with a technique of this disclosure.

DETAILED DESCRIPTION

[0033] According to some video block partitioning techniques, chroma blocks of video data are partitioned independently of luma blocks of video data, such that some chroma blocks may not be directly aligned with a single corresponding luma block. As such, it becomes difficult to reuse syntax elements related to luma blocks for chroma blocks, as there may not be a one-to-one correspondence between luma blocks and chroma blocks. This disclosure describes techniques for coding a chroma block of video data using information (e.g., syntax elements) corresponding to a luma block of video data in situations where luma and chroma blocks are partitioned independently.

[0034] This disclosure also describes techniques for determining coding parameters for a position dependent intra prediction combination (PDPC) coding mode. In one example, this disclosure describes techniques for determining PDPC parameters for video blocks partitioned into non-square blocks (e.g., non-square, rectangular blocks).

[0035] FIG. 1 is a block diagram illustrating an example video encoding and decoding system 10 that may be configured to perform the techniques of this disclosure. As shown in FIG. 1, system 10 includes source device 12 that provides encoded video data to be decoded at a later time by destination device 14. In particular, source device 12 provides the video data to destination device 14 via computer-readable medium 16. Source device 12 and destination device 14 may comprise any of a wide range of devices, including desktop computers, notebook (e.g., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called “smart” phones (or more generally, mobile stations), tablet computers, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or the like. A mobile station may be any device capable of communicating over a wireless network. In some cases, source device 12 and destination device 14 may be equipped for wireless communication. Thus, source device 12 and destination device 14 may be wireless communication devices (e.g., mobile stations). Source device 12 is an example video encoding device (i.e., a device for encoding video data). Destination device 14 is an example video decoding device (i.e., a device for decoding video data).

[0036] In the example of FIG. 1, source device 12 includes video source 18, storage media 20 configured to store video data, video encoder 22, and output interface 24. Destination device 14 includes input interface 26, storage media 28 configured to store encoded video data, video decoder 30, and display device 32. In other examples, source device 12 and destination device 14 include other components or arrangements. For example, source device 12 may receive video data from an external video source, such as an external camera. Likewise, destination device 14 may interface with an external display device, rather than including an integrated display device 32.

[0037] The illustrated system 10 of FIG. 1 is merely one example. Techniques for processing and/or coding (e.g., encoding and/or decoding) video data may be performed by any digital video encoding and/or decoding device. Although the techniques of this disclosure are generally performed by a video encoding device and/or video decoding device, the techniques may also be performed by a video encoder/decoder, typically referred to as a "CODEC." Source device 12 and destination device 14 are merely examples of such coding devices in which source device 12 generates coded video data for transmission to destination device 14. In some examples, source device 12 and destination device 14 may operate in a substantially symmetrical manner such that each of source device 12 and destination device 14 include video encoding and decoding components. Hence, system 10 may support one-way or two-way video transmission between source device 12 and destination device 14, e.g., for video streaming, video playback, video broadcasting, or video telephony.

[0038] Video source 18 of source device 12 may include a video capture device, such as a video camera, a video archive containing previously captured video, and/or a video feed interface to receive video data from a video content provider. As a further alternative, video source 18 may generate computer graphics-based data as the source video, or a combination of live video, archived video, and computer-generated video. Source device 12 may comprise one or more data storage media (e.g., storage media 20) configured to store the video data. The techniques described in this disclosure may be applicable to video coding in general, and may be applied to wireless and/or wired applications. In each case, the captured, pre-captured, or computer-generated video may be encoded by video encoder 22. Output interface 24 may output the encoded video information (e.g., a bitstream of encoded video data) to computer-readable medium 16.

[0039] Destination device 14 may receive the encoded video data to be decoded via computer-readable medium 16. Computer-readable medium 16 may comprise any type of medium or device capable of moving the encoded video data from source device 12 to destination device 14. In some examples, computer-readable medium 16 comprises a communication medium to enable source device 12 to transmit encoded video data directly to 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 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 source device 12 to destination device 14. Destination device 14 may comprise one or more data storage media configured to store encoded video data and decoded video data.

[0040] In some examples, encoded data may be output from output interface 24 to a storage device. Similarly, encoded data may be accessed from the storage device by input interface. The storage device may include any of a variety of distributed or locally accessed data storage media such as a hard drive, Blu-ray discs, DVDs, CD-ROMs, flash memory, volatile or non-volatile memory, or any other suitable digital storage media for storing encoded video data. In a further example, the storage device may correspond to a file server or another intermediate storage device that may store the encoded video generated by source device 12. Destination device 14 may access stored video data from the storage device via streaming or download. The file server may be any type of server capable of storing encoded video data and transmitting that encoded video data to the destination device 14. Example file servers include a web server (e.g., for a website), an FTP server, network attached storage (NAS) devices, or a local disk drive. Destination device 14 may access the encoded video data through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired connection (e.g., 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 encoded video data from the storage device may be a streaming transmission, a download transmission, or a combination thereof.

[0041] The techniques described in this disclosure may be applied to video coding in support of any of a variety of multimedia applications, such as over-the-air television broadcasts, cable television transmissions, satellite television transmissions, Internet streaming video transmissions, such as dynamic adaptive streaming over HTTP (DASH), digital video that is encoded onto a data storage medium, decoding of digital video stored on a data storage medium, or other applications. In some examples, system 10 may be configured to support one-way or two-way video transmission to support applications such as video streaming, video playback, video broadcasting, and/or video telephony.

[0042] Computer-readable medium 16 may include transient media, such as a wireless broadcast or wired network transmission, or storage media (that is, non-transitory storage media), such as a hard disk, flash drive, compact disc, digital video disc, Blu-ray disc, or other computer-readable media. In some examples, a network server (not shown) may receive encoded video data from source device 12 and provide the encoded video data to destination device 14, e.g., via network transmission. Similarly, a computing device of a medium production facility, such as a disc stamping facility, may receive encoded video data from source device 12 and produce a disc containing the encoded video data. Therefore, computer-readable medium 16 may be understood to include one or more computer-readable media of various forms, in various examples.

[0043] Input interface 26 of destination device 14 receives information from computer-readable medium 16. The information of computer-readable medium 16 may include syntax information defined by video encoder 22 of video encoder 22, which is also used by video decoder 30, that includes syntax elements that describe characteristics and/or processing of blocks and other coded units, e.g., groups of pictures (GOPs). Storage media 28 may store encoded video data received by input interface 26. Display device 32 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a cathode ray tube (CRT), a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0044] Video encoder 22 and video decoder 30 each may be implemented as any of a variety of suitable video encoder and/or video 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 the techniques are implemented partially in software, a 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 techniques of this disclosure. Each of video encoder 22 and video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined CODEC in a respective device.

[0045] In some examples, video encoder 22 and video decoder 30 may operate according to a video coding standard. Example video coding standards include, but are not limited to, ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 or ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual and ITU-T H.264 (also known as ISO/IEC MPEG-4 AVC), including its Scalable Video Coding (SVC) and Multi-View Video Coding (MVC) extensions. In addition, a new video coding standard, namely High Efficiency Video Coding (HEVC) or ITU-T H.265, including its range and screen content coding extensions, 3D video coding (3D-HEVC) and multiview extensions (MV-HEVC) and scalable extension (SHVC), has been developed by the Joint Collaboration Team on Video Coding (JCT-VC) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Motion Picture Experts Group (MPEG).

[0046] In other examples, video encoder 22 and video decoder 30 may be configured to operate according to other video coding techniques and/or standards, including new video coding techniques being explored by the Joint Video Exploration Team (JVET). In some examples of the disclosure, video encoder 22 and video decoder 30 may be configured to operate according to video coding techniques that use independent luma and chroma partitioning, such that luma and chroma blocks of video data are not required to be aligned. Such partitioning techniques may lead to the situation where a chroma block is not aligned, within a particular location of a picture, to a single luma block. In other examples of the disclosure, video encoder 22 and video decoder 30 may be configured to operate according to video coding techniques that use partitioning frameworks that allow for non-square blocks.

[0047] In accordance with the techniques of this disclosure, as will be described in more detail below, video decoder 30 may be configured to receive the bitstream of encoded video data, the encoded video data representing partitioned luma blocks and partitioned chroma blocks, wherein the chroma blocks are partitioned independently of the luma blocks, determine a respective coding mode corresponding to the respective partitioned luma blocks, decode the respective partitioned luma blocks according the determined respective coding modes, decode a first syntax element indicating that the respective coding modes associated with the respective partitioned luma blocks are to be used for decoding a first partitioned chroma block, wherein the first partitioned chroma block is aligned with two or more partitioned luma blocks, determine a chroma coding mode for the first partitioned chroma block according to a function of the respective coding modes of the two or more partitioned luma blocks, and decode the first partitioned chroma block in accordance with the determined chroma coding mode. Video encoder 22 may be configured to perform techniques reciprocal to that of video decoder 30. In some examples, video encoder 22 may be configured to generate a syntax element that indicates whether or not a chroma block is to reuse coding mode information from two or more luma blocks based the function of the respective coding modes of the two or more partitioned luma blocks.

[0048] In HEVC and other video coding specifications, a video sequence typically includes a series of pictures. Pictures may also be referred to as “frames.” A picture may include three sample arrays, denoted S_L, S_{Cb}, and S_{Cr}. S_L is a two-dimensional array (e.g., a block) of luma samples. S_{Cb} is a two-dimensional array of Cb chrominance samples. S_{Cr} is a two-dimensional array of Cr chrominance samples. Chrominance samples may also be referred to herein as “chroma” samples. In other instances, a picture may be monochrome and may only include an array of luma samples.

[0049] To generate an encoded representation of a picture (e.g., an encoded video bitstream), video encoder 22 may generate a set of coding tree units (CTUs). Each of the CTUs may comprise a coding tree block of luma samples, two corresponding coding tree blocks of chroma samples, and syntax structures used to code the samples of the coding tree blocks. In monochrome pictures or pictures having three separate color planes, a CTU may comprise a single coding tree block and syntax structures used to code the samples of the coding tree block. A coding tree block may be an NxN block of samples. A CTU may also be referred to as a “tree block” or a “largest coding unit” (LCU). The CTUs of HEVC may be broadly analogous to the macroblocks of other standards, such as H.264/AVC. However, a CTU is not necessarily limited to a particular size and may include one or more coding units (CUs). A slice may include an integer number of CTUs ordered consecutively in a raster scan order.

[0050] To generate a coded CTU, video encoder 22 may recursively perform quadtree partitioning on the coding tree blocks of a CTU to divide the coding tree blocks into coding blocks, hence the name “coding tree units.” A coding block is an NxN block of samples. A CU may comprise a coding block of luma samples and two corresponding coding blocks of chroma samples of a picture that has a luma sample array, a Cb sample array, and a Cr sample array, and syntax structures 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.

[0051] Video encoder 22 may partition a coding block of a CU into one or more prediction blocks. A prediction block is a rectangular (i.e., square or non-square) block of samples on which the same prediction is applied. A prediction unit (PU) of a CU may comprise a prediction block of luma samples, two corresponding prediction blocks of chroma samples, and syntax structures used to predict the prediction blocks. In monochrome pictures or pictures having three separate color planes, a PU may comprise a single prediction block and syntax structures used to predict the prediction block. Video encoder 22 may generate predictive blocks (e.g., luma, Cb, and Cr predictive blocks) for prediction blocks (e.g., luma, Cb, and Cr prediction blocks) of each PU of the CU.

[0052] Video encoder 22 may use intra prediction or inter prediction to generate the predictive blocks for a PU. If video encoder 22 uses intra prediction to generate the predictive blocks of a PU, video encoder 22 may generate the predictive blocks of the PU based on decoded samples of the picture that includes the PU.

[0053] After video encoder 22 generates predictive blocks (e.g., luma, Cb, and Cr predictive blocks) for one or more PUs of a CU, video encoder 22 may generate one or more residual blocks for the CU. As one example, video encoder 22 may generate a luma residual block for the CU. 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. In addition, video encoder 22 may generate a Cb residual block for the CU. In one example of chroma prediction, each sample in the Cb residual block of a CU may indicate 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. Video encoder 22 may also generate a Cr residual block for the CU. 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. However, it should be understood that other techniques for chroma prediction may be used.

[0054] Furthermore, video encoder 22 may use quadtree partitioning to decompose the residual blocks (e.g., the luma, Cb, and Cr residual blocks) of a CU into one or more transform blocks (e.g., luma, Cb, and Cr transform blocks). A transform block is a rectangular (e.g., square or non-square) block of samples on which the same transform is applied. A transform unit (TU) of a CU may comprise a transform block of luma samples, two corresponding transform blocks of chroma samples, and syntax structures used to transform the transform block samples. Thus, each TU of a CU may have a luma transform block, a Cb transform block, and a Cr transform block. The luma transform block of 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.

[0055] Video encoder 22 may apply one or more transforms a transform block of a TU to generate a coefficient block for the TU. For instance, video encoder 22 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. Video encoder 22 may apply one or more transforms to a Cb transform block of a TU to generate a Cb coefficient block for the TU. Video encoder 22 may apply one or more transforms to a Cr transform block of a TU to generate a Cr coefficient block for the TU.

[0056] After generating a coefficient block (e.g., a luma coefficient block, a Cb coefficient block or a Cr coefficient block), video encoder 22 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 video encoder 22 quantizes a coefficient block, video encoder 22 may entropy encode syntax elements indicating the quantized transform coefficients. For example, video encoder 22 may perform context-adaptive binary arithmetic coding (CABAC) on the syntax elements indicating the quantized transform coefficients.

[0057] Video encoder 22 may output a bitstream that includes a sequence of bits that forms a representation of coded pictures and associated data. Thus, the bitstream comprises an encoded representation of video data. The bitstream may comprise a sequence of network abstraction layer (NAL) units. A NAL unit is a syntax structure containing an indication of the type of data in the NAL unit and bytes containing that data in the form of a raw byte sequence payload (RBSP) interspersed as necessary with emulation prevention bits. Each of the NAL units may include a NAL unit header and encapsulates a RBSP. The NAL unit header may include a syntax element indicating a NAL unit type code. The NAL unit type code specified by the NAL unit header of a NAL unit indicates the type of the NAL unit. A RBSP may be a syntax structure containing an integer number of bytes that is encapsulated within a NAL unit. In some instances, an RBSP includes zero bits.

[0058] Video decoder 30 may receive an encoded video bitstream generated by video encoder 22. In addition, video decoder 30 may parse the bitstream to obtain syntax elements from the bitstream. Video decoder 30 may reconstruct the pictures of the video data based at least in part on the syntax elements obtained from the bitstream. The process to reconstruct the video data may be generally reciprocal to the process performed by video encoder 22. For instance, video decoder 30 may use motion vectors of PUs to determine predictive blocks for the PUs of a current CU. In addition, video decoder 30 may inverse quantize coefficient blocks of TUs of the current CU. Video decoder 30 may perform inverse transforms on the coefficient blocks to reconstruct transform blocks of the TUs of the current CU. Video decoder 30 may reconstruct 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. By reconstructing the coding blocks for each CU of a picture, video decoder 30 may reconstruct the picture.

[0059] In some example video codec frameworks, such as the quadtree partitioning framework of HEVC, partitioning of video data into blocks for the color components (e.g., luma blocks and chroma blocks) is performed jointly. That is, in some examples, luma blocks and chroma blocks are partitioned in the same manner such that no more than one luma block corresponds to a chroma block in a particular location within a picture. In one example, a partition of a block of video data may be further divided into sub-blocks. Information (e.g., sample values and syntax elements indicating how the video block is to be coded) relating to the video block or partition of the video block is stored at the sub-block level. Or, more generally, information relating to video blocks or partitions of the video block may be stored with relation to one or more representative locations (e.g., corresponding to any sample(s) or sub-samples(s)) of a block of video data. For example, if a partition is 16x16 pixels, and each sub-block in the partition is 4x4 pixels, then there are 16 sub-blocks in the partition. Information is stored at sub-block granularity, 4x4 in this example, and all 16 sub-blocks may have the same information.

[0060] In the context of this disclosure, the terms "partition," "block," and "partitioned block" may be used interchangeably. In general, a block is a group of samples (e.g., luma or chroma samples) on which video coding is performed. In the context of this disclosure, a "sub-block" is a division of a block having an associated memory location that stores coding mode information for the block.

[0061] Video encoder 22 and video decoder 30 may allocate locations in a memory for storing the information for each representative location (e.g., sub-block). In some examples, the values of the information (e.g., values of particular syntax elements for a particular coding mode) may be stored in a separate memory location associated with each representative location (e.g., sub-block). In other examples, the information may be stored once for one of a plurality of representative locations (e.g., sub-blocks) of a partition. The memory locations of the other sub-blocks of the partition may include pointers to the memory location that stores the actual values of the information. Techniques of this disclosure will be described below with reference to sub-blocks, though it should be understood that any representative location of a block may be used.

[0062] As mentioned above, the information stored at the sub-block level can be any information that is used to perform coding processes on the partition. Such information may be signaled syntax information or derived supplemental information. One example of derived supplemental information may be information used to code chroma blocks that is derived from information related to coding luma blocks. One example of derived supplemental information for use in HEVC is direct mode information, where luma intra prediction information (e.g., intra prediction direction) is used for chroma prediction without signaling the intra prediction direction itself for chroma blocks. Other examples of the information may be mode decision, such as intra prediction or inter prediction, intra prediction direction, motion information, and the like.

[0063] When luma and chroma partition sizes are compared, chroma color format (e.g., chroma sub-sampling format), such as 4:4:4, 4:2:2, 4:2:0, can be taken into the account.

For example, if a luma partition is 16x16 pixels, the corresponding or collocated chroma partition is 8x8 pixels for the 4:2:0 color format, and is 16x16 pixels for the 4:4:4 chroma color format. Partitions are not necessarily square, and can be, for example, rectangular in shape. As such, for a 4:2:0 chroma sub-sampling format, luma and chroma partitions will not be the same size. However, when luma and chroma blocks are partitioned jointly, the resulting partitioning still results in only one luma block corresponding to any particular chroma block.

[0064] A quadtree plus binary tree (QTBT) partition structure is currently being studied by the Joint Video Exploration Team (JVET). In J. An et al., “Block partitioning structure for next generation video coding”, International Telecommunication Union, COM16-C966, Sep. 2015 (hereinafter, “VCEG proposal COM16-C966”), QTBT partitioning techniques were described for future video coding standard beyond HEVC. Simulations have shown that the proposed QTBT structure may be more efficient than the quadtree structure used in HEVC.

[0065] In the QTBT structure described in VCEG proposal COM16-C966, a CTB is first partitioned using quadtree partitioning techniques, where the quadtree splitting of one node can be iterated until the node reaches the minimum allowed quadtree leaf node size. The minimum allowed quadtree leaf node size may be indicated to video decoder 30 by the value of the syntax element MinQTSize. If the quadtree leaf node size is not larger than the maximum allowed binary tree root node size (e.g., as denoted by a syntax element MaxBTSize), the quadtree leaf node can be further partitioned using binary tree partitioning. The binary tree partitioning of one node can be iterated until the node reaches the minimum allowed binary tree leaf node size (e.g., as denoted by a syntax element MinBTSize) or the maximum allowed binary tree depth (e.g., as denoted by a syntax element MaxBTDepth). VCEG proposal COM16-C966 uses the term “CU” to refer to binary-tree leaf nodes. In VCEG proposal COM16-C966, CUs are used for prediction (e.g., intra prediction, inter prediction, etc.) and transform without any further partitioning. In general, according to QTBT techniques, there are two splitting types for binary tree splitting: symmetric horizontal splitting and symmetric vertical splitting. In each case, a block is split by dividing the block down the middle, either horizontally or vertically. This differs from quadtree partitioning, which divides a block into four blocks.

[0066] In one example of the QTBT partitioning structure, the CTU size is set as 128x128 (e.g., a 128x128 luma block and two corresponding 64x64 chroma blocks), the

MinQTSize is set as 16x16, the MaxBTSize is set as 64x64, the MinBTSize (for both width and height) is set as 4, and the MaxBTDepth is set as 4. Quadtree partitioning is applied to the CTU first to generate quadtree leaf nodes. The quadtree leaf nodes may have a size from 16x16 (i.e., the MinQTSize is 16x16) to 128x128 (i.e., the CTU size). According to one example of QTBT partitioning, if the leaf quadtree node is 128x128, the leaf quadtree node cannot be further split by the binary tree, since the size of the leaf quadtree node exceeds the MaxBTSize (i.e., 64x64). Otherwise, the leaf quadtree node is further partitioned by the binary tree. Therefore, the quadtree leaf node is also the root node for the binary tree and has the binary tree depth as 0. The binary tree depth reaching MaxBTDepth (e.g., 4) implies that there is no further splitting. The binary tree node having a width equal to the MinBTSize (e.g., 4) implies that there is no further horizontal splitting. Similarly, the binary tree node having a height equal to MinBTSize implies no further vertical splitting. The leaf nodes of the binary tree (CUs) are further processed (e.g., by performing a prediction process and a transform process) without any further partitioning.

[0067] FIG. 2A illustrates an example of a block 50 (e.g., a CTB) partitioned using QTBT partitioning techniques. As shown in FIG. 2A, using QTBT partition techniques, each of the resultant blocks is split symmetrically through the center of each block. FIG. 2B illustrates the tree structure corresponding to the block partitioning of FIG. 2A. The solid lines in FIG. 2B indicate quadtree splitting and dotted lines indicate binary tree splitting. In one example, in each splitting (i.e., non-leaf) node of the binary tree, a syntax element (e.g., a flag) is signaled to indicate the type of splitting performed (e.g., horizontal or vertical), where 0 indicates horizontal splitting and 1 indicates vertical splitting. For the quadtree splitting, there is no need to indicate the splitting type, as quadtree splitting always splits a block horizontally and vertically into 4 sub-blocks with an equal size.

[0068] As shown in FIG. 2B, at node 70, block 50 is split into the four blocks 51, 52, 53, and 54, shown in FIG. 2A, using quadtree partitioning. Block 54 is not further split, and is therefore a leaf node. At node 72, block 51 is further split into two blocks using binary tree partitioning. As shown in FIG. 2B, node 72 is marked with a 1, indicating vertical splitting. As such, the splitting at node 72 results in block 57 and the block including both blocks 55 and 56. Blocks 55 and 56 are created by a further vertical splitting at node 74. At node 76, block 52 is further split into two blocks 58 and 59 using binary tree partitioning. As shown in FIG. 2B, node 76 is marked with a 1, indicating horizontal splitting.

[0069] At node 78, block 53 is split into 4 equal size blocks using quadtree partitioning. Blocks 63 and 66 are created from this quadtree partitioning and are not further split. At node 80, the upper left block is first split using vertical binary tree splitting resulting in block 60 and a right vertical block. The right vertical block is then split using horizontal binary tree splitting into blocks 61 and 62. The lower right block created from the quadtree splitting at node 78, is split at node 84 using horizontal binary tree splitting into blocks 64 and 65.

[0070] In one example of QTBT partitioning, luma and chroma partitioning may be performed independently of each other for I-slices, contrary, for example, to HEVC, where the quadtree partitioning is performed jointly for luma and chroma blocks. That is, in some examples being studied, luma blocks and chroma blocks may be partitioned separately such that luma blocks and chroma blocks do not directly overlap. As such, in some examples of QTBT partitioning, chroma blocks may be partitioned in a manner such that at least one partitioned chroma block is not spatially aligned with a single partitioned luma block. That is, the luma samples that are co-located with a particular chroma block may be within two or more different luma partitions.

[0071] As described above, in some examples, information relating to how a chroma block is to be coded can be derived from information relating to a corresponding luma block. However if luma and chroma partitioning is performed independently, the luma and chroma blocks may not be aligned (e.g., the luma and chroma blocks may not correspond to the same set of pixels). For example, chroma partitioning can be such that the chroma blocks are larger or smaller than a corresponding luma partition. In addition, chroma blocks may spatially overlap two or more luma blocks. As explained above, if a partitioned chroma block is larger than a partitioned luma block, it can be the case than there is more than one luma block that spatially corresponds to a particular chroma block, and thus more than one set of luma information (e.g., syntax elements and the like) associated with the luma partitions corresponding to the size of the chroma partition. In such cases, it is unclear how to derive chroma information from luma information. It should be understood that such a situation may arise with any partitioning structure where luma and chroma blocks are partitioned independently, and not just with example QTBT partitioning structures being studied by the JVET.

[0072] In view of these drawbacks, this disclosure describes methods and devices for deriving chroma information from luma information for pictures partitioned using separate and/or independent luma and chroma partitioning. As described above, luma and chroma partitioning can be misaligned, e.g., being of different sizes or shapes. Derived information (e.g., determined coding mode information) from a luma block can be used as a predictor for chroma information (e.g., the coding mode to be used for a chroma block) or be used to code a chroma block of video data. Alternatively or additionally, luma information can be used for context modelling in the context coding of the chroma information. Optionally, context modelling can be combined with the prediction information. It should be understood that each of the techniques described below may be used independently or may be combined with the other techniques in any combination.

[0073] As one example of the disclosure, video encoder 22 encodes a luma block of video data using a particular coding mode (e.g., a particular intra prediction mode, a particular inter prediction mode, a particular filtering mode, a particular motion vector prediction mode, etc.). In some examples, video encoder 22 may further encode syntax elements that indicate what coding mode(s) were used to encode a particular luma block. Video decoder 30 may be configured to decode the syntax elements to determine the coding mode(s) to use to decode the luma block. In other examples, video decoder 30 may not receive syntax elements that explicitly indicate a particular coding mode. Rather, video decoder 30 may be configured to derive a particular coding mode for a luma block based on various video characteristics (e.g., block size, information from neighboring blocks, etc.) and a set of predetermined rules. In other examples, video decoder 30 may determine coding modes based on a combination of explicitly signaled syntax elements and predetermined rules.

[0074] In one example, video encoder 22 may optionally encode chroma blocks (e.g., a Cr block and/or a Cb block) using the same coding mode as a corresponding luma block. Rather than video encoder 22 simply signaling syntax elements and/or other information indicating the coding mode for the chroma blocks, video encoder 22 may signal a syntax element (e.g., a flag) that indicates to video decoder 30 to reuse any signaled or derived information for determining the coding mode(s) for the luma block as a predictor for the coding mode(s) of one or more corresponding chroma blocks. For example, a flag may be coded for one or more chroma blocks to indicate whether the chroma blocks are coded with the same coding mode as a corresponding luma block. If not, than video encoder 22 generates syntax elements indicating the coding mode for the chroma block independently, where video encoder 22 can take into account that the chroma mode is not equal to the luma mode. That is, video encoder 22 and video decoder may be able to determine that the coding mode for the chroma block is not the same as the coding mode for the luma block, and therefore, the coding mode for the luma block can be excluded as a possibility for the chroma block. In a further example, a separate context can be used to code the flag that indicates whether chroma component is coded using the same mode as the luma component.

[0075] In the example describe above, it is assumed video encoder 22 and video decoder 30 code the luma block first, followed by coding the one or more chroma blocks. In this example, luma information is already available when a chroma block is being coded. If video encoder 22 and video decoder 30 are configured to code the blocks in another order (e.g., a chroma block is coded first), then luma and chroma terms can be simply be swapped in the following examples.

[0076] In one example of the disclosure, video decoder 30 may be configured to receive a bitstream of encoded video data and store the encoded video data in a memory (e.g., storage media 28 of FIG. 1). The encoded video data may represent both partitioned luma blocks and partitioned chroma blocks. In some examples, the partitioned chroma blocks may include both Cr chroma blocks and Cb chroma blocks. As used in the disclosure, the term “chroma block” may refer to any type of block that includes any type of chroma information. In the examples of this disclosure, the chroma blocks are partitioned independently of the luma blocks. That is, video encoder 22 may be configured to encode the video data using a separate partitioning structure for luma blocks and chroma blocks.

[0077] Such a separate partitioning structure may result in at least one partitioned chroma block not being aligned with a single partitioned luma block. As such, for a particular spatial location of a picture, video encoder 22 may partition a single chroma block, but partition multiple luma blocks. However, it should be understood that for other spatial locations of the picture, there may be a 1 to 1 correspondence between luma and chroma blocks, or there may be multiple chroma blocks for a single luma block. The QTBT partitioning structure described above is a type of partitioning structure where luma and chroma blocks are partitioned independently/separately. However, the techniques of this disclosure may be applied to video data partitioned according to any partitioning structure where luma and chroma blocks are partitioned independently.

[0078] Video decoder 30 may be further configured to determine a coding mode for the respective partitioned luma blocks received in the encoded video bitstream and decode the respective partitioned luma blocks according the determined respective coding modes. Video decoder 30 may be configured to determine the coding mode from information indicated by syntax elements received in the encoded video bitstream. Such syntax elements may indicate the coding modes explicitly. In other examples, video decoder 30 may be configured to implicitly determine the coding mode for the luma blocks from characteristics of the video data and some predetermined rules. In other examples, video decoder 30 may determine the coding modes for the luma blocks using a combination of explicitly signaled syntax elements and implicitly determined coding modes from predetermined rules and video data characteristics.

[0079] In the context of this disclosure, the coding modes can be any information that indicates to video decoder 30 how video encoder 22 encoded the encoded video data, and how video decoder 30 should decode the video data. Example coding modes may include direct mode for chroma intra prediction, a position-dependent intra prediction combination (PDPC) flag (e.g., indicating if a PDPC mode is used), PDPC parameters, secondary transform sets for a non-separable secondary transforms (NSST), enhanced multiple transform (EMT), adaptive multiple transforms (AMT), and contexts for selecting entropy coding data models. The above are examples of chroma coding modes that can be derived from the luma coding modes (coding modes used to code luma blocks) determined by video decoder 30, and which have been used in the JEM test model studied in JVET. However, it should be understood that the coding modes may include any coding mode used for coding the luma blocks that may be reused for coding chroma blocks or used to predict the coding mode for a chroma block.

[0080] Regardless of the type of coding mode, or the manner in which the coding mode was determined, video decoder 30 may be configured to store the determined coding mode for a particular partitioned luma blocks in a plurality of different memory locations associated with the particular partitioned luma block. As will be explained in more detail below with reference to FIG. 3, a particular partitioned luma block may be divided into sub-blocks, and video decoder 30 may store the coding mode determined for the entire particular partitioned luma block in memory locations corresponding to each of the sub-blocks. Accordingly, for a particular partitioned luma block divided into N sub-blocks, the coding mode is stored in N different memory locations, each memory location corresponding to a particular spatially located sub-block within the partitioned luma block. Sub-blocks can be a rectangular or square block of any size. In some examples, a sub-block may be just one sample, i.e., a block of size of 1x1. In some examples, each memory location may store data that explicitly indicates the coding mode for the particular partitioned luma block. In other examples, one or more memory location associated with the particular partitioned luma block explicitly stores information indicating the coding mode, while the other memory locations associated with the particular partitioned luma block store pointers to the memory location(s) that explicitly stores the coding mode.

[0081] According to some examples of the disclosure, video decoder 30 may be configured to reuse coding mode information received for a luma block for use when decoding a chroma block. In some examples, video decoder 30 may receive and decode a first syntax element indicating if coding modes associated with the respective partitioned luma blocks are to be used for decoding a particular partitioned chroma block. As described above, a partitioned chroma block may be spatially aligned with two or more different partitioned luma blocks. As such, it may be difficult to determine how to reuse luma coding modes for such a chroma block, as it is unclear from which luma block to inherit (e.g., reuse) the coding mode information. By storing the luma coding mode information in multiple memory locations corresponding to multiple sub-block locations, video decoder 30 may be configured to determine which coding mode information from the luma block to reuse for a particular chroma block using a function of the coding mode information stored for sub-blocks that correspond to the spatial location of the particular chroma block. In this context, the function may be a set of predetermined rules and analysis techniques that video decoder 30 uses to determine which coding mode(s) of the two or more co-located partitioned luma blocks to reuse for the partitioned chroma block. Once the luma coding mode to reuse is determined, video decoder 30 may decode the particular partitioned chroma block with the determined coding mode.

[0082] Video decoder 30 may be configured to determine which coding mode to reuse from two or more spatially aligned luma blocks based on a function of coding mode information stored in memory locations associated with sub-blocks of the partitioned luma blocks. Several different functions may be used. In one example of the disclosure, video decoder 30 may be configured to perform a statistical analysis of the luma coding mode information corresponding to the two or more luma partitions that are collocated with chroma partition. The determined luma coding mode information from the luma block for use by the chroma block may be a function of the whole luma coding mode information (e.g., the coding mode information contained in every luma sub-block that is co-located with the chroma block) and how luma coding mode information varies (e.g., to what extent the information is similar or different) across the co-located sub-blocks in the luma block. One difference from the previous techniques for coding mode reuse and the techniques of this disclosure, is that the luma coding mode information can be related to more than one luma block (or partition) due to the separate luma and chroma partitioning.

[0083] Examples of a function that may be used for determining what luma coding mode information from two or more partitioned luma blocks can be reused for a co-located chroma block may include, but is not limited to, one or more of the functions described below. In one example, video decoder 30 may perform a function that includes a statistical analysis of the coding mode information indicating the determined respective coding mode of two or more partitioned luma blocks. The coding mode information is stored in respective memory locations associated with the respective sub-blocks of the two or more partitioned luma blocks. In one example, video decoder 30 may analyze the coding mode information and return (e.g., to determine to use and obtain from memory) the coding mode information that appears most often for the co-located sub-blocks of the two or more partitioned luma blocks. That is, the function returns the majority of the luma information used in the corresponding luma blocks. In this manner, the function indicates that particular luma coding mode information, from two or more partitioned luma blocks, to be reused for a co-located chroma block.

[0084] In other examples, video decoder 30 may use a function performs an analysis of the coding mode information for the sub-blocks of the two or more partitioned luma blocks, the analysis includes measuring the gradient or higher derivatives of the luma coding mode information to measure the smoothness of that information. For example, some extreme (e.g., outlier) coding mode information in the two or more luma blocks, which may be much different from the majority of the coding mode information related to the two or more luma blocks, can be ignored and not reused for the chroma blocks. In other examples, video decoder 30 may assign different weights to the luma coding mode information stored for respective sub-blocks, and use a weighted average of coding modes to determine which mode to reuse for the chroma block. The weights may be assigned based on the relative locations of the luma sub-blocks that are co-located with the chroma block.

[0085] In other examples, the functions used to determine what luma coding mode(s) to reuse for a chroma block may include one or more of the following video coding characteristics. The functions video decoder 30 may use to determine what luma coding mode(s) to reuse for a chroma block may include the shape of the block (rectangular, square), a block orientation (vertical or horizontal oriented rectangular block), a shape of the chroma block, a shape of the luma block containing representative location, the width or height of the luma and/or chroma blocks, more frequently used luma mode of the area corresponding to the chroma block. In other examples, the function may be based on the prediction mode or luma information in the representative location. For example, if video decoder 30 is configured to reuse a luma intra mode for a chroma block, but the luma block in the representative location is coded with an inter mode, video decoder 30 may configured to select another representative location to determine the luma intra mode to reuse for the chroma block. More generally, if video decoder 30 is determine what luma information to reuse to code a chroma block, but the luma information may not be valid for this chroma block, video decoder 30 may consider luma information from another representative location in the luma block, or otherwise may use some default luma information.

[0086] In another example of the disclosure, video decoder 30 may reuse coding mode information based on a predetermined sub-block location. For example, video decoder 30 may simply reuse the coding mode information stored for a particular sub-block location that is co-located with the chroma block. As one example, video decoder 30 may reuse coding mode information stored at a luma sub-block that is co-located with a particular corner of the partitioned chroma block. Any corner sub-block may be used. As another example, video decoder 30 may reuse coding mode information stored at a luma sub-block that is co-located with a center of the partitioned chroma block. In other examples, video decoder 30 may perform a statistical analysis (e.g., as described above) of the coding mode information stored for some predetermined number of luma sub-block locations. That is, video decoder 30 may use a function that analyzes certain sub-blocks of a luma partition and derives chroma information based on the luma information contained therein.

[0087] In another example, video decoder 30 may divide the partitioned chroma block into multiples of sub-blocks, e.g., 1x1, 2x2, 4x4, or other sized sub-blocks. Then, for each sub-block, video decoder 30 may inherit (e.g., reuse) the luma coding mode information stored for a particular luma sub-block for a co-located chroma sub-block. In this way, different coding information from the corresponding luma block can be applied in a single chroma block.

[0088] As discussed above, video decoder 30 may use one of a plurality of predetermined functions for determining how to reuse luma coding mode information for chroma blocks. In some examples, video decoder 30 may be configured to use a single predetermined function, and use that function for all pictures. In other examples, video decoder 30 may determine which function to use based on some video coding characteristics. In other examples, video encoder 22 may be configured to signal a syntax element that indicates to video decoder 30 what function to use to determine how to reuse luma coding mode information. Such a syntax element may be signaled at any level, e.g., a sequence level, a picture level, a slice level, a tile level, a CTB level, etc.

[0089] FIG. 3 is a conceptual diagram illustrating an example of luma and chroma relative partitioning in accordance with the techniques of this disclosure. As shown in FIG. 3, information is stored per partition (e.g., partitioned luma blocks) in sub-blocks of the partitions (dashed boxes). The sub-blocks may be of any size, down to the size of an individual sample. FIG. 3 shows that one chroma partition, whether in a 4:4:4 or 4:2:0 sub-sampling format, may have more than one associated luma partition. As such, a single chroma partition may have more than one corresponding set of luma information. Certain representative locations (e.g., sub-blocks) of the luma partitions may be used to analyze the luma information (e.g., coding modes) so as to derive the chroma information for the corresponding chroma partition, as described above.

[0090] FIG. 4 is a conceptual diagram illustrating another example of luma and chroma relative partitioning in accordance with the techniques of this disclosure. As shown in FIG. 4, information is stored per partition in sub-blocks (dashed boxes) of the luma partition. FIG. 4 shows that each sub-block in one chroma partition may have one associated luma sub-block, and the luma information of the associated one luma sub-block can be analyzed to derive the chroma information for the corresponding chroma sub-block.

[0091] The following section describes some examples that may use the techniques of the disclosure. In chroma direct mode, the luma intra direction is used for chroma intra prediction. An example of this mode was used in HEVC. In accordance with one example technique of this disclosure, when the chroma and luma structure is not aligned (e.g., due to independent chroma and luma partitioning), the center representative luma sub-block is selected to obtain the luma intra prediction mode, which is then applied for the chroma partition as the direct mode. Other luma sub-blocks of the corresponding luma partition may have other intra directions different from the selected sub-block. Other functions instead of using the center representative sub-block, as explained above, may also be used.

[0092] In another example, in chroma direct mode, when the chroma and luma structure is not aligned, the chroma intra prediction is performed in 2x2 (or 4x4) sub-block units. For each 2x2 (or 4x4) chroma sub-block, one associated 4x4 luma sub-block is identified, and the intra prediction mode of this identified 4x4 luma sub-block is applied for the current chroma 2x2 (or 4x4) sub-block.

[0093] In another example, a chroma PDPC control flag (i.e., PDPC mode is applied or not) and PDPC parameters are derived, for example, from the center representative luma sub-block, and applied for the chroma partition. Other functions instead of using the center representative sub-block, as explained above, may also be used.

[0094] In another example, a secondary transform (NSST) set is selected by video decoder 30 from the center representative luma sub-block and is applied for the chroma partition. Other functions instead of using the center representative sub-block, as explained above, may also be used.

[0095] Similar techniques may be applied by video decoder 30 to any chroma information that is derived from luma information.

[0096] The foregoing examples were described with reference to video decoder 30. However, video encoder 22 may employ the same techniques for determining how to reuse information generated, derived, and/or signaled for luma blocks for chroma blocks. In particular, video encoder 22 may determine whether or not to signal a syntax element that indicates whether or not luma coding mode information should be reused for chroma blocks based on the function the video decoder will use to determine what luma coding mode information to reuse.

[0097] The following sections describes techniques for determining parameters for a position-dependent intra prediction combination (PDPC) coding mode for blocks of video data that may be partitioned into non-square, rectangular partitions. The QTBT partitioning structure described above is an example of a partitioning structure that allows for non-square, rectangular blocks. However, the techniques of this disclosure may be used with any partitioning structure that produces non-square, rectangular blocks.

[0098] When coding video data using the PDPC coding mode, video encoder 22 and/or video decoder 30 may use one or more parameterized equations that define how to combine predictions based on filtered and unfiltered reference values and based on the position of the predicted pixel. The present disclosure describes several sets of parameters, such that video encoder 22 may be configured to test the sets of parameters (via, e.g., using rate-distortion analysis) and signal to video decoder 30 the optimal parameters (e.g., the parameters resulting in the best rate-distortion performance among those parameters that are tested). In other examples, video decoder 30 may be configured to determine PDPC parameters from characteristics of the video data (e.g., block size, block height, block width, etc.).

[0099] FIG. 5A illustrates a prediction of a 4x4 block (p) using an unfiltered reference (r) according to techniques of the present disclosure. FIG. 5B illustrates a prediction of a 4x4 block (q) using a filtered reference (s) according to techniques of the present disclosure. While both FIGS. 5A and 5B illustrate a 4x4 pixel block and 17 (4x4+1) respective reference values, the techniques of the present disclosure may be applied to any block size and number of reference values.

[0100] Video encoder 22 and/or video decoder 30, when performing the PDPC coding mode, may utilize a combination between the filtered (q) and unfiltered (p) predictions, such that a predicted block for a current block to be coded can be computed using pixel values from both the filtered (s) and unfiltered (r) reference arrays.

[0101] In one example of the techniques of PDPC, given any two set of pixel predictions $p_{\mathbf{r}}[x, y]$ and $q_{\mathbf{s}}[x, y]$, computed using only the unfiltered and filtered references **r** and **s**, respectively, the combined predicted value of a pixel, denoted by $v[x, y]$, is defined by

$$v[x, y] = c[x, y]\, p_{\mathbf{r}}[x, y] + (1 - c[x, y])\, q_{\mathbf{s}}[x, y] \qquad (1)$$

where $c[x, y]$ is the set of combination parameters. The value of the weight $c[x, y]$ may be a value between 0 and 1. The sum of the weights $c[x, y]$ and $(1 - c[x, y])$ may be equal to one.

[0102] In certain examples it may not be practical to have a set of parameters as large as the number of pixels in the block. In such examples $c[x, y]$ may be defined by a much smaller set of parameters, plus an equation to compute all combination values from those parameters. In such an example the following formula may be used:

$$v[x,y] = \left\lfloor \frac{c_1^{(v)} r[x,-1] - c_2^{(v)} r[-1,-1]}{2^{\left\lfloor \frac{y}{d_v} \right\rfloor}} \right\rfloor + \left\lfloor \frac{c_1^{(h)} r[-1,y] - c_2^{(h)} r[-1,-1]}{2^{\left\lfloor \frac{x}{d_h} \right\rfloor}} \right\rfloor + \left(\frac{N - \min(x,y)}{N} \right) g \; p_{\mathbf{r}}^{(\mathrm{HEVC})}[x,y] + b[x,y] \, q_{\mathbf{s}}^{(\mathrm{HEVC})}[x,y] \quad (2)$$

where $c_1^v, c_2^v, c_1^h, c_2^h, g,$ and $d_v, d_h \in \{1,2\}$, are prediction parameters, N is the block size, $p_{\mathbf{r}}\ [x,y]$ and $q_{\mathbf{s}}\ [x,y]$ are prediction values computed using the according to the HEVC standard, for the specific mode, using respectively the nonfiltered and filtered references, and

$$b[x,y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor - \left(\frac{N - \min(x,y)}{N} \right) g \quad (3)$$

is a normalization factor (i.e., to make the overall weights assigned to $p_{\mathbf{r}}^{(\mathrm{HEVC})}[x,y]$ and $q_{\mathbf{s}}^{(\mathrm{HEVC})}[x,y]$ add to 1), defined by the prediction parameters.

[0103] Formula 2 may be generalized for any video coding standard in formula 2A:

$$v[x,y] = \left\lfloor \frac{c_1^{(v)} r[x,-1] - c_2^{(v)} r[-1,-1]}{2^{\left\lfloor \frac{y}{d_v} \right\rfloor}} \right\rfloor + \left\lfloor \frac{c_1^{(h)} r[-1,y] - c_2^{(h)} r[-1,-1]}{2^{\left\lfloor \frac{x}{d_h} \right\rfloor}} \right\rfloor + \left(\frac{N - \min(x,y)}{N} \right) g \; p_{\mathbf{r}}^{(\mathrm{STD})}[x,y] + b[x,y] \, q_{\mathbf{s}}^{(\mathrm{STD})}[x,y] \quad (2A)$$

where $c_1^v, c_2^v, c_1^h, c_2^h, g,$ and $d_v, d_h \in \{1,2\}$, are prediction parameters, N is the block size, $p_{\mathbf{r}}^{(\mathrm{STD})}[x,y]$ and $q_{\mathbf{s}}^{(\mathrm{STD})}[x,y]$ are prediction values computed using the according to a video coding standard (or video coding scheme or algorithm), for the specific mode, using respectively the nonfiltered and filtered references, and

$$b[x,y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor - \left(\frac{N - \min(x,y)}{N} \right) g \quad (3A)$$

is a normalization factor (i.e., to make the overall weights assigned to $p_{\mathbf{r}}^{(\mathrm{STD})}[x,y]$ and $q_{\mathbf{s}}^{(STD)}[x,y]$ add to 1), defined by the prediction parameters.

[0104] These prediction parameters may include weights to provide an optimal linear combination of the predicted terms according to the type of prediction mode used (e.g., DC, planar, and 33 directional modes of HEVC). For example, HEVC contains 35 prediction modes. A lookup table may be constructed with values for each of the prediction parameters $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v,$ and $d_h$for each of the prediction modes (i.e., 35 values of $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v,$ and $d_h$for each prediction mode). Such values may be encoded in a bitstream with the video or may be constant values known by the encoder and decoder ahead of time and may not need to be transmitted in a file or bitstream. The values for $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v,$ and $d_h$may be determined by an optimization training algorithm by finding the values for the prediction parameters that give best compression for a set of training videos.

[0105] In another example, there are a plurality of predefined prediction parameter sets for each prediction mode (in e.g. a lookup table) and the prediction parameter set selected (but not the parameters themselves) is transmitted to a decoder in an encoded file or bitstream. In another example the values for $c_1^v, c_2^v, c_1^h, c_2^h, g, d_v,$ and d_h may be generated on the fly by a video encoder and transmitted to a decoder in an encoded file or bitstream.

[0106] In another example, instead of using HEVC prediction, a video coding device performing these techniques may use a modified version of HEVC, like one that uses 65 directional predictions instead of 33 directional predictions. In fact, any type of intra-frame prediction can be used.

[0107] In another example, the formula can be chosen to facilitate computations. For example, we can use the following type of predictor

$$v[x,y] = \left\lfloor \frac{c_1^{(v)} r[x,-1] - c_2^{(v)} r[-1,-1]}{2^{\lfloor y/d_v \rfloor}} \right\rfloor + \left\lfloor \frac{c_1^{(h)} r[-1,y] - c_2^{(h)} r[-1,-1]}{2^{\lfloor x/d_h \rfloor}} \right\rfloor + b[x,y]\, p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{HEVC})}[x,y] \quad (4)$$

where

$$b[x,y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor \quad (5)$$

and

$$p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{HEVC})}[x,y] = a\, p_{\mathbf{r}}^{(\mathrm{HEVC})}[x,y] + (1-a)\, q_{\mathbf{s}}^{(\mathrm{HEVC})}[x,y]. \quad (6)$$

[0108] Such an approach may exploit the linearity of the HEVC (or other) prediction. Defining **h** as the impulse response of a filter *k* from a predefined set, if we have

$$\mathbf{s} = a\,\mathbf{r} + (1-a)(\mathbf{h} * \mathbf{r}) \quad (7)$$

where "*" represents convolution, then

$$p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{HEVC})}[x,y] = p_{\mathbf{s}}^{(\mathrm{HEVC})}[x,y] \quad (8)$$

i.e., the linearly combined prediction may be computed from the linearly combined reference.

[0109] Formulas 4, 6 and 8 may be may be generalized for any video coding standard in formula 4A, 6A, and 8A:

$$v[x,y] = \left\lfloor \frac{c_1^{(v)} r[x,-1] - c_2^{(v)} r[-1,-1]}{2^{\lfloor y/d_v \rfloor}} \right\rfloor + \left\lfloor \frac{c_1^{(h)} r[-1,y] - c_2^{(h)} r[-1,-1]}{2^{\lfloor x/d_h \rfloor}} \right\rfloor + b[x,y]\, p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{STD})}[x,y] \quad (4A)$$

where

$$b[x,y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor \quad (5A)$$

and

$$p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{STD})}[x,y] = a\, p_{\mathbf{r}}^{(\mathrm{STD})}[x,y] + (1-a)\, q_{\mathbf{s}}^{(\mathrm{STD})}[x,y]. \quad (6A)$$

Such an approach may exploit the linearity of the prediction of the coding standard. Defining **h** as the impulse response of a filter k from a predefined set, if we have

$$\mathbf{s} = a\,\mathbf{r} + (1-a)(\mathbf{h} * \mathbf{r}) \quad (7A)$$

where "*" represents convolution, then

$$p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{STD})}[x,y] = p_{\mathbf{s}}^{(\mathrm{STD})}[x,y] \quad (8A)$$

i.e., the linearly combined prediction may be computed from the linearly combined reference.

[0110] In an example, prediction functions may use the reference vector (e.g., **r** and **s**) only as input. In this example, the behavior of the reference vector does not change if the reference has been filtered or not filtered. If **r** and **s** are equal (e.g., some unfiltered reference **r** happens to be the same as another filtered reference **s**) then predictive functions, e.g. $p_{\mathbf{r}}[x,y]$ (also written as p(x,y,**r**)) is equal to $p_{\mathbf{s}}[x,y]$ (also written as p(x,y,**s**))), applied to filtered and unfiltered references are equal. Additionally, pixel predictions p and q may be equivalent (e.g., produce the same output given the same input). In such an example, formulas (1)-(8) may be rewritten with pixel prediction $p\,[x,y]$ replacing pixel prediction $q\,[x,y]$.

[0111] In another example, the prediction (e.g., the sets of functions) may change depending on the information that a reference has been filtered. In this example, different sets of functions can be denoted (e.g., $p_{\mathbf{r}}[x,y]$ and $q_{\mathbf{s}}[x,y]$). In this case, even if **r** and **s** are equal, $p_{\mathbf{r}}[x,y]$ and $q_{\mathbf{s}}[x,y]$ may not be equal. In other words, the same input can create different output depending on whether the input has been filtered or not. In such an example, $p\,[x,y]$ may not be able to be replaced by $q\,[x,y]$.

[0112] An advantage of the prediction equations shown is that, with the parameterized formulation, sets of optimal parameters can be determined (i.e., those that optimize the prediction accuracy), for different types of video textures, using techniques such as training. This approach, in turn, may be extended in some examples by computing several sets of predictor parameters, for some typical types of textures, and having a compression scheme where the encoder test predictors from each set, and encodes as side information the one that yields best compression.

[0113] In some example of the techniques described above, when the PDPC coding mode is enabled, PDPC parameters used for intra prediction weighting and controlling, for example, using filtered or unfiltered samples, of PDPC mode are precomputed and stored in a look up table (LUT). In one example, video decoder 30 determines the PDPC parameters according to the block size and intra prediction direction. Previous techniques for PDPC coding mode assumed that intra predicted blocks are always square in size.

[0114] The JVET test model includes the PDPC coding mode. As discussed above, the JVET test model uses QTBT partitioning, which allows for non-square rectangular blocks. The following section discusses example techniques for the extension of PDPC coding for rectangular blocks. However, it should be understood that the techniques of this disclosure may be used for determining prediction mode parameters for any prediction mode that uses non-square blocks, including prediction modes that apply a weighted average to the predictor and reference samples according to sample position.

[0115] It is suggested to modify the structure of the LUT used for determining PDPC parameters, or the techniques used for deriving parameters from the LUT, in a way that for horizontal-related parameters, the width of the block is used to store or access PDPC parameters from the LUT, and for vertical-related parameters, the height of the block is used to store or access PDPC parameters from the LUT. For other parameters that do not have horizontal or vertical relation, a function of the block width and height can be applied to store or access those parameters from the LUT.

[0116] Video decoder 30 may receive a block of video data that has been encoded using the PDPC mode. In this example, the block of video data may have a non-square, rectangular shape defined by a width and a height. Video decoder 30 may determine horizontally-related PDPC parameters as a function of the intra prediction mode and the width of the block. Video decoder 30 may determine vertically-related PDPC parameters as a function of the intra prediction mode and the height of the block. In addition, video decoder 30 may determine non-directional PDPC parameters (e.g., PDPC parameters that are neither horizontally-related nor vertically-related) based on the intra prediction mode and a function of the height and width of the block. Example vertically-related parameters are indicated above with a superscript v. Example horizontally-related parameters are indicated above with a superscript h. The function, for example can be, but is not limited to, the max or min of width and height of the block, or a weighted average of the height and width of the block, where the weighting can be dependent on how one dimension is larger than another dimension of the block.

For example, the larger dimension (width or height) can have a larger weight than the other dimension in the weighted average.

[0117] Since there are only a certain number of block shapes (width and height) that are allowed, this function can also be represented explicitly for all possible block widths and heights, or sub-combinations. For example, if we have N_w and N_h possible block widths and heights, tables of size $N_w \times N_h$ can store the data to be used in the intra prediction process, for each rectangular or square block.

[0118] FIG. 6 is a conceptual diagram illustrating the use of nested tables for determining a set of prediction parameters used in a rectangular block in accordance with one example of the disclosure. Video decoder 30 may determine the PDPC parameters using one or more LUTs whose entries are indexed on both the width and height of a block. As shown in FIG. 6, the width (W) and/or height (H) may be used as inputs to size-to-parameter table 90. Size-to-parameter table 90 may be configured as a (LUT) that contains indices that point to entries in prediction parameter table 92. As discussed above, size-to-parameter table 90 may be of size $N_w \times N_h$ to account for N_w and N_h possible block widths and heights. In this example, size-to-parameter table 90 may be used for a single intra prediction mode (e.g., DC, planar, or other prediction directions). In other examples, size-to-parameter table 90 may contain entries for all intra prediction modes and use block height, block width, and intra prediction mode as entries in to the table. In general, to minimize memory, video decoder 30 and video encoder 22 may be configured to combine table entries in rows and columns, reducing the size of the tables (e.g., size-to-parameter table 90), and possibly creating several tables, of different sizes.

[0119] As one example, assuming a particular intra prediction mode, video decoder 30 may use the width of the block of video data being decoded to access an entry in size-to-parameter table 90 to determine one or more horizontally-related PDPC parameters. Based on the width of the block, the corresponding entry in size-to-parameter table 90 may be used as an input to prediction parameter table 92. Prediction parameter table 92 is of size N_p x N_e and contains entries of the actual PDPC parameters. As such, the entry obtained from size-to-parameter table 90 is an index that points to the actual horizontally-related PDPC parameter in prediction parameter table 92 that is then used in decoding block 94.

[0120] Likewise, video decoder 30 may use the height of the block of video data being decoded to access an entry in size-to-parameter table 90 to determine one or more vertically-related PDPC parameters. Based on the width of the block, the corresponding entry in size-to-parameter table 90 may be used as an input to prediction parameter table 92 to obtain the actual vertically-related PDPC parameter in prediction parameter table 92 that is then used in decoding block 94. The same process may be applied for non-directional PDPC parameters that are index based on a function of the height and width of the block.

[0121] FIG. 7 is a block diagram illustrating an example video encoder 22 that may implement the techniques of this disclosure. FIG. 7 is provided for purposes of explanation and should not be considered limiting of the techniques as broadly exemplified and described in this disclosure. The techniques of this disclosure may be applicable to various coding standards or methods.

[0122] In the example of FIG. 7, video encoder 22 includes a prediction processing unit 100, video data memory 101, a residual generation unit 102, a transform processing unit 104, a quantization unit 106, an inverse quantization unit 108, an inverse transform processing unit 110, a reconstruction unit 112, a filter unit 114, a decoded picture buffer 116, and an entropy encoding unit 118. Prediction processing unit 100 includes an inter prediction processing unit 120 and an intra prediction processing unit 126. Inter prediction processing unit 120 may include a motion estimation unit and a motion compensation unit (not shown).

[0123] Video data memory 101 may be configured to store video data to be encoded by the components of video encoder 22. The video data stored in video data memory 101 may be obtained, for example, from video source 18. Decoded picture buffer 116 may be a reference picture memory that stores reference video data for use in encoding video data by video encoder 22, e.g., in intra- or inter-coding modes. Video data memory 101 and decoded picture buffer 116 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM), magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 101 and decoded picture buffer 116 may be provided by the same memory device or separate memory devices. In various examples, video data memory 101 may be on-chip with other components of video encoder 22, or off-chip relative to those components. Video data memory 101 may be the same as or part of storage media 20 of FIG. 1.

[0124] Video encoder 22 receives video data. Video encoder 22 may encode each CTU in a slice of a picture of the video data. Each of the CTUs may be associated with equally-sized luma coding tree blocks (CTBs) and corresponding CTBs of the picture. As part of encoding a CTU, prediction processing unit 100 may perform partitioning to divide the CTBs of the CTU into progressively-smaller blocks. In some examples, video encoder 22 may partition blocks using a QTBT structure. The smaller blocks may be coding blocks of CUs. For example, prediction processing unit 100 may partition a CTB associated with a CTU according to a tree structure. In accordance with one or more techniques of this disclosure, for each respective non-leaf node of the tree structure at each depth level of the tree structure, there are a plurality of allowed splitting patterns for the respective non-leaf node and the video block corresponding to the respective non-leaf node is partitioned into video blocks corresponding to the child nodes of the respective non-leaf node according to one of the plurality of allowable splitting patterns.

[0125] Video encoder 22 may encode CUs of a CTU to generate encoded representations of the CUs (i.e., coded CUs). As part of encoding a CU, prediction processing unit 100 may partition the coding blocks associated with the CU among one or more PUs of the CU. Thus, each PU may be associated with a luma prediction block and corresponding chroma prediction blocks. Video encoder 22 and video decoder 30 may support PUs having various sizes. As indicated above, the size of a CU may refer to the size of the luma coding block of the CU and the size of a PU may refer to the size of a luma prediction block of the PU. Assuming that the size of a particular CU is 2Nx2N, video encoder 22 and video decoder 30 may support PU sizes of 2Nx2N or NxN for intra prediction, and symmetric PU sizes of 2Nx2N, 2NxN, Nx2N, NxN, or similar for inter prediction. Video encoder 22 and video decoder 30 may also support asymmetric partitioning for PU sizes of 2NxnU, 2NxnD, nLx2N, and nRx2N for inter prediction.

[0126] Inter prediction processing unit 120 may generate predictive data for a PU by performing inter prediction on each PU of a CU. The predictive data for the PU may include predictive blocks of the PU and motion information for the PU. Inter prediction processing unit 120 may perform different operations for a PU of a CU depending on whether the PU is in an I slice, a P slice, or a B slice. In an I slice, all PUs are intra predicted. Hence, if the PU is in an I slice, inter prediction processing unit 120 does not perform inter prediction on the PU. Thus, for blocks encoded in I-mode, the predicted block is formed using spatial prediction from previously-encoded neighboring blocks within the same frame. If a PU is in a P slice, inter prediction processing unit 120 may use uni-directional inter prediction to generate a predictive block of the PU. If a PU is in a B slice, inter prediction processing unit 120 may use uni-directional or bi-directional inter prediction to generate a predictive block of the PU.

[0127] Intra prediction processing unit 126 may generate predictive data for a PU by performing intra prediction on the PU. The predictive data for the PU may include predictive blocks of the PU and various syntax elements. Intra prediction processing unit 126 may perform intra prediction on PUs in I slices, P slices, and B slices.

[0128] To perform intra prediction on a PU, intra prediction processing unit 126 may use multiple intra prediction modes to generate multiple sets of predictive data for the PU. Intra prediction processing unit 126 may use samples from sample blocks of neighboring PUs to generate a predictive block for a PU. The neighboring PUs may be above, above and to the right, above and to the left, or to the left of the PU, assuming a left-to-right, top-to-bottom encoding order for PUs, CUs, and CTUs. Intra prediction processing unit 126 may use various numbers of intra prediction modes, e.g., 33 directional intra prediction modes. In some examples, the number of intra prediction modes may depend on the size of the region associated with the PU. In addition, as will be described in more detail below with reference to FIG. 11, intra prediction processing unit 126 may be configured to determine PDPC parameters for encoding a block of video data as a function of a height and/or width of a block of video data.

[0129] Prediction processing unit 100 may select the predictive data for PUs of a CU from among the predictive data generated by inter prediction processing unit 120 for the PUs or the predictive data generated by intra prediction processing unit 126 for the PUs. In some examples, prediction processing unit 100 selects the predictive data for the PUs of the CU based on rate/distortion metrics of the sets of predictive data. The predictive blocks of the selected predictive data may be referred to herein as the selected predictive blocks.

[0130] Residual generation unit 102 may generate, based on the coding blocks (e.g., luma, Cb and Cr coding blocks) for a CU and the selected predictive blocks (e.g., predictive luma, Cb and Cr blocks) for the PUs of the CU, residual blocks (e.g., luma, Cb and Cr residual blocks) for the CU. For instance, residual generation unit 102 may generate the residual blocks of the CU such that each sample in the residual blocks has a value equal to a difference between a sample in a coding block of the CU and a corresponding sample in a corresponding selected predictive block of a PU of the CU.

[0131] Transform processing unit 104 may perform quadtree partitioning to partition the residual blocks associated with a CU into transform blocks associated with TUs of the CU. Thus, a TU may be associated with a luma transform block and two chroma transform blocks. The sizes and positions of the luma and chroma transform blocks of TUs of a CU may or may not be based on the sizes and positions of prediction blocks of the PUs of the CU. A quadtree structure known as a "residual quadtree" (RQT) may include nodes associated with each of the regions. The TUs of a CU may correspond to leaf nodes of the RQT.

[0132] Transform processing unit 104 may generate transform coefficient blocks for each TU of a CU by applying one or more transforms to the transform blocks of the TU. Transform processing unit 104 may apply various transforms to a transform block associated with a TU. For example, transform processing unit 104 may apply a discrete cosine transform (DCT), a directional transform, or a conceptually similar transform to a transform block. In some examples, transform processing unit 104 does not apply transforms to a transform block. In such examples, the transform block may be treated as a transform coefficient block.

[0133] Quantization unit 106 may quantize the transform coefficients in a coefficient block. The quantization process may reduce the bit depth associated with some or all of the transform coefficients. For example, an n-bit transform coefficient may be rounded down to an m-bit transform coefficient during quantization, where n is greater than m. Quantization unit 106 may quantize a coefficient block associated with a TU of a CU based on a quantization parameter (QP) value associated with the CU. Video encoder 22 may adjust the degree of quantization applied to the coefficient blocks associated with a CU by adjusting the QP value associated with the CU. Quantization may introduce loss of information. Thus, quantized transform coefficients may have lower precision than the original ones.

[0134] Inverse quantization unit 108 and inverse transform processing unit 110 may apply inverse quantization and inverse transforms to a coefficient block, respectively, to reconstruct a residual block from the coefficient block. Reconstruction unit 112 may add the reconstructed residual block to corresponding samples from one or more predictive blocks generated by prediction processing unit 100 to produce a reconstructed transform block associated with a TU. By reconstructing transform blocks for each TU of a CU in this way, video encoder 22 may reconstruct the coding blocks of the CU.

[0135] Filter unit 114 may perform one or more deblocking operations to reduce blocking artifacts in the coding blocks associated with a CU. Decoded picture buffer 116 may store the reconstructed coding blocks after filter unit 114 performs the one or more deblocking operations on the reconstructed coding blocks. Inter prediction processing unit 120 may use a reference picture that contains the reconstructed coding blocks to perform inter prediction on PUs of other pictures. In addition, intra prediction processing unit 126 may use reconstructed coding blocks in decoded picture buffer 116 to perform intra prediction on other PUs in the same picture as the CU.

[0136] Entropy encoding unit 118 may receive data from other functional components of video encoder 22. For example, entropy encoding unit 118 may receive coefficient blocks from quantization unit 106 and may receive syntax elements from prediction processing unit 100. Entropy encoding unit 118 may perform one or more entropy encoding operations on the data to generate entropy-encoded data. For example, entropy encoding unit 118 may perform a CABAC operation, a context-adaptive variable length coding (CAVLC) operation, a variable-to-variable (V2V) length coding operation, a syntax-based context-adaptive binary arithmetic coding (SBAC) operation, a Probability Interval Partitioning Entropy (PIPE) coding operation, an Exponential-Golomb encoding operation, or another type of entropy encoding operation on the data. Video encoder 22 may output a bitstream that includes entropy-encoded data generated by entropy encoding unit 118. For instance, the bitstream may include data that represents a RQT for a CU.

[0137] FIG. 8 is a block diagram illustrating an example video decoder 30 that is configured to implement the techniques of this disclosure. FIG. 8 is provided for purposes of explanation and is not limiting on the techniques as broadly exemplified and described in this disclosure. For purposes of explanation, this disclosure describes video decoder 30 in the context of HEVC coding. However, the techniques of this disclosure may be applicable to other coding standards or methods, including techniques that allow for non-square partitioning and/or independent luma and chroma partitioning.

[0138] In the example of FIG. 8, video decoder 30 includes an entropy decoding unit 150, video data memory 151, a prediction processing unit 152, an inverse quantization unit 154, an inverse transform processing unit 156, a reconstruction unit 158, a filter unit 160, and a decoded picture buffer 162. Prediction processing unit 152 includes a motion compensation unit 164 and an intra prediction processing unit 166. In other examples, video decoder 30 may include more, fewer, or different functional components.

[0139] Video data memory 151 may store encoded video data, such as an encoded video bitstream, to be decoded by the components of video decoder 30. The video data stored in video data memory 151 may be obtained, for example, from computer-readable medium 16, e.g., 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. Video data memory 151 may form a coded picture buffer (CPB) that stores encoded video data from an encoded video bitstream. Decoded picture buffer 162 may be a reference picture memory that stores reference video data for use in decoding video data by video decoder 30, e.g., in intra- or inter-coding modes, or for output. Video data memory 151 and decoded picture buffer 162 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM), magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 151 and decoded picture buffer 162 may be provided by the same memory device or separate memory devices. In various examples, video data memory 151 may be on-chip with other components of video decoder 30, or off-chip relative to those components. Video data memory 151 may be the same as or part of storage media 28 of FIG. 1.

[0140] Video data memory 151 receives and stores encoded video data (e.g., NAL units) of a bitstream. Entropy decoding unit 150 may receive encoded video data (e.g., NAL units) from video data memory 151 and may parse the NAL units to obtain syntax elements. Entropy decoding unit 150 may entropy decode entropy-encoded syntax elements in the NAL units. Prediction processing unit 152, inverse quantization unit 154, inverse transform processing unit 156, reconstruction unit 158, and filter unit 160 may generate decoded video data based on the syntax elements extracted from the bitstream. Entropy decoding unit 150 may perform a process generally reciprocal to that of entropy encoding unit 118.

[0141] In accordance with some examples of this disclosure, entropy decoding unit 150 may determine a tree structure as part of obtaining the syntax elements from the bitstream. The tree structure may specify how an initial video block, such as a CTB, is partitioned into smaller video blocks, such as coding units. In accordance with one or more techniques of this disclosure, for each respective non-leaf node of the tree structure at each depth level of the tree structure, there are a plurality of allowed splitting patterns for the respective non-leaf node and the video block corresponding to the respective non-leaf node is partitioned into video blocks corresponding to the child nodes of the respective non-leaf node according to one of the plurality of allowable splitting patterns.

[0142] In addition, as will be explained in more detail below with reference to FIG. 10, video decoder 30 may be configured to determine how to reuse coding mode information received for a luma block for use when decoding chroma blocks in situations where there are two or more luma blocks that correspond to a single chroma block.

[0143] In addition to obtaining syntax elements from the bitstream, video decoder 30 may perform a reconstruction operation on a non-partitioned CU. To perform the reconstruction operation on a CU, video decoder 30 may perform a reconstruction operation on each TU of the CU. By performing the reconstruction operation for each TU of the CU, video decoder 30 may reconstruct residual blocks of the CU.

[0144] As part of performing a reconstruction operation on a TU of a CU, inverse quantization unit 154 may inverse quantize, i.e., de-quantize, coefficient blocks associated with the TU. After inverse quantization unit 154 inverse quantizes a coefficient block, inverse transform processing unit 156 may apply one or more inverse transforms to the coefficient block in order to generate a residual block associated with the TU. For example, inverse transform processing unit 156 may apply an inverse DCT, an inverse integer transform, an inverse Karhunen-Loeve transform (KLT), an inverse rotational transform, an inverse directional transform, or another inverse transform to the coefficient block.

[0145] If a PU is encoded using intra prediction, intra prediction processing unit 166 may perform intra prediction to generate predictive blocks of the PU. Intra prediction processing unit 166 may use an intra prediction mode to generate the predictive blocks of the PU based on samples spatially-neighboring blocks. Intra prediction processing unit 166 may determine the intra prediction mode for the PU based on one or more syntax elements obtained from the bitstream. In addition, as will be described in more detail below with reference to FIG. 12, intra prediction processing unit 166 may be configured to determine PDPC parameters for encoding a block of video data as a function of a height and/or width of a block of video data.

[0146] If a PU is encoded using inter prediction, entropy decoding unit 150 may determine motion information for the PU. Motion compensation unit 164 may determine, based on the motion information of the PU, one or more reference blocks. Motion compensation unit 164 may generate, based on the one or more reference blocks, predictive blocks (e.g., predictive luma, Cb and Cr blocks) for the PU.

[0147] Reconstruction unit 158 may use transform blocks (e.g., luma, Cb and Cr transform blocks) for TUs of a CU and the predictive blocks (e.g., luma, Cb and Cr blocks) of the PUs of the CU, i.e., either intra prediction data or inter prediction data, as applicable, to reconstruct the coding blocks (e.g., luma, Cb and Cr coding blocks) for the CU. For example, reconstruction unit 158 may add samples of the transform blocks (e.g., luma, Cb and Cr transform blocks) to corresponding samples of the predictive blocks (e.g., luma, Cb and Cr predictive blocks) to reconstruct the coding blocks (e.g., luma, Cb and Cr coding blocks) of the CU.

[0148] Filter unit 160 may perform a deblocking operation to reduce blocking artifacts associated with the coding blocks of the CU. Video decoder 30 may store the coding blocks of the CU in decoded picture buffer 162. Decoded picture buffer 162 may provide reference pictures for subsequent motion compensation, intra prediction, and presentation on a display device, such as display device 32 of FIG. 1. For instance, video decoder 30 may perform, based on the blocks in decoded picture buffer 162, intra prediction or inter prediction operations for PUs of other CUs.

[0149] FIG. 9 is a flowchart illustrating an example operation of a video coder in accordance with a technique of this disclosure. The video coder may be video encoder 22 and/or video decoder 30. In accordance with the techniques of this disclosure, encoder 22 and/or video decoder 30 may be configured to partition video data into partitions of luma components (200), partition the video data into partitions of chroma components, wherein the chroma components are partitioned independently of the luma components (202), code a first partition of the luma components (204), code a syntax element indicating if information associated with coding the first partition of the luma components is to be used for coding a second partition of the chroma components (206), and code the second partition of the chroma components in accordance with the syntax element (208).

[0150] FIG. 10 is a flowchart illustrating an example operation of a video decoder in accordance with a technique of this disclosure. The techniques of FIG. 10 may be performed by one or more hardware structures of video decoder 30.

[0151] In one example of the disclosure, video decoder 30 may be configured to receive the bitstream of encoded video data, the encoded video data representing partitioned luma blocks and partitioned chroma blocks, wherein the chroma blocks are partitioned independently of the luma blocks (212). Video decoder 30 may be further configured to determine a respective coding mode corresponding to the respective partitioned luma blocks (214), and decode the respective partitioned luma blocks according the determined respective coding modes (216).

[0152] Video decoder 30 may be further configured to decode a first syntax element indicating that the respective coding modes associated with the respective partitioned luma blocks are to be used for decoding a first partitioned chroma block, wherein the first partitioned chroma block is aligned with two or more partitioned luma blocks (218). Video decoder 30 may further determine a chroma coding mode for the first partitioned chroma block according to a function of the respective coding modes of the two or more partitioned luma blocks (220), and decode the first partitioned chroma block in accordance with the determined chroma coding mode (222).

[0153] In one example of the disclosure, the chroma blocks are partitioned independently of the luma blocks such that at least one partitioned chroma block is not aligned with a single partitioned luma block.

[0154] In another example of the disclosure, to determine the respective coding mode corresponding to the respective partitioned luma blocks, video decoder 30 may be further configured to receive second syntax elements corresponding to the respective partitioned luma blocks, the second syntax elements indicating the respective coding mode, and decode the second syntax elements corresponding to the respective partitioned luma blocks to determine the respective coding mode.

[0155] In another example of the disclosure, to determine the respective coding mode corresponding to the respective partitioned luma blocks, video decoder 30 is further configured to select one or more respective coding modes from one or more representative locations of the respective partitioned luma blocks. In another example, video decoder 30 is configured to select the one or more respective coding modes according to the function.

[0156] In another example, the one or more representative locations include a center representative location of the respective partitioned luma blocks, and wherein to determine the chroma coding mode for the first partitioned chroma block according to the function, video decoder 30 is further configured to obtain information indicating the determined respective coding mode stored for the center representative location.

[0157] In another example, the one or more representative locations include a corner representative location of the respective partitioned luma blocks, and wherein to determine the chroma coding mode for the first partitioned chroma block according to the function, video decoder 30 is further configured to obtain information indicating the determined respective coding mode stored for the corner representative location. In one example, the one or more representative locations comprise one or more sub-blocks.

[0158] In another example of the disclosure, video decoder 30 may be further configured to divide the respective partitioned luma blocks into respective sub-blocks, and store information indicating the determined respective coding mode in respective memory locations associated with the respective sub-blocks.

[0159] In another example of the disclosure, the function includes a location of one or more respective sub-blocks of the two or more partitioned luma blocks. In another example of the disclosure, the location of the one or more respective sub-blocks is a center sub-block of the two or more partitioned luma blocks, and wherein to determine the chroma coding mode for the first partitioned chroma block according to the function, video decoder 30 is configured to obtain the information indicating the determined respective coding mode stored for the center sub-block.

[0160] In another example of the disclosure, the location of the one or more respective sub-blocks is a corner sub-block of the two more partitioned luma blocks, and wherein to determine the chroma coding mode for the first partitioned chroma block according to the function, the video decoder 30 is configured to obtain the information indicating the determined respective coding mode stored for the corner sub-block.

[0161] In another example of the disclosure, the function includes a statistical analysis of the information indicating the determined respective coding mode in respective memory locations associated with the respective sub-blocks.

[0162] In another example of the disclosure, to determine the chroma coding mode for the first partitioned chroma block according to the function, video decoder 30 is further configured to analyze the information stored in the respective memory locations using one of a gradient or a higher derivative.

[0163] In another example of the disclosure, the information includes one or more of an indication of a direct mode for chroma prediction, a prediction direction, motion information, a flag for a position dependent intra prediction combination mode, one or more parameters for the position dependent intra prediction combination mode, one or more second transform sets for a non-separable transform, an enhanced multiple transform, an adaptive multiple transform, or one or more contexts for determining entropy coding data models.

[0164] In another example of the disclosure, video decoder 30 may configured to receive a third syntax element indicating the function.

[0165] In another example of the disclosure, video decoder 30 is part of a wireless communication device, the wireless communication device further comprising a receiver configured to receive the bitstream of encoded video data. In one example, the wireless communication device is a mobile station and the bitstream of encoded video data is received by the receiver and modulated according to a cellular communication standard.

[0166] FIG. 11 is a flowchart illustrating an example operation of video encoder 22 in accordance with a technique of this disclosure. The techniques of FIG. 12 may be performed by one or more hardware structures of video encoder 22.

[0167] In one example of the disclosure, video encoder 22 may be configured to receive a block of video data, , the block of video data having a non-square shape defined by a width and a height (230), determine one or more PDPC parameters based on one or more of the width or the height of the block of video data (232), and encode the block of video data using a PDPC mode and the determined PDPC parameters (234). As discussed above, it should be understood that the techniques FIG. 11 may be used for determining prediction mode parameters for any prediction mode that uses non-square blocks, including prediction modes that apply a weighted average to the predictor and reference samples according to sample position.

[0168] In one example, the one or more PDPC parameters include one or more horizontally-related PDPC parameters and one or more vertically-related PDPC parameters, and wherein to determine the one or more PDPC parameters, video encoder 22 is further configured to determine the one or more horizontally-related PDPC parameters based on the width of the block of video data, and determine the one or more vertically-related PDPC parameters based on the height of the block of video data.

[0169] In another example of the disclosure, to determine the one or more horizontally-related PDPC parameters, video encoder 22 is further configured to retrieve one or more entries of one or more lookup tables as a function of the width of the block of video data, and wherein to determine the one or more vertically-related PDPC parameters, video encoder 22 is further configured to retrieve one or more entries of the one or more lookup tables as a function of the height of the block of video data.

[0170] In another example of the disclosure, to retrieve the one or more entries of the one or more lookup tables as the function of the width of the block of video data, video encoder 22 is further configured to retrieve a first index in a first lookup table based on the width of the block of video data, the first index pointing to a first entry in a second lookup table, and retrieve the one or more horizontally-related PDPC parameters in the second lookup table based on the retrieved first index. In a further example, to retrieve the one or more entries of the one or more lookup tables as the function of the of the height of the block of video data, video encoder 22 is further configured to retrieve a second index in the first lookup table based on the height of the block of video data, the second index pointing to a second entry in the second lookup table, and retrieve the one or more vertically-related PDPC parameters in the second lookup table based on the retrieved second index.

[0171] In another example of the disclosure, the one or more PDPC parameters include one or more non-directional PDPC parameters that are not horizontally-related and are not vertically-related, and wherein to determine the one or more PDPC parameters, video encoder 22 is further configured to determine the one or more non-directional PDPC parameters based on a function of the width and the height of the block of video data.

[0172] In another example of the disclosure, the function is one or more of a minimum of the width and the height of the block of video data, a maximum of the width and the height of the block of video data, or a weighted average of the width and the height of the block of video data. In a further example, to determine the one or more non-directional PDPC parameters, video encoder 22 is further configured to access one or more entries of one or more lookup tables as a function of the width and the height of the block of video data.

[0173] In another example of the disclosure, video encoder 22 is included in a wireless communication device, the wireless communication device further comprising a transmitter configured to transmit the encoded block of video data. In another example, the wireless communication device is a mobile station and the encoded block of video data is transmitted by the transmitter and modulated according to a cellular communication standard.

[0174] FIG. 12 is a flowchart illustrating an example operation of video decoder 30 in accordance with a technique of this disclosure. The techniques of FIG. 12 may be performed by one or more hardware structures of video decoder 30.

[0175] In one example of the disclosure, video decoder 30 is configured to receive a block of video data encoded using a PDPC mode, the block of video data having a non-square shape defined by a width and a height (240), determine one or more PDPC parameters based on one or more of the width or the height of the block of video data (242), and decode the block of video data using the PDPC mode and the determined PDPC parameters (244). As discussed above, it should be understood that the techniques FIG. 12 may be used for determining prediction mode parameters for any prediction mode that uses non-square blocks, including prediction modes that apply a weighted average to the predictor and reference samples according to sample position.

[0176] In one example of the disclosure, the one or more PDPC parameters include one or more horizontally-related PDPC parameters and one or more vertically-related PDPC parameters, and wherein to determine the one or more PDPC parameters, video decoder 30 is further configured to determine the one or more horizontally-related PDPC parameters based on the width of the block of video data, and determine the one or more vertically-related PDPC parameters based on the height of the block of video data.

[0177] In another example of the disclosure, to determine the one or more horizontally-related PDPC parameters, video decoder 30 is further configured to retrieve one or more entries of one or more lookup tables as a function of the width of the block of video data, and wherein to determine the one or more vertically-related PDPC parameters, video decoder 30 is further configured to retrieve one or more entries of the one or more lookup tables as a function of the height of the block of video data.

[0178] In another example of the disclosure, to retrieve the one or more entries of the one or more lookup tables as the function of the width of the block of video data, video decoder 30 is further configured to retrieve a first index in a first lookup table based on the width of the block of video data, the first index pointing to a first entry in a second lookup table, and retrieve the one or more horizontally-related PDPC parameters in the second lookup table based on the retrieved first index. In a further example, to retrieve the one or more entries of the one or more lookup tables as the function of the of the height of the block of video data, video decoder 30 is further configured to retrieve a second index in the first lookup table based on the height of the block of video data, the second index pointing to a second entry in the second lookup table, and retrieve the one or more vertically-related PDPC parameters in the second lookup table based on the retrieved second index.

[0179] In another example, the one or more PDPC parameters include one or more non-directional PDPC parameters that are not horizontally-related and are not vertically-related, and wherein to determine the one or more PDPC parameters, video decoder 30 is further configured to determine the one or more non-directional PDPC parameters based on a function of the width and the height of the block of video data.

[0180] In another example, the function is one or more of a minimum of the width and the height of the block of video data, a maximum of the width and the height of the block of video data, or a weighted average of the width and the height of the block of video data.

[0181] In another example of the disclosure, to determine the one or more non-directional PDPC parameters, video decoder 30 is further configured to access one or more entries of one or more lookup tables as a function of the width and the height of the block of video data.

[0182] In another example of the disclosure, video decoder 30 is part of is a wireless communication device, the wireless communication device further comprising a receiver configured to receive the block of video data. In a further example, the wireless communication device is a mobile station and the block of video data is received by the receiver and modulated according to a cellular communication standard.

[0183] Certain aspects of this disclosure have been described with respect to extensions of the HEVC standard for purposes of illustration. However, the techniques described in this disclosure may be useful for other video coding processes, including other standard or proprietary video coding processes under development or not yet developed.

[0184] A video coder, as described in this disclosure, may refer to a video encoder or a video decoder. Similarly, a video coding unit may refer to a video encoder or a video decoder. Likewise, video coding may refer to video encoding or video decoding, as applicable.

[0185] It is to be recognized that depending on the example, certain acts or events of any of the techniques described herein can be performed in a different sequence, may be added, merged, or left out altogether (e.g., not all described acts or events are necessary for the practice of the techniques). Moreover, in certain examples, acts or events may be performed concurrently, e.g., through multi-threaded processing, interrupt processing, or multiple processors, rather than sequentially.

[0186] In one or more examples, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium and executed by a hardware-based processing unit. Computer-readable media may include computer-readable storage media, which corresponds to a tangible medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to another, e.g., according to a communication protocol. In this manner, computer-readable media generally may correspond to (1) tangible computer-readable storage media which is non-transitory or (2) a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or data structures for implementation of the techniques described in this disclosure. A computer program product may include a computer-readable medium.

[0187] By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that computer-readable storage media and data storage media do not include connections, carrier waves, signals, or other transitory media, but are instead directed to non-transitory, tangible storage media. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and Blu-ray disc, where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0188] Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term “processor,” as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules configured for encoding and decoding, or incorporated in a combined codec. Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0189] The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, modules, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

[0190] Various examples have been described. These and other examples are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method of decoding video data, the method comprising:

receiving a block of video data encoded using a prediction mode, the block of video data having a non-square shape defined by a width and a height;

determining one or more parameters for the prediction mode based on one or more of the width or the height of the block of video data; and

decoding the block of video data using the PDPC mode and the determined PDPC parameters.

2. The method of claim 1, wherein the prediction mode is a position dependent intra prediction combination (PDPC) mode, and wherein the one or more parameters for the prediction mode are one or more PDPC parameters.

3. The method of claim 2, wherein the one or more PDPC parameters include one or more horizontally-related PDPC parameters and one or more vertically-related PDPC parameters, and wherein determining the one or more PDPC parameters comprises:

determining the one or more horizontally-related PDPC parameters based on the width of the block of video data; and

determining the one or more vertically-related PDPC parameters based on the height of the block of video data.

4. The method of claim 3, wherein determining the one or more horizontally-related PDPC parameters comprises retrieving one or more entries of one or more lookup tables as a function of the width of the block of video data, and

wherein determining the one or more vertically-related PDPC parameters comprises retrieving one or more entries of the one or more lookup tables as a function of the height of the block of video data.

5. The method of claim 4, wherein retrieving the one or more entries of the one or more lookup tables as the function of the width of the block of video data comprises:

retrieving a first index in a first lookup table based on the width of the block of video data, the first index pointing to a first entry in a second lookup table; and

retrieving the one or more horizontally-related PDPC parameters in the second lookup table based on the retrieved first index, and

wherein retrieving the one or more entries of the one or more lookup tables as the function of the of the height of the block of video data comprises:

retrieving a second index in the first lookup table based on the height of the block of video data, the second index pointing to a second entry in the second lookup table; and

retrieving the one or more vertically-related PDPC parameters in the second lookup table based on the retrieved second index.

6. The method of claim 2, wherein the one or more PDPC parameters include one or more non-directional PDPC parameters that are not horizontally-related and are not vertically-related, and wherein determining the one or more PDPC parameters comprises:

determining the one or more non-directional PDPC parameters based on a function of the width and the height of the block of video data.

7. The method of claim 6, wherein the function is one or more of a minimum of the width and the height of the block of video data, a maximum of the width and the height of the block of video data, or a weighted average of the width and the height of the block of video data.

8. The method of claim 7, wherein determining the one or more non-directional PDPC parameters comprises accessing one or more entries of one or more lookup tables as a function of the width and the height of the block of video data.

9. The method of claim 1, the method being executable on a wireless communication device, wherein the device comprises:

a memory configured to store the block of video data;

a processor configured to execute instructions to process the block of video data stored in the memory; and

a receiver configured to receive the block of video data.

10. The method of claim 9, wherein the wireless communication device is a mobile station and the block of video data is received by the receiver and modulated according to a cellular communication standard.

11. An apparatus configured to decode video data, the apparatus comprising:

a memory configured to store a block of video data encoded using a prediction mode, the block of video data having a non-square shape defined by a width and a height; and

one or more processors configured to:

receive the block of video data;

determine one or more parameters for the prediction mode based on one or more of the width or the height of the block of video data; and

decode the block of video data using the PDPC mode and the determined PDPC parameters.

12. The apparatus of claim 11, wherein the prediction mode is a position dependent intra prediction combination (PDPC) mode, and wherein the one or more parameters for the prediction mode are one or more PDPC parameters.

13. The apparatus of claim 12, wherein the one or more PDPC parameters include one or more horizontally-related PDPC parameters and one or more vertically-related PDPC parameters, and wherein to determine the one or more PDPC parameters, the one or more processors are further configured to:

determine the one or more horizontally-related PDPC parameters based on the width of the block of video data; and

determine the one or more vertically-related PDPC parameters based on the height of the block of video data.

14. The apparatus of claim 13, wherein to determine the one or more horizontally-related PDPC parameters, the one or more processors are further configured to retrieve one or more entries of one or more lookup tables as a function of the width of the block of video data, and

wherein to determine the one or more vertically-related PDPC parameters, the one or more processors are further configured to retrieve one or more entries of the one or more lookup tables as a function of the height of the block of video data.

15. The apparatus of claim 14, wherein to retrieve the one or more entries of the one or more lookup tables as the function of the width of the block of video data, the one or more processors are further configured to:

retrieve a first index in a first lookup table based on the width of the block of video data, the first index pointing to a first entry in a second lookup table; and

retrieve the one or more horizontally-related PDPC parameters in the second lookup table based on the retrieved first index, and

wherein to retrieve the one or more entries of the one or more lookup tables as the function of the of the height of the block of video data, the one or more processors are further configured to:

retrieve a second index in the first lookup table based on the height of the block of video data, the second index pointing to a second entry in the second lookup table; and

retrieve the one or more vertically-related PDPC parameters in the second lookup table based on the retrieved second index.

16. The apparatus of claim 12, wherein the one or more PDPC parameters include one or more non-directional PDPC parameters that are not horizontally-related and are not vertically-related, and wherein to determine the one or more PDPC parameters, the one or more processors are further configured to determine the one or more non-directional PDPC parameters based on a function of the width and the height of the block of video data.

17. The apparatus of claim 16, wherein the function is one or more of a minimum of the width and the height of the block of video data, a maximum of the width and the height of the block of video data, or a weighted average of the width and the height of the block of video data.

18. The apparatus of claim 17, wherein to determine the one or more non-directional PDPC parameters, the one or more processors are further configured to access one or more entries of one or more lookup tables as a function of the width and the height of the block of video data.

19. The apparatus of claim 11, wherein the apparatus is a wireless communication device, the apparatus further comprising:

a receiver configured to receive the block of video data.

20. The apparatus of claim 19, wherein the wireless communication device is a mobile station and the block of video data is received by the receiver and modulated according to a cellular communication standard.

21. An apparatus configured to decode video data, the apparatus comprising:

means for receiving a block of video data encoded using a prediction mode, the block of video data having a non-square shape defined by a width and a height;

means for determining one or more parameters for the prediction mode based on one or more of the width or the height of the block of video data; and

means for decoding the block of video data using the PDPC mode and the determined PDPC parameters.

22. The apparatus of claim 21, wherein the prediction mode is a position dependent intra prediction combination (PDPC) mode, and wherein the one or more parameters for the prediction mode are one or more PDPC parameters.

23. A non-transitory computer-readable storage medium having instructions stored thereon that, when executed, cause one or more processors of a device to configured to decode video data to:

receive a block of video data encoded using a prediction mode, the block of video data having a non-square shape defined by a width and a height;

determine one or more parameters for the prediction mode based on one or more of the width or the height of the block of video data; and

decode the block of video data using the PDPC mode and the determined PDPC parameters.

24. The non-transitory computer-readable storage medium of claim 23, wherein the prediction mode is a position dependent intra prediction combination (PDPC) mode, and wherein the one or more parameters for the prediction mode are one or more PDPC parameters.

25. A method of encoding video data, the method comprising:

receiving a block of video data, the block of video data having a non-square shape defined by a width and a height;

determining one or more parameters for a prediction mode based on one or more of the width or the height of the block of video data; and

encoding the block of video data using the prediction mode and the determined one or more parameters.

26. The method of claim 25, wherein the prediction mode is a position dependent intra prediction combination (PDPC) mode, and wherein the one or more parameters for the prediction mode are one or more PDPC parameters.

27. The method of claim 26, wherein the one or more PDPC parameters include one or more horizontally-related PDPC parameters and one or more vertically-related PDPC parameters, and wherein determining the one or more PDPC parameters comprises:

determining the one or more horizontally-related PDPC parameters based on the width of the block of video data; and

determining the one or more vertically-related PDPC parameters based on the height of the block of video data.

28. The method of claim 27, wherein determining the one or more horizontally-related PDPC parameters comprises retrieving one or more entries of one or more lookup tables as a function of the width of the block of video data, and

wherein determining the one or more vertically-related PDPC parameters comprises retrieving one or more entries of the one or more lookup tables as a function of the height of the block of video data.

29. The method of claim 28, wherein retrieving the one or more entries of the one or more lookup tables as the function of the width of the block of video data comprises:

retrieving a first index in a first lookup table based on the width of the block of video data, the first index pointing to a first entry in a second lookup table; and

retrieving the one or more horizontally-related PDPC parameters in the second lookup table based on the retrieved first index, and

wherein retrieving the one or more entries of the one or more lookup tables as the function of the of the height of the block of video data comprises:

retrieving a second index in the first lookup table based on the height of the block of video data, the second index pointing to a second entry in the second lookup table; and

retrieving the one or more vertically-related PDPC parameters in the second lookup table based on the retrieved second index.

30. The method of claim 26, wherein the one or more PDPC parameters include one or more non-directional PDPC parameters that are not horizontally-related and are not vertically-related, and wherein determining the one or more PDPC parameters comprises:

determining the one or more non-directional PDPC parameters based on a function of the width and the height of the block of video data.

31. The method of claim 30, wherein the function is one or more of a minimum of the width and the height of the block of video data, a maximum of the width and the height of the block of video data, or a weighted average of the width and the height of the block of video data.

32. The method of claim 31, wherein determining the one or more non-directional PDPC parameters comprises accessing one or more entries of one or more lookup tables as a function of the width and the height of the block of video data.

33. The method of claim 25, the method being executable on a wireless communication device, wherein the device comprises:

a memory configured to store the block of video data;

a processor configured to execute instructions to process the block of video data stored in the memory; and

a transmitter configured to transmit the encoded block of video data.

34. The method of claim 33, wherein the wireless communication device is a mobile station and the encoded block of video data is transmitted by the transmitter and modulated according to a cellular communication standard.

35. An apparatus configured to encode video data, the apparatus comprising:

a memory configured to store a block of video data, the block of video data having a non-square shape defined by a width and a height; and

one or more processors configured to:

receive the block of video data;

determine one or more parameters for a prediction mode based on one or more of the width or the height of the block of video data; and

encode the block of video data using the prediction mode and the determined one or more parameters.

36. The apparatus of claim 35, wherein the prediction mode is a position dependent intra prediction combination (PDPC) mode, and wherein the one or more parameters for the prediction mode are one or more PDPC parameters.

37. The apparatus of claim 36, wherein the one or more PDPC parameters include one or more horizontally-related PDPC parameters and one or more vertically-related PDPC parameters, and wherein to determine the one or more PDPC parameters, the one or more processors are further configured to:

determine the one or more horizontally-related PDPC parameters based on the width of the block of video data; and

determine the one or more vertically-related PDPC parameters based on the height of the block of video data.

38. The apparatus of claim 37, wherein to determine the one or more horizontally-related PDPC parameters, the one or more processors are further configured to retrieve one or more entries of one or more lookup tables as a function of the width of the block of video data, and

wherein to determine the one or more vertically-related PDPC parameters, the one or more processors are further configured to retrieve one or more entries of the one or more lookup tables as a function of the height of the block of video data.

39. The apparatus of claim 38, wherein to retrieve the one or more entries of the one or more lookup tables as the function of the width of the block of video data, the one or more processors are further configured to:

retrieve a first index in a first lookup table based on the width of the block of video data, the first index pointing to a first entry in a second lookup table; and

retrieve the one or more horizontally-related PDPC parameters in the second lookup table based on the retrieved first index, and

wherein to retrieve the one or more entries of the one or more lookup tables as the function of the of the height of the block of video data, the one or more processors are further configured to:

retrieve a second index in the first lookup table based on the height of the block of video data, the second index pointing to a second entry in the second lookup table; and

retrieve the one or more vertically-related PDPC parameters in the second lookup table based on the retrieved second index.

40. The apparatus of claim 36, wherein the one or more PDPC parameters include one or more non-directional PDPC parameters that are not horizontally-related and are not vertically-related, and wherein to determine the one or more PDPC parameters, the one or more processors are further configured to determine the one or more non-directional PDPC parameters based on a function of the width and the height of the block of video data.

41. The apparatus of claim 40, wherein the function is one or more of a minimum of the width and the height of the block of video data, a maximum of the width and the height of the block of video data, or a weighted average of the width and the height of the block of video data.

42. The apparatus of claim 41, wherein to determine the one or more non-directional PDPC parameters, the one or more processors are further configured to access one or more entries of one or more lookup tables as a function of the width and the height of the block of video data.

43. The apparatus of claim 35, wherein the apparatus is a wireless communication device, the apparatus further comprising:

a transmitter configured to transmit the encoded block of video data.

44. The apparatus of claim 43, wherein the wireless communication device is a mobile station and the encoded block of video data is transmitted by the transmitter and modulated according to a cellular communication standard.

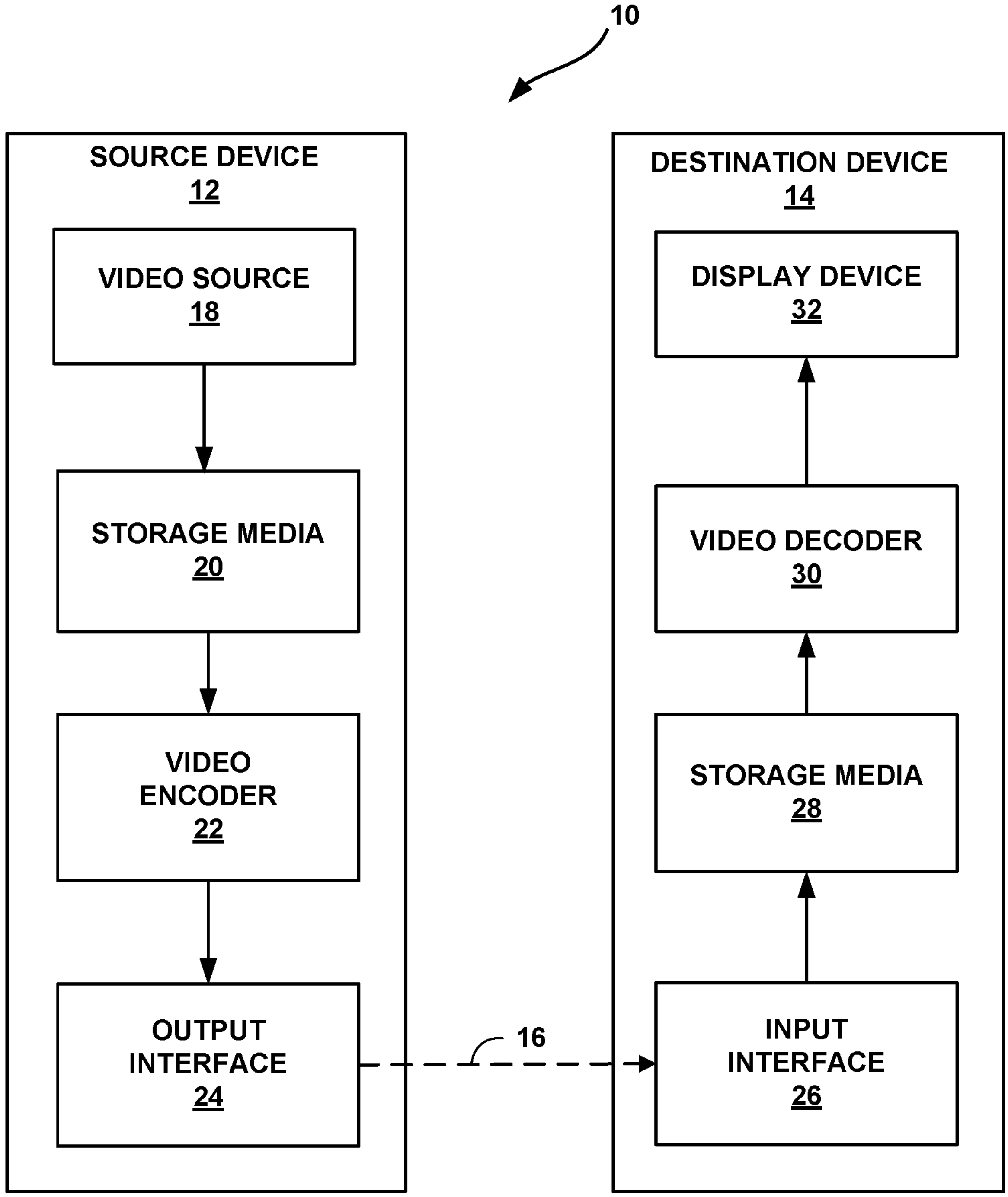

10
SOURCE DEVICE
12
VIDEO SOURCE
18
STORAGE MEDIA
20
VIDEO
ENCODER
22
OUTPUT
INTERFACE
24
16
DESTINATION DEVICE
14
DISPLAY DEVICE
32
VIDEO DECODER
30
STORAGE MEDIA
28
INPUT
INTERFACE
26

FIG. 1

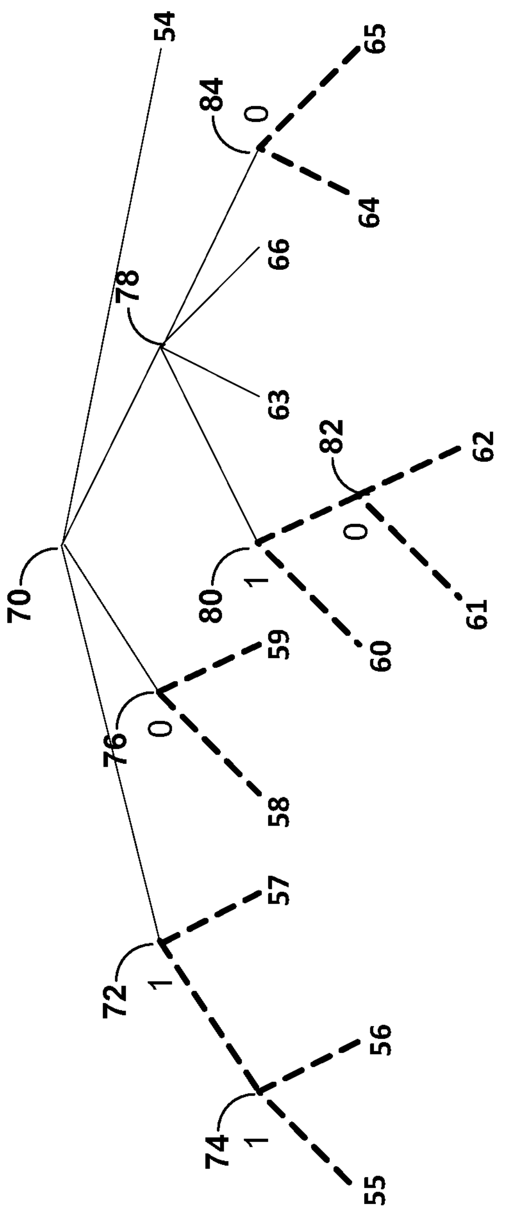

54
65
84
0
64
78
66
63
82
62
70
80
1
0
61
60
59
76
0
58
57
72
1
56
74
1
55

FIG. 2B

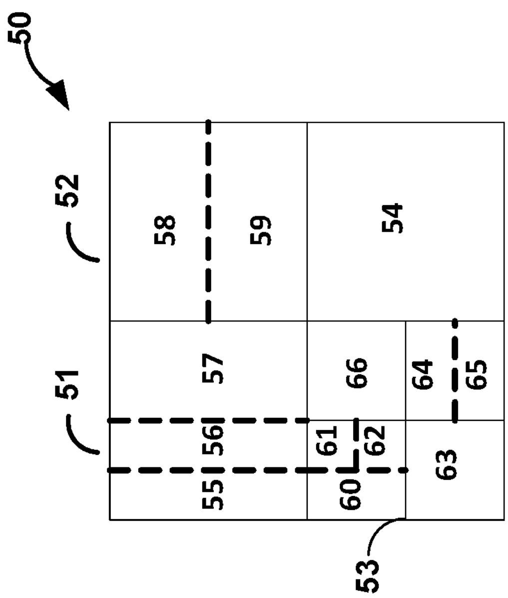

50
52
58
59
54
51
57
66
64
65
56
61
62
55
60
63
53

FIG. 2A

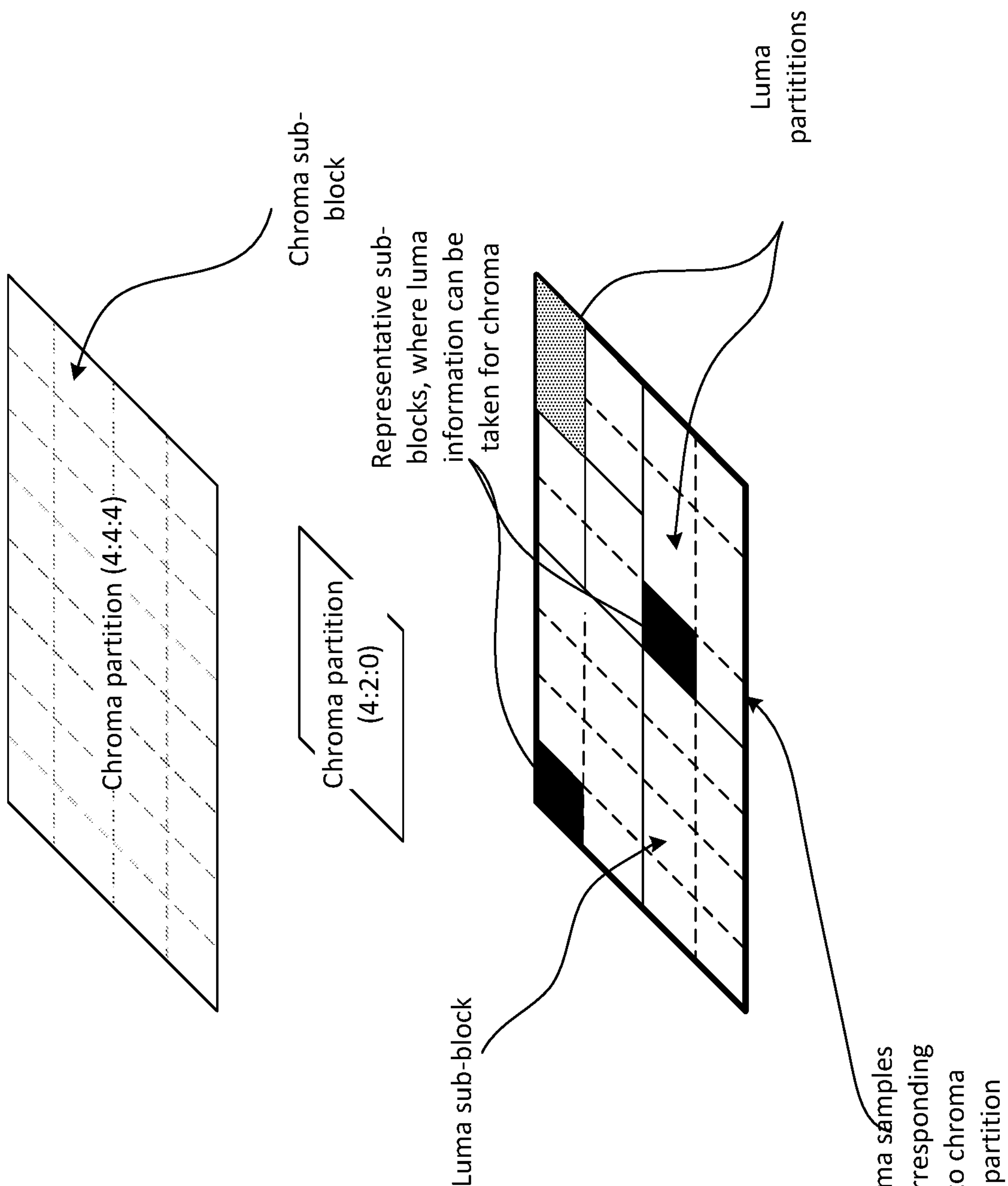

Chroma sub-block
Chroma partition (4:4:4)
Chroma partition (4:2:0)
Representative sub-blocks, where luma information can be taken for chroma
Luma partitions
Luma sub-block
Luma samples corresponding to chroma partition

FIG. 3

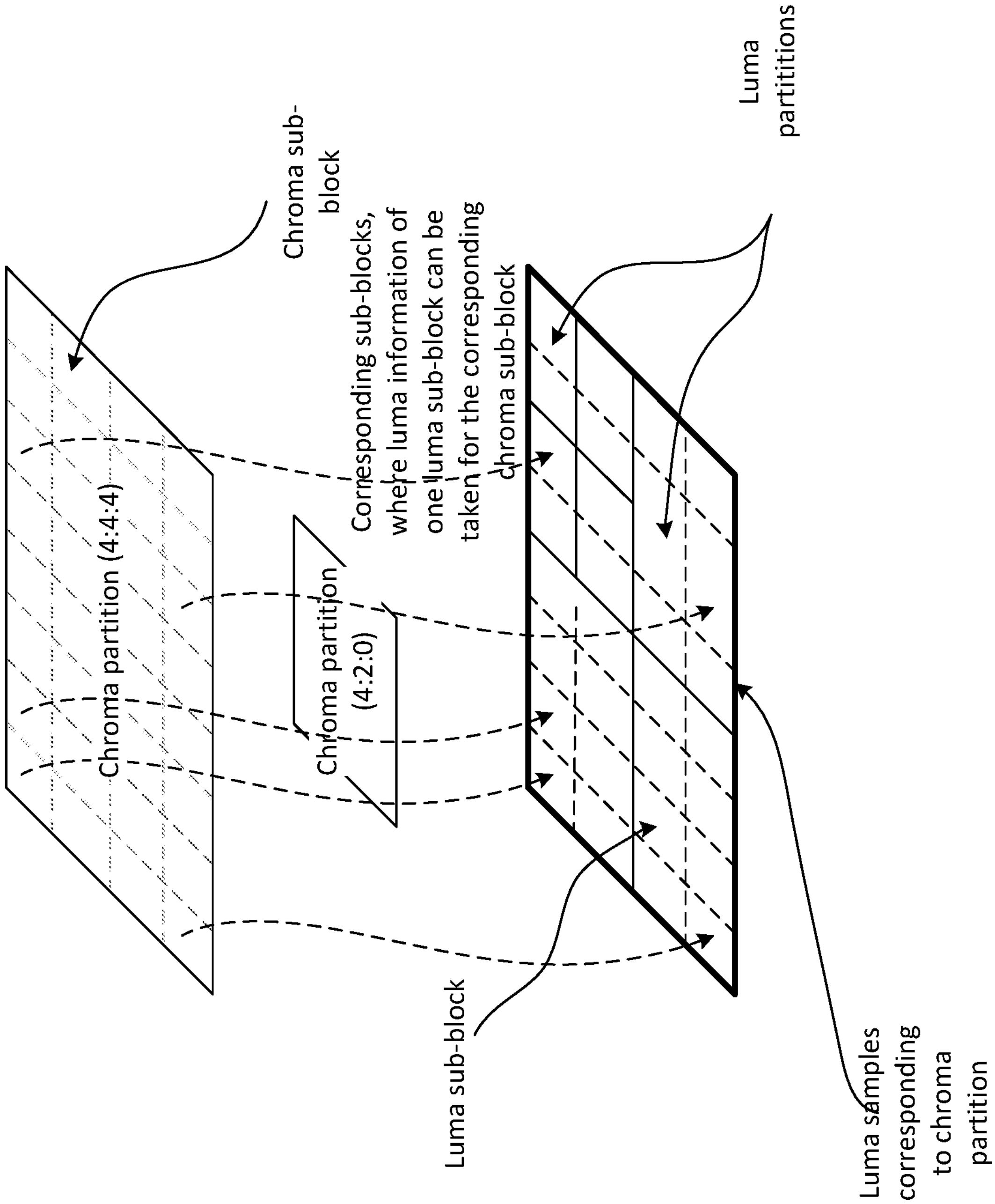

Chroma sub-block
Chroma partition (4:4:4)
Chroma partition (4:2:0)
Corresponding sub-blocks, where luma information of one luma sub-block can be taken for the corresponding chroma sub-block
Luma partitions
Luma sub-block
Luma samples corresponding to chroma partition

FIG. 4

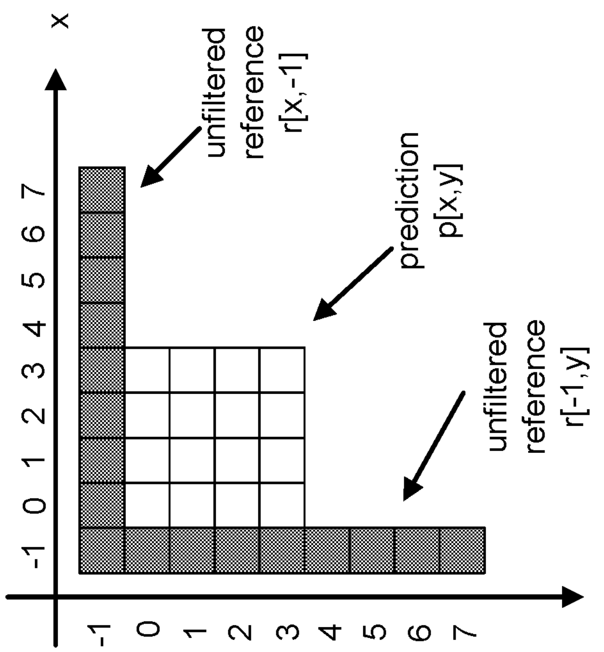

x
y
-1 0 1 2 3 4 5 6 7
-1 0 1 2 3 4 5 6 7
unfiltered reference r[x,-1]
prediction p[x,y]
unfiltered reference r[-1,y]

FIG. 5A

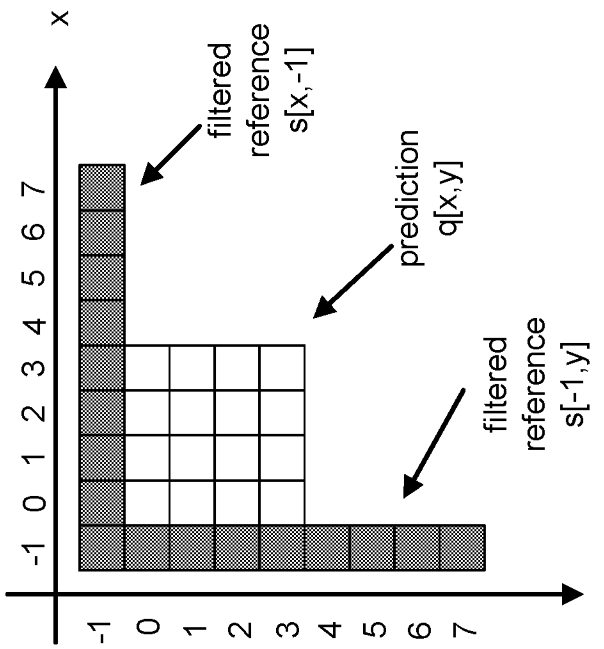

x
y
-1 0 1 2 3 4 5 6 7
-1 0 1 2 3 4 5 6 7
filtered reference s[x,-1]
prediction q[x,y]
filtered reference s[-1,y]

FIG. 5B

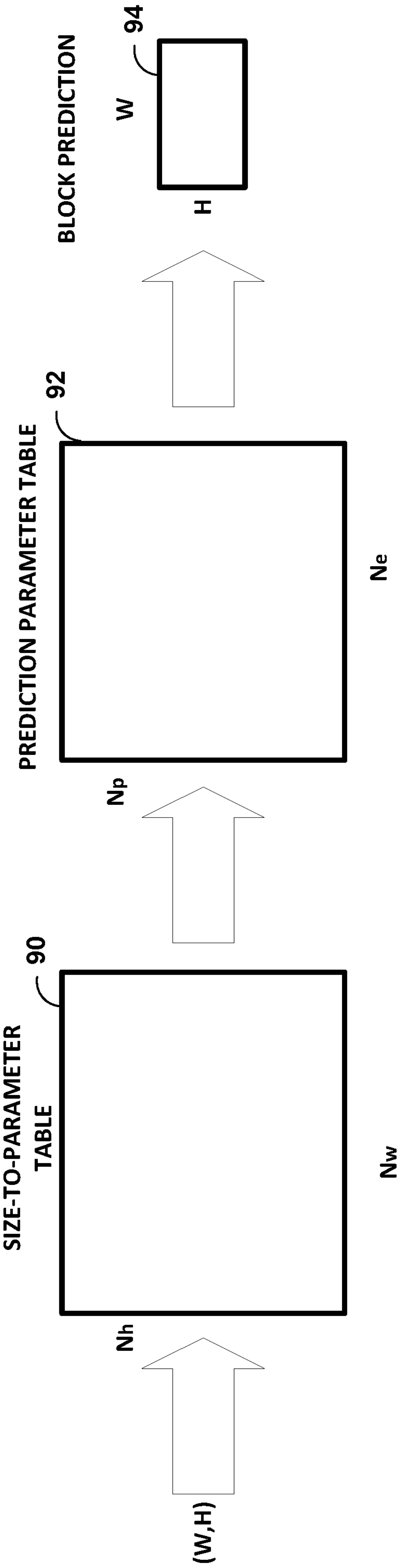

(W,H)
SIZE-TO-PARAMETER TABLE
90
Nh
Nw
PREDICTION PARAMETER TABLE
92
Np
Ne
BLOCK PREDICTION
94
W
H

FIG. 6

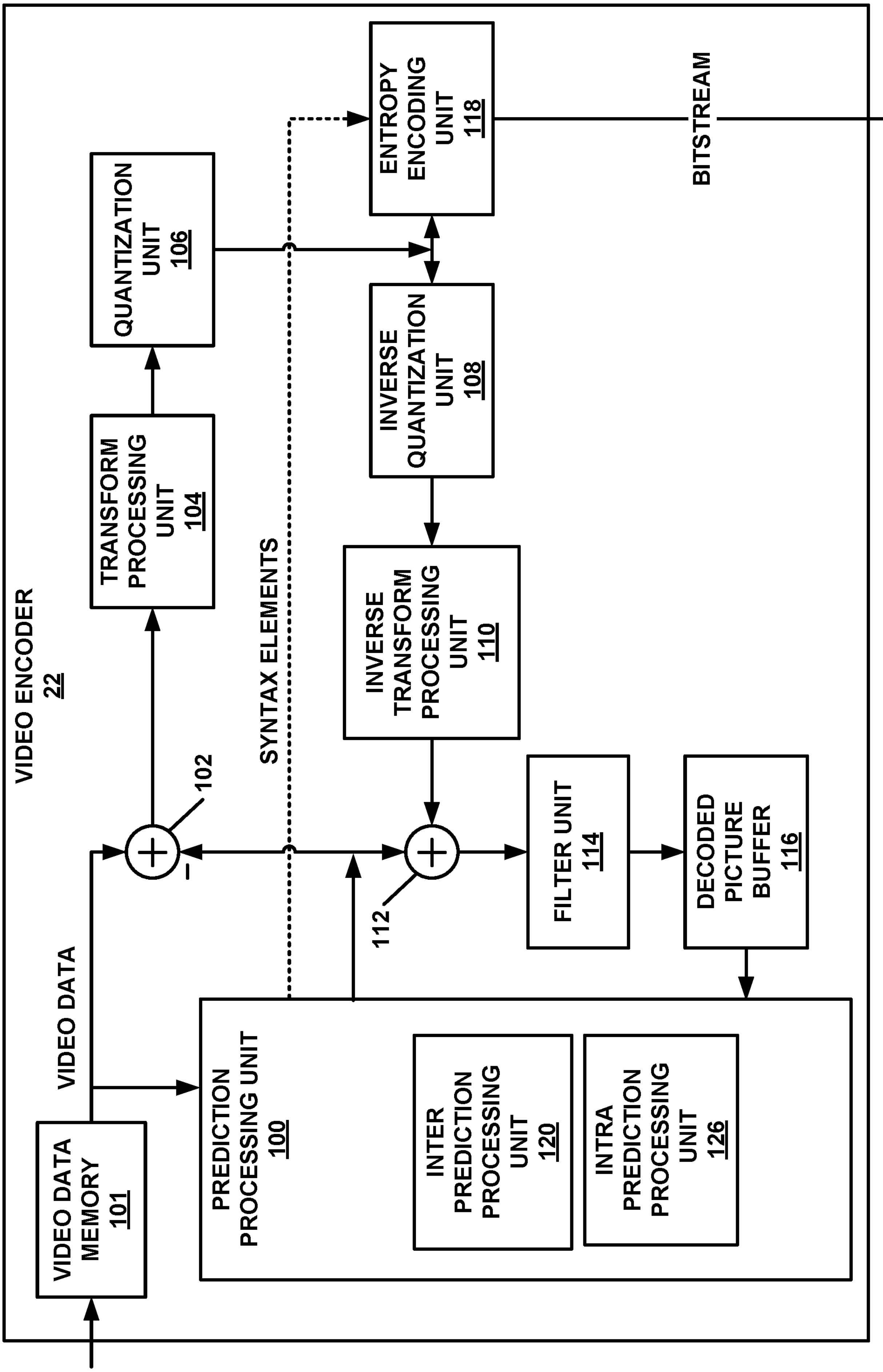

VIDEO ENCODER 22
VIDEO DATA
VIDEO DATA MEMORY 101
PREDICTION PROCESSING UNIT 100
INTER PREDICTION PROCESSING UNIT 120
INTRA PREDICTION PROCESSING UNIT 126
102
–
112
TRANSFORM PROCESSING UNIT 104
QUANTIZATION UNIT 106
SYNTAX ELEMENTS
ENTROPY ENCODING UNIT 118
BITSTREAM
INVERSE QUANTIZATION UNIT 108
INVERSE TRANSFORM PROCESSING UNIT 110
FILTER UNIT 114
DECODED PICTURE BUFFER 116

FIG. 7

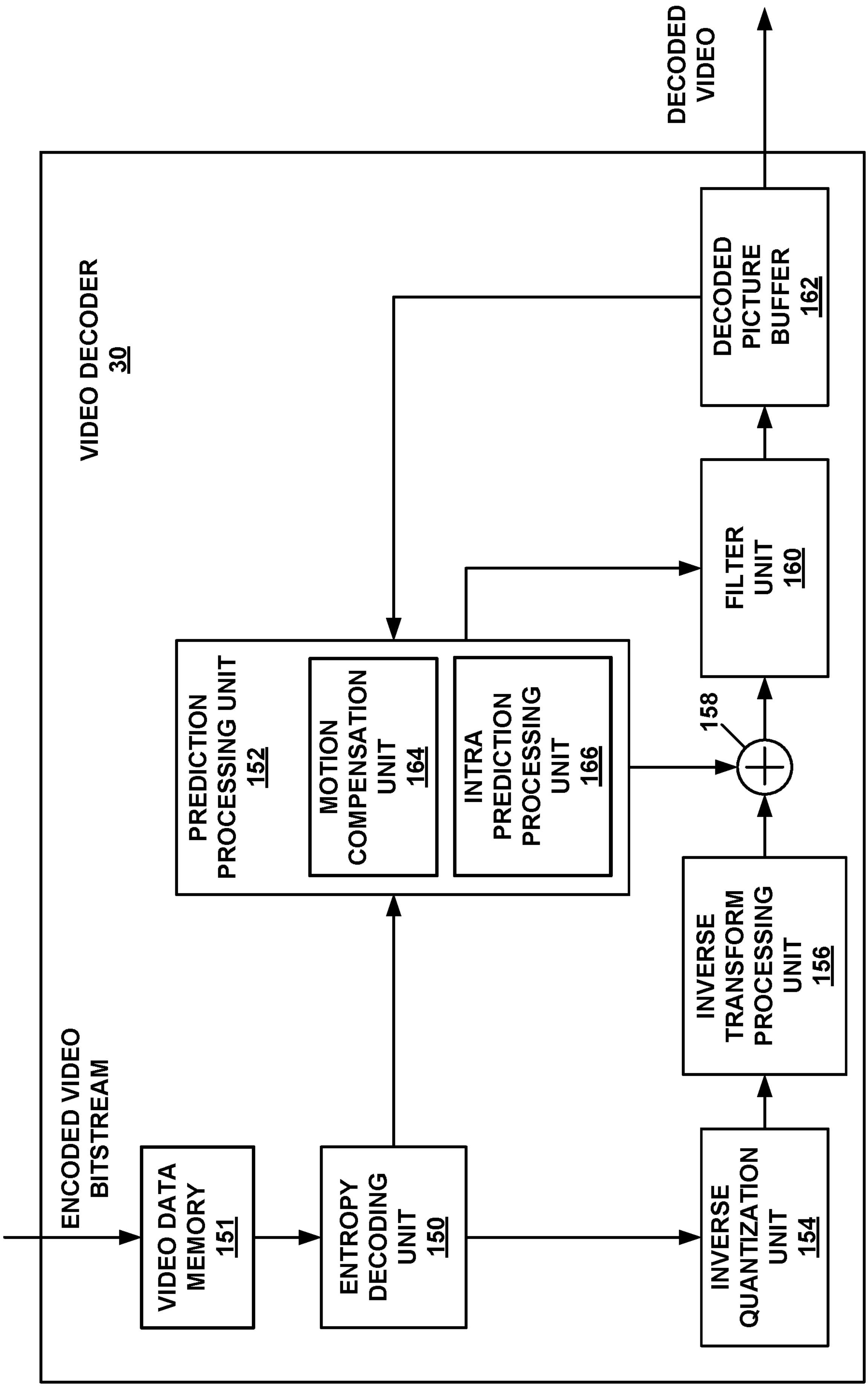

VIDEO DECODER 30
ENCODED VIDEO BITSTREAM
VIDEO DATA MEMORY 151
ENTROPY DECODING UNIT 150
PREDICTION PROCESSING UNIT 152
MOTION COMPENSATION UNIT 164
INTRA PREDICTION PROCESSING UNIT 166
INVERSE QUANTIZATION UNIT 154
INVERSE TRANSFORM PROCESSING UNIT 156
158
FILTER UNIT 160
DECODED PICTURE BUFFER 162
DECODED VIDEO

FIG. 8

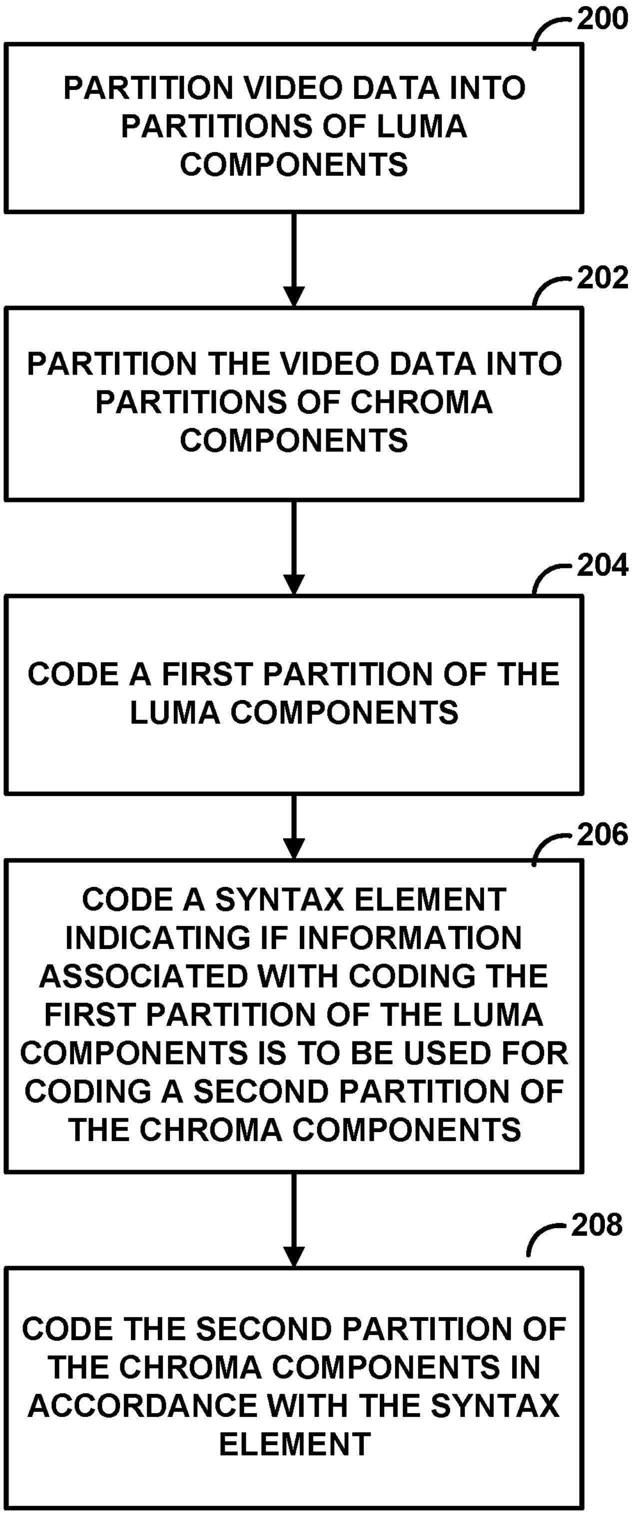

200
PARTITION VIDEO DATA INTO PARTITIONS OF LUMA COMPONENTS
202
PARTITION THE VIDEO DATA INTO PARTITIONS OF CHROMA COMPONENTS
204
CODE A FIRST PARTITION OF THE LUMA COMPONENTS
206
CODE A SYNTAX ELEMENT INDICATING IF INFORMATION ASSOCIATED WITH CODING THE FIRST PARTITION OF THE LUMA COMPONENTS IS TO BE USED FOR CODING A SECOND PARTITION OF THE CHROMA COMPONENTS
208
CODE THE SECOND PARTITION OF THE CHROMA COMPONENTS IN ACCORDANCE WITH THE SYNTAX ELEMENT

FIG. 9

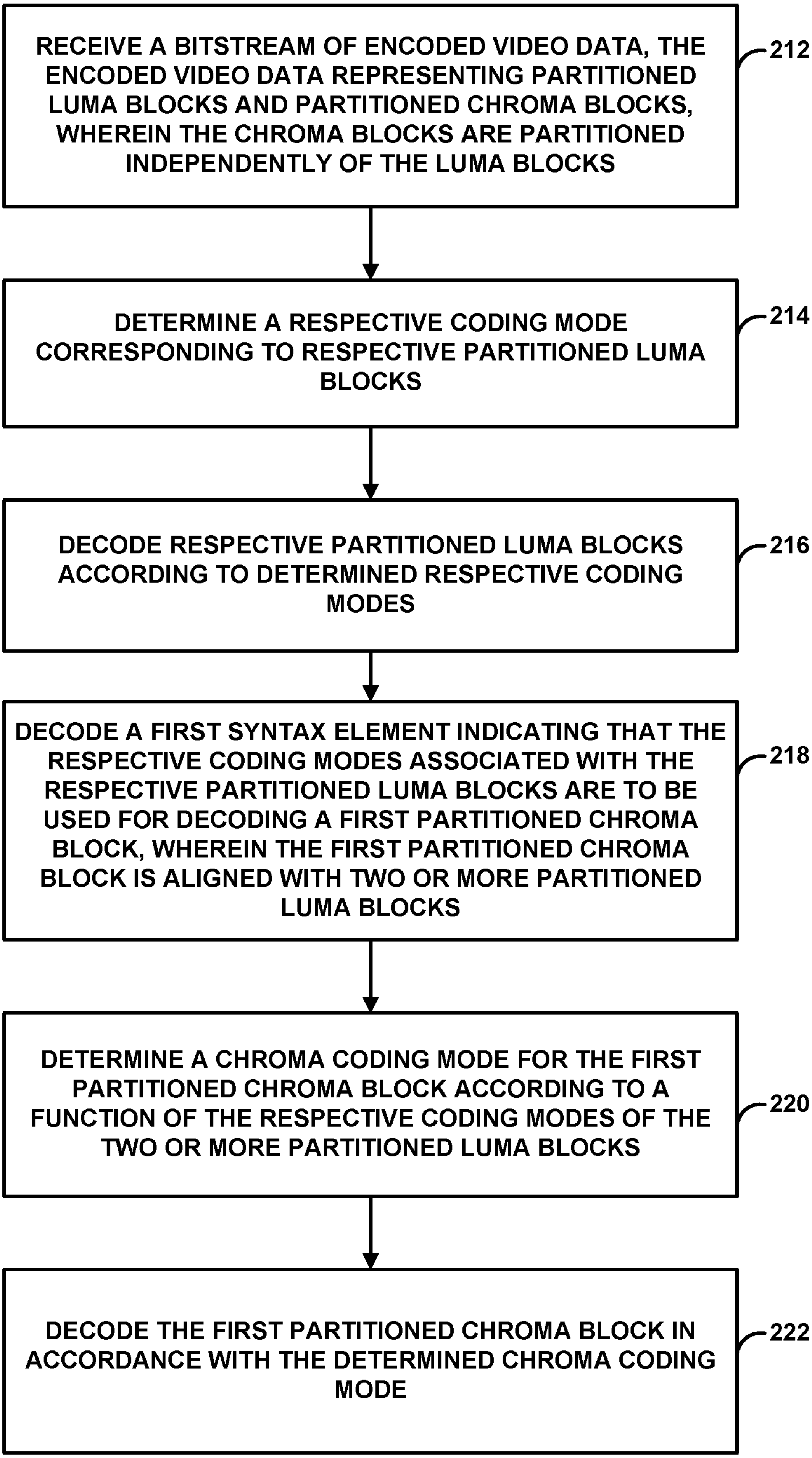

RECEIVE A BITSTREAM OF ENCODED VIDEO DATA, THE ENCODED VIDEO DATA REPRESENTING PARTITIONED LUMA BLOCKS AND PARTITIONED CHROMA BLOCKS, WHEREIN THE CHROMA BLOCKS ARE PARTITIONED INDEPENDENTLY OF THE LUMA BLOCKS
212
DETERMINE A RESPECTIVE CODING MODE CORRESPONDING TO RESPECTIVE PARTITIONED LUMA BLOCKS
214
DECODE RESPECTIVE PARTITIONED LUMA BLOCKS ACCORDING TO DETERMINED RESPECTIVE CODING MODES
216
DECODE A FIRST SYNTAX ELEMENT INDICATING THAT THE RESPECTIVE CODING MODES ASSOCIATED WITH THE RESPECTIVE PARTITIONED LUMA BLOCKS ARE TO BE USED FOR DECODING A FIRST PARTITIONED CHROMA BLOCK, WHEREIN THE FIRST PARTITIONED CHROMA BLOCK IS ALIGNED WITH TWO OR MORE PARTITIONED LUMA BLOCKS
218
DETERMINE A CHROMA CODING MODE FOR THE FIRST PARTITIONED CHROMA BLOCK ACCORDING TO A FUNCTION OF THE RESPECTIVE CODING MODES OF THE TWO OR MORE PARTITIONED LUMA BLOCKS
220
DECODE THE FIRST PARTITIONED CHROMA BLOCK IN ACCORDANCE WITH THE DETERMINED CHROMA CODING MODE
222

FIG. 10

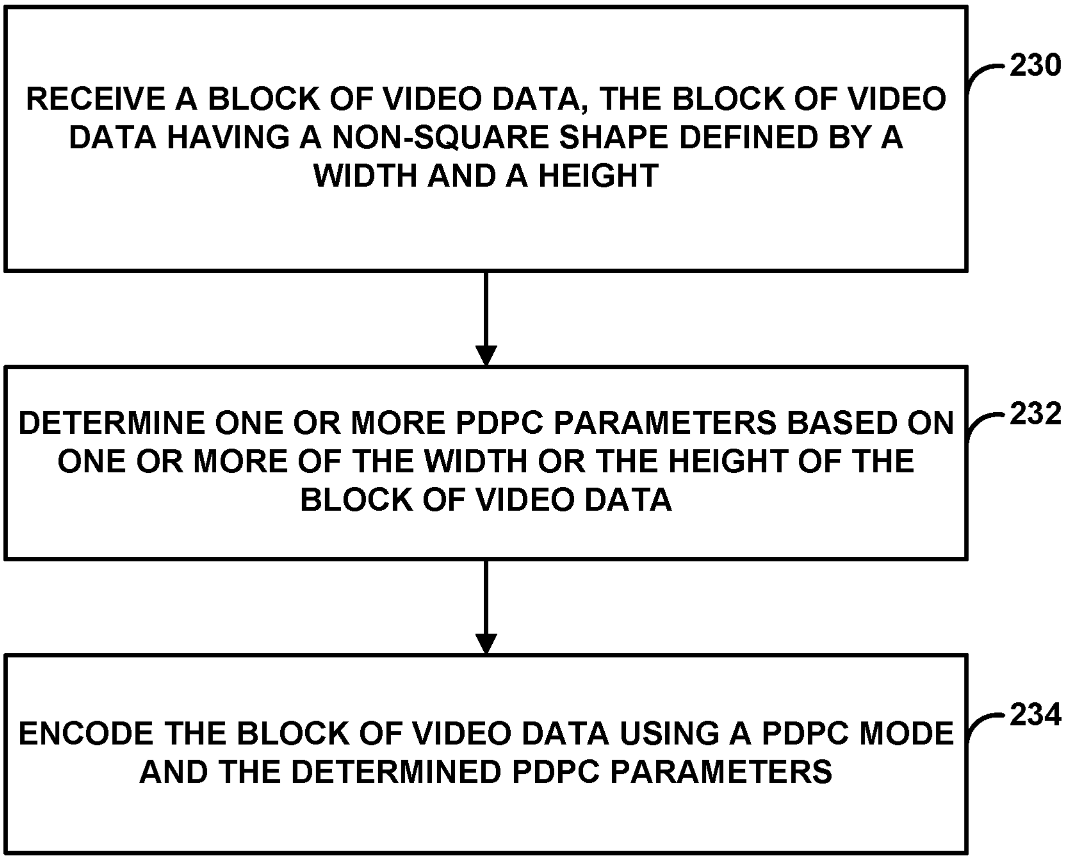

RECEIVE A BLOCK OF VIDEO DATA, THE BLOCK OF VIDEO DATA HAVING A NON-SQUARE SHAPE DEFINED BY A WIDTH AND A HEIGHT
230
DETERMINE ONE OR MORE PDPC PARAMETERS BASED ON ONE OR MORE OF THE WIDTH OR THE HEIGHT OF THE BLOCK OF VIDEO DATA
232
ENCODE THE BLOCK OF VIDEO DATA USING A PDPC MODE AND THE DETERMINED PDPC PARAMETERS
234

FIG. 11

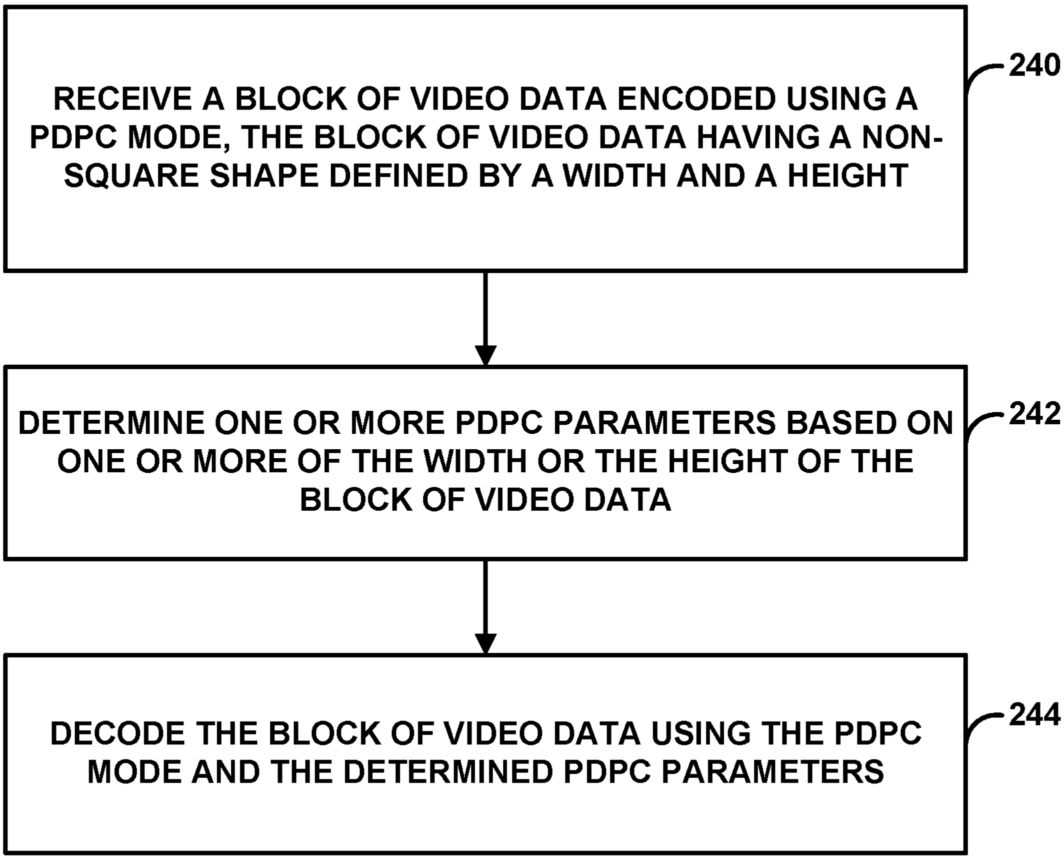

RECEIVE A BLOCK OF VIDEO DATA ENCODED USING A PDPC MODE, THE BLOCK OF VIDEO DATA HAVING A NON-SQUARE SHAPE DEFINED BY A WIDTH AND A HEIGHT
240
DETERMINE ONE OR MORE PDPC PARAMETERS BASED ON ONE OR MORE OF THE WIDTH OR THE HEIGHT OF THE BLOCK OF VIDEO DATA
242
DECODE THE BLOCK OF VIDEO DATA USING THE PDPC MODE AND THE DETERMINED PDPC PARAMETERS
244

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/023378

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/593 H04N19/11 H04N19/167 H04N19/176
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04N

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	SAID A ET AL: "Position dependent intra prediction combination", 113. MPEG MEETING; 19-10-2015 - 23-10-2015; GENEVA; (MOTION PICTURE EXPERT GROUP OR ISO/IEC JTC1/SC29/WG11),, no. m37502, 23 October 2015 (2015-10-23), XP030065870, section 2.1	1-44

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 May 2017	07/06/2017

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Regidor Arenales, R

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	AN J ET AL: "Quadtree plus binary tree structure integration with JEM tools", 2. JVET MEETING; 20-2-2016 - 26-2-2016; SAN DIEGO; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://PHENIX.INT-EVRY.FR/JVET/,, no. JVET-B0023-v2, 20 February 2016 (2016-02-20), XP030150012, page 2, line 14 - line 20 figure 1	1-44
E	WO 2017/058635 A1 (QUALCOMM INC [US]) 6 April 2017 (2017-04-06) paragraph [0041] paragraph [0054] paragraph [0064] paragraph [0073] paragraph [0081] paragraph [0092]	1-5, 8-15, 18-29, 32-39, 42-44

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2017/023378

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2017058635 A1	06-04-2017	US 2017094285 A1	30-03-2017
		WO 2017058635 A1	06-04-2017

(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 108781296 A
(43)申请公布日 2018.11.09

(21)申请号 201780016511.4

(22)申请日 2017.03.21

(30)优先权数据
62/311,265 2016.03.21 US
15/463,474 2017.03.20 US

(85)PCT国际申请进入国家阶段日
2018.09.11

(86)PCT国际申请的申请数据
PCT/US2017/023378 2017.03.21

(87)PCT国际申请的公布数据
WO2017/165395 EN 2017.09.28

(71)申请人 高通股份有限公司
地址 美国加利福尼亚州

(72)发明人 V•谢廖金 赵欣 A•赛义德
M•卡切维奇

(74)专利代理机构 北京律盟知识产权代理有限责任公司 11287
代理人 杨林勳

(51)Int.Cl.
H04N 19/593(2006.01)
H04N 19/11(2006.01)
H04N 19/167(2006.01)
H04N 19/176(2006.01)

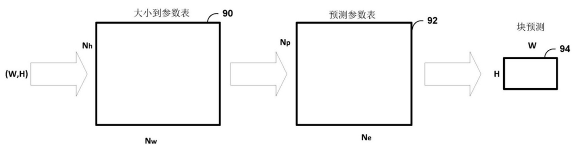
权利要求书6页 说明书27页 附图13页

(54)发明名称

为视频译码中非方形块确定预测参数

(57)摘要

一种解码视频数据的方法，其包括：接收使用位置相依帧内预测组合PDPC模式而编码的视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；基于所述视频数据块的所述宽度或所述高度中的一或多者来确定一或多个PDPC参数；及使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块。



CN 108781296 A

1.一种解码视频数据的方法,所述方法包括:

接收使用预测模式而编码的视频数据块,所述视频数据块具有由宽度及高度界定的非方形形状;

基于所述视频数据块的所述宽度或所述高度中的一或多者来确定用于所述预测模式的一或多个参数;及

使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块。

2.根据权利要求1所述的方法,其中所述预测模式为位置相依帧内预测组合PDPC模式,且其中用于所述预测模式的所述一或多个参数为一或多个PDPC参数。

3.根据权利要求2所述的方法,其中所述一或多个PDPC参数包含一或多个水平相关PDPC参数及一或多个垂直相关PDPC参数,且其中确定所述一或多个PDPC参数包括:

基于所述视频数据块的所述宽度来确定所述一或多个水平相关PDPC参数;及

基于所述视频数据块的所述高度来确定所述一或多个垂直相关PDPC参数。

4.根据权利要求3所述的方法,其中确定所述一或多个水平相关PDPC参数包括依据所述视频数据块的所述宽度来检索一或多个查找表的一或多个条目,且

其中确定所述一或多个垂直相关PDPC参数包括依据所述视频数据块的所述高度来检索所述一或多个查找表的一或多个条目。

5.根据权利要求4所述的方法,其中依据所述视频数据块的所述宽度来检索所述一或多个查找表的所述一或多个条目包括:

基于所述视频数据块的所述宽度来检索第一查找表中的第一索引,所述第一索引指向第二查找表中的第一条目;及

基于所述经检索第一索引来检索所述第二查找表中的所述一或多个水平相关PDPC参数,且

其中依据所述视频数据块的所述高度来检索所述一或多个查找表的所述一或多个条目包括:

基于所述视频数据块的所述高度来检索所述第一查找表中的第二索引,所述第二索引指向所述第二查找表中的第二条目;及

基于所述经检索第二索引来检索所述第二查找表中的所述一或多个垂直相关PDPC参数。

6.根据权利要求2所述的方法,其中所述一或多个PDPC参数包含不水平相关且不垂直相关的一或多个非方向性PDPC参数,且其中确定所述一或多个PDPC参数包括:

基于所述视频数据块的所述宽度及所述高度的函数来确定所述一或多个非方向性PDPC参数。

7.根据权利要求6所述的方法,其中所述函数为以下各者中的一或多者:所述视频数据块的所述宽度及所述高度的最小值、所述视频数据块的所述宽度及所述高度的最大值,或所述视频数据块的所述宽度及所述高度的加权平均值。

8.根据权利要求7所述的方法,其中确定所述一或多个非方向性PDPC参数包括依据所述视频数据块的所述宽度及所述高度来存取一或多个查找表的一或多个条目。

9.根据权利要求1所述的方法,所述方法在无线通信装置上可执行,其中所述装置包括:

存储器，其经配置以存储所述视频数据块；

处理器，其经配置以执行指令以处理存储在所述存储器中的所述视频数据块；及

接收器，其经配置以接收所述视频数据块。

10.根据权利要求9所述的方法，其中所述无线通信装置为移动站，且所述视频数据块是由所述接收器接收且根据蜂窝式通信标准予以调制。

11.一种经配置以解码视频数据的设备，所述设备包括：

存储器，其经配置以存储使用预测模式而编码的视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；及

一或多个处理器，其经配置以进行以下操作：

接收所述视频数据块；

基于所述视频数据块的所述宽度或所述高度中的一或多者来确定用于所述预测模式的一或多个参数；及

使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块。

12.根据权利要求11所述的设备，其中所述预测模式为位置相依帧内预测组合PDPC模式，且其中用于所述预测模式的所述一或多个参数为一或多个PDPC参数。

13.根据权利要求12所述的设备，其中所述一或多个PDPC参数包含一或多个水平相关PDPC参数及一或多个垂直相关PDPC参数，且其中为了确定所述一或多个PDPC参数，所述一或多个处理器经进一步配置以进行以下操作：

基于所述视频数据块的所述宽度来确定所述一或多个水平相关PDPC参数；及

基于所述视频数据块的所述高度来确定所述一或多个垂直相关PDPC参数。

14.根据权利要求13所述的设备，其中为了确定所述一或多个水平相关PDPC参数，所述一或多个处理器经进一步配置以依据所述视频数据块的所述宽度来检索一或多个查找表的一或多个条目，且

其中为了确定所述一或多个垂直相关PDPC参数，所述一或多个处理器经进一步配置以依据所述视频数据块的所述高度来检索所述一或多个查找表的一或多个条目。

15.根据权利要求14所述的设备，其中为了依据所述视频数据块的所述宽度来检索所述一或多个查找表的所述一或多个条目，所述一或多个处理器经进一步配置以进行以下操作：

基于所述视频数据块的所述宽度来检索第一查找表中的第一索引，所述第一索引指向第二查找表中的第一条目；及

基于所述经检索第一索引来检索所述第二查找表中的所述一或多个水平相关PDPC参数，且

其中为了依据所述视频数据块的所述高度来检索所述一或多个查找表的所述一或多个条目，所述一或多个处理器经进一步配置以进行以下操作：

基于所述视频数据块的所述高度来检索所述第一查找表中的第二索引，所述第二索引指向所述第二查找表中的第二条目；及

基于所述经检索第二索引来检索所述第二查找表中的所述一或多个垂直相关PDPC参数。

16.根据权利要求12所述的设备，其中所述一或多个PDPC参数包含不水平相关且不垂

直相关的一或多个非方向性PDPC参数，且其中为了确定所述一或多个PDPC参数，所述一或多个处理器经进一步配置以基于所述视频数据块的所述宽度及所述高度的函数来确定所述一或多个非方向性PDPC参数。

17.根据权利要求16所述的设备，其中所述函数为以下各者中的一或多者：所述视频数据块的所述宽度及所述高度的最小值、所述视频数据块的所述宽度及所述高度的最大值，或所述视频数据块的所述宽度及所述高度的加权平均值。

18.根据权利要求17所述的设备，其中为了确定所述一或多个非方向性PDPC参数，所述一或多个处理器经进一步配置以依据所述视频数据块的所述宽度及所述高度来存取一或多个查找表的一或多个条目。

19.根据权利要求11所述的设备，其中所述设备为无线通信装置，所述设备进一步包括：

接收器，其经配置以接收所述视频数据块。

20.根据权利要求19所述的设备，其中所述无线通信装置为移动站，且所述视频数据块是由所述接收器接收且根据蜂窝式通信标准予以调制。

21.一种经配置以解码视频数据的设备，所述设备包括：

用于接收使用预测模式而编码的视频数据块的装置，所述视频数据块具有由宽度及高度界定的非方形形状；

用于基于所述视频数据块的所述宽度或所述高度中的一或多者来确定用于所述预测模式的一或多个参数的装置；及

用于使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块的装置。

22.根据权利要求21所述的设备，其中所述预测模式为位置相依帧内预测组合PDPC模式，且其中用于所述预测模式的所述一或多个参数为一或多个PDPC参数。

23.一种非暂时性计算机可读存储媒体，其上存储有指令，所述指令在被执行时致使经配置以解码视频数据的装置的一或多个处理器进行以下操作：

接收使用预测模式而编码的视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；

基于所述视频数据块的所述宽度或所述高度中的一或多者来确定用于所述预测模式的一或多个参数；及

使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块。

24.根据权利要求23所述的非暂时性计算机可读存储媒体，其中所述预测模式为位置相依帧内预测组合PDPC模式，且其中用于所述预测模式的所述一或多个参数为一或多个PDPC参数。

25.一种编码视频数据的方法，所述方法包括：

接收视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；

基于所述视频数据块的所述宽度或所述高度中的一或多者来确定用于预测模式的一或多个参数；及

使用所述预测模式及所述经确定的一或多个参数来编码所述视频数据块。

26.根据权利要求25所述的方法，其中所述预测模式为位置相依帧内预测组合PDPC模式，且其中用于所述预测模式的所述一或多个参数为一或多个PDPC参数。

27.根据权利要求26所述的方法，其中所述一或多个PDPC参数包含一或多个水平相关PDPC参数及一或多个垂直相关PDPC参数，且其中确定所述一或多个PDPC参数包括：

基于所述视频数据块的所述宽度来确定所述一或多个水平相关PDPC参数；及

基于所述视频数据块的所述高度来确定所述一或多个垂直相关PDPC参数。

28.根据权利要求27所述的方法，其中确定所述一或多个水平相关PDPC参数包括依据所述视频数据块的所述宽度来检索一或多个查找表的一或多个条目，且

其中确定所述一或多个垂直相关PDPC参数包括依据所述视频数据块的所述高度来检索所述一或多个查找表的一或多个条目。

29.根据权利要求28所述的方法，其中依据所述视频数据块的所述宽度来检索所述一或多个查找表的所述一或多个条目包括：

基于所述视频数据块的所述宽度来检索第一查找表中的第一索引，所述第一索引指向第二查找表中的第一条目；及

基于所述经检索第一索引来检索所述第二查找表中的所述一或多个水平相关PDPC参数，且

其中依据所述视频数据块的所述高度来检索所述一或多个查找表的所述一或多个条目包括：

基于所述视频数据块的所述高度来检索所述第一查找表中的第二索引，所述第二索引指向所述第二查找表中的第二条目；及

基于所述经检索第二索引来检索所述第二查找表中的所述一或多个垂直相关PDPC参数。

30.根据权利要求26所述的方法，其中所述一或多个PDPC参数包含不水平相关且不垂直相关的一或多个非方向性PDPC参数，且其中确定所述一或多个PDPC参数包括：

基于所述视频数据块的所述宽度及所述高度的函数来确定所述一或多个非方向性PDPC参数。

31.根据权利要求30所述的方法，其中所述函数为以下各者中的一或多者：所述视频数据块的所述宽度及所述高度的最小值、所述视频数据块的所述宽度及所述高度的最大值，或所述视频数据块的所述宽度及所述高度的加权平均值。

32.根据权利要求31所述的方法，其中确定所述一或多个非方向性PDPC参数包括依据所述视频数据块的所述宽度及所述高度来存取一或多个查找表的一或多个条目。

33.根据权利要求25所述的方法，所述方法在无线通信装置上可执行，其中所述装置包括：

存储器，其经配置以存储所述视频数据块；

处理器，其经配置以执行指令以处理存储在所述存储器中的所述视频数据块；及

发射器，其经配置以发射所述经编码视频数据块。

34.根据权利要求33所述的方法，其中所述无线通信装置为移动站，且所述经编码视频数据块是由所述发射器发射且根据蜂窝式通信标准予以调制。

35.一种经配置以编码视频数据的设备，所述设备包括：

存储器，其经配置以存储视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；及

一或多个处理器，其经配置以进行以下操作：

接收所述视频数据块；

基于所述视频数据块的所述宽度或所述高度中的一或多者来确定用于预测模式的一或多个参数；及

使用所述预测模式及所述经确定的一或多个参数来编码所述视频数据块。

36.根据权利要求35所述的设备，其中所述预测模式为位置相依帧内预测组合PDPC模式，且其中用于所述预测模式的所述一或多个参数为一或多个PDPC参数。

37.根据权利要求36所述的设备，其中所述一或多个PDPC参数包含一或多个水平相关PDPC参数及一或多个垂直相关PDPC参数，且其中为了确定所述一或多个PDPC参数，所述一或多个处理器经进一步配置以进行以下操作：

基于所述视频数据块的所述宽度来确定所述一或多个水平相关PDPC参数；及

基于所述视频数据块的所述高度来确定所述一或多个垂直相关PDPC参数。

38.根据权利要求37所述的设备，其中为了确定所述一或多个水平相关PDPC参数，所述一或多个处理器经进一步配置以依据所述视频数据块的所述宽度来检索一或多个查找表的一或多个条目，且

其中为了确定所述一或多个垂直相关PDPC参数，所述一或多个处理器经进一步配置以依据所述视频数据块的所述高度来检索所述一或多个查找表的一或多个条目。

39.根据权利要求38所述的设备，其中为了依据所述视频数据块的所述宽度来检索所述一或多个查找表的所述一或多个条目，所述一或多个处理器经进一步配置以进行以下操作：

基于所述视频数据块的所述宽度来检索第一查找表中的第一索引，所述第一索引指向第二查找表中的第一条目；及

基于所述经检索第一索引来检索所述第二查找表中的所述一或多个水平相关PDPC参数，且

其中为了依据所述视频数据块的所述高度来检索所述一或多个查找表的所述一或多个条目，所述一或多个处理器经进一步配置以进行以下操作：

基于所述视频数据块的所述高度来检索所述第一查找表中的第二索引，所述第二索引指向所述第二查找表中的第二条目；及

基于所述经检索第二索引来检索所述第二查找表中的所述一或多个垂直相关PDPC参数。

40.根据权利要求36所述的设备，其中所述一或多个PDPC参数包含不水平相关且不垂直相关的一或多个非方向性PDPC参数，且其中为了确定所述一或多个PDPC参数，所述一或多个处理器经进一步配置以基于所述视频数据块的所述宽度及所述高度的函数来确定所述一或多个非方向性PDPC参数。

41.根据权利要求40所述的设备，其中所述函数为以下各者中的一或多者：所述视频数据块的所述宽度及所述高度的最小值、所述视频数据块的所述宽度及所述高度的最大值，或所述视频数据块的所述宽度及所述高度的加权平均值。

42.根据权利要求41所述的设备，其中为了确定所述一或多个非方向性PDPC参数，所述一或多个处理器经进一步配置以依据所述视频数据块的所述宽度及所述高度来存取一或

多个查找表的一或多个条目。

43.根据权利要求35所述的设备,其中所述设备为无线通信装置,所述设备进一步包括:

发射器,其经配置以发射所述经编码视频数据块。

44.根据权利要求43所述的设备,其中所述无线通信装置为移动站,且所述经编码视频数据块是由所述发射器发射且根据蜂窝式通信标准予以调制。

为视频译码中非方形块确定预测参数

[0001] 本申请案主张2016年3月21日申请的美国临时申请案第62/311,265号的权益，所述临时申请案的全部内容以引用的方式并入本文中。

技术领域

[0002] 本发明涉及视频编码及视频解码。

背景技术

[0003] 数字视频能力可结合到广泛范围的装置中，所述装置包含数字电视、数字直播系统、无线广播系统、个人数字助理(PDA)、膝上型或桌上型计算机、平板计算机、电子书阅读器、数码相机、数字记录装置、数字媒体播放器、视频游戏装置、视频游戏控制台、蜂窝式或卫星无线电电话、所谓的“智能电话”、视频电话会议装置、视频流式处理装置等等。数字视频装置实施视频译码技术，诸如由MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4、Part 10、高级视频译码(AVC)、高效率视频译码(HEVC或H.265)标准及此类标准的扩展所定义的标准中所描述的视频译码技术。视频装置可通过实施此类视频译码技术来更有效地发射、接收、编码、解码及/或存储数字视频信息。

[0004] 视频译码技术包含空间(图片内)预测及/或时间(图片间)预测以缩减或移除为视频序列所固有的冗余。对于基于块的视频译码，可将视频切片(例如，视频帧或视频帧的部分)分割成视频块，其也可被称作树型块、译码单元(CU)及/或译码节点。图片可被称作帧，且参考图片可被称作参考帧。

[0005] 空间或时间预测产生用于待译码块的预测性块。残余数据表示原始经译码块与预测性块之间的像素差。为了进行进一步压缩，可将残余数据从像素域变换为变换域，从而产生残余变换系数，其接着可被量化。可应用熵译码以达成甚至更多的压缩。

发明内容

[0006] 本发明描述用于译码已使用独立亮度及色度分割框架而分割的视频数据的技术。在一些实例中，本发明描述用于在存在对应于色度块的两个或多于两个亮度块时(例如，在两个或多于两个亮度块与色度块共置时)确定如何将来自亮度块的译码信息再用于色度块的技术。

[0007] 在其它实例中，本发明描述用于在视频数据块可被分割成非方形块时确定用于位置相依帧内预测比较(PDPC)模式的参数的技术。在一些实例中，可使用多个查找表来确定PDPC参数，所述查找表包含用于垂直相关参数及水平相关参数的单独表。

[0008] 在本发明的一个实例中，一种解码视频数据的方法包括：接收经编码视频数据的位流，所述经编码视频数据表示经分割亮度块及经分割色度块，其中所述色度块是独立于所述亮度块被分割；确定对应于所述相应经分割亮度块的相应译码模式；根据所述经确定相应译码模式来解码所述相应经分割亮度块；解码指示与所述相应经分割亮度块相关联的所述相应译码模式待用于解码第一经分割色度块的第一语法元素，其中所述第一经分割色

度块与两个或多于两个经分割亮度块对准；根据所述两个或多于两个经分割亮度块的所述相应译码模式的函数而为所述第一经分割色度块确定色度译码模式；及根据所述经确定色度译码模式来解码所述第一经分割色度块。

[0009] 在本发明的另一实例中，一种经配置以解码视频数据的设备包括：存储器，其经配置以存储经编码视频数据的位流；及一或多个处理器，其经配置以进行以下操作：接收经编码视频数据的所述位流，所述经编码视频数据表示经分割亮度块及经分割色度块，其中所述色度块是独立于所述亮度块被分割；确定对应于所述相应经分割亮度块的相应译码模式；根据所述经确定相应译码模式来解码所述相应经分割亮度块；解码指示与所述相应经分割亮度块相关联的所述相应译码模式待用于解码第一经分割色度块的第一语法元素，其中所述第一经分割色度块与两个或多于两个经分割亮度块对准；根据所述两个或多于两个经分割亮度块的所述相应译码模式的函数而为所述第一经分割色度块确定色度译码模式；及根据所述经确定色度译码模式来解码所述第一经分割色度块。

[0010] 在本发明的另一实例中，一种经配置以解码视频数据的设备包括：用于接收经编码视频数据的位流的装置，所述经编码视频数据表示经分割亮度块及经分割色度块，其中所述色度块是独立于所述亮度块被分割；用于确定对应于所述相应经分割亮度块的相应译码模式的装置；用于根据所述经确定相应译码模式来解码所述相应经分割亮度块的装置；用于解码指示与所述相应经分割亮度块相关联的所述相应译码模式待用于解码第一经分割色度块的第一语法元素的装置，其中所述第一经分割色度块与两个或多于两个经分割亮度块对准；用于根据所述两个或多于两个经分割亮度块的所述相应译码模式的函数而为所述第一经分割色度块确定色度译码模式的装置；及用于根据所述经确定色度译码模式来解码所述第一经分割色度块的装置。

[0011] 在另一实例中，本发明描述一种非暂时性计算机可读存储媒体，其存储指令，所述指令在被执行时致使经配置以解码视频数据的一或多个处理器进行以下操作：接收经编码视频数据的位流，所述经编码视频数据表示经分割亮度块及经分割色度块，其中所述色度块是独立于所述亮度块被分割；确定对应于所述相应经分割亮度块的相应译码模式；根据所述经确定相应译码模式来解码所述相应经分割亮度块；解码指示与所述相应经分割亮度块相关联的所述相应译码模式待用于解码第一经分割色度块的第一语法元素，其中所述第一经分割色度块与两个或多于两个经分割亮度块对准；根据所述两个或多于两个经分割亮度块的所述相应译码模式的函数而为所述第一经分割色度块确定色度译码模式；及根据所述经确定色度译码模式来解码所述第一经分割色度块。

[0012] 在本发明的另一实例中，一种解码视频数据的方法包括：接收使用位置相依帧内预测组合(PDPC)模式而编码的视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；基于所述视频数据块的所述宽度或所述高度中的一或多者来确定一或多个PDPC参数；及使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块。

[0013] 在本发明的另一实例中，一种经配置以解码视频数据的设备包括：存储器，其经配置以存储使用PDPC模式而编码的视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；及一或多个处理器，其经配置以进行以下操作：接收所述视频数据块；基于所述视频数据块的所述宽度或所述高度中的一或多者来确定一或多个PDPC参数；及使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块。

[0014] 在本发明的另一实例中，一种经配置以解码视频数据的设备包括：用于接收使用PDPC模式而编码的视频数据块的装置，所述视频数据块具有由宽度及高度界定的非方形形状；用于基于所述视频数据块的所述宽度或所述高度中的一或多者来确定一或多个PDPC参数的装置；及用于使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块的装置。

[0015] 在另一实例中，本发明描述一种非暂时性计算机可读存储媒体，其存储指令，所述指令在被执行时致使经配置以解码视频数据的装置的一或多个处理器进行以下操作：接收使用PDPC模式而编码的视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；基于所述视频数据块的所述宽度或所述高度中的一或多者来确定一或多个PDPC参数；及使用所述PDPC模式及所述经确定PDPC参数来解码所述视频数据块。

[0016] 在本发明的另一实例中，一种编码视频数据的方法包括：接收视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；基于所述视频数据块的所述宽度或所述高度中的一或多者来确定一或多个PDPC参数；及使用PDPC模式及所述经确定PDPC参数来编码所述视频数据块。

[0017] 在本发明的另一实例中，一种经配置以编码视频数据的设备包括：存储器，其经配置以存储视频数据块，所述视频数据块具有由宽度及高度界定的非方形形状；及一或多个处理器，其经配置以进行以下操作：接收所述视频数据块；基于所述视频数据块的所述宽度或所述高度中的一或多者来确定一或多个PDPC参数；及使用PDPC模式及所述经确定PDPC参数来编码所述视频数据块。

[0018] 附图及以下描述中阐述本发明的一或多个方面的细节。本发明中所描述的技术的其它特征、目标及优点将从描述及附图且从权利要求书显而易见。

附图说明

[0019] 图1为绘示经配置以实施本发明的技术的实例视频编码及解码系统的框图。

[0020] 图2A为绘示使用四元树加二元树(QTBT)结构的块分割的实例的概念图。

[0021] 图2B为绘示对应于使用图2A的QTBT结构的块分割的实例树型结构的概念图。

[0022] 图3为绘示根据本发明的技术的亮度及色度相对分割的实例的概念图。

[0023] 图4为绘示根据本发明的技术的亮度及色度相对分割的另一实例的概念图。

[0024] 图5A绘示根据本发明的技术使用未经滤波参考的4×4块的预测。

[0025] 图5B绘示根据本发明的技术使用经滤波参考的4×4块的预测。

[0026] 图6为绘示根据本发明的一个实例的用于确定矩形块中使用的预测参数集合的嵌套表的使用的概念图。

[0027] 图7为绘示经配置以实施本发明的技术的视频编码器的实例的框图。

[0028] 图8为绘示经配置以实施本发明的技术的视频解码器的实例的框图。

[0029] 图9为绘示根据本发明的技术的视频译码器的实例操作的流程图。

[0030] 图10为绘示根据本发明的技术的视频解码器的实例操作的流程图。

[0031] 图11为绘示根据本发明的技术的视频编码器的实例操作的流程图。

[0032] 图12为绘示根据本发明的技术的视频解码器的实例操作的流程图。

具体实施方式

[0033] 根据一些视频块分割技术，视频数据的色度块是独立于视频数据的亮度块被分割，使得一些色度块可不与单一对应亮度块直接对准。因而，变得难以将与亮度块相关的语法元素再用于色度块，这是由于亮度块与色度块之间可不存在一对一对应性。本发明描述用于在亮度及色度块被独立地分割的情形中使用对应于视频数据的亮度块的信息(例如，语法元素)来译码视频数据的色度块的技术。

[0034] 本发明还描述用于确定用于位置相依帧内预测组合(PDPC)译码模式的译码参数的技术。在一个实例中，本发明描述用于为分割成非方形块(例如，非方形、矩形块)的视频块确定PDPC参数的技术。

[0035] 图1为绘示可经配置以执行本发明的技术的实例视频编码及解码系统10的框图。如图1所展示，系统10包含源装置12，其提供稍后将由目的地装置14解码的经编码视频数据。具体地说，源装置12经由计算机可读媒体16将视频数据提供到目的地装置14。源装置12及目的地装置14可包括广泛范围的装置中的任一者，包含桌上型计算机、笔记本(例如，膝上型)计算机、平板计算机、机顶盒、电话手机(诸如所谓的“智能”电话(或更一般化地为移动站))、平板计算机、电视、相机、显示装置、数字媒体播放器、视频游戏控制台、视频流式处理装置等等。移动站可为能够经由无线网络而通信的任何装置。在一些状况下，源装置12及目的地装置14可经装备以用于无线通信。因此，源装置12及目的地装置14可为无线通信装置(例如，移动站)。源装置12为实例视频编码装置(即，用于编码视频数据的装置)。目的地装置14为实例视频解码装置(即，用于解码视频数据的装置)。

[0036] 在图1的实例中，源装置12包含视频源18、经配置以存储视频数据的存储媒体20、视频编码器22，及输出接口24。目的地装置14包含输入接口26、经配置以存储经编码视频数据的存储媒体28、视频解码器30，及显示装置32。在其它实例中，源装置12及目的地装置14可包含其它组件或布置。举例来说，源装置12可从外部视频源(诸如外部相机)接收视频数据。同样地，目的地装置14可与外部显示装置介接，而非包含集成显示装置32。

[0037] 图1的所绘示系统10仅仅为一个实例。用于处理及/或译码(例如，编码及/或解码)视频数据的技术可由任何数字视频编码及/或解码装置执行。尽管本发明的技术通常是由视频编码装置及/或视频解码装置执行，但所述技术也可由视频编码器/解码器(通常被称作“编解码器(CODEC)”)执行。源装置12及目的地装置14仅仅为源装置12产生经译码视频数据以供发射到目的地装置14的此类译码装置的实例。在一些实例中，源装置12及目的地装置14可以基本上对称方式而操作，使得源装置12及目的地装置14中的每一者包含视频编码及解码组件。因此，系统10可支持源装置12与目的地装置14之间的单向或双向视频发射，例如，用于视频流式处理、视频回放、视频广播或视频电话。

[0038] 源装置12的视频源18可包含视频捕获装置，诸如摄像机、含有经先前捕获视频的视频存档，及/或用以从视频内容提供者接收视频数据的视频馈送接口。作为另外替代方案，视频源18可产生基于计算机图形的数据作为源视频，或实况视频、经存档视频与经计算机产生视频的组合。源装置12可包括经配置以存储视频数据的一或多个数据存储媒体(例如，存储媒体20)。本发明中所描述的技术可大体上适用于视频译码，且可应用于无线及/或有线应用。在每一状况下，可由视频编码器22编码经捕获、经预捕获或经计算机产生视频。

输出接口24可将经编码视频信息(例如,经编码视频数据的位流)输出到计算机可读媒体16。

[0039] 目的地装置14可经由计算机可读媒体16接收待解码的经编码视频数据。计算机可读媒体16可包括能够将经编码视频数据从源装置12移动到目的地装置14的任何类型的媒体或装置。在一些实例中,计算机可读媒体16包括用以使源装置12能够实时地将经编码视频数据直接发射到目的地装置14的通信媒体。可根据通信标准(诸如无线通信协议)来调制经编码视频数据,且将其发射到目的地装置14。通信媒体可包括任何无线或有线通信媒体,诸如射频(RF)频谱或一或多个物理传输线。通信媒体可形成基于数据包的网络(诸如局域网、广域网,或诸如因特网的全局网络)的部分。通信媒体可包含路由器、交换机、基站,或可有用于促进从源装置12到目的地装置14的通信的任何其它设备。目的地装置14可包括经配置以存储经编码视频数据及经解码视频数据的一或多个数据存储媒体。

[0040] 在一些实例中,可将经编码数据从输出接口24输出到存储装置。相似地,可由输入接口从存储装置存取经编码数据。存储装置可包含多种分布式或本地存取数据存储媒体中的任一者,诸如硬盘驱动器、蓝光光盘、DVD、CD-ROM、闪速存储器、易失性或非易失性存储器,或用于存储经编码视频数据的任何其它合适数字存储媒体。在另外实例中,存储装置可对应于文件服务器或可存储由源装置12产生的经编码视频的另一中间存储装置。目的地装置14可经由流式处理或下载而从存储装置存取经存储视频数据。文件服务器可为能够存储经编码视频数据且将所述经编码视频数据发射到目的地装置14的任何类型的服务器。实例文件服务器包含web服务器(例如,用于网站)、FTP服务器、网络连接存储(NAS)装置,或本地磁盘驱动器。目的地装置14可经由任何标准数据连接(包含因特网连接)来存取经编码视频数据。此数据连接可包含适合于存取存储在文件服务器上的经编码视频数据的无线信道(例如,Wi-Fi连接)、有线连接(例如,DSL、电缆调制解调器等等)或此两者的组合。来自存储装置的经编码视频数据的发射可为流式处理发射、下载发射或其组合。

[0041] 本发明中所描述的技术可应用于视频译码以支持多种多媒体应用中的任一者,诸如空中电视广播、有线电视发射、卫星电视发射、因特网流式处理视频发射(诸如HTTP动态自适应流式处理(DASH))、编码到数据存储媒体上的数字视频、存储在数据存储媒体上的数字视频的解码,或其它应用。在一些实例中,系统10可经配置以支持单向或双向视频发射以支持诸如视频流式处理、视频回放、视频广播及/或视频电话的应用。

[0042] 计算机可读媒体16可包含:暂时性媒体,诸如无线广播或有线网络发射;或存储媒体(即,非暂时性存储媒体),诸如硬盘、闪存驱动器、压缩光盘、数字视频光盘、蓝光光盘或其它计算机可读媒体。在一些实例中,网络服务器(未展示)可从源装置12接收经编码视频数据,且例如经由网络发射将经编码视频数据提供到目的地装置14。相似地,媒体生产设施(诸如光盘冲压设施)的计算装置可从源装置12接收经编码视频数据且生产含有经编码视频数据的光盘。因此,在各种实例中,计算机可读媒体16可被理解为包含各种形式的一或多个计算机可读媒体。

[0043] 目的地装置14的输入接口26从计算机可读媒体16接收信息。计算机可读媒体16的信息可包含由视频编码器22的视频编码器22定义的语法信息,所述语法信息也是由视频解码器30使用,所述语法信息包含描述块及其它经译码单元(例如,图片群组(GOP))的特性及/或处理的语法元素。存储媒体28可存储由输入接口26接收的经编码视频数据。显示装置

32将经解码视频数据显示给用户，且可包括多种显示装置中的任一者，诸如阴极射线管(CRT)、液晶显示器(LCD)、等离子体显示器、有机发光二极管(OLED)显示器，或另一类型的显示装置。

[0044] 视频编码器22及视频解码器30各自可被实施为多种合适视频编码器及/或视频解码器电路系统中的任一者，诸如一或多个微处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、离散逻辑、软件、硬件、固件或其任何组合。在技术是部分地以软件予以实施时，装置可将用于软件的指令存储在合适非暂时性计算机可读媒体中，且在硬件中使用一或多个处理器来执行指令以执行本发明的技术。视频编码器22及视频解码器30中的每一者可包含在一或多个编码器或解码器中，所述一或多个编码器或解码器中的任一者可被集成为相应装置中的组合式编解码器的部分。

[0045] 在一些实例中，视频编码器22及视频解码器30可根据视频译码标准而操作。实例视频译码标准包含但不限于ITU-T H.261、ISO/IEC MPEG-1Visual、ITU-T H.262或ISO/IEC MPEG-2Visual、ITU-T H.263、ISO/IEC MPEG-4Visual及ITU-T H.264(也被称为ISO/IEC MPEG-4AVC)，包含其可伸缩视频译码(SVC)及多视图视频译码(MVC)扩展。另外，ITU-T视频译码专家团体(VCEG)及ISO/IEC动画专家团体(MPEG)的视频译码联合协作小组(JCT-VC)已开发新视频译码标准，即，高效率视频译码(HEVC)或ITU-T H.265，包含其范围及屏幕内容译码扩展、3D视频译码(3D-HEVC)以及多视图扩展(MV-HEVC)及可伸缩扩展(SHVC)。

[0046] 在其它实例中，视频编码器22及视频解码器30可经配置以根据其它视频译码技术及/或标准(包含正由联合视频探索小组(JVET)探索的新视频译码技术)而操作。在本发明的一些实例中，视频编码器22及视频解码器30可经配置以根据使用独立亮度及色度分割使得视频数据的亮度及色度块无需对准的视频译码技术而操作。此类分割技术可导致色度块在图片的特定位置内不对准于单一亮度块的情形。在本发明的其它实例中，视频编码器22及视频解码器30可经配置以根据使用允许非方形块的分割框架的视频译码技术而操作。

[0047] 根据本发明的技术，如下文将更详细地所描述，视频解码器30可经配置以进行以下操作：接收经编码视频数据的位流，经编码视频数据表示经分割亮度块及经分割色度块，其中色度块是独立于亮度块被分割；确定对应于相应经分割亮度块的相应译码模式；根据经确定相应译码模式来解码相应经分割亮度块；解码指示与相应经分割亮度块相关联的相应译码模式待用于解码第一经分割色度块的第一语法元素，其中第一经分割色度块与两个或多于两个经分割亮度块对准；根据两个或多于两个经分割亮度块的相应译码模式的函数而为第一经分割色度块确定色度译码模式；及根据经确定色度译码模式来解码第一经分割色度块。视频编码器22可经配置以执行与视频解码器30的技术互逆的技术。在一些实例中，视频编码器22可经配置以基于两个或多于两个经分割亮度块的相应译码模式的函数来产生指示色度块是否将再用来自两个或多于两个亮度块的译码模式信息的语法元素。

[0048] 在HEVC及其它视频译码规范中，视频序列通常包含一系列图片。图片也可被称作“帧”。图片可包含三个样本阵列，被表示为S_L、S_{Cb}及S_{Cr}。S_L为亮度样本的二维阵列(例如，块)。S_{Cb}为Cb彩度样本的二维阵列。S_{Cr}为Cr彩度样本的二维阵列。彩度样本也可在本文中被称作“色度”样本。在其它情况下，图片可为单色的，且可仅包含亮度样本阵列。

[0049] 为了产生图片的经编码表示(例如，经编码视频位流)，视频编码器22可产生译码树型单元(CTU)集合。所述CTU中的每一者可包括亮度样本的译码树型块、色度样本的两个

对应译码树型块，及用以译码所述译码树型块的样本的语法结构。在单色图片或具有三个单独颜色平面的图片中，CTU可包括单一译码树型块及用以译码所述译码树型块的样本的语法结构。译码树型块可为样本的N×N块。CTU也可被称作“树型块”或“最大译码单元”(LCU)。HEVC的CTU可大致地类似于诸如H.264/AVC的其它标准的宏块。然而，CTU未必限于特定大小，且可包含一或多个译码单元(CU)。切片可包含按光栅扫描次序连续地排序的整数数目个CTU。

[0050] 为了产生经译码CTU，视频编码器22可对CTU的译码树型块递归地执行四元树分割以将译码树型块划分成译码块，因此名称为“译码树型单元”。译码块为样本的N×N块。CU可包括具有亮度样本阵列、Cb样本阵列及Cr样本阵列的图片的亮度样本的译码块及色度样本的两个对应译码块，及用以译码所述译码块的样本的语法结构。在单色图片或具有三个单独颜色平面的图片中，CU可包括单一译码块及用以译码所述译码块的样本的语法结构。

[0051] 视频编码器22可将CU的译码块分割成一或多个预测块。预测块为被应用相同预测的样本的矩形(即，方形或非方形)块。CU的预测单元(PU)可包括亮度样本的预测块、色度样本的两个对应预测块，及用以预测所述预测块的语法结构。在单色图片或具有三个单独颜色平面的图片中，PU可包括单一预测块及用以预测所述预测块的语法结构。视频编码器22可针对CU的每一PU的预测块(例如，亮度、Cb及Cr预测块)产生预测性块(例如，亮度、Cb及Cr预测性块)。

[0052] 视频编码器22可使用帧内预测或帧间预测以产生用于PU的预测性块。如果视频编码器22使用帧内预测以产生PU的预测性块，那么视频编码器22可基于包含PU的图片的经解码样本来产生PU的预测性块。

[0053] 在视频编码器22产生用于CU的一或多个PU的预测性块(例如，亮度、Cb及Cr预测性块)之后，视频编码器22可产生用于CU的一或多个残余块。作为一个实例，视频编码器22可产生用于CU的亮度残余块。CU的亮度残余块中的每一样本指示CU的预测性亮度块中的一者中的亮度样本与CU的原始亮度译码块中的对应样本之间的差。另外，视频编码器22可产生用于CU的Cb残余块。在色度预测的一个实例中，CU的Cb残余块中的每一样本可指示CU的预测性Cb块中的一者中的Cb样本与CU的原始Cb译码块中的对应样本之间的差。视频编码器22还可产生用于CU的Cr残余块。CU的Cr残余块中的每一样本可指示CU的预测性Cr块中的一者中的Cr样本与CU的原始Cr译码块中的对应样本之间的差。然而，应理解，可使用用于色度预测的其它技术。

[0054] 此外，视频编码器22可使用四元树分割以将CU的残余块(例如，亮度、Cb及Cr残余块)分解成一或多个变换块(例如，亮度、Cb及Cr变换块)。变换块为被应用相同变换的样本的矩形(例如，方形或非方形)块。CU的变换单元(TU)可包括亮度样本的变换块、色度样本的两个对应变换块，及用以变换所述变换块样本的语法结构。因此，CU的每一TU可具有亮度变换块、Cb变换块及Cr变换块。TU的亮度变换块可为CU的亮度残余块的子块。Cb变换块可为CU的Cb残余块的子块。Cr变换块可为CU的Cr残余块的子块。在单色图片或具有三个单独颜色平面的图片中，TU可包括单一变换块及用以变换所述变换块的样本的语法结构。

[0055] 视频编码器22可将一或多个变换应用于TU的变换块以产生用于TU的系数块。举例来说，视频编码器22可将一或多个变换应用于TU的亮度变换块以产生用于TU的亮度系数块。系数块可为变换系数的二维阵列。变换系数可为标量。视频编码器22可将一或多个变换

应用于TU的Cb变换块以产生用于TU的Cb系数块。视频编码器22可将一或多个变换应用于TU的Cr变换块以产生用于TU的Cr系数块。

[0056] 在产生系数块(例如,亮度系数块、Cb系数块或Cr系数块)之后,视频编码器22可量化系数块。量化通常是指变换系数经量化以可能地缩减用以表示变换系数的数据的量而提供进一步压缩的过程。在视频编码器22量化系数块之后,视频编码器22可熵编码指示经量化变换系数的语法元素。举例来说,视频编码器22可对指示经量化变换系数的语法元素执行上下文自适应二进制算术译码(CABAC)。

[0057] 视频编码器22可输出包含形成经译码图片及关联数据的表示的位序列的位流。因此,位流包括视频数据的经编码表示。位流可包括网络抽象层(NAL)单元序列。NAL单元为含有NAL单元中的数据的类型的指示及含有所述数据的呈在必要时散置有仿真预防位的原始字节序列有效负载(RBSP)的形式的字节的语法结构。NAL单元中的每一者可包含NAL单元标头且封装RBSP。NAL单元标头可包含指示NAL单元类型码的语法元素。由NAL单元的NAL单元标头指定的NAL单元类型码指示NAL单元的类型。RBSP可为含有封装在NAL单元内的整数数目个字节的语法结构。在一些情况下,RBSP包含零个位。

[0058] 视频解码器30可接收由视频编码器22产生的经编码视频位流。另外,视频解码器30可分析位流以从位流获得语法元素。视频解码器30可至少部分地基于从位流获得的语法元素来重构视频数据的图片。用以重构视频数据的过程可与由视频编码器22执行的过程大体上互逆。举例来说,视频解码器30可使用PU的运动向量以确定用于当前CU的PU的预测性块。另外,视频解码器30可反量化当前CU的TU的系数块。视频解码器30可对系数块执行反变换以重构当前CU的TU的变换块。视频解码器30可通过将用于当前CU的PU的预测性块的样本与当前CU的TU的变换块的对应样本相加来重构当前CU的译码块。通过重构用于图片的每一CU的译码块,视频解码器30可重构图片。

[0059] 在一些实例视频编解码器框架(诸如HEVC的四元树分割框架)中,联合地执行将视频数据分割成用于颜色分量的块(例如,亮度块及色度块)。即,在一些实例中,以相同方式分割亮度块及色度块,使得不多于一个亮度块对应于图片内的特定位置中的色度块。在一个实例中,视频数据块的分割区可被进一步划分成子块。与视频块或视频块的分割区相关的信息(例如,样本值及指示视频块将被如何译码的语法元素)存储在子块级处。或,更一般化地,与视频块或视频块的分割区相关的信息可与视频数据块的一或多个代表性位置(例如,对应于任何样本或子样本)相关地被存储。举例来说,如果分割区为16×16像素,且分割区中的每一子块为4×4像素,那么在分割区中存在16个子块。信息是以子块粒度(在此实例中为4×4)被存储,且所有16个子块可具有相同信息。

[0060] 在本发明的上下文中,术语“分割区”、“块”及“经分割块”可被互换地使用。一般来说,块为被执行视频译码的样本(例如,亮度或色度样本)群组。在本发明的上下文中,“子块”为块的具有存储用于块的译码模式信息的关联存储器位置的分区。

[0061] 视频编码器22及视频解码器30可分配存储器中的位置以用于存储用于每一代表性位置(例如,子块)的信息。在一些实例中,信息的值(例如,用于特定译码模式的特定语法元素的值)可存储在与每一代表性位置(例如,子块)相关联的单独存储器位置中。在其它实例中,信息可经一次存储以用于分割区的多个代表性位置(例如,子块)中的一者。分割区的其它子块的存储器位置可包含指向存储信息的实际值的存储器位置的指针。下文将参考子

块来描述本发明的技术,但应理解,可使用块的任何代表性位置。

[0062] 如上文所提及,存储在子块级处的信息可为用以对分割区执行译码处理的任何信息。此类信息可为经发信语法信息或经派生补充信息。经派生补充信息的一个实例可为从与译码亮度块相关的信息派生的用以译码色度块的信息。供在HEVC中使用的经派生补充信息的一个实例为直接模式信息,其中亮度帧内预测信息(例如,帧内预测方向)用于色度预测而不发信用于色度块的帧内预测方向自身。信息的其它实例可为模式决策,诸如帧内预测或帧间预测、帧内预测方向、运动信息等等。

[0063] 在比较亮度及色度分割区大小时,可考虑色度颜色格式(例如,色度子取样格式),诸如4:4:4、4:2:2、4:2:0。举例来说,如果亮度分割区为16×16像素,那么对应或共置型色度分割区对于4:2:0颜色格式为8×8,且对于4:4:4色度颜色格式为16×16。分割区未必为方形,且可为例如矩形形状。因而,对于4:2:0色度子取样格式,亮度及色度分割区将不为相同的大小。然而,在亮度及色度块被联合地分割时,所得分割仍产生对应于任何特定色度块的仅一个亮度块。

[0064] 联合视频探索小组(JVET)当前正在研究四元树加二元树(QTBT)分割结构。在J.An等人的“Block partitioning structure for next generation video coding”(国际电信联盟,COM16-C966,2015年9月(在下文中为“VCEG提案COM16-C966”))中,针对除了HEVC以外的未来视频译码标准描述QTBT分割技术。模拟已展示所提议的QTBT结构可比HEVC中使用的四元树结构更有效。

[0065] 在VCEG提案COM16-C966中所描述的QTBT结构中,首先使用四元树分割技术来分割CTB,其中一个节点的四元树拆分可被反复直到节点达到最小允许四元树叶节点大小。可通过语法元素MinQTSize的值向视频解码器30指示最小允许四元树叶节点大小。如果四元树叶节点大小不大于最大允许二元树根节点大小(例如,如由语法元素MaxBTSize所指示),那么可使用二元树分割来进一步分割四元树叶节点。一个节点的二元树分割可被反复直到节点达到最小允许二元树叶节点大小(例如,如由语法元素MinBTSize所指示)或最大允许二元树深度(例如,如由语法元素MaxBTDepth所指示)。VCEG提案COM16-C966使用术语“CU”来指二元树叶节点。在VCEG提案COM16-C966中,CU用于预测(例如,帧内预测、帧间预测等等)及变换而无任何进一步分割。一般来说,根据QTBT技术,对于二元树拆分存在两种拆分类型:对称水平拆分及对称垂直拆分。在每一状况下,通过顺着中间水平地或垂直地划分块来拆分块。这不同于四元树分割,四元树分割将块划分成四个块。

[0066] 在QTBT分割结构的一个实例中,CTU大小经设置为128×128(例如,128×128亮度块及两个对应64×64色度块),MinQTSize经设置为16×16,MaxBTSize经设置为64×64,MinBTSize(对于宽度及高度两者)经设置为4,且MaxBTDepth经设置为4。四元树分割首先应用于CTU以产生四元树叶节点。四元树叶节点可具有从16×16(即,MinQTSize为16×16)到128×128(即,CTU大小)的大小。根据QTBT分割的一个实例,如果叶四元树节点为128×128,那么叶四元树节点不能由二元树进一步拆分,这是由于叶四元树节点的大小超过MaxBTSize(即,64×64)。否则,叶四元树节点由二元树进一步分割。因此,四元树叶节点也为二元树的根节点且具有为0的二元树深度。二元树深度达到MaxBTDepth(例如,4)暗示不存在进一步拆分。二元树节点具有等于MinBTSize(例如,4)的宽度暗示不存在进一步水平拆分。相似地,二元树节点具有等于MinBTSize的高度暗示无进一步垂直拆分。二元树的叶

节点(CU)被进一步处理(例如,通过执行预测过程及变换过程)而无任何进一步分割。

[0067] 图2A绘示使用QTBT分割技术而分割的块50(例如,CTB)的实例。如图2A所展示,在使用QTBT分割技术的情况下,通过每一块的中心对称地拆分所得块中的每一者。图2B绘示对应于图2A的块分割的树型结构。图2B中的实线指示四元树拆分且点线指示二元树拆分。在一个实例中,在二元树的每一拆分(即,非叶)节点中,发信语法元素(例如,旗标)以指示所执行的拆分的类型(例如,水平或垂直),其中0指示水平拆分且1指示垂直拆分。对于四元树拆分,无需指示拆分类型,这是由于四元树拆分始终将块水平地及垂直地拆分成具有相等大小的4个子块。

[0068] 如图2B所展示,在节点70处,使用四元树分割将块50拆分成图2A所展示的四个块51、52、53及54。块54未被进一步拆分,且因此为叶节点。在节点72处,使用二元树分割将块51进一步拆分成两个块。如图2B所展示,节点72是以1予以标记,这指示垂直拆分。因而,节点72处的拆分产生块57以及包含块55及56两者的块。在节点74处通过进一步垂直拆分来产生块55及56。在节点76处,使用二元树分割将块52进一步拆分成两个块58及59。如图2B所展示,节点76是以1予以标记,这指示水平拆分。

[0069] 在节点78处,使用四元树分割将块53拆分成4个相等大小块。块63及66是根据此四元树分割而产生且未被进一步拆分。在节点80处,首先使用垂直二元树拆分来拆分左上部块,从而产生块60及右垂直块。接着使用水平二元树拆分将右垂直块拆分成块61及62。在节点84处,使用水平二元树拆分将在节点78处根据四元树拆分而产生的右下块拆分成块64及65。

[0070] 在QTBT分割的一个实例中,可针对I切片彼此独立地执行亮度及色度分割,这与例如HEVC相反,在HEVC中,针对亮度及色度块联合地执行四元树分割。即,在正被研究的一些实例中,亮度块及色度块可被单独地分割,使得亮度块及色度块不会直接重叠。因而,在QTBT分割的一些实例中,可以使得至少一个经分割色度块不与单一经分割亮度块空间上对准的方式来分割色度块。即,与特定色度块共置的亮度样本可在两个或多于两个不同亮度分割区内。

[0071] 如上文所描述,在一些实例中,可从与对应亮度块相关的信息派生与色度块将被如何译码相关的信息。然而,如果亮度及色度分割被独立地执行,那么亮度及色度块可不对准(例如,亮度及色度块可不对应于同一像素集合)。举例来说,色度分割可使得色度块大于或小于对应亮度分割区。另外,色度块可空间上重叠于两个或多于两个亮度块。如上文所阐释,如果经分割色度块大于经分割亮度块,那么状况可为:存在空间上对应于特定色度块的多于一个亮度块,且因此存在与对应于色度分割区的大小的亮度分割区相关联的多于一个亮度信息(例如,语法元素等等)集合。在此类状况下,不清楚的是如何从亮度信息派生色度信息。应理解,亮度及色度块被独立地分割的任何分割结构可出现此类情形,而非仅仅是正由JVET研究的实例QTBT分割结构才出现此类情形。

[0072] 鉴于这些缺点,本发明描述用于从用于使用单独及/或独立亮度及色度分割而分割的图片的亮度信息派生色度信息的方法及装置。如上文所描述,亮度及色度分割可未对准,例如,具有不同大小或形状。来自亮度块的经派生信息(例如,经确定译码模式信息)可用作色度信息(例如,待用于色度块的译码模式)的预测值或用以译码视频数据的色度块。替代地或另外,亮度信息可用于色度信息的上下文译码中的上下文建模。任选地,上下文建

模可与预测信息组合。应理解,下文所描述的技术中的每一者可被独立地使用或可以任何组合而与其它技术组合。

[0073] 作为本发明的一个实例,视频编码器22使用特定译码模式(例如,特定帧内预测模式、特定帧间预测模式、特定滤波模式、特定运动向量预测模式等等)来编码视频数据的亮度块。在一些实例中,视频编码器22可进一步编码指示什么译码模式用以编码特定亮度块的语法元素。视频解码器30可经配置以解码语法元素以确定用以解码亮度块的译码模式。在其它实例中,视频解码器30可不接收明确地指示特定译码模式的语法元素。更确切地说,视频解码器30可经配置以基于各种视频特性(例如,块大小、来自相邻块的信息等等)及预定规则集合来派生用于亮度块的特定译码模式。在其它实例中,视频解码器30可基于经明确发信的语法元素与预定规则的组合来确定译码模式。

[0074] 在一个实例中,视频编码器22可任选地使用与对应亮度块相同的译码模式来编码色度块(例如,Cr块及/或Cb块)。视频编码器22可发信向视频解码器30指示将用于确定用于亮度块的译码模式的任何经发信或经派生信息再用作一或多个对应色度块的译码模式的预测值的语法元素(例如,旗标),而非视频编码器22仅仅发信指示用于色度块的译码模式的语法元素及/或其它信息。举例来说,可针对一或多个色度块译码旗标以指示是否以与对应亮度块相同的译码模式译码色度块。如果否,那么视频编码器22独立地产生指示用于色度块的译码模式的语法元素,其中视频编码器22可考虑色度模式不等于亮度模式。即,视频编码器22及视频解码器可能够确定用于色度块的译码模式不与用于亮度块的译码模式相同,且因此,用于亮度块的译码模式可作为用于色度块的可能性而被排除。在另外实例中,单独上下文可用以译码指示是否使用与亮度分量相同的模式来译码色度分量的旗标。

[0075] 在上文所描述的实例中,假定视频编码器22及视频解码器30首先译码亮度块,接着译码一或多个色度块。在此实例中,在色度块正被译码时,亮度信息已经可用。如果视频编码器22及视频解码器30经配置以按另一次序译码块(例如,首先译码色度块),那么可仅仅在以下实例中调换亮度及色度术语。

[0076] 在本发明的一个实例中,视频解码器30可经配置以接收经编码视频数据的位流且在存储器(例如,图1的存储媒体28)中存储经编码视频数据。经编码视频数据可表示经分割亮度块及经分割色度块两者。在一些实例中,经分割色度块可包含Cr色度块及Cb色度块两者。如本发明中所使用,术语“色度块”可指包含任何类型的色度信息的任何类型的块。在本发明的实例中,独立于亮度块来分割色度块。即,视频编码器22可经配置以使用用于亮度块及色度块的单独分割结构来编码视频数据。

[0077] 此类单独分割结构可引起至少一个经分割色度块不与单一经分割亮度块对准。因而,对于图片的特定空间位置,视频编码器22可分割单一色度块,但分割多个亮度块。然而,应理解,对于图片的其它空间位置,亮度块与色度块之间可存在1对1对应性,或对于单一亮度块可存在多个色度块。上文所描述的QTBT分割结构为亮度及色度块被独立地/单独地分割的一种类型的分割结构。然而,本发明的技术可应用于根据亮度及色度块被独立地分割的任何分割结构而分割的视频数据。

[0078] 视频解码器30可经进一步配置以为经编码视频位流中接收的相应经分割亮度块确定译码模式,且根据经确定相应译码模式来解码相应经分割亮度块。视频解码器30可经配置以从由经编码视频位流中接收的语法元素指示的信息确定译码模式。此类语法元素可

明确地指示译码模式。在其它实例中，视频解码器30可经配置以从视频数据的特性及一些预定规则隐含地为亮度块确定译码模式。在其它实例中，视频解码器30可使用经明确发信的语法元素与来自预定规则及视频数据特性的经隐含确定的译码模式的组合而为亮度块确定译码模式。

[0079] 在本发明的上下文中，译码模式可为向视频解码器30指示视频编码器22如何编码经编码视频数据及视频解码器30应如何解码视频数据的任何信息。实例译码模式可包含用于色度帧内预测的直接模式、位置相依帧内预测组合(PDPC)旗标(例如，指示是否使用PDPC模式)、PDPC参数、用于非可分离次级变换(NSST)的次级变换集、增强型多重变换(EMT)、自适应多重变换(ATM)，及用于选择熵译码数据模型的上下文。以上各者为色度译码模式的实例，所述色度译码模式可从由视频解码器30确定的亮度译码模式(用以译码亮度块的译码模式)派生，且已用于JVET中研究的JEM测试模型中。然而，应理解，译码模式可包含用于译码亮度块的任何译码模式，所述译码模式可再用于译码色度块或用以预测用于色度块的译码模式。

[0080] 不管译码模式的类型，或译码模式被确定的方式，视频解码器30皆可经配置以将用于特定经分割亮度块的经确定译码模式存储在与特定经分割亮度块相关联的多个不同存储器位置中。如下文将参考图3更详细地所阐释，特定经分割亮度块可被划分成子块，且视频解码器30可将为整个特定经分割亮度块而确定的译码模式存储在对应于子块中的每一者的存储器位置中。因此，对于划分成N个子块的特定经分割亮度块，译码模式存储在N个不同存储器位置中，每一存储器位置对应于经分割亮度块内的特定空间上定位的子块。子块可为任何大小的矩形或方形块。在一些实例中，子块可为仅仅一个样本，即，大小为1×1的块。在一些实例中，每一存储器位置可存储明确地指示用于特定经分割亮度块的译码模式的数据。在其它实例中，与特定经分割亮度块相关联的一或多个存储器位置明确地存储指示译码模式的信息，而与特定经分割亮度块相关联的其它存储器位置存储指向明确地存储译码模式的存储器位置的指针。

[0081] 根据本发明的一些实例，视频解码器30可经配置以再用经接收用于亮度块的译码模式信息以供在解码色度块时使用。在一些实例中，视频解码器30可接收及解码指示与相应经分割亮度块相关联的译码模式是否待用于解码特定经分割色度块的第一语法元素。如上文所描述，经分割色度块可与两个或多于两个不同经分割亮度块空间上对准。因而，可难以确定如何将亮度译码模式再用于此类色度块，这是由于不清楚从哪一亮度块继承(例如，再用)译码模式信息。通过在对应于多个子块位置的多个存储器位置中存储亮度译码模式信息，视频解码器30可经配置以使用经存储用于对应于特定色度块的空间位置的子块的译码模式信息的函数来确定来自亮度块的哪些译码模式信息再用于特定色度块。在此上下文中，函数可为由视频解码器30使用以确定两个或多于两个共置型分割亮度块中的哪一者(些)再用于经分割色度块的预定规则及分析技术集合。一旦确定将再用的亮度译码模式，视频解码器30就可运用经确定译码模式来解码特定经分割色度块。

[0082] 视频解码器30可经配置以基于存储在与经分割亮度块的子块相关联的存储器位置中的译码模式信息的函数而从两个或多于两个空间上对准亮度块确定将再用哪一译码模式。可使用若干不同函数。在本发明的一个实例中，视频解码器30可经配置以执行对应于与色度分割区共置的两个或多于两个亮度分割区的亮度译码模式信息的统计分析。供色度

块使用的来自亮度块的经确定亮度译码模式信息可为整个亮度译码模式信息(例如,与色度块共置的每一亮度子块中所含有的译码模式信息)及亮度译码模式信息跨越亮度块中的共置型子块如何变化(例如,信息相似或不同到什么程度)的函数。用于译码模式再用的先前技术与本发明的技术的一个差异为亮度译码模式信息可由于单独亮度及色度分割而与多于一个亮度块(或分割区)相关。

[0083] 可用于确定来自两个或多于两个经分割亮度块的什么亮度译码模式信息可再用于共置型色度块的函数的实例可包含但不限于下文所描述的函数中的一或多者。在一个实例中,视频解码器30可执行包含指示两个或多于两个经分割亮度块的经确定相应译码模式的译码模式信息的统计分析的函数。译码模式信息存储在与两个或多于两个经分割亮度块的相应子块相关联的相应存储器位置中。在一个实例中,视频解码器30可分析译码模式信息且返回(例如,用以确定用途及从存储器获得)对于两个或多于两个经分割亮度块的共置型子块最常出现的译码模式信息。即,函数返回用于对应亮度块中的大部分亮度信息。以此方式,函数指示来自两个或多于两个经分割亮度块的特定亮度译码模式信息待再用于共置型色度块。

[0084] 在其它实例中,视频解码器30可使用执行用于两个或多于两个经分割亮度块的子块的译码模式信息的分析的函数,分析包含测量亮度译码模式信息的梯度或高阶导数以测量所述信息的平滑度。举例来说,两个或多于两个亮度块中的一些极端(例如,离群值)译码模式信息(其可非常不同于与两个或多于两个亮度块相关的大部分译码模式信息)可被忽略且不再用于色度块。在其它实例中,视频解码器30可将不同权重指派给经存储用于相应子块的亮度译码模式信息,且使用译码模式的加权平均值以确定哪一模式再用于色度块。可基于与色度块共置的亮度子块的相对位置来指派权重。

[0085] 在其它实例中,用以确定什么亮度译码模式再用于色度块的函数可包含以下视频译码特性中的一或多者。可由视频解码器30使用以确定什么亮度译码模式再用于色度块的函数可包含块的形状(矩形、方形)、块定向(垂直或水平定向的矩形块)、色度块的形状、含有代表性位置的亮度块的形状、亮度及/或色度块的宽度或高度、对应于色度块的区域的更频繁使用的亮度模式。在其它实例中,函数可基于预测模式或代表性位置中的亮度信息。举例来说,如果视频解码器30经配置以再用亮度帧内模式以用于色度块,但代表性位置中的亮度块是以帧间模式予以译码,那么视频解码器30可经配置以选择另一代表性位置以确定再用于色度块的亮度帧内模式。更一般化地,如果视频解码器30确定什么亮度信息再用于译码色度块,但亮度信息可对于此色度块并非有效,那么视频解码器30可考虑来自亮度块中的另一代表性位置的亮度信息,或以其它方式可使用某一默认亮度信息。

[0086] 在本发明的另一实例中,视频解码器30可基于预定子块位置来再用译码模式信息。举例来说,视频解码器30可仅仅再用经存储用于与色度块共置的特定子块位置的译码模式信息。作为一个实例,视频解码器30可再用存储在与分割色度块的特定拐角共置的亮度子块处的译码模式信息。可使用任何拐角子块。作为另一实例,视频解码器30可再用存储在与分割色度块的中心共置的亮度子块处的译码模式信息。在其它实例中,视频解码器30可执行经存储用于某一预定数目个亮度子块位置的译码模式信息的统计分析(例如,如上文所描述)。即,视频解码器30可使用分析亮度分割区的某些子块且基于其中所含有的亮度信息来派生色度信息的函数。

[0087] 在另一实例中，视频解码器30可将经分割色度块划分成多个子块，例如，1×1、2×2、4×4或其它大小的子块。接着，对于每一子块，视频解码器30可继承(例如，再用)经存储用于特定亮度子块的亮度译码模式信息以用于共置型色度子块。以此方式，可在单一色度块中应用来自对应亮度块的不同译码信息。

[0088] 如上文所论述，视频解码器30可使用多个预定函数中的一者以用于确定如何将亮度译码模式信息再用于色度块。在一些实例中，视频解码器30可经配置以使用单一预定函数，且针对所有图片使用所述函数。在其它实例中，视频解码器30可基于一些视频译码特性来确定将使用哪一函数。在其它实例中，视频编码器22可经配置以发信向视频解码器30指示将使用什么函数来确定如何再用亮度译码模式信息的语法元素。可在任何级(例如，序列级、图片级、切片级、图像块级、CTB级等等)处发信此类语法元素。

[0089] 图3为绘示根据本发明的技术的亮度及色度相对分割的实例的概念图。如图3所展示，按照分割区(例如，经分割亮度块)将信息存储在分割区的子块(虚线方框)中。子块可具有任何大小，下至个别样本的大小。图3展示一个色度分割区(无论呈4:4:4还是4:2:0子取样格式)可具有多于一个关联亮度分割区。因而，单一色度分割区可具有多于一个对应亮度信息集合。亮度分割区的某些代表性位置(例如，子块)可用以分析亮度信息(例如，译码模式)，以便派生用于对应色度分割区的色度信息，如上文所描述。

[0090] 图4为绘示根据本发明的技术的亮度及色度相对分割的另一实例的概念图。如图4所展示，按照分割区将信息存储在亮度分割区的子块(虚线方框)中。图4展示一个色度分割区中的每一子块可具有一个关联亮度子块，且可分析关联的一个亮度子块的亮度信息以派生用于对应色度子块的色度信息。

[0091] 以下章节描述可使用本发明的技术的一些实例。在色度直接模式中，亮度帧内方向用于色度帧内预测。此模式的实例用于HEVC中。根据本发明的一个实例技术，在色度及亮度结构不对准(例如，由于独立色度及亮度分割)时，选择中心代表性亮度子块以获得亮度帧内预测模式，亮度帧内预测模式接着作为直接模式而应用于色度分割区。对应亮度分割区的其它亮度子块可具有不同于选定子块的其它帧内方向。还可使用其它函数以代替使用如上文所阐释的中心代表性子块。

[0092] 在另一实例中，在色度直接模式中，在色度及亮度结构不对准时，在2×2(或4×4)子块单元中执行色度帧内预测。对于每一2×2(或4×4)色度子块，识别一个关联4×4亮度子块，且此经识别4×4亮度子块的帧内预测模式应用于当前色度2×2(或4×4)子块。

[0093] 在另一实例中，色度PDPC控制旗标(即，PDPC模式被应用或未被应用)及PDPC参数是例如从中心代表性亮度子块派生，且应用于色度分割区。还可使用其它函数以代替使用如上文所阐释的中心代表性子块。

[0094] 在另一实例中，次级变换(NSST)集是由视频解码器30从中心代表性亮度子块选择且应用于色度分割区。还可使用其它函数以代替使用如上文所阐释的中心代表性子块。

[0095] 相似技术可由视频解码器30应用于从亮度信息派生的任何色度信息。

[0096] 参考视频解码器30而描述前述实例。然而，视频编码器22可使用相同技术以用于确定如何将针对亮度块所产生、派生及/或发信的信息再用于色度块。具体地说，视频编码器22可基于将由视频解码器使用以确定将再用什么亮度译码模式信息的函数来确定是否发信指示亮度译码模式信息是否应再用于色度块的语法元素。

[0097] 以下章节描述用于为可被分割成非方形矩形分割区的视频数据块确定用于位置相依帧内预测组合(PDPC)译码模式的参数的技术。上文所描述的QTBT分割结构为允许非方形矩形块的分割结构的实例。然而，本发明的技术可与产生非方形矩形块的任何分割结构一起使用。

[0098] 在使用PDPC译码模式来译码视频数据时，视频编码器22及/或视频解码器30可使用定义如何基于经滤波及未经滤波参考值及基于经预测像素的位置来组合预测的一或多个参数化方程式。本发明描述若干参数集合，使得视频编码器22可经配置以测试所述参数集合(例如，经由使用速率-失真分析)且将最佳参数(例如，被测试的那些参数当中引起最佳速率-失真效能的参数)发信到视频解码器30。在其它实例中，视频解码器30可经配置以从视频数据的特性(例如，块大小、块高度、块宽度等等)确定PDPC参数。

[0099] 图5A绘示根据本发明的技术使用未经滤波参考(r)的4×4块(p)的预测。图5B绘示根据本发明的技术使用经滤波参考(s)的4×4块(q)的预测。尽管图5A及5B两者绘示4×4像素块及17(4×4+1)个相应参考值，但本发明的技术可应用于任何块大小及任何数目个参考值。

[0100] 在执行PDPC译码模式时，视频编码器22及/或视频解码器30可利用经滤波(q)预测与未经滤波(p)预测之间的组合，使得可使用来自经滤波(s)及未经滤波(r)参考阵列的像素值来计算用于待译码的当前块的经预测块。

[0101] 在PDPC的技术的一个实例中，在给出仅分别使用未经滤波及经滤波参考r及s而计算的任何两个像素预测$p_r[x,y]$及$q_s[x,y]$集合的情况下，由v[x,y]表示的像素的组合式经预测值是由下式定义：

[0102] $v[x,y]=c[x,y]p_r[x,y]+(1-c[x,y])q_s[x,y]$ (1)

[0103] 其中c[x,y]为组合参数集合。权重c[x,y]的值可为介于0与1之间的值。权重c[x,y]及(1-c[x,y])的总和可等于1。

[0104] 在某些实例中，可不实际的是具有与块中的像素的数目一样大的参数集合。在此类实例中，c[x,y]可由小得多的参数集合加方程式定义以计算来自那些参数的所有组合值。在此类实例中，可使用以下公式：

[0105]

$$v[x,y]=\left\lfloor\frac{c_1^{(v)}r[x,-1]-c_2^{(v)}r[-1,-1]}{2^{\left\lfloor\frac{y}{d_v}\right\rfloor}}\right\rfloor+\left\lfloor\frac{c_1^{(h)}r[-1,y]-c_2^{(h)}r[-1,-1]}{2^{\left\lfloor\frac{x}{d_h}\right\rfloor}}\right\rfloor+\left(\frac{N-\min(x,y)}{N}\right)g\ p_{\mathbf{r}}^{(\mathrm{HEVC})}[x,y]+b[x,y]\,q_{\mathbf{s}}^{(\mathrm{HEVC})}[x,y] \quad (2)$$

[0106] 其中c_1^v,c_2^v,c_1^h,c_2^h,g及$d_v,d_h\in\{1,2\}$为预测参数，N为块大小，$p_r[x,y]$及$q_s[x,y]$为使用根据HEVC标准所计算的预测值，对于特定模式，分别使用未经滤波及经滤波参考，且

[0107] $$b[x,y]=1-\left\lfloor\frac{c_1^{(v)}-c_2^{(v)}}{2^{\lfloor y/d_v\rfloor}}\right\rfloor-\left\lfloor\frac{c_1^{(h)}-c_2^{(h)}}{2^{\lfloor x/d_h\rfloor}}\right\rfloor-\left(\frac{N-\min(x,y)}{N}\right)g \quad (3)$$

[0108] 为由预测参数定义的标准化因数(即，用以使经指派到$p_{\mathbf{r}}^{(\mathrm{HEVC})}[x,y]$及$q_{\mathbf{s}}^{(\mathrm{HEVC})}[x,y]$的总权重增加到1)。

[0109] 公式2可经一般化用于公式2A中的任何视频译码标准：

[0110] $$v[x,y] = \left\lfloor \frac{c_1^{(v)} r[x,-1] - c_2^{(v)} r[-1,-1]}{2^{\left\lfloor \frac{y}{d_v} \right\rfloor}} \right\rfloor + \left\lfloor \frac{c_1^{(h)} r[-1,y] - c_2^{(h)} r[-1,-1]}{2^{\left\lfloor \frac{x}{d_h} \right\rfloor}} \right\rfloor + \left(\frac{N-\min(x,y)}{N}\right) g\ p_{\mathbf{r}}^{(\mathrm{STD})}[x,y] + b[x,y]\ q_{\mathbf{s}}^{(\mathrm{STD})}[x,y] \quad (2A)$$

[0111] 其中$c_1^v, c_2^v, c_1^h, c_2^h$,g及$d_v$,$d_h \in \{1,2\}$为预测参数,N为块大小,$p_{\mathbf{r}}^{(\mathrm{STD})}[x,y]$及$q_{\mathbf{s}}^{(\mathrm{STD})}[x,y]$为使用根据视频译码标准(或视频译码方案或算法)所计算的预测值,对于特定模式,分别使用未经滤波及经滤波参考,且

[0112] $$b[x,y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor - \left(\frac{N-\min(x,y)}{N}\right) g \quad (3A)$$

[0113] 为由预测参数定义的标准化因数(即,用以使经指派到$p_{\mathbf{r}}^{(\mathrm{STD})}[x,y]$及$q_{\mathbf{s}}^{(STD)}[x,y]$的总权重增加到1)。

[0114] 这些预测参数可包含用以根据所使用的预测模式的类型(例如,HEVC的DC、平面及33方向性模式)来提供经预测项的最佳线性组合的权重。举例来说,HEVC含有35个预测模式。查找表可以用于预测模式中的每一者的预测参数$c_1^v, c_2^v, c_1^h, c_2^h$,g,$d_v$及$d_h$中的每一者的值(即,用于每一预测模式的$c_1^v, c_2^v, c_1^h, c_2^h$,g,$d_v$及$d_h$的35个值)予以构造。此类值可在具有视频的位流中被编码,或可为事先由编码器及解码器所知的常量值且可无需在文件或位流中被发射。$c_1^v, c_2^v, c_1^h, c_2^h$,g,$d_v$及$d_h$的值可由优化训练算法通过找到用于针对训练视频集合给出最佳压缩的预测参数的值而确定。

[0115] 在另一实例中,对于每一预测模式存在多个预定义预测参数集(例如,在查找表中),且选定预测参数集(而非参数自身)在经编码文件或位流中发射到解码器。在另一实例中,$c_1^v, c_2^v, c_1^h, c_2^h$,g,$d_v$及$d_h$的值可由视频编码器在运作中产生且在经编码文件或位流中发射到解码器。

[0116] 在另一实例中,代替使用HEVC预测,执行这些技术的视频译码装置可使用HEVC的经修改版本,类似于使用65方向性预测以代替33方向性预测的版本。事实上,可使用任何类型的帧内预测。

[0117] 在另一实例中,可选择公式以促进计算。举例来说,我们可使用以下类型的预测值

[0118]

$$v[x,y] = \left\lfloor \frac{c_1^{(v)} r[x,-1] - c_2^{(v)} r[-1,-1]}{2^{\lfloor y/d_v \rfloor}} \right\rfloor + \left\lfloor \frac{c_1^{(h)} r[-1,y] - c_2^{(h)} r[-1,-1]}{2^{\lfloor x/d_h \rfloor}} \right\rfloor + b[x,y]\ p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{HEVC})}[x,y] \quad (4)$$

[0119] 其中

[0120] $$b[x,y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor \quad (5)$$

[0121] 且

[0122] $$p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{HEVC})}[x,y] = a\ p_{\mathbf{r}}^{(\mathrm{HEVC})}[x,y] + (1-a)\ q_{\mathbf{s}}^{(\mathrm{HEVC})}[x,y]。 \quad (6)$$

[0123] 此类途径可采用HEVC(或其它)预测的线性。在将h定义为来自预定义集合的滤波器k的脉冲响应的情况下,如果我们具有

[0124] s=ar+(1-a)(h*r) (7)

[0125] 其中“*”表示卷积，那么

[0126] $$p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{HEVC})}[x,y] = p_{\mathbf{s}}^{(\mathrm{HEVC})}[x,y] \qquad (8)$$

[0127] 即，可从线性组合参考计算线性组合预测。

[0128] 公式4、6及8可针对公式4A、6A及8A中的任何视频译码标准而被一般化：

[0129]

$$v[x,y] = \left\lfloor \frac{c_1^{(v)} r[x,-1] - c_2^{(v)} r[-1,-1]}{2^{\lfloor y/d_v \rfloor}} \right\rfloor + \left\lfloor \frac{c_1^{(h)} r[-1,y] - c_2^{(h)} r[-1,-1]}{2^{\lfloor x/d_h \rfloor}} \right\rfloor + b[x,y]\, p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{STD})}[x,y] \qquad (4A)$$

[0130] 其中

[0131] $$b[x,y] = 1 - \left\lfloor \frac{c_1^{(v)} - c_2^{(v)}}{2^{\lfloor y/d_v \rfloor}} \right\rfloor - \left\lfloor \frac{c_1^{(h)} - c_2^{(h)}}{2^{\lfloor x/d_h \rfloor}} \right\rfloor \qquad (5A)$$

[0132] 且

[0133] $$p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{STD})}[x,y] = a\, p_{\mathbf{r}}^{(\mathrm{STD})}[x,y] + (1-a)\, q_{\mathbf{s}}^{(\mathrm{STD})}[x,y]。 \qquad (6A)$$

[0134] 此类途径可采用译码标准的预测的线性。在将h定义为来自预定义集合的滤波器k的脉冲响应的情况下，如果我们具有

[0135] s＝ar+(1-a)(h*r) (7A)

[0136] 其中“*”表示卷积，那么

[0137] $$p_{a,\mathbf{r},\mathbf{s}}^{(\mathrm{STD})}[x,y] = p_{\mathbf{s}}^{(\mathrm{STD})}[x,y] \qquad (8A)$$

[0138] 即，可从线性组合参考计算线性组合预测。

[0139] 在实例中，预测函数可仅使用参考向量(例如，r及s)作为输入。在此实例中，如果参考已被滤波或尚未被滤波，那么参考向量的行为不会改变。如果r与s相等(例如，某一未经滤波参考r恰好与另一经滤波参考s相同)，那么应用于经滤波及未经滤波参考的预测性函数(例如，$p_r[x,y]$(也被写为p(x,y,r))等于$p_s[x,y]$(也被写为p(x,y,s)))相等。另外，像素预测p及q可等效(例如，在给出相同输入的情况下产生相同输出)。在此类实例中，可运用替换像素预测q[x,y]的像素预测p[x,y]来重写公式(1)到(8)。

[0140] 在另一实例中，预测(例如，函数集合)可取决于参考已被滤波的信息而改变。在此实例中，可表示不同函数集合(例如，$p_r[x,y]$及$q_s[x,y]$)。在此状况下，即使r与s相等，$p_r[x,y]$及$q_s[x,y]$仍可不相等。换句话说，取决于输入是否已被滤波，相同输入可产生不同输出。在此类实例中，p[x,y]可不能够由q[x,y]替换。

[0141] 所展示的预测方程式的优点为：在运用参数化公式化的情况下，可使用诸如训练的技术而针对不同类型的视频纹理确定最佳参数集合(即，优化预测准确度的最佳参数集合)。又可在一些实例中通过针对一些典型类型的纹理计算若干预测值参数集合且具有编码器测试来自每一集合的预测值且将得到最佳压缩的预测值编码为边信息的压缩方案来扩展此途径。

[0142] 在上文所描述的技术的一些实例中，在启用PDPC译码模式时，用于PDPC模式的帧内预测加权及控制(例如，使用经滤波或未经滤波样本)的PDPC参数被预计算且存储在查找表(LUT)中。在一个实例中，视频解码器30根据块大小及帧内预测方向来确定PDPC参数。用于PDPC译码模式的先前技术假定经帧内预测块的大小始终为方形。

[0143] JVET测试模型包含PDPC译码模式。如上文所论述，JVET测试模型使用QTBT分割，其

允许非方形矩形块。以下章节论述用于矩形块的PDPC译码的扩展的实例技术。然而，应理解，本发明的技术可用于确定用于使用非方形块的任何预测模式的预测模式参数，所述预测模式包含根据样本位置将加权平均值应用于预测值及参考样本的预测模式。

[0144] 建议以对于水平相关参数使用块的宽度以存储或存取来自LUT的PDPC参数且对于垂直相关参数使用块的高度以存储或存取来自LUT的PDPC参数的方式修改用于确定PDPC参数的LUT的结构，或用于派生来自LUT的参数的技术。对于不具有水平或垂直关系的其它参数，可应用块宽度及高度的函数以存储或存取来自LUT的那些参数。

[0145] 视频解码器30可接收已使用PDPC模式而编码的视频数据块。在此实例中，视频数据块可具有由宽度及高度界定的非方形矩形形状。视频解码器30可依据帧内预测模式及块的宽度来确定水平相关PDPC参数。视频解码器30可依据帧内预测模式及块的高度来确定垂直相关PDPC参数。另外，视频解码器30可基于帧内预测模式以及块的高度及宽度的函数来确定非方向性PDPC参数(例如，既不水平相关也不垂直相关的PDPC参数)。上文以上标v指示实例垂直相关参数。上文以上标h指示实例水平相关参数。举例来说，函数可为但不限于块的宽度及高度的最大值或最小值，或块的高度及宽度的加权平均值，其中加权可取决于一个尺寸如何大于块的另一尺寸。举例来说，相比于加权平均值中的另一尺寸，较大尺寸(宽度或高度)可具有较大权重。

[0146] 由于存在被允许的仅某一数目个块形状(宽度及高度)，故还可明确地针对所有可能的块宽度及高度或子组合来表示此函数。举例来说，如果我们具有N_w及N_h可能块宽度及高度，那么大小为$N_w \times N_h$的表可针对每一矩形或方形块存储待在帧内预测过程中使用的数据。

[0147] 图6为绘示根据本发明的一个实例的用于确定矩形块中使用的预测参数集合的嵌套表的使用的概念图。视频解码器30可使用一或多个LUT来确定PDPC参数，所述一或多个LUT的条目是关于块的宽度及高度两者被加索引。如图6所展示，宽度(W)及/或高度(H)可用作对大小到参数表90的输入。大小到参数表90可经配置为含有指向预测参数表92中的条目的索引的(LUT)。如上文所论述，大小到参数表90可具有大小$N_w \times N_h$以考虑N_w及N_h可能块宽度及高度。在此实例中，大小到参数表90可用于单一帧内预测模式(例如，DC、平面或其它预测方向)。在其它实例中，大小到参数表90可含有用于所有帧内预测模式的条目，且使用块高度、块宽度及帧内预测模式作为到表中的条目。一般来说，为了最小化存储器，视频解码器30及视频编码器22可经配置以组合行及列中的表条目，从而缩减表(例如，大小到参数表90)的大小，且可能地产生具有不同大小的若干表。

[0148] 作为一个实例，在假定特定帧内预测模式的情况下，视频解码器30可使用正被解码的视频数据块的宽度来存取大小到参数表90中的条目以确定一或多个水平相关PDPC参数。基于块的宽度，大小到参数表90中的对应条目可用作对预测参数表92的输入。预测参数表92具有大小$N_p \times N_e$且含有实际PDPC参数的条目。因而，从大小到参数表90获得的条目为指向预测参数表92中的实际水平相关PDPC参数的索引，其接着用于解码块94中。

[0149] 同样地，视频解码器30可使用正被解码的视频数据块的高度来存取大小到参数表90中的条目以确定一或多个垂直相关PDPC参数。基于块的宽度，大小到参数表90中的对应条目可用作对预测参数表92的输入以获得预测参数表92中的实际垂直相关PDPC参数，其接着用于解码块94中。相同过程可应用于基于块的高度及宽度的函数而加索引的非方向性PDPC参数。

[0150] 图7为绘示可实施本发明的技术的实例视频编码器22的框图。图7是出于阐释的目的而被提供，且不应被视为限制如本发明中广泛地示范及描述的技术。本发明的技术可适用于各种译码标准或方法。

[0151] 在图7的实例中，视频编码器22包含预测处理单元100、视频数据存储器101、残余产生单元102、变换处理单元104、量化单元106、反量化单元108、反变换处理单元110、重构单元112、滤波器单元114、经解码图片缓冲器116及熵编码单元118。预测处理单元100包含帧间预测处理单元120及帧内预测处理单元126。帧间预测处理单元120可包含运动估计单元及运动补偿单元(未展示)。

[0152] 视频数据存储器101可经配置以存储待由视频编码器22的组件编码的视频数据。存储在视频数据存储器101中的视频数据可例如从视频源18获得。经解码图片缓冲器116可为参考图片存储器，其存储用于由视频编码器22在编码视频数据(例如，在帧内或帧间译码模式中)时使用的参考视频数据。视频数据存储器101及经解码图片缓冲器116可由多种存储器装置中的任一者形成，诸如动态随机存取存储器(DRAM)，包含同步DRAM(SDRAM)、磁阻式RAM(MRAM)、电阻式RAM(RRAM)或其它类型的存储器装置。可由同一存储器装置或单独存储器装置提供视频数据存储器101及经解码图片缓冲器116。在各种实例中，视频数据存储器101可与视频编码器22的其它组件一起在芯片上，或相对于那些组件在芯片外。视频数据存储器101可与图1的存储媒体20相同或为图1的存储媒体20的部分。

[0153] 视频编码器22接收视频数据。视频编码器22可编码视频数据的图片的切片中的每一CTU。CTU中的每一者可与相等大小的亮度译码树型块(CTB)及图片的对应CTB相关联。作为编码CTU的部分，预测处理单元100可执行分割以将CTU的CTB划分成逐渐较小的块。在一些实例中，视频编码器22可使用QTBT结构来分割块。较小块可为CU的译码块。举例来说，预测处理单元100可根据树型结构来分割与CTU相关联的CTB。根据本发明的一或多种技术，对于树型结构的每一深度级处的树型结构的每一相应非叶节点，对于相应非叶节点存在多个允许拆分图案，且对应于相应非叶节点的视频块根据多个允许拆分图案中的一者而分割成对应于相应非叶节点的子节点的视频块。

[0154] 视频编码器22可编码CTU的CU以产生CU的经编码表示(即，经译码CU)。作为编码CU的部分，预测处理单元100可分割与CU的一或多个PU中的CU相关联的译码块。因此，每一PU可与亮度预测块及对应色度预测块相关联。视频编码器22及视频解码器30可支持具有各种大小的PU。如上文所指示，CU的大小可指CU的亮度译码块的大小，且PU的大小可指PU的亮度预测块的大小。假定特定CU的大小为2N×2N，那么视频编码器22及视频解码器30可支持用于帧内预测的2N×2N或N×N的PU大小，及用于帧间预测的2N×2N、2N×N、N×2N、N×N或相似者的对称PU大小。视频编码器22及视频解码器30还可支持用于帧间预测的2N×nU、2N×nD、nL×2N及nR×2N的PU大小的不对称分割。

[0155] 帧间预测处理单元120可通过对CU的每一PU执行帧间预测来产生用于PU的预测性数据。用于PU的预测性数据可包含PU的预测性块及用于PU的运动信息。取决于PU在I切片中、在P切片中还是在B切片中，帧间预测处理单元120可针对CU的PU执行不同操作。在I切片中，所有PU被帧内预测。因此，如果PU在I切片中，那么帧间预测处理单元120不对PU执行帧间预测。因此，对于在I模式中编码的块，使用空间预测而从同一帧内的经先前编码的相邻块形成经预测块。如果PU在P切片中，那么帧间预测处理单元120可使用单向帧间预测以产

生PU的预测性块。如果PU在B切片中，那么帧间预测处理单元120可使用单向或双向帧间预测以产生PU的预测性块。

[0156] 帧内预测处理单元126可通过对PU执行帧内预测来产生用于PU的预测性数据。用于PU的预测性数据可包含PU的预测性块及各种语法元素。帧内预测处理单元126可对I切片、P切片及B切片中的PU执行帧内预测。

[0157] 为了对PU执行帧内预测，帧内预测处理单元126可使用多个帧内预测模式以产生用于PU的多个预测性数据集合。帧内预测处理单元126可使用来自相邻PU的样本块的样本以产生用于PU的预测性块。对于PU、CU及CTU，假定从左到右、从上而下的编码次序，那么相邻PU可在PU上方、右上方、左上方或左边。帧内预测处理单元126可使用各种数目个帧内预测模式，例如，33个方向性帧内预测模式。在一些实例中，帧内预测模式的数目可取决于与PU相关联的区的大小。另外，如下文将参考图11更详细地所描述，帧内预测处理单元126可经配置以依据视频数据块的高度及/或宽度来确定用于编码视频数据块的PDPC参数。

[0158] 预测处理单元100可从由帧间预测处理单元120产生的用于PU的预测性数据当中或从由帧内预测处理单元126产生的用于PU的预测性数据当中选择用于CU的PU的预测性数据。在一些实例中，预测处理单元100基于预测性数据集合的速率/失真度量来选择用于CU的PU的预测性数据。选定预测性数据的预测性块可在本文中被称作选定预测性块。

[0159] 残余产生单元102可基于用于CU的译码块(例如，亮度、Cb及Cr译码块)及用于CU的PU的选定预测性块(例如，预测性亮度、Cb及Cr块)来产生用于CU的残余块(例如，亮度、Cb及Cr残余块)。举例来说，残余产生单元102可产生CU的残余块，使得残余块中的每一样本具有等于CU的译码块中的样本与CU的PU的对应选定预测性块中的对应样本之间的差的值。

[0160] 变换处理单元104可执行四元树分割以将与CU相关联的残余块分割成与CU的TU相关联的变换块。因此，TU可与亮度变换块及两个色度变换块相关联。CU的TU的亮度及色度变换块的大小及位置可或可不基于CU的PU的预测块的大小及位置。被称为“残余四元树”(RQT)的四元树结构可包含与区中的每一者相关联的节点。CU的TU可对应于RQT的叶节点。

[0161] 变换处理单元104可通过将一或多个变换应用于TU的变换块来产生用于CU的每一TU的变换系数块。变换处理单元104可将各种变换应用于与TU相关联的变换块。举例来说，变换处理单元104可将离散余弦变换(DCT)、方向性变换或概念上相似变换应用于变换块。在一些实例中，变换处理单元104不将变换应用于变换块。在此类实例中，变换块可被视为变换系数块。

[0162] 量化单元106可量化系数块中的变换系数。量化过程可缩减与变换系数中的一些或全部相关联的位深度。举例来说，n位变换系数可在量化期间被降值舍入到m位变换系数，其中n大于m。量化单元106可基于与CU相关联的量化参数(QP)值来量化与CU的TU相关联的系数块。视频编码器22可通过调整与CU相关联的QP值来调整应用于与CU相关联的系数块的量化程度。量化可引入信息损失。因此，经量化变换系数相比于原始变换系数可具有较低精确度。

[0163] 反量化单元108及反变换处理单元110可分别将反量化及反变换应用于系数块，以从系数块重构残余块。重构单元112可将经重构残余块与来自由预测处理单元100产生的一或多个预测性块的对应样本相加以产生与TU相关联的经重构变换块。通过以此方式重构用于CU的每一TU的变换块，视频编码器22可重构CU的译码块。

[0164] 滤波器单元114可执行一或多个解块操作以缩减与CU相关联的译码块中的块假影。在滤波器单元114对经重构译码块执行一或多个解块操作之后，经解码图片缓冲器116可存储经重构译码块。帧间预测处理单元120可使用含有经重构译码块的参考图片以对其它图片的PU执行帧间预测。另外，帧内预测处理单元126可使用经解码图片缓冲器116中的经重构译码块以对与CU相同的图片中的其它PU执行帧内预测。

[0165] 熵编码单元118可从视频编码器22的其它功能组件接收数据。举例来说，熵编码单元118可从量化单元106接收系数块，且可从预测处理单元100接收语法元素。熵编码单元118可对数据执行一或多个熵编码操作以产生经熵编码数据。举例来说，熵编码单元118可对数据执行CABAC操作、上下文自适应可变长度译码(CAVLC)操作、可变到可变(V2V)长度译码操作、基于语法的上下文自适应二进制算术译码(SBAC)操作、概率区间分割熵(PIPE)译码操作、指数哥伦布(Exponential-Golomb)编码操作，或另一类型的熵编码操作。视频编码器22可输出包含由熵编码单元118产生的经熵编码数据的位流。举例来说，位流可包含表示用于CU的RQT的数据。

[0166] 图8为绘示经配置以实施本发明的技术的实例视频解码器30的框图。图8是出于阐释的目的而被提供，且并不限制如本发明中广泛地示范及描述的技术。出于阐释的目的，本发明在HEVC译码的上下文中描述视频解码器30。然而，本发明的技术可适用于其它译码标准或方法，包含允许非方形分割及/或独立亮度及色度分割的技术。

[0167] 在图8的实例中，视频解码器30包含熵解码单元150、视频数据存储器151、预测处理单元152、反量化单元154、反变换处理单元156、重构单元158、滤波器单元160，及经解码图片缓冲器162。预测处理单元152包含运动补偿单元164及帧内预测处理单元166。在其它实例中，视频解码器30可包含更多、更少或不同的功能组件。

[0168] 视频数据存储器151可存储待由视频解码器30的组件解码的经编码视频数据，诸如经编码视频位流。可例如从计算机可读媒体16(例如，从本地视频源(诸如相机))经由视频数据的有线或无线网络通信或通过存取物理数据存储媒体而获得存储在视频数据存储器151中的视频数据。视频数据存储器151可形成存储来自经编码视频位流的经编码视频数据的经译码图片缓冲器(CPB)。经解码图片缓冲器162可为存储用于由视频解码器30在解码视频数据(例如，以帧内或帧间译码模式)时使用的参考视频数据的参考图片存储器。视频数据存储器151及经解码图片缓冲器162可由多种存储器装置中的任一者形成，诸如动态随机存取存储器(DRAM)，包含同步DRAM(SDRAM)、磁阻式RAM(MRAM)、电阻式RAM(RRAM)或其它类型的存储器装置。可由同一存储器装置或单独存储器装置提供视频数据存储器151及经解码图片缓冲器162。在各种实例中，视频数据存储器151可与视频解码器30的其它组件一起在芯片上，或相对于那些组件在芯片外。视频数据存储器151可与图1的存储媒体28相同或为图1的存储媒体28的部分。

[0169] 视频数据存储器151接收及存储位流的经编码视频数据(例如，NAL单元)。熵解码单元150可从视频数据存储器151接收经编码视频数据(例如，NAL单元)，且可分析NAL单元以获得语法元素。熵解码单元150可熵解码NAL单元中的经熵编码语法元素。预测处理单元152、反量化单元154、反变换处理单元156、重构单元158及滤波器单元160可基于从位流提取的语法元素来产生经解码视频数据。熵解码单元150可执行与熵编码单元118的过程大体上互逆的过程。

[0170] 根据本发明的一些实例，熵解码单元150可确定树型结构作为从位流获得语法元素的部分。树型结构可指定如何将初始视频块(诸如CTB)分割成较小视频块(诸如译码单元)。根据本发明的一或多种技术，对于树型结构的每一深度级处的树型结构的每一相应非叶节点，对于相应非叶节点存在多个允许拆分图案，且对应于相应非叶节点的视频块根据多个允许拆分图案中的一者而分割成对应于相应非叶节点的子节点的视频块。

[0171] 另外，如下文将参考图10更详细地所阐释，视频解码器30可经配置以确定在存在对应于单一色度块的两个或多于两个亮度块的情形中如何再用经接收用于亮度块的译码模式信息以供在解码色度块时使用。

[0172] 除了从位流获得语法元素以外，视频解码器30还可对未经分割CU执行重构操作。为了对CU执行重构操作，视频解码器30可对CU的每一TU执行重构操作。通过针对CU的每一TU执行重构操作，视频解码器30可重构CU的残余块。

[0173] 作为对CU的TU执行重构操作的部分，反量化单元154可反量化(即，解量化)与TU相关联的系数块。在反量化单元154反量化系数块之后，反变换处理单元156可将一或多个反变换应用于系数块以便产生与TU相关联的残余块。举例来说，反变换处理单元156可将反DCT、反整数变换、反卡洛南-洛伊(Karhunen-Loeve)变换(KLT)、反旋转变换、反方向性变换或另一反变换应用于系数块。

[0174] 如果使用帧内预测来编码PU，那么帧内预测处理单元166可执行帧内预测以产生PU的预测性块。帧内预测处理单元166可使用帧内预测模式以基于样本空间上相邻块来产生PU的预测性块。帧内预测处理单元166可基于从位流获得的一或多个语法元素来确定用于PU的帧内预测模式。另外，如下文将参考图12更详细地所描述，帧内预测处理单元166可经配置以依据视频数据块的高度及/或宽度来确定用于编码视频数据块的PDPC参数。

[0175] 如果使用帧间预测来编码PU，那么熵解码单元150可确定用于PU的运动信息。运动补偿单元164可基于PU的运动信息来确定一或多个参考块。运动补偿单元164可基于一或多个参考块来产生用于PU的预测性块(例如，预测性亮度、Cb及Cr块)。

[0176] 重构单元158可使用用于CU的TU的变换块(例如，亮度、Cb及Cr变换块)及CU的PU的预测性块(例如，亮度、Cb及Cr块)(即，在适用时为帧内预测数据或帧间预测数据)以重构用于CU的译码块(例如，亮度、Cb及Cr译码块)。举例来说，重构单元158可将变换块(例如，亮度、Cb及Cr变换块)的样本与预测性块(例如，亮度、Cb及Cr预测性块)的对应样本相加以重构CU的译码块(例如，亮度、Cb及Cr译码块)。

[0177] 滤波器单元160可执行解块操作以缩减与CU的译码块相关联的块假影。视频解码器30可将CU的译码块存储在经解码图片缓冲器162中。经解码图片缓冲器162可提供参考图片以用于后续运动补偿、帧内预测及在显示装置(诸如图1的显示装置32)上的呈现。举例来说，视频解码器30可基于经解码图片缓冲器162中的块而针对其它CU的PU执行帧内预测或帧间预测操作。

[0178] 图9为绘示根据本发明的技术的视频译码器的实例操作的流程图。视频译码器可为视频编码器22及/或视频解码器30。根据本发明的技术，编码器22及/或视频解码器30可经配置以进行以下操作：将视频数据分割成亮度分量的分割区(200)；将视频数据分割成色度分量的分割区，其中色度分量是独立于亮度分量被分割(202)；译码亮度分量的第一分割区(204)；译码指示与译码亮度分量的第一分割区相关联的信息是否待用于译码色度分量

的第二分割区的语法元素(206);及根据语法元素来译码色度分量的第二分割区(208)。

[0179] 图10为绘示根据本发明的技术的视频解码器的实例操作的流程图。图10的技术可由视频解码器30的一或多个硬件结构执行。

[0180] 在本发明的一个实例中,视频解码器30可经配置以接收经编码视频数据的位流,经编码视频数据表示经分割亮度块及经分割色度块,其中色度块是独立于亮度块被分割(212)。视频解码器30可经进一步配置以确定对应于相应经分割亮度块的相应译码模式(214),及根据经确定相应译码模式来解码相应经分割亮度块(216)。

[0181] 视频解码器30可经进一步配置以解码指示与相应经分割亮度块相关联的相应译码模式待用于解码第一经分割色度块的第一语法元素,其中第一经分割色度块与两个或多于两个经分割亮度块对准(218)。视频解码器30可进一步根据两个或多于两个经分割亮度块的相应译码模式的函数而为第一经分割色度块确定色度译码模式(220),及根据经确定色度译码模式来解码第一经分割色度块(222)。

[0182] 在本发明的一个实例中,色度块是独立于亮度块被分割,使得至少一个经分割色度块不与单一经分割亮度块对准。

[0183] 在本发明的另一实例中,为了确定对应于相应经分割亮度块的相应译码模式,视频解码器30可经进一步配置以进行以下操作:接收对应于相应经分割亮度块的第二语法元素,第二语法元素指示相应译码模式;及解码对应于相应经分割亮度块的第二语法元素以确定相应译码模式。

[0184] 在本发明的另一实例中,为了确定对应于相应经分割亮度块的相应译码模式,视频解码器30经进一步配置以从相应经分割亮度块的一或多个代表性位置选择一或多个相应译码模式。在另一实例中,视频解码器30经配置以根据函数来选择一或多个相应译码模式。

[0185] 在另一实例中,一或多个代表性位置包含相应经分割亮度块的中心代表性位置,且其中为了根据函数而为第一经分割色度块确定色度译码模式,视频解码器30经进一步配置以获得经存储用于中心代表性位置的指示经确定相应译码模式的信息。

[0186] 在另一实例中,一或多个代表性位置包含相应经分割亮度块的拐角代表性位置,且其中为了根据函数而为第一经分割色度块确定色度译码模式,视频解码器30经进一步配置以获得经存储用于拐角代表性位置的指示经确定相应译码模式的信息。在一个实例中,一或多个代表性位置包括一或多个子块。

[0187] 在本发明的另一实例中,视频解码器30可经进一步配置以将相应经分割亮度块划分成相应子块,及将指示经确定相应译码模式的信息存储在与相应子块相关联的相应存储器位置中。

[0188] 在本发明的另一实例中,函数包含两个或多于两个经分割亮度块的一或多个相应子块的位置。在本发明的另一实例中,一或多个相应子块的位置为两个或多于两个经分割亮度块的中心子块,且其中为了根据函数而为第一经分割色度块确定色度译码模式,视频解码器30经配置以获得经存储用于中心子块的指示经确定相应译码模式的信息。

[0189] 在本发明的另一实例中,一或多个相应子块的位置为多于两个经分割亮度块的拐角子块,且其中为了根据函数而为第一经分割色度块确定色度译码模式,视频解码器30经配置以获得经存储用于拐角子块的指示经确定相应译码模式的信息。

[0190] 在本发明的另一实例中,函数包含与相应子块相关联的相应存储器位置中指示经确定相应译码模式的信息的统计分析。

[0191] 在本发明的另一实例中,为了根据函数而为第一经分割色度块确定色度译码模式,视频解码器30经进一步配置以使用梯度或高阶导数中的一者来分析存储在相应存储器位置中的信息。

[0192] 在本发明的另一实例中,信息包含以下各者中的一或多者:用于色度预测的直接模式的指示、预测方向、运动信息、用于位置相依帧内预测组合模式的旗标、用于位置相依帧内预测组合模式的一或多个参数、用于非可分离变换的一或多个第二变换集、增强型多重变换、自适应多重变换,或用于确定熵译码数据模型的一或多个上下文。

[0193] 在本发明的另一实例中,视频解码器30可经配置以接收指示函数的第三语法元素。

[0194] 在本发明的另一实例中,视频解码器30为无线通信装置的部分,无线通信装置进一步包括经配置以接收经编码视频数据的位流的接收器。在一个实例中,无线通信装置为移动站,且经编码视频数据的位流是由接收器接收且根据蜂窝式通信标准予以调制。

[0195] 图11为绘示根据本发明的技术的视频编码器22的实例操作的流程图。图12的技术可由视频编码器22的一或多个硬件结构执行。

[0196] 在本发明的一个实例中,视频编码器22可经配置以进行以下操作:接收视频数据块,视频数据块具有由宽度及高度界定的非方形形状(230);基于视频数据块的宽度或高度中的一或多者来确定一或多个PDPC参数(232);及使用PDPC模式及经确定PDPC参数来编码视频数据块(234)。如上文所论述,应理解,图11的技术可用于确定用于使用非方形块的任何预测模式的预测模式参数,所述预测模式包含根据样本位置将加权平均值应用于预测值及参考样本的预测模式。

[0197] 在一个实例中,一或多个PDPC参数包含一或多个水平相关PDPC参数及一或多个垂直相关PDPC参数,且其中为了确定一或多个PDPC参数,视频编码器22经进一步配置以基于视频数据块的宽度来确定一或多个水平相关PDPC参数,及基于视频数据块的高度来确定一多个垂直相关PDPC参数。

[0198] 在本发明的另一实例中,为了确定一或多个水平相关PDPC参数,视频编码器22经进一步配置以依据视频数据块的宽度来检索一或多个查找表的一或多个条目,且其中为了确定一或多个垂直相关PDPC参数,视频编码器22经进一步配置以依据视频数据块的高度来检索一或多个查找表的一或多个条目。

[0199] 在本发明的另一实例中,为了依据视频数据块的宽度来检索一或多个查找表的一或多个条目,视频编码器22经进一步配置以进行以下操作:基于视频数据块的宽度而在第一查找表中检索第一索引,第一索引指向第二查找表中的第一条目;及基于经检索第一索引而在第二查找表中检索一或多个水平相关PDPC参数。在另外实例中,为了依据视频数据块的高度来检索一或多个查找表的一或多个条目,视频编码器22经进一步配置以进行以下操作:基于视频数据块的高度而在第一查找表中检索第二索引,第二索引指向第二查找表中的第二条目;及基于经检索第二索引而在第二查找表中检索一或多个垂直相关PDPC参数。

[0200] 在本发明的另一实例中,一或多个PDPC参数包含不水平相关且不垂直相关的一或

多个非方向性PDPC参数，且其中为了确定一或多个PDPC参数，视频编码器22经进一步配置以基于视频数据块的宽度及高度的函数来确定一或多个非方向性PDPC参数。

[0201] 在本发明的另一实例中，函数为以下各者中的一或多者：视频数据块的宽度及高度的最小值、视频数据块的宽度及高度的最大值，或视频数据块的宽度及高度的加权平均值。在另外实例中，为了确定一或多个非方向性PDPC参数，视频编码器22经进一步配置以依据视频数据块的宽度及高度来存取一或多个查找表的一或多个条目。

[0202] 在本发明的另一实例中，视频编码器22包含在无线通信装置中，无线通信装置进一步包括经配置以发射经编码视频数据块的发射器。在另一实例中，无线通信装置为移动站，且经编码视频数据块是由发射器发射且根据蜂窝式通信标准予以调制。

[0203] 图12为绘示根据本发明的技术的视频解码器30的实例操作的流程图。图12的技术可由视频解码器30的一或多个硬件结构执行。

[0204] 在本发明的一个实例中，视频解码器30经配置以进行以下操作：接收使用PDPC模式而编码的视频数据块，视频数据块具有由宽度及高度界定的非方形形状(240)；基于视频数据块的宽度或高度中的一或多者来确定一或多个PDPC参数(242)；及使用PDPC模式及经确定PDPC参数来解码视频数据块(244)。如上文所论述，应理解，图12的技术可用于确定用于使用非方形块的任何预测模式的预测模式参数，所述预测模式包含根据样本位置将加权平均值应用于预测值及参考样本的预测模式。

[0205] 在本发明的一个实例中，一或多个PDPC参数包含一或多个水平相关PDPC参数及一或多个垂直相关PDPC参数，且其中为了确定一或多个PDPC参数，视频解码器30经进一步配置以基于视频数据块的宽度来确定一或多个水平相关PDPC参数，及基于视频数据块的高度来确定一或多个垂直相关PDPC参数。

[0206] 在本发明的另一实例中，为了确定一或多个水平相关PDPC参数，视频解码器30经进一步配置以依据视频数据块的宽度来检索一或多个查找表的一或多个条目，且其中为了确定一或多个垂直相关PDPC参数，视频解码器30经进一步配置以依据视频数据块的高度来检索一或多个查找表的一或多个条目。

[0207] 在本发明的另一实例中，为了依据视频数据块的宽度来检索一或多个查找表的一或多个条目，视频解码器30经进一步配置以进行以下操作：基于视频数据块的宽度而在第一查找表中检索第一索引，第一索引指向第二查找表中的第一条目；及基于经检索第一索引而在第二查找表中检索一或多个水平相关PDPC参数。在另外实例中，为了依据视频数据块的高度来检索一或多个查找表的一或多个条目，视频解码器30经进一步配置以进行以下操作：基于视频数据块的高度而在第一查找表中检索第二索引，第二索引指向第二查找表中的第二条目；及基于经检索第二索引而在第二查找表中检索一或多个垂直相关PDPC参数。

[0208] 在另一实例中，一或多个PDPC参数包含不水平相关且不垂直相关的一或多个非方向性PDPC参数，且其中为了确定一或多个PDPC参数，视频解码器30经进一步配置以基于视频数据块的宽度及高度的函数来确定一或多个非方向性PDPC参数。

[0209] 在另一实例中，函数为以下各者中的一或多者：视频数据块的宽度及高度的最小值、视频数据块的宽度及高度的最大值，或视频数据块的宽度及高度的加权平均值。

[0210] 在本发明的另一实例中，为了确定一或多个非方向性PDPC参数，视频解码器30经

进一步配置以依据视频数据块的宽度及高度来存取一或多个查找表的一或多个条目。

[0211] 在本发明的另一实例中，视频解码器30为无线通信装置的部分，无线通信装置进一步包括经配置以接收视频数据块的接收器。在另一实例中，无线通信装置为移动站，且视频数据块是由接收器接收且根据蜂窝式通信标准予以调制。

[0212] 已出于说明的目的而关于HEVC标准的扩展来描述本发明的某些方面。然而，本发明中所描述的技术可有用于其它视频译码过程，包含在开发中或尚未开发的其它标准或专有视频译码过程。

[0213] 如本发明中所描述的视频译码器可指视频编码器或视频解码器。相似地，视频译码单元可指视频编码器或视频解码器。同样地，在适用时，视频译码可指视频编码或视频解码。

[0214] 应认识到，取决于实例，本文中所描述的技术中的任一者的某些动作或事件可以不同序列被执行、可被添加、合并或完全省去(例如，并非所有所描述动作或事件对于所述技术的实践是必要的)。此外，在某些实例中，可同时地(例如，经由多线程处理、中断处理或多个处理器)而非依序地执行动作或事件。

[0215] 在一或多个实例中，所描述功能可以硬件、软件、固件或其任何组合予以实施。如果以软件予以实施，那么所述功能可作为一或多个指令或代码而存储在计算机可读媒体上或经由计算机可读媒体进行发射，且由基于硬件的处理单元执行。计算机可读媒体可包含：计算机可读存储媒体，其对应于有形媒体，诸如数据存储媒体；或通信媒体，其包含促进将计算机程序从一处传送到另一处(例如，根据通信协议)的任何媒体。以此方式，计算机可读媒体通常可对应于(1)为非暂时性的有形计算机可读存储媒体，或(2)诸如信号或载波的通信媒体。数据存储媒体可为可由一或多个计算机或一或多个处理器存取以检索指令、代码及/或数据结构以用于实施本发明中所描述的技术的任何可用媒体。计算机程序产品可包含计算机可读媒体。

[0216] 作为实例而非限制，此类计算机可读存储媒体可包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储装置、磁盘存储装置或其它磁性存储装置、闪速存储器，或可用以存储呈指令或数据结构形式的所要程序代码且可由计算机存取的任何其它媒体。此外，任何连接被适当地称为计算机可读媒体。举例来说，如果使用同轴电缆、光缆、双绞线、数字订户线(DSL)或无线技术(诸如红外线、无线电及微波)而从网站、服务器或其它远程源发射指令，那么同轴电缆、光缆、双绞线、DSL或无线技术(诸如红外线、无线电及微波)包含在媒体的定义中。然而，应理解，计算机可读存储媒体及数据存储媒体并不包含连接、载波、信号或其它暂时性媒体，而是涉及非暂时性有形存储媒体。如本文中所使用，磁盘及光盘包含压缩光盘(CD)、激光光盘、光学光盘、数字多功能光盘(DVD)、软盘及蓝光光盘，其中磁盘通常以磁性方式再现数据，而光盘运用激光以光学方式再现数据。以上各者的组合也应包含在计算机可读媒体的范围内。

[0217] 可由诸如一或多个数字信号处理器(DSP)、通用微处理器、专用集成电路(ASIC)、现场可编程逻辑阵列(FPGA)或其它等效集成或离散逻辑电路系统的一或多个处理器执行指令。因此，如本文中所使用的术语“处理器”可指上述结构或适合于实施本文中所描述的技术的任何其它结构中的任一者。另外，在一些方面中，本文中所描述的功能性可提供在经配置用于编码及解码的专用硬件及/或软件模块内，或结合在组合式编解码器中。此外，所

述技术可完全地实施在一或多个电路或逻辑元件中。

[0218] 本发明的技术可实施在多种装置或设备中,所述装置或设备包含无线手机、集成电路(IC)或IC集合(例如,芯片集)。在本发明中描述各种组件、模块或单元以强调经配置以执行所揭示技术的装置的功能方面,但未必要求由不同硬件单元来实现。更确切地说,如上文所描述,各种单元可组合在编解码器硬件单元中,或由互操作性硬件单元(包含如上文所描述的一或多个处理器)的集合结合合适的软件及/或固件而提供。

[0219] 已描述了各种实例。这些及其它实例在所附权利要求书的范围内。

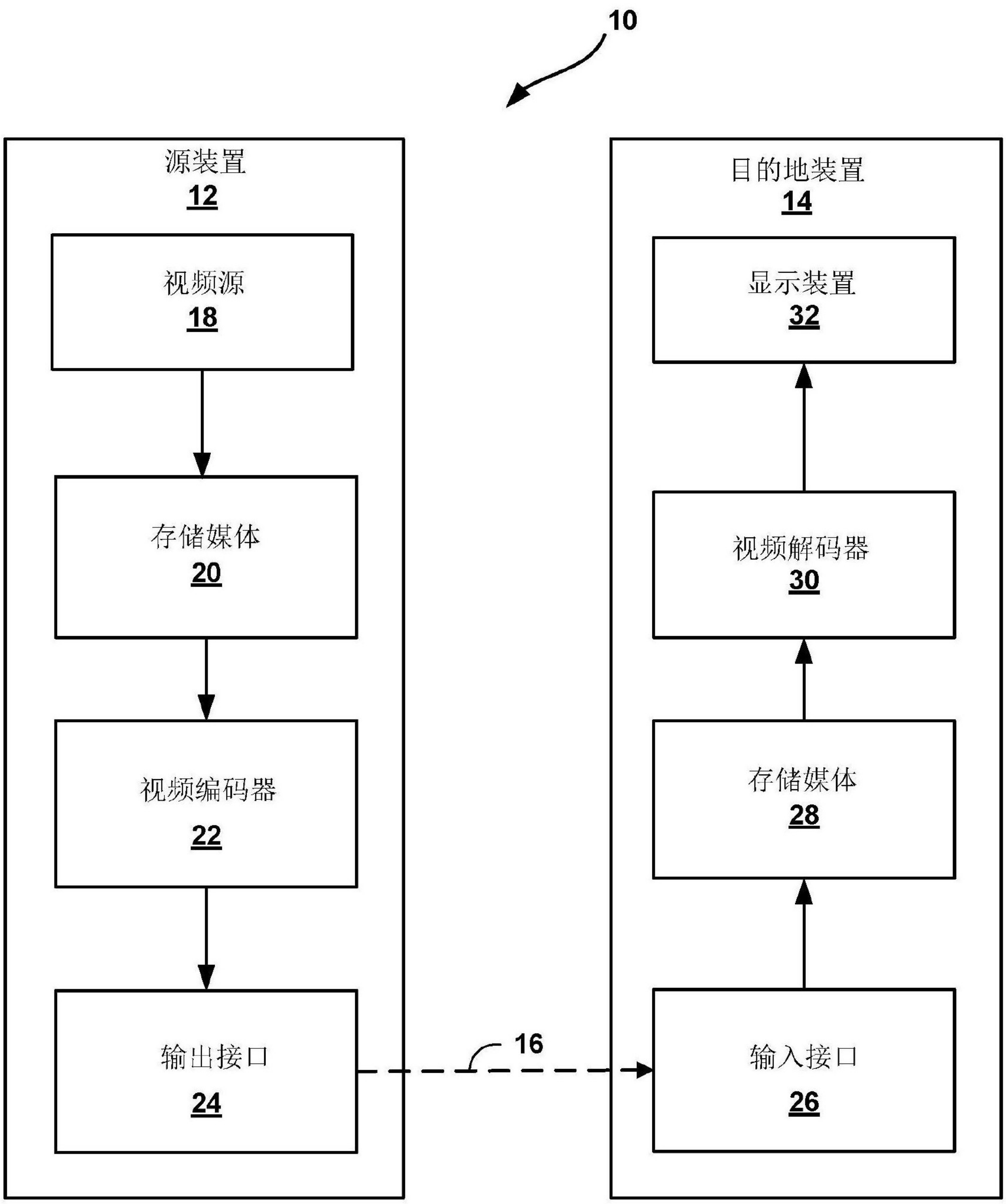

10
源装置
12
视频源
18
存储媒体
20
视频编码器
22
输出接口
24
16
目的地装置
14
显示装置
32
视频解码器
30
存储媒体
28
输入接口
26

图1

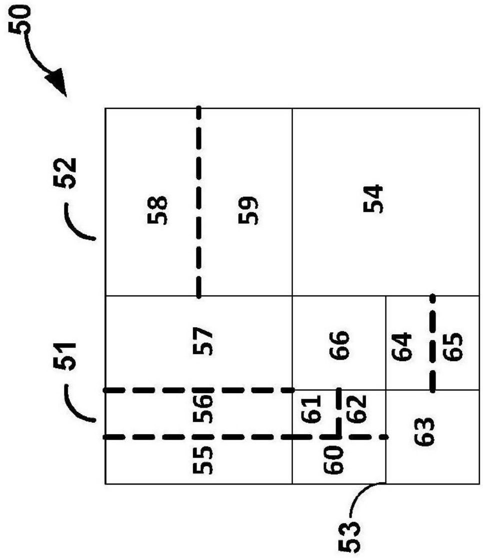


图2A

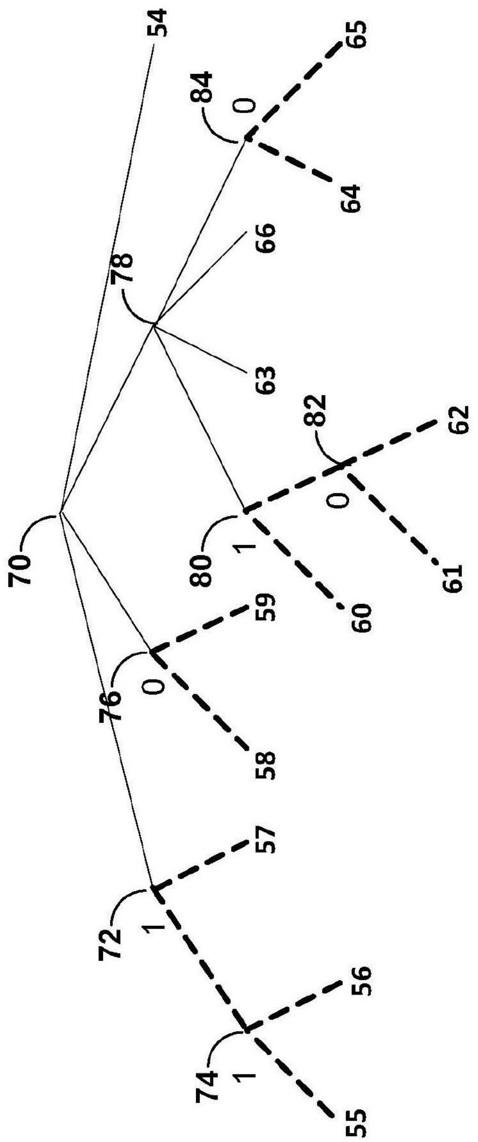


图2B

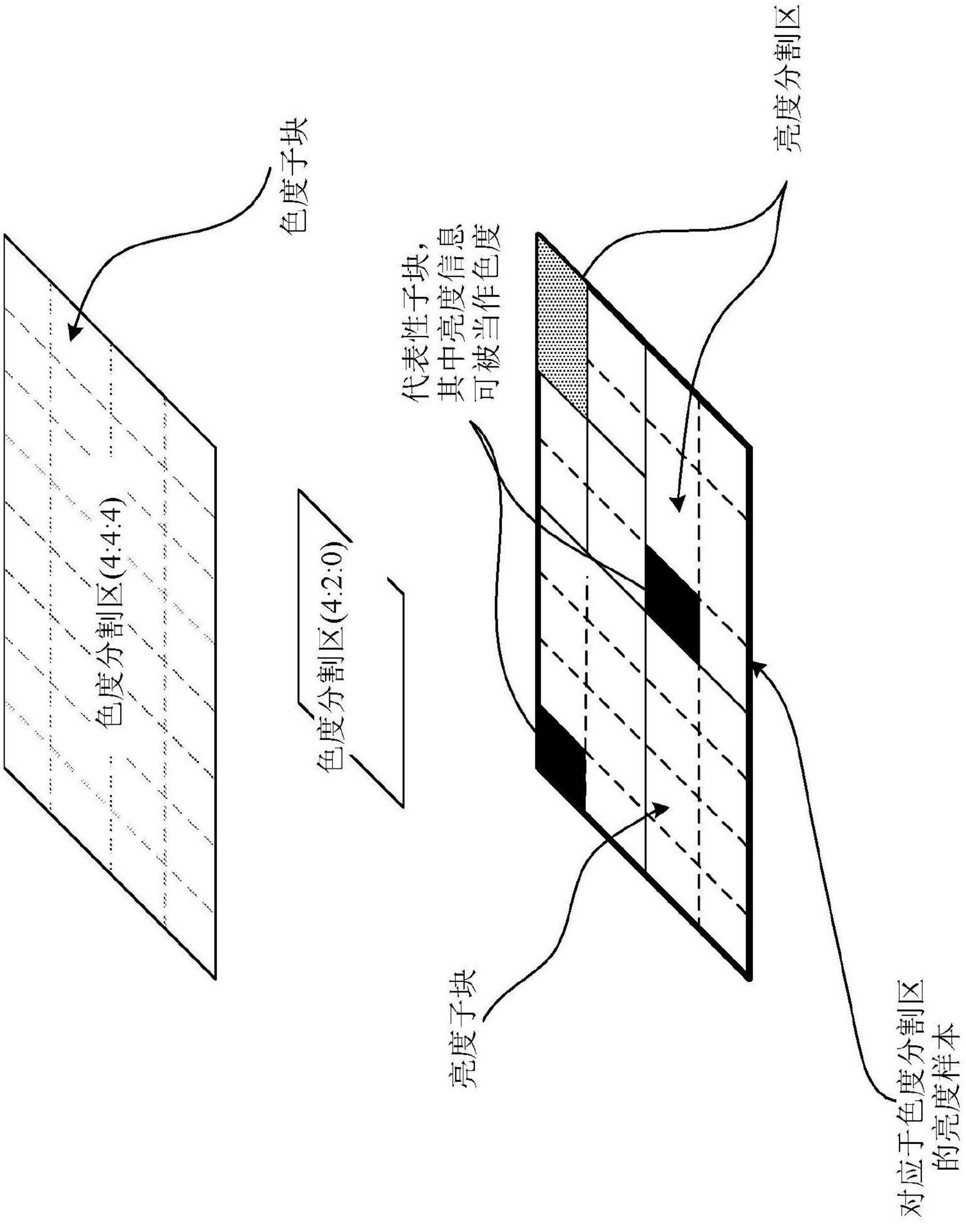

色度子块
色度分割区(4:4:4)
色度分割区(4:2:0)
代表性子块，其中亮度信息可被当作色度
亮度分割区
亮度子块
对应于色度分割区的亮度样本

图3

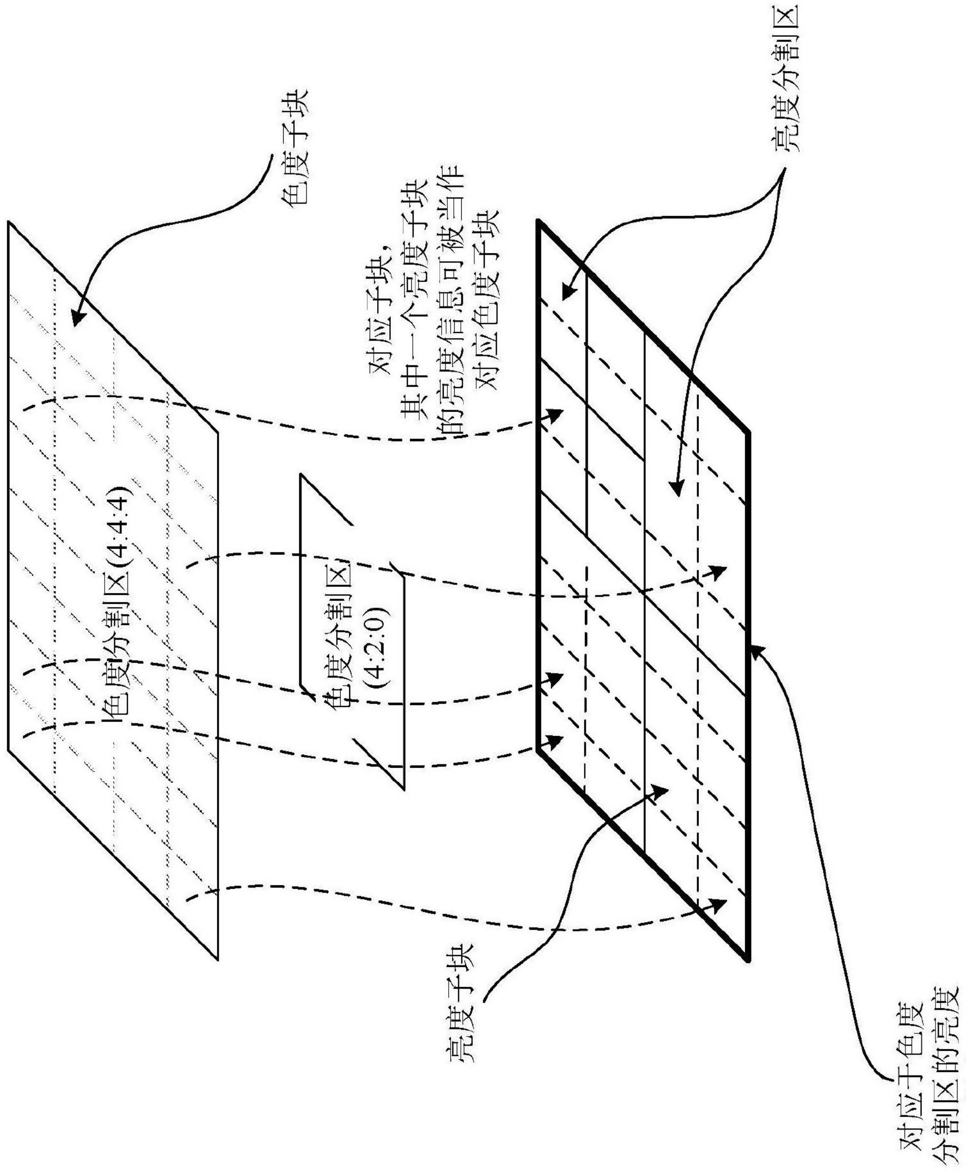

色度子块
亮度分割区
对应子块，
其中一个亮度子块
的亮度信息可被当作
对应色度子块
色度分割区(4:4:4)
色度分割区
(4:2:0)
亮度子块
对应于色度
分割区的亮度
样本

图4

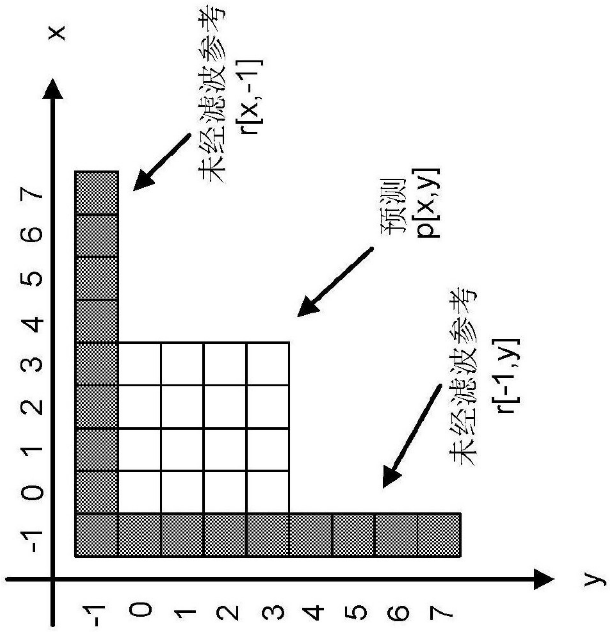

x
y
-1 0 1 2 3 4 5 6 7
-1 0 1 2 3 4 5 6 7
未经滤波参考
r[x,-1]
预测
p[x,y]
未经滤波参考
r[-1,y]

图5A

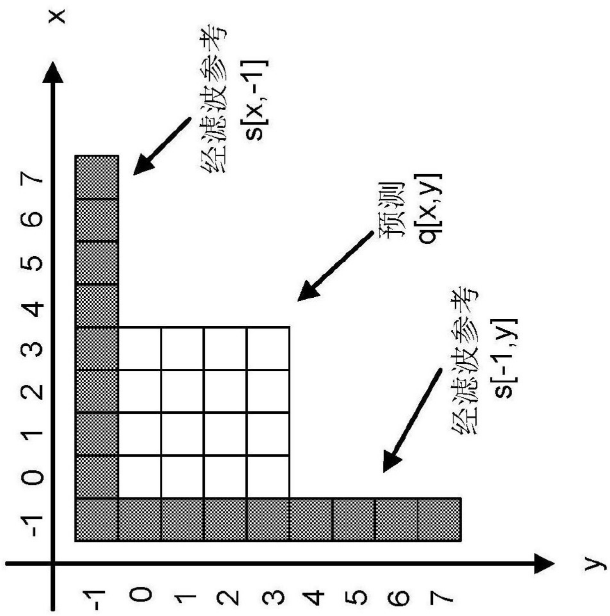

x
y
-1 0 1 2 3 4 5 6 7
-1 0 1 2 3 4 5 6 7
经滤波参考
s[x,-1]
预测
q[x,y]
经滤波参考
s[-1,y]

图5B

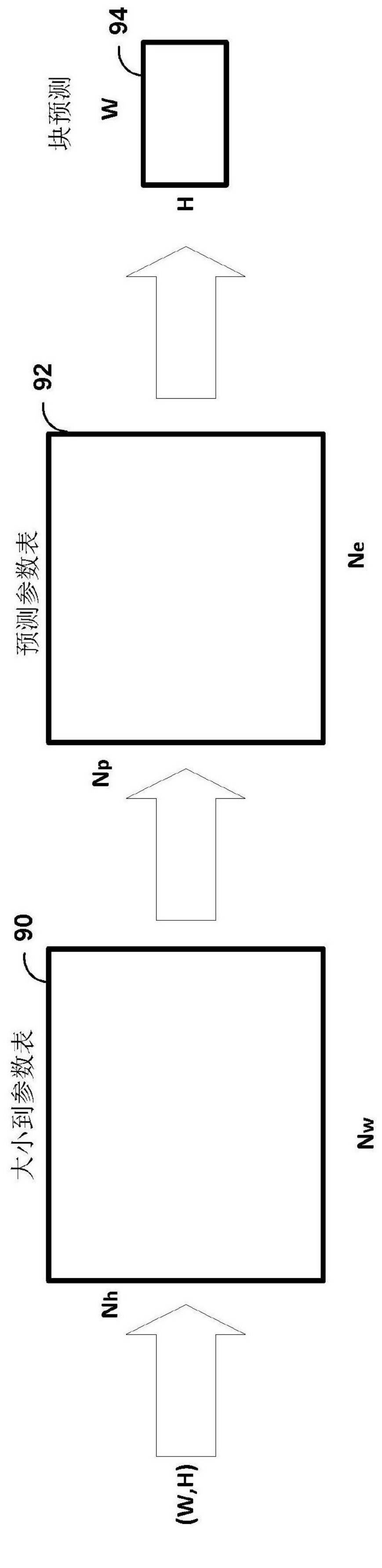


图6

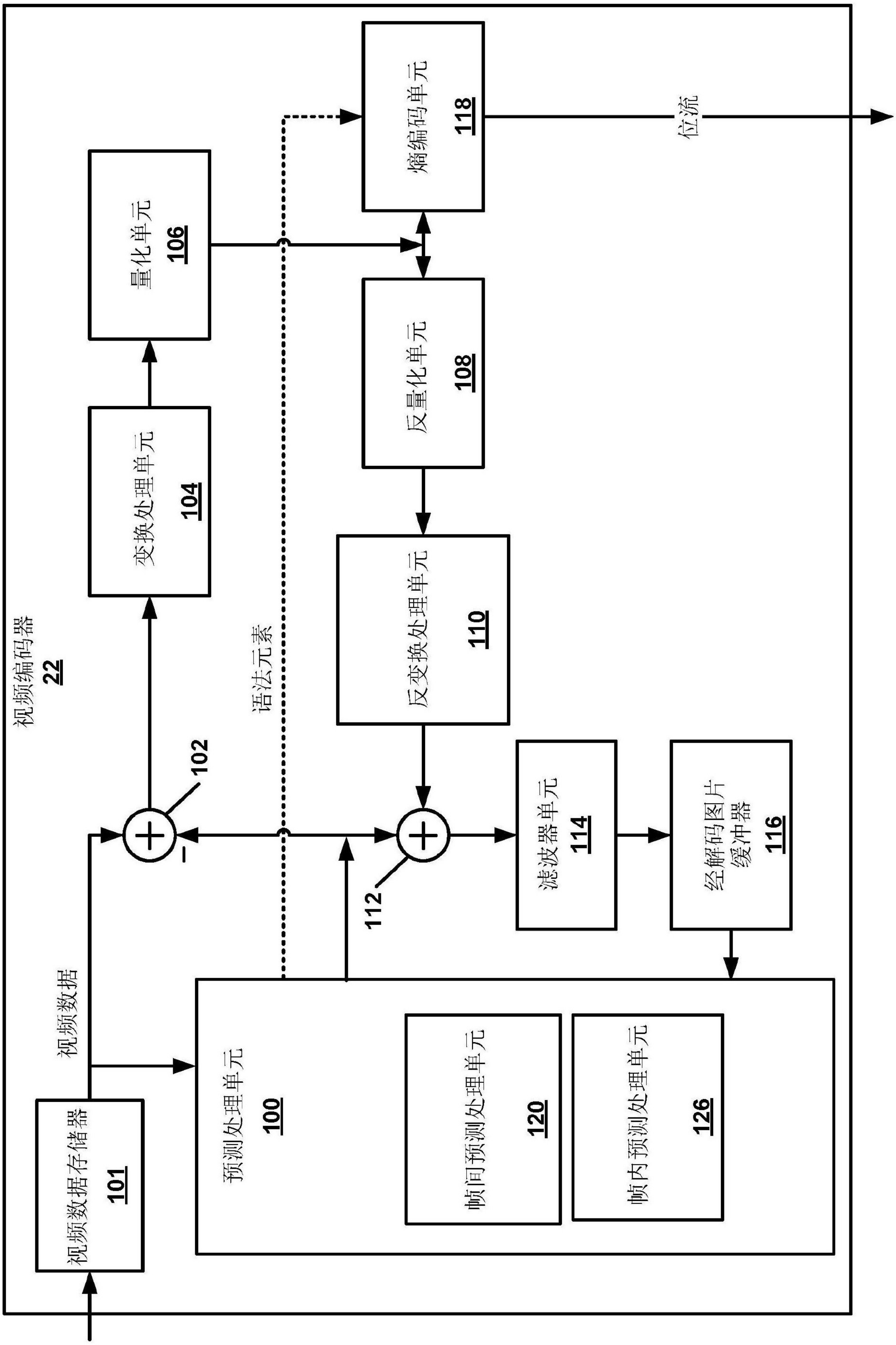

视频编码器
22
视频数据存储器
101
视频数据
预测处理单元
100
帧间预测处理单元
120
帧内预测处理单元
126
102
−
112
变换处理单元
104
量化单元
106
语法元素
熵编码单元
118
位流
反量化单元
108
反变换处理单元
110
滤波器单元
114
经解码图片
缓冲器
116

图7

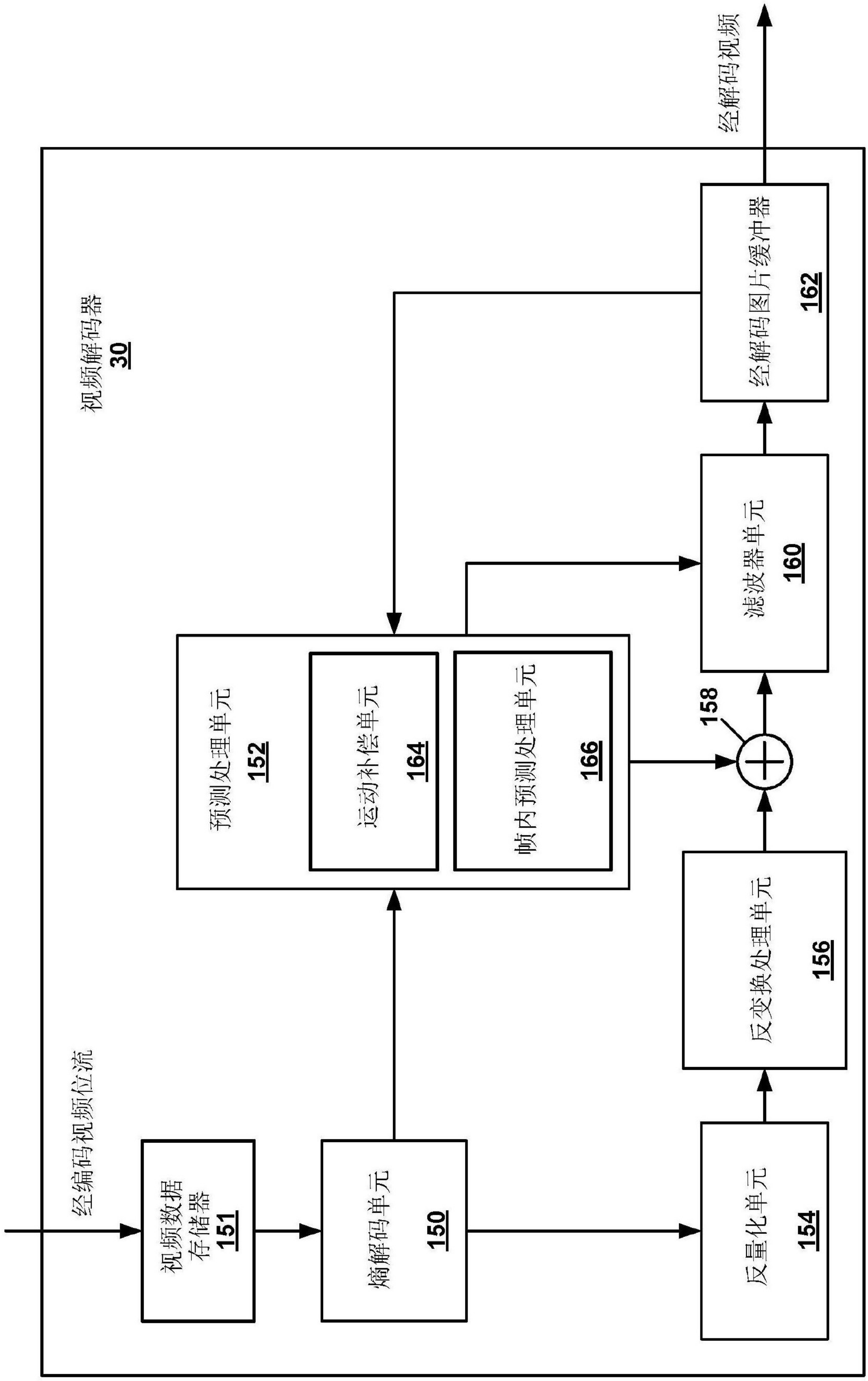

视频解码器
30
经编码视频位流
视频数据存储器
151
熵解码单元
150
预测处理单元
152
运动补偿单元
164
帧内预测处理单元
166
反量化单元
154
反变换处理单元
156
158
滤波器单元
160
经解码图片缓冲器
162
经解码视频

图8

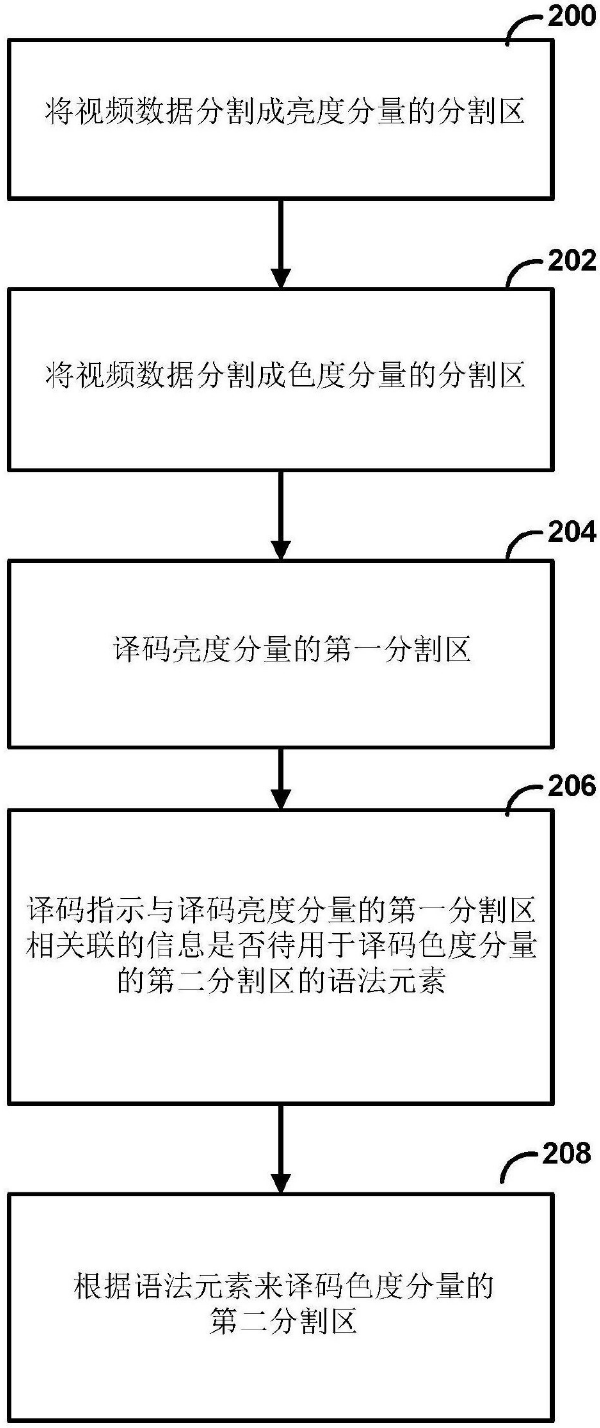

200
将视频数据分割成亮度分量的分割区
202
将视频数据分割成色度分量的分割区
204
译码亮度分量的第一分割区
206
译码指示与译码亮度分量的第一分割区相关联的信息是否待用于译码色度分量的第二分割区的语法元素
208
根据语法元素来译码色度分量的第二分割区

图9

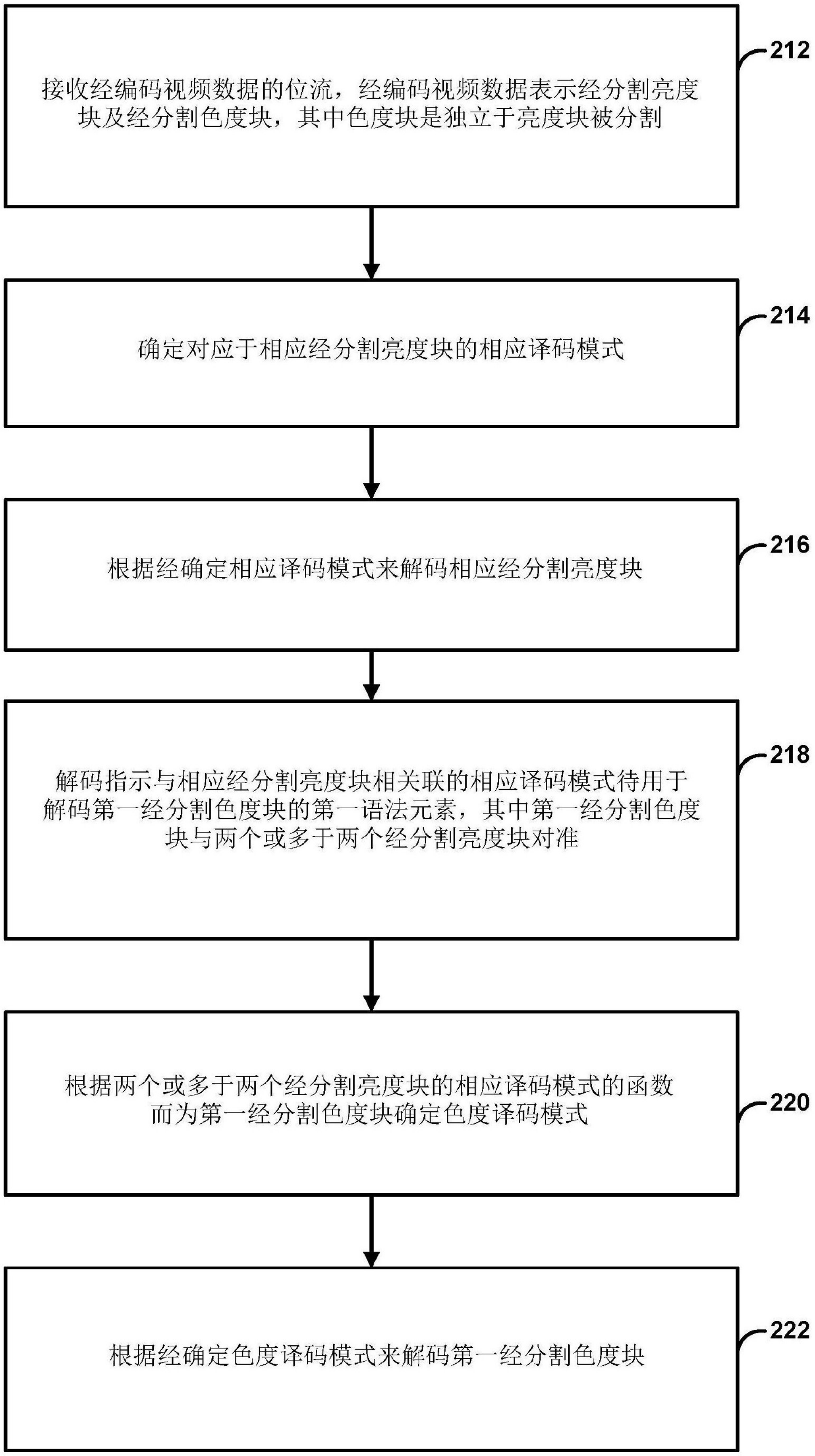

212
接收经编码视频数据的位流，经编码视频数据表示经分割亮度块及经分割色度块，其中色度块是独立于亮度块被分割
214
确定对应于相应经分割亮度块的相应译码模式
216
根据经确定相应译码模式来解码相应经分割亮度块
218
解码指示与相应经分割亮度块相关联的相应译码模式待用于解码第一经分割色度块的第一语法元素，其中第一经分割色度块与两个或多于两个经分割亮度块对准
220
根据两个或多于两个经分割亮度块的相应译码模式的函数而为第一经分割色度块确定色度译码模式
222
根据经确定色度译码模式来解码第一经分割色度块

图10

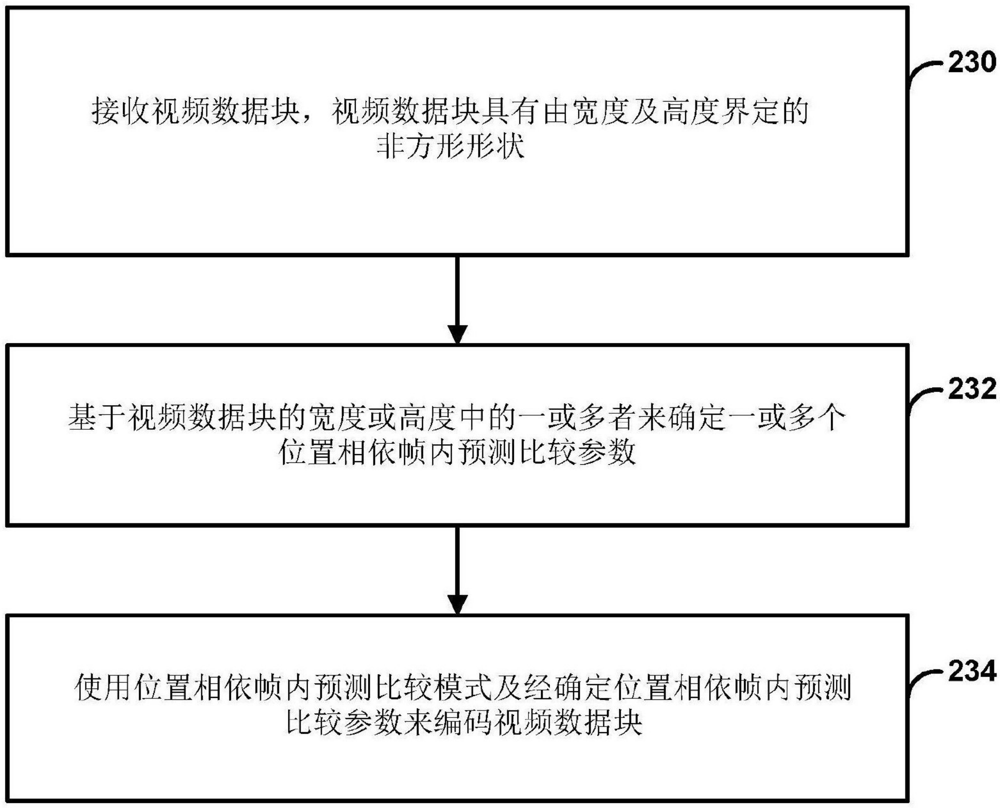

接收视频数据块，视频数据块具有由宽度及高度界定的非方形形状
230
基于视频数据块的宽度或高度中的一或多者来确定一或多个位置相依帧内预测比较参数
232
使用位置相依帧内预测比较模式及经确定位置相依帧内预测比较参数来编码视频数据块
234

图11

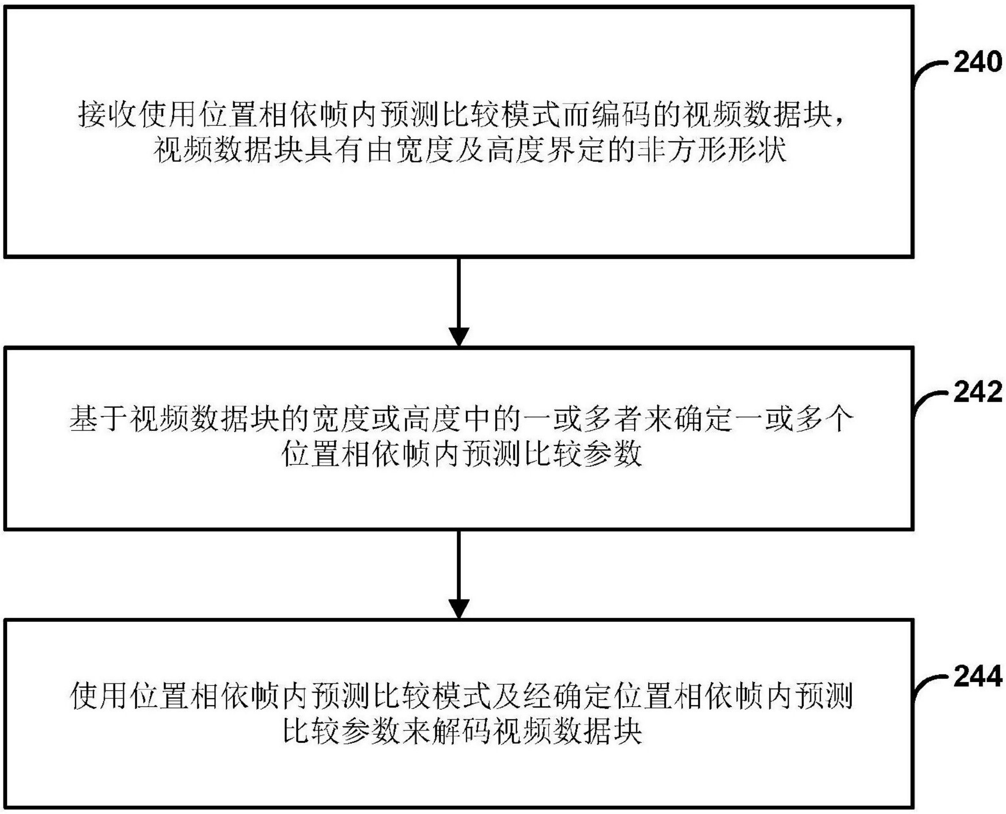

接收使用位置相依帧内预测比较模式而编码的视频数据块，视频数据块具有由宽度及高度界定的非方形形状
240
基于视频数据块的宽度或高度中的一或多者来确定一或多个位置相依帧内预测比较参数
242
使用位置相依帧内预测比较模式及经确定位置相依帧内预测比较参数来解码视频数据块
244

图12